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	January 5, 2011	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
0	05/01/2011	Formatting example	B.Isnard

Contents

Chapter 1

Document presentation

1.1 Document objectives

This document presents the detailed specifications of the Vishnu C++ API.

These specifications include the definition of all methods and all data types provided by the API.

1.2 Document structure

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

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

Each module corresponds to a chapter in the document, and each chapter contains the following sections:

- A first section describing the definition of all methods provided by the library
- A second section describing the definition of all data types provided by the library
- TODO - A third section describing the dependencies of the library

1.3 References

TODO - ajouter ref. vers les specs générales

1.4 Glossary

Chapter 2

API specification for User Management System (UMS)

2.1 Definition of the functions of the API

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
ClosePolicy	SessionCloseType	It allows the user to choose the way for closing the session automatically on TIMEOUT or on DISCONNECT	IN	no
options	ConnectOptions	options is an object which encapsulates the options available for the connect method. It allows the user to set the TIMEOUT when the CLOSE_ON_TIMEOUT option is chosen or for an admin to open a session as he/she was a specific user	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 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
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 TIMEOUT treshold)
UNKNOWN_LOGIN	The login is unknown
NO_ADMIN	The user is not an administrator
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect

Name	Description
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **connect**(const string& login, const string& password, const SessionCloseType& ClosePolicy, 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
sessionId	string	sessionId is the identifier of the session defined 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
UNKNOWN_SESSION_ID	The session Id is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.3 createUser

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
newUser	User	newUser is an object which encapsulates the new user information	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
USERID_REQUIRED	The userId must to be defined
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.4 updateUser

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
options	UpdateUserOptions	options is an object which encapsulates user information updated	IN	no

Description

The updateUser() function updates the user information in VISHNU except login and 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
UNKNOWN_LOGIN	The login is unknown
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.
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **updateUser**(const string& sessionKey, const UpdateUserOptions& options=UpdateUserOptions());

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 removes a 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
UNKNOWN_LOGIN	The login is unknown
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_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

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 a user 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_ERROR	A problem occurs with the database

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
passwordNew	string	passwordNew 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_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.8 resetPassword

Parameters

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
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 an admin to reset the password of a user identified by his/her userId

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_LOGIN	The login is unknown
INCORRECT_LOGIN_SIZE	The size of the login is incorrect

Name	Description
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_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.9 createLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
newAccount	LocalAccount	newAccount is the object which encapsulates the new local user configuration	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
USERID_REQUIRED	The userId must to be defined
MACHINEID_REQUIRED	The machineId must to be defined
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_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.10 updateLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
options	LocalAccount	options is an object which encapsulates the change of the local user configuration except the machineId and the userId	IN	no
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The updateLocalAccount() function allows the user to update his/her local user configuration 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.
UNKNOWN_LOCAL_ACCOUNT	The local configuration for the given user on the given machine is unknown
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **updateLocalAccount**(const LocalAccount& options, const string& sessionKey);

2.1.11 deleteLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
userId	string	userId represents the login of the user which will be deleted according to a specific machine	IN	yes
machineId	string	machineId represents the identifier of the machine which will be deleted according to a specific user	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

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

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_LOCAL_ACCOUNT	The local configuration for the given user on the given machine is unknown
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **deleteLocalAccount**(const string& userId, const string& machineId, const string& sessionKey);

2.1.12 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 an object which encapsulates the configuration description	OUT	yes

Description

The saveConfiguration() function allows an admin to save the configuration of 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
SAVE_CONFIG_ERROR	A problem occurs during the configuration saving
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.13 restoreConfiguration

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	IN	yes

Description

The restoreConfiguration() function allows to restores the configuration of 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
RESTORE_CONFIG_ERROR	A problem occurs during the configuration restoring
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.14 createMachine

Parameters

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

Description

The createMachine() function adds a new machine 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
MACHINE_EXISTING	The machineId already exists in the database
MACHINEID_REQUIRED	The machineId must to be defined
NO_ADMIN	The user is not an administrator
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **createMachine**(const string& sessionKey, const Machine& newMachine);

