

DMI Solution Guide for Juniper Networks Secure Access Platforms



Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
408 745 2000 or 888 JUNIPER
www.juniper.net

Table of Contents

| | |
|--|----|
| Table of Contents | 2 |
| Introduction | 3 |
| Related Information | 3 |
| Inbound DMI | 4 |
| Host System and Logical Systems | 5 |
| Device Specific RPCs | 6 |
| create-logical-system | 7 |
| delete-logical-system | 17 |
| get-user-stats | 19 |
| get-failed-login-count | 22 |
| get-role-count | 24 |
| get-resource-profile-count | 26 |
| get-vlan-throughput | 27 |
| get-ivs-throughput | 30 |
| get-rollback-partition-information | 33 |
| validate-custom-expression | 35 |
| get-active-users | 38 |
| disable-all-users | 44 |
| enable-all-users | 45 |
| delete-active-sessions | 46 |
| refresh-roles | 50 |
| IVE Schema | 52 |
| Sample Code | 53 |
| Get DMI Agent Configuration | 53 |
| Configure DMI Agent | 54 |
| Get Client Types | 55 |
| Add Client Type | 56 |
| Get Network Configuration | 57 |
| Configure Network Settings | 58 |
| Create a Realm | 59 |
| Create a Realm in Logical System | 62 |
| Delete a Realm | 64 |
| Create a Role | 65 |
| Create a Role in Logical System | 70 |
| Delete a Role | 76 |
| Create a Resource Profile | 77 |
| Create a Resource Profile in Logical System | 78 |
| Delete a Resource Profile | 78 |
| Create a Resource Policy | 79 |
| Create a Resource Policy in Logical System | 80 |
| Delete a Resource Policy | 81 |
| Create a Web Bookmark for a Role | 81 |
| Create a Web Bookmark for a Role in Logical System | 82 |
| Delete a Web Bookmark for a Role | 83 |
| Get Syslog Events | 84 |
| Error conditions | 86 |
| References | 87 |

Introduction

The Device Management Interface (DMI) is an XML-RPC-based protocol used to manage Juniper devices. The protocol allows administrators and third-party applications to configure and manage Juniper devices bypassing their native interfaces. The Juniper Secure Access product, with IVE version 6.4, is compliant with DMI v1.3 specification. The readers of this document are urged to read the DMI specification before using this guide.

IMPORTANT: This feature is geared toward service providers. Juniper Networks Technical Support does not offer developer support for this feature. If you require assistance, contact your Juniper Networks account team.

DMI clients can be stand-alone applications, or can be embedded in larger applications, such as network management solutions and service provider OSS's. DMI clients can connect to the IVE in one of two ways: inbound and outbound. Inbound connection is initiated into the device by the client, while outbound connection is initiated by the device into an always-available application hosting a DMI client. Juniper's NSM product uses the outbound connection.

The DMI inbound and outbound connection features in the IVE enable the IVE administrator to connect to and manage the system without having to use the browser as the administrator's interface to the IVE. IVE version 6.3 supported the outbound connection type. 6.4 introduces support for the inbound connection type.

With the new inbound DMI feature, the administrator can now connect to the IVE using an SSH secure shell Command Line Interface (CLI) to manage the device. The IVE can also be managed by integrating any SSH-aware, netconf¹ supporting application by programming the application to comply with DMI version 1.3. More information about DMI is available in the DMI specification document².

This document serves as a reference guide for achieving the following tasks in IVE:

- Configuring the inbound DMI agent
- Issuing RPC requests to retrieve the configuration of the device
- Issuing RPC requests to configure the device
- Issuing RPC requests to receive real time logs and alerts from the device
- Issuing IVE specific RPCs to get state parameter data from the device
- Issuing RPC requests for software image management
- Issuing RPC requests to backup/restore device configuration

Related Information

In addition to this guide, the following should be referred to, while administering the IVE using inbound DMI connection.

- ◆ DMI specification document²
 - The specification document for the Juniper-wide Device Management Interface
- ◆ IVE Schema

- The XML configuration schema of the IVE. More information about the schema is available in the later part of this document
- ◆ Juniper Update repository
 - The repository contains the common schema of all DMI compliant devices and the main configuration schema of IVE. For each release of the product, the schema is updated in the repository. More information about the repository can be found in section 5.7.2 of DMI specification document.
- ◆ RFC 4741: NETCONF Configuration Protocol¹
 - The protocol specification RFC document of NETCONF, the protocol that is used extensively by DMI.

Inbound DMI

The inbound DMI connection is available to the administrator of the root IVS in the IVE. The base license for the IVE will enable DMI Agent configuration option available.

Once the base license is installed, the DMI agent in the IVE can be configured in the DMI Agent page under the Configuration menu. The page can be used to configure both inbound and outbound DMI agent.

To enable the inbound DMI agent, the following needs to be configured:

- ◆ The network interface on which the inbound agent should be enabled
- ◆ The TCP port on which the inbound agent should accept connections
- ◆ The administrator realm to be used for authenticating the inbound DMI users

While the internal interface is available for all SA devices, the management interface is available for inbound in the SA6000 and SA6500 devices. The TCP port needs to be a valid value between 1 and 65535 and it is important that the port configured is not used by any other process in the IVE. It is recommended that either the default value or a value higher than 1024 be used for the TCP port. The default choice for the interface is the internal interface and the default value for the TCP port is 22.

DMI uses SSH protocol for communication¹. To connect to the IVE using inbound connection, the standard SSH shell² can be used as the command line interface. For a better user experience, a simple client can be built around the standard SSH client. Since netconf protocol is used by DMI, while connecting to the IVE using inbound, netconf channel needs to be specified as a parameter in the `ssh` command.

The following command invokes ssh to connect to the IVE's inbound DMI agent

```
ssh -l <user> <ip address> -p <port> -s netconf
```

The `-s` parameter tells the ssh server to use the netconf channel for this connection. DMI relies on the Netconf protocol for managing device configurations.

After the user is authenticated, the IVE responds with “system:” capability string to the client. The SSH client displays this to the user. At this point, the user can execute RPC commands to configure, manage and get information from the IVE. The standard schema for the RPCs and the schema for the RPC-replies are elaborated in the DMI specification document.

To close the inbound session, close-session RPC can be used. More information about close-session RPC is available in section 7.8 of NETCONF Configuration Protocol RFC¹

Host System and Logical Systems

For DMI purposes, the root IVS system is called the “host system” and virtual systems are called “logical systems”. The connection is said to be either in the host system context or in the logical system context. Some RPCs are available in both contexts, while others are available only in host system context. The following table lists the standard (ie, non-product-specific) DMI RPCs and the contexts in which they are available.

| DMI RPC | Host System | Logical System |
|--------------------------------|-------------|----------------|
| get-system-information | ✓ | ✗ |
| get-cluster-information | ✓ | ✗ |
| get-hardware-inventory | ✓ | ✗ |
| get-software-inventory | ✓ | ✗ |
| get-license-inventory | ✓ | ✗ |
| edit-config | ✓ | ✗ |
| get-config | ✓ | ✗ |
| get-configuration-information | ✓ | ✓ |
| get-alarm-information | ✓ | ✗ |
| get-syslog-events | ✓ | ✓ |
| set-logical-system | ✓ | ✓ |
| clear-logical-system | ✓ | ✓ |
| get-logical-system-information | ✓ | ✗ |
| request-package-add | ✓ | ✗ |
| request-reboot | ✓ | ✗ |
| backup | ✓ | ✗ |
| restore | ✓ | ✗ |

The DMI specification document describes the schema for the standard DMI RPCs and their replies. The *Sample Code* section contains examples of some of the RPCs listed in the table.

Device Specific RPCs

DMI also allows products to define their own non-standard RPCs, called device-specific RPC's. IVE makes use of this option and supports a set of Remote Procedure Calls that are specific only to IVE. These are mainly used in getting runtime state information from the IVE.

The table below contains the list of device specific RPCs of IVE and the context in which the calls are available.

| IVE specific RPC | Host System | Logical System |
|------------------------------------|-------------|----------------|
| create-logical-system | ✓ | ✗ |
| delete-logical-system | ✓ | ✗ |
| get-user-stats | ✓ | ✓ |
| get-failed-login-count | ✓ | ✓ |
| get-role-count | ✓ | ✓ |
| get-resource-profile-count | ✓ | ✓ |
| get-vlan-throughput | ✓ | ✓ |
| get-ivs-throughput | ✓ | ✓ |
| get-rollback-partition-information | ✓ | ✗ |
| validate-custom-expression | ✓ | ✗ |
| get-active-users | ✓ | ✗ |
| disable-all-users | ✓ | ✗ |
| enable-all-users | ✓ | ✗ |
| refresh-roles | ✓ | ✗ |
| delete-active-sessions | ✓ | ✗ |

The following subsections elaborate these IVE-specific RPCs, outline the schema for the requests and the replies and also illustrate each of the calls with examples.

create-logical-system

The create-logical-system RPC is used to create a new IVS. This RPC can be issued only in the Root IVS context.

