

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

Table of Contents

<u>Web Service: VishnuFMSService</u>	1
<u>Web Service: VishnuFMSService</u>	1
<u>Port VishnuFMSPort Port typeSource code</u>	1
<u>Operations</u>	2
<u>Port type VishnuFMSPortTypeSource code</u>	2
<u>WSDL source code</u>	24
<u>About wsdl-viewer.xsl</u>	81

Web Service: VishnuFMSService

Web Service: VishnuFMSService

Target Namespace:

urn:ResourceProxy

Port VishnuFMSPort Port typeSource code

Location:

http://127.0.0.1:8080/ResourceProxy/VishnuFMS

Protocol:

SOAP

Default style:

document

Transport protocol:

SOAP over HTTP

Operations:

1. *chGrp*DetailSource code
2. *chMod*DetailSource code
3. *contentOfFile*DetailSource code
4. *copyAsyncFile*DetailSource code
5. *copyFile*DetailSource code
6. *createDir*DetailSource code
7. *createFile*DetailSource code
8. *getFileInfo*DetailSource code
9. *headOfFile*DetailSource code
10. *listDir*DetailSource code
11. *listFileTransfers*DetailSource code
12. *moveAsyncFile*DetailSource code
13. *moveFile*DetailSource code
14. *removeDir*DetailSource code
15. *removeFile*DetailSource code
16. *stopFileTransfer*DetailSource code
17. *tailOfFile*DetailSource code

Operations

Port type VishnuFMSPortTypeSource code

1. **chGrp**

Source code

Description:

changes group owner of remote files/directories.

Operation type:

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

Input:

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

parameters type *chGrpRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **group** type *string*
the new group owner of file/directory
- ◆ **path** type *string*
The file/directory following the pattern [host:]file path

Output:

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

parameters type *chGrpResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)Source code

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)Source code

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)Source code

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)Source code

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)Source code

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)Source code

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)Source code

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

2. **chMod**

[Source code](#)

Description:

changes access rights of remote files/directories.

Operation type:

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

Input:

chModInput (soapbind:body, use = literal)[Source code](#)

parameters type *chModRequest*

- ◆ sessionKey type *string*
The session key
- ◆ mode type *integer*
the access rights of file/directory in octal system.
- ◆ path type *string*
The file/directory following the pattern [host:]file path

Output:

chModOutput (soapbind:body, use = literal)[Source code](#)

parameters type *chModResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

3. **contentOfFile**

[Source code](#)

Description:

displays content of files located on remote machines

Operation type:

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

Input:

contentOfFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *contentOfFileRequest*

◆ sessionKey type *string*

The session key

◆ path type *string*

The file to display following the pattern [host:]file path

Output:

contentOfFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *contentOfFileResponse*

◆ fileContent type *string*

The content of the file

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

4. **copyAsyncFile**

[Source code](#)

Description:

executes an asynchronous copy of file.

Operation type:

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

Input:

copyAsyncFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *copyAsyncFileRequest*

- ◆ sessionKey type *string*
The session key
- ◆ src type *string*
The source file to copy following the pattern [host:]file path
- ◆ dest type *string*
The path of the destination file
- ◆ isRecursive type *boolean*
It specifies when the copy is recursive (case of directory) or not.
- ◆ trCommand type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }
the command to use to perform file transfer.

Output:

copyAsyncFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *copyAsyncFileResponse*

- ◆ transferId type *string*

The file transfer identifier

- ◆ status type *Status* - type *string* with restriction - enum { 'INPROGRESS', 'COMPLETED', 'CANCELLED', 'FAILED', 'UNDEFINED' }

The file transfer status

- ◆ userId type *string*

The user identifier

- ◆ clientMachineId type *string*

The client machine identifier

- ◆ sourceMachineId type *string*

The source machine identifier

- ◆ destinationMachineId type *string*

The destination machine identifier

- ◆ sourceFilePath type *string*

The source file path

- ◆ destinationFilePath type *string*

The destination file path

- ◆ size type *long*

The size of the file

- ◆ start_time type *long*

The start time of the file transfer

- ◆ trCommand type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }

The command used for the file transfer (scp or rsync)

- ◆ errorMsg type *string*

The eventual error message if the file transfer failed

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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_FOUND*Fault

Fault:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINED*Fault

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFER*Fault

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNT*Fault

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINE*Fault

5. **copyFile**

[Source code](#)

Description:

executes a synchronous copy of file.

Operation type:

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

Input:

copyFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *copyFileRequest*

- ◆ sessionKey type *string*

The session key

- ◆ src type *string*

The source file to copy following the pattern [host:]file path

- ◆ dest type *string*

The path of the destination file

- ◆ isRecursive - optional; type *boolean*

It specifies when the copy is recursive (case of directory) or not.

- ◆ trCommand - optional; type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }

the command to use to perform file transfer.

Output:

copyFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *copyFileResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUND*Fault

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERR*Fault

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIET*Fault

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAM*Fault

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

6. **createDir**

[Source code](#)

Description:

creates directories on remote machines.

Operation type:

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

Input:

createDirInput (soapbind:body, use = literal)[Source code](#)

parameters type *createDirRequest*

◆ sessionKey type *string*

The session key

◆ path type *string*

The directory to create following the pattern [host:]directory path.

◆ isRecursive - optional; type *boolean*

It specifies when the create command is recursive (create parent directory also) or not

Output:

createDirOutput (soapbind:body, use = literal)[Source code](#)

parameters type *createDirResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

7. **createFile**

[Source code](#)

Description:

creates files on remote machines.