2.1.15 deleteMachine

Parameters

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

Description

The deleteMachine() function removes a machine 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
NO_ADMIN	The user is not an administrator
UNKNOWN_MACHINE	The machineId is unknown
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **deleteMachine**(const string& machineId, const string& sessionKey);

2.1.16 listLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
machineIdOpt	string	allows a user to list a local user configuration on a specific machine	IN	no
adminOption	AdminOptions	allows an admin to list all local config of all users in the database	IN	no
listLocalAcct	ListLocalAccounts	listLocalAccount is the list of the local user configuations	OUT	yes

Description

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

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_LOGIN	The login is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **listLocalAccount**(const string& sessionKey, const string& machineIdOpt, const AdminOptions& adminOption=AdminOptions(), ListLocalAccounts& listLocalAcct);

2.1.17 listMachine

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
machineIdOpt	string	allows a user to list one machine identified by its machineId	IN	no
adminOption	AdminOptions	allows a user to list all machines in the database	IN	no
listMachine	ListMachines	listLocalAccount is the list of the local configs	OUT	yes

Description

The listMachine() function allows the user to get an object which encapsulates the list of the machine objects 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
UNKNOWN_LOGIN	The login is unknown
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
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **listMachine**(const string& sessionKey, const string& machineIdOpt, const AdminOptions& adminOption=AdminOptions(), ListMachines& listMachine);

2.1.18 listHistoryCmd

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
startDateOptions	long	allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used).	IN	no
endDateOptions	long	allows the user to organize the commands listed by providing the end date (the timestamp of the end date is used). By default, the end date is the current day.	IN	no
sessionIdOpt	string	allows the user to list all commands launched in the session identifier by a session Id	IN	no
adminOption	AdminOptions	allows the users	IN	no
listCommands	ListCommands	listCommands is the list of commands	OUT	yes

Description

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

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_LOGIN	The login is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **listHistoryCmd**(const string& sessionKey, const long& startDateOptions, const long& endDateOptions, const string& sessionIdOpt, const AdminOptions& adminOption=AdminOptions(), ListCommands& listCommands);

2.1.19 listOptions

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
adminOption	AdminOptions	allows an admin to list all options of all users in VIHSNU	IN	no
optionName	string	represents the name on an option. It allows the user to list a specific option whose name is optionName	IN	no
listOptValues	ListOptionsValues	listOptions is the list of options	OUT	yes

Description

The listOptions() function allows the user to get an object which encapsulates the list of the 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
NO_ADMIN	The user is not an administrator
UNKNOWN_LOGIN	The login is unknown
UNKNOWN_OPTION	the name of the option is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **listOptions**(const string& sessionKey, const AdminOptions& adminOption=AdminOptions(), const string& optionName, ListOptionsValues& listOptValues=ListOptionsValues());

2.1.20 listUsers

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
userIdOption	string	allows an admin to list one user identified by his/her userId	IN	no
listuser	ListUsers	listuser is the list of users	OUT	yes

Description

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

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

Name	Description
UNKNOWN_LOGIN	The login is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

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

2.1.21 listSessions

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
options	ListSessionOption	allows the user to list all sessions or only inactives. By default, only actives sessions are listed.	IN	no
startDateOption	long	allows the user to organize the sessions listed by providing the start date (the UNIX timestamp of the start Date is used).	IN	no
endDateOption	long	allows the user to organize the sessions listed by providing the end date (the UNIX timestamp of the end Date is used). By default, the end date is the current day.	IN	no
adminOption	AdminOptions	allows an admin to list all sessions of all users in the database or sessions from a specific user by providing his/her userId	IN	no
sessionIdOpt	string	allows the user to list one session by providing its session identifier	IN	no
listsession	ListSessions	listsession is the list of sessions	OUT	yes

Description

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

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_LOGIN	The login is unknown
UNKNOWN_SESSION_OPTION	the name of the session option is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
INCORRECT_DATE_OPTION	The date option is incorrect
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **listSessions**(const string& sessionKey, const ListSessionOption& options, const long& startDateOption, const long& endDa-

teOption, const AdminOptions& adminOption=AdminOptions(), const string& sessionIdOpt, ListSessions& listsession);

