IPVS DeviceAdminAgent API

IPVS DeviceAdminAgent API

Table of Contents

List of Examples

Glossary

AppController
AppController running on managementserver. Controller will manage all modules of the system.

DeviceAdminManager
DeviceAdminManager Running on Appserver. Manager will do policy check of device requests and send the requests to DeviceAgent

DeviceAdminAgent
DeviceAdminAgent Running on Device. Handles the Request from DeviceAdminManager or Internal DeviceServer.

DeviceAdminServer
DeviceAdminServer Running on Device. Which will send or receive commands or responses from JBUS/proxyclient/cli.

Introduction

This document contains the IPVS Device Admin Agent API. Agent will accept requests from AppController/Clients or internal DeviceAdminServer

Chapter 1. Discovery API

API Calls used in Discovery process

External API

Requests Received from DeviceAdminManager or Clients

GetDevice

Returns the Device information in RAW format.

Example 1.1. GetDeviceRequest

```
  <GetDevice selectFormat='RAW' />
```

Example 1.2. GetDeviceResponse

```
  <GetDevice from='0' total='0'>
 <Device  title='' type='' state='' password='' ownerUserJID=''>
  <Groups />
  <DevicePrivateData />
  <TemplateNID />
  <PortList />
  <DeviceConfig> <!-- Device specific response --> </DeviceConfig>
  <DeviceStatus> <!-- Device specific response --> </DeviceStatus>
  <DeviceList />
 </Device>
</GetDevice>
```

SetDeviceConfig

update the device with given config

Example 1.3. SetDeviceConfigRequest

```
   <SetDeviceConfig doNotSaveToFlash='true/false'>
   <DeviceConfig Dirty="CLEAN">
            <ServiceElementsConfig>
              <XMPPAgentServiceConfig Dirty="MODIFY">
                    <Enable>true</Enable>
                    <XMPPServer>0.0.0.0</XMPPServer>
                    <DeviceName>xp2001</DeviceName>
                    <DevicePassword>Ipvs1234</DevicePassword>
                    <Domain>default</Domain>
                    <ServiceDomain>default</ServiceDomain>
              </XMPPAgentServiceConfig>
              <NetworkElementsConfig>
                 <RouteTableConfig Dirty="MODIFY">
                   <RouteEntry Dirty="MODIFY" ListIndex="1">
                         <Destination>0.0.0.0</Destination>
                         <Netmask>0.0.0.0</Netmask>
                         <Gateway>192.168.1.1</Gateway>
                   </RouteEntry>
                 </RouteTableConfig>
                 <EthernetPortTableConfig>
                   <EthernetPortConfig Dirty="MODIFY" ListIndex="1">
                         <PortID>1</PortID>
                         <EnableInterface>true</EnableInterface>
                               <IsPrimary>true</IsPrimary>
                               <EnableMulticast>false</
EnableMulticast>
                         <IPConfig>
                               <UseDHCP>false</UseDHCP>
                               <IPAddress>192.168.1.196</IPAddress>
                               <Netmask>255.255.0.0</Netmask>
                         </IPConfig>
                   </EthernetPortConfig>
              </EthernetPortTableConfig>
         </NetworkElementsConfig>
         </ServiceElementsConfig>
   </DeviceConfig>
 </SetDeviceConfig>
```

Example 1.4. SetDeviceConfigResponse

```
   <SetDeviceConfig/>
```

Internal API

Requests Received from DeviceAdminServer

Chapter 2. Startup API

API Calls used on device startup

External API

Requests Received from DeviceAdminManager or Clients

Internal API

Requests Received from DeviceAdminClient

GetPublishInfo

Returns the Device information Which needs to be published to appserver

Example 2.1. GetPublishInfoRequest

```
  <GetPublishInfo/>
```

Example 2.2. GetPublishInfoResponse

```
  <GetPublishInfo>
 <Device ownerUserJID=''>
  <PortList>
     <Port portID='' portType=''/>
  </PortList>
  <DeviceConfig> <!-- Device specific response --> </DeviceConfig>
  <DeviceStatus> <!-- Device specific response --> </DeviceStatus>
 </Device>
</GetPublishInfo>
```

Chapter 3. DeviceConfig API

API Calls used to update Device Config From Clients/AppControler or publish the Config to App Controler

External API

Requests Received from DeviceAdminManager or Clients

SetDeviceConfig

Set the given Config input

Example 3.1. SetDeviceConfigRequest

```
    <SetDeviceConfig doNotSaveToFlash='true/false'>
  <DeviceConfig Dirty="CLEAN">
      <SystemElementsConfig Dirty="CLEAN">
         <!-- Choice of device specific  System Info -->
      </SystemElementsConfig>
      <ServiceElementsConfig Dirty="CLEAN">
         <!-- Choice of device specific  Services-->
      </ServiceElementsConfig>
      <NetworkElementsConfig Dirty="CLEAN">
         <!-- Network Config Common for all devices-->
      </NetworkElementsConfig>
       <StorageElementsConfig Dirty="CLEAN">
         <!-- Common for all DMS and MS Devices-->
      </StorageElementsConfig>
      <AVCodecElementsConfig Dirty="CLEAN">
         <!-- Choice of device specific  AVCodec Elements -->
      </AVCodecElementsConfig>
  </DeviceConfig>
</SetDeviceConfig>
```

