DeviceAdminService API

DeviceAdminService API

Table of Contents

List of Examples

Glossary

Device Interface	This represents the external interface to a device.
Device Interface API	The API accepted by the device on its external interface
DeviceAdminService API	The API accepted by the device on its external interface for DeviceAdminService

Introduction

This document contains the description of the device interface API for deviceadmin services

Chapter 1. Discovery API

API Calls used in Discovery process

GetDevice

Returns the Device information in RAW format.

Example 1.1. GetDeviceRequest

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

Example 1.2. GetDeviceResponse

```
   <GetDevice>
  <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

- *Dirty: possible values are CLEAN/ADD/MODIFY/DELETE. (Applies the config only if Dirty!=CLEAN)*
  - CLEAN: No update required
  - ADD: Add given entry to table. Example: Adding a route to route table.
  - MODIFY: Modifies the given config. Example: Updating exiting route config
  - DELETE: Deletes the given entry from table. Example deleting a route from route table

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/>
```

Chapter 2. DeviceConfig API

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

GetDeviceConfig

Returns the Device configuration information

Example 2.1. GetDeviceConfigRequest

- *xpath: (optional) use when you want to get specific config information*

```
  <GetDeviceConfig xpath=""/>
```

Example 2.2. GetDeviceConfigResponse

```
  <GetDeviceConfig>
  <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>
</GetDevice>
```

SetDeviceConfig

Updates the given Configuration

- *Dirty: possible values are CLEAN/ADD/MODIFY/DELETE. (Applies the config only if Dirty!=CLEAN)*

- CLEAN: No update required
- ADD: Add given entry to table. Example: Adding a route to route table.
- MODIFY: Modifies the given config. Example: Updating exiting route config
- DELETE: Deletes the given entry from table. Example deleting a route from route table

- *doNotSaveToFlash: (optional) if true it will not update the given configuration to flash. Default value is false*

Example 2.3. 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 2.4. SetDeviceConfigResponse

```
 <SetDeviceConfig/>
```

Example 2.5. 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>
```

Chapter 3. DeviceStatus API

API Calls used to publish the device status to AppController

GetDeviceStatus

Returns current status of device

Example 3.1. GetDynamicDeviceStatusRequest

- *xpath: (optional) use when you want to get specific status information*

```
          <GetDeviceStatus xpath=""/>
```

Example 3.2. GetDeviceStatusResponse

```
   <GetDeviceStatus>
  <DeviceStatus>
   <SystemElementsStatus>
    <!-- Choice of device specific System Status -->
   </SystemElementsStatus>
   <ServiceElementsStatus>
    <!-- Choice of device specific Services status -->
   </ServiceElementsStatus>
   <NetworkElementsStatus>
    <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>
    <!-- Common for all DMS and MS Devices -->
   </StorageElementsStatus>
   <AVCodecElementsStatus>
    <!-- Choice of device specific AVCodec Elements -->
   </AVCodecElementsStatus>
  </DeviceStatus>
 </GetDeviceStatus>
```

Chapter 4. DeviceCommand API

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

GetDeviceCommand

Returns the supported commands information

Example 4.1. GetDeviceCommandRequest

```
<GetDeviceCommand/>
```

Example 4.2. GetDeviceCommandResponse

```
  <GetDeviceCommand>
   <DeviceCommand>
       <SystemElementsCommand">
          <!-- Choice of device specific  System Info -->
       </SystemElementsCommand>
       <ServiceElementsCommand">
          <!-- Choice of device specific  Services-->
       </ServiceElementsCommand>
       <NetworkElementsCommand">
          <!-- Network Config Common for all devices-->
       </NetworkElementsCommand>
        <StorageElementsCommand">
          <!-- Common for all DMS and MS Devices-->
       </StorageElementsCommand>
       <AVCodecElementsCommand">
          <!-- Choice of device specific  AVCodec Elements -->
       </AVCodecElementsCommand>
   </DeviceConfig>
</GetDevice>
```

ExecuteDeviceCommand

Execute the given action

Example 4.3. ExecuteDeviceCommandRequest

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

Example 4.4. ExecuteDeviceCommandResponse

```
       <ExecuteDeviceCommand/>
```