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

Table of Contents

Web Service: VishnuUMSService

Web Service: VishnuUMSService

Target Namespace:
urn:ResourceProxy

Port VishnuUMSPort Port typeSource code

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

1. *addAuthAccount*DetailSource code
2. *addLocalAccount*DetailSource code
3. *changePassword*DetailSource code
4. *close*DetailSource code
5. *configureOption*DetailSource code
6. *connect*DetailSource code
7. *deleteAuthAccount*DetailSource code
8. *deleteLocalAccount*DetailSource code
9. *listAuthAccounts*DetailSource code
10. *listAuthSystems*DetailSource code
11. *listHistoryCmd*DetailSource code
12. *listLocalAccounts*DetailSource code
13. *listMachines*DetailSource code
14. *listOptions*DetailSource code
15. *listSessions*DetailSource code
16. *reconnect*DetailSource code
17. *updateAuthAccount*DetailSource code
18. *updateLocalAccount*DetailSource code

Operations

Port type VishnuUMSPortTypeSource code

1. ### addAuthAccount

Source code
Description:
adds a new local user-authentication configuration
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
addAuthAccountInput (soapbind:body, use = literal)Source code
parameters type *addAuthAccountRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- authSystemId type *string*
  is the identifier of the user-authentication system
- userId type *string*
  represents an admin option to add a user-authentication account of a specific user
- acLogin type *string*
  acLogin represents the login of the user on the associated user-authentication system

Output:
addAuthAccountOutput (soapbind:body, use = literal)Source code
parameters type *addAuthAccountResponse*
Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNKNOWN_AUTH_SYSTEMMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_SYSTEMFault*
Fault:
AUTH_ACCOUNT_EXISTMessage (documentation, use = literal)Source code
fault type *AUTH_ACCOUNT_EXISTFault*
Fault:
LOGIN_ALREADY_USEDMessage (documentation, use = literal)Source code
fault type *LOGIN_ALREADY_USEDFault*
Fault:

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

2. addLocalAccount

Source code
Description:
adds a new local user configuration
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
addLocalAccountInput (soapbind:body, use = literal)Source code
parameters type *addLocalAccountRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ userId type *string*
  The userId represents the VISHNU user identifier of the user of the local user configuration
- ♦ machineId type *string*
  The MachineId represents the identifier of the machine associated to the local user configuration
- ♦ acLogin type *string*
  acLogin represents the login of the user on the associated machine
- ♦ sshKeyPath type *string*
  sshKeyPath is the path of the ssh key of the user on the associated machine
- ♦ homeDirectory type *string*
  HomeDirectory is the path of the home directory of the user on the associated machine

Output:
addLocalAccountOutput (soapbind:body, use = literal)Source code
parameters type *addLocalAccountResponse*

- ♦ sshPublicKey type *string*
  The SSH public key generated by VISHNU for accessing a local account

Fault:
LOCAL_ACCOUNT_EXISTMessage (documentation, use = literal)Source code
fault type *LOCAL_ACCOUNT_EXISTFault*
Fault:
MACHINE_LOCKEDMessage (documentation, use = literal)Source code
fault type *MACHINE_LOCKEDFault*
Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
LOGIN_ALREADY_USEDMessage (documentation, use = literal)Source code
fault type *LOGIN_ALREADY_USEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
UNUSABLE_MACHINEMessage (documentation, use = literal)Source code
fault type *UNUSABLE_MACHINEFault*

3. changePassword

Source code
Description:
changes the password
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
changePasswordInput (soapbind:body, use = literal)Source code
parameters type *changePasswordRequest*

- userId type *string*
  userId represents the VISHNU user identifier
- password type *string*
  password represents the password of the user
- passwordNew type *string*
  passwordNew represents the new password of the user

Output:
changePasswordOutput (soapbind:body, use = literal)Source code
parameters type *changePasswordResponse*

Fault:
UNKNOWN_USERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
READONLY_ACCOUNTMessage (documentation, use = literal)Source code
fault type *READONLY_ACCOUNTFault*
Fault:
AUTHENTERRMessage (documentation, use = literal)Source code
fault type *AUTHENTERRFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

4. **close**

Source code
Description:
closes the session
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
closeInput (soapbind:body, use = literal)Source code
parameters type *closeRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU

Output:
closeOutput (soapbind:body, use = literal)Source code
parameters type *closeResponse*
Fault:
COMMAND_RUNNINGMessage (documentation, use = literal)Source code
fault type *COMMAND_RUNNINGFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*

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

5. configureOption

Source code
Description:
configures an option of the user
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
configureOptionInput (soapbind:body, use = literal)Source code
parameters type *configureOptionRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ optionName type *string*
  represents the name of an option
- ♦ value type *string*
  represents the value of an option

Output:
configureOptionOutput (soapbind:body, use = literal)Source code
parameters type *configureOptionResponse*
Fault:
UNKNOWN_OPTIONMessage (documentation, use = literal)Source code
fault type *UNKNOWN_OPTIONFault*
Fault:
UNKNOWN_CLOSURE_MODEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_CLOSURE_MODEFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
INCORRECT_TIMEOUTMessage (documentation, use = literal)Source code
fault type *INCORRECT_TIMEOUTFault*
Fault:
INCORRECT_TRANSFER_CMDMessage (documentation, use = literal)Source code

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

6. connect

Source code
Description:
opens a session
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
connectInput (soapbind:body, use = literal)Source code
parameters type *connectRequest*

- userId type *string*
  userId represents the VISHNU user identifier. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user
- password type *string*
  password represents the password of the user. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user
- closePolicy - optional; type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is an option for closing session automatically
- sessionInactivityDelay - optional; type *integer*
  is the maximum delay in seconds between two user requests when the CLOSE_ON_TIMEOUT policy is set
- substituteUserId - optional; type *string*
  is an admin option which allows an admin to open a session as if she was another user identified by her userId

connectInput (soapbind:body, use = literal)Source code
parameters type *connectRequest*

- userId type *string*
  userId represents the VISHNU user identifier. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user
- password type *string*
  password represents the password of the user. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user
- closePolicy - optional; type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is an option for closing session automatically

- sessionInactivityDelay - optional; type *integer*
  is the maximum delay in seconds between two user requests when the CLOSE_ON_TIMEOUT policy is set
- substituteUserId - optional; type *string*
  is an admin option which allows an admin to open a session as if she was another user identified by her userId

Output:
connectOutput (soapbind:body, use = literal)Source code
parameters type *connectResponse*

- sessionId type *string*
  represents the VISHNU session identifier of the session
- userId type *string*
  represents the VISHNU user identifier of the user who has opened the session
- sessionKey type *string*
  is the key of the session generated by VISHNU
- dateLastConnect type *long*
  is the date of the last connection to the session (the UNIX timestamps is used)
- dateCreation type *long*
  is the date of the first connection to the session (the UNIX timestamps is used)
- dateClosure type *long*
  is the date of the closure of the session (the UNIX timestamps is used)
- status type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
  represents the status of the session
- closePolicy type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is the way to close the session
- timeout type *long*
  is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT option
- authenId type *string*
  represents the identifier defined on a user-authentication system and used for the current session

connectOutput (soapbind:body, use = literal)Source code
parameters type *connectResponse*

- sessionId type *string*
  represents the VISHNU session identifier of the session
- userId type *string*
  represents the VISHNU user identifier of the user who has opened the session
- sessionKey type *string*
  is the key of the session generated by VISHNU
- dateLastConnect type *long*
  is the date of the last connection to the session (the UNIX timestamps is used)
- dateCreation type *long*
  is the date of the first connection to the session (the UNIX timestamps is used)
- dateClosure type *long*
  is the date of the closure of the session (the UNIX timestamps is used)
- status type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
  represents the status of the session
- closePolicy type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is the way to close the session
- timeout type *long*

is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT option

- authenId type *string*
  represents the identifier defined on a user-authentication system and used for the current session

Fault:
UNKNOWN_USERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERFault*
Fault:
UNKNOWN_CLOSURE_MODEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_CLOSURE_MODEFault*
Fault:
INCORRECT_TIMEOUTMessage (documentation, use = literal)Source code
fault type *INCORRECT_TIMEOUTFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
AUTHENTERRMessage (documentation, use = literal)Source code
fault type *AUTHENTERRFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
UNKNOWN_USERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERFault*
Fault:
UNKNOWN_CLOSURE_MODEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_CLOSURE_MODEFault*
Fault:
INCORRECT_TIMEOUTMessage (documentation, use = literal)Source code
fault type *INCORRECT_TIMEOUTFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code

fault type *UNKNOWN_USERIDFault*
Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
AUTHENTERRMessage (documentation, use = literal)Source code
fault type *AUTHENTERRFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