Operation type:

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

Input:

createFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *createFileRequest*

◆ sessionKey type *string*

The session key.

◆ path type *string*

The file to create following the pattern [host:]file path.

Output:

createFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *createFileResponse*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_USERIDFault*

8. **getFileInfo**

[Source code](#)

Description:

displays the information of files.

Operation type:

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

Input:

getFileInfoInput (soapbind:body, use = literal)[Source code](#)

parameters type *getFileInfoRequest*

◆ sessionKey type *string*

The session key

◆ path type *string*

The file whose inode information will be displayed

Output:

getFileInfoOutput (soapbind:body, use = literal)[Source code](#)

parameters type *getFileInfoResponse*

◆ path type *string*

The path of the file

◆ owner type *string*

The name of the owner of the file

◆ group type *string*

The group name of the owner

◆ perms type *integer*

The protection of the file

◆ uid type *long*

The user identifier of the owner

◆ gid type *long*

The group identifier of the owner

◆ size type *long*

The size of the file in bytes

◆ atime type *long*

The time of last access

◆ mtime type *long*

The time of last modification

◆ ctime type *long*

The time of the last change of the inode information

◆ type type *FileType* - type *string* with restriction - enum { 'BLOCK', 'CHARACTER', 'DIRECTORY', 'SYMBOLICLINK', 'SCKT', 'FIFO', 'REGULAR' }

The file type

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

9. **headOfFile**

[Source code](#)

Description:

displays a few first lines of files located on remote machines.

Operation type:

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

Input:

headOfFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *headOfFileRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **path** type *string*
The file following the pattern [host:]file path
- ◆ **nline** - optional; type *integer*
the number of line to get

Output:

headOfFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *headOfFileResponse*

- ◆ **fileContent** type *string*
The first "nLine" lines of the file

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

10. **listDir**

[Source code](#)

Description:

displays the content of a remote directory.

Operation type:

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

Input:

listDirInput (soapbind:body, use = literal)[Source code](#)

parameters type *listDirRequest*

- ◆ sessionKey type *string*
The session key
- ◆ path type *string*
The directory to list following the pattern [host:]directory path
- ◆ longFormat type *boolean*
It specifies the long display format (all available file informations)
- ◆ allFiles type *boolean*
Allows to display all files including hidden files

Output:

listDirOutput (soapbind:body, use = literal)[Source code](#)

parameters type *listDirResponse*

- ◆ data - optional;
- ◆ direntry - optional, unbounded, nillable;
 - path - optional; type *string*
The path of the remote file

- **owner** - optional; type *string*
The remote file owner's name
- **group** - optional; type *string*
The group name of the owner
- **perms** - optional; type *integer*
The protection of the remote file
- **size** - optional; type *long*
The size of the remote file in bytes
- **creationTime** - optional; type *string*
The file creation time (in remote host time)
- **type** - optional; type *FileType* - type *string* with restriction - enum
{ 'BLOCK', 'CHARACTER', 'DIRECTORY', 'SYMBOLICLINK',
'SCKT', 'FIFO', 'REGULAR' }
The file type

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type UNKNOWN_MACHINEFault

11. listFileTransfers

[Source code](#)

Description:

displays the history of all file transfers submitted by User.

Operation type:

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

Input:

listFileTransfersInput (soapbind:body, use = literal)[Source code](#)

parameters type listFileTransfersRequest

- ◆ sessionKey type *string*
The session key
- ◆ transferId - optional; type *string*
a given transfer id
- ◆ fromMachineId - optional; type *string*
the machine that is the source of the file transfer
- ◆ userId - optional; type *string*
allows the admin to list file transfers initiated by a specific user
- ◆ status - optional; type *Status* - type *string* with restriction - enum { 'INPROGRESS', 'COMPLETED', 'CANCELLED', 'FAILED', 'UNDEFINED' }
the file transfer status

Output:

listFileTransfersOutput (soapbind:body, use = literal)[Source code](#)

parameters type listFileTransfersResponse

- ◆ data - optional;
 - ◆ filetransfer - optional, unbounded, nillable;
 - transferId - optional; type *string*
The file transfer identifier
 - status - optional; type *Status* - type *string* with restriction - enum { 'INPROGRESS', 'COMPLETED', 'CANCELLED', 'FAILED', 'UNDEFINED' }
The file transfer status
 - userId - optional; type *string*
The user identifier
 - clientMachineId - optional; type *string*
The client machine identifier
 - sourceMachineId - optional; type *string*
The source machine identifier
 - destinationMachineId - optional; type *string*
The destination machine identifier
 - sourceFilePath - optional; type *string*
The source file path
 - destinationFilePath - optional; type *string*
The destination file path
 - size - optional; type *long*
The size of the file
 - start_time - optional; type *long*
The start time of the file transfer
 - trCommand - optional; type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }

The command used for the file transfer (scp or rsync)

- **errorMsg** - optional; type *string*

The eventual error message if the file transfer failed

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_USERIDFault*

12. **moveAsyncFile**

[Source code](#)

Description:

executes an asynchronous move of file.

Operation type:

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

Input:

moveAsyncFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *moveAsyncFileRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **src** type *string*
The source file to move following the pattern [host:]file path
- ◆ **dest** type *string*
The path of the destination file
- ◆ **isRecursive** - optional; type *boolean*
It specifies when the copy is recursive (case of directory) or not.
- ◆ **trCommand** - optional; type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }
the command to use to perform file transfer.

