



CHAPTER 2

Cisco Multicast Manager API Reference

This chapter contains the following sections:

- [API Operations, page 2-1](#)
- [Spring Framework Beans, page 2-46](#)

API Operations

The WSDL definition file for the CMM API— *cmmNB.wsdl*, contains a set of definitions that define the following elements:

- Data types and associated messages
- A list of portTypes and associated operations
- Bindings for the SOAP protocol
- Services and associated ports

See [Appendix B, “cmmNB.wsdl File”](#) for a listing of the *cmmNB.wsdl* file.

The CMM API set comprises a set of operations that you can perform using these elements. Using the specific API operations, you can perform the operations shown in [Table 2-1](#).

Table 2-1 CMM API Operations

Operation	See
Add Interface Polling	See Add Interface Polling, page 2-2
Add Layer 2 Polling	See Add Layer 2 Polling, page 2-4
Add Layer 3 and Layer 2 devices in Bulk	Add Layer 3 and Layer 2 Devices in Bulk, page 2-5
Add RP Polling	See Add RP Polling, page 2-6
Add RPF Polling	See Add RPF Polling, page 2-7
Add S,G Polling	See Add S,G Polling, page 2-8
Add S,G Time-Based Polling	See Add S,G Time-Based Polling, page 2-9
Add SSM Polling	See Add SSM Polling, page 2-11
Add Tree Polling	See Add Tree Polling, page 2-12
Add Video Probe Polling	See Add Video Probe Polling, page 2-14

Table 2-1CMM API Operations (continued)

Operation	See
Add Video Probes in Bulk	See Add Video Probes in Bulk , page 2-13
Add VidMon Polling	See Add VidMon Polling , page 2-15
Delete Devices in Bulk	See Delete Devices in Bulk , page 2-17
Delete Interface Polling	See Delete Interface Polling , page 2-18
Delete Layer 2 Polling	See Delete Layer 2 Polling , page 2-19
Delete RP Polling	See Delete RP Polling , page 2-20
Delete RPF Polling	See Delete RPF Polling , page 2-21
Delete S, G Polling	See Delete S,G Polling , page 2-22
Delete SSM Polling	See Delete SSM Polling , page 2-23
Delete Tree Polling	See Delete Tree Polling , page 2-25
Delete Video Probe Polling	See Delete Video Probe Polling , page 2-27
Delete VidMon Polling	See Delete VidMon Polling , page 2-28
Get a Multicast Trace Image File	See Get Multicast Trace Image File , page 2-31
Get Interface Polling Configuration	See Get Layer 2 Polling Configuration , page 2-30
Get Layer 2 Polling Configuration	See Get Layer 2 Polling Configuration , page 2-30
Get Multicast Devices in a Domain	See Get Multicast Devices in a Domain , page 2-32
Get Multicast Events	See Get Multicast Events , page 2-33
Get RP Polling Configuration	See Get RP Polling Configuration , page 2-34
Get RPF Polling Configuration	See Get RPF Polling Configuration , page 2-35
Get S,G on a Device	See Get S,G on a Device , page 2-36
Get S,G in a Domain	See Get S,G in a Domain , page 2-38
Get S,G Polling Configuration	See Get S,G on a Device , page 2-36
Get S,G Time-Based Polling Configuration	See Get S,G Time-Based Polling Configuration , page 2-39
Get SSM Polling Configuration	See Get SSM Polling Configuration , page 2-41
Get Tree Polling Configuration	See Get Tree Polling Configuration , page 2-41
Get Video Probe Polling Configuration	See Get Video Probe Polling Configuration , page 2-42
Get VidMon Polling Configuration	See Get VidMon Polling Configuration , page 2-43
Set Global Polling Configuration	See Set Global Polling Configuration , page 2-44

Add Interface Polling

Use this operation to add polling configuration for a specified interface on a device.

Request Message

JAXBElement<IntPollingListType>

- *IntPollingListType*—A JAXB-generated class that encapsulates an array list of *IntPollingType*.
- *IntPollingType*—A JAXB-generated wrapper class that encapsulates an interface name, interface index, bandwidth, router name, polling direction, high PPS threshold and low PPS threshold for polling of the interface.

Response Message

The response message contains a list of the interfaces that have been configured for polling.

JAXBElement<IntPollingListType>

- *IntPollingListType*—A JAXB-generated class that encapsulates an array list of *IntPollingType*.
- *IntPollingType*—A JAXB-generated wrapper class that encapsulates an interface name, interface index, bandwidth, router name, polling direction, high PPS threshold and low PPS threshold for polling of the interface.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void addIntPolling() {
    try {
        IntPollingListType sgplt = new IntPollingListType();
        sgplt.setDomainName("test-01");

        IntPollingType sgt = new IntPollingType();
        sgt.setInterfaceName("Vlan1");
        sgt.setIfindex(52);
        sgt.setBandwidth(100000);
        sgt.setRouterName("cmm-6503-c2.cisco.com");
        sgt.setDirection(0);
        sgt.setHighThreshold(12312);
        sgt.setLowThreshold(233);
        sgplt.getInterfaces().add(sgt);

        sgt = new IntPollingType();
        sgt.setInterfaceName("Vlan1");
        sgt.setIfindex(52);
        sgt.setBandwidth(100000);
        sgt.setRouterName("cmm-6503-c2.cisco.com");
        sgt.setDirection(1);
        sgt.setHighThreshold(1111);
        sgt.setLowThreshold(99);
        sgplt.getInterfaces().add(sgt);

        JAXBElement<IntPollingListType> req =
objf.createMulticastIntPollingAddRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("addIntPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());
    }
}
```

```

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Add Layer 2 Polling

Use this operation to add Layer 2 polling for a specified switch in a domain.

Request Message

JAXBElement<L2PollingConfigType>

- *L2PollingConfigType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address or hostname, device type enumeration, polling direction, high PPS threshold and low PPS threshold.

Response Message:

The response message contains an integer specifying the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void addL2Polling() {
    try {
        L2PollingConfigType lt = new L2PollingConfigType();
        lt.setDomainName("test-01");

        L2PollingType sgt = new L2PollingType();
        sgt.setSwitchName("cmm-1801-sa4");
        sgt.setPortName("Fa3");
        sgt.setPortIfIndex(7);
        sgt.setDirection(0);
        sgt.setHighThreshold(12312);
        sgt.setLowThreshold(233);
        lt.setL2(sgt);

        JAXBElement<L2PollingConfigType> req =
objf.createMulticastL2PollingAddRequest(lt);
        JAXBElement<Integer> res = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(req);