Schema for RPC

```

<!-- create-logical-system -->
<xs:complexType name="create-logical-system">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Create Logical System</name>
        <avail>
          <matches>
            <match>
              <operational-mode>logical-
systems</operational-mode>
              <value>>false</value>
            </match>
            <match>
              <value>>true</value>
            </match>
          </matches>
        </avail>
        <description>
          This command creates a new logical system
        </description>
        <rpc-reply-tag>create-logical-system-reply</rpc-
reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="name" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Logical System Name</name>
            <description>
              The name of the logical system to create
            </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="description" type="xs:string"
minOccurs="0">
      <xs:annotation>
        <xs:appinfo>

```

```

    <dmi:param-info>
      <name>Logical System Description</name>
      <description>
        The detail description of the logical system
      </description>
    </dmi:param-info>
  </xs:appinfo>
</xs:annotation>
</xs:element>
<xs:element name="enabled" type="xs:boolean">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Enabled</name>
        <description>
          The enable/disable state of the logical
system.
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
<xs:element name="initial-configuration"
type="xs:string">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Logical System Initial
configuration</name>
        <description>
          Initialize the IVS using the default
configuration or copy the configuration from an existing
IVS. Specify the name of an existing logical system, or "-
Default Config -"
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
<xs:element name="admin-username" type="xs:string"
minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Logical System Admin Username</name>
        <description>
          The default admin username for the logical
system
        </description>
      </dmi:param-info>
    </xs:appinfo>

```

```

    </xs:annotation>
  </xs:element>
  <xs:element name="admin-password" type="xs:string"
minOccurs="0">
    <xs:annotation>
      <xs:appinfo>
        <dmi:param-info>
          <name>Logical System Admin Password</name>
          <description>
            The default admin password for the logical
system
          </description>
        </dmi:param-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:element name="minimum-guaranteed-users"
type="xs:int">
    <xs:annotation>
      <xs:appinfo>
        <dmi:param-info>
          <name>Minimum Guaranteed Users</name>
          <description>
            The number of concurrent user logins
          </description>
        </dmi:param-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:element name="burstable-maximum-users"
type="xs:int">
    <xs:annotation>
      <xs:appinfo>
        <dmi:param-info>
          <name>Burstable Maximum Users</name>
          <description>
            The maximum concurrent user logins during
peak time
          </description>
        </dmi:param-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:element name="total-maximum-bandwidth"
type="xs:int" minOccurs="0">
    <xs:annotation>
      <xs:appinfo>
        <dmi:param-info>
          <name>Total Maximum Bandwidth</name>
          <description>
            The maximum bandwidth available to this

```

```

logical system
    </description>
    </dmi:param-info>
    </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:element name="nc-maximum-bandwidth" type="xs:int"
minOccurs="0">
    <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>NC Maximum Bandwidth</name>
        <description>
          The maximum bandwidth available to Network
Connect in this logical system
        </description>
      </dmi:param-info>
    </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="vlans">
    <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>VLANs</name>
        <description>
          VLANs available to this logical system
        </description>
      </dmi:param-info>
    </xs:appinfo>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs="1" maxOccurs="unbounded">
          <xs:element name="vlan" minOccurs="1"
maxOccurs="unbounded">
            <xs:annotation>
            <xs:appinfo>
              <dmi:param-info>
                <name>VLAN</name>
                <description>
                  Selected VLAN
                </description>
              </dmi:param-info>
            </xs:appinfo>
            </xsd:annotation>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>

```

```

</xs:element>

<xs:element name="default-vlan" type="xs:string">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Default VLAN</name>
        <description>
          The default VLAN in this logical system
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>

<xs:element name="sign-in-url-prefix" type="xs:string"
minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Sign-in URL Prefix</name>
        <description>
          The sign-in URL prefix used for logical
system sign-in
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>

<xs:element name="internal-interface-virtual-ports"
minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Virtual Ports (Internal
Interface)</name>
        <description>
          The virtual port on internal interface used
for logical system sign-in
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="internal-interface-virtual-
port" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:appinfo>

```

```

        <dmi:param-info>
            <name>Virtual Port</name>
            <description>
                Selected virtual port
            </description>
        </dmi:param-info>
    </xs:appinfo>
</xsd:annotation>
</xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="external-interface-virtual-ports"
minOccurs="0">
    <xs:annotation>
        <xs:appinfo>
            <dmi:param-info>
                <name>Virtual Ports (External
Interface)</name>
                <description>
                    The virtual port on external interface used
for logical system sign-in
                </description>
            </dmi:param-info>
        </xs:appinfo>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="external-interface-virtual-
port" minOccurs="0" maxOccurs="unbounded">
                    <xs:annotation>
                        <xs:appinfo>
                            <dmi:param-info>
                                <name>Virtual Port</name>
                                <description>
                                    Selected virtual port
                                </description>
                            </dmi:param-info>
                        </xs:appinfo>
                    </xsd:annotation>
                </xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="nc-ip-pools" minOccurs="0">
    <xs:annotation>

```

```

<xs:appinfo>
  <dmi:param-info>
    <name>NC IP Ranges</name>
    <description>
      Network Connect Connection Profile IP
address pools are restricted to the IP ranges listed here
    </description>
  </dmi:param-info>
</xs:appinfo>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="nc-ip-pool" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:appinfo>
            <dmi:param-info>
              <name>NC IP Range</name>
              <description>
                Network Connect connection profile
IP address pool
              </description>
            </dmi:param-info>
          </xs:appinfo>
        </xsd:annotation>
      </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Schema for RPC-REPLY

```

<!-- logical-system-rpc-reply -->
<xs:complexType name="logical-system-rpc-reply">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Reply to the create-logical-system and delete-
logical-system RPCs
        </description>
        <rpc-list>
          <rpc-tag>create-logical-system</rpc-tag>
          <rpc-tag>delete-logical-system</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>

```

```

        </rpc-list>
    </dmi:rpc-reply-info>
</xs:appinfo>
</xs:annotation>

<xs:choice>
  <xs:element name="ok">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>OK</name>
          <desc>Success return</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
    <xs:complexType/> <!-- empty element -->
  </xs:element>

  <xs:element name="rpc-error">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>RPC Error</name>
          <desc>Error return</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="error-type">
          <xs:annotation>
            <xs:appinfo>
              <dmi:field-info>
                <name>Error Type</name>
                <desc>Error Type</desc>
              </dmi:field-info>
            </xs:appinfo>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="transport"/>
              <xs:enumeration value="rpc"/>
              <xs:enumeration value="protocol"/>
              <xs:enumeration value="application"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="error-tag">
          <xs:annotation>
            <xs:appinfo>
              <dmi:field-info>

```

```

        <name>Error Tag</name>
        <desc>The reason for error</desc>
    </dmi:field-info>
</xs:appinfo>
</xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="in-use"/>
        <xs:enumeration value="invalid-value"/>
        <xs:enumeration value="too-big"/>
        <xs:enumeration value="missing-attribute"/>
        <xs:enumeration value="bad-attribute"/>
        <xs:enumeration value="unknown-attribute"/>
        <xs:enumeration value="missing-element"/>
        <xs:enumeration value="bad-element"/>
        <xs:enumeration value="unknown-element"/>
        <xs:enumeration value="unknown-namespace"/>
        <xs:enumeration value="access-denied"/>
        <xs:enumeration value="lock-denied"/>
        <xs:enumeration value="resource-denied"/>
        <xs:enumeration value="rollback-failed"/>
        <xs:enumeration value="data-exists"/>
        <xs:enumeration value="data-missing"/>
        <xs:enumeration value="operation-not-
supported"/>
        <xs:enumeration value="operation-failed"/>
        <xs:enumeration value="partial-operation"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="error-severity">
    <xs:annotation>
        <xs:appinfo>
            <dmi:field-info>
                <name>Error Severity</name>
                <desc>Error Severity</desc>
            </dmi:field-info>
        </xs:appinfo>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="error"/>
            <xs:enumeration value="warning"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>

```

The following is an example of creating a new logical system, passing only the mandatory parameters for the RPC. The XML code creates a logical system with default config, setting Internal Port as the default vlan port for the newly created IVS.

Example for RPC

```
<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <create-logical-system>
    <name>test</name>
    <initial-configuration>- Default Config -</initial-
configuration>
    <enabled>>true</enabled>
    <minimum-guaranteed-users>3</minimum-guaranteed-users>
    <burstable-maximum-users>4</burstable-maximum-users>
    <vlans>
      <vlan>Internal Port</vlan>
    </vlans>
    <default-vlan>Internal Port</default-vlan>
  </create-logical-system>
</rpc>
```

If the RPC is successful, the following is the response received. On error conditions, the error message explains the reason the command failed.

Example for RPC-REPLY

```
<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

An example of the same RPC with all the parameters passed is given below. This assumes that the virtual ports and the NC IP pools are already configured in the IVE, without which the command would fail. The RPC creates an IVS with configuration copied from the Root IVS.

Example for RPC

```
<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <create-logical-system>
    <name>test</name>
    <initial-configuration>Root</initial-
```

```

configuration>
  <enabled>>true</enabled>
  <admin-username>admin</admin-username>
  <admin-password>dana123</admin-password>
  <minimum-guaranteed-users>3</minimum-
guaranteed-users>
  <burstable-maximum-users>4</burstable-maximum-
users>
  <vlans>
    <vlan>Internal Port</vlan>
  </vlans>
  <default-vlan>Internal Port</default-vlan>
  <internal-virtual-ports>
    <internal-virtual-port>int_vp1</internal-
virtual-port>
    <internal-virtual-port>int_vp2</internal-
virtual-port>
  </internal-virtual-ports>
  <nc-ip-pools>
    <nc-ip-pool>10.10.10.10-20</nc-ip-pool>
    <nc-ip-pool>10.10.10.50</nc-ip-pool>
  </nc-ip-pools>
</create-logical-system>
</rpc>

```

delete-logical-system

The delete-logical-system RPC, as the name implies, deletes an IVS in the IVE. This command requires the name of the IVS to be specified as the parameter in the call.

Schema for the RPC

```

<!-- delete-logical-system -->
<xs:complexType name="delete-logical-system">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Delete Logical System</name>
        <avail>
          <matches>
            <match>
              <operational-mode>logical-
systems</operational-mode>
              <value>>false</value>
            </match>
            <match>
              <value>>true</value>

```

```

        </match>
    </matches>
</avail>
<description>
    This command deletes an existing logical system
</description>
    <rpc-reply-tag>delete-logical-system-reply</rpc-
reply-tag>
    </dmi:rpc-info>
</xs:appinfo>
</xs:annotation>
<xs:sequence>
    <xs:element name="name" type="xs:string">
        <xs:annotation>
            <xs:appinfo>
                <dmi:param-info>
                    <name>Logical System Name</name>
                    <description>
                        The name of the logical system to delete
                    </description>
                </dmi:param-info>
            </xs:appinfo>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>

```

Delete logical system RPC takes the name of the IVS as the parameter and if the IVS with the given name is present, deletes it from the IVE.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
    <delete-logical-system>
        <name>test</name>
    </delete-logical-system>
</rpc>

```

If the RPC is successful the following reply is received.

Example for RPC-REPLY

```

<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>

```

get-user-stats

The `get-user-stats` RPC retrieves the number of users existing presently and in the last 24 hour interval in the IVE. Optionally, the RPC takes a parameter if the data has to be reset after the retrieval. This call can be executed in both the host-system context and in the logical-system context and the data is pertinent to the appropriate IVS.

Schema for RPC

```

<!-- get-user-stats -->
<xs:complexType name="get-user-stats">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get user statistics</name>
        <description>
          This command returns AllocatedUserCount
          CurrentUserCount MaxUsersin24Hrs MinUsersin24Hrs
        </description>
        <rpc-reply-tag>user-stats</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="reset" type="xs:boolean"
      minOccurs="0">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Reset Stats</name>
            <description>
              This will govern the resetting of this
              statistics data. By default, the data is not reset.
            </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

As shown in the schema below, the following are the data sent back by the IVE:

- ◆ Total number of allocated users
- ◆ Total number of current users

- ◆ Maximum number of active users in the last 24 hour period
- ◆ Minimum number of active users in the last 24 hour period

Schema for RPC-REPLY

```

<!-- user-stats -->
<xs:complexType name="user-stats">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          User Statistics
        </description>
        <rpc-list>
          <rpc-tag>get-user-stats</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="allocated-user-count"
type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Allocated User Count</name>
            <desc>The Allocated User Count for the logical
system</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="current-user-count"
type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Current user count</name>
            <desc>The number of users logged in
currently</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="max-active-user-count-24hrs"
type="xs:string">
      <xs:annotation>
        <xs:appinfo>

```

```

        <dmi:field-info>
          <name>Max active user count in the last 24
Hrs</name>
          <desc>The Max active user count for a 24 Hrs
moving window</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="min-active-user-count-24hrs"
type="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>Min active user count in the last 24
Hrs</name>
          <desc>The Min active user count for a 24 Hrs
moving window</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>

```

An example of the get-user-stats RPC call and the reply are rendered below.