2.1.22 configureOption

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
optionValue	OptionValue	The optionValue is an object which encapsulates the option information	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_OPTION	the name of the option is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_ERROR	A problem occurs with the database
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **configureOption**(const string& sessionKey, const OptionValue& optionValue);

2.1.23 configureDefaultOption

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
optionValue	OptionValue	The optionValue is an object which encapsulates the option information	IN	yes

Description

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

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_OPTION	the name of the option is unknown
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_ERROR	A problem occurs with the database

Name	Description
UMS_SERVER_NOT_AVAILABLE	The server UMS is not available

Signature

int **configureDefaultOption**(const string& sessionKey, const OptionValue& optionValue);

2.1.24 vishnuInitialize

Parameters

Parameter	Type	Description	Mode	Required
configPath	string	configPath is the path of VISHNU configuration file	IN	yes

Description

The vishnuInitialize() function allows the user to initialize 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
INCORRECT_CFG_PATH	The configuration file path is incorrect

Signature

int **vishnuInitialize**(const string& configPath);

2.1.25 vishnuFinalize

Parameters

Parameter	Type	Description	Mode	Required

Description

The vishnuFinalize() function allows the user to go out properly 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
VISHNU_FINALIZE_ERROR	An error occurs during vishnuFinalize

Signature

int **vishnuFinalize**

2.2 Class definitions

SessionCloseType Enumeration Type

Name	Value
CLOSE_ON_DISCONNECT	0
CLOSE_ON_TIMEOUT	1

ConnectOptions Class Content

Name	Type	Description
sessionInactivityDelay	int	The sessionInactivityDelay is the maximum delay between two API commands when the CLOSE_ON_TIMEOUT option is set
substituteUserId	string	is an option which allows an admin to open a session as he/she was a specific user identified by his/her userId

User Class Content

Name	Type	Description
userId	string	represents the login of the user
password	string	is the password of the user. At the beginning, an admin can give a temporary password or it is automatically generated by the System.
firstname	string	is the firstname of the user
lastname	string	is the lastname of the user
privilege	int	is the privilege of the user (admin or simple user)
email	string	is the email of the user
passwordState	int	is the state of the password which allows the System to inform the user to change his/her password

UpdateUserOptions Class Content

Name	Type	Description
userId	string	represents the login of the user
firstname	string	is an option to update the firstname of the user
lastname	string	is an option to update the lastname of the user
privilege	string	is an option to update the privilege of the user
email	string	is an option to update the email adress of the user

LocalAccount Class Content

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

Configuration Class Content

Name	Type	Description
descConf	string	represents the description of the configuration
listConfUsers	List of User	is the list of users objects
listConfMachines	List of Machine	is a list of machines objects
listConfLocalAccounts	List of LocalAccount	is the list of LocalAccount objects

Machine Class Content

Name	Type	Description
machineId	string	represents the identifier of the machine
name	string	represents the name of the machine
site	string	represents the location of the machine
totalDiskSpace	string	represents the total disk space of the machine
totalMemory	string	represents the total memory of the machine
machineDescription	string	represents the description of the machine

AdminOptions Class Content

Name	Type	Description
AdminListOption	AdminListOptionType	is an admin option for listing all information in the database during a list API commands
userId	string	is an admin option for listing information about a specific user identified by his/her userId

AdminListOptionType Enumeration Type

Name	Value
ALL	0

ListLocalAccount Class Content

Name	Type	Description
accounts	List of LocalAccount	is a list of LocalAccount objects which encapsulates local user configurations

ListMachines Class Content

Name	Type	Description
machines	List of Machine	is a list of machines objects which encapsulates the machines information

ListCommands Class Content

Name	Type	Description
Commands	List of Command	is the list of commands objects

Command Class Content

Name	Type	Description
commandId	string	is the identifier of a command
sessionId	string	The sessionId is the identifier of the session define in the database

Name	Type	Description
machineId	string	The machineId is the identifier of the machine used by the command
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