        System.out.println("status = " + res.getValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
    }
}
```

Add Layer 3 and Layer 2 Devices in Bulk

Use this operation for bulk addition of layer 3 and layer 2 devices. All devices must be added in the same domain.

Request Message

JAXBElement<DeviceListType>

- *DeviceListType*—A JAXB-generated class that encapsulates an array list of *DeviceType*.
- *DeviceType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model, and device name.

Response Message:

JAXBElement<DeviceListType>

- *DeviceListType*—A JAXB-generated class that encapsulates an array list of *DeviceType*.
- *DeviceType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model and device name.

This message contains a list of devices that have been successfully scheduled for discovery. Note that this message does not offer any guarantee that all devices listed in message have been completely discovered at the moment when the message is received by the client program.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void AddUpdateMulticastDevices() {
    DeviceListType dl= objf.createDeviceListType();
    DeviceType dt1= objf.createDeviceType();
    DeviceType dt2= objf.createDeviceType();
    //DeviceType dt3= objf.createDeviceType();
    //DeviceType dt4= objf.createDeviceType();
    //DeviceType dt5= objf.createDeviceType();
    //DeviceType dt6= objf.createDeviceType();

    dt1.setDomainName("test-01");
    dt1.setIp("172.20.111.198");
    dt1.setModel("7206");
    dt1.setName("cmm-7206-sd1");
    dt1.setCommunityString("public");
    dt1.setType(DeviceTypeEnum.ROUTER);

    //dt2.setDomainName("test-01");
    //dt2.setIp("172.20.111.199");
    //dt2.setModel("7604");
    //dt2.setName("cmm-7204-sd2");
    // dt2.setCommunityString("public");
    // dt2.setType(DeviceTypeEnum.ROUTER);
}
/*
```

```

dt3.setDomainName("test");
dt3.setIp("10.10.10.30");
dt3.setModel("7600");
dt3.setName("cmmdev3");
dt3.setType("router");

dt4.setDomainName("test");
dt4.setIp("10.10.10.40");
dt4.setModel("7600");
dt4.setName("cmmdev4");
dt4.setType("router");

dt5.setDomainName("test");
dt5.setIp("10.10.10.50");
dt5.setModel("7600");
dt5.setName("cmmdev5");
dt5.setType("router");

dt6.setDomainName("test");
dt6.setIp("10.10.10.60");
dt6.setModel("7600");
dt6.setName("cmmdev6");
dt6.setType("router");
*/
dl.getDevice().add(dt1);
dl.getDevice().add(dt2);
//dl.getDevice().add(dt3);
//dl.getDevice().add(dt4);
//dl.getDevice().add(dt5);
//dl.getDevice().add(dt6);
try{
    JAXBElement<DeviceListType> sgresp
=(JAXBElement<DeviceListType>getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastAddUpdateDevicesRequest(dl));
    DeviceListType sgt = sgresp.getValue();
    System.out.println("=====");
    System.out.println("Getting Successfully added devices for domain ");
    System.out.println("=====");

    for(int i=0;i<sgt.getDevice().size();i++)
    {
        DeviceType sg = sgt.getDevice().get(i);
        System.out.println("Name "+sg.getName()+ " ip "+sg.getIp());
    }

}
catch(SoapFaultClientException ex) {
    System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
    System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
}

}

```

Add RP Polling

Use this operation to add Rendezvous Point (RP) polling for a specified router.

Request Message

JAXBElement<RPPollingListType>

- *RPPollingListType*—A JAXB-generated class that indicates the domain name and encapsulates an array list of *RPPollingType*.
- *RPPollingType*—A JAXB-generated wrapper class that encapsulates a device IP address or hostname, and a group limit for monitoring.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void addRPPolling() {
    try {
        RPPollingListType lt = new RPPollingListType();
        lt.setDomainName("test-01");

        RPPollingType sgt = new RPPollingType();
        sgt.setRouter("cmm-6506-c3");
        sgt.setGroupCount(3);
        lt.getRppoll().add(sgt);

        JAXBElement<RPPollingListType> req =
objf.createMulticastRPPollingAddRequest(lt);
        JAXBElement<Integer> res = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(req);

        System.out.println("status = " + res.getValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Add RPF Polling

Use this operation to add Reverse Path Forwarding (RPF) polling configuration for a specified source and group on a router.

Request Message

JAXBElement<RPFPollingConfigType>

- *RPFPollingConfigType*—A JAXB-generated wrapper class that encapsulates a domain name, router name or IP address, source and group IP addresses, and a delta value that specifies the number of RPF failures per sampling period that trigger a report.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

/public void addRPFPolling() {
    try {
        RPFPollingConfigType lt = new RPFPollingConfigType();
        lt.setDomainName("test-01");

        RPFPollingType sgt = new RPFPollingType();
        sgt.setRouterName("cmm-6503-c2.cisco.com");
        sgt.setGroupIp("126.0.1.18");
        sgt.setSourceIp("239.232.2.0");
        sgt.setDelta(7);
        lt.setRpf(sgt);

        JAXBElement<RPFPollingConfigType> req =
objf.createMulticastRPFPollingAddRequest(lt);
        JAXBElement<Integer> res = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(req);

        System.out.println("status = " + res.getValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Add S,G Polling

Use this operation to add S,G polling for specified sources and groups on specified routers.

Request Message

JAXBElement<*SGPollingListType*>

- *SGPollingListType*—A JAXB-generated wrapper class that encapsulates a domain name and array list of *SGPollingType*.
- *SGPollingType*—A JAXB-generated class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, and a low threshold.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void addSGPolling() {
    try {
        SGPollingListType sgplt = new SGPollingListType();
        sgplt.setDomainName("test-01");

        SGPollingType sgt = new SGPollingType();
        sgt.setSourceIp("126.32.129.28");
        sgt.setGroupIp("239.254.1.11");
        sgt.setRouterName("cmm-7604-sd2.dns-sj.cisco.com");
        sgt.setUnits(0);
        sgt.setHigh(12312);
        sgt.setLow(233);
        sgplt.getSg().add(sgt);

        sgt = new SGPollingType();
        sgt.setSourceIp("126.32.129.29");
        sgt.setGroupIp("239.254.1.13");
        sgt.setRouterName("cmm-7604-sd2.dns-sj.cisco.com");
        sgt.setUnits(1);
        sgt.setHigh(45623);
        sgt.setLow(342);
        sgplt.getSg().add(sgt);

        JAXBElement<SGPollingListType> req =
objf.createMulticastSGPollingAddRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("addSGPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Add S,G Time-Based Polling

Use this operation to add time-based S,G polling for specified sources and groups on specified routers.

Request Message

JAXBElement<*SGTimePollingListType*>

- *SGTimePollingListType*—A JAXB-generated class that encapsulates an a domain name and an array list of *SGTimePollingType*.
- *SGTimePollingType*—A JAXB-generated wrapper class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, a low threshold, a start time hour and minutes, stop time hour and minutes, and a day value.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void addSGTimePolling() {
    try {
        SGTimePollingListType sgplt = new SGTimePollingListType();
        sgplt.setDomainName("test-01");

        SGTimePollingType sgt = new SGTimePollingType();
        sgt.setSource("126.0.1.18");
        sgt.setGroup("239.232.2.0");
        sgt.setRouter("cmm-7206-sd1");
        sgt.setUnits(1);
        sgt.setHighThreshold(66);
        sgt.setLowThreshold(55);
        sgt.setStartTimeHur(1);
        sgt.setStartTimeMin(2);
        sgt.setStopTimeHur(3);
        sgt.setStopTimeMin(4);
        sgt.setDays("Tue");
        sgplt.getSgtime().add(sgt);

        sgt = new SGTimePollingType();
        sgt.setSource("126.0.1.12");
        sgt.setGroup("239.232.0.0");
        sgt.setRouter("4503");
        sgt.setUnits(1);
        sgt.setHighThreshold(45623);
        sgt.setLowThreshold(342);
        sgt.setStartTimeHur(5);
        sgt.setStartTimeMin(6);
        sgt.setStopTimeHur(7);
        sgt.setStopTimeMin(8);
        sgt.setDays("Sun");
        sgplt.getSgtime().add(sgt);

        JAXBElement<SGTimePollingListType> req =
objf.createMulticastSGTimePollingAddRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("addSGPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());
    }
}
```

```

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
                ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
                ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Add SSM Polling

Use this operation to add Source Specific Multicast (SSM) polling configuration for a specified source and group.