Example for RPC

```

<rpc message-id="14">
  <get-user-stats/>
</rpc>

```

Example for RPC-REPLY

```

<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <user-stats>
    <allocated-user-count>10</allocated-user-count>
    <current-user-count>3</current-user-count>
    <max-active-user-count-24hrs>2</max-active-user-
count-24hrs>
    <min-active-user-count-24hrs>0</min-active-user-
count-24hrs>
  </user-stats>
</rpc-reply>

```

get-failed-login-count

The get-failed-login-count RPC is used to retrieve the number of failures in the last 24 hour interval due to number of users exceeding the limit and due to authentication failure. Similar to get-user-stats RPC, this also takes the reset option as a parameter.

Schema for the RPC

```

<!-- get-failed-login-count -->
<xs:complexType name="get-failed-login-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get failed login count for Authentication
failure and Exceeded user</name>
        <description>
          This command returns the Number of Logins
refused due to exceeding allowed limits and Auth failure
(24 hour moving window)
        </description>
        <rpc-reply-tag>failed-login-count</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="reset" type="xs:boolean"
minOccurs="0">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Reset Stats</name>
            <description>
              This will govern the resetting of this
statistics data. By default, the data is not reset.
            </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Schema for RPC-REPLY

```

<!-- failed-login-count -->
<xs:complexType name="failed-login-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Failed Login statistics Info
        </description>
        <rpc-list>
          <rpc-tag>get-failed-login-count</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>

  <xs:sequence>

    <xs:element name="exceeded-user-count"
type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Number of login failures due to exceeded
login user limit</name>
            <desc>The Number of user logins refused due to
exceeded user count.</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="failed-auth-count" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Number of login failures due to
authentication failure</name>
            <desc>The Number of user logins refused due to
authentication failure.</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

An example of the get-failed-login-count RPC and its response are given below.

Example for RPC

```
<rpc message-id="12">
  <get-failed-login-count/>
</rpc>
```

Example for RPC-REPLY

```
<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <failed-login-count>
    <exceeded-user-count>2</exceeded-user-count>
    <failed-auth-count>4</failed-auth-count>
  </failed-login-count>
</rpc-reply>
```

get-role-count

To retrieve the number of administrative roles and the user roles available in the IVS, the `get-role-count` RPC can be used. The RPC can be executed in both the host-system and in the logical-system context and the RPC reply contains the statistics pertinent to the IVS currently set.

Schema for RPC

```
<!-- get-role-count -->
<xs:complexType name="get-role-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get The roles count</name>
        <description>
          This command returns the admin and user role
          count.
        </description>
        <rpc-reply-tag>role-count</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

Schema for RPC-REPLY

```

<!-- role-count -->
<xs:complexType name="role-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Number for roles configured
        </description>
        <rpc-list>
          <rpc-tag>get-role-count</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="admin-role-count" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Admin roles Count</name>
            <desc>The total number of admin roles
configured for the logical system</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="user-role-count" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>User roles Count</name>
            <desc>The total number of user roles
configured for the logical system</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

  </xs:sequence>
</xs:complexType>

```

An example of the RPC and its reply follow.

Example for RPC

```

<rpc message-id="12">
  <get-role-count/>

```

```
</rpc>
```

Example for RPC-REPLY

```
<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <role-count>
    <admin-role-count>2</admin-role-count>
    <user-role-count>1</user-role-count>
  </role-count>
</rpc-reply>
```

get-resource-profile-count

The number of resource profiles in the IVS can be retrieved with the `get-resource-profile-count` RPC. Here is the schema for the RPC and its reply.

Schema for the RPC

```
<!-- get-resource-profile-count -->
<xs:complexType name="get-resource-profile-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get the resource profile count</name>
        <description>
          This command returns the number of resource
          profiles in the logical system.
        </description>
        <rpc-reply-tag>resource-profile-count</rpc-reply-
tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

Schema for RPC-REPLY

```
<!-- resource-profile-count -->
<xs:complexType name="resource-profile-count">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
```

```

        Number for resource profiles configured.
    </description>
    <rpc-list>
        <rpc-tag>get-resource-profile-count</rpc-tag>
    </rpc-list>
</dmi:rpc-reply-info>
</xs:appinfo>
</xs:annotation>
<xs:sequence>
    <xs:element name="profile-count" type="xs:string">
        <xs:annotation>
            <xs:appinfo>
                <dmi:field-info>
                    <name>Resource profile Count</name>
                    <desc>The total number of resource profiles
configured for the logical system</desc>
                </dmi:field-info>
            </xs:appinfo>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>

```

Example of the RPC request and its response are rendered below.

Example for RPC

```

<rpc message-id="12">
    <get-resource-profile-count/>
</rpc>

```

Example for RPC-REPLY

```

<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <resource-profile-count>
        <profile-count>20</profile-count>
    </resource-profile-count>
</rpc-reply>

```

get-vlan-throughput

The throughput of a specific VLAN can be retrieved using the get-vlan-throughput RPC. The RPC reply contains the throughput for the VLAN in bytes.

The schema for the RPC and its reply are given below.

Schema for the RPC

```

<!-- get-vlan-throughput -->
<xs:complexType name="get-vlan-throughput">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get VLAN Throughput</name>
        <description>
          This command returns the throughput for the VLAN
          id sent as parameter
        </description>
        <rpc-reply-tag>vlan-throughput</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="vlanid" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>VLAN ID</name>
            <description>
              The ID of the VLAN whose throughput is
              required. The values should be in the range 0-4094. 0
              Indicated the internal interface.
            </decription>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="reset" type="xs:boolean"
      minOccurs="0">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Reset Stats</name>
            <description>
              This will govern the resetting of the
              statistics data. By default, the data is not reset.
            </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Schema for RPC-REPLY

```

<!-- vlan-throughput -->
<xs:complexType name="vlan-throughput">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          VLAN throughput information
        </description>
        <rpc-list>
          <rpc-tag>get-vlan-throughput</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="max-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Maximum throughput over the last 24
Hrs</name>
            <desc>Maximum throughput over the last 24
Hrs</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="min-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Minimum throughput over the last 24
Hrs</name>
            <desc>Minimum throughput over the last 24
Hrs</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="avg-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Average throughput over the last 24

```

```

Hrs</name>
      <desc>Average throughput over the last 24
Hrs</desc>
      </dmi:field-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>

</xs:sequence>
</xs:complexType>

```

An example of the RPC and its response are given below.

Example for RPC

```

<rpc message-id="12">
  <get-vlan-throughput>
    <vlanid>0</vlanid>
  </get-vlan-throughput>
</rpc>

```

Example for RPC-REPLY

```

<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <vlan-throughput>
    <max-throughput>8591642</max-throughput>
    <min-throughput>0</min-throughput>
    <avg-throughput>4918466.471698</avg-throughput>
  </vlan-throughput>
</rpc-reply>

```

get-ivs-throughput

A variation to getting the throughput in IVE is to retrieve the value for a given IVS. If there are multiple VLANs assigned for an IVS, then the throughput will be a consolidated value of all the IVSes. The RPC also takes the reset parameter, which if set would reset the current throughput values.

The schema for the RPC and its reply follow.

Schema for the RPC

```

<!-- get-throughput -->
<xs:complexType name="get-throughput">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get throughput for the logical system</name>
        <description>
          This command returns the consolidated throughput
          for all the VLANS for a logical system
        </description>
        <rpc-reply-tag>ivs-throughput</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="reset" type="xs:boolean"
      minOccurs="0">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Reset Stats</name>
            <description>
              This will govern the resetting of the
              statistics data. By default, the data is not reset.
            </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Schema for RPC-REPLY

```

<!-- ivs-throughput -->
<xs:complexType name="ivs-throughput">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          IVS throughput information
        </description>
        <rpc-list>
          <rpc-tag>get-throughput</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>

```

```

    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="max-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Maximum throughput over the last 24
Hrs</name>
            <desc>Maximum throughput over the last 24
Hrs</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="min-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Minimum throughput over the last 24
Hrs</name>
            <desc>Minimum throughput over the last 24
Hrs</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="avg-throughput" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Average throughput over the last 24
Hrs</name>
            <desc>Average throughput over the last 24
Hrs</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

  </xs:sequence>
</xs:complexType>

```

An example of the RPC and its response are rendered below.

Example for RPC

```
<rpc message-id="12">
  <get-ivs-throughput>
    <name>test</name>
  </get-ivs-throughput>
</rpc>
```

Example for RPC-REPLY

```
<rpc-reply message
id="12" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ivsthroughput>
    <maxthroughput>10972025</maxthroughput>
    <minthroughput>0</minthroughput>
    <avgthroughput>5527986.533333</avgthroughput>
  </ivsthroughput>
</rpc-reply>
```

get-rollback-partition-information

The `get-rollback-partition-information` RPC retrieves the device rollback version information such as `os-name`, `os-version` and `os build number`.

