



CHAPTER 2

ANM Web Services API

This chapter describes the ANM Web Services APIs, which allows you to manage the Cisco Application Control Engine (ACE) module, Cisco Application Control Engine (ACE) appliance, Cisco Content Services Switch (CSS) and Cisco Content Switching Module (CSM), Cisco Content Switching Module with SSL (CSM-S), and Cisco Global Site Selector (GSS) using ANM. ANM Web Services provide APIs to support configuration functionalities. All APIs are implemented as synchronous calls and each call is a transaction.

This chapter contains the mapping table and describes the following ANM APIs:

- [Mapping Table](#)
- [OperationManager Web Service APIs](#)
- [ApplicationTemplateManager Web Service APIs](#)

Mapping Table

Table 2-1 displays the mapping of each API with the specified role and the domain, where

- F = Full access
- N = No access

Table 2-1 Mapping Table

API Name	Role Name											
	ACE-Admin	ANM-Admin	Network-Admin	Network-Monitor	Org-Admin	Security-Admin	Server-Appln-Maintenance	Server-Maintenance	SLB-Admin	SSL-Admin	SSL-Cert-Key-Admin	VM-Mapper
listDeviceids	N	F	N	F	F	N	N	N	N	N	N	N
listVirtualContexts	F	F	F	F	F	N	N	N	N	N	N	N
listVirtualServers	F	F	N	F	F	N	F	F	F	N	N	N

Table 2-1 Mapping Table (continued)

API Name	Role Name											
	ACE-Admin	ANM-Admin	Network-Admin	Network-Monitor	Org-Admin	Security-Admin	Server-Appln-Maintenance	Server-Maintenance	SLB-Admin	SSL-Admin	SSL-Cert-Key-Admin	VM-Mapper
listServerFarms	F	F	N	F	F	N	F	F	F	N	N	N
listRservers	F	F	N	F	F	N	F	F	F	N	N	N
listServerfarmRservers	F	F	N	F	F	N	F	F	F	N	N	N
listAnswers	N	F	N	N	F	N	F	F	F	N	N	N
listDnsRules	N	F	N	N	F	N	F	F	F	N	N	N
activateVirtualServer	F	F	N	N	F	N	N	N	F	N	N	N
suspendVirtualServer	F	F	N	N	F	N	N	N	F	N	N	N
addRserverToServerfarm	F	F	N	F	F	N	F	F	F	N	N	N
removeRserverFromServerfarm	F	F	N	N	F	N	F	N	F	N	N	N
activateServerfarmRserver	F	F	N	N	F	N	F	F	F	N	N	N
suspendServerfarmRserver	F	F	N	N	F	N	F	F	F	N	N	N
changeServerfarmRserverWeight	F	F	N	N	F	N	F	F	F	N	N	N
activateAnswer	N	F	N	N	F	N	N	F	F	N	N	N
suspendAnswer	N	F	N	N	F	N	N	F	F	N	N	N
activateDnsRule	N	F	N	N	F	N	N	F	F	N	N	N
suspendDnsRule	N	F	N	N	F	N	N	F	F	N	N	N
getVMMappingInfo	N	F	N	N	F	N	N	N	N	N	N	F
listTemplateDefinitions	F	F	N	N	F	N	N	N	N	N	N	N
getTemplateDefinitionMetadata	F	F	N	N	F	N	N	N	N	N	N	N
createTemplateInstance	F	F	N	N	F	N	N	N	N	N	N	N
listTemplateInstances	F	F	N	N	F	N	N	N	N	N	N	N
deleteTemplateInstance	F	F	N	N	F	N	N	N	N	N	N	N
listCheckpoints	F	F	N	N	F	N	N	N	N	N	N	N
createCheckpoint	F	F	N	N	F	N	N	N	N	N	N	N
deleteCheckpoint	F	F	N	N	F	N	N	N	N	N	N	N
rollbackCheckpoint	F	F	N	N	F	N	N	N	N	N	N	N
freeFormShow	F	F	N	N	F	N	N	N	N	N	N	N

OperationManager Web Service APIs

This section describes the APIs associated with the OperationManager Web Service, which are as follows:

- [login](#)
- [logout](#)
- [listDeviceids](#)
- [listVirtualContexts](#)
- [listVirtualServers](#)
- [listServerFarms](#)
- [listRservers](#)
- [listServerfarmRservers](#)
- [listAnswers](#)
- [listDnsRules](#)
- [activateVirtualServer](#)
- [suspendVirtualServer](#)
- [addRserverToServerfarm](#)
- [removeRserverFromServerfarm](#)
- [activateServerfarmRserver](#)
- [suspendServerfarmRserver](#)
- [changeServerfarmRserverWeight](#)
- [activateAnswer](#)
- [suspendAnswer](#)
- [activateDnsRule](#)
- [suspendDnsRule](#)
- [getVMMappingInfo](#)

login

This request logs in an application user to ANM. The login API validates the credentials and returns a session token. The session times out after being idle for 30 minutes even if the logout is not called. By default, ANM allows 10 concurrent API logins, but you can modify this number (see the [“Modifying the Number of Concurrent Sessions Allowed”](#) section on page 4-3).


Input Parameters

Parameter	Description
Username	Username of the application user.
Password	Password of the application user.

Output Parameters

Parameter	Description
SessionToken	Session key of the application user.

Exceptions

Type	Exception String
String	Maximum session limit <i>configured_limit</i> reached for WS.  Note The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Invalid username or password. Please try again.
String	Session ID is in use. Please try again later.
String	Username cannot be null.
String	Password cannot be null.

Example Request

The following example shows an XML-formatted SOAP request to perform the [login](#) action. The input parameters for this particular action are username and password.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <login
      xmlns="http://anm.cisco.com">
      <user
        xmlns="">admin</user>
      <password xmlns="">admin</password>
    </login>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response:

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

```
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:loginResponse
      xmlns:ns2="http://anm.cisco.com">
      <SessionToken xmlns="">
        <sid>5546099606143734326</sid>
      </SessionToken>
    </ns2:loginResponse>
  </S:Body>
</S:Envelope>
```

logout

This request logs out an application user from the ANM. The session times out after being idle for 30 minutes even if the logout is not called. In case of a session timeout, the following exception string is generated: Invalid Session Id.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.

Example Request

The following example shows an XML-formatted SOAP request to perform the logout action. Session token is the input parameter for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <logout
      xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>-60553659690519291</sid>
      </sessionToken>
    </logout>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:logoutResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

listDeviceids

This request lists all device IDs for a device type from the ANM inventory. If none of the devices is imported in ANM, it returns an empty list.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceType	List of device types including the device name and status. For example, CSS, CSM, CAT6K, GSS, VIRTUAL_CONTEXT, ACE4710, and ACE_BLADE.

Output Parameters

Parameter	Description
DeviceIDs	List of device IDs including the device name, device type, IP address, slot, and chassis IP address attributes.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Unsupported Device Type: <i>device_type</i> .
String	Device Type should not be null.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listDeviceids](#) action. SessionToken and DeviceType are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listDeviceids
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-81548021995146358</sid>
      </sessionToken>
      <deviceType xmlns="">VIRTUAL_CONTEXT</deviceType>
    </listDeviceids>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains DeviceID as the output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listDeviceIdsResponse
      xmlns:ns2="http://anm.cisco.com">
      <DeviceIDs xmlns="">
```