7. deleteAuthAccount

Source code
Description:
removes a local user-authentication configuration from VISHNU
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
deleteAuthAccountInput (soapbind:body, use = literal)Source code
parameters type *deleteAuthAccountRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ authSystemId type *string*
  authSystemId is the identifier of the user-authentication system
- ♦ userId - optional; type *string*
  is an admin option which represents the VISHNU identifier of the user whose local user-authentication configuration will be deleted

Output:
deleteAuthAccountOutput (soapbind:body, use = literal)Source code
parameters type *deleteAuthAccountResponse*
Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*

Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNKNOWN_AUTH_SYSTEMMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_SYSTEMFault*
Fault:
UNKNOWN_AUTH_ACCOUNTMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_ACCOUNTFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

8. deleteLocalAccount

Source code
Description:
removes a local user configuration (for a given user on a given machine) from VISHNU
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
deleteLocalAccountInput (soapbind:body, use = literal)Source code
parameters type *deleteLocalAccountRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- userId type *string*
  userId represents the VISHNU user identifier of the user whose local configuration will be deleted for the given machine
- machineId type *string*
  machineId represents the identifier of the machine whose local configuration will be deleted for the given user

Output:
deleteLocalAccountOutput (soapbind:body, use = literal)Source code
parameters type *deleteLocalAccountResponse*

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

9. listAuthAccounts

Source code
Description:
lists local user-authentication configurations
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listAuthAccountsInput (soapbind:body, use = literal)Source code
parameters type *listAuthAccountsRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ listAll - optional; type *boolean*
  is an admin option for listing all local user-authentication configurations of VISHNU
- ♦ userId - optional; type *string*
  is an admin option for listing all local user-authentication configurations of a specific user identified by his/her userId
- ♦ authSystemId - optional; type *string*
  is an option for listing local user-configuration of a specific user-authentication system

Output:
listAuthAccountsOutput (soapbind:body, use = literal)Source code
parameters type *listAuthAccountsResponse*

- ♦ data - optional;
  - ♦ authaccount - optional, unbounded, nillable;
    - • authSystemId - required; type *string*
      is the identifier of the user-authentication system

- userId - optional; type *string*
  represents an admin option to add a user-authentication account of a specific user
- acLogin - required; type *string*
  acLogin represents the login of the user on the associated user-authentication system

Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNKNOWN_AUTH_SYSTEMMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_SYSTEMFault*
Fault:
UNKNOWN_AUTH_ACCOUNTMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_ACCOUNTFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

10. listAuthSystems

Source code
Description:
lists VISHNU user-authentification systems
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listAuthSystemsInput (soapbind:body, use = literal)Source code

parameters type *listAuthSystemsRequest*

- ♦ sessionKey type *string*
  The sessionKey is the identifier of the session generated by VISHNU
- ♦ listAllAuthSystems - optional; type *boolean*
  is an option for listing all VISHNU user-authentication systems
- ♦ listFullInfo - optional; type *boolean*
  is an admin option for listing full VISHNU user-authentication systems information such as all concerned only the administrator: authLogin, authPassword and userPasswordEncryption
- ♦ userId - optional; type *string*
  is an admin option for listing all user-authentication systems in which a specific user has local user-authentication configs
- ♦ authSystemId - optional; type *string*
  is an option for listing a specific user-authentication system

Output:
listAuthSystemsOutput (soapbind:body, use = literal)Source code
parameters type *listAuthSystemsResponse*

- ♦ data - optional;
  - ♦ authsystem - optional, unbounded, nillable;
    - ● authSystemId - required; type *string*
      is the identifier, generated by VISHNU, of the user-authentication system
    - ● name - required; type *string*
      corresponds to the user-authentication system's name
    - ● URI - required; type *string*
      is the URI of the user-authentication systems (by the form host:port for LDAP)
    - ● authLogin - required; type *string*
      is the login used to connect to the user-authentication system
    - ● authPassword - required; type *string*
      is the password used to connect to the user-authentication system
    - ● userPasswordEncryption - required; type *EncryptionMethod* - type *string* with restriction - enum { 'UNDEFINED', 'SSHA' }
      represents the encryption method used to encrypt user's password
    - ● type - optional; type *AuthType* - type *string* with restriction - enum { 'UNDEFINED', 'LDAP' }
      represents the type of the user-authentication system
    - ● status - optional; type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
      represents the status of the user-authentication system
    - ● ldapBase - optional; type *string*
      is an option for user-authentication system based on LDAP which specifies the DN of the root entry

Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
UNKNOWN_AUTH_SYSTEM_TYPEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_SYSTEM_TYPEFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

11. listHistoryCmd

Source code
Description:
lists the commands
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listHistoryCmdInput (soapbind:body, use = literal)Source code
parameters type *listHistoryCmdRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- adminListOption - optional; type *boolean*
  is an admin option for listing all commands of all users
- userId - optional; type *string*
  is an admin option for listing commands launched by a specific user identified by his/her userId
- sessionId - optional; type *string*
  lists all commands launched within a specific session
- startDateOption - optional; type *long*
  allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used)
- endDateOption - optional; type *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

Output:
listHistoryCmdOutput (soapbind:body, use = literal)Source code
parameters type *listHistoryCmdResponse*

- ♦ data - optional;
  - ♦ command - optional, unbounded, nillable;
    - commandId - optional; type *string*
      is the identifier of a command
    - sessionId - optional; type *string*
      The sessionId is the identifier of the session define in the database
    - machineId - optional; type *string*
      The machineId is the identifier of the machine used by the command
    - cmdDescription - optional; type *string*
      cmdDescription is the description of the command
    - cmdStartTime - optional; type *long*
      cmdStartTime is the date of the command beginning (the UNIX timestamps is used)
    - cmdEndTime - optional; type *long*
      cmdEndTime is the date of the command end (the UNIX timestamps is used)
    - status - optional; type *CommandStatusType* - type *string*
      with restriction - enum { 'FAILURE', 'SUCCESS' }
      The machineId is the identifier of the machine used by the command

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

12. listLocalAccounts

Source code

Description:

lists the local user configurations

Operation type:

Request-response. The endpoint receives a message, and sends a correlated message.

Input:

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

parameters type *listLocalAccountsRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ adminListOption - optional; type *boolean*
  is an admin option for listing all local configurations of all users
- ♦ userId - optional; type *string*
  is an admin option for listing the local configurations of a specific user
- ♦ machineId - optional; type *string*
  is an option for listing local user configurations on a specific machine

Output:

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

parameters type *listLocalAccountsResponse*

- ♦ data - optional;
  - ♦ localaccount - optional, unbounded, nillable;
    - userId - required; type *string*
      The userId represents the VISHNU user identifier of the user of the local user configuration
    - machineId - required; type *string*
      The MachineId represents the identifier of the machine associated to the local user configuration
    - acLogin - optional; type *string*
      acLogin represents the login of the user on the associated machine
    - sshKeyPath - optional; type *string*
      sshKeyPath is the path of the ssh key of the user on the associated machine
    - homeDirectory - optional; type *string*
      HomeDirectory is the path of the home directory of the user on the associated machine

Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)Source code

fault type *UNKNOWN_USERIDFault*

Fault:

SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code

fault type *SESSIONKEY_EXPIREDFault*

Fault:

SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code

fault type *SESSIONKEY_NOT_FOUNDFault*

Fault:

DIETMessage (documentation, use = literal)Source code

fault type *DIETFault*

Fault:

DBERRMessage (documentation, use = literal)Source code

fault type *DBERRFault*

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

13. listMachines

Source code
Description:
lists the machines that are accessible through VISHNU
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listMachinesInput (soapbind:body, use = literal)Source code
parameters type *listMachinesRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- userId - optional; type *string*
  is an admin option for listing machines in which a specific user has a local configuration
- listAllMachine - optional; type *boolean*
  is an option for listing all VISHNU machines
- machineId - optional; type *string*
  is an option for listing information about a specific machine

Output:
listMachinesOutput (soapbind:body, use = literal)Source code
parameters type *listMachinesResponse*