Schema for RPC

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- get-rollback-partition-information -->
<xs:complexType name="get-rollback-partition-information">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Get Rollback Partition Information</name>
        <description>
          This command returns IVE's rollback partition
information
        </description>
        <rpc-reply-tag>rollback-partition-
information</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

```
</xs:schema>
```

Schema for RPC-REPLY

```
<xs:complexType name="rollback-partition-information">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Rollback Software Image Information
        </description>
        <rpc-list>
          <rpc-tag>get-rollback-partition-
information</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="os-name" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Software Image OS Name</name>
            <desc>Software Image OS Name</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="os-version" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Software Image OS Version</name>
            <desc>Software Image OS Version</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="build" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Software Image Build Number</name>
            <desc>Software Image Build Number</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```

    </xs:appinfo>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

The following is an example to retrieve rollback software image information.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <get-rollback-partition-information/>
</rpc>

```

If the RPC is successful, the following is the response received. On error conditions, the error message explains the reason the command failed.

Example for RPC-REPLY

```

<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rollback-partition-information>
    <os-name>ive-sa</os-name>
    <os-version>6.4R1</os-version>
    <build>14063</build>
  </rollback-partition-information>
</rpc-reply>

```

validate-custom-expression

The validate-custom-expression RPC is used to validate custom expression. This RPC validates realm role mapping rule, policy detailed rule, log-filter-settings-query and sensor event expressions.

Schema for RPC

```

<!--validate-custom-expression--!>
<xs:complexType name="validate-custom-expression">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Validate Custom Expression</name>
        <description>This command will validate the

```

```

expression</description>
    <rpc-reply-tag>validate-custom-expression-
reply</rpc-reply-tag>
    </dmi:rpc-info>
    </xs:appinfo>
</xs:annotation>
<xs:sequence>
    <xs:element name="expression" type="xs:string">
        <xs:annotation>
            <xs:appinfo>
                <dmi:param-info>
                    <name>Custom Expression</name>
                    <description>Custom Expression to
validate</description>
                </dmi:param-info>
            </xs:appinfo>
        </xs:annotation>
    </xs:element>
    <xs:element name="expression-usage">
<xs:annotation>
    <xs:appinfo>
        <dmi:field-info>
            <name>Custom Expression Usage</name>
            <desc>Identify the place in Admin UI where
the custom expression is being used.</desc>
        </dmi:field-info>
    </xs:appinfo>
</xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="realm-role-mapping-rule"/>
            <xs:enumeration value="policy-detailed-rule"/>
            <xs:enumeration value="log-filter-settings-
query"/>
            <xs:enumeration value="sensor-event"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

This RPC instructs device to validate expression provided in the <expression> parameter for realm role mapping.

Example for RPC

```

<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <validate-custom-expression>
    <expression>cacheCleanerStatus = 1</expression>

```

```

    <expression-usage>realm-role-mapping-rule</expression-usage>
  </validate-custom-expression>
</rpc>

```

This RPC instructs device to validate expression provided in the <expression> parameter for policy detailed rule.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <validate-custom-expression>
    <expression>loginURL =
"partners.company.com/"</expression>
    <expression-usage>policy-detailed-rule</expression-usage>
  </validate-custom-expression>
</rpc>

```

This RPC instructs device to validate log filter setting expression provided in the <expression> parameter.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <validate-custom-expression>
    <expression>result = 200 OR port = 210 OR method =
"GET"</expression>
    <expression-usage>log-filter-settings-
query</expression-usage>
  </validate-custom-expression>
</rpc>

```

This RPC instructs device to validate sensor event expression provided in the <expression> parameter.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <validate-custom-expression><expression>idp.action =
'ignore' OR idp.action = 'none'</expression>
    <expression-usage>sensor-event</expression-usage>
  </validate-custom-expression>
</rpc>

```

If the RPC is successful, the following is the response received. On error conditions, the error message explains the reason the command failed.

Example for RPC-REPLY

```
<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

get-active-users

The get-active-users RPC retrieves current active user sessions in host system and virtual system context.

Schema for RPC

```
<!--get-active-users -->
<xs:complexType name="get-active-users">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Search Active Users</name>
        <description>This RPC requests for a fixed number
of active users currently in the system based on a
specified name pattern. </description>
        <rpc-reply-tag>active-users</rpc-reply-tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="name" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>User Name</name>
            <description>This is the user name to search for. A
regular expression may be used. </description>
          </dmi:param-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="number" type="xs:integer">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
```

```

    <name>Maximum Number of Users to Display </name>
    <description>This is the maximum number of users to
return.</description>
    </dmi:param-info>
  </xs:appinfo>
</xs:annotation>
</xs:element>
<xs:element name="sortby" type="xs:string">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>Column to Sort By</name>
        <description>This is the column name the IVE will
use to sort. IVE will use name + number + sortby +
sortorder to return the correct result to
NSM.</description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
<xs:element name="sortorder" type="xs:integer">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>The Sort Order</name>
        <description> This is the requested sort order for the
reply. The meaning of the value: 1 for ascending, 2 for
descending and 0 for unsorted.    <description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

Schema for RPC-REPLY

```

<!-- active-users -->
<xs:complexType name="active-users">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Active Users
        </description>
        <rpc-list>
          <rpc-tag>get-active-users</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>

```

```

    </dmi:rpc-reply-info>
  </xs:appinfo>
</xs:annotation>
<xs:sequence>

  <xs:element name="user-login-permission"
type="xs:boolean">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>User Login Permission</name>
          <desc> The current system setting for allowing
or disallowing users to login: true for allowing and false
for disallowing. </desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="total-matched-record-number"
type="xs:integer">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>Total Matched Record Number</name>
          <desc> Total number of active user records
that matched the search criterion </desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="total-returned-record-number"
type="xs:integer">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name> Total Returned Record Number </name>
          <desc> Number of active user records returned
in this reply </desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="active-user-records">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="active-user-record" minOccurs="0"
maxOccurs="unbounded">
          <xs:complexType>

```

```

<xs:sequence>
  <xs:element name="active-user-name" type="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>Active User Name</name>
          <desc>This is the name of an active
user</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="authentication-realm"
type="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>Authentication Realm</name>
          <desc>The authentication realm the user logged
in</desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="user-roles" type="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>User Roles </name>
          <desc>The roles assigned to the user </desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="user-sign-in-time"
type="xs:dateTime">
    <xs:annotation>
      <xs:appinfo>
        <dmi:field-info>
          <name>User Sign-in Time </name>
          <desc>The time the user signed in </desc>
        </dmi:field-info>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

  <xs:element name="events" type="xs:integer">
    <xs:annotation>

```

```

        <xs:appinfo>
          <dmi:field-info>
            <name>Events </name>
            <desc> Events for this session. Its value can
            be 0, 1 or 2. If there are no IDP (intrusion detection)
            events for the session, it.s set to 0. If there are IDP
            events for the session, but the session has not been
            quarantined, it.s set to 1. If the session has been
            quarantined, it.s set to 2 </desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="login-node" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Login Node </name>
            <desc>The node from which the user signed in
</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="network-connect-ip"
type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name>Network Connect IP </name>
            <desc>The IP address assigned for the user's
network connect session </desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

    <xs:element name="session-id" type="xs:string">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name><User Session ID/name>
            <desc>The unique user session ID</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

  </xs:sequence>

```

```

</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

Example for RPC

```

<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
    <get-active-users><name>*</name>
      <number>32</number>
    <orderby>userName</orderby>
    <sortorder>2</sortorder>
  </get-active-users>
</rpc>

```

An example of the RPC reply is shown below.

Example for RPC-REPLY

```

<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<get-active-users>
  <user-login-permission>true</user-login-permission>
  <total-matched-record-number>3</total-matched-record-
number>
  <total-returned-record-number>3</total-returned-record-
number>
  <active-user-records>
    <active-user-record>
      <active-user-name>newadmin</active-user-name>
      <authentication-realm>Admin Users</authentication-
realm>
      <user-roles>.Administrators</user-roles>
      <user-sign-in-time>2009/05/28 13:37:31</user-sign-in-
time>
      <events>0</events>
      <login-node>localhost2</login-node>
      <network-connect-ip></network-connect-ip>
      <session-
id>userid25b50923ef52dcc2bc8d113a5e5d17a02a74fde8</session-
id>
    </active-user-record>

```

```

    <active-user-record>
      <active-user-name>admindb</active-user-name>
      <authentication-realm>Admin Users</authentication-
realm>
      <user-roles>.Administrators</user-roles>
      <user-sign-in-time>2009/05/29 15:22:33</user-sign-in-
time>
      <events>0</events>
      <login-node>localhost2</login-node>
      <network-connect-ip></network-connect-ip>
      <session-
id>useruida9d896a45d2f2dd7e4e64a04ca57d070c5dc9228</session-
id>
    </active-user-record>
    <active-user-record>
      <active-user-name>admin</active-user-name>
      <authentication-realm>Admin Users</authentication-
realm>
      <user-roles>.Administrators</user-roles>
      <user-sign-in-time>2009/05/29 09:55:21</user-sign-in-
time>
      <events>0</events>
      <login-node>localhost2</login-node>
      <network-connect-ip></network-connect-ip>
      <session-
id>userid5c142fc2be8e15de7e4015257f1f14af386d6a1e</session-
id>
    </active-user-record>
  </active-user-records>
</get-active-users>
</rpc-reply>

```

disable-all-users

The `disable-all-users` RPC is used to disable all active user sessions in host system and virtual system context.

Schema for RPC

```

<!-- disable-all-users -->
<xs:complexType name="disable-all-users">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Disable All End User </name>
        <description>
          This RPC disables all the end users on the
system
        </description>

```

```

    <rpc-reply-tag>disable-all-users-reply</rpc-reply-
tag>
    </dmi:rpc-info>
    </xs:appinfo>
    </xs:annotation>
</xs:complexType>

```

Example for RPC

```

<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <disable-all-users/>
</rpc>

```

If the RPC is successful, the following is the response received. On error conditions, the error message explains the reason the command failed.

Example for RPC-REPLY

```

<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

```

enable-all-users

The enable-all-users RPC is used to enable all user sessions in host system and virtual system context.

Schema for RPC

```

<!--enable-all-users-->
<xs:complexType name="enable-all-users">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Enable All End User </name>
        <description>
          This RPC enables all the end users on the system
        </description>
        <rpc-reply-tag>enable-all-users-reply</rpc-reply-
tag>

```

```

    </dmi:rpc-info>
  </xs:appinfo>
</xs:annotation>
</xs:complexType>

```

Example for RPC

```

<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <enable-all-users/>
</rpc>

```

If the RPC is successful an “ok” reply is sent. On error conditions, the error message explains the reason the command failed.

delete-active-sessions

The delete-active-sessions RPC is used to delete selected user sessions in host system and virtual system context.

Schema for RPC