```

<item>
  <name>Admin</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <ipAddr>10.77.241.46</ipAddr>
</item>
<item>
  <name>VC1</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <ipAddr>10.77.241.46</ipAddr>
</item>
<item>
  <name>VC2</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <ipAddr>10.77.241.46</ipAddr>
</item>
<item>
  <name>VC3</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <ipAddr>10.77.241.46</ipAddr>
</item>
<item>
  <name>VC4</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <ipAddr>10.77.241.46</ipAddr>
</item>
<item>
  <name>Admin</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <slot>3</slot>
  <chassisIPAddr>10.77.247.2</chassisIPAddr>
</item>
<item>
  <name>VC1</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <slot>3</slot>
  <chassisIPAddr>10.77.247.2</chassisIPAddr>
</item>
<item>
  <name>Admin</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <slot>4</slot>
  <chassisIPAddr>10.77.247.2</chassisIPAddr>
</item>
<item>
  <name>VC1</name>
  <deviceType>VIRTUAL_CONTEXT</deviceType>
  <slot>4</slot>
  <chassisIPAddr>10.77.247.2</chassisIPAddr>
</item>
</DeviceIDs>
</ns2:listDeviceIdsResponse>
</S:Body>
</S:Envelope>

```


listVirtualContexts

This request lists all the virtual contexts for an Application Control Engine (ACE) module or ACE appliance from the ANM inventory.

This API is supported only on ACE modules and appliances (device type: ACE_BLADE and ACE4710).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	ACE module: Device ID includes the device type, slot, and chassis IP address attributes. ACE appliance: Device ID includes the device type, IP address, and name attributes.

Output Parameters

Parameter	Description
DeviceIDs	List of virtual context device IDs including the device type, IP address, slot, and chassis IP address attributes.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPAddress</i> .
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as ACE_BLADE, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as ACE4710, but the mandatory field “ipAddr” is set as null or empty.
String	Invalid IP Address: <i>IPAddress</i> .
String	DeviceType is set as ACE_BLADE, but the mandatory field “slot” is set as null or empty.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listVirtualContexts](#) action. SessionToken and DeviceID are the input parameters for this particular action.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listVirtualContexts
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-81548021995146358</sid>
      </sessionToken>
      <deviceIDs xmlns="">
        <deviceType>ACE_BLADE</deviceType>
        <slot>3</slot>
        <chassisIPAddr>10.77.247.2</chassisIPAddr>
      </deviceIDs>
    </listVirtualContexts>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that contains the list of virtual context DeviceIDs as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listVirtualContextsResponse
      xmlns:ns2="http://anm.cisco.com">
      <DeviceIDs xmlns="">
        <item>
          <name>Admin</name>
          <deviceType>VIRTUAL_CONTEXT</deviceType>
          <slot>3</slot>
          <chassisIPAddr>10.77.247.2</chassisIPAddr>
        </item>
        <item>
          <name>VC1</name>
          <deviceType>VIRTUAL_CONTEXT</deviceType>
          <slot>3</slot>
          <chassisIPAddr>10.77.247.2</chassisIPAddr>
        </item>
      </DeviceIDs>
    </ns2:listVirtualContextsResponse>
  </S:Body>
</S:Envelope>

```

listVirtualServers

This request lists all the virtual servers configured on the specified device. The device can be a CSS, CSM, or a virtual context (VIRTUAL_CONTEXT) on an ACE module or ACE appliance.


Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context DeviceID, which contains the virtual server name, device type, IP address, chassis slot, and chassis IP address attributes.
allVSs	(Optional) Input parameter that instructs ANM to return a list of all virtual servers that match ANM's virtual server definition and those that do not match this definition but that ANM can recognize as virtual servers using SNMP polling. By default, ANM returns only the virtual servers that match ANM's virtual server definition.

Output Parameters

Parameter	Description
vservers	List of all virtual servers including the virtual server name, policy-map name, and IP address attributes.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Maximum session limit <i>configured_limit</i> reached for WS.
	 Note The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPAddress</i> .
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.

String	Invalid IP Address: <i>IPaddress</i> .
String	Given virtual context name “[<i>VC name</i>]” does not exist in the ACE Devices.
String	DeviceType is set as CSS, but the mandatory field “ipAddr” is set as null or empty.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “chassisIPAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “slot” is set as null or empty.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listVirtualServers](#) action. SessionToken, DeviceID, and allVSs are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listVirtualServers
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>7299761084673772498</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr/>
        <slot>2</slot>
        <chassisIPAddr>10.77.247.2</chassisIPAddr>
      </deviceID>
      <allVS xmlns="">false</allVS>
    </listVirtualServers>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains the list of virtual servers as the output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listVirtualServersResponse
      xmlns:ns2="http://anm.cisco.com">
      <listVirtualServers xmlns="">
        <item>
          <virtualserverName>14_simple_vs</virtualserverName>
          <policyMapName>int2</policyMapName>
          <matchConditions>
            <ipAddress>15.5.3.2</ipAddress>
            <protocol>tcp</protocol>
```

```
<port>2</port>
</matchConditions>
<serverfarms>14_simple_sf</serverfarms>
<vlans>2</vlans>
<adminState>DOWN</adminState>
<operState>DOWN</operState>
<lastPolledTime>Fri May 27 16:52:31 IST 2011</lastPolledTime>
</item>
<item>
<virtualserverName>ds_simple_vs</virtualserverName>
<policyMapName>int161</policyMapName>
<matchConditions>
<ipAddress>13.3.3.3</ipAddress>
<protocol>tcp</protocol>
<port>6</port>
</matchConditions>
<serverfarms>ds_simple_sf</serverfarms>
<vlans>161</vlans>
<adminState>DOWN</adminState>
<operState>DOWN</operState>
<lastPolledTime>Fri May 27 16:52:31 IST 2011</lastPolledTime>
</item>
<item>
<virtualserverName>test</virtualserverName>
<policyMapName>global</policyMapName>
<matchConditions>
<ipAddress>4.5.6.7</ipAddress>
<protocol>tcp</protocol>
<port>443</port>
</matchConditions>
<serverfarms>gh</serverfarms>
<vlans/>
<adminState>UP</adminState>
<operState>DOWN</operState>
<lastPolledTime>Fri May 27 16:52:31 IST 2011</lastPolledTime>
</item>
</listVirtualServers>
</ns2:listVirtualServersResponse>
</S:Body>
</S:Envelope>
```

listServerFarms

This request lists all server farms on a virtual context from the ANM inventory. This API is supported only on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.

Output Parameters

Parameter	Description
Serverfarms	List of server farm objects including the name and type (host or redirect) attributes.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Maximum session limit [<i>configured_limit</i>] reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPAddress</i>
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: <i>IPAddress</i> .
String	Given virtual context name <i>VC_name</i> does not exist in the ACE Devices.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listServerFarms](#) action. SessionToken and DeviceID are the input parameters for this particular action.

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

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listServerFarms
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>8368662778638735743</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
    </listServerFarms>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that contains the list of Serverfarm objects as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listServerFarmsResponse
      xmlns:ns2="http://anm.cisco.com">
      <serverfarms xmlns="">
        <item>
          <name>SF1</name>
          <serverFarmType>host</serverFarmType>
        </item>
        <item>
          <name>SF2</name>
          <serverFarmType>host</serverFarmType>
        </item>
      </serverfarms>
    </ns2:listServerFarmsResponse>
  </S:Body>
</S:Envelope>

```

listRservers

This request lists all the real servers on a virtual context from the ANM inventory. This API is supported only on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.

Output Parameters

Parameter	Description
Rservers	List of real server objects including the name, IP address, admin state, and weight attributes.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given ACE module is not yet imported in ANM.
String	Given ACE appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPAddress</i> .
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: <i>IPAddress</i> .
String	Given virtual context name <i>VC_name</i> does not exist in the ACE devices.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listRservers](#) action. SessionToken and DeviceID are the input parameters for this particular action.

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



```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listRservers
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>8368662778638735743</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
    </listRservers>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that contains the list of real server objects as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listRserversResponse
      xmlns:ns2="http://anm.cisco.com">
      <Rservers xmlns="">
        <item>
          <name>RS1</name>
          <type>host</type>
          <state>IS</state>
          <weight>8</weight>
        </item>
        <item>
          <name>RS2</name>
          <type>redirect</type>
          <state>IS</state>
        </item>
      </Rservers>
    </ns2:listRserversResponse>
  </S:Body>
</S:Envelope>

```

listServerfarmRservers

This request lists all the real servers present in the given server farm from the ANM inventory. If the server farm name is provided as empty or null in the input parameters, it lists all the real server objects in the given virtual context. This API is supported on CSS, CSM, and ACE devices (device type: CSS, CSM, and VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Device ID of the virtual context including the device name, device type, IP address, slot and chassis IP address attributes.
Serverfarmname	Name of an existing server farm. It can be empty, null, or a specific server farm configured in the device.

Output Parameters

Parameter	Description
SfRservers	List of server farm real server objects including server farm name, server farm real server name, IP address, admin state, weight, and port value attributes.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given ACE module is not yet imported in ANM.
String	Given ACE appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: <i>IPaddress</i> .
String	Given virtual context name <i>VC_name</i> does not exist in the ACE Devices.

String	DeviceType is set as CSS, but the mandatory field “ipAddr” is set as null or empty.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “chassisIPAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “slot” is set as null or empty.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listServerfarmRservers](#) action. SessionToken, DeviceID, and Serverfarmname are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listServerfarmRservers
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-2221649013806878130</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
      <serverfarmname xmlns="">SF1</serverfarmname>
    </listServerfarmRservers>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains Sfrserver as the output parameter:

```
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listServerfarmRserversResponse
      xmlns:ns2="http://anm.cisco.com">
      <Sfrservers xmlns="">
        <item>
          <serverfarmName>SF1</serverfarmName>
          <realserverName>04122009sanitytesting</realserverName>
          <adminState>IS</adminState>
          <operationState>UP</operationState>
          <weight>8</weight>
          <port>0</port>
          <lastPolledTime>Wed Jun 08 20:53:18 IST 2011</lastPolledTime>
        </item>
        <item>
          <serverfarmName>1</serverfarmName>
          <realserverName>anm-vm-118</realserverName>
          <adminState>IS</adminState>
        </item>
      </Sfrservers>
    </ns2:listServerfarmRserversResponse>
  </S:Body>
</S:Envelope>
```

```
<weight>19</weight>
<port>0</port>
</item>
<item>
<serverfarmName>SF1</serverfarmName>
<realserverName>test</realserverName>
<adminState>OOS</adminState>
<operationState>UP</operationState>
<weight>8</weight>
<port>443</port>
<ipAddr>10.2.1.11</ipAddr>
<lastPolledTime>Wed Jun 08 20:53:18 IST 2011</lastPolledTime>
</item>
</SfRservers>
</ns2:listServerfarmRserversResponse>
</S:Body>
</S:Envelope>
```

listAnswers

This request lists all the VIP answers configured on the specified given GSS.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, which contains the GSS name, device type, and IP address attributes.