- data - optional;
  - machine - optional, unbounded, nillable;
    - machineId - required; type *string*
      represents the identifier of the machine
    - name - optional; type *string*
      represents the name of the machine
    - site - optional; type *string*
      represents the location of the machine
    - machineDescription - optional; type *string*
      represents the description of the machine
    - language - optional; type *string*
      represents the language used for the description of the machine
    - status - optional; type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
      represents the status of the machine
    - sshPublicKey - optional; type *string*
      contains the SSH public key used by VISHNU to access local user accounts

Fault:

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

14. listOptions

Source code
Description:
lists the options of the user
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listOptionsInput (soapbind:body, use = literal)Source code
parameters type *listOptionsRequest*

- ♦ sessionKey type *string*
  The sessionKey is the identifier of the session generated by VISHNU
- ♦ listAllDeftValue - optional; type *boolean*
  is an option for listing all default option values defined by VISHNU administrator
- ♦ userId - optional; type *string*
  is an admin option for listing the options of a specific user
- ♦ optionName - optional; type *string*
  allows the user to get the value of a specific option identified by its name

Output:
listOptionsOutput (soapbind:body, use = literal)Source code
parameters type *listOptionsResponse*

- ♦ data - optional;
  - ♦ optionvalue - optional, unbounded, nillable;
    - • optionName - required; type *string*
      represents the name of an option
    - • value - required; type *string*
      represents the value of an option

Fault:

NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
UNKNOWN_OPTIONMessage (documentation, use = literal)Source code
fault type *UNKNOWN_OPTIONFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

15. listSessions

Source code
Description:
lists all sessions of the user
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listSessionsInput (soapbind:body, use = literal)Source code
parameters type *listSessionsRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- status - optional; type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
  specifies the status of the sessions which will be listed
- sessionClosePolicy - optional; type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  specifies the closure mode of the sessions which will be listed (CLOSE_ON_TIMEOUT or CLOSE_ON_DISCONNECT)
- sessionInactivityDelay - optional; type *integer*
  specifies the inactivity delay in seconds of the sessions which will be listed
- machineId - optional; type *string*
  allows the user to list sessions opened on a specific client's machine by using its name

- adminListOption - optional; type *boolean*
  is an admin option for listing all sessions of all users
- userId - optional; type *string*
  is an admin option for listing sessions opened by a specific user
- sessionId - optional; type *string*
  allows the user to list all commands launched within a specific session
- startDateOption - optional; type *long*
  allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used)
- endDateOption - optional; type *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

Output:
listSessionsOutput (soapbind:body, use = literal)Source code
parameters type *listSessionsResponse*

- data - optional;
  - session - optional, unbounded, nillable;
    - sessionId - optional; type *string*
      represents the VISHNU session identifier of the session
    - userId - optional; type *string*
      represents the VISHNU user identifier of the user who has opened the session
    - sessionKey - optional; type *string*
      is the key of the session generated by VISHNU
    - dateLastConnect - optional; type *long*
      is the date of the last connection to the session (the UNIX timestamps is used)
    - dateCreation - optional; type *long*
      is the date of the first connection to the session (the UNIX timestamps is used)
    - dateClosure - optional; type *long*
      is the date of the closure of the session (the UNIX timestamps is used)
    - status - optional; type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
      represents the status of the session
    - closePolicy - optional; type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
      is the way to close the session
    - timeout - optional; type *long*
      is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT option
    - authenId - optional; type *string*
      represents the identifier defined on a user-authentication system and used for the current session

Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:

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

16. reconnect

Source code
Description:
reconnects to a session that is still active
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
reconnectInput (soapbind:body, use = literal)Source code
parameters type *reconnectRequest*

- ♦ userId type *string*
  userId represents the VISHNU user identifier. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user
- ♦ password type *string*
  password represents the password of the user. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user
- ♦ sessionId type *string*
  sessionId is the identifier of the session defined in the database

reconnectInput (soapbind:body, use = literal)Source code
parameters type *reconnectRequest*

- ♦ userId type *string*
  userId represents the VISHNU user identifier. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user
- ♦ password type *string*
  password represents the password of the user. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user
- ♦ sessionId type *string*
  sessionId is the identifier of the session defined in the database

Output:

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

parameters type *reconnectResponse*

- sessionId type *string*
  represents the VISHNU session identifier of the session
- userId type *string*
  represents the VISHNU user identifier of the user who has opened the session
- sessionKey type *string*
  is the key of the session generated by VISHNU
- dateLastConnect type *long*
  is the date of the last connection to the session (the UNIX timestamps is used)
- dateCreation type *long*
  is the date of the first connection to the session (the UNIX timestamps is used)
- dateClosure type *long*
  is the date of the closure of the session (the UNIX timestamps is used)
- status type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
  represents the status of the session
- closePolicy type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is the way to close the session
- timeout type *long*
  is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT option
- authenId type *string*
  represents the identifier defined on a user-authentication system and used for the current session

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

parameters type *reconnectResponse*

- sessionId type *string*
  represents the VISHNU session identifier of the session
- userId type *string*
  represents the VISHNU user identifier of the user who has opened the session
- sessionKey type *string*
  is the key of the session generated by VISHNU
- dateLastConnect type *long*
  is the date of the last connection to the session (the UNIX timestamps is used)
- dateCreation type *long*
  is the date of the first connection to the session (the UNIX timestamps is used)
- dateClosure type *long*
  is the date of the closure of the session (the UNIX timestamps is used)
- status type *StatusType* - type *string* with restriction - enum { 'UNDEFINED', 'INACTIVE', 'ACTIVE' }
  represents the status of the session
- closePolicy type *SessionCloseType* - type *string* with restriction - enum { 'DEFAULT', 'CLOSE_ON_TIMEOUT', 'CLOSE_ON_DISCONNECT' }
  is the way to close the session
- timeout type *long*
  is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT option
- authenId type *string*
  represents the identifier defined on a user-authentication system and used for the current session

Fault:

UNKNOWN_USERMessage (documentation, use = literal)Source code

fault type *UNKNOWN_USERFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNUSABLE_MACHINEMessage (documentation, use = literal)Source code
fault type *UNUSABLE_MACHINEFault*
Fault:
UNKNOWN_SESSION_IDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_SESSION_IDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
AUTHENTERRMessage (documentation, use = literal)Source code
fault type *AUTHENTERRFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
UNKNOWN_USERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNUSABLE_MACHINEMessage (documentation, use = literal)Source code
fault type *UNUSABLE_MACHINEFault*
Fault:
UNKNOWN_SESSION_IDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_SESSION_IDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code

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

17. updateAuthAccount

Source code
Description:
updates a local user-authentication configuration
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
updateAuthAccountInput (soapbind:body, use = literal)Source code
parameters type *updateAuthAccountRequest*

- ♦ sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- ♦ authSystemId type *string*
  is the identifier of the user-authentication system
- ♦ userId type *string*
  represents an admin option to add a user-authentication account of a specific user
- ♦ acLogin type *string*
  acLogin represents the login of the user on the associated user-authentication system

Output:
updateAuthAccountOutput (soapbind:body, use = literal)Source code
parameters type *updateAuthAccountResponse*
Fault:
NO_ADMINMessage (documentation, use = literal)Source code
fault type *NO_ADMINFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code

fault type *UNKNOWN_USERIDFault*
Fault:
USER_LOCKEDMessage (documentation, use = literal)Source code
fault type *USER_LOCKEDFault*
Fault:
UNKNOWN_AUTH_SYSTEMMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_SYSTEMFault*
Fault:
UNKNOWN_AUTH_ACCOUNTMessage (documentation, use = literal)Source code
fault type *UNKNOWN_AUTH_ACCOUNTFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*

18. updateLocalAccount

Source code
Description:
updates a local user configuration
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
updateLocalAccountInput (soapbind:body, use = literal)Source code
parameters type *updateLocalAccountRequest*

- sessionKey type *string*
  The sessionKey is the encrypted identifier of the session generated by VISHNU
- userId type *string*
  The userId represents the VISHNU user identifier of the user of the local user configuration
- machineId type *string*
  The MachineId represents the identifier of the machine associated to the local user configuration
- acLogin type *string*
  acLogin represents the login of the user on the associated machine
- sshKeyPath type *string*
  sshKeyPath is the path of the ssh key of the user on the associated machine
- homeDirectory type *string*
  HomeDirectory is the path of the home directory of the user on the associated machine