Request Message

JAXBElement<*SSMPollingConfigType*>

- *SSMPollingConfigType*—A JAXB-generated class that encapsulates an array list of *SSMPollingType*.
- *SSMPollingType*—A JAXB-generated wrapper class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, a low threshold, a start time hour and minutes, stop time hour and minutes, and a day value.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void addSSMPolling() {
    try {
        //126.0.1.13 224.0.1.40 1 0 19 0 Sun 0 0 -
        SSMPollingConfigType lt = new SSMPollingConfigType();
        lt.setDomainName("test-01");

        SSMPollingType ssmt = new SSMPollingType();
        ssmt.setSourceIp("126.0.1.12");
        ssmt.setGroupIp("239.232.0.0");
        ssmt.setConfigureThreshold(false);
        ssmt.setUnits(0);
        ssmt.setHigh(800);
        ssmt.setLow(700);
        ssmt.setDays("Sun");
        ssmt.setStartTimeHur(1);
    }
}

```

```

        ssmt.setStartTimeMin(59);
        ssmt.setStopTimeHur(23);
        ssmt.setStopTimeMin(59);
        lt.setSsm(ssmt);

        JAXBElement<SSMPollingConfigType> req =
objf.createMulticastSSMPollingAddRequest(lt);
        JAXBElement<Integer> res = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(req);

        System.out.println("status for SSM Polling = " + res.getValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Add Tree Polling

Use this operation to add Tree Polling configuration for a specified domain and trace baseline.

Request Message

JAXBElement<*TreePollingConfigType*>

- *TreePollingConfigType*—A JAXB-generated class that contains a domain name and a *TreePollingType* object.
- *TreePollingType*—A JAXB-generated wrapper class that encapsulates a saved baseline and indicates whether it is to be used as a comparison baseline.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void addTreePolling() {
    try {
        TreePollingConfigType lt = new TreePollingConfigType();
        lt.setDomainName("test-01");

        TreePollingType sgt = new TreePollingType();
        sgt.setBaseLine("baseline2.trace");
        sgt.setIsCompareBaseLine(true);
        lt.setTree(sgt);

        JAXBElement<TreePollingConfigType> req =
objf.createMulticastTreePollingAddRequest(lt);
        JAXBElement<Integer> res = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(req);

        System.out.println("status = " + res.getValue());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
}

```

Add Video Probes in Bulk

Description

Adding video probes is slightly different from adding other devices in CMM. Your code must identify both the router and the interface where the video probe is connected. The Request message must specify the community strings of both the router and the video probe. You can use this operation to add video probes in bulk. Your code also must associate probes with their monitoring applications on the CMM GUI.

This API operation returns a list of successfully added video probes. Since video probe addition is a synchronous operation, the returned list guarantees that the probes have been added to CMM.

Request Message

JAXBElement<VideoProbeListType>

- *VideoProbeListType*—A JAXB generated class that encapsulates an array list of *VideoProbeType* objects.
- *VideoProbeType*—A JAXB-generated class that encapsulates the probe name, probe IP address, probe ro community string, probe rw community string, router IP address, router ro community string, domain name, and video probe type.

Response Message

JAXBElement<VideoProbeListType>

The response message contains a list of successfully added video probes (*VideoProbeListType* objects).

- *VideoProbeListType*—A JAXB generated class that encapsulates an array list of *VideoProbeType* objects.
- *VideoProbeType*—A JAXB-generated class that encapsulates the probe name, probe IP address, probe ro community string, probe rw community string, router IP address, router ro community string, domain name, and video probe type.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code sample

// Create video probe object and fill in parameters
VideoProbeType videoProbe= new VideoProbeType();
    videoProbe.setProbeName("Bridge3");
    videoProbe.setProbeIpAddress("10.86.1.233");
    videoProbe.setRoCommunity("public");
    videoProbe.setRwCommunity("private");
    videoProbe.setRouterIpAddress("126.1.9.15");
    videoProbe.setRouterRoCommunity("lab");
    videoProbe.setInterfaceDescription("Gil/1");
    videoProbe.setDomainName("test");
    videoProbe.setApplicationType(VideoApplicationTypeEnum.BRIDGETECH);
    videoProbe.setSnmpRetries(8);
```

```

        videoProbe.setSnmptimeout(2);

// Create video probe list
VideoProbeListType vl= objf.createVideoProbeListType();
vl.getVideoProbe().add(videoProbe);

// create requestMessage
JAXBElement<VideoProbeListType> requestMessage =
objf.createMulticastAddUpdateVideoProbeRequest(vl);

    try{
// Send and Receive
        JAXBElement<VideoProbeListType> responseMessage
=(JAXBElement<VideoProbeListType>)getWebServiceTemplate().marshalSendAndReceive(requestMes
sage);
// Get list of successfully added probes.
        VideoProbeListType vt = responseMessage.getValue();
        VideoProbeType vp = (VideoProbeType) vt.getVideoProbe().get(0);
        System.out.println("Video probe name is Name is :"+vp.getProbeName());
    }
    catch(SoapFaultClientException ex) {
        // Error
        System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
        System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
    }
    catch(Exception e) {
    }

```

Add Video Probe Polling

Use this operation to add video probe polling for a specified video probe.

Request Message

JAXBElement<VideoProbePollingListType>

- *VideoProbePollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VideoProbePollingType*.
- *VideoProbePollingType*—A JAXB-generated wrapper class that encapsulates a video probe name, delay Factor (DF) threshold in milliseconds, and Media Loss Rate (MLR) value.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

// Code Sample
public void addVideoProbePolling() {
    try {
        VideoProbePollingListType sgplt = new VideoProbePollingListType();
        sgplt.setDomainName("test-01");

        VideoProbePollingType sgt = new VideoProbePollingType();
        sgt.setProbeName("cmm-ckt-01");
        sgt.setDf(2);
        sgt.setLoss(98);
    }
}

```

```

sgplt.getVideoProbes().add(sgt);

sgt = new VideoProbePollingType();
sgt.setProbeName("cmm-glt-01");
sgt.setDf(0);
sgt.setLoss(50);
sgplt.getVideoProbes().add(sgt);

JAXBElement<VideoProbePollingListType> req =
objf.createMulticastVideoProbePollingAddRequest(sgplt);
JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

BulkMessageReturnWS ret = res.getValue();
System.out.println("=====");
System.out.println("addVideoProbePolling");
System.out.println("=====");

System.out.println("status = " + ret.getReturnStatus());

for (int i = 0; i < ret.getFailureList().size(); i++) {
    System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
}

for (int i = 0; i < ret.getSuccessList().size(); i++) {
    System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