```

<!-- delete-active-sessions -->
<xs:complexType name="delete-active-sessions">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Delete Specified Active User Sessions
</name>
        <description>
          This RPC deletes some active user sessions
currently on the system
        </description>
        <rpc-reply-tag>delete-sessions-reply</rpc-reply-
tag>
      </dmi:rpc-info>
    </xs:appinfo>
  </xs:annotation>
  <xs:choice>
    <xs:element name="session-records">
      <xs:annotation>
        <xs:appinfo>
          <dmi:param-info>
            <name>Some Users</name>
            <description>
              This parameter tells the IVE to delete the

```

```

specified users on the system
    </description>
  </dmi:param-info>
</xs:appinfo>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="session-record" minOccurs="0"
maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string">
        <xs:annotation>
          <xs:appinfo>
            <dmi:param-info>
              <name>User Name</name>
              <description>
                This is the user name. No regular expression
may be used.
              </description>
            </dmi:param-info>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
<xs:element name="authentication-realm" type="xs:string">
  <xs:annotation>
    <xs:appinfo>
      <dmi:field-info>
        <name>Authentication Realm</name>
        <desc>The authentication realm the user logged
in</desc>
      </dmi:field-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
<xs:element name="session-id" type="xs:string">
  <xs:annotation>
    <xs:appinfo>
      <dmi:field-info>
        <name>User Session ID</name>
        <desc>The unique user session ID </desc>
      </dmi:field-info>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="all">
  <xs:annotation>
    <xs:appinfo>
      <dmi:param-info>
        <name>All Users</name>
        <description>
          This parameter tells the IVE to delete all
          active users on the system
        </description>
      </dmi:param-info>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType/>
</xs:element>
</xs:choice>
</xs:complexType>

```

Schema for RPC-REPLY

```

<!-- delete-sessions -->
<xs:complexType name="delete-sessions-reply" >
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-reply-info>
        <description>
          Reply from the 'delete session' request
        </description>
        <rpc-list>
          <rpc-tag>delete-active-sessions</rpc-tag>
        </rpc-list>
      </dmi:rpc-reply-info>
    </xs:appinfo>
  </xs:annotation>

  <xs:sequence>

    <xs:element name="failed-session-number"
      type="xs:integer">
      <xs:annotation>
        <xs:appinfo>
          <dmi:field-info>
            <name> Number of Failed Sessions </name>
            <desc> Number of sessions the RPC failed to
            delete</desc>
          </dmi:field-info>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>

```

```

<xs:element name="failed-session-records">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="failed-session-record" minOccurs="0"
maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>

            <xs:element name="name" type="xs:string" >
              <xs:annotation>
                <xs:appinfo>
                  <dmi:param-info>
                    <name>User Name</name>
                    <description>
                      This is the user name. No regular expression
may be used.
                    </description>
                  </dmi:param-info>
                </xs:appinfo>
              </xs:annotation>
            </xs:element>

            <xs:element name="authentication-realm"
type="xs:string">
              <xs:annotation>
                <xs:appinfo>
                  <dmi:field-info>
                    <name>Authentication Realm</name>
                    <desc>The authentication realm the user logged
in</desc>
                  </dmi:field-info>
                </xs:appinfo>
              </xs:annotation>
            </xs:element>

            <xs:element name="session-id" type="xs:string">
              <xs:annotation>
                <xs:appinfo>
                  <dmi:field-info>
                    <name><User Session ID/name>
                    <desc>The unique user session ID</desc>
                  </dmi:field-info>
                </xs:appinfo>
              </xs:annotation>
            </xs:element>

          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>
</xs:sequence>

```

Example for RPC

```

<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'\
  <delete-active-sessions>
    <session-records><session-record>
      <name>newadmin</name>
      <authentication-realm>Admin Users</authentication-realm>
      <session-
id>userid5c142fc2be8e15de7e4015257f1f14af386d6a1e</session-
id>
      </session-record></session-records>
    </delete-active-sessions>
  </rpc>

```

If the RPC is successful, the following is the response received. On error conditions, the error message explains the reason the command failed.

Example for RPC-REPLY

```

<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <delete-active-sessions>
    <failed-session-number>0</failed-session-number>
  </delete-active-sessions>
</rpc-reply>

```

refresh-roles

The refresh-roles RPC is used to refresh active user session roles in host system and virtual system context.

Schema for RPC

```

<!-- refresh-roles -->
<xs:complexType name="refresh-roles">
  <xs:annotation>
    <xs:appinfo>
      <dmi:rpc-info>
        <name>Refresh User Roles</name>

```

```
<description>
  The Refresh Roles request results in dynamic policy
  evaluation being triggered for the selected users.
</description>
  <rpc-reply-tag>refresh-roles-reply</rpc-reply-tag>
</dmi:rpc-info>
</xs:appinfo>
</xs:annotation>
</xs:complexType>
```

Example for RPC

```
<rpc message-id='101'>
  xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <refresh-roles/>
</rpc
```

If the RPC is successful an “ok” reply is sent. On error conditions, the error message explains the reason the command failed.

IVE Schema

The IVE XML schema describes the configuration tree of the IVE and is useful in constructing the requests to configure and manage the IVE. The schema can be downloaded from the Admin UI in the Export XML page, in the Maintenance menu.

As described in the DMI specification, the standard NETCONF RPCs such as get-config, edit-config etc. can be used to retrieve and update the configuration. The standard NETCONF operations such as create, delete, merge and replace are supported by the IVE.

Sample Code

In this section, we illustrate some of the operations that are typically executed by the IVE administrator, by issuing RPCs using the inbound DMI. These examples will not cover all possible cases and might need certain preconditions for the operation to be successful.

The examples given include operations that are possible only on the host system, or the Root IVS and operations that are possible in both host and logical systems.

Most of these examples use the standard NETCONF operations `get-config` and `edit-config`.

Get DMI Agent Configuration

The following snippet gets the configuration of the DMI agent settings from the IVE.

Example for RPC

```
<rpc message-id="12">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <configuration>
        <system>
          <configuration>
            <dmi-agent/>
          </configuration>
        </system>
      </configuration>
    </filter>
  </get-config>
</rpc>
```

The following is an example of the reply received for the RPC requesting information about DMI agent configuration.

Example for RPC-REPLY

```
<rpc-reply message-id="12"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <configuration xmlns="http://xml.juniper.net/ive-
sa/6.4R1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" iveData="1508" saData="1407">
      <system>
        <configuration>
          <dm-agent>
            <enabled>true</enabled>
            <primary-
server>10.209.117.19</primary-server>
            <device-id>C7B91C</device-id>
            <hmac-key>netscreen</hmac-key>
            <in-enabled>true</in-enabled>
            <in-int-port-enabled>true</in-int-
port-enabled>
            <inbound-port>22</inbound-port>
          </dm-agent>
        </configuration>
      </system>
    </configuration>
  </data>
</rpc-reply>
```

Note that the parameters are returned only if the values differ from the default values for the attributes.

Configure DMI Agent

The following snippet configures some of the inbound and outbound settings for DMI agent in IVE.

Example for RPC

```
<rpc message-id="12">
<edit-config>
  <target>
    <running/>
  </target>
</edit-config>
```

```

<configuration>
  <system>
    <configuration>
      <dmi-agent>
        <enabled>>true</enabled>
        <primary-server>10.20.30.50</primary-server>
        <primary-port>7804</primary-port>
        <backup-server>10.20.30.60</backup-server>
        <backup-port>7805</backup-port>
        <device-id>TheDeviceId</device-id>
        <hmac-key>TheHMAC</hmac-key>
        <admin-realm>Admin Users</admin-realm>
        <in-enabled>>false</in-enabled>
        <in-mgt-port-enabled>>false</in-mgt-port-
enabled>
        <in-int-port-enabled>>true</in-int-port-enabled>
        <inbound-port>2222</inbound-port>
      </dmi-agent>
    </configuration>
  </system>
</configuration>
</config>
</edit-config>
</rpc>

```

Note that changing some of the DMI agent settings could make the server close the current DMI session.

Get Client Types

To get the list of client types available in the IVE, the following get-config RPC can be issued.

Example for RPC

```

<rpc message-id="12">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <configuration>
        <system>
          <configuration>
            <client-types/>
          </configuration>
        </system>
      </configuration>
    </filter>
  </get-config>
</rpc>

```

```

        </system>
      </configuration>
    </filter>
  </get-config>
</rpc>

```

Add Client Type

The following snippet adds a client type to the client types list. By default, the client type is added to the top of the list.

Example for RPC

```

<rpc message-id="12">
  <edit-config>
    <target><running/></target>
  <config>
    <configuration
      xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
      <system>
        <configuration>
          <client-types>
            <client-type-data operation="create">
              <user-agent-string-pattern>*new client
type*</user-agent-string-pattern>
              <client-type>smart-phone-html-basic</client-
type>
            </client-type-data>
          </client-types>
        </configuration>
      </system>
    </configuration>
  </config>
</edit-config>
</rpc>

```

On success, the following reply is received.

Example for RPC-REPLY

```

<rpc-reply message-id="12"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>

```

```
</rpc-reply>
```

Get Network Configuration

To get only the network configuration of the device, the network sub tree can be specified in the filter in the get-config call. The following snippet shows the method to retrieve network configuration in IVE.

Example for RPC

```
<rpc message-id="12">  
<get-config>  
  <source>  
    <running/>  
  </source>  
  <filter>  
    <configuration>  
      <system>  
        <network/>  
      </system>  
    </configuration>  
  </filter>  
</get-config>  
</rpc>
```

Configure Network Settings

The network configuration can be updated using the edit-config RPC and specifying the additions or modifications in the network sub tree. For example, the following snippet creates a new VLAN.

Example for RPC

```
<rpc message-id="12">
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <configuration>
      <system>
        <network>
          <vlans>
            <node>localhost2</node>
            <vlan operation="create">
              <name>vlan200</name>
              <settings>
                <vlan-id>200</vlan-id>
                <ip-address>10.20.30.40</ip-address>
                <netmask>255.255.0.0</netmask>
                <default-gateway>10.20.31.1</default-
gateway>
              </settings>
            </vlan>
          </vlans>
        </network>
      </system>
    </configuration>
  </config>
</edit-config>
</rpc>
```

The VLAN created with the previous RPC can be deleted with the following snippet. Only the name of the VLAN is needed for deleting it.

Example for RPC