Output Parameters

Parameter	Description
answers	List of answer objects, including the answer name, IP address, configuration state, and type attributes. The type attribute is always "vip."

Exceptions

Type	Exception String
String	Invalid session token.
String	GSS does not exist.
String	Session timeout.
String	Cannot reach the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listAnswers](#) action. SessionToken and DeviceID are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listAnswers
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>7299761084673772498</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>GSS</name>
        <deviceType>GSS</deviceType>
        <ipAddr>172.23.244.135</ipAddr>
        <slot>0</slot>
        <chassisIPAddr/>
      </deviceID>
    </listAnswers>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains answers as the output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listAnswersResponse
      xmlns:ns2="http://anm.cisco.com">
      <Answers xmlns="">
        <item>
          <ipAddr>10.10.9.99</ipAddr>
          <answerName>vip2131</answerName>
          <type>null</type>
          <configState>ACTIVATE</configState>
          <anserGroups>vipgrp1066</anserGroups>
          <location>null</location>
          <device>GSS</device>
          <operState>ACTIVE</operState>
          <pGssmTime>17:50:19</pGssmTime>
        </item>
      </DeviceIDs>
    </ns2:listAnswersResponse>
  </S:Body>
</S:Envelope>
```

listDnsRules

This request lists all the DNS rules configured on the specified given GSS.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, which contains the GSS name, device type, and IP address attributes.

Output Parameters

Parameter	Description
dnsRules	List of DNS rule objects, including the DNS rule name, source IP address, owner, and configuration state attributes.

Exceptions

Type	Exception String
String	Invalid session token.
String	GSS does not exist.
String	Session timeout.
String	Cannot reach the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listDnsRules](#) action. SessionToken and DeviceID are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listDnsRules
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>6600024266770332702</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>40</name>
        <deviceType>GSS</deviceType>
        <ipAddr>10.77.241.40</ipAddr>
      </deviceID>
    </listDnsRules>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains dnsRules as the output parameter:

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

```
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
  <ns2:listDnsRulesResponse
    xmlns:ns2="http://anm.cisco.com">
    <listDnsRules xmlns="">
      <item>
        <dnsRuleName>dns1</dnsRuleName>
        <sourceAddress>Anywhere</sourceAddress>
        <domains>dom1</domains>
        <dnsConfigState>Active</dnsConfigState>
        <answerGroups>null</answerGroups>
        <owner>System</owner>
      </item>
    </listDnsRules>
  </ns2:listDnsRulesResponse>
</S:Body>
</S:Envelope>
```


activateVirtualServer

This request activates an existing virtual server, putting it into service. This API is supported on CSS, CSM, and virtual contexts (VIRTUAL_CONTEXT) on ACE modules and appliances.


Note

For CSS and CSM devices, after changing the state, SNMP polling is triggered. The API does not return until the polling is complete.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual server device ID, including the device name, device type, IP address, slot, and chassis IP address attributes.
vServer	Virtual server object, including the virtual server name, policy-map name, and IP address attributes.
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Virtual context does not exist.
String	Session timeout.
String	Cannot reach the virtual context.
String	Virtual server does not exist.
String	Config errors from the ACE, CSS, or CSM.

Example Request

The following example shows an XML-formatted SOAP request to perform the [activateVirtualServer](#) action. SessionToken, DeviceID, and virtualServer (vip1) are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <activateVirtualServer
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>2713248117084982682</sid>
      </sessionToken>
      <deviceID
        xmlns="">
```

```

    <name>SysTestApp</name>
    <deviceType>VIRTUAL_CONTEXT</deviceType>
    <slot>2</slot>
    <chassisIPAddr>10.77.247.2</chassisIPAddr>
  </deviceID>
  <vserver
    xmlns="">
    <virtualserverName>12_simple_vs</virtualserverName>
    <policyMapName>NAT_Policy</policyMapName>
  </vserver>
  <reason xmlns="">testing</reason>
</activateVirtualServer>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that contains no output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:activateVirtualServerResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>

```

suspendVirtualServer

This request suspends an existing virtual server, putting it out of service. This API is supported on CSS, CSM, and virtual contexts (VIRTUAL_CONTEXT) on ACE modules and appliances.


Note

For CSS and CSM devices, after changing the state, SNMP polling is triggered. The API does not return until the polling is complete.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual server device ID, including the device name, device type, IP address, slot, and chassis IP address attributes.
vServer	Virtual server object, including the virtual server name, policy-map name, and IP address attributes.
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Virtual context does not exist.
String	Session timeout.
String	Cannot reach the virtual context.
String	Virtual server does not exist.
String	Config errors from the ACE, CSS, or CSM.

Example Request

The following example shows an XML-formatted SOAP request to perform the [suspendVirtualServer](#) action. SessionToken, DeviceID, and virtualServer (vip1) are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <suspendVirtualServer
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>2713248117084982688</sid>
      </sessionToken>
      <deviceID
        xmlns="">
```