Example 3.2. SetDeviceConfigResponse

```
<SetDeviceConfig/>
```

Example 3.3. Enable DHCP to get local ipaddress

```
    <SetDeviceConfig doNotSaveToFlash='true/false'>
  <DeviceConfig Dirty="CLEAN">
       <NetworkElementsConfig Dirty="CLEAN">
         <EthernetPortTableConfig>
            <EthernetPortConfig Dirty="MODIFY" ListIndex="1">
               <PortID>1</PortID>
               <EnableInterface>true</EnableInterface>
               <IsPrimary>true</IsPrimary>
               <EnableMulticast>true</EnableMulticast>
               <MTU>1500</MTU>

               <IPConfig>
                  <UseDHCP>true</UseDHCP>
                  <IPAddress>169.254.0.1</IPAddress>
                  <Netmask>255.255.0.0</Netmask>
               </IPConfig>
            </EthernetPortConfig>
         </EthernetPortTableConfig>
      </NetworkElementsConfig>
  </DeviceConfig>
</SetDeviceConfig>
```

Internal API

Requests Received from DeviceAdminClient

PublishDeviceConfig

Sends Request to appserver to update DeviceConfig in database

Example 3.4. Publish the DHCP config to AppController

```
 <PublishDeviceConfig  byNID='1234' xpath='//EthernetPortConfig'
operation='UPDATE'>
 <EthernetPortConfig Dirty="CLEAN" ListIndex="1">
        <PortID>1</PortID>
        <EnableInterface>true</EnableInterface>
        <IsPrimary>true</IsPrimary>
        <EnableMulticast>true</EnableMulticast>
        <MTU>1500</MTU>

        <IPConfig>
           <UseDHCP>true</UseDHCP>
           <IPAddress>169.254.0.1</IPAddress>
           <Netmask>255.255.0.0</Netmask>
        </IPConfig>
     </EthernetPortConfig>
</PublishDeviceConfig>
```

Example 3.5. PublishDeviceConfigResponse

```
 <PublishDeviceConfig/>
```

Chapter 4. DeviceStatus API

API Calls used to publish the device status to AppController

External API

Requests Received from DeviceAdminManager

GetDynamicDeviceStatus

Returns current status of device

Example 4.1. GetDynamicDeviceStatusRequest

```
     <GetDynamicDeviceStatus selectFormat='RAW'/>
```

Example 4.2. GetDynamicDeviceStatusResponse

```
   <GetDynamicDeviceStatus>
  <DeviceStatus Dirty="CLEAN">
   <SystemElementsStatus Dirty="CLEAN">
    <!-- Choice of device specific System Status -->
   </SystemElementsStatus>
   <ServiceElementsStatus Dirty="CLEAN">
    <!-- Choice of device specific Services status -->
   </ServiceElementsStatus>
   <NetworkElementsStatus Dirty="CLEAN">
    <NATZoneTableStatus Queryable="true" State="OK">
     <ParametersList>
      <Parameters>
       <Parameter name="" type="Float" units="" source=""></Parameter>
      </Parameters>
     </ParametersList>
    </NATZoneTableStatus>
    <EthernetPortTableStatus>
     <EthernetPortStatus Queryable="true" State="OK">
      <ParametersList>
       <Parameters>
        <Parameter name="" type="Float" units="" source=""></
Parameter>
       </Parameters>
      </ParametersList>
     </EthernetPortStatus>
    </EthernetPortTableStatus>
    <!-- Network Status Common for all devices -->
   </NetworkElementsStatus>
   <StorageElementsStatus Dirty="CLEAN">
    <!-- Common for all DMS and MS Devices -->
   </StorageElementsStatus>
   <AVCodecElementsStatus Dirty="CLEAN">
    <!-- Choice of device specific AVCodec Elements -->
   </AVCodecElementsStatus>
  </DeviceStatus>
 </GetDynamicDeviceStatus>
```

Internal API

Requests Received from DeviceAdminServer

PublishDeviceStatus

Sends Request to appserver to update DeviceStatus in database

Example 4.3. Publishing network status to appserver

```
 <PublishDeviceStatus  byNID='1234' xpath='//EthernetPortTableStatus'
operation='UPDATE'>
   <EthernetPortTableStatus>
           <EthernetPortStatus Queryable="true" State="OK">
              <ParametersList>
                 <Parameters>
                    <Parameter name="" type="Float" units=""
source=""></Parameter>
                 </Parameters>
              </ParametersList>
           </EthernetPortStatus>
        </EthernetPortTableStatus>
</PublishDeviceStatus>
```

Example 4.4. PublishDevicStatusResponse

```
 <PublishDeviceStatus/>
```

Chapter 5. DeviceCommand API

API Calls used to Run commands on devices like Upgrade,Reboot,RevertDefault, SaveToFlash

External API

Requests Received from DeviceAdminManager

ExecuteDeviceCommand

Execute the given action

Example 5.1. ExecuteDeviceCommandRequest

```
     <ExecuteDeviceCommand command='UPGRADE/REBOOT/REVERTDEFAULT/
SAVE'>
      <CommandParam name='' value=''/>
      <CommandParam name='' value=''/>
      <CommandParam name='' value=''/>
     </ExecuteDeviceCommand>
```

Example 5.2. ExecuteDeviceCommandResponse

```
   <ExecuteDeviceCommand/>
```

Internal API

Requests Received from DeviceAdminServer