```
<rpc message-id="12">
<edit-config>
  <target>
```

```

    <running/>
  </target>
  <config>
    <configuration>
      <system>
        <network>
          <vlans>
            <node>localhost2</node>
            <vlan operation="delete">
              <name>vlan200</name>
            </vlan>
          </vlans>
        </network>
      </system>
    </configuration>
  </config>
</edit-config>
</rpc>

```

Create a Realm

The following samples are configurations that are available to both the host system and the logical systems. To retrieve and edit configuration for IVSes, the configuration subtree needs to be embedded inside the <logical-systems> tag and the name of the IVS needs to be specified.

The example below creates a realm with name *Admin Users*. If a realm with the name already exists, the RPC will fail.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <edit-config>
    <target><running/></target>
    <config><configuration>
      <administrators>
        <admin-realms>
          <realm operation="create">
            <name>Admin Users</name>
            <authentication-policy>
              <source-ip>
                <customized>any-ip</customized>
              </source-ip>
            </authentication-policy>
          </realm>
        </admin-realms>
      </administrators>
    </config>
  </edit-config>
</rpc>

```

```

    </allow-admin-signin-internal-port>
</source-ip>
<browser>
  <customized>any-user-agent</customized>
  <user-agent-patterns>
  </user-agent-patterns>
</browser>
<certificate>
  <customized>allow-all-users</customized>
  <cert-key-value-pairs>
  </cert-key-value-pairs>
</certificate>
<password>
  <primary-password-restricted>
    allow-passwords-of-minimum-length
  </primary-password-restricted>
  <primary-password-management>false
  </primary-password-management>
  <primary-password-minimum-length>4
  </primary-password-minimum-length>
  <primary-password-expiration-warning-
days>14</primary-password-expiration-warning-days>
</password>
<host-checker>
  <evaluate-all-policies>false
  </evaluate-all-policies>
  <enforce-all-policies>false
  </enforce-all-policies>
  <evaluate-policy-list>
  </evaluate-policy-list>
  <enforce-policy-list>
  </enforce-policy-list>
  <evaluate-logic>all-policies-must-succeed
  </evaluate-logic>
</host-checker>
<limits>
  <limit-concurrent-users>false
  </limit-concurrent-users>
  <guaranteed-minimum xsi:nil="true"/>
  <maximum>-1</maximum>
</limits>
</authentication-policy>
<role-mapping-rules>
<rule>
  <name>rule 0</name>
  <user-name>
    <test>is</test>
    <user-names>*</user-names>
  </user-name>
  <roles>.Administrators</roles>
  <stop-rules-processing>false

```

```

        </stop-rules-processing>
    </rule>
    <user-selects-role>>false
</user-selects-role>
    <user-selects-roleset>>false
</user-selects-roleset>
</role-mapping-rules>
    <description>Default authentication
realm for administrators
    </description>
    <editing-description>>false
</editing-description>
    <authentication-server>Administrators
</authentication-server>
    <directory-server>None
</directory-server>
    <accounting-server>None
</accounting-server>
    <secondary-authentication-settings>
        <name>-</name>
        <authentication-must-succeed>>true
</authentication-must-succeed>
        <user-name-input>user
</user-name-input>
        <predefined-user-name>
</predefined-user-name>
        <predefined-password>
</predefined-password>
    </secondary-authentication-settings>
    <dynamic-policy>
        <dynamic-policy-evaluation>>false
</dynamic-policy-evaluation>
        <refresh-roles>>false</refresh-roles>
        <refresh-policies>>false
</refresh-policies>
        <refresh-interval>60
</refresh-interval>
    </dynamic-policy>
</realm>
</admin-realms>
</administrators>
</configuration>
</config>
</edit-config>
</rpc>

```

Create a Realm in Logical System

The snippet below creates a realm with name *Admin Users*, in the IVS named *IVS1*.

Example for RPC

```
<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config><configuration>
    <logical-systems>
      <logical-system>
        <name>IVS1</name>
        <administrators>
          <admin-realms>
            <realm operation="create">
              <name>Admin Users</name>
              <authentication-policy>
                <source-ip>
                  <customized>any-ip</customized>
                  <ips></ips>
                  <allow-admin-signin-external-port>>false
                  </allow-admin-signin-external-port>
                  <allow-admin-signin-internal-port>>true
                  </allow-admin-signin-internal-port>
                </source-ip>
                <browser>
                  <customized>any-user-agent</customized>
                  <user-agent-patterns>
                    </user-agent-patterns>
                  </browser>
                <certificate>
                  <customized>allow-all-users</customized>
                  <cert-key-value-pairs>
                    </cert-key-value-pairs>
                  </certificate>
                <password>
                  <primary-password-restricted>
                    allow-passwords-of-minimum-length
                  </primary-password-restricted>
                  <primary-password-management>>false
                  </primary-password-management>
                  <primary-password-minimum-length>4
                  </primary-password-minimum-length>
                  <primary-password-expiration-warning-
days>14</primary-password-expiration-warning-days>
                </password>
              </authentication-policy>
            </realm>
          </admin-realms>
        </administrators>
      </logical-system>
    </logical-systems>
  </config>
</edit-config>
</rpc>
```

```

<evaluate-all-policies>>false
</evaluate-all-policies>
<enforce-all-policies>>false
</enforce-all-policies>
<evaluate-policy-list>
</evaluate-policy-list>
<enforce-policy-list>
</enforce-policy-list>
<evaluate-logic>all-policies-must-succeed
</evaluate-logic>
</host-checker>
<limits>
  <limit-concurrent-users>>false
  </limit-concurrent-users>
  <guaranteed-minimum xsi:nil="true"/>
  <maximum>-1</maximum>
</limits>
</authentication-policy>
<role-mapping-rules>
  <rule>
    <name>rule 0</name>
    <user-name>
      <test>is</test>
      <user-names>*</user-names>
    </user-name>
    <roles>.Administrators</roles>
    <stop-rules-processing>>false
    </stop-rules-processing>
  </rule>
  <user-selects-role>>false
  </user-selects-role>
  <user-selects-roleset>>false
  </user-selects-roleset>
</role-mapping-rules>
  <description>Default authentication
realm for administrators
  </description>
  <editing-description>>false
  </editing-description>
  <authentication-server>Administrators
  </authentication-server>
  <directory-server>None
  </directory-server>
  <accounting-server>None
  </accounting-server>
  <secondary-authentication-settings>
    <name>-</name>
    <authentication-must-succeed>>true
    </authentication-must-succeed>
    <user-name-input>user
    </user-name-input>
  </secondary-authentication-settings>

```

```

        <predefined-user-name>
        </predefined-user-name>
        <predefined-password>
        </predefined-password>
    </secondary-authentication-settings>
    <dynamic-policy>
        <dynamic-policy-evaluation>>false
        </dynamic-policy-evaluation>
        <refresh-roles>>false</refresh-roles>
        <refresh-policies>>false
        </refresh-policies>
        <refresh-interval>60
        </refresh-interval>
    </dynamic-policy>
</realm>
</admin-realms>
</administrators>
</logical-system>
</logical-systems>
</configuration>
</config>
</edit-config>
</rpc>

```

Delete a Realm

This XML snippet deletes an administrator realm with name *Admin Users3*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config><configuration>
    <administrators>
      <admin-realms>
        <realm operation="delete">
          <name>Admin Users3</name>
        </realm>
      </admin-realms>
    </administrators>
  </configuration>
</config>
</edit-config>
</rpc>

```

Create a Role

This XML snippet creates an administrator role with name *role1*.

Example for RPC

```
<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <administrators>
        <admin-roles>
          <admin-role operation="create">
            <name>role1</name>
            <general>
              <overview>
                <description></description>
                <options>
                  <session-options>false</session-options>
                  <ui-options>false</ui-options>
                </options>
              </overview>
              <restrictions>
                <source-ip>
                  <customized>any-ip</customized>
                  <ips>
                    </ips>
                </source-ip>
                <browser>
                  <customized>any-user-agent</customized>
                  <user-agent-patterns>
                    </user-agent-patterns>
                </browser>
                <certificate>
                  <customized>allow-all-users</customized>
                  <cert-key-value-pairs>
                    </cert-key-value-pairs>
                </certificate>
                <host-checker>
                  <host-check-enforce>disable</host-check-
enforce>
                  <host-check-policies></host-check-
policies>
                  <host-check-match>all</host-check-match>
                </host-checker>
              </restrictions>
              <session-options>
                <idle-timeout>9</idle-timeout>
              </session-options>
            </general>
          </admin-role>
        </admin-roles>
      </administrators>
    </configuration>
  </config>
</edit-config>
</rpc>
```

```

        <max-timeout>59</max-timeout>
        <roaming>disabled</roaming>
        <netmask></netmask>
    </session-options>
    <ui-options>
        <header-background-color>#336699</header-
background-color>
        <navigation-menus>auto-enabled</navigation-
menus>
        <show-copyright-notice>>true</show-copyright-
notice>
    </ui-options>
</general>
<administrators>
    <manage-admin-roles>
        <enable>>false</enable>
        <allow-add-remove-admin-
roles>>false</allow-add-remove-admin-roles>
        <access>deny-all</access>
        <role-pages-custom-settings>
            <general>deny</general>
            <system>deny</system>
            <users>deny</users>
            <administrators>deny</administrators>
            <resource-policies>deny</resource-
policies>
            <resource-profiles>deny</resource-
profiles>
        </role-pages-custom-settings>
    </manage-admin-roles>
    <manage-admin-realms>
        <enable>>false</enable>
        <allow-add-remove-admin-
realms>>false</allow-add-remove-admin-realms>
        <access>deny-all</access>
        <realm-pages-custom-settings>
            <general>deny</general>
            <authentication-
policy>deny</authentication-policy>
            <role-mapping>deny</role-mapping>
        </realm-pages-custom-settings>
    </manage-admin-realms>
</administrators>
<users>
    <roles>
        <delegate-user-roles>
            <apply-to-all-roles>>false</apply-to-
all-roles>
            <selected-roles>Users</selected-roles>
            <access>custom-settings</access>
            <role-pages-custom-settings>

```

