

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

Table of Contents

| | |
|--|----------|
| <u>Web Service: VishnuIMSService</u> | 1 |
| <u>Web Service: VishnuIMSService</u> | 1 |
| <u>Port VishnuIMSPort Port typeSource code</u> | 1 |
| <u>Operations</u> | 1 |
| <u>Port type VishnuIMSPortTypeSource code</u> | 1 |
| <u>WSDL source code</u> | 5 |
| <u>About wsdl-viewer.xsl</u> | 17 |

Web Service: VishnuIMSService

Web Service: VishnuIMSService

Target Namespace:

urn:ResourceProxy

Port VishnuIMSPort [Port typeSource code](#)

Location:

http://127.0.0.1:8080/ResourceProxy/VishnuIMS

Protocol:

SOAP

Default style:

document

Transport protocol:

SOAP over HTTP

Operations:

1. *exportCommands*[DetailSource code](#)
2. *getMetricCurrentValue*[DetailSource code](#)
3. *getMetricHistory*[DetailSource code](#)
4. *getSystemInfo*[DetailSource code](#)
5. *getUpdateFrequency*[DetailSource code](#)

Operations

Port type VishnuIMSPortTypeSource code

1. **exportCommands**

[Source code](#)

Description:

exports all the commands made by a user during a session

Operation type:

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

Input:

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

parameters type *exportCommandsRequest*

◆ sessionKey type *string*

The session key

◆ oldSessionId type *string*

The id of the session to export (session has ended)

◆ exportType - optional; type *ExportType* - type *string* with restriction - enum { 'UNDEFINED', 'SHELL' }

The type to export

Output:

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

parameters type *exportCommandsResponse*

Fault:

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

fault type *DBERRFault*

Fault:

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

fault type *INVALID_PARAMETERFault*

Fault:

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

fault type *UNDEFINEDFault*

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*

2. **getMetricCurrentValue**

[Source code](#)

Description:

displays the current values of system metrics

Operation type:

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

Input:

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

parameters type *getMetricCurrentValueRequest*

- ◆ sessionKey type *string*
The session key
- ◆ machineId type *string*
The id of the machine
- ◆ metricType type *MetricType* - type *string* with restriction - enum { 'UNDEFINED', 'CPUUSE', 'FREEDISKSPACE', 'FREEMEMORY' }
The type of the metric

Output:

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

parameters type *getMetricCurrentValueResponse*

- ◆ data - optional;
- ◆ metric - optional, unbounded, nillable;
 - type - optional; type *MetricType* - type *string* with restriction - enum { 'UNDEFINED', 'CPUUSE', 'FREEDISKSPACE', 'FREEMEMORY' }
The type of the metric
 - value - optional; type *long*
The value of the metric
 - time - optional; type *long*
The timestamp the metric had the value

Fault:

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

fault type *DBERRFault*

Fault:

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

fault type *INVALID_PARAMETERFault*

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*

3. **getMetricHistory**

[Source code](#)

Description:

displays the history of values of a system metric

Operation type:

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

Input:

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

parameters type *getMetricHistoryRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **machineId** type *string*
The id of the machine
- ◆ **startTime** - optional; type *long*
The start time to get the history
- ◆ **endTime** - optional; type *long*
The end time to get the history
- ◆ **type** - optional; type *MetricType* - type *string* with restriction - enum { 'UNDEFINED', 'CPUUSE', 'FREEDISKSPACE', 'FREEMEMORY' }
The type of metric searched

Output:

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

parameters type *getMetricHistoryResponse*

- ◆ **data** - optional;
- ◆ **metric** - optional, unbounded, nillable;
 - **type** - optional; type *MetricType* - type *string* with restriction - enum { 'UNDEFINED', 'CPUUSE', 'FREEDISKSPACE', 'FREEMEMORY' }
The type of the metric
 - **value** - optional; type *long*
The value of the metric
 - **time** - optional; type *long*
The timestamp the metric had the value

Fault:

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

fault type *DBERRFault*

Fault:

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

fault type *INVALID_PARAMETERFault*

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*

4. **getSystemInfo**

[Source code](#)

Description:

To get the system info on a machine

Operation type:

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

Input:

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

parameters type *getSystemInfoRequest*

- ◆ **sessionKey** type *string*
The session key
- ◆ **machineId** - optional; type *string*
The machine id

Output:

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

parameters type *getSystemInfoResponse*

- ◆ **data** - optional;
- ◆ **systeminfo** - optional, unbounded, nillable;
 - **memory** - optional; type *long*
Amount of RAM memory available on the machine (in Bytes)
 - **diskSpace** - optional; type *long*
Amount of disk space available on the machine (in Bytes)
 - **machineId** - optional; type *string*
The id of the machine

Fault:

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

fault type *DBERRFault*

Fault:

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

fault type *INVALID_PARAMETERFault*

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*

5. **getUpdateFrequency**

Source code

Description:

gets the update frequency of the IMS database

Operation type:

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

Input:

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

parameters type *getUpdateFrequencyRequest*

◆ sessionKey type *string*

The session key

Output:

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

parameters type *getUpdateFrequencyResponse*

◆ freq type *integer*

Frequency the data are updated, in second

Fault:

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

fault type *DBERRFault*

Fault:

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

fault type *INVALID_PARAMETERFault*

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*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="VishnuIMS" 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="exportCommandsRequest">
<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="oldSessionId" type="xs:string">
<xs:annotation>
<xs:documentation>The id of the session to export (session has ended)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="exportType" type="Q1:ExportType">
<xs:annotation>
<xs:documentation>The type to export</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="ExportType">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="SHELL" />
</xs:restriction>
</xs:simpleType>
<xs:element name="exportCommandsResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="getMetricCurrentValueRequest">
<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="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>The id of the machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="metricType" type="Q1:MetricType">
<xs:annotation>
<xs:documentation>The type of the metric</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="MetricType">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />

```



```

<xs:enumeration value="CPUUSE" />
<xs:enumeration value="FREEDISKSPACE" />
<xs:enumeration value="FREEMEMORY" />
</xs:restriction>
</xs:simpleType>
<xs:element name="getMetricCurrentValueResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="metric" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="type" type="Q1:MetricType" use="optional">
<xs:annotation>
<xs:documentation>The type of the metric</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="value" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>The value of the metric</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="time" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>The timestamp the metric had the value</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="getMetricHistoryRequest">
<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="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>The id of the machine</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="startTime" type="xs:long">

```

```

<xs:annotation>
<xs:documentation>The start time to get the history</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="endTime" type="xs:long">
<xs:annotation>
<xs:documentation>The end time to get the history</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="type" type="Q1:MetricType">
<xs:annotation>
<xs:documentation>The type of metric searched</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getMetricHistoryResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="metric" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="type" type="Q1:MetricType" use="optional">
<xs:annotation>
<xs:documentation>The type of the metric</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="value" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>The value of the metric</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="time" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>The timestamp the metric had the value</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="getUpdateFrequencyRequest">
<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:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getUpdateFrequencyResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="freq" type="xs:integer">
        <xs:annotation>
          <xs:documentation>Frequency the data are updated, in second</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getSystemInfoRequest">
  <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="machineId" type="xs:string">
        <xs:annotation>
          <xs:documentation>The machine id</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getSystemInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="data">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="systeminfo" nillable="true">
              <xs:complexType>
                <xs:sequence />
              </xs:complexType>
            <xs:attribute name="memory" type="xs:long" use="optional">
              <xs:annotation>
                <xs:documentation>Amount of RAM memory available on the machine (in Bytes)</xs:documentation>
              </xs:annotation>
            </xs:attribute>
            <xs:attribute name="diskSpace" type="xs:long" use="optional">

```

```

<xs:annotation>
<xs:documentation>Amount of disk space available on the machine (in Bytes)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="machineId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>The id of the machine</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="INVALID_PARAMETERFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DBERRFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="COMPONENT_ERRORFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SESSIONKEY_EXPIREDFault">
<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="UNDEFINEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
</xs:schema>
</types>
<message name="exportCommandsInput">
<part element="tns:exportCommandsRequest" name="parameters" />

```

```

</message>
<message name="exportCommandsOutput">
  <part element="tns:exportCommandsResponse" name="parameters" />
</message>
<message name="getMetricCurrentValueInput">
  <part element="tns:getMetricCurrentValueRequest" name="parameters" />
</message>
<message name="getMetricCurrentValueOutput">
  <part element="tns:getMetricCurrentValueResponse" name="parameters" />
</message>
<message name="getMetricHistoryInput">
  <part element="tns:getMetricHistoryRequest" name="parameters" />
</message>
<message name="getMetricHistoryOutput">
  <part element="tns:getMetricHistoryResponse" name="parameters" />
</message>
<message name="getUpdateFrequencyInput">
  <part element="tns:getUpdateFrequencyRequest" name="parameters" />
</message>
<message name="getUpdateFrequencyOutput">
  <part element="tns:getUpdateFrequencyResponse" name="parameters" />
</message>
<message name="getSystemInfoInput">
  <part element="tns:getSystemInfoRequest" name="parameters" />
</message>
<message name="getSystemInfoOutput">
  <part element="tns:getSystemInfoResponse" name="parameters" />
</message>
<message name="DBERRMessage">
  <part element="tns:DBERRFault" name="fault" />
</message>
<message name="INVALID_PARAMETERMessage">
  <part element="tns:INVALID_PARAMETERFault" name="fault" />
</message>
<message name="UNDEFINEDMessage">
  <part element="tns:UNDEFINEDFault" 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>
<portType name="VishnuIMSPortType">
  <operation name="exportCommands">
    <documentation>exports all the commands made by a user during a session</documentation>
    <input message="tns:exportCommandsInput" name="exportCommandsInput" />
    <output message="tns:exportCommandsOutput" name="exportCommandsOutput" />
    <fault message="tns:DBERRMessage" name="DBERR">
      <documentation>The database generated an error</documentation>
    </fault>
  </operation>

```