Output:

moveAsyncFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *moveAsyncFileResponse*

- ◆ **transferId** type *string*
The file transfer identifier
- ◆ **status** type *Status* - type *string* with restriction - enum { 'INPROGRESS', 'COMPLETED', 'CANCELLED', 'FAILED', 'UNDEFINED' }
The file transfer status
- ◆ **userId** type *string*
The user identifier
- ◆ **clientMachineId** type *string*
The client machine identifier
- ◆ **sourceMachineId** type *string*
The source machine identifier
- ◆ **destinationMachineId** type *string*
The destination machine identifier
- ◆ **sourceFilePath** type *string*
The source file path
- ◆ **destinationFilePath** type *string*
The destination file path
- ◆ **size** type *long*
The size of the file
- ◆ **start_time** type *long*
The start time of the file transfer
- ◆ **trCommand** type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }
The command used for the file transfer (scp or rsync)
- ◆ **errorMsg** type *string*
The eventual error message if the file transfer failed

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

13. **moveFile**

[Source code](#)

Description:

executes a synchronous move of file.

Operation type:

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

Input:

moveFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *moveFileRequest*

◆ sessionKey type *string*

The session key

◆ src type *string*

The source file to move following the pattern [host:]file path

◆ dest type *string*

The path of the destination file

◆ isRecursive type *boolean*

It specifies when the copy is recursive (case of directory) or not.

◆ trCommand type *TransferCommand* - type *string* with restriction - enum { 'SCP', 'RSYNC', 'UNDEFINED' }

the command to use to perform file transfer.

Output:

moveFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *moveFileResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

14. **removeDir**

[Source code](#)

Description:

removes directories (and subdirectories) from remote machines.

Operation type:

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

Input:

removeDirInput (soapbind:body, use = literal)[Source code](#)

parameters type *removeDirRequest*

◆ sessionKey type *string*

The session key

◆ path type *string*

The directory to remove following the pattern [host:]directory path

Output:

removeDirOutput (soapbind:body, use = literal)[Source code](#)

parameters type *removeDirResponse*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

15. **removeFile**

[Source code](#)

Description:

removes files from remote hosts.

Operation type:

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

Input:

removeFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *removeFileRequest*

◆ sessionKey type *string*

The session key

◆ path type *string*

The file to remove following the pattern [host:]file path

◆ isRecursive - optional; type *boolean*

It specifies when the remove command is recursive (case of directory) or not

Output:

removeFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *removeFileResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

16. **stopFileTransfer**

[Source code](#)

Description:

stops an execution of a set of file transfers.

Operation type:

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

Input:

stopFileTransferInput (soapbind:body, use = literal)[Source code](#)

parameters type *stopFileTransferRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **transferId** - optional; type *string*
a given transfer id
- ◆ **fromMachineId** - optional; type *string*
the machine that is the source or destination of the file transfer
- ◆ **userId** - optional; type *string*
allows an admin to stop file transfers of a specific user

Output:

stopFileTransferOutput (soapbind:body, use = literal)[Source code](#)

parameters type *stopFileTransferResponse*

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUNDFault*

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERRFault*

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIETFault*

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAMFault*

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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_FOUND*Fault

Fault:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINED*Fault

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFER*Fault

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNT*Fault

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINE*Fault

Fault:

UNKNOWN_USERIDMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_USERID*Fault

17. **tailOfFile**

[Source code](#)

Description:

displays a few last lines of files located on remote machines

Operation type:

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

Input:

tailOfFileInput (soapbind:body, use = literal)[Source code](#)

parameters type *tailOfFileRequest*

- ◆ sessionKey type *string*
The session key
- ◆ path type *string*
The file following the pattern [host:]file path
- ◆ nline - optional; type *integer*
the number of line to get

Output:

tailOfFileOutput (soapbind:body, use = literal)[Source code](#)

parameters type *tailOfFileResponse*

- ◆ fileContent type *string*
The last "nLine" lines of the file

Fault:

CONFIGNOTFOUNDMessage (documentation, use = literal)[Source code](#)

fault type *CONFIGNOTFOUND*Fault

Fault:

DBERRMessage (documentation, use = literal)[Source code](#)

fault type *DBERR*Fault

Fault:

DIETMessage (documentation, use = literal)[Source code](#)

fault type *DIET*Fault

Fault:

INVALID_PARAMMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PARAM*Fault

Fault:

INVALID_PATHMessage (documentation, use = literal)[Source code](#)

fault type *INVALID_PATHFault*

Fault:

NO_ADMINMessage (documentation, use = literal)[Source code](#)

fault type *NO_ADMINFault*

Fault:

RUNTIME_ERRORMessage (documentation, use = literal)[Source code](#)

fault type *RUNTIME_ERRORFault*

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:

UNDEFINEDMessage (documentation, use = literal)[Source code](#)

fault type *UNDEFINEDFault*

Fault:

UNKNOWN_FILE_TRANSFERMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_FILE_TRANSFERFault*

Fault:

UNKNOWN_LOCAL_ACCOUNTMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_LOCAL_ACCOUNTFault*

Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)[Source code](#)

fault type *UNKNOWN_MACHINEFault*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="VishnuFMS" 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="createFileRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
```