```

    <name>SysTestApp</name>
    <deviceType>VIRTUAL_CONTEXT</deviceType>
    <ipAddr/>
    <slot>2</slot>
    <chassisIPAddr>10.77.247.2</chassisIPAddr>
  </deviceID>
  <vserver
    xmlns="">
    <virtualserverName>12_simple_vs</virtualserverName>
    <policyMapName>NAT_Policy</policyMapName>
  </vserver>
  <reason xmlns="">test</reason>
</suspendVirtualServer>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that contains no output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:suspendVirtualServerResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>

```

addRserverToServerfarm

This request adds a real server to an existing server farm. If it does not find the real server, it creates a real server and adds it to the server farm. This API is supported on ACE devices (device type: 'VIRTUAL_CONTEXT').

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Device ID of the virtual context including the device name, device type, IP address, slot and chassis IP address attributes.
Rserver	Real server object including name, IP address, admin state, and weight attributes.
Port	Port of the server farm real server.
Serverfarm	Name of an existing server farm.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Given ACE module is not yet imported in ANM.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field "slot" is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field "name" is set as null or empty.
String	Invalid IP Address: <i>IPaddress</i> .
String	Given virtual context name <i>VC_name</i> does not exist in the ACE Devices.
String	Given ACE Appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Device Type should be not null.
String	"chassisIPAddr" and "ipAddr" are not added in device ID. Either should be provided.
String	Both "chassisIPAddr" and "ipAddr" are added in device ID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field "ipAddr" is set as null or empty.

String	Given Server Farm “ServerFarmname” is not available in the given Virtual Context.
String	Server Farm name should not be null or empty.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “ Modifying the Number of Concurrent Sessions Allowed ” section on page 4-3).
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 64 characters.
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Invalid weight value: <i>weight_value</i> . Weight value should be numeric.
String	Invalid weight value: <i>weight_value</i> . Weight value is outside the range <0-100>.
String	Invalid Name: <i>ServerFarm_name</i> . Name should match the format [^a-zA-Z0-9._-].
String	Invalid Name: <i>RealServer_name</i> . Name should match the format [^a-zA-Z0-9._-].
String	Invalid Name: <i>ServerFarm_name</i> . Name should not hold any reserved keywords.
String	Invalid Name: <i>RealServer_name</i> . Name should not hold any reserved keywords.
String	IP address must be provided for the Real Server.
String	Given Real Server type “Real Server Type” is not associated with Server Farm type “Server Farm type.”
String	Real Server “Real Server Name/ Port value” already exist in the Server Farm <i>ServerFarm_name</i> .

Example Request

The following example shows an XML-formatted SOAP request to perform the [addRserverToServerfarm](#) action. SessionToken, DeviceID, Rserver, Port, and Serverfarm are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <addRserverToServerfarm
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>5731694438459731003</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
      <serverfarmname
```

```
xmlns="">SF1</serverfarmname>
<rserver
  xmlns="">
  <name>sampletest</name>
  <type>host</type>
  <ipAddr>2.3.6.53</ipAddr>
  <state>IS</state>
  <weight>34</weight>
</rserver>
<port xmlns="">0</port>
</addRserverToServerfarm>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:addRserverToServerfarmResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

removeRserverFromServerfarm

This request removes an existing server farm real server from a server farm. It removes the real server, if there is no server farm real server reference to the real server. This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.
SfRserver	Server farm real server object including the server farm name, real server name, IP address, admin state, weight, and port attributes.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Given ACE module does not exist in the chassis: <i>Chassis_IPAddress</i> .
String	Device Type should be not null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be not null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: <i>IPAddress</i> .
String	Given virtual context name <i>VC_name</i> does not exist in the ACE Devices.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.

String	Given Server Farm <i>ServerFarm_name</i> is not available in the given Virtual Context.
String	Server Farm name should not be null or empty.
String	Given Name : <i>ServerFarm_name</i> exceeds the maximum length of 64 characters.
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Invalid weight value: <i>weight_value</i> . Weight value should be numeric.
String	Invalid weight value: <i>weight_value</i> . Weight value is outside the range <0-100>.
String	Invalid Name: <i>ServerFarm_name</i> . Name should match the format [^a-zA-Z0-9._-].
String	Invalid Name: <i>RealServer_name</i> . Name should match the format [^a-zA-Z0-9._-].
String	Invalid Name: <i>ServerFarm_name</i> . Name should not hold any reserved keywords.
String	Invalid Name: <i>RealServer_name</i> . Name should not hold any reserved keywords.
String	Given Server Farm <i>ServerFarm_name</i> is not available in the given Virtual Context.
String	Given Real Server <i>RealServer_name</i> / <i>port_number</i> does not exist in the Server Farm <i>ServerFarm_name</i> .

Example Request

The following example shows an XML-formatted SOAP request to perform the [removeRserverFromServerfarm](#) action. SessionToken, DeviceID, and Sfrserver are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <removeRserverFromServerfarm
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-81548021995146358</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
      <rserver xmlns="">
        <serverfarmName>SF1</serverfarmName>
        <realserverName>sampletest</realserverName>
        <adminState>IS</adminState>
        <weight>0</weight>
        <port>0</port>
        <ipAddr>2.3.6.53</ipAddr>
      </rserver>
    </removeRserverFromServerfarm>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
    </rserver>
  </removeRserverFromServerfarm>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:removeRserverFromServerfarmResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

activateServerfarmRserver

This request activates an existing server farm real server in service. This API is supported on CSS, CSM, and ACE devices (device type: CSS, CSM and VIRTUAL_CONTEXT).


Note

For CSS and CSM devices, after changing the state, SNMP polling is triggered. The API does not return until the polling is complete.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.
SfRserver	Server farm real server object including the server farm name, real server name, IP address, admin state, weight and port attributes
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Server Farm name cannot be null.
String	Maximum session limit [<i>configured_limit</i>] reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given Server farm Real Server name cannot be null.
String	Given Real Server state cannot be null.
String	Given Real server weight cannot be null.
String	Server Farm Real server port value should not be null.
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	Given Cat6k does not exist in ANM.
String	Given ACE module does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSM does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSS does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Device Type should be null.

String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: “IP address.”
String	DeviceType is set as CSS, but the mandatory field “ipAddr” is set as null or empty.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “chassisIPAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “slot” is set as null or empty.
String	Given Server Farm <i>ServerFarm_name</i> is not available in the given Virtual Context.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 64 characters.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 15 characters.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 31 characters.
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Given Real Server <i>RealServer_name/port_number</i> does not exist in the Server Farm <i>ServerFarm_name</i> .

Example Request

The following example shows an XML-formatted SOAP request to perform the [activateServerfarmRserver](#) action. SessionToken, DeviceID, Sfrserver, and Reason are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <activateServerfarmRserver
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-5217915770163489503</sid>
      </sessionToken>
      <deviceID
        xmlns="">
```

```
<name>Admin</name>
<deviceType>VIRTUAL_CONTEXT</deviceType>
<ipAddr>10.77.241.46</ipAddr>
</deviceID>
<rserver
  xmlns="">
  <serverfarmName>SF1</serverfarmName>
  <realserverName>test</realserverName>
  <adminState>OOS</adminState>
  <weight>44</weight>
  <port>443</port>
  <ipAddr>10.2.1.11</ipAddr>
</rserver>
  <reason xmlns="">sample Webservices testing</reason>
</activateServerfarmRserver>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:activateServerfarmRserverResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

suspendServerfarmRserver

This request suspends the existing server farm real server. This API is supported on CSS, CSM, and ACE devices (device type: CSS, CSM and VIRTUAL_CONTEXT).



Note

For CSS and CSM devices, after changing the state, SNMP polling is triggered. The API does not return until the polling is complete.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.
SfRserver	Server farm real server object including the server farm name, real server name, IP address, admin state, weight and port attributes
State	List of options supported are Graceful, Suspend, Suspend_Clear_Connections.
Reason	Reason for suspension.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Server Farm name cannot be null.
String	Given Server farm Real Server name cannot be null.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given Real Server state cannot be null.
String	Given Real server weight cannot be null.
String	Server Farm Real server port value should not be null.
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	Given Cat6k does not exist in ANM.
String	Given ACE module does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSM does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSS does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .

String	Device Type should be null.
String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: <i>IPaddress</i> .
String	DeviceType is set as CSS, but the mandatory field “ipAddr” is set as null or empty.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “chassisIPAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “slot” is set as null or empty.
String	Given Server Farm <i>ServerFarm_name</i> is not available in the given Virtual Context.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 64 characters.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 15 characters.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 31 characters.
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Given Real Server <i>RealServer_name/port_number</i> does not exist in the Server Farm <i>ServerFarm_name</i> .
String	Invalid suspend state: <i>suspend_state</i> .

Example Request

The following example shows an XML-formatted SOAP request to perform the [suspendServerfarmRserver](#) action. SessionToken, DeviceID, SfrServer, Reason, and State are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <suspendServerfarmRserver
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
```

```

    <sid>2658073681808034239</sid>
  </sessionToken>
  <deviceID
    xmlns="">
    <name>Admin</name>
    <deviceType>VIRTUAL_CONTEXT</deviceType>
    <ipAddr>10.77.241.46</ipAddr>
  </deviceID>
  <rserver
    xmlns="">
    <serverfarmName>SF1</serverfarmName>
    <realserverName>test</realserverName>
    <adminState>IS</adminState>
    <weight>44</weight>
    <port>443</port>
    <ipAddr>10.2.1.11</ipAddr>
  </rserver>
  <suspendState
    xmlns="">Suspend</suspendState>
    <reason xmlns="" />
  </suspendServerfarmRserver>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response that has no output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:suspendServerfarmRserverResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>

```


changeServerfarmRserverWeight

This request changes the weight of an existing server farm real server. This API is supported on CSS, CSM, and ACE devices (device type: CSS, CSM, and VIRTUAL_CONTEXT).


Note

For CSS & CSM devices, after changing the weight, SNMP polling is triggered. The API does not return until the polling is complete.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	Virtual context device ID including the device name, device type, IP address, slot, and chassis IP address attributes.
SfRserver	Server farm real server object including the server farm name, real server name, IP address, admin state, weight and port attributes.
Reason	Reason for the operation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid Session ID.
String	Server Farm name cannot be null.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “Modifying the Number of Concurrent Sessions Allowed” section on page 4-3).
String	Given Server farm Real Server name cannot be null.
String	Given Real Server state cannot be null.
String	Given Real server weight cannot be null.
String	Server Farm Real server port value should not be null.
String	Given ACE module is not yet imported in ANM.
String	Given ACE Appliance does not exist in ANM.
String	Given Cat6k does not exist in ANM.
String	Given ACE module does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSM does not exist in the chassis: <i>Chassis_IPaddress</i> .
String	Given CSS does not exist in ANM.
String	None of the devices are imported.
String	Unsupported Device Type: <i>device_type</i> .
String	Device Type should be null.

String	“chassisIPAddr” and “ipAddr” are not added in deviceID. Either should be provided.
String	Both “chassisIPAddr” and “ipAddr” are added in deviceID. Only one of them can be added.
String	DeviceID should not be null.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “slot” is set as null or empty.
String	DeviceType is set as VIRTUALCONTEXT, but the mandatory field “name” is set as null or empty.
String	Invalid IP Address: “Ip address.”
String	DeviceType is set as CSS, but the mandatory field “ipAddr” is set as null or empty.
String	DeviceType is set as VIRTUAL_CONTEXT, but the mandatory field “ipAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “chassisIPAddr” is set as null or empty.
String	Device Type is set as CSM, but the mandatory field “slot” is set as null or empty.
String	Given Server Farm <i>ServerFarm_name</i> is not available in the given Virtual Context.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 64 characters.
String	Given Name: <i>ServerFarm_name</i> exceeds the maximum length of 15 characters
String	Given Name : <i>ServerFarm_name</i> exceeds the maximum length of 31 characters.
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Given Real Server <i>RealServer_name/port_number</i> does not exist in the Server Farm <i>ServerFarm_name</i> .
String	Invalid Port: <i>port_number</i> . Port number should be numeric.
String	Invalid port value: <i>port_number</i> . Port value is outside the range <0-65535>.
String	Given Real Server <i>RealServer_name/port_number</i> does not exist in the Server Farm <i>ServerFarm_name</i> .
String	Invalid weight value: <i>weight_value</i> . Weight value is outside the range <1-100> [ace module, ace appliance].
String	Invalid weight value: <i>weight_value</i> . Weight value is outside the range <0-100> [csm].
String	Invalid weight value: <i>weight_value</i> . Weight value is outside the range <0-10> [css].