```

        <general>write</general>
        <web>read</web>
        <files>write</files>
        <sam>deny</sam>
        <telnet-ssh>write</telnet-ssh>
        <terminal-services>write</terminal-
services>
        <network-connect>read</network-
connect>
        <meetings>deny</meetings>
        <email-client>deny</email-client>
        </role-pages-custom-settings>
    </delegate-user-roles>
    <delegate-read-only-roles>
        <apply-to-all-roles>false</apply-to-
all-roles>
        <selected-roles></selected-roles>
    </delegate-read-only-roles>
</roles>
<realms>
    <delegate-user-realms>
        <apply-to-all-realms>false</apply-
to-all-realms>
        <selected-realms>Users</selected-
realms>
        <access>custom-settings</access>
        <realm-pages-custom-settings>
            <general>read</general>
            <role-mapping>deny</role-mapping>
            <authentication-
policy>write</authentication-policy>
            </realm-pages-custom-settings>
        </delegate-user-realms>
        <delegate-read-only-realms>
            <apply-to-all-realms>false</apply-
to-all-realms>
            <selected-realms></selected-realms>
        </delegate-read-only-realms>
    </realms>
</users>
<system>
    <system-tasks>
        <access>deny-all</access>
        <custom-settings>
            <status>deny</status>
            <configuration>deny</configuration>
            <network>deny</network>
            <clustering>deny</clustering>
        </custom-settings>
    </system-tasks>
    <log-monitoring>

```

```

        <access>deny-all</access>
        <custom-settings>
            <events>deny</events>
            <user-access>deny</user-access>
            <admin-access>deny</admin-access>
            <sensors>deny</sensors>
            <client-logs>deny</client-logs>
            <snmp>deny</snmp>
            <statistics>deny</statistics>
        </custom-settings>
    </log-monitoring>
    <authentication>
        <access>deny-all</access>
        <custom-settings>
            <sign-in-policies>deny</sign-in-
policies>
                <sign-in-pages>deny</sign-in-pages>
                <endpoint-security>deny</endpoint-
security>
                <servers>deny</servers>
            </custom-settings>
        </authentication>
        <maintenance-tasks>
            <access>deny-all</access>
            <custom-settings>
                <system>deny</system>
                <archiving>deny</archiving>
            </custom-settings>
        </maintenance-tasks>
    </system>
    <resource-policies>
        <access>custom-settings</access>
        <web>
            <access>read</access>
            <additional-access-policies>
                <policies></policies>
                <access>read</access>
            </additional-access-policies>
        </web>
        <file>
            <access>write</access>
            <additional-access-policies>
                <policies></policies>
                <access>read</access>
            </additional-access-policies>
        </file>
        <sam>
            <access>deny</access>
            <additional-access-policies>

```

```

        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</sam>
<telnet-ssh>
    <access>write</access>
    <additional-access-policies>
        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</telnet-ssh>
<terminal-services>
    <access>read</access>
    <additional-access-policies>
        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</terminal-services>
<network-connect>
    <access>deny</access>
    <additional-access-policies>
        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</network-connect>
<email-client>
    <access>deny</access>
</email-client>
</resource-policies>
<resource-profiles>
    <access>deny-all</access>
    <web>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>
    </web>
    <file>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>
    </file>
</sam>
    <access>deny</access>
    <additional-access-profiles>
        <profiles></profiles>
        <access>read</access>
    </additional-access-profiles>

```

```

        </sam>
        <telnet_ssh>
            <access>deny</access>
            <additional-access-profiles>
                <profiles></profiles>
                <access>read</access>
            </additional-access-profiles>
        </telnet_ssh>
        <terminal-services>
            <access>deny</access>
            <additional-access-profiles>
                <profiles></profiles>
                <access>read</access>
            </additional-access-profiles>
        </terminal-services>
    </resource-profiles>
</admin-role>
<admin-role-default-options>
    <session-options>
        <idle-timeout>10</idle-timeout>
        <max-timeout>60</max-timeout>
        <roaming>enabled</roaming>
        <netmask></netmask>
    </session-options>
    <ui-options>
        <header-background-color>#336699</header-
background-color>
        <navigation-menus>auto-enabled</navigation-
menus>
        <show-copyright-notice>>true</show-copyright-
notice>
    </ui-options>
</admin-role-default-options>
</admin-roles>
</administrators>
</configuration>
</config>
</edit-config>
</rpc>

```

Create a Role in Logical System

The snippet below creates an admin role *role1*, in IVS named *IVS1*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>

```

```

<target><running/></target>
<config>
  <configuration>
    <logical-systems>
      <logical-system>
        <name>IVS1</name>
        <administrators>
          <admin-roles>
            <admin-role operation="create">
              <name>role1</name>
              <general>
                <overview>
                  <description></description>
                  <options>
                    <session-options>false</session-options>
                    <ui-options>false</ui-options>
                  </options>
                </overview>
                <restrictions>
                  <source-ip>
                    <customized>any-ip</customized>
                    <ips>
                      </ips>
                    </source-ip>
                  <browser>
                    <customized>any-user-agent</customized>
                    <user-agent-patterns>
                      </user-agent-patterns>
                    </browser>
                  <certificate>
                    <customized>allow-all-users</customized>
                    <cert-key-value-pairs>
                      </cert-key-value-pairs>
                    </certificate>
                  <host-checker>
                    <host-check-enforce>disable</host-check-
enforce>
                    <host-check-policies></host-check-
policies>
                    <host-check-match>all</host-check-match>
                  </host-checker>
                </restrictions>
                <session-options>
                  <idle-timeout>9</idle-timeout>
                  <max-timeout>59</max-timeout>
                  <roaming>disabled</roaming>
                  <netmask></netmask>
                </session-options>
                <ui-options>
                  <header-background-color>#336699</header-
background-color>

```

```

        <navigation-menus>auto-enabled</navigation-
menus>
        <show-copyright-notice>>true</show-copyright-
notice>
        </ui-options>
    </general>
    <administrators>
        <manage-admin-roles>
            <enable>>false</enable>
            <allow-add-remove-admin-
roles>>false</allow-add-remove-admin-roles>
            <access>deny-all</access>
            <role-pages-custom-settings>
                <general>deny</general>
                <system>deny</system>
                <users>deny</users>
                <administrators>deny</administrators>
                <resource-policies>deny</resource-
policies>
                <resource-profiles>deny</resource-
profiles>
            </role-pages-custom-settings>
        </manage-admin-roles>
        <manage-admin-realms>
            <enable>>false</enable>
            <allow-add-remove-admin-
realms>>false</allow-add-remove-admin-realms>
            <access>deny-all</access>
            <realm-pages-custom-settings>
                <general>deny</general>
                <authentication-
policy>deny</authentication-policy>
                <role-mapping>deny</role-mapping>
            </realm-pages-custom-settings>
        </manage-admin-realms>
    </administrators>
    <users>
        <roles>
            <delegate-user-roles>
                <apply-to-all-roles>>false</apply-to-
all-roles>
                <selected-roles>Users</selected-roles>
                <access>custom-settings</access>
                <role-pages-custom-settings>
                    <general>write</general>
                    <web>read</web>
                    <files>write</files>
                    <sam>deny</sam>
                    <telnet-ssh>write</telnet-ssh>
                    <terminal-services>write</terminal-
services>

```

```

connect>
    <network-connect>read</network-
connect>
    <meetings>deny</meetings>
    <email-client>deny</email-client>
    </role-pages-custom-settings>
</delegate-user-roles>
<delegate-read-only-roles>
    <apply-to-all-roles>>false</apply-to-
all-roles>
    <selected-roles></selected-roles>
    </delegate-read-only-roles>
</roles>
<realms>
    <delegate-user-realms>
to-all-realms>
    <apply-to-all-realms>>false</apply-
realms>
    <selected-realms>Users</selected-
realms>
    <access>custom-settings</access>
    <realm-pages-custom-settings>
    <general>read</general>
    <role-mapping>deny</role-mapping>
    <authentication-
policy>write</authentication-policy>
    </realm-pages-custom-settings>
    </delegate-user-realms>
    <delegate-read-only-realms>
to-all-realms>
    <apply-to-all-realms>>false</apply-
realms>
    <selected-realms></selected-realms>
    </delegate-read-only-realms>
</realms>
</users>
<system>
    <system-tasks>
    <access>deny-all</access>
    <custom-settings>
    <status>deny</status>
    <configuration>deny</configuration>
    <network>deny</network>
    <clustering>deny</clustering>
    </custom-settings>
</system-tasks>
<log-monitoring>
    <access>deny-all</access>
    <custom-settings>
    <events>deny</events>
    <user-access>deny</user-access>
    <admin-access>deny</admin-access>
    <sensors>deny</sensors>
    <client-logs>deny</client-logs>

```

```

        <snmp>deny</snmp>
        <statistics>deny</statistics>
    </custom-settings>
</log-monitoring>
<authentication>
    <access>deny-all</access>
    <custom-settings>
        <sign-in-policies>deny</sign-in-
policies>
        <sign-in-pages>deny</sign-in-pages>
        <endpoint-security>deny</endpoint-
security>
        <servers>deny</servers>
    </custom-settings>
</authentication>
<maintenance-tasks>
    <access>deny-all</access>
    <custom-settings>
        <system>deny</system>
        <archiving>deny</archiving>
</maintenance-tasks>
<troubleshooting>deny</troubleshooting>
</troubleshooting>
</custom-settings>
</maintenance-tasks>
</system>
<resource-policies>
    <access>custom-settings</access>
    <web>
        <access>read</access>
        <additional-access-policies>
            <policies></policies>
            <access>read</access>
        </additional-access-policies>
    </web>
    <file>
        <access>write</access>
        <additional-access-policies>
            <policies></policies>
            <access>read</access>
        </additional-access-policies>
    </file>
    <sam>
        <access>deny</access>
        <additional-access-policies>
            <policies></policies>
            <access>read</access>
        </additional-access-policies>
    </sam>
    <telnet-ssh>
        <access>write</access>
        <additional-access-policies>

```

```

        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</telnet-ssh>
<terminal-services>
    <access>read</access>
    <additional-access-policies>
        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</terminal-services>
<network-connect>
    <access>deny</access>
    <additional-access-policies>
        <policies></policies>
        <access>read</access>
    </additional-access-policies>
</network-connect>
<email-client>
    <access>deny</access>
</email-client>
</resource-policies>
<resource-profiles>
    <access>deny-all</access>
    <web>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>
    </web>
    <file>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>
    </file>
    <sam>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>
    </sam>
    <telnet_ssh>
        <access>deny</access>
        <additional-access-profiles>
            <profiles></profiles>
            <access>read</access>
        </additional-access-profiles>