// Create device list type object
DeviceListType dl= objf.createDeviceListType();
// Create individual device objects

```

Add VidMon Polling

Use this operation add VidMon polling for a specified Cisco 76xx device or ASR 9000 device.

Request Message

JAXBElement<VidmonPollingListType>

- *VidmonPollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VidmonPollingType*.
- *VidmonPollingType*—A JAXB-generated wrapper class that encapsulates a device name, VidMon type, delay factor (DF) in milliseconds, a Media Loss Rate (MLR) threshold value (number of packets), a milli-percentage value that specifies a MRV maximum threshold., and a milli-percentage value that specifies a MRV minimum threshold.

**Note**

The *setVidmonType* field in the *VidmonPollingType* class indicates the type of VidMon device for which you are adding polling. If the device is a Cisco 76xx series router, specify 1. If the device is an ASR 9000 series device, specify 0. Only 76xx devices support MLR monitoring; therefore you should only specify an MLR value in the *setMlr* call if the device is a 76xx device.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void addVidmonPolling() {
    try {
        VidmonPollingListType sgplt = new VidmonPollingListType();
        sgplt.setDomainName("test-01");

        VidmonPollingType sgt = new VidmonPollingType();
        sgt.setVidmonName("cmm-7604-d2.dns-sj.cisco.com");
        sgt.setVidmonType(1);
        sgt.setDf(2);
        sgt.setMlr(24);
        sgt.setMrvMin(42);
        sgt.setMrvMax(98);
        sgplt.getVidmons().add(sgt);

        JAXBElement<VidmonPollingListType> req =
objf.createMulticastVidmonPollingAddRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("addVidmonPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
// multicastDevicesRequest
public void getAllSGPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<SGPollingListType> sgresp = (JAXBElement<SGPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllSGPollingRequest(d
t));
    }
}
```



```

SGPollingListType sgt = sgresp.getValue();
System.out.println("=====");
System.out.println("Getting SG Polling Configuration for domain " + dt);
System.out.println("=====");

for (int i = 0; i < sgt.getSg().size(); i++) {
    SGPollingType sg = sgt.getSg().get(i);
    System.out.println(sg.getSourceIp() + "\t" + sg.getGroupIp() + "\t" +
sg.getRouterName()+ "\t" +sg.getUnits()+ "\t" +sg.getHigh()+ "\t" +sg.getLow());
}

} catch (SoapFaultClientException ex) {
    System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
    System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
}

}

```

Delete Devices in Bulk

Use this operation to delete devices in bulk. You can also use this operation to delete all type of devices.

Request Message

JAXBElement<*DeviceListType*>

- *DeviceListType*—A JAXB-generated class that encapsulates an array list of *DeviceType*.
- *DeviceType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model and device name.

Response Message

JAXBElement<*DeviceListType*>

The response message contains list of successfully deleted devices.

- *DeviceListType*—A JAXB-generated class that encapsulates an array list of *DeviceType*.
- *DeviceType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model and device name.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

// Code sample

public void DeleteMulticastDevices() {
    DeviceListType dl= objf.createDeviceListType();
    DeviceType dt1= objf.createDeviceType();

    dt1.setDomainName("test-01");
    dt1.setIp("172.20.111.198");
    // dt1.setModel("7600");
    dt1.setName("cmm-7206-sd1");
    dt1.setType(DeviceTypeEnum.ROUTER);
}

```

```

dl.getDevice().add(dt1);
try{
    JAXBElement<DeviceListType> sgresp
= (JAXBElement<DeviceListType>)getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastDeleteDevicesRequest(dl));
    DeviceListType sgt = sgresp.getValue();
    System.out.println("=====");
    System.out.println("Getting Successfully Deleted devices for domain ");
    System.out.println("=====");

    for(int i=0;i<sgt.getDevice().size();i++)
    {
        DeviceType sg = sgt.getDevice().get(i);
        System.out.println("Name "+sg.getName()+ " ip "+sg.getIp());
    }

}
catch(SoapFaultClientException ex) {
    System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
    System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
}
}

```

Delete Interface Polling

Use this operation to delete interface polling for a specified interface on a specified router.

Request Message

JAXBElement<IntPollingListType>

- *IntPollingListType*—A JAXB-generated class that encapsulates an array list of *IntPollingType*.
- *IntPollingType*—A JAXB-generated wrapper class that encapsulates an interface name, interface index, bandwidth, router name, polling direction, high PPS threshold and low PPS threshold for polling of the interface.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void deleteIntPolling() {
    try {
        IntPollingListType sgplt = new IntPollingListType();
        sgplt.setDomainName("test-01");

        IntPollingType sgt = new IntPollingType();
        sgt.setInterfaceName("GigabitEthernet1/1");
        sgt.setRouterName("4503");
        sgt.setIfindex(2);
    }
}

```

```

        sgt.setDirection(0);
        sgplt.getInterfaces().add(sgt);

        JAXBElement<IntPollingListType> req =
objf.createMulticastIntPollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteIntPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Delete Layer 2 Polling

Use this operation to delete Layer 2 polling for a specified switch in a domain.

Request Message

JAXBElement<L2PollingConfigType>

- *L2PollingConfigType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address or hostname, device type enumeration, polling direction, high PPS threshold and low PPS threshold.

Response Message:

The response message contains an integer specifying the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void deleteL2Polling() {
    try {
        L2PollingListType lt = new L2PollingListType();
        lt.setDomainName("test-01");

        L2PollingType sgt = new L2PollingType();
        sgt.setSwitchName("cmm-1801-sa4");
        sgt.setPortName("Fa3");
    }
}

```

```

        sgt.setPortIfIndex(7);
        sgt.setDirection(0);
        sgt.setHighThreshold(12312);
        sgt.setLowThreshold(233);
        lt.getL2().add(sgt);

        JAXBElement<L2PollingListType> req =
objf.createMulticastL2PollingDeleteRequest(lt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteSGPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = "+
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Delete RP Polling

Use this operation to delete Rendezvous Point (RP) polling for a specified router.

Request Message

JAXBElement<*RPPollingListType*>

- *RPPollingListType*—A JAXB-generated class that indicates the domain name and encapsulates an array list of *RPPollingType*.
- *RPPollingType*—A JAXB-generated wrapper class that encapsulates a device IP address or hostname, and a group limit for monitoring.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

// Code Sample
public void deleteRPPolling() {
    try {
        RPPollingListType sgplt = new RPPollingListType();

```

```

sgplt.setDomainName("test-01");

RPPollingType sgt = new RPPollingType();
sgt.setRouter("cmm-6506-c3");

sgplt.getRppoll().add(sgt);

JAXBElement<RPPollingListType> req =
objf.createMulticastRPPollingDeleteRequest(sgplt);
JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

BulkMessageReturnWS ret = res.getValue();
System.out.println("=====");
System.out.println("delete RPPolling");
System.out.println("=====");

System.out.println("status = " + ret.getReturnStatus());

for (int i = 0; i < ret.getFailureList().size(); i++) {
    System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
}

for (int i = 0; i < ret.getSuccessList().size(); i++) {
    System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

```

Delete RPF Polling

Use this operation to delete Reverse Path Forwarding (RPF) polling configuration for a specified source and group on a route.