ListOptionsValue Class Content

Name	Type	Description
optionValues	List of OptionValue	is a list of optionValue objects which encapsulates the optionValue information

OptionValue Class Content

Name	Type	Description
optionName	string	represents the name of an option
value	string	represents the value of an option

ListUsers Class Content

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

ListSessionOption Class Content

Name	Type	Description
sessionListOption	SessionListOptionType	is an option which precise the type of session listed (ACTIVE, INACTIVE or ALL)
sessionClosePolicy	SessionCloseType	is an option for closing session automatically
sessionInactivityDelay	int	The sessionInactivityDelay is the maximum delay between two API commands when the CLOSE_ON_TIMEOUT option is set
machineId	string	represents the identifier of the machine

SessionListOptionType Enumeration Type

Name	Value
ACTIVE_SESSION	0
INACTIVE_SESSION	1
ALL_SESSION	2

ListSessions Class Content

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

Session Class Content

Name	Type	Description
userId	string	represents the login of the user
sessionKey	string	is the key of the session generated by VISHNU

Name	Type	Description
dateLastConnect	int	is the date of the last connection to the session
dateCreation	int	is the date of the first connection to the session
state	int	is the state of the session (ACTIVE OR INACTIVE)
closePolicy	SessionCloseType	is the way to close the session
timeoutDate	long	is the maximum delay between two API commands when the CLOSE_ON_TIMEOUT option is set (the UNIX timestamps is used)

Chapter 3

API specification for Tasks Management System (TMS)

3.1 Definition of the functions of the API

3.1.1 submitJob

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
machineId	string	Is the id of the machine where the job must be submitted.	IN	yes
scriptFilePath	string	The path to the file containing the characteristics (job command, and batch scheduler directive required or optional) of the job to submit.	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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, const string& scriptFilePath, string& jobId, string& jobPath, const SubmitOptions& options=SubmitOptions());

3.1.2 listJobs

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, ListJobs& listOfJobs, const ListJobsOptions& options=ListJobsOptions

3.1.3 getJobInfo

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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_RESPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_SESSION_KEY	The session key is not valid to perform the service.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, const string& jobId, Job& jobInfos);

3.1.4 cancelJob

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, const string& jobId, string& infoMsg);

3.1.5 getJobOutPut

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_OK	The service was performed successfully.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, const string& jobId, string& outputPath, string& errorPath);

3.1.6 getAllJobsOutPut

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
machineId	string	Machine hash key	IN	yes
listOfResults	ListJobResults	Is the list of jobs results	OUT	yes

Description

The getAllJobsOutPut() 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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_OK	The service was performed successfully.
TMS_PERMISSION_DENIED	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 **getAllJobsOutPut**(const string& sessionKey, const string& machineId, ListJobResults& listOfResults);

3.1.7 listQueues

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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_RESPONSE	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_KEY	The session key is not valid to perform the service.
TMS_PERMISSION_DENIED	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& sessionKey, const string& machineId, ListQueues& listofQueues);

3.1.8 setMachineEnv

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
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& sessionKey, const string& machineId, const string& listEnv);

3.1.9 setMachineRefreshPeriod

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
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& sessionKey, const string& machineId, const int& value);

3.1.10 getJobProgress

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	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 getJobProgress() function get 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_KEY	The session key 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 **getJobProgress**(const string& sessionKey, const string& machineId, Progression& progress, const ProgressOptions& options=ProgressOptions());

3.2 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.

Name	Type	Description
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
sessionId	string	Is the id of the session that contained the job submission command
submitMachineId	string	Is the id of the machine on which the job has been submitted.
submitMachineName	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 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.
status	JobStatus	The current status of the job.
submitDate	long	Date and time when job was submitted (unix timestamp)
endDate	long	Represents the execution end date of the job (unix timestamp)
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 (in seconds)
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	long	List jobs submitted after this date (unix timestamp).
toSubmitDate	long	List jobs submitted before this date (unix timestamp)
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 JobResult	Represents the list of completed jobs results.

JobResult 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 Queue	Represents the list of queues.