```

<xs:documentation>The file to create following the pattern [host:]file path.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="createFileResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="createDirRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The directory to create following the pattern [host:]directory path.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="isRecursive" type="xs:boolean">
<xs:annotation>
<xs:documentation>It specifies when the create command is recursive (create parent directory also) or
not</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="createDirResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="removeFileRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The file to remove following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>

```

```

<xs:element minOccurs="0" name="isRecursive" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>It specifies when the remove command is recursive (case of directory) or not</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="removeFileResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="removeDirRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="path" type="xs:string">
        <xs:annotation>
          <xs:documentation>The directory to remove following the pattern [host:]directory path</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="removeDirResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="chGrpRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="group" type="xs:string">
        <xs:annotation>
          <xs:documentation>the new group owner of file/directory</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="path" type="xs:string">
        <xs:annotation>
          <xs:documentation>The file/directory following the pattern [host:]file path</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="chGrpResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="chModRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="mode" type="xs:integer">
<xs:annotation>
<xs:documentation> the access rigths of file/directory in octal system.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The file/directory following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="chModResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="headOfFileRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The file following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="nline" type="xs:integer">
<xs:annotation>

```

```

<xs:documentation>the number of line to get</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="headOfFileResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="fileContent" type="xs:string">
<xs:annotation>
<xs:documentation>The first "nLine" lines of the file</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="tailOfFileRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The file following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="nline" type="xs:integer">
<xs:annotation>
<xs:documentation>the number of line to get</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="tailOfFileResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="fileContent" type="xs:string">
<xs:annotation>
<xs:documentation>The last "nLine" lines of the file</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="contentOfFileRequest">
<xs:complexType>

```

```

<xs:sequence>
  <xs:element minOccurs="1" name="sessionKey" type="xs:string">
    <xs:annotation>
      <xs:documentation>The session key</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="1" name="path" type="xs:string">
    <xs:annotation>
      <xs:documentation>The file to display following the pattern [host:]file path</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="contentOfFileResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="fileContent" type="xs:string">
        <xs:annotation>
          <xs:documentation>The content of the file</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="listDirRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="path" type="xs:string">
        <xs:annotation>
          <xs:documentation>The directory to list following the pattern [host:]directory path</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="longFormat" type="xs:boolean">
        <xs:annotation>
          <xs:documentation>It specifies the long display format (all available file informations)</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="allFiles" type="xs:boolean">
        <xs:annotation>
          <xs:documentation>Allows to display all files including hidden files</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="listDirResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="data">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="direntry" nillable="true">
              <xs:complexType>
                <xs:sequence />
                <xs:attribute name="path" type="xs:string" use="optional">
                  <xs:annotation>
                    <xs:documentation>The path of the remote file</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="owner" type="xs:string" use="optional">
                  <xs:annotation>
                    <xs:documentation>The remote file owner's name</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="group" type="xs:string" use="optional">
                  <xs:annotation>
                    <xs:documentation>The group name of the owner</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="perms" type="xs:integer" use="optional">
                  <xs:annotation>
                    <xs:documentation>The protection of the remote file</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="size" type="xs:long" use="optional">
                  <xs:annotation>
                    <xs:documentation>The size of the remote file in bytes</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="creationTime" type="xs:string" use="optional">
                  <xs:annotation>
                    <xs:documentation>The file creation time (in remote host time)</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
                <xs:attribute name="type" type="Q1:FileType" use="optional">
                  <xs:annotation>
                    <xs:documentation>The file type</xs:documentation>
                  </xs:annotation>
                </xs:attribute>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

</xs:element>
<xs:simpleType name="FileType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BLOCK" />
    <xs:enumeration value="CHARACTER" />
    <xs:enumeration value="DIRECTORY" />
    <xs:enumeration value="SYMBOLICLINK" />
    <xs:enumeration value="SCKT" />
    <xs:enumeration value="FIFO" />
    <xs:enumeration value="REGULAR" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="copyFileRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="src" type="xs:string">
        <xs:annotation>
          <xs:documentation>The source file to copy following the pattern [host:]file path</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="dest" type="xs:string">
        <xs:annotation>
          <xs:documentation>The path of the destination file</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" name="isRecursive" type="xs:boolean">
        <xs:annotation>
          <xs:documentation>It specifies when the copy is recursive (case of directory) or not.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" name="trCommand" type="Q1:TransferCommand">
        <xs:annotation>
          <xs:documentation>the command to use to perform file transfer.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="TransferCommand">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SCP" />
    <xs:enumeration value="RSYNC" />
    <xs:enumeration value="UNDEFINED" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="copyFileResponse">

```

```

<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="copyAsyncFileRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="src" type="xs:string">
<xs:annotation>
<xs:documentation>The source file to copy following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dest" type="xs:string">
<xs:annotation>
<xs:documentation>The path of the destination file</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="isRecursive" type="xs:boolean">
<xs:annotation>
<xs:documentation>It specifies when the copy is recursive (case of directory) or not.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="trCommand" type="Q1:TransferCommand">
<xs:annotation>
<xs:documentation>the command to use to perform file transfer.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="copyAsyncFileResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="transferId" type="xs:string">
<xs:annotation>
<xs:documentation>The file transfer identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:Status">
<xs:annotation>
<xs:documentation>The file transfer status</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>The user identifier</xs:documentation>

```