Request Message

`JAXBElement<RPFPollingConfigType>`

- *RPFPollingConfigType*—A JAXB-generated wrapper class that encapsulates a domain name, router name or IP address, source and group IP addresses, and a delta value that specifies the number of RPF failures per sampling period that trigger a report.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void deleteRPFPolling() {
    try {

```

```

RPFPollingListType sgplt = new RPFPollingListType();
sgplt.setDomainName("test-01");

RPFPollingType sgt = new RPFPollingType();
sgt.setGroupIp("239.232.2.0");
sgt.setRouterName("cmm-6503-c2.cisco.com");
sgt.setSourceIp("126.0.1.18");
sgt.setDelta(7);
sgplt.getRpf().add(sgt);

sgt = new RPFPollingType();
sgt.setGroupIp("239.232.0.0");
sgt.setRouterName("cmm-6503-c2.cisco.com");
sgt.setSourceIp("126.0.0.18");
sgt.setDelta(23);
sgplt.getRpf().add(sgt);

JAXBElement<RPFPollingListType> req =
objf.createMulticastRPFPollingDeleteRequest(sgplt);
JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

BulkMessageReturnWS ret = res.getValue();
System.out.println("=====");
System.out.println("deleteRPFPolling");
System.out.println("=====");

System.out.println("status = " + ret.getReturnStatus());

for (int i = 0; i < ret.getFailureList().size(); i++) {
    System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
}

for (int i = 0; i < ret.getSuccessList().size(); i++) {
    System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

```

Delete S,G Polling

Use this operation to delete S,G polling for specified sources and groups on specified routers.

Request Message

JAXBElement<*SGPollingListType*>

- *SGPollingListType*—A JAXB-generated wrapper class that encapsulates a domain name and array list of *SGPollingType*.
- *SGPollingType*—A JAXB-generated class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, and a low threshold.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void deleteSGPolling() {
    try {
        SGPollingListType sgplt = new SGPollingListType();
        sgplt.setDomainName("test-01");

        SGPollingType sgt = new SGPollingType();
        sgt.setSourceIp("126.32.129.28");
        sgt.setGroupIp("239.254.1.11");
        sgt.setRouterName("cmm-7604-sd2.dns-sj.cisco.com");
        sgt.setUnits(0);
        sgplt.getSg().add(sgt);

        sgt = new SGPollingType();
        sgt.setSourceIp("126.32.129.29");
        sgt.setGroupIp("239.254.1.13");
        sgt.setRouterName("cmm-7604-sd2.dns-sj.cisco.com");
        sgt.setUnits(1);
        sgplt.getSg().add(sgt);

        JAXBElement<SGPollingListType> req =
objf.createMulticastSGPollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteSGPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Delete SSM Polling

Use this operation to delete Source Specific Multicast (SSM) polling configuration for a specified source and group.

Request Message

JAXBElement<SSMPollingConfigType>

- *SSMPollingConfigType*—A JAXB-generated class that encapsulates an array list of *SSMPollingType*.
- *SSMPollingType*—A JAXB-generated wrapper class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, a low threshold, a start time hour and minutes, stop time hour and minutes, and a day value.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void deleteSSMPolling() {
    try {
        SSMPollingListType ssmplt = new SSMPollingListType();
        ssmplt.setDomainName("test-01");
        //126.0.1.13 224.0.1.40 1 0 19 0 Sun 0 0 -
        //String t = ssmpolling.getSourceIp() + ":" + ssmpolling.getGroupIp() + ":" +
        ssmpolling.getUnits()
            // + ":" + ssmpolling.getStartHour() + ":" +
        ssmpolling.getStartHourMin() + ":" +
            // ssmpolling.getStopHour() + ":" + ssmpolling.getStopHourMin() +
        ":" + ssmpolling.getDays();
        SSMPollingType ssmt = new SSMPollingType();
        ssmt.setSourceIp("126.0.1.13");
        ssmt.setGroupIp("224.0.1.40");
        ssmt.setConfigureThreshold(false);
        ssmt.setHigh(1);
        ssmt.setLow(0);
        ssmt.setUnits(2);
        ssmt.setDays("Sun");
        ssmt.setStartHour(1);
        ssmt.setStartHourMin(0);
        ssmt.setStopHour(19);
        ssmt.setStopHourMin(0);
        ssmplt.getSsm().add(ssmt);

        /* ssmt = new SSMPollingType();
        ssmt.setSourceIp("0.0.0.0");
        ssmt.setGroupIp("224.0.1.40");
        ssmt.setConfigureThreshold(true);
        ssmt.setHigh(800);
        ssmt.setLow(700);
        ssmt.setDays("Sun");
        ssmt.setStartHour(1);
        ssmt.setStartHourMin(59);
        ssmt.setStopHour(23);
        ssmt.setStopHourMin(59);
        ssmplt.getSsm().add(ssmt);
        ssmplt.getSsm().add(ssmt);*/
    }
}
```



```

        JAXBElement<SSMPollingListType> req =
objf.createMulticastSSMPollingDeleteRequest(ssmplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteSSMPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Delete Tree Polling

Use this operation to delete Tree Polling configuration for a specified domain and trace baseline.

Request Message

JAXBElement<*TreePollingConfigType*>

- *TreePollingConfigType*—A JAXB-generated class that contains a domain name and a *TreePollingType* object.
- *TreePollingType*—A JAXB-generated wrapper class that encapsulates a saved baseline and indicates whether it is to be used as a comparison baseline.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```

public void deleteTreePolling() {
    try {
        TreePollingListType sgplt = new TreePollingListType();
        sgplt.setDomainName("test-01");

        TreePollingType sgt = new TreePollingType();
        sgt.setBaseLine("baseline2.trace");
        sgt.setIsCompareBaseLine(true);
        sgplt.getTree().add(sgt);
    }
}

```

```

        JAXBElement<TreePollingListType> req =
objf.createMulticastTreePollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteTreePolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void deleteVideoProbePolling() {
    try {
        VideoProbePollingListType sgplt = new VideoProbePollingListType();
        sgplt.setDomainName("test-01");

        VideoProbePollingType sgt = new VideoProbePollingType();
        sgt.setProbeName("cmm-glt-01");
        sgplt.getVideoProbes().add(sgt);

        JAXBElement<VideoProbePollingListType> req =
objf.createMulticastVideoProbePollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteVideoProbePolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Delete Video Probe Polling

Use this operation to delete video probe polling for a specified video probe.

Request Message

JAXBElement<VideoProbePollingListType>

- *VideoProbePollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VideoProbePollingType*.
- *VideoProbePollingType*—A JAXB-generated wrapper class that encapsulates a video probe name, delay Factor (DF) threshold in milliseconds, and Media Loss Rate (MLR) value.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void deleteVideoProbePolling() {
    try {
        VideoProbePollingListType sgplt = new VideoProbePollingListType();
        sgplt.setDomainName("test-01");

        VideoProbePollingType sgt = new VideoProbePollingType();
        sgt.setProbeName("cmm-glt-01");
        sgplt.getVideoProbes().add(sgt);

        JAXBElement<VideoProbePollingListType> req =
objf.createMulticastVideoProbePollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteVideoProbePolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void deleteVidmonPolling() {
    try {
        VidmonPollingListType sgplt = new VidmonPollingListType();
        sgplt.setDomainName("test-01");
```

```

VidmonPollingType sgt = new VidmonPollingType();
sgt.setVidmonName("cmm-7604-sd2.dns-sj.cisco.com");
sgplt.getVidmons().add(sgt);

JAXBElement<VidmonPollingListType> req =
objf.createMulticastVidmonPollingDeleteRequest(sgplt);
JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

BulkMessageReturnWS ret = res.getValue();
System.out.println("=====");
System.out.println("deleteVidmonPolling");
System.out.println("=====");

System.out.println("status = " + ret.getReturnStatus());

for (int i = 0; i < ret.getFailureList().size(); i++) {
    System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
}

for (int i = 0; i < ret.getSuccessList().size(); i++) {
    System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
}
} catch (Exception e) {
    e.printStackTrace();
}
}