Output:

updateLocalAccountOutput (soapbind:body, use = literal)Source code
parameters type *updateLocalAccountResponse*
Fault:
UNKNOWN_USERIDMessage (documentation, use = literal)Source code
fault type *UNKNOWN_USERIDFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
SESSIONKEY_NOT_FOUNDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_NOT_FOUNDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)Source code
fault type *UNKNOWN_LOCAL_ACCOUNTFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
LOGIN_ALREADY_USEDMessage (documentation, use = literal)Source code
fault type *LOGIN_ALREADY_USEDFault*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="VishnuUMS" targetNamespace="urn:ResourceProxy"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="urn:ResourceProxy"
xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"
>
<types>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
targetNamespace="urn:ResourceProxy"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
```

```
<xs:element name="connectRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>userId represents the VISHNU user identifier. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="password" type="xs:string">
<xs:annotation>
<xs:documentation>password represents the password of the user. If userId and password are empty, vishnu connect will read them in the .netrc file located in the home of the user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="closePolicy" type="Q1:SessionCloseType">
<xs:annotation>
<xs:documentation>is an option for closing session automatically</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sessionInactivityDelay" type="xs:integer">
<xs:annotation>
<xs:documentation>is the maximum delay in seconds between two user requests when the CLOSE_ON_TIMEOUT policy is set</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="substituteUserId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option which allows an admin to open a session as if she was another user identified by her userId</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="SessionCloseType">
<xs:restriction base="xs:string">
<xs:enumeration value="DEFAULT" />
<xs:enumeration value="CLOSE_ON_TIMEOUT" />
<xs:enumeration value="CLOSE_ON_DISCONNECT" />
</xs:restriction>
</xs:simpleType>
<xs:element name="connectResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>represents the VISHNU session identifier of the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
```

```
<xs:annotation>
<xs:documentation>represents the VISHNU user identifier of the user who has opened the
session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>is the key of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateLastConnect" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the last connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateCreation" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the first connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateClosure" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the closure of the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:StatusType">
<xs:annotation>
<xs:documentation>represents the status of the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="closePolicy" type="Q1:SessionCloseType">
<xs:annotation>
<xs:documentation>is the way to close the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="timeout" type="xs:long">
<xs:annotation>
<xs:documentation>is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT
option</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="authenId" type="xs:string">
<xs:annotation>
<xs:documentation>represents the identifier defined on a user-authentication system and used for the current
session </xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
```

```
</xs:complexType>
</xs:element>
<xs:simpleType name="StatusType">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="INACTIVE" />
<xs:enumeration value="ACTIVE" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="PrivilegeType">
<xs:restriction base="xs:string">
<xs:enumeration value="USER" />
<xs:enumeration value="ADMIN" />
</xs:restriction>
</xs:simpleType>
<xs:element name="reconnectRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>userId represents the VISHNU user identifier. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="password" type="xs:string">
<xs:annotation>
<xs:documentation>password represents the password of the user. If userId and password are empty, vishnu reconnect will read them in the .netrc file located in the home of the user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>sessionId is the identifier of the session defined in the database</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="reconnectResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>represents the VISHNU session identifier of the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>represents the VISHNU user identifier of the user who has opened the session</xs:documentation>
</xs:annotation>
```

```
</xs:element>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>is the key of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateLastConnect" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the last connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateCreation" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the first connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dateClosure" type="xs:long">
<xs:annotation>
<xs:documentation>is the date of the closure of the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:StatusType">
<xs:annotation>
<xs:documentation>represents the status of the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="closePolicy" type="Q1:SessionCloseType">
<xs:annotation>
<xs:documentation>is the way to close the session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="timeout" type="xs:long">
<xs:annotation>
<xs:documentation>is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT
option</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="authenId" type="xs:string">
<xs:annotation>
<xs:documentation>represents the identifier defined on a user-authentication system and used for the current
session </xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="closeRequest">
<xs:complexType>
```

```
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="closeResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="changePasswordRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>userId represents the VISHNU user identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="password" type="xs:string">
<xs:annotation>
<xs:documentation>password represents the password of the user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="passwordNew" type="xs:string">
<xs:annotation>
<xs:documentation>passwordNew represents the new password of the user</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="changePasswordResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="addLocalAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
```

```
<xs:annotation>
<xs:documentation>The userId represents the VISHNU user identifier of the user of the local user configuration</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>The MachineId represents the identifier of the machine associated to the local user configuration</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="acLogin" type="xs:string">
<xs:annotation>
<xs:documentation>acLogin represents the login of the user on the associated machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sshKeyPath" type="xs:string">
<xs:annotation>
<xs:documentation>sshKeyPath is the path of the ssh key of the user on the associated machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="homeDirectory" type="xs:string">
<xs:annotation>
<xs:documentation>HomeDirectory is the path of the home directory of the user on the associated machine</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="addLocalAccountResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sshPublicKey" type="xs:string">
<xs:annotation>
<xs:documentation>The SSH public key generated by VISHNU for accessing a local account</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="updateLocalAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by VISHNU</xs:documentation>
</xs:annotation>
```

```
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>The userId represents the VISHNU user identifier of the user of the local user
configuration</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>The MachineId represents the identifier of the machine associated to the local user
configuration</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="acLogin" type="xs:string">
<xs:annotation>
<xs:documentation>acLogin represents the login of the user on the associated machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sshKeyPath" type="xs:string">
<xs:annotation>
<xs:documentation>sshKeyPath is the path of the ssh key of the user on the associated
machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="homeDirectory" type="xs:string">
<xs:annotation>
<xs:documentation>HomeDirectory is the path of the home directory of the user on the associated
machine</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="updateLocalAccountResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="deleteLocalAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>userId represents the VISHNU user identifier of the user whose local configuration will
be deleted for the given machine </xs:documentation>
```

```
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>machineId represents the identifier of the machine whose local configuration will be
deleted for the given user </xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="deleteLocalAccountResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="listLocalAccountsRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="adminListOption" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an admin option for listing all local configurations of all users </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing the local configurations of a specific user
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>is an option for listing local user configurations on a specific
machine</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listLocalAccountsResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
```

```
<xs:element maxOccurs="unbounded" minOccurs="0" name="localaccount" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="userId" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>The userId represents the VISHNU user identifier of the user of the local user
configuration</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="machineId" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>The MachineId represents the identifier of the machine associated to the local user
configuration</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="acLogin" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>acLogin represents the login of the user on the associated machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="sshKeyPath" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>sshKeyPath is the path of the ssh key of the user on the associated
machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="homeDirectory" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>HomeDirectory is the path of the home directory of the user on the associated
machine</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listMachinesRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
```

```
<xs:documentation>is an admin option for listing machines in which a specific user has a local configuration</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="listAllMachine" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an option for listing all VISHNU machines</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>is an option for listing information about a specific machine</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listMachinesResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="machine" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="machineId" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>represents the identifier of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="name" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the name of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="site" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the location of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="machineDescription" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the description of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="language" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the language used for the description of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
```

```
<xs:attribute name="status" type="Q1:StatusType" use="optional">
<xs:annotation>
<xs:documentation>represents the status of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="sshPublicKey" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>contains the SSH public key used by VISHNU to access local user accounts</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listHistoryCmdRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="adminListOption" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an admin option for listing all commands of all users</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing commands launched by a specific user identified by his/her userId</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>lists all commands launched within a specific session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="startDateOption" type="xs:long">
<xs:annotation>
<xs:documentation>allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="endDateOption" type="xs:long">
```

```
<xs:annotation>
<xs:documentation>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</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listHistoryCmdResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="command" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="commandId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>is the identifier of a command</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="sessionId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>The sessionId is the identifier of the session define in the database</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="machineId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>The machineId is the identifier of the machine used by the command</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="cmdDescription" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>cmdDescription is the description of the command</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="cmdStartTime" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>cmdStartTime is the date of the command beginning (the UNIX timestamps is used)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="cmdEndTime" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>cmdEndTime is the date of the command end (the UNIX timestamps is used)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="status" type="Q1:CommandStatusType" use="optional">
```

```
<xs:annotation>
<xs:documentation>The machineId is the identifier of the machine used by the
command</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="CommandStatusType">
<xs:restriction base="xs:string">
<xs:enumeration value="FAILURE" />
<xs:enumeration value="SUCCESS" />
</xs:restriction>
</xs:simpleType>
<xs:element name="listOptionsRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="listAllDeftValue" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an option for listing all default option values defined by VISHNU
administrator</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing the options of a specific user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="optionName" type="xs:string">
<xs:annotation>
<xs:documentation> allows the user to get the value of a specific option identified by its name
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listOptionsResponse">
<xs:complexType>
<xs:sequence>
```