```

</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="clientMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The client machine identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sourceMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The source machine identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="destinationMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The destination machine identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sourceFilePath" type="xs:string">
<xs:annotation>
<xs:documentation>The source file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="destinationFilePath" type="xs:string">
<xs:annotation>
<xs:documentation>The destination file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="size" type="xs:long">
<xs:annotation>
<xs:documentation>The size of the file </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="start_time" type="xs:long">
<xs:annotation>
<xs:documentation>The start time of the file transfer</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="trCommand" type="Q1:TransferCommand">
<xs:annotation>
<xs:documentation>The command used for the file transfer (scp or rsync)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="errorMsg" type="xs:string">
<xs:annotation>
<xs:documentation>The eventual error message if the file transfer failed</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="Status">

```

```

<xs:restriction base="xs:string">
  <xs:enumeration value="INPROGRESS" />
  <xs:enumeration value="COMPLETED" />
  <xs:enumeration value="CANCELLED" />
  <xs:enumeration value="FAILED" />
  <xs:enumeration value="UNDEFINED" />
</xs:restriction>
</xs:simpleType>
<xs:element name="moveFileRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="src" type="xs:string">
        <xs:annotation>
          <xs:documentation>The source file to move following the pattern [host:]file path</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="dest" type="xs:string">
        <xs:annotation>
          <xs:documentation>The path of the destination file</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="isRecursive" type="xs:boolean">
        <xs:annotation>
          <xs:documentation>It specifies when the copy is recursive (case of directory) or not.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="trCommand" type="Q1:TransferCommand">
        <xs:annotation>
          <xs:documentation>the command to use to perform file transfer.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="moveFileResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="moveAsyncFileRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="sessionKey" type="xs:string">
        <xs:annotation>
          <xs:documentation>The session key</xs:documentation>
        </xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="1" name="src" type="xs:string">
<xs:annotation>
<xs:documentation>The source file to move following the pattern [host:]file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="dest" type="xs:string">
<xs:annotation>
<xs:documentation>The path of the destination file</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="isRecursive" type="xs:boolean">
<xs:annotation>
<xs:documentation>It specifies when the copy is recursive (case of directory) or not.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="trCommand" type="Q1:TransferCommand">
<xs:annotation>
<xs:documentation>the command to use to perform file transfer.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="moveAsyncFileResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="transferId" type="xs:string">
<xs:annotation>
<xs:documentation>The file transfer identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:Status">
<xs:annotation>
<xs:documentation>The file transfer status</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>The user identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="clientMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The client machine identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sourceMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The source machine identifier</xs:documentation>
</xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="1" name="destinationMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>The destination machine identifier</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="sourceFilePath" type="xs:string">
<xs:annotation>
<xs:documentation>The source file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="destinationFilePath" type="xs:string">
<xs:annotation>
<xs:documentation>The destination file path</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="size" type="xs:long">
<xs:annotation>
<xs:documentation>The size of the file </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="start_time" type="xs:long">
<xs:annotation>
<xs:documentation>The start time of the file transfer</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="trCommand" type="Q1:TransferCommand">
<xs:annotation>
<xs:documentation>The command used for the file transfer (scp or rsync)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="errorMsg" type="xs:string">
<xs:annotation>
<xs:documentation>The eventual error message if the file transfer failed</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="stopFileTransferRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="transferId" type="xs:string">
<xs:annotation>
<xs:documentation>a given transfer id</xs:documentation>
</xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="0" name="fromMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>the machine that is the source or destination of the file transfer</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>allows an admin to stop file transfers of a specific user</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="stopFileTransferResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="listFileTransfersRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="transferId" type="xs:string">
<xs:annotation>
<xs:documentation>a given transfer id</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="fromMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>the machine that is the source of the file transfer</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="userId" type="xs:string">
<xs:annotation>
<xs:documentation>allows the admin to list file transfers initiated by a specific user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="status" type="Q1:Status">
<xs:annotation>
<xs:documentation>the file transfer status</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listFileTransfersResponse">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="0" name="data">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="filetransfer" nillable="true">
            <xs:complexType>
              <xs:sequence />
              <xs:attribute name="transferId" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The file transfer identifier</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="status" type="Q1:Status" use="optional">
                <xs:annotation>
                  <xs:documentation>The file transfer status</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="userId" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The user identifier</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="clientMachineId" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The client machine identifier</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="sourceMachineId" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The source machine identifier</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="destinationMachineId" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The destination machine identifier</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="sourceFilePath" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The source file path</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="destinationFilePath" type="xs:string" use="optional">
                <xs:annotation>
                  <xs:documentation>The destination file path</xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="size" type="xs:long" use="optional">
                <xs:annotation>
                  <xs:documentation>The size of the file</xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:sequence>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```



```

</xs:annotation>
</xs:attribute>
<xs:attribute name="start_time" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>The start time of the file transfer</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="trCommand" type="Q1:TransferCommand" use="optional">
<xs:annotation>
<xs:documentation>The command used for the file transfer (scp or rsync)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="errorMsg" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>The eventual error message if the file transfer failed</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="getFileInfoRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The file whose inode information will be displayed</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getFileInfoResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="path" type="xs:string">
<xs:annotation>
<xs:documentation>The path of the file</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="owner" type="xs:string">
<xs:annotation>