Example Request

The following example shows an XML-formatted SOAP request to perform the [changeServerfarmRserverWeight](#) action. SessionToken, DeviceID, Sfrserver, and Reason are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <changeServerfarmRserverWeight
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>-5217915770163489503</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.241.46</ipAddr>
      </deviceID>
      <rserver
        xmlns="">
        <serverfarmName>SF1</serverfarmName>
        <realserverName>test</realserverName>
        <adminState>OOS</adminState>
        <weight>44</weight>
        <port>443</port>
        <ipAddr>10.2.1.11</ipAddr>
      </rserver>
      <reason xmlns="" />
    </changeServerfarmRserverWeight>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:changeServerfarmRserverWeightResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

activateAnswer

This request activates an existing answer on the specified GSS, putting the answer into service.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, including the device name, device type, and IP address attributes.
answer	Answer object, including the answer name, IP address, and type attributes. The type is always "vip."
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Session timeout.
String	Cannot reach the GSS.
String	GSS does not exist.
String	Config errors from the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [activateAnswer](#) action. SessionToken, DeviceID, and answer are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <activateAnswer
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>991427603955168471</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>172.23.244.135</name>
        <deviceType>GSS</deviceType>
        <ipAddr>172.23.244.135</ipAddr>
      </deviceID>
      <answer
        xmlns="">
        <ipAddr>2.3.4.5</ipAddr>
        <answerName>vip1</answerName>
      </answer>
    </activateAnswer>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<reason xmlns="">test</reason>
</activateAnswer>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:activateAnswerResponse xmlns:ns2="http://anm.cisco.com" />
  </S:Body>
</S:Envelope>
```

suspendAnswer

This request suspends an existing answer on the specified GSS, putting the answer out of service.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, including the device name, device type, and IP address attributes.
answer	Answer object, including the answer name, IP address, and type attributes. The type is always "vip."
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Session timeout.
String	Cannot reach the GSS.
String	GSS does not exist.
String	Config errors from the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [suspendAnswer](#) action. SessionToken, DeviceID, and answer are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <suspendAnswer
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>991427603955168481</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>172.23.244.135</name>
        <deviceType>GSS</deviceType>
        <ipAddr>172.23.244.135</ipAddr>
      </deviceID>
      <answer
        xmlns="">
        <ipAddr>2.3.4.5</ipAddr>
        <answerName>vip1</answerName>
      </answer>
    </suspendAnswer>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<reason xmlns="">test</reason>
</suspendAnswer>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:suspendAnswerResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

activateDnsRule

This request activates an existing DNS rule on the specified GSS, putting the DNS rule into service.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, including the device name, device type, and IP address attributes.
dnsRule	DNS rule object, including the DNS rule name, source IP address, owner, and configuration state attributes.
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Session timeout.
String	Cannot reach the GSS.
String	GSS does not exist.
String	Config errors from the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [activateDnsRule](#) action. SessionToken, DeviceID, and dnsRule are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <activateDnsRule
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>6195441786283682375</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>172.23.244.135</name>
        <deviceType>GSS</deviceType>
        <ipAddr>172.23.244.135</ipAddr>
      </deviceID>
      <dnsrule
        xmlns="">
        <dnsRuleName>dns1</dnsRuleName>
        <sourceAddress>Anywhere</sourceAddress>
      </dnsrule>
    </activateDnsRule>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
<reason xmlns="">test</reason>
</activateDnsRule>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:activateDnsRuleResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

suspendDnsRule

This request suspend an existing DNS rule on the specified GSS, putting the DNS rule out of service.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
DeviceID	GSS device ID, including the device name, device type, and IP address attributes.
dnsRule	DNS rule object, including the DNS rule name, source IP address, owner, and configuration state attributes.
Reason	Reason for the activation.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session token.
String	Session timeout.
String	Cannot reach the GSS.
String	GSS does not exist.
String	Config errors from the GSS.

Example Request

The following example shows an XML-formatted SOAP request to perform the [suspendDnsRule](#) action. SessionToken, DeviceID, and dnsRule are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <suspendDnsRule
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>6195441786283682370</sid>
      </sessionToken>
      <deviceID
        xmlns="">
        <name>172.23.244.135</name>
        <deviceType>GSS</deviceType>
        <ipAddr>172.23.244.135</ipAddr>
      </deviceID>
      <dnsrule
        xmlns="">
        <dnsRuleName>dns1</dnsRuleName>
        <sourceAddress>Anywhere</sourceAddress>
      </dnsrule>
    </suspendDnsRule>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<reason xmlns="">test</reason>
</suspendDnsRule>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that has no output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:suspendDnsRuleResponse xmlns:ns2="http://anm.cisco.com" />
  </S:Body>
</S:Envelope>
```

getVMMappingInfo

This request returns all the server farm real server mapping information for a given VM. This API is supported on CSS, CSM, and ACE devices (device type: CSS, CSM, and VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
VMName	Unique name of the VM. The format is “/Vcenter Name given in ANM>/full path for vmname as in the Vcenter>/VM name>”.

Output Parameters

Parameter	Description
VMMappingInfos	Object attributes including the device ID, server farm name, server farm real server name, IP address, and port attributes.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Maximum session limit <i>configured_limit</i> reached for WS. The default is 10 (see the “ Modifying the Number of Concurrent Sessions Allowed ” section on page 4-3).
String	None of the vCenter is imported in ANM.
String	Given VM <i>VM_name</i> does not exist.
String	Unable to get DeviceId for <i>device_id</i> .
String	Vmname should not be null.

Example Request

The following example shows an XML-formatted SOAP request to perform the [getVMMappingInfo](#) action. SessionToken and VMName are the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <getVMMappingInfo
      xmlns="http://anm.cisco.com">
      <sessionToken
        xmlns="">
        <sid>3920090311151112722</sid>
      </sessionToken>
      <vmname xmlns="">/Vcenter/ANM_DC/vm/Discovered virtual machine/anm-vm-115</vmname>
    </getVMMappingInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains the list of VMMappingInfo objects as the output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getVMMappingInfoResponse
      xmlns:ns2="http://anm.cisco.com">
      <VMMappingInfos xmlns="">
        <item>
          <deviceID>
            <deviceType>CSS</deviceType>
            <ipAddr>172.23.244.250</ipAddr>
          </deviceID>
          <sfname>myReal8</sfname>
          <sfrsname>myReal8</sfrsname>
          <port>0</port>
          <ipaddr>0.0.0.0</ipaddr>
        </item>
        <item>
          <deviceID>
            <name>Admin</name>
            <deviceType>VIRTUAL_CONTEXT</deviceType>
            <slot>4</slot>
            <chassisIPAddr>10.77.247.2</chassisIPAddr>
          </deviceID>
          <sfname>WEB-SF</sfname>
          <sfrsname>webservicesample</sfrsname>
          <port>0</port>
          <ipaddr>3.3.3.4</ipaddr>
        </item>
        <item>
          <deviceID>
            <deviceType>CSM</deviceType>
            <slot>4</slot>
            <chassisIPAddr>172.23.244.130</chassisIPAddr>
          </deviceID>
          <sfname>F002</sfname>
          <sfrsname>RS1</sfrsname>
          <port>0</port>
          <ipaddr/>
        </item>
        <item>
          <deviceID>
            <name>Admin</name>
            <deviceType>VIRTUAL_CONTEXT</deviceType>
            <ipAddr>10.77.241.46</ipAddr>
          </deviceID>
          <sfname>tst_sf_dup</sfname>
          <sfrsname>testing_dup</sfrsname>
          <port>0</port>
        </item>
      </VMMappingInfos>
    </ns2:getVMMappingInfoResponse>
  </S:Body>
</S:Envelope>
```

ApplicationTemplateManager Web Service APIs

This section describes the APIs associated with the ApplicationTemplateManager Web Service, which are as follows:

- [listTemplateDefinitions](#)
- [getTemplateDefinitionMetadata](#)
- [createTemplateInstance](#)
- [listTemplateInstances](#)
- [deleteTemplateInstance](#)
- [listCheckpoints](#)
- [createCheckpoint](#)
- [deleteCheckpoint](#)
- [rollbackCheckpoint](#)
- [freeFormShow](#)

listTemplateDefinitions

This API lists all the available ANM templates, including both ANM-defined templates (or *system templates*) and user-defined templates. Template definitions include the template name, description, application, application version, and so forth.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.

Output Parameters

Parameter	Description
templateDefinitionArray	List of template definitions.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listTemplateDefinitions](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header />
  <S:Body>
    <ns2:listTemplateDefinitions xmlns:ns2="http://anm.cisco.com">
      <sessionToken>
        <sid>-4754037077725100881</sid>
      </sessionToken>
    </ns2:listTemplateDefinitions>
  </S:Body>
</S:Envelope>
```

```

    </ns2:listTemplateDefinitions>
  </S:Body>
</S:Envelope>