```

Delete VidMon Polling

Use this operation to delete VidMon polling for a specified Cisco 76xx device or ASR 9000 device.

Request Message

JAXBElement<*VidmonPollingListType*>

- *VidmonPollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VidmonPollingType*.
- *VidmonPollingType*—A JAXB-generated wrapper class that encapsulates a device name, VidMon type, delay factor (DF) in milliseconds, a Media Loss Rate (MLR) threshold value (number of packets), a milli-percentage value that specifies a MRV maximum threshold., and a milli-percentage value that specifies a MRV minimum threshold.



Note

The *setVidmonType* field in the *VidmonPollingType* class indicates the type of VidMon device for which you are adding polling. If the device is a Cisco 76xx series router, specify 1. If the device is an ASR 9000 series device, specify 0. Only 76xx devices support MLR monitoring; therefore you should only specify an MLR value in the setMLr call if the device is a 76xx device.

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void deleteVidmonPolling() {
    try {
        VidmonPollingListType sgplt = new VidmonPollingListType();
        sgplt.setDomainName("test-01");

        VidmonPollingType sgt = new VidmonPollingType();
        sgt.setVidmonName("cmm-7604-sd2.dns-sj.cisco.com");
        sgplt.getVidmons().add(sgt);

        JAXBElement<VidmonPollingListType> req =
objf.createMulticastVidmonPollingDeleteRequest(sgplt);
        JAXBElement<BulkMessageReturnWS> res = (JAXBElement<BulkMessageReturnWS>)
getWebServiceTemplate().marshalSendAndReceive(req);

        BulkMessageReturnWS ret = res.getValue();
        System.out.println("=====");
        System.out.println("deleteVidmonPolling");
        System.out.println("=====");

        System.out.println("status = " + ret.getReturnStatus());

        for (int i = 0; i < ret.getFailureList().size(); i++) {
            System.out.println("failure item = " +
ret.getFailureList().get(i).toString());
        }

        for (int i = 0; i < ret.getSuccessList().size(); i++) {
            System.out.println("success item = " +
ret.getSuccessList().get(i).toString());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Get Interface Polling Configuration

Use this operation to retrieve the interface configuration for a specified domain.

Request Message

JAXBElement<string>

string—Specifies the domain name.

Response Message

The response message contains a list of the interfaces that have been configured for polling.

JAXBElement<IntPollingListType>

- *IntPollingListType*—A JAXB-generated class that encapsulates an array list of *IntPollingType*.

- *IntPollingType*—A JAXB-generated wrapper class that encapsulates an interface name, interface index, bandwidth, router name, polling direction, high PPS threshold and low PPS threshold for polling of the interface.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllIntPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<IntPollingListType> sgresp = (JAXBElement<IntPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllIntPollingRequest(
dt));

        IntPollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting Interface Polling Configuration for domain " +
dt);

        System.out.println("=====");

        for (int i = 0; i < sgt.getInterfaces().size(); i++) {
            IntPollingType sg = sgt.getInterfaces().get(i);
            System.out.println(sg.getRouterName() + "\t" + sg.getInterfaceName() +
"\t" + sg.getIfindex() + "\t" + sg.getDirection() + "\t" + sg.getHighThreshold() + "\t"
+ sg.getLowThreshold() + "\t" + sg.getBandwidth());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
```

Get Layer 2 Polling Configuration

Use this operation to retrieve the Layer 2 polling configuration for a specified domain.

Request Message

JAXBElement<*string*>

string—Specifies the domain name.

Response Message:

AXBELEMENT<*L2PollingConfigType*>

- *L2PollingConfigType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address or hostname, device type enumeration, polling direction, high PPS threshold and low PPS threshold.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllL2PollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<L2PollingListType> sgresp = (JAXBElement<L2PollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllL2PollingRequest(d
t));

        L2PollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting L2 Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getL2().size(); i++) {
            L2PollingType sg = sgt.getL2().get(i);
            System.out.println(sg.getSwitchName() + "\t" + sg.getPortName() + "\t" +
sg.getPortIfIndex() + "\t" +sg.getDirection()+ "\t" +sg.getHighThreshold()+ "\t"
+sg.getLowThreshold());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
```

Get Multicast Trace Image File

Description

Use this operation to get a multicast trace image file from the CMM server. You can use this operation to create a current snapshot of the multicast tree for S,G or *.G and uploading of the image file in PNG format.

Request Message

JAXBElement<*multicastTraceType*>

multicastTraceType—A JAXB-generated class that encapsulates source, group, first hop router, last hop router, domain name, and *serviceType*.

Response Message

JAXBElement<*ImageType*>

ImageType —A JAXB-generated class that encapsulates binary data for uploading an image file from the server to the client application. The binary data in the message contains the trace image in MTOM binary format.

SOAP Errors

SOAP fault describing error.

Code Example:

```
public void getMulticastTrace()
```

```

{
    System.out.println("getMulticastTrace : ");

    MulticastTraceType mt = new MulticastTraceType();

    mt.setMulticastSource("126.32.193.29");
    mt.setMulticastGroup("239.254.1.12");
    mt.setFhr("SOURCE");
    mt.setLhr("crs1.cisco.com");
    mt.setDomainName("test-01");

    try
    {
        JAXBElement<ImageType> sgresp =
            (JAXBElement<ImageType>) getWebServiceTemplate().marshalSendAndReceive(
objf.createMulticastTraceRequest(mt));

ImageType        im = sgresp.getValue();
BufferedImage bim = (BufferedImage) im.getImage();

        System.out.println("Image Name is : " + im.getName());

        File f = new File("c:/tmp/cmmJaxBWTClientTrace.png");

        ImageIO.write(bim, "png", f);
    }
    catch (SoapFaultClientException ex)
    {
        System.out.println("getMulticastTrace : SOAP Fault Code:" +
ex.getFaultCode().getLocalPart());
        System.out.println("getMulticastTrace : SOAP Fault String:" +
ex.getFaultStringOrReason());
    }
    catch (Exception e)
    {
        System.out.println("getMulticastTrace : Exception : " +
e.getLocalizedMessage());
    }
}
}

```

Get Multicast Devices in a Domain

Description

Use this operation to get a list of all devices in a specified domain.