```

```

<xs:documentation>The name of the owner of the file</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="group" type="xs:string">
<xs:annotation>
<xs:documentation>The group name of the owner</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="perms" type="xs:integer">
<xs:annotation>
<xs:documentation>The protection of the file</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="uid" type="xs:long">
<xs:annotation>
<xs:documentation>The user identifier of the owner</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="gid" type="xs:long">
<xs:annotation>
<xs:documentation>The group identifier of the owner</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="size" type="xs:long">
<xs:annotation>
<xs:documentation>The size of the file in bytes</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="atime" type="xs:long">
<xs:annotation>
<xs:documentation>The time of last access</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="mtime" type="xs:long">
<xs:annotation>
<xs:documentation>The time of last modification</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="ctime" type="xs:long">
<xs:annotation>
<xs:documentation>The time of the last change of the inode information</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="type" type="Q1:FileType">
<xs:annotation>
<xs:documentation>The file type</xs:documentation>
</xs:annotation>
</xs:element>
</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="UNDEFINEDFault">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="INVALID_PARAMFault">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="CONFIGNOTFOUNDFault">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="NO_ADMINFault">
  <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_MACHINEFault">
  <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="INVALID_PATHFault">

```

```

<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="RUNTIME_ERRORFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_FILE_TRANSFERFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_USERIDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
</xs:schema>
</types>
<message name="createFileInput">
<part element="tns:createFileRequest" name="parameters" />
</message>
<message name="createFileOutput">
<part element="tns:createFileResponse" name="parameters" />
</message>
<message name="createDirInput">
<part element="tns:createDirRequest" name="parameters" />
</message>
<message name="createDirOutput">
<part element="tns:createDirResponse" name="parameters" />
</message>
<message name="removeFileInput">
<part element="tns:removeFileRequest" name="parameters" />
</message>
<message name="removeFileOutput">
<part element="tns:removeFileResponse" name="parameters" />
</message>
<message name="removeDirInput">
<part element="tns:removeDirRequest" name="parameters" />
</message>
<message name="removeDirOutput">
<part element="tns:removeDirResponse" name="parameters" />
</message>
<message name="chGrpInput">
<part element="tns:chGrpRequest" name="parameters" />
</message>
<message name="chGrpOutput">
<part element="tns:chGrpResponse" name="parameters" />
</message>

```

```

<message name="chModInput">
  <part element="tns:chModRequest" name="parameters" />
</message>
<message name="chModOutput">
  <part element="tns:chModResponse" name="parameters" />
</message>
<message name="headOfFileInput">
  <part element="tns:headOfFileRequest" name="parameters" />
</message>
<message name="headOfFileOutput">
  <part element="tns:headOfFileResponse" name="parameters" />
</message>
<message name="tailOfFileInput">
  <part element="tns:tailOfFileRequest" name="parameters" />
</message>
<message name="tailOfFileOutput">
  <part element="tns:tailOfFileResponse" name="parameters" />
</message>
<message name="contentOfFileInput">
  <part element="tns:contentOfFileRequest" name="parameters" />
</message>
<message name="contentOfFileOutput">
  <part element="tns:contentOfFileResponse" name="parameters" />
</message>
<message name="listDirInput">
  <part element="tns:listDirRequest" name="parameters" />
</message>
<message name="listDirOutput">
  <part element="tns:listDirResponse" name="parameters" />
</message>
<message name="copyFileInput">
  <part element="tns:copyFileRequest" name="parameters" />
</message>
<message name="copyFileOutput">
  <part element="tns:copyFileResponse" name="parameters" />
</message>
<message name="copyAsyncFileInput">
  <part element="tns:copyAsyncFileRequest" name="parameters" />
</message>
<message name="copyAsyncFileOutput">
  <part element="tns:copyAsyncFileResponse" name="parameters" />
</message>
<message name="moveFileInput">
  <part element="tns:moveFileRequest" name="parameters" />
</message>
<message name="moveFileOutput">
  <part element="tns:moveFileResponse" name="parameters" />
</message>
<message name="moveAsyncFileInput">
  <part element="tns:moveAsyncFileRequest" name="parameters" />
</message>

```

```

<message name="moveAsyncFileOutput">
  <part element="tns:moveAsyncFileResponse" name="parameters" />
</message>
<message name="stopFileTransferInput">
  <part element="tns:stopFileTransferRequest" name="parameters" />
</message>
<message name="stopFileTransferOutput">
  <part element="tns:stopFileTransferResponse" name="parameters" />
</message>
<message name="listFileTransfersInput">
  <part element="tns:listFileTransfersRequest" name="parameters" />
</message>
<message name="listFileTransfersOutput">
  <part element="tns:listFileTransfersResponse" name="parameters" />
</message>
<message name="getFileInfoInput">
  <part element="tns:getFileInfoRequest" name="parameters" />
</message>
<message name="getFileInfoOutput">
  <part element="tns:getFileInfoResponse" name="parameters" />
</message>
<message name="INVALID_PARAMMessage">
  <part element="tns:INVALID_PARAMFault" name="fault" />
</message>
<message name="INVALID_PATHMessage">
  <part element="tns:INVALID_PATHFault" name="fault" />
</message>
<message name="NO_ADMINMessage">
  <part element="tns:NO_ADMINFault" name="fault" />
</message>
<message name="RUNTIME_ERRORMessage">
  <part element="tns:RUNTIME_ERRORFault" 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="UNDEFINEDMessage">
  <part element="tns:UNDEFINEDFault" name="fault" />
</message>
<message name="UNKNOWN_FILE_TRANSFERMessage">
  <part element="tns:UNKNOWN_FILE_TRANSFERFault" name="fault" />
</message>
<message name="UNKNOWN_LOCAL_ACCOUNTMessage">
  <part element="tns:UNKNOWN_LOCAL_ACCOUNTFault" name="fault" />
</message>
<message name="UNKNOWN_MACHINEMessage">
  <part element="tns:UNKNOWN_MACHINEFault" name="fault" />
</message>