```
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="optionvalue" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="optionName" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>represents the name of an option</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="value" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>represents the value of an option</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listSessionsRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="status" type="Q1:StatusType">
<xs:annotation>
<xs:documentation>specifies the status of the sessions which will be listed</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sessionClosePolicy" type="Q1:SessionCloseType">
<xs:annotation>
<xs:documentation>specifies the closure mode of the sessions which will be listed (CLOSE_ON_TIMEOUT or CLOSE_ON_DISCONNECT)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sessionInactivityDelay" type="xs:integer">
<xs:annotation>
<xs:documentation>specifies the inactivity delay in seconds of the sessions which will be listed</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="machineId" type="xs:string">
```

```
<xs:annotation>
<xs:documentation>allows the user to list sessions opened on a specific client's machine by using its name</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="adminListOption" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an admin option for listing all sessions of all users</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing sessions opened by a specific user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>allows the user to list all commands launched within a specific session</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="startDateOption" type="xs:long">
<xs:annotation>
<xs:documentation>allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="endDateOption" type="xs:long">
<xs:annotation>
<xs:documentation>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</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listSessionsResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="session" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="sessionId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the VISHNU session identifier of the session</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="userId" type="xs:string" use="optional">
```

```
<xs:annotation>
<xs:documentation>represents the VISHNU user identifier of the user who has opened the
session</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="sessionKey" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>is the key of the session generated by VISHNU</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="dateLastConnect" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>is the date of the last connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="dateCreation" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>is the date of the first connection to the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="dateClosure" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>is the date of the closure of the session (the UNIX timestamps is
used)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="status" type="Q1:StatusType" use="optional">
<xs:annotation>
<xs:documentation>represents the status of the session</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="closePolicy" type="Q1:SessionCloseType" use="optional">
<xs:annotation>
<xs:documentation>is the way to close the session</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="timeout" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>is the inactivity delay in seconds associated to the CLOSE_ON_TIMEOUT
option</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="authenId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents the identifier defined on a user-authentication system and used for the current
session </xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
```

```xml
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="configureOptionRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="optionName" type="xs:string">
<xs:annotation>
<xs:documentation>represents the name of an option</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="value" type="xs:string">
<xs:annotation>
<xs:documentation>represents the value of an option</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="configureOptionResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="listAuthSystemsRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="listAllAuthSystems" type="xs:boolean">
<xs:annotation>
<xs:documentation>is an option for listing all VISHNU user-authentication systems</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="listFullInfo" type="xs:boolean">
<xs:annotation>
```

```xml
<xs:documentation>is an admin option for listing full VISHNU user-authentication systems information such as all concerned only the administrator: authLogin, authPassword and userPasswordEncryption
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing all user-authentication systems in which a specific user has local user-authentication configs</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="authSystemId" type="xs:string">
<xs:annotation>
<xs:documentation>is an option for listing a specific user-authentication system</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listAuthSystemsResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="authsystem" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="authSystemId" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>is the identifier, generated by VISHNU, of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="name" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>corresponds to the user-authentication system's name</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="URI" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>is the URI of the user-authentication systems (by the form host:port for LDAP)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="authLogin" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>is the login used to connect to the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="authPassword" type="xs:string" use="required">
```

```
<xs:annotation>
<xs:documentation>is the password used to connect to the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="userPasswordEncryption" type="Q1:EncryptionMethod" use="required">
<xs:annotation>
<xs:documentation>represents the encryption method used to encrypt user's password</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="type" type="Q1:AuthType" use="optional">
<xs:annotation>
<xs:documentation>represents the type of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="status" type="Q1:StatusType" use="optional">
<xs:annotation>
<xs:documentation>represents the status of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="ldapBase" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>is an option for user-authentication system based on LDAP which specifies the DN of the
root entry </xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="EncryptionMethod">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="SSHA" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="AuthType">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="LDAP" />
</xs:restriction>
</xs:simpleType>
<xs:element name="addAuthAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
```

```
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="authSystemId" type="xs:string">
<xs:annotation>
<xs:documentation>is the identifier of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>represents an admin option to add a user-authentication account of a specific
user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="acLogin" type="xs:string">
<xs:annotation>
<xs:documentation>acLogin represents the login of the user on the associated user-authentication
system</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="addAuthAccountResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="updateAuthAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="authSystemId" type="xs:string">
<xs:annotation>
<xs:documentation>is the identifier of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>represents an admin option to add a user-authentication account of a specific
user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="acLogin" type="xs:string">
<xs:annotation>
```

```
<xs:documentation>acLogin represents the login of the user on the associated user-authentication
system</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="updateAuthAccountResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="deleteAuthAccountRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="authSystemId" type="xs:string">
<xs:annotation>
<xs:documentation>authSystemId is the identifier of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option which represents the VISHNU identifier of the user whose local
user-authentication configuration will be deleted</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="deleteAuthAccountResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="listAuthAccountsRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The sessionKey is the encrypted identifier of the session generated by
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="listAll" type="xs:boolean">
<xs:annotation>
```

```
<xs:documentation>is an admin option for listing all local user-authentication configurations of
VISHNU</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>is an admin option for listing all local user-authentication configurations of a specific user
identified by his/her userId</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="authSystemId" type="xs:string">
<xs:annotation>
<xs:documentation>is an option for listing local user-configuration of a specific user-authentication
system</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listAuthAccountsResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="authaccount" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="authSystemId" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>is the identifier of the user-authentication system</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="userId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>represents an admin option to add a user-authentication account of a specific
user</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="acLogin" type="xs:string" use="required">
<xs:annotation>
<xs:documentation>acLogin represents the login of the user on the associated user-authentication
system</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
```

```
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_USERFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_USERIDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="USERID_EXISTINGFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="USER_LOCKEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="NO_ADMINFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="TEMPORARY_PASSWORDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="INVALID_MAIL_ADRESSFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SESSIONKEY_NOT_FOUNDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SESSIONKEY_EXPIREDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_SESSION_IDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
```

```
</xs:element>
<xs:element name="COMMAND_RUNNINGFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_MACHINEFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="MACHINE_EXISTINGFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="MACHINE_LOCKEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNUSABLE_MACHINEFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="LOCAL_ACCOUNT_EXISTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_LOCAL_ACCOUNTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SAVE_CONFIG_ERRORFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="RESTORE_CONFIG_ERRORFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_OPTIONFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
```

```
<xs:element name="UNKNOWN_CLOSURE_MODEFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="INCORRECT_TIMEOUTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="LOGIN_ALREADY_USEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="AUTHENTERRFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DIETFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DBERRFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DBCONNFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SYSTEMFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNDEFINEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="ROOT_USER_ONLYFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="USER_ALREADY_LOCKEDFault">
```

```
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="MACHINE_ALREADY_LOCKEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="INCORRECT_TRANSFER_CMDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_AUTH_SYSTEM_TYPEFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_AUTH_SYSTEMFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="AUTH_SYSTEM_ALREADY_LOCKEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="AUTH_SYSTEM_ALREADY_EXISTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="AUTH_ACCOUNT_EXISTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_AUTH_ACCOUNTFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_ENCRYPTION_METHODFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="READONLY_ACCOUNTFault">
<xs:complexType>
```

```
<xs:sequence />
</xs:complexType>
</xs:element>
</xs:schema>
</types>
<message name="connectInput">
<part element="tns:connectRequest" name="parameters" />
</message>
<message name="connectOutput">
<part element="tns:connectResponse" name="parameters" />
</message>
<message name="connectInput">
<part element="tns:connectRequest" name="parameters" />
</message>
<message name="connectOutput">
<part element="tns:connectResponse" name="parameters" />
</message>
<message name="reconnectInput">
<part element="tns:reconnectRequest" name="parameters" />
</message>
<message name="reconnectOutput">
<part element="tns:reconnectResponse" name="parameters" />
</message>
<message name="reconnectInput">
<part element="tns:reconnectRequest" name="parameters" />
</message>
<message name="reconnectOutput">
<part element="tns:reconnectResponse" name="parameters" />
</message>
<message name="closeInput">
<part element="tns:closeRequest" name="parameters" />
</message>
<message name="closeOutput">
<part element="tns:closeResponse" name="parameters" />
</message>
<message name="changePasswordInput">
<part element="tns:changePasswordRequest" name="parameters" />
</message>
<message name="changePasswordOutput">
<part element="tns:changePasswordResponse" name="parameters" />
</message>
<message name="addLocalAccountInput">
<part element="tns:addLocalAccountRequest" name="parameters" />
</message>
<message name="addLocalAccountOutput">
<part element="tns:addLocalAccountResponse" name="parameters" />
</message>
<message name="updateLocalAccountInput">
<part element="tns:updateLocalAccountRequest" name="parameters" />
</message>
<message name="updateLocalAccountOutput">
```