```

Example Response

The following example shows the XML response that contains the list of templateDefinitions objects as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listTemplateDefinitionsResponse xmlns:ns2="http://anm.cisco.com">
      <templateDefinitions>
        <item>
          <id>1</id>
          <name>FTP</name>
          <description>FTP LB</description>
          <application>FTP</application>
          <appVersion>1.1</appVersion>
          <createTime>2012-02-08 15:06:51</createTime>
          <createUser>Management Task</createUser>
          <updateTime>2012-02-08 15:06:51</updateTime>
          <updateUser>Management Task</updateUser>
        </item>
        <item>
          <id>10</id>
          <name>Java Application Server</name>
          <description>HTTP server, Java Servlet and JSP technologies, ie Apache
Tomcat</description>
          <application>Java Application Server</application>
          <appVersion>1.1</appVersion>
          <createTime>2012-02-08 15:06:51</createTime>
          <createUser>Management Task</createUser>
          <updateTime>2012-02-08 15:06:51</updateTime>
          <updateUser>Management Task</updateUser>
        </item>
        <item>
          <id>11</id>
          <name>Basic HTTP</name>
          <description>Basic L4 LB for HTTP</description>
          <application>Basic HTTP</application>
          <appVersion>1.1</appVersion>
          <createTime>2012-02-08 15:06:51</createTime>
          <createUser>Management Task</createUser>
          <updateTime>2012-02-08 15:06:51</updateTime>
          <updateUser>Management Task</updateUser>
        </item>
        <item>
          <id>2</id>
          <name>Layer 3 LB</name>
          <description>Simple L3 all ports no mapping</description>
          <application>Layer 3 LB</application>
          <appVersion>1.1</appVersion>
          <createTime>2012-02-08 15:06:51</createTime>
          <createUser>Management Task</createUser>
          <updateTime>2012-02-08 15:06:51</updateTime>
          <updateUser>Management Task</updateUser>
        </item>
        <item>
          <id>3</id>
          <name>Secure Webserver</name>
          <description>Generic Secure Webserver with redirect</description>
          <application>Secure Webserver</application>

```

```

        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>4</id>
        <name>Microsoft Exchange 2010</name>
        <description>MS Exchange Template</description>
        <application>Microsoft Exchange 2010</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>5</id>
        <name>DWS with Nexus 7000 OTV</name>
        <description>Dynamic Workload Scaling with Nexus 7000 OTV for Basic LB
with HTTP or Secure Webserver</description>
        <application>DWS with Nexus 7000 OTV</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>6</id>
        <name>RDP</name>
        <description>Remote Desktop Protocol</description>
        <application>RDP</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>7</id>
        <name>Microsoft SharePoint 2010</name>
        <description>Microsoft SharePoint Template</description>
        <application>Microsoft SharePoint 2010</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>8</id>
        <name>Layer 4 LB</name>
        <description>Simple L4 LB</description>
        <application>Layer 4 LB</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
    </item>
    <item>
        <id>9</id>

```



```
<name>DNS</name>
<description>DNS LB</description>
<application>DNS</application>
<appVersion>1.1</appVersion>
<createTime>2012-02-08 15:06:51</createTime>
<createUser>Management Task</createUser>
<updateTime>2012-02-08 15:06:51</updateTime>
<updateUser>Management Task</updateUser>
</item>
</templateDefinitions>
</ns2:listTemplateDefinitionsResponse>
</S:Body>
</S:Envelope>
```

getTemplateDefinitionMetadata

This API takes an available ANM template definition and returns an array of template input objects, which contain the template name, default value, userData, hasChild, and the childElements details. The names are the variable names defined in the template definition file.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
TemplateDefinition	Template definition, which includes the template name, description, application, application version, and so forth. To retrieve a list of available template definitions, use the listTemplateDefinitions API.

Output Parameters

Parameter	Description
templateInputArray	Array of TemplateInput objects.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid template definition.

Example Request

The following example shows an XML-formatted SOAP request to perform the [getTemplateDefinitionMetadata](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header />
  <S:Body>
    <ns2:getTemplateDefinitionMetadata xmlns:ns2="http://anm.cisco.com">
      <sessionToken>
        <sid>-986099040932183901</sid>
      </sessionToken>
      <templateDefinition>
        <id>7</id>
        <name>Microsoft SharePoint 2010</name>
        <description>Microsoft SharePoint Template</description>
        <application>Microsoft SharePoint 2010</application>
        <appVersion>1.1</appVersion>
        <createTime>2012-02-08 15:06:51</createTime>
        <createUser>Management Task</createUser>
        <updateTime>2012-02-08 15:06:51</updateTime>
        <updateUser>Management Task</updateUser>
      </templateDefinition>
    </ns2:getTemplateDefinitionMetadata>
  </S:Body>
</S:Envelope>
```

Example Response

The following example shows the XML response that contains the list of TemplateDefinitionMetadata objects as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getTemplateDefinitionMetadataResponse xmlns:ns2="http://anm.cisco.com">
      <templateDefinitionMetadata>
        <item>
          <tagName>GROUP</tagName>
          <name>service</name>
          <type>group</type>
          <displayString>Application Configuration</displayString>
          <childElements>
            <tagName>VARIABLE</tagName>
            <name>name</name>
            <type>name</type>
            <displayString>Application Config Name:</displayString>
          </childElements>
          <childElements>
            <tagName>VARIABLE</tagName>
            <name>vip</name>
            <type>ipaddresswithprefix</type>
            <displayString>VIP Address:</displayString>
          </childElements>
          <childElements>
            <tagName>ARRAY</tagName>
            <name>reals</name>
            <displayString>SharePoint Application Servers:</displayString>
            <childElements>
              <tagName>VARIABLE</tagName>
              <name>ipAddr</name>
              <type>ipaddress</type>
              <displayString />
            </childElements>
          </childElements>
          <childElements>
            <tagName>VARIABLE</tagName>
            <name>realport</name>
            <type>port</type>
            <displayString>SSL Offload Port:</displayString>
            <defaultValue>80</defaultValue>
          </childElements>
          <childElements>
            <tagName>IMG</tagName>
            <name>img1</name>
            <type>img</type>
          </childElements>
          <childElements>
            <tagName>VARIABLE</tagName>
            <name>pkcs12</name>
            <type>enum</type>
            <displayString>Key Type:</displayString>
            <defaultValue>PKCS12</defaultValue>
          </childElements>
          <childElements>
            <tagName>IF</tagName>
            <childElements>
              <tagName>VARIABLE</tagName>
              <name>url</name>
              <type>url</type>
              <displayString>SSL Key URL:</displayString>
              <defaultValue>sftp://1.2.3.4/dir/myfile.pkcs12</defaultValue>
            </childElements>
            <childElements>
              <tagName>VARIABLE</tagName>
              <name>url.name</name>
            </childElements>
          </childElements>
        </item>
      </templateDefinitionMetadata>
    </ns2:getTemplateDefinitionMetadataResponse>
  </S:Body>
</S:Envelope>

```

```

        <type>Hidden</type>
        <displayString>SSL Key URL:</displayString>
        <defaultValue>sftp://1.2.3.4/dir/myfile.pkcs12</defaultValue>
    </childElements>
</childElements>
<childElements>
    <tagName>VARIABLE</tagName>
    <name>username</name>
    <type>string</type>
    <displayString>Key Server Username:</displayString>
</childElements>
<childElements>
    <tagName>VARIABLE</tagName>
    <name>password</name>
    <type>password</type>
    <displayString>Key Server Password:</displayString>
</childElements>
</childElements>
<childElements>
    <tagName>IF</tagName>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_key_url</name>
        <type>url</type>
        <displayString>SSL Key URL:</displayString>
        <defaultValue>sftp://1.2.3.4/dir/myfile.key</defaultValue>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_key_url.name</name>
        <type>Hidden</type>
        <displayString>SSL Key URL:</displayString>
        <defaultValue>sftp://1.2.3.4/dir/myfile.key</defaultValue>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_cer_url</name>
        <type>url</type>
        <displayString>SSL Cert URL:</displayString>
        <defaultValue>sftp://1.2.3.4/dir/myfile.der</defaultValue>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_cer_url.name</name>
        <type>Hidden</type>
        <displayString>SSL Cert URL:</displayString>
        <defaultValue>sftp://1.2.3.4/dir/myfile.der</defaultValue>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_username</name>
        <type>string</type>
        <displayString>Key Server Username:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>der_password</name>
        <type>password</type>
        <displayString>Key Server Password:</displayString>
    </childElements>
</childElements>
<childElements>
    <tagName>IF</tagName>
    <childElements>
        <tagName>VARIABLE</tagName>

```