```

```

<message name="CONFIGNOTFOUNDMessage">
<part element="tns:CONFIGNOTFOUNDFault" name="fault" />
</message>
<message name="DBERRMessage">
<part element="tns:DBERRFault" name="fault" />
</message>
<message name="DIETMessage">
<part element="tns:DIETFault" name="fault" />
</message>
<message name="UNKNOWN_USERIDMessage">
<part element="tns:UNKNOWN_USERIDFault" name="fault" />
</message>
<portType name="VishnuFMSPortType">
<operation name="createFile">
<documentation>creates files on remote machines.</documentation>
<input message="tns:createFileInput" name="createFileInput" />
<output message="tns:createFileOutput" name="createFileOutput" />
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>

```

```

<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
</operation>
<operation name="createDir">
<documentation>creates directories on remote machines.</documentation>
<input message="tns:createDirInput" name="createDirInput" />
<output message="tns:createDirOutput" name="createDirOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>

```



```

<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="removeFile">
<documentation>removes files from remote hosts.</documentation>
<input message="tns:removeFileInput" name="removeFileInput" />
<output message="tns:removeFileOutput" name="removeFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="removeDir">
<documentation>removes directories (and subdirectories) from remote machines.</documentation>

```

```

<input message="tns:removeDirInput" name="removeDirInput" />
<output message="tns:removeDirOutput" name="removeDirOutput" />
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
</operation>
<operation name="chGrp">
<documentation>changes group owner of remote files/directories. </documentation>
<input message="tns:chGrpInput" name="chGrpInput" />
<output message="tns:chGrpOutput" name="chGrpOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">

```

```

<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="chMod">
<documentation>changes access rights of remote files/directories.</documentation>
<input message="tns:chModInput" name="chModInput" />
<output message="tns:chModOutput" name="chModOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">

```

```

<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</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 session key has expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="headOfFile">
<documentation>displays a few first lines of files located on remote machines.</documentation>
<input message="tns:headOfFileInput" name="headOfFileInput" />
<output message="tns:headOfFileOutput" name="headOfFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">

```

```

<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="tailOfFile">
<documentation>displays a few last lines of files located on remote machines</documentation>
<input message="tns:tailOfFileInput" name="tailOfFileInput" />
<output message="tns:tailOfFileOutput" name="tailOfFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">

```

```

<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="contentOfFile">
<documentation>displays content of files located on remote machines</documentation>
<input message="tns:contentOfFileInput" name="contentOfFileInput" />
<output message="tns:contentOfFileOutput" name="contentOfFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">

```

```

<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="listDir">
<documentation>displays the content of a remote directory. </documentation>
<input message="tns:listDirInput" name="listDirInput" />
<output message="tns:listDirOutput" name="listDirOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>

```

```

<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="copyFile">
<documentation>executes a synchronous copy of file.</documentation>
<input message="tns:copyFileInput" name="copyFileInput" />
<output message="tns:copyFileOutput" name="copyFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>

```



```

</fault>
</operation>
<operation name="copyAsyncFile">
<documentation>executes an asynchronous copy of file.</documentation>
<input message="tns:copyAsyncFileInput" name="copyAsyncFileInput" />
<output message="tns:copyAsyncFileOutput" name="copyAsyncFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="moveFile">
<documentation>executes a synchronous move of file.</documentation>
<input message="tns:moveFileInput" name="moveFileInput" />
<output message="tns:moveFileOutput" name="moveFileOutput" />

```

```

<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="moveAsyncFile">
<documentation>executes an asynchronous move of file.</documentation>
<input message="tns:moveAsyncFileInput" name="moveAsyncFileInput" />
<output message="tns:moveAsyncFileOutput" name="moveAsyncFileOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>

```

```

<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
<operation name="stopFileTransfer">
<documentation>stops an execution of a set of file transfers.</documentation>
<input message="tns:stopFileTransferInput" name="stopFileTransferInput" />
<output message="tns:stopFileTransferOutput" name="stopFileTransferOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>

```

```

<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
</operation>
<operation name="listFileTransfers">
<documentation>displays the history of all file transfers submitted by User.</documentation>
<input message="tns:listFileTransfersInput" name="listFileTransfersInput" />
<output message="tns:listFileTransfersOutput" name="listFileTransfersOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>

```

```

<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>
<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_USERIDMessage" name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
</fault>
</operation>
<operation name="getFileInfo">
<documentation>displays the information of files.</documentation>
<input message="tns:getFileInfoInput" name="getFileInfoInput" />
<output message="tns:getFileInfoOutput" name="getFileInfoOutput" />
<fault message="tns:CONFIGNOTFOUNDMessage" name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
</fault>
<fault message="tns:INVALID_PATHMessage" name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
</fault>
<fault message="tns:NO_ADMINMessage" name="NO_ADMIN">
<documentation>The user is not an administrator</documentation>
</fault>

```

```

<fault message="tns:RUNTIME_ERRORMessage" name="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:UNKNOWN_FILE_TRANSFERMessage" name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_LOCAL_ACCOUNTMessage"
name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
</operation>
</portType>
<binding name="VishnuFMSSOAPBinding" type="tns:VishnuFMSPortType">
<soapbind:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="createFile">
<documentation>creates files on remote machines.</documentation>
<soapbind:operation soapAction="" />
<input name="createFileInput">
<soapbind:body use="literal" />
</input>
<output name="createFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>