```
<part element="tns:updateLocalAccountResponse" name="parameters" />
</message>
<message name="deleteLocalAccountInput">
<part element="tns:deleteLocalAccountRequest" name="parameters" />
</message>
<message name="deleteLocalAccountOutput">
<part element="tns:deleteLocalAccountResponse" name="parameters" />
</message>
<message name="listLocalAccountsInput">
<part element="tns:listLocalAccountsRequest" name="parameters" />
</message>
<message name="listLocalAccountsOutput">
<part element="tns:listLocalAccountsResponse" name="parameters" />
</message>
<message name="listMachinesInput">
<part element="tns:listMachinesRequest" name="parameters" />
</message>
<message name="listMachinesOutput">
<part element="tns:listMachinesResponse" name="parameters" />
</message>
<message name="listHistoryCmdInput">
<part element="tns:listHistoryCmdRequest" name="parameters" />
</message>
<message name="listHistoryCmdOutput">
<part element="tns:listHistoryCmdResponse" name="parameters" />
</message>
<message name="listOptionsInput">
<part element="tns:listOptionsRequest" name="parameters" />
</message>
<message name="listOptionsOutput">
<part element="tns:listOptionsResponse" name="parameters" />
</message>
<message name="listSessionsInput">
<part element="tns:listSessionsRequest" name="parameters" />
</message>
<message name="listSessionsOutput">
<part element="tns:listSessionsResponse" name="parameters" />
</message>
<message name="configureOptionInput">
<part element="tns:configureOptionRequest" name="parameters" />
</message>
<message name="configureOptionOutput">
<part element="tns:configureOptionResponse" name="parameters" />
</message>
<message name="listAuthSystemsInput">
<part element="tns:listAuthSystemsRequest" name="parameters" />
</message>
<message name="listAuthSystemsOutput">
<part element="tns:listAuthSystemsResponse" name="parameters" />
</message>
<message name="addAuthAccountInput">
```

```
<part element="tns:addAuthAccountRequest" name="parameters" />
</message>
<message name="addAuthAccountOutput">
<part element="tns:addAuthAccountResponse" name="parameters" />
</message>
<message name="updateAuthAccountInput">
<part element="tns:updateAuthAccountRequest" name="parameters" />
</message>
<message name="updateAuthAccountOutput">
<part element="tns:updateAuthAccountResponse" name="parameters" />
</message>
<message name="deleteAuthAccountInput">
<part element="tns:deleteAuthAccountRequest" name="parameters" />
</message>
<message name="deleteAuthAccountOutput">
<part element="tns:deleteAuthAccountResponse" name="parameters" />
</message>
<message name="listAuthAccountsInput">
<part element="tns:listAuthAccountsRequest" name="parameters" />
</message>
<message name="listAuthAccountsOutput">
<part element="tns:listAuthAccountsResponse" name="parameters" />
</message>
<message name="UNKNOWN_USERMessage">
<part element="tns:UNKNOWN_USERFault" name="fault" />
</message>
<message name="UNKNOWN_CLOSURE_MODEMessage">
<part element="tns:UNKNOWN_CLOSURE_MODEFault" name="fault" />
</message>
<message name="INCORRECT_TIMEOUTMessage">
<part element="tns:INCORRECT_TIMEOUTFault" name="fault" />
</message>
<message name="UNKNOWN_USERIDMessage">
<part element="tns:UNKNOWN_USERIDFault" name="fault" />
</message>
<message name="NO_ADMINMessage">
<part element="tns:NO_ADMINFault" name="fault" />
</message>
<message name="USER_LOCKEDMessage">
<part element="tns:USER_LOCKEDFault" name="fault" />
</message>
<message name="AUTHENTERRMessage">
<part element="tns:AUTHENTERRFault" name="fault" />
</message>
<message name="DIETMessage">
<part element="tns:DIETFault" name="fault" />
</message>
<message name="DBERRMessage">
<part element="tns:DBERRFault" name="fault" />
</message>
<message name="DBCONNMessage">
```

```
<part element="tns:DBCONNFault" name="fault" />
</message>
<message name="SYSTEMMessage">
<part element="tns:SYSTEMFault" name="fault" />
</message>
<message name="UNDEFINEDMessage">
<part element="tns:UNDEFINEDFault" name="fault" />
</message>
<message name="UNUSABLE_MACHINEMessage">
<part element="tns:UNUSABLE_MACHINEFault" name="fault" />
</message>
<message name="UNKNOWN_SESSION_IDMessage">
<part element="tns:UNKNOWN_SESSION_IDFault" name="fault" />
</message>
<message name="SESSIONKEY_EXPIREDMessage">
<part element="tns:SESSIONKEY_EXPIREDFault" name="fault" />
</message>
<message name="SESSIONKEY_NOT_FOUNDMessage">
<part element="tns:SESSIONKEY_NOT_FOUNDFault" name="fault" />
</message>
<message name="COMMAND_RUNNINGMessage">
<part element="tns:COMMAND_RUNNINGFault" name="fault" />
</message>
<message name="READONLY_ACCOUNTMessage">
<part element="tns:READONLY_ACCOUNTFault" name="fault" />
</message>
<message name="LOCAL_ACCOUNT_EXISTMessage">
<part element="tns:LOCAL_ACCOUNT_EXISTFault" name="fault" />
</message>
<message name="MACHINE_LOCKEDMessage">
<part element="tns:MACHINE_LOCKEDFault" name="fault" />
</message>
<message name="UNKNOWN_MACHINEMessage">
<part element="tns:UNKNOWN_MACHINEFault" name="fault" />
</message>
<message name="LOGIN_ALREADY_USEDMessage">
<part element="tns:LOGIN_ALREADY_USEDFault" name="fault" />
</message>
<message name="UNKNOWN_LOCAL_ACCOUNTMessage">
<part element="tns:UNKNOWN_LOCAL_ACCOUNTFault" name="fault" />
</message>
<message name="UNKNOWN_OPTIONMessage">
<part element="tns:UNKNOWN_OPTIONFault" name="fault" />
</message>
<message name="INCORRECT_TRANSFER_CMDMessage">
<part element="tns:INCORRECT_TRANSFER_CMDFault" name="fault" />
</message>
<message name="UNKNOWN_AUTH_SYSTEM_TYPEMessage">
<part element="tns:UNKNOWN_AUTH_SYSTEM_TYPEFault" name="fault" />
</message>
<message name="UNKNOWN_AUTH_SYSTEMMessage">
```

```
<part element="tns:UNKNOWN_AUTH_SYSTEMFault" name="fault" />
</message>
<message name="AUTH_ACCOUNT_EXISTMessage">
<part element="tns:AUTH_ACCOUNT_EXISTFault" name="fault" />
</message>
<message name="UNKNOWN_AUTH_ACCOUNTMessage">
<part element="tns:UNKNOWN_AUTH_ACCOUNTFault" name="fault" />
</message>
<portType name="VishnuUMSPortType">
<operation name="connect">
<documentation>opens a session</documentation>
<input message="tns:connectInput" name="connectInput" />
<output message="tns:connectOutput" name="connectOutput" />
<fault message="tns:UNKNOWN_USERMessage" name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
</fault>
<fault message="tns:UNKNOWN_CLOSURE_MODEMessage" name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
</fault>
<fault message="tns:INCORRECT_TIMEOUTMessage" name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:AUTHENTERRMessage" name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_USERMessage" name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
```

```
</fault>
<fault message="tns:UNKNOWN_CLOSURE_MODEMessage" name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
</fault>
<fault message="tns:INCORRECT_TIMEOUTMessage" name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:AUTHENTERRMessage" name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="reconnect">
<documentation>reconnects to a session that is still active</documentation>
<input message="tns:reconnectInput" name="reconnectInput" />
<output message="tns:reconnectOutput" name="reconnectOutput" />
<fault message="tns:UNKNOWN_USERMessage" name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:UNUSABLE_MACHINEMessage" name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_SESSION_IDMessage" name="UNKNOWN_SESSION_ID">
<documentation>The session Id is unknown</documentation>
</fault>
```

```
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:AUTHENTERRMessage" name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_USERMessage" name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:UNUSABLE_MACHINEMessage" name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_SESSION_IDMessage" name="UNKNOWN_SESSION_ID">
<documentation>The session Id is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:AUTHENTERRMessage" name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
```