```

        <name>key</name>
        <type>ssl_key</type>
        <displayString>SSL Key:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>key.name</name>
        <type>Hidden</type>
        <displayString>SSL Key:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>key.contents</name>
        <type>Hidden</type>
        <displayString>SSL Key:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>cert</name>
        <type>certificate</type>
        <displayString>SSL Certificate:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>cert.name</name>
        <type>Hidden</type>
        <displayString>SSL Certificate:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>cert.contents</name>
        <type>Hidden</type>
        <displayString>SSL Certificate:</displayString>
    </childElements>
</childElements>
<childElements>
    <tagName>VARIABLE</tagName>
    <name>passphrase</name>
    <type>password</type>
    <displayString>Cert/Key Passphrase:</displayString>
</childElements>
<childElements>
    <tagName>VARIABLE</tagName>
    <name>compress</name>
    <type>boolean</type>
    <displayString>Enable Compression:</displayString>
    <defaultValue>true</defaultValue>
</childElements>
</item>
<item>
    <tagName>GROUP</tagName>
    <name>network</name>
    <type>group</type>
    <displayString>Network Configuration</displayString>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>device</name>
        <type>device</type>
        <displayString>Load Balancer
(Device:VirtualContext):</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>vlans</name>

```

```
        <type>vlans</type>
        <displayString>Client VLANs:</displayString>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>autoNat</name>
        <type>boolean</type>
        <displayString>Enable Source NAT:</displayString>
    </childElements>
</item>
</templateDefinitionMetadata>
</ns2:getTemplateDefinitionMetadataResponse>
</S:Body>
</S:Envelope>
```

createTemplateInstance

This API takes a `TemplateDefinition` and `TemplateInputArray` object, generates a configuration based on the `TemplateDefinition`, and then deploys the configuration to the device. The `TemplateInput` object contains the template name and an array of name and value pairs. The names are the variable names defined in the template definition file and the values are all empty.

This API is supported on ACE devices (device type: `VIRTUAL_CONTEXT`).

Input Parameters

Parameter	Description
sessionToken	Session key of the application user.
deviceID	Device name, deviceType, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
templateDefinition	Template definition, which includes the template name, description, application, application version, and so forth. To retrieve a list of available template definitions, use the listTemplateDefinitions API.
templateInputArray	Array of <code>TemplateInput</code> objects, which includes the template name, default value, userData, hasChild, and the childElements details. The names are the variable names defined in the template definition file and the values are the values set to the variable names.

Output Parameters

Parameter	Description
instanceId	Unique integer value of the created template.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid template definition.
String	Invalid template input.

Example Request

The following example shows an XML-formatted SOAP request to perform the [createTemplateInstance](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header />
  <S:Body>
    <ns2:createTemplateInstance xmlns:ns2="http://anm.cisco.com">
      <sessionToken>
        <sid>-8257363202654966035</sid>
      </sessionToken>
      <deviceID>
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <slot>3</slot>
        <chassisIPAddr>10.77.247.2</chassisIPAddr>
      </deviceID>
      <templateDefinition>
```

```

<id>7</id>
<name>Microsoft SharePoint 2010</name>
<description>Microsoft SharePoint Template</description>
<application>Microsoft SharePoint 2010</application>
<appVersion>1.1</appVersion>
<createTime>2012-02-08 15:06:51</createTime>
<createUser>Management Task</createUser>
<updateTime>2012-02-08 15:06:51</updateTime>
<updateUser>Management Task</updateUser>
</templateDefinition>
<templateInputs>
  <item>
    <tagName>GROUP</tagName>
    <name>service</name>
    <type>group</type>
    <displayString>Application Configuration</displayString>
    <childElements>
      <tagName>VARIABLE</tagName>
      <name>name</name>
      <type>name</type>
      <displayString>Application Config Name:</displayString>
      <userData>sp_vs_01</userData>
    </childElements>
    <childElements>
      <tagName>VARIABLE</tagName>
      <name>vip</name>
      <type>ipaddresswithprefix</type>
      <displayString>VIP Address:</displayString>
      <userData>210.2.3.2</userData>
    </childElements>
    <childElements>
      <tagName>ARRAY</tagName>
      <name>reals</name>
      <displayString>SharePoint Application Servers:</displayString>
      <childElements>
        <tagName>VARIABLE</tagName>
        <name>ipAddr</name>
        <type>ipaddress</type>
        <displayString />
        <userDatas>210.2.3.4</userDatas>
        <userDatas>210.2.3.5</userDatas>
        <userDatas>210.2.3.6</userDatas>
        <userDatas>210.2.3.7</userDatas>
      </childElements>
    </childElements>
    <childElements>
      <tagName>VARIABLE</tagName>
      <name>realport</name>
      <type>port</type>
      <displayString>SSL Offload Port:</displayString>
      <defaultValue>80</defaultValue>
      <userData>810</userData>
    </childElements>
    <childElements>
      <tagName>IMG</tagName>
      <name>img1</name>
      <type>img</type>
    </childElements>
    <childElements>
      <tagName>VARIABLE</tagName>
      <name>pkcs12</name>
      <type>enum</type>
      <displayString>Key Type:</displayString>
      <defaultValue>PKCS12</defaultValue>
    </childElements>
  </item>
</templateInputs>

```



```

    <userData>PKCS12</userData>
  </childElements>
<childElements>
  <tagName>IF</tagName>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>url</name>
    <type>url</type>
    <displayString>SSL Key URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.pkcs12</defaultValue>
    <userData>sftp://10.77.240.100/p12_rsa768.p12</userData>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>url.name</name>
    <type>Hidden</type>
    <displayString>SSL Key URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.pkcs12</defaultValue>
    <userData>p12_rsa768.p12</userData>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>username</name>
    <type>string</type>
    <displayString>Key Server Username:</displayString>
    <userData>root</userData>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>password</name>
    <type>password</type>
    <displayString>Key Server Password:</displayString>
    <userData>roZes123</userData>
  </childElements>
</childElements>
<childElements>
  <tagName>IF</tagName>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>der_key_url</name>
    <type>url</type>
    <displayString>SSL Key URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.key</defaultValue>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>der_key_url.name</name>
    <type>Hidden</type>
    <displayString>SSL Key URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.key</defaultValue>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>der_cer_url</name>
    <type>url</type>
    <displayString>SSL Cert URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.der</defaultValue>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>der_cer_url.name</name>
    <type>Hidden</type>
    <displayString>SSL Cert URL:</displayString>
    <defaultValue>sftp://1.2.3.4/dir/myfile.der</defaultValue>
  </childElements>

```

```

</childElements>
<childElements>
  <tagName>VARIABLE</tagName>
  <name>der_username</name>
  <type>string</type>
  <displayString>Key Server Username:</displayString>
</childElements>
<childElements>
  <tagName>VARIABLE</tagName>
  <name>der_password</name>
  <type>password</type>
  <displayString>Key Server Password:</displayString>
</childElements>
</childElements>
<childElements>
  <tagName>IF</tagName>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>key</name>
    <type>ssl_key</type>
    <displayString>SSL Key:</displayString>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>key.name</name>
    <type>Hidden</type>
    <displayString>SSL Key:</displayString>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>key.contents</name>
    <type>Hidden</type>
    <displayString>SSL Key:</displayString>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>cert</name>
    <type>certificate</type>
    <displayString>SSL Certificate:</displayString>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>cert.name</name>
    <type>Hidden</type>
    <displayString>SSL Certificate:</displayString>
  </childElements>
  <childElements>
    <tagName>VARIABLE</tagName>
    <name>cert.contents</name>
    <type>Hidden</type>
    <displayString>SSL Certificate:</displayString>
  </childElements>
</childElements>
<childElements>
  <tagName>VARIABLE</tagName>
  <name>passphrase</name>
  <type>password</type>
  <displayString>Cert/Key Passphrase:</displayString>
  <userData />
</childElements>
<childElements>
  <tagName>VARIABLE</tagName>
  <name>compress</name>
  <type>boolean</type>