```

<fault message="tns:INVALID_PARAMETERMessage" name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Undefined error code</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>
</operation>
<operation name="getMetricCurrentValue">
<documentation>displays the current values of system metrics</documentation>
<input message="tns:getMetricCurrentValueInput" name="getMetricCurrentValueInput" />
<output message="tns:getMetricCurrentValueOutput" name="getMetricCurrentValueOutput" />
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>The database generated an error</documentation>
</fault>
<fault message="tns:INVALID_PARAMETERMessage" name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</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>Undefined error code</documentation>
</fault>
</operation>
<operation name="getMetricHistory">
<documentation>displays the history of values of a system metric</documentation>
<input message="tns:getMetricHistoryInput" name="getMetricHistoryInput" />
<output message="tns:getMetricHistoryOutput" name="getMetricHistoryOutput" />
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>The database generated an error</documentation>
</fault>
<fault message="tns:INVALID_PARAMETERMessage" name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</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>Undefined error code</documentation>

```

```

</fault>
</operation>
<operation name="getUpdateFrequency">
  <documentation>gets the update frequency of the IMS database</documentation>
  <input message="tns:getUpdateFrequencyInput" name="getUpdateFrequencyInput" />
  <output message="tns:getUpdateFrequencyOutput" name="getUpdateFrequencyOutput" />
  <fault message="tns:DBERRMessage" name="DBERR">
    <documentation>The database generated an error</documentation>
  </fault>
  <fault message="tns:INVALID_PARAMETERMessage" name="INVALID_PARAMETER">
    <documentation>If a parameter is invalid</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>Undefined error code</documentation>
  </fault>
</operation>
<operation name="getSystemInfo">
  <documentation>To get the system info on a machine</documentation>
  <input message="tns:getSystemInfoInput" name="getSystemInfoInput" />
  <output message="tns:getSystemInfoOutput" name="getSystemInfoOutput" />
  <fault message="tns:DBERRMessage" name="DBERR">
    <documentation>The database generated an error</documentation>
  </fault>
  <fault message="tns:INVALID_PARAMETERMessage" name="INVALID_PARAMETER">
    <documentation>If a parameter is invalid</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>Undefined error code</documentation>
  </fault>
</operation>
</portType>
<binding name="VishnuIMSSOAPBinding" type="tns:VishnuIMSPortType">
  <soapbind:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="exportCommands">
    <documentation>exports all the commands made by a user during a session</documentation>
    <soapbind:operation soapAction="" />
    <input name="exportCommandsInput">
      <soapbind:body use="literal" />
    </input>
  </operation>

```

```

<output name="exportCommandsOutput">
<soapbind:body use="literal" />
</output>
<fault name="DBERR">
<documentation>The database generated an error</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
<soapbind:fault name="INVALID_PARAMETER" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Undefined error code</documentation>
<soapbind:fault name="UNDEFINED" 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>
</operation>
<operation name="getMetricCurrentValue">
<documentation>displays the current values of system metrics</documentation>
<soapbind:operation soapAction="" />
<input name="getMetricCurrentValueInput">
<soapbind:body use="literal" />
</input>
<output name="getMetricCurrentValueOutput">
<soapbind:body use="literal" />
</output>
<fault name="DBERR">
<documentation>The database generated an error</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
<soapbind:fault name="INVALID_PARAMETER" 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>Undefined error code</documentation>

```



```

<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="getMetricHistory">
<documentation>displays the history of values of a system metric</documentation>
<soapbind:operation soapAction="" />
<input name="getMetricHistoryInput">
<soapbind:body use="literal" />
</input>
<output name="getMetricHistoryOutput">
<soapbind:body use="literal" />
</output>
<fault name="DBERR">
<documentation>The database generated an error</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
<soapbind:fault name="INVALID_PARAMETER" 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>Undefined error code</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="getUpdateFrequency">
<documentation>gets the update frequency of the IMS database</documentation>
<soapbind:operation soapAction="" />
<input name="getUpdateFrequencyInput">
<soapbind:body use="literal" />
</input>
<output name="getUpdateFrequencyOutput">
<soapbind:body use="literal" />
</output>
<fault name="DBERR">
<documentation>The database generated an error</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
<soapbind:fault name="INVALID_PARAMETER" 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>Undefined error code</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
<operation name="getSystemInfo">
<documentation>To get the system info on a machine</documentation>
<soapbind:operation soapAction="" />
<input name="getSystemInfoInput">
<soapbind:body use="literal" />
</input>
<output name="getSystemInfoOutput">
<soapbind:body use="literal" />
</output>
<fault name="DBERR">
<documentation>The database generated an error</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="INVALID_PARAMETER">
<documentation>If a parameter is invalid</documentation>
<soapbind:fault name="INVALID_PARAMETER" 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>Undefined error code</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
</operation>
</binding>
<service name="VishnuIMSService">
<port binding="tns:VishnuIMSSOAPBinding" name="VishnuIMSPort">
<soapbind:address location="http://127.0.0.1:8080/ResourceProxy/VishnuIMS" />
</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>)