```xml
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="close">
<documentation>closes the session</documentation>
<input message="tns:closeInput" name="closeInput" />
<output message="tns:closeOutput" name="closeOutput" />
<fault message="tns:COMMAND_RUNNINGMessage" name="COMMAND_RUNNING">
<documentation>Commands are running</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="changePassword">
<documentation>changes the password</documentation>
<input message="tns:changePasswordInput" name="changePasswordInput" />
<output message="tns:changePasswordOutput" name="changePasswordOutput" />
<fault message="tns:UNKNOWN_USERMessage" name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:READONLY_ACCOUNTMessage" name="READONLY_ACCOUNT">
<documentation>You can modify information. This account is read-only</documentation>
```

```
</fault>
<fault message="tns:AUTHENTERRMessage" name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="addLocalAccount">
<documentation>adds a new local user configuration </documentation>
<input message="tns:addLocalAccountInput" name="addLocalAccountInput" />
<output message="tns:addLocalAccountOutput" name="addLocalAccountOutput" />
<fault message="tns:LOCAL_ACCOUNT_EXISTMessage" name="LOCAL_ACCOUNT_EXIST">
<documentation>The local account already exists</documentation>
</fault>
<fault message="tns:MACHINE_LOCKEDMessage" name="MACHINE_LOCKED">
<documentation>The machine is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:LOGIN_ALREADY_USEDMessage" name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
```

```
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNUSABLE_MACHINEMessage" name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
</fault>
</operation>
<operation name="updateLocalAccount">
<documentation>updates a local user configuration</documentation>
<input message="tns:updateLocalAccountInput" name="updateLocalAccountInput" />
<output message="tns:updateLocalAccountOutput" name="updateLocalAccountOutput" />
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:LOGIN_ALREADY_USEDMessage" name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
</fault>
```

```
</operation>
<operation name="deleteLocalAccount">
<documentation>removes a local user configuration (for a given user on a given machine) from
VISHNU</documentation>
<input message="tns:deleteLocalAccountInput" name="deleteLocalAccountInput" />
<output message="tns:deleteLocalAccountOutput" name="deleteLocalAccountOutput" />
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listLocalAccounts">
<documentation>lists the local user configurations</documentation>
<input message="tns:listLocalAccountsInput" name="listLocalAccountsInput" />
<output message="tns:listLocalAccountsOutput" name="listLocalAccountsOutput" />
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
```

```
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listMachines">
<documentation>lists the machines that are accessible through VISHNU</documentation>
<input message="tns:listMachinesInput" name="listMachinesInput" />
<output message="tns:listMachinesOutput" name="listMachinesOutput" />
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listHistoryCmd">
<documentation>lists the commands</documentation>
<input message="tns:listHistoryCmdInput" name="listHistoryCmdInput" />
<output message="tns:listHistoryCmdOutput" name="listHistoryCmdOutput" />
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
```

```
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listOptions">
<documentation>lists the options of the user</documentation>
<input message="tns:listOptionsInput" name="listOptionsInput" />
<output message="tns:listOptionsOutput" name="listOptionsOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_OPTIONMessage" name="UNKNOWN_OPTION">
<documentation>The name of the user option is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
```

```
</operation>
<operation name="listSessions">
<documentation>lists all sessions of the user</documentation>
<input message="tns:listSessionsInput" name="listSessionsInput" />
<output message="tns:listSessionsOutput" name="listSessionsOutput" />
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_CLOSURE_MODEMessage" name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="configureOption">
<documentation>configures an option of the user</documentation>
<input message="tns:configureOptionInput" name="configureOptionInput" />
<output message="tns:configureOptionOutput" name="configureOptionOutput" />
<fault message="tns:UNKNOWN_OPTIONMessage" name="UNKNOWN_OPTION">
<documentation>The name of the user option is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_CLOSURE_MODEMessage" name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:INCORRECT_TIMEOUTMessage" name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
```

```
</fault>
<fault message="tns:INCORRECT_TRANSFER_CMDMessage"
name="INCORRECT_TRANSFER_CMD">
<documentation>The value of the transfer command is incorrect</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listAuthSystems">
<documentation>lists VISHNU user-authentification systems</documentation>
<input message="tns:listAuthSystemsInput" name="listAuthSystemsInput" />
<output message="tns:listAuthSystemsOutput" name="listAuthSystemsOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_SYSTEM_TYPEMessage"
name="UNKNOWN_AUTH_SYSTEM_TYPE">
<documentation>The type of the user-authentication system is unknown</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
```

```
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="addAuthAccount">
<documentation>adds a new local user-authentication configuration</documentation>
<input message="tns:addAuthAccountInput" name="addAuthAccountInput" />
<output message="tns:addAuthAccountOutput" name="addAuthAccountOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_SYSTEMMessage" name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
</fault>
<fault message="tns:AUTH_ACCOUNT_EXISTMessage" name="AUTH_ACCOUNT_EXIST">
<documentation>The user-authentication account already exists</documentation>
</fault>
<fault message="tns:LOGIN_ALREADY_USEDMessage" name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="updateAuthAccount">
```

```
<documentation>updates a local user-authentication configuration</documentation>
<input message="tns:updateAuthAccountInput" name="updateAuthAccountInput" />
<output message="tns:updateAuthAccountOutput" name="updateAuthAccountOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_SYSTEMMessage" name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_ACCOUNTMessage"
name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="deleteAuthAccount">
<documentation>removes a local user-authentication configuration from VISHNU</documentation>
<input message="tns:deleteAuthAccountInput" name="deleteAuthAccountInput" />
<output message="tns:deleteAuthAccountOutput" name="deleteAuthAccountOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
```

```
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_SYSTEMMessage" name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_ACCOUNTMessage"
name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
<operation name="listAuthAccounts">
<documentation>lists local user-authentication configurations</documentation>
<input message="tns:listAuthAccountsInput" name="listAuthAccountsInput" />
<output message="tns:listAuthAccountsOutput" name="listAuthAccountsOutput" />
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:SESSIONKEY_NOT_FOUNDMessage" name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
<fault message="tns:USER_LOCKEDMessage" name="USER_LOCKED">
<documentation>The user is locked</documentation>
</fault>
```

```
<fault message="tns:UNKNOWN_AUTH_SYSTEMMessage" name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
</fault>
<fault message="tns:UNKNOWN_AUTH_ACCOUNTMessage"
name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
</operation>
</portType>
<binding name="VishnuUMSSOAPBinding" type="tns:VishnuUMSPortType">
<soapbind:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="connect">
<documentation>opens a session</documentation>
<soapbind:operation soapAction="" />
<input name="connectInput">
<soapbind:body use="literal" />
</input>
<output name="connectOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
<soapbind:fault name="UNKNOWN_USER" use="literal" />
</fault>
<fault name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
<soapbind:fault name="UNKNOWN_CLOSURE_MODE" use="literal" />
</fault>
<fault name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
<soapbind:fault name="INCORRECT_TIMEOUT" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
```

```
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
<soapbind:fault name="AUTHENTERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
<soapbind:fault name="UNKNOWN_USER" use="literal" />
</fault>
<fault name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
<soapbind:fault name="UNKNOWN_CLOSURE_MODE" use="literal" />
</fault>
<fault name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
<soapbind:fault name="INCORRECT_TIMEOUT" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
```

```
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
<soapbind:fault name="AUTHENTERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="reconnect">
<documentation>reconnects to a session that is still active</documentation>
<soapbind:operation soapAction="" />
<input name="reconnectInput">
<soapbind:body use="literal" />
</input>
<output name="reconnectOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
<soapbind:fault name="UNKNOWN_USER" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
<soapbind:fault name="UNUSABLE_MACHINE" use="literal" />
</fault>
```

```
<fault name="UNKNOWN_SESSION_ID">
<documentation>The session Id is unknown</documentation>
<soapbind:fault name="UNKNOWN_SESSION_ID" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
<soapbind:fault name="AUTHENTERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
<soapbind:fault name="UNKNOWN_USER" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
<soapbind:fault name="UNUSABLE_MACHINE" use="literal" />
</fault>
<fault name="UNKNOWN_SESSION_ID">
<documentation>The session Id is unknown</documentation>
<soapbind:fault name="UNKNOWN_SESSION_ID" use="literal" />
```