```

```

        <displayName>Enable Compression:</displayName>
        <defaultValue>true</defaultValue>
        <userData>true</userData>
    </childElements>
</item>
<item>
    <tagName>GROUP</tagName>
    <name>network</name>
    <type>group</type>
    <displayName>Network Configuration</displayName>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>device</name>
        <type>device</type>
        <displayName>Load Balancer
(Device:VirtualContext) :</displayName>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>vlans</name>
        <type>vlans</type>
        <displayName>Client VLANs:</displayName>
        <userData>ALL_VLAN</userData>
    </childElements>
    <childElements>
        <tagName>VARIABLE</tagName>
        <name>autoNat</name>
        <type>boolean</type>
        <displayName>Enable Source NAT:</displayName>
        <userData>>false</userData>
    </childElements>
    </item>
</templateInputs>
</ns2:createTemplateInstance>
</S:Body>
</S:Envelope>

```

Example Response

The following example shows the XML response that contains the instanceId object as the output parameter:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:createTemplateInstanceResponse xmlns:ns2="http://anm.cisco.com">
      <instanceId>5</instanceId>
    </ns2:createTemplateInstanceResponse>
  </S:Body>
</S:Envelope>

```

listTemplateInstances

This API lists all template instances, which includes the template name, description, application, application version, and so forth.

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.

Output Parameters

Parameter	Description
templateInstanceArray	List of template instances.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listTemplateInstances](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header />
  <S:Body>
    <ns2:listTemplateInstances xmlns:ns2="http://anm.cisco.com">
      <sessionToken>
        <sid>4918324415074336347</sid>
      </sessionToken>
    </ns2:listTemplateInstances>
  </S:Body>
</S:Envelope>
```

Example Response

The following example shows the XML response that contains the list of templateInstances objects as the output parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:listTemplateInstancesResponse xmlns:ns2="http://anm.cisco.com">
      <TemplateInstances>
        <item>
          <instanceId>1</instanceId>
          <templateName>FTP1</templateName>
          <applicationType>FTP</applicationType>
          <type>Deployed</type>
          <status>Complete</status>
          <deviceId>10.77.247.20:new</deviceId>
          <lastUpdatedTime>2012-02-09 00:25:58</lastUpdatedTime>
        </item>
      </TemplateInstances>
    </ns2:listTemplateInstancesResponse>
  </S:Body>
```

```
</S:Envelope>
```

deleteTemplateInstance

This API takes an instance ID and deviceID and deletes the configuration based on the instance ID.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
instanceId	Template instance ID. To retrieve a list of available template instances, use the listTemplateInstances API.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device ID.
String	Invalid template input.

Example Request

The following example shows an XML-formatted SOAP request to perform the [deleteTemplateInstance](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header />
  <S:Body>
    <ns2:deleteTemplateInstance xmlns:ns2="http://anm.cisco.com">
      <sessionToken>
        <sid>4918324415074336347</sid>
      </sessionToken>
      <deviceID>
        <name>new</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>10.77.247.20</ipAddr>
      </deviceID>
      <instanceId>1</instanceId>
    </ns2:deleteTemplateInstance>
  </S:Body>
</S:Envelope>
```

Example Response The following example shows the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:deleteTemplateInstanceResponse xmlns:ns2="http://anm.cisco.com">
      <status>success</status>
    </ns2:deleteTemplateInstanceResponse>
  </S:Body>
</S:Envelope>
```

listCheckpoints

This API lists all the checkpoints in ANM inventory for a specified device.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.

Output Parameters

Parameter	Description
Checkpoints	List of checkpoints.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device type.
String	Invalid device ID.

Example Request

The following example shows an XML-formatted SOAP request to perform the [listCheckpoints](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <listCheckpoints xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>-8567831601320378522</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>172.16.244.214</ipAddr>
      </deviceID>
    </listCheckpoints>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

The following example shows the XML response that contains the list of Checkpoints objects as the output parameter:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
```

```
<ns2:listCheckpointsResponse xmlns:ns2="http://anm.cisco.com">
  <Checkpoints xmlns="">
    <item>
      <name>35Admin</name>
      <size>11128</size>
      <date>Sun May 1 15:10:04 2011</date>
    </item>
    <item>
      <name>allcontexts</name>
      <size>49294</size>
      <date>Sun May 1 15:10:04 2011</date>
    </item>
    <item>
      <name>app-tmpl-1314729175709</name>
      <size>3917</size>
      <date>Mon Sep 5 10:54:45 2011</date>
    </item>
    <item>
      <name>beforeupgrade</name>
      <size>8233</size>
      <date>Sun May 1 15:10:04 2011</date>
    </item>
  </Checkpoints>
</ns2:listCheckpointsResponse>
</S:Body>
</S:Envelope>
```


createCheckpoint

This API creates a checkpoint on a specified device.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
checkpointName	Name of the checkpoint to create.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device type.
String	Invalid device ID.
String	Checkpoint already exists: <i>checkpoint_name</i> .

Example Request

The following example shows an XML-formatted SOAP request to perform the [createCheckpoint](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <createCheckpoint xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>-5488932206595059144</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>172.23.244.214</ipAddr>
      </deviceID>
      <checkpointName xmlns="">test-checkpoint</checkpointName>
    </createCheckpoint>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response The following example shows the XML response:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:createCheckpointResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

deleteCheckpoint

This API deletes a checkpoint from a specified device.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
checkpointName	Name of the checkpoint to delete. To retrieve a list of available checkpoints, use the listCheckpoints API.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device type.
String	Invalid device ID.
String	Invalid checkpoint name.
String	Checkpoint doesn't exist: <i>checkpoint_name</i>

Example Request

The following example shows an XML-formatted SOAP request to perform the [deleteCheckpoint](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <deleteCheckpoint xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>-5019678635804080238</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>172.16.244.214</ipAddr>
      </deviceID>
      <checkpointName xmlns="">test-checkpoint</checkpointName>
    </deleteCheckpoint>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response The following example shows the XML response:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:deleteCheckpointResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

rollbackCheckpoint

This API rolls back a device's running configuration to a specified checkpoint.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
checkpointName	Checkpoint to which you want to roll back.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device type.
String	Invalid device ID.
String	Invalid checkpoint name.
String	Checkpoint doesn't exist: <i>checkpoint_name</i>

Example Request

The following example shows an XML-formatted SOAP request to perform the [rollbackCheckpoint](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <rollbackCheckpoint xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>4270884926038298576</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>172.16.244.214</ipAddr>
      </deviceID>
      <checkpointName xmlns="">test-checkpoint</checkpointName>
    </rollbackCheckpoint>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response The following example shows the XML response:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:rollbackCheckpointResponse xmlns:ns2="http://anm.cisco.com"/>
  </S:Body>
</S:Envelope>
```

freeFormShow

This API returns the output of a specified **show** command sent to a specified device. The API includes the **xml-show** command, which specifies that the **show** command output is either regular CLI display output or it is in XML format.

This API is supported on ACE devices (device type: VIRTUAL_CONTEXT).

Input Parameters

Parameter	Description
SessionToken	Session key of the application user.
deviceID	Virtual context name, devicetype, and IP address attributes. For the ACE module only, the device ID includes the host chassis slot and chassis IP address attributes.
command	Name of the show command. Specify the command without using the word show . For example, to specify the show version command output, just use the word version .
isXMLized	Command output format: <ul style="list-style-type: none"> On (true)—XML format. Off (false)—CLI display format.

Output Parameters

Parameter	Description
None	None.

Exceptions

Type	Exception String
String	Invalid session ID.
String	Session timeout.
String	Invalid device type.
String	Invalid device ID.
String	Internal error: command parse error.

Example Request

The following example shows an XML-formatted SOAP request to perform the [freeFormShow](#) action. SessionToken is the input parameters for this particular action.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <freeFormShow xmlns="http://anm.cisco.com">
      <sessionToken xmlns="">
        <sid>-869899350867169519</sid>
      </sessionToken>
      <deviceID xmlns="">
        <name>Admin</name>
        <deviceType>VIRTUAL_CONTEXT</deviceType>
        <ipAddr>172.16.244.214</ipAddr>
      </deviceID>
    </freeFormShow>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    </deviceID>
    <command xmlns="">access-list ALL</command>
    <isXMLized xmlns="">>false</isXMLized>
  </freeFormShow>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Example Response

The following example shows the XML response:

```

<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:freeFormShowResponse xmlns:ns2="http://anm.cisco.com">
      <output xmlns="">access-list:ALL, elements: 3, status: ACTIVE
      remark :
      access-list ALL line 8 extended permit ip any any (hitcount=333800)
      access-list ALL line 18 extended permit ip any host 172.16.244.230 (hitcount=0)
      access-list ALL line 28 extended permit ip any host 10.10.10.1 (hitcount=0)</output>
    </ns2:freeFormShowResponse>
  </S:Body>
</S:Envelope>

```