Queue 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.

Name	Type	Description
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	long	Start date and time of the job (unix timestamp)
endTime	long	End date and time of the job (unix timestamp)
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 Definition of the functions of the API

4.1.1 getUpdateFrequency

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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

Signature

int **getUpdateFrequency**(const string& sessionKey, int& freq);

4.1.2 export

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command (current session key)	IN	yes
oldSessionKey	string	The key of the session to export (session has ended)	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

Signature

int **export**(const string& sessionKey, const string& oldSessionKey, const ExportType& exportType, string& filename);

4.1.3 replay

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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

Signature

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

4.1.4 getMetricVal

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
machineId	string	The id of the machine	IN	yes
startTime	long	The start time to look for the metric	IN	yes
endTime	long	The end time to search	IN	yes
metricType	MetricType	The metric types	IN	yes
res	ListMetric	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

Signature

int **getMetricVal**(const string& sessionKey, const string& machineId, const long& startTime, const long& endTime, const MetricType& metricType, ListMetric& res);

4.1.5 getCurrentData

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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 data corresponding to the datatype 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

Signature

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

4.1.6 getProcesses

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
machineId	string	The id of the machine the user wants the running processes	IN	yes
process	ListProcesses	The list of the processes 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

Signature

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

4.1.7 setSystemThreshold

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
machineId	string	The id of the machine the user wants to set the treshold over	IN	yes
tresholdType	MetricType	The type of the metric to set	IN	yes
value	double	The treshold value	IN	yes

Description

The setSystemThreshold() function sets a threshold 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

Signature

int **setSystemThreshold**(const string& sessionKey, const string& machineId, const MetricType& tresholdType, const double& value);

4.1.8 getSystemThreshold

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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 getSystemThreshold() function gets a System threshold 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

Signature

int **getSystemThreshold**(const string& sessionKey, const string& machineId, const MetricType& type, double& value);

4.1.9 defineUserIdentifier

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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

Signature

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

4.1.10 defineMachineIdentifier

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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

Signature

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

4.1.11 defineJobIdentifier

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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

Signature

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

4.1.12 defineTransferIdentifier

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
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& sessionKey, const string& format);

4.1.13 loadShed

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
machineId	string	The id of the machine to stop	IN	yes
loadShedType	LoadShedType	Type of load shedding	IN	yes

Description

The loadShed() function loads shed a machine, 2 modes are available, soft 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 **loadShed**(const string& sessionKey, const string& machineId, const LoadShedType& loadShedType);

4.1.14 setUpdateFrequency

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
freq	int	Frequency the data are updated, in second	IN	yes

Description

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

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 **setUpdateFrequency**(const string& sessionKey, 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
sessionKey	string	The key of the session of the user that submits the command	IN	yes
type	RestartType	The type of restart desired	IN	yes
componentName	string	If not restarting all, the name of the component to restart	IN	no

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

Signature

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

4.1.17 updateMachine

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The key of the session of the user that submits the command	IN	yes
machineId	string	The id of the machine to update	IN	yes
CompoMachine	CompoMachine	A description of the type of component to change	IN	yes
val	object	The value to set	IN	yes
lang	string	For machine description, the language used	IN	no

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
SUCCESS	Error code returned if success

Signature

int **updateMachine**(const string& sessionKey, const string& machineId, const CompoMachine& CompoMachine, const object& val, const string& lang);

4.2 Class definitions

ExportType Enumeration Type

Name	Value
PYTHON	0
SHELL	1

MetricType Enumeration Type

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

ListMetric Class Content

Name	Type	Description
metric	List of Metric	The metrics of the list

Metric Class Content

Name	Type	Description
type	MetricType	The type of the metric
value	double	The value of the metric
time	int	The timestamp the metric had the value

ListProcesses Class Content

Name	Type	Description
processName	List of string	The processes of the list

LoadShedType Enumeration Type

Name	Value
SOFT	0
HARD	1

RestartType Enumeration Type

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

CompoMachine Enumeration Type

Name	Value
DESC	0
DISKSPACE	1
RAM	2