```
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
<soapbind:fault name="AUTHENTERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="close">
<documentation>closes the session</documentation>
<soapbind:operation soapAction="" />
<input name="closeInput">
<soapbind:body use="literal" />
</input>
<output name="closeOutput">
<soapbind:body use="literal" />
</output>
<fault name="COMMAND_RUNNING">
<documentation>Commands are running</documentation>
<soapbind:fault name="COMMAND_RUNNING" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
```

```
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="changePassword">
<documentation>changes the password</documentation>
<soapbind:operation soapAction="" />
<input name="changePasswordInput">
<soapbind:body use="literal" />
</input>
<output name="changePasswordOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USER">
<documentation>The user is unknown or the password is wrong</documentation>
<soapbind:fault name="UNKNOWN_USER" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="READONLY_ACCOUNT">
<documentation>You can modify information. This account is read-only</documentation>
<soapbind:fault name="READONLY_ACCOUNT" use="literal" />
</fault>
<fault name="AUTHENTERR">
<documentation>Vishnu not available (Authenticator error)</documentation>
<soapbind:fault name="AUTHENTERR" use="literal" />
</fault>
<fault name="DIET">
```

```
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="addLocalAccount">
<documentation>adds a new local user configuration </documentation>
<soapbind:operation soapAction="" />
<input name="addLocalAccountInput">
<soapbind:body use="literal" />
</input>
<output name="addLocalAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="LOCAL_ACCOUNT_EXIST">
<documentation>The local account already exists</documentation>
<soapbind:fault name="LOCAL_ACCOUNT_EXIST" use="literal" />
</fault>
<fault name="MACHINE_LOCKED">
<documentation>The machine is locked</documentation>
<soapbind:fault name="MACHINE_LOCKED" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
```

```
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
<soapbind:fault name="LOGIN_ALREADY_USED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNUSABLE_MACHINE">
<documentation>The machine does not exist or it is locked</documentation>
<soapbind:fault name="UNUSABLE_MACHINE" use="literal" />
</fault>
</operation>
<operation name="updateLocalAccount">
<documentation>updates a local user configuration</documentation>
<soapbind:operation soapAction="" />
<input name="updateLocalAccountInput">
<soapbind:body use="literal" />
</input>
<output name="updateLocalAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
```

```
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
<soapbind:fault name="UNKNOWN_LOCAL_ACCOUNT" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
<soapbind:fault name="LOGIN_ALREADY_USED" use="literal" />
</fault>
</operation>
<operation name="deleteLocalAccount">
<documentation>removes a local user configuration (for a given user on a given machine) from
VISHNU</documentation>
<soapbind:operation soapAction="" />
<input name="deleteLocalAccountInput">
<soapbind:body use="literal" />
</input>
<output name="deleteLocalAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
<soapbind:fault name="UNKNOWN_LOCAL_ACCOUNT" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
```

```
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listLocalAccounts">
<documentation>lists the local user configurations</documentation>
<soapbind:operation soapAction="" />
<input name="listLocalAccountsInput">
<soapbind:body use="literal" />
</input>
<output name="listLocalAccountsOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
```

```
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listMachines">
<documentation>lists the machines that are accessible through VISHNU</documentation>
<soapbind:operation soapAction="" />
<input name="listMachinesInput">
<soapbind:body use="literal" />
</input>
<output name="listMachinesOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
```

```
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listHistoryCmd">
<documentation>lists the commands</documentation>
<soapbind:operation soapAction="" />
<input name="listHistoryCmdInput">
<soapbind:body use="literal" />
</input>
<output name="listHistoryCmdOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listOptions">
```

```
<documentation>lists the options of the user</documentation>
<soapbind:operation soapAction="" />
<input name="listOptionsInput">
<soapbind:body use="literal" />
</input>
<output name="listOptionsOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="UNKNOWN_OPTION">
<documentation>The name of the user option is unknown</documentation>
<soapbind:fault name="UNKNOWN_OPTION" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listSessions">
<documentation>lists all sessions of the user</documentation>
```

```
<soapbind:operation soapAction="" />
<input name="listSessionsInput">
<soapbind:body use="literal" />
</input>
<output name="listSessionsOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
<soapbind:fault name="UNKNOWN_CLOSURE_MODE" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="configureOption">
<documentation>configures an option of the user</documentation>
<soapbind:operation soapAction="" />
<input name="configureOptionInput">
<soapbind:body use="literal" />
</input>
<output name="configureOptionOutput">
```

```
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_OPTION">
<documentation>The name of the user option is unknown</documentation>
<soapbind:fault name="UNKNOWN_OPTION" use="literal" />
</fault>
<fault name="UNKNOWN_CLOSURE_MODE">
<documentation>The closure policy is unknown</documentation>
<soapbind:fault name="UNKNOWN_CLOSURE_MODE" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="INCORRECT_TIMEOUT">
<documentation>The value of the timeout is incorrect</documentation>
<soapbind:fault name="INCORRECT_TIMEOUT" use="literal" />
</fault>
<fault name="INCORRECT_TRANSFER_CMD">
<documentation>The value of the transfer command is incorrect</documentation>
<soapbind:fault name="INCORRECT_TRANSFER_CMD" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listAuthSystems">
<documentation>lists VISHNU user-authentification systems</documentation>
<soapbind:operation soapAction="" />
<input name="listAuthSystemsInput">
```

```
<soapbind:body use="literal" />
</input>
<output name="listAuthSystemsOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_SYSTEM_TYPE">
<documentation>The type of the user-authentication system is unknown</documentation>
<soapbind:fault name="UNKNOWN_AUTH_SYSTEM_TYPE" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="addAuthAccount">
<documentation>adds a new local user-authentication configuration</documentation>
<soapbind:operation soapAction="" />
<input name="addAuthAccountInput">
<soapbind:body use="literal" />
```

```
</input>
<output name="addAuthAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
<soapbind:fault name="UNKNOWN_AUTH_SYSTEM" use="literal" />
</fault>
<fault name="AUTH_ACCOUNT_EXIST">
<documentation>The user-authentication account already exists</documentation>
<soapbind:fault name="AUTH_ACCOUNT_EXIST" use="literal" />
</fault>
<fault name="LOGIN_ALREADY_USED">
<documentation>The system account login is already used by another vishnu user</documentation>
<soapbind:fault name="LOGIN_ALREADY_USED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
```

```
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="updateAuthAccount">
<documentation>updates a local user-authentication configuration</documentation>
<soapbind:operation soapAction="" />
<input name="updateAuthAccountInput">
<soapbind:body use="literal" />
</input>
<output name="updateAuthAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
<soapbind:fault name="UNKNOWN_AUTH_SYSTEM" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
<soapbind:fault name="UNKNOWN_AUTH_ACCOUNT" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
```

```
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="deleteAuthAccount">
<documentation>removes a local user-authentication configuration from VISHNU</documentation>
<soapbind:operation soapAction="" />
<input name="deleteAuthAccountInput">
<soapbind:body use="literal" />
</input>
<output name="deleteAuthAccountOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
<soapbind:fault name="UNKNOWN_AUTH_SYSTEM" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
<soapbind:fault name="UNKNOWN_AUTH_ACCOUNT" use="literal" />
</fault>
<fault name="DIET">
```

```
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="listAuthAccounts">
<documentation>lists local user-authentication configurations</documentation>
<soapbind:operation soapAction="" />
<input name="listAuthAccountsInput">
<soapbind:body use="literal" />
</input>
<output name="listAuthAccountsOutput">
<soapbind:body use="literal" />
</output>
<fault name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
<soapbind:fault name="NO_ADMIN" use="literal" />
</fault>
<fault name="SESSIONKEY_NOT_FOUND">
<documentation>The session key is unrecognized</documentation>
<soapbind:fault name="SESSIONKEY_NOT_FOUND" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
<fault name="USER_LOCKED">
<documentation>The user is locked</documentation>
<soapbind:fault name="USER_LOCKED" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_SYSTEM">
<documentation>The user-authentication system is unknown or locked</documentation>
```

```
<soapbind:fault name="UNKNOWN_AUTH_SYSTEM" use="literal" />
</fault>
<fault name="UNKNOWN_AUTH_ACCOUNT">
<documentation>The user-authentication account is unknown</documentation>
<soapbind:fault name="UNKNOWN_AUTH_ACCOUNT" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
</binding>
<service name="VishnuUMSService">
<port binding="tns:VishnuUMSSOAPBinding" name="VishnuUMSPort">
<soapbind:address location="http://127.0.0.1:8080/ResourceProxy/VishnuUMS" />
</port>
</service>
</definitions>
```

About *wsdl-viewer.xsl*

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

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

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