```

```

        </telnet_ssh>
        <terminal-services>
            <access>deny</access>
            <additional-access-profiles>
                <profiles></profiles>
                <access>read</access>
            </additional-access-profiles>
        </terminal-services>
    </resource-profiles>
</admin-role>
<admin-role-default-options>
    <session-options>
        <idle-timeout>10</idle-timeout>
        <max-timeout>60</max-timeout>
        <roaming>enabled</roaming>
        <netmask></netmask>
    </session-options>
    <ui-options>
        <header-background-color>#336699</header-
background-color>
        <navigation-menus>auto-enabled</navigation-
menus>
        <show-copyright-notice>true</show-copyright-
notice>
    </ui-options>
</admin-role-default-options>
</admin-roles>
</administrators>
</logical-system></logical-systems>
</configuration>
</config>
</edit-config>
</rpc>

```

Delete a Role

The following snippet deletes an administrator role with name *role1*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
    <target><running/></target>
    <config>
        <configuration>
            <administrators>
                <admin-roles>

```

```

        <admin-role operation="delete">
            <name>role1</name>
        </admin-role>
    </admin-roles>
</administrators>
</configuration>
</config>
</edit-config>
</rpc>

```

Create a Resource Profile

The snippet below creates a telnet resource profile for the role named *Users*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
    <target><running/></target>
    <config>
        <configuration>
            <users>
                <resource-profiles>
                    <telnet-ssh-profiles>
                        <telnet-ssh-profile operation="create">
                            <name>New Telnet Profile</name>
                            <description>Created through
Inbound</description>
                            <host>10.20.30.40</host>
                            <port>2345</port>
                            <roles>Users</roles>
                        </telnet-ssh-profile>
                    </telnet-ssh-profiles>
                </resource-profiles>
            </users>
        </configuration>
    </config>
</edit-config>
</rpc>

```

Create a Resource Profile in Logical System

The snippet below creates a telnet resource profile for the role named *Users*, in IVS named *IVS1*.

Example for RPC

```
<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <logical-systems>
        <logical-system>
          <name>IVS1</name>
          <users>
            <resource-profiles>
              <telnet-ssh-profiles>
                <telnet-ssh-profile operation="create">
                  <name>New Telnet Profile</name>
                  <description>Created through
Inbound</description>
                  <host>10.20.30.40</host>
                  <port>2345</port>
                  <roles>Users</roles>
                </telnet-ssh-profile>
              </telnet-ssh-profiles>
            </resource-profiles>
          </users>
        </logical-system>
      </logical-systems>
    </configuration>
  </config>
</edit-config>
</rpc>
```

Delete a Resource Profile

The snippet below deletes a telnet resource profile named *New Telnet Profile* for the role named *Users*.

Example for RPC

```
<rpc message-id='101'
```

```

xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <users>
        <resource-profiles>
          <telnet-ssh-profiles>
            <telnet-ssh-profile operation="delete">
              <name>New Telnet Profile</name>
            </telnet-ssh-profile>
          </telnet-ssh-profiles>
        </resource-profiles>
      </users>
    </configuration>
  </config>
</edit-config>
</rpc>

```

Create a Resource Policy

The example below creates a Windows resource access control list and assigns it to the *Users* role.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <edit-config>
    <target><running/></target>
    <config>
      <configuration>
        <users>
          <resource-policies>
            <file-policies>
              <file-win-acls>
                <file-win-acl operation="create">
                  <name>My Domain Windows Policy</name>
                  <description>Created through Inbound
DMI</description>
                  <resources>https://*.mydomain.com/*</resources>

                  <parent-type>none</parent-type>
                  <apply>selected</apply>
                  <role>Users</role>
                  <read-only>>false</read-only>
                </file-win-acl>
              </file-win-acls>
            </file-policies>
          </resource-policies>
        </users>
      </configuration>
    </config>
  </edit-config>
</rpc>

```

```

        </file-win-acls>
    </file-policies>
</resource-policies>
</users>
</configuration>
</config>
</edit-config>
</rpc>

```

Create a Resource Policy in Logical System

The example below creates a Windows resource access control list in *IVS1* and assigns it to the *Users* role.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <edit-config>
    <target><running/></target>
    <config>
      <configuration>
        <logical-systems>
          <logical-system>
            <name>IVS1</name>
            <users>
              <resource-policies>
                <file-policies>
                  <file-win-acls>
                    <file-win-acl operation="create">
                      <name>My Domain Windows Policy</name>
                      <description>Created through Inbound
DMI</description>
                      <resources>https://*.mydomain.com/*</resources>

                      <parent-type>none</parent-type>
                      <apply>selected</apply>
                      <role>Users</role>
                      <read-only>>false</read-only>
                    </file-win-acl>
                  </file-win-acls>
                </file-policies>
              </resource-policies>
            </users>
          </logical-system>
        </logical-systems>
      </configuration>

```

```

    </config>
  </edit-config>
</rpc>

```

Delete a Resource Policy

The following snippet deletes a Windows resource access control list resource policy with name *My Domain Windows Policy*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
  <edit-config>
    <target><running/></target>
    <config>
      <configuration>
        <users>
          <resource-policies>
            <file-policies>
              <file-win-acls>
                <file-win-acl operation="create">
                  <name>My Domain Windows Policy</name>
                  <description>Created through Inbound
DMI</description>
                  <resources>https://*.mydomain.com/*</resources>

                  <parent-type>none</parent-type>
                  <apply>selected</apply>
                  <role>Users</role>
                  <read-only>false</read-only>
                </file-win-acl>
              </file-win-acls>
            </file-policies>
          </resource-policies>
        </users>
      </configuration>
    </config>
  </edit-config>
</rpc>

```

Create a Web Bookmark for a Role

The following example creates a web bookmark for the user role *Users*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <users>
        <user-roles>
          <user-role>
            <name>Users</name>
            <web>
              <web-bookmarks>
                <bookmark operation='create'>
                  <name>Google</name>
                  <parent>--none--</parent>
                  <description>Google web site
</description>
                  <standard>
                    <url>http://www.google.com</url>
                  </standard>
                </bookmark>
              </web-bookmarks>
            </web>
          </user-role>
        </user-roles>
      </users>
    </configuration>
  </config>
</edit-config>
</rpc>

```

Create a Web Bookmark for a Role in Logical System

The following example creates a web bookmark for the user role *Users* in IVS named *IVS1*.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <logical-systems>

```

```

<logical-system>
  <name>IVS1</name>
  <users>
    <user-roles>
      <user-role>
        <name>Users</name>
        <web>
          <web-bookmarks>
            <bookmark operation='create'>
              <name>Gmail</name>
              <parent>--none--</parent>
              <description>Google mail</description>
              <standard>
                <url>http://mail.google.com</url>
              </standard>
            </bookmark>
          </web-bookmarks>
        </web>
      </user-role>
    </user-roles>
  </users>
</logical-system>
</logical-systems>
</configuration>
</config>
</edit-config>
</rpc>

```

Delete a Web Bookmark for a Role

The following snippet deletes a web bookmark named *Google* which is created for the user role *Users*. Note that multiple identifiers are specified in the RPC. Both the role name and the bookmark name need to be specified.

Example for RPC

```

<rpc message-id='101'
xmlns='urn:ietf:params:xml:ns:netconf:base:1.0'>
<edit-config>
  <target><running/></target>
  <config>
    <configuration>
      <users>
        <user-roles>
          <user-role>
            <name>Users</name>
            <web>

```

```

        <web-bookmarks>
            <bookmark operation="delete">
                <name>Google</name>
            </bookmark>
        </web-bookmarks>
    </web>
</user-role>
</user-roles>
</users>
</configuration>
</config>
</edit-config>
</rpc>

```

Get Syslog Events

The `get-syslog-events` RPC can be used to receive the syslog events. The administrator is advised to use a separate dedicated channel for receiving the events, as sending other RPCs in the same channel could potentially create confusion in interpreting the RPC replies received. **default-log-messages** is the only log stream supported by an IVE device.

The following example illustrates getting syslog events from the IVE.

Example for RPC

```

<rpc message-id='101'>
  <get-syslog-events>
    <stream>default-log-messages</stream>
  </get-syslog-events>
</rpc>

```

All the syslog events that occur after the RPC is issued are notified as they occur in the inbound session. Filters can be applied to be notified of specific syslog events. For example, the following snippet notifies only the configuration change events.

Example for RPC

```

<rpc message-id='101'>
  <get-syslog-events>
    <stream>default-log-messages</stream>
    <event>configuration-change</event>
  </get-syslog-events>
</rpc>

```

Syslog events can be filtered for logical systems. The following snippet sets the IVE to notify those syslog events that are pertinent only to *IVS1*.

Example for RPC

```
<rpc message-id='101'>
  <get-syslog-events>
    <stream>default-log-messages</stream>
    <parameter>ivs='IVS1'</parameter>
  </get-syslog-events>
</rpc>
```

Limitations in get-syslog-events

DMI recommends following the draft “draft-shafer-netconf-syslog-00.txt” for implementing the get-syslog-events RPC. Some limitations in the implementation of the RPC from the specifications in the draft are mentioned below:

1. **recorded** argument is not supported in the rpc request. This argument is ignored if specified in the request.
2. **text-pattern** argument is not supported in the rpc request. This argument is ignored if specified in the request.
3. **process** argument is not supported in the rpc request. This argument is ignored if specified in the request.
4. **priority** argument is not supported in the rpc request. This argument is ignored if specified in the request.
5. The time format specified in **start-time** and **stop-time** parameters have to be in UTC time format. Local time formats with a clock difference from UTC time is not supported.
6. If **event** parameter has **configuration-change** as the value in the RPC request, all the other filters are ignored. This is used to capture asynchronous configuration notifications from the IVE.
7. **parameter** argument can be used to filter IVE logs. The filter implementation supports custom expressions that can be used as a query for log filters. The implementation does not support generic regular expressions.

Error conditions

This section documents some of the probable error conditions and the reasons they could occur.

| Error | Possible reason |
|---|--|
| In the DMI page in UI, the Inbound status reports “Error”, when the Inbound is enabled. | The port configured for inbound might already be in use by another process. |
| “Connection Refused” error when trying to connect to inbound | The inbound might not have been enabled or the port might be used by some other process other than DMI. |
| Login unsuccessful when an inbound connection attempt is made. | If a DMI session already exists for a user, another session can not be established. Check in the Admin Access log for information. |
| When an RPC is issued to get or edit IVS related configuration, there is no error, but no data is returned in the rpc-reply | It is possible that the data requested is pertinent only for the host system or the root IVS. |

References

1. RFC 4741: NETCONF Configuration Protocol
<http://www.ietf.org/rfc/rfc4741.txt>
2. DMI Specification version 1.2
3. Secure Shell connectivity tool:
<http://www.openssh.com/>