```

```

<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
</operation>
<operation name="createDir">
<documentation>creates directories on remote machines.</documentation>
<soapbind:operation soapAction="" />
<input name="createDirInput">
<soapbind:body use="literal" />
</input>
<output name="createDirOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">

```

```

<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>

```



```

</operation>
<operation name="removeFile">
<documentation>removes files from remote hosts.</documentation>
<soapbind:operation soapAction="" />
<input name="removeFileInput">
<soapbind:body use="literal" />
</input>
<output name="removeFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">

```

```

<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="removeDir">
<documentation>removes directories (and subdirectories) from remote machines.</documentation>
<soapbind:operation soapAction="" />
<input name="removeDirInput">
<soapbind:body use="literal" />
</input>
<output name="removeDirOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_LOCAL_ACCOUNT">
<documentation>The local account is unknown</documentation>
<soapbind:fault name="UNKNOWN_LOCAL_ACCOUNT" use="literal" />
</fault>
</operation>
<operation name="chGrp">
<documentation>changes group owner of remote files/directories. </documentation>
<soapbind:operation soapAction="" />
<input name="chGrpInput">
<soapbind:body use="literal" />
</input>
<output name="chGrpOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="chMod">
<documentation>changes access rights of remote files/directories.</documentation>
<soapbind:operation soapAction="" />
<input name="chModInput">
<soapbind:body use="literal" />
</input>
<output name="chModOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>

```

```

<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" 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 session key has expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="headOfFile">
<documentation>displays a few first lines of files located on remote machines.</documentation>
<soapbind:operation soapAction="" />
<input name="headOfFileInput">
<soapbind:body use="literal" />
</input>

```

```

<output name="headOfFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="tailOfFile">
<documentation>displays a few last lines of files located on remote machines</documentation>
<soapbind:operation soapAction="" />
<input name="tailOfFileInput">
<soapbind:body use="literal" />
</input>
<output name="tailOfFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">

```

```

<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="contentOfFile">
<documentation>displays content of files located on remote machines</documentation>
<soapbind:operation soapAction="" />
<input name="contentOfFileInput">
<soapbind:body use="literal" />
</input>
<output name="contentOfFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>

```



```

<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="listDir">
<documentation>displays the content of a remote directory. </documentation>
<soapbind:operation soapAction="" />
<input name="listDirInput">
<soapbind:body use="literal" />
</input>
<output name="listDirOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />

```

```

</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="copyFile">
<documentation>executes a synchronous copy of file.</documentation>
<soapbind:operation soapAction="" />
<input name="copyFileInput">
<soapbind:body use="literal" />
</input>
<output name="copyFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>

```

```

<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="copyAsyncFile">
<documentation>executes an asynchronous copy of file.</documentation>

```

```

<soapbind:operation soapAction="" />
<input name="copyAsyncFileInput">
<soapbind:body use="literal" />
</input>
<output name="copyAsyncFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="moveFile">
<documentation>executes a synchronous move of file.</documentation>
<soapbind:operation soapAction="" />
<input name="moveFileInput">
<soapbind:body use="literal" />
</input>
<output name="moveFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="moveAsyncFile">
<documentation>executes an asynchronous move of file.</documentation>
<soapbind:operation soapAction="" />
<input name="moveAsyncFileInput">
<soapbind:body use="literal" />
</input>
<output name="moveAsyncFileOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
<operation name="stopFileTransfer">
<documentation>stops an execution of a set of file transfers.</documentation>
<soapbind:operation soapAction="" />
<input name="stopFileTransferInput">
<soapbind:body use="literal" />
</input>
<output name="stopFileTransferOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />

```

```

</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
</operation>
<operation name="listFileTransfers">
<documentation>displays the history of all file transfers submitted by User.</documentation>
<soapbind:operation soapAction="" />
<input name="listFileTransfersInput">
<soapbind:body use="literal" />

```



```

</input>
<output name="listFileTransfersOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNKNOWN_USERID">
<documentation>The userId is unknown</documentation>
<soapbind:fault name="UNKNOWN_USERID" use="literal" />
</fault>
</operation>
<operation name="getFileInfo">
<documentation>displays the information of files.</documentation>
<soapbind:operation soapAction="" />
<input name="getFileInfoInput">
<soapbind:body use="literal" />
</input>
<output name="getFileInfoOutput">
<soapbind:body use="literal" />
</output>
<fault name="CONFIGNOTFOUND">
<documentation>Undefined configuration parameter</documentation>
<soapbind:fault name="CONFIGNOTFOUND" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>an option or a parameter provided is invalid for this service</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="INVALID_PATH">
<documentation>The path provided is invalid.</documentation>
<soapbind:fault name="INVALID_PATH" 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="RUNTIME_ERROR">
<documentation>Runtime error</documentation>
<soapbind:fault name="RUNTIME_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The session key has 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="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="UNKNOWN_FILE_TRANSFER">
<documentation>The transfer id is unknown</documentation>
<soapbind:fault name="UNKNOWN_FILE_TRANSFER" 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="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
</operation>
</binding>
<service name="VishnuFMSService">
<port binding="tns:VishnuFMSSOAPBinding" name="VishnuFMSPort">
<soapbind:address location="http://127.0.0.1:8080/ResourceProxy/VishnuFMS" />
</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>)