Request Message

JAXBElement<*string*>

string—Specifies the domain name.

Response Message

JAXBElement<*DeviceListType*>

- *DeviceListType*—A JAXB-generated class that encapsulates an array list of *DeviceType*.
- *DeviceType*—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model, and device name.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
public void getMulticastDevices() {
    String dt = "test-01";
    try {
        JAXBElement<DeviceListType> sgresp = (JAXBElement<DeviceListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastDevicesRequest(dt));
        DeviceListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting Multicast devices for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getDevice().size(); i++) {
            DeviceType sg = sgt.getDevice().get(i);
            System.out.println("Name " + sg.getName() + " ip " + sg.getIp());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
```

Get Multicast Events

Description

Use this operation to initiate a query about events generated by CMM.

Request Message

JAXBElement<*EventsQueryType*>

EventsQueryType—A class that encapsulates query parameter for filtering events. It encapsulates a number of query parameters such as the domain name, device IP address, and source and group IP address, which are used to filter the results of the query.

Response Message

JAXBElement<*EventListType*>

- *EventListType*—A JAXB-generated class that encapsulates an array list of *CmmEventType*.
- *CmmEventType*—A JAXB-generated class that encapsulates an event ID, *eventType*, date, device name (if applicable), and event details (as seen on the CMM GUI).

Example Code

```
public void getMulticastEventsList() {
    EventsQueryType et=objf.createEventsQueryType();
    et.setDomainName("test-01");
    et.setSource("126.32.193.29");
    et.setGroup("239.254.1.12");
    et.setDeviceIp("126.1.4.12");
}
```

```

//et.setDeviceName("cmm-7604-sd2.dns-sj.cisco.com");
et.setEventType(EventTypeEnumeration.VIDMON_EVENTS);
Calendar calendar1 = new GregorianCalendar();
calendar1.set(2008,11,1,1,1,1);

Calendar calendar2 = new GregorianCalendar();
Date endtTime = new Date();
calendar2.setTime(endtTime);
et.setStartTime(calendar1);
//et.setEndTime(calendar2);
et.setMaxNumberOfEvents(19);

//Date startTime = new Date();
//calendar.setTime(startTime);
//et.setStartTime(calendar);
try{
    JAXBElement<EventListType> sgresp
=(JAXBElement<EventListType>)getWebServiceTemplate().marshalSendAndReceive(objf.createMult
icastEventsRequest(et));
    EventListType sgt = sgresp.getValue();
    System.out.println("=====");
    System.out.println("Getting events for domain test");
    System.out.println("=====");

    for(int i=0;i<sgt.getEvents().size();i++)
    {
        CmmEventType sg = sgt.getEvents().get(i);
        System.out.println("DeviceName "+sg.getDeviceName()+ " type "+sg.getType()+
details: "+sg.getDetails());
    }

}
catch(SoapFaultClientException ex) {
    System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
    System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
}
}

```

Get RP Polling Configuration

Use this operation to retrieve Rendezvous Point (RP) polling configuration for a specified domain.

Request Message

JAXBElement<*string*>

string—Specifies the domain name.

Response Message:

JAXBElement<*RPPollingListType*>

- *RPPollingListType*—A JAXB-generated class that indicates the domain name and encapsulates an array list of *RPPollingType*.
- *RPPollingType*—A JAXB-generated wrapper class that encapsulates a device IP address or hostname, and a group limit for monitoring.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllRPPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<RPPollingListType> sgresp = (JAXBElement<RPPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllRPPollingRequest(d
t));

        RPPollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting RP Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getRppoll().size(); i++) {
            RPPollingType sg = sgt.getRppoll().get(i);
            System.out.println(sg.getRouter() + "\t" + sg.getGroupCount());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```

Get RPF Polling Configuration

Use this operation to add Reverse Path Forwarding (RPF) polling configuration for a specified source and group on a router.

Request Message

JAXBElement<string>

string—Specifies the domain name.

Response Message:

JAXBElement<RPPollingListType>

- *RPPollingListType*—A JAXB-generated class that indicates the domain name and encapsulates an array list of *RPPollingType*.
- *RPPollingType*—A JAXB-generated wrapper class that encapsulates a device IP address or hostname, and a group limit for monitoring.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllRPFPPollingConfigs() {
    String dt = "test-01";
```

```

    try {
        JAXBElement<RPFPollingListType> sgresp = (JAXBElement<RPFPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllRPFPollingRequest(
dt));

        RPFPollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting RPF Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getRpf().size(); i++) {
            RPFPollingType sg = sgt.getRpf().get(i);
            System.out.println(sg.getRouterName() + "\t" + sg.getGroupIp() + "\t" +
sg.getSourceIp() + "\t" + sg.getDelta());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}

```

Get S,G on a Device

Description

Use this operation to query a layer three device about currently active S,G on the device.

Request Message

JAXBElement< *DeviceType* >

DeviceType—A JAXB-generated wrapper class that encapsulates a domain name, device IP address, device type enumeration, model, and device name.

Response Message

JAXBElement<*SGListType*>

- *SGListType*—A JAXB-generated class that encapsulates an array list of *sourceGroupType*.
- *sourceGroupType*—A JAXB-generated class that encapsulates a string type for a source (source IP address) and for a group (group IP address).

SOAP Errors

A SOAP fault describing the error.

Code Example

```

public void getDeviceSg() {
    DeviceType dt = objf.createDeviceType();
    dt.setDomainName("test-01");
    dt.setIp("172.20.111.198");
    dt.setModel("7600");
    dt.setName("cmm-7206-sd1");

    try{

```

```

    JAXBElement<SGListType> sgresp
=(JAXBElement<SGListType>)getWebServiceTemplate().marshalSendAndReceive(objf.createMultica
stDeviceSGRequest(dt));
    SGListType sgt = sgresp.getValue();
    System.out.println("=====");
    System.out.println("Getting device SG list of " + dt.getIp());
    System.out.println("=====");

    for(int i=0;i<sgt.getSg().size();i++)
    {
    SourceGroupType sg = sgt.getSg().get(i);
    System.out.println("source "+sg.getSource()+ " Group "+sg.getGroup());
    }

    }
    catch(SoapFaultClientException ex) {
    System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
    System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
    }

    }

    /*public void getMulticastTraceSoapFault() {

    MulticastTraceType mt= new MulticastTraceType();
    mt.setMulticastSource("10.10.10.20");
    //mt.setMulticastGroup("239.20.20.20");
    mt.setMulticastGroup("");
    mt.setFhr("30.30.30.30");
    mt.setLhr("10.20.10.20");
    mt.setSeed("multicastTrace");

    try{
    JAXBElement<ImageType> sgresp
=(JAXBElement<ImageType>)getWebServiceTemplate().marshalSendAndReceive(objf.createMulticas
tTraceRequest(mt));
    ImageType im = sgresp.getValue();
    BufferedImage bim =(BufferedImage) im.getImage();
    System.out.println("Image Name is :"+im.getName());
    File f = new File("c:/cmmJaxBWTClientTrace.png");
    ImageIO.write(bim, "png", f);
    }
    catch(SoapFaultClientException ex) {
    //System.out.println("SOAP Fault Code " + ex.getFault().getFaultCodeAsQName());
    //System.out.println("SOAP Fault String:"+ ex.getFault().getFaultString());
    System.out.println("SOAP Fault Code:"+ ex.getFaultCode().getLocalPart() );
    System.out.println("SOAP Fault String:"+ ex.getFaultStringOrReason() );
    }
    catch(Exception e) {
    }

    }*/

```

Get S,G in a Domain

Description

Use this operation to get the S,G in a domain. This API operation initiates a query about available multicast streams in a particular domain.

Request Message

JAXBElement<*string*>

string—A string that specifies the domain name.

Response Message

JAXBElement<*SGListType*>

- *SGListType*—A JAXB-generated class that encapsulates an array list of *sourceGroupType*.
- *sourceGroupType*—A JAXB-generated class that encapsulates a string type for a source (source IP address) and for a group (group IP address).

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// multicastDomainSGRequest
public void getDomainSg() {
    String dt = "test-01";
    try {
        JAXBElement<SGListType> sgresp = (JAXBElement<SGListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastDomainSGRequest(dt));
        SGListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting domain SG list");
        System.out.println("=====");

        for (int i = 0; i < sgt.getSg().size(); i++) {
            SourceGroupType sg = sgt.getSg().get(i);
            System.out.println("source " + sg.getSource() + " Group " +
sg.getGroup());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
```

Get S,G Polling Configuration

Use this operation to retrieve the S,G polling configuration for a specified domain.

Request Message

JAXBElement<*string*>

string—A string that specifies the domain name.

Response Message:

- *SGPollingListType*—A JAXB-generated wrapper class that encapsulates a domain name and array list of *SGPollingType*.
- *SGPollingType*—A JAXB-generated class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, and a low threshold.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllSGPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<SGPollingListType> sgresp = (JAXBElement<SGPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllSGPollingRequest(d
t));

        SGPollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting SG Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getSg().size(); i++) {
            SGPollingType sg = sgt.getSg().get(i);
            System.out.println(sg.getSourceIp() + "\t" + sg.getGroupIp() + "\t" +
sg.getRouterName()+ "\t" +sg.getUnits()+ "\t" +sg.getHigh()+ "\t" +sg.getLow());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```

Get S,G Time-Based Polling Configuration

Use this operation for to retrieve the S,G time-based polling configuration for a specified domain.

Request Message

JAXBElement<string>

string—A string that specifies the domain name.

Response Message:

- *SGTimePollingListType*—A JAXB-generated class that encapsulates an a domain name and an array list of *SGTimePollingType*.
- *SGTimePollingType*—A JAXB-generated wrapper class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, a low threshold, a start time hour and minutes, stop time hour and minutes, and a day value.

- The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllSGTimePollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<SGTimePollingListType> sgresp =
        (JAXBElement<SGTimePollingListType>)
        getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllSGTimePollingReque
        st(dt));

        SGTimePollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting sgtime poll for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getSgtime().size(); i++) {
            SGTimePollingType sg = sgt.getSgtime().get(i);
            System.out.println(sg.getSource() + "\t" + sg.getGroup() + "\t" +
            sg.getRouter() + "\t" + sg.getUnits() + "\t" + sg.getHighThreshold() + "\t" +
            sg.getLowThreshold() + "\t" + sg.getStartTimeHur() + "\t" + sg.getStartTimeMin() + "\t" +
            sg.getStopTimeHur() + "\t" + sg.getStopTimeMin());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}

public void getAllL2PollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<L2PollingListType> sgresp = (JAXBElement<L2PollingListType>)
        getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllL2PollingRequest(d
        t));

        L2PollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting L2 Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getL2().size(); i++) {
            L2PollingType sg = sgt.getL2().get(i);
            System.out.println(sg.getSwitchName() + "\t" + sg.getPortName() + "\t" +
            sg.getPortIfIndex() + "\t" + sg.getDirection() + "\t" + sg.getHighThreshold() + "\t"
            + sg.getLowThreshold());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```


Get SSM Polling Configuration

Use this operation to retrieve the Source Specific Multicast (SSM) polling configuration for a specified domain.

Request Message

JAXBElement<string>

string—A string that specifies the domain name.

Response Message:

- *SSMPollingConfigType*—A JAXB-generated class that encapsulates an array list of *SSMPollingType*.
- *SSMPollingType*—A JAXB-generated wrapper class that encapsulates a source IP address, group IP address, units value specifying the units for monitoring, a high threshold, a low threshold, a start time hour and minutes, stop time hour and minutes, and a day value.
 - The *setUnits* value specifies the type units for the high and low threshold. To monitor packets per second, specify 0; to monitor bits per second, specify 1.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllSSMPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<SSMPollingListType> ssmresp = (JAXBElement<SSMPollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllSSMPollingRequest(
dt));

        SSMPollingListType ssmt = ssmresp.getValue();
        System.out.println("=====");
        System.out.println("Getting SSM Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < ssmt.getSsm().size(); i++) {
            SSMPollingType ssm = ssmt.getSsm().get(i);
            System.out.println(ssm.getGroupIp() + "\t" + ssm.getSourceIp() + "\t"
+ssm.getDays());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
```

Get Tree Polling Configuration

Use this operation to retrieve the tree polling configuration for a specified domain.

Request MessageJAXBElement<*string*>*string*—A string that specifies the domain name.**Response Message:**JAXBElement<*TreePollingListType*>

- *TreePollingListType*—A JAXB-generated class that contains a domain name and a *TreePollingType* object..
- *TreePollingType*—A JAXB-generated wrapper class that encapsulates a saved baseline and indicates whether it is to be used as a comparison baseline.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void getAllTreePollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<TreePollingListType> sgresp = (JAXBElement<TreePollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllTreePollingRequest
(dt));

        TreePollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting Tree Polling Configuration for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getTree().size(); i++) {
            TreePollingType sg = sgt.getTree().get(i);
            System.out.println(sg.getBaseLine() + "\t" + sg.isIsCompareBaseLine());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```

Get Video Probe Polling Configuration

Use this operation to retrieve the video probe polling configuration for a specified domain.

Request MessageJAXBElement<*string*>*string*—A string that specifies the domain name.**Response Message:**JAXBElement<*VideoProbePollingListType*>

- *VideoProbePollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VideoProbePollingType*.
- *VideoProbePollingType*—A JAXB-generated wrapper class that encapsulates a video probe name, delay Factor (DF) threshold in milliseconds, and Media Loss Rate (MLR) value.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample

public void getAllVideoProbePollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<VideoProbePollingListType> sgresp =
(JAXBElement<VideoProbePollingListType>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllVideoProbePollingR
equest(dt));
        VideoProbePollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting Video Probe Configuratin for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getVideoProbes().size(); i++) {
            VideoProbePollingType sg = sgt.getVideoProbes().get(i);
            System.out.println(sg.getProbeName() + "\t" + sg.getDf() + "\t" +
sg.getLoss());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```

Get VidMon Polling Configuration

Use this operation to retrieve the VidMon polling configuration for a specified domain.

Request Message

JAXBElement<string>

string—A string that specifies the domain name.

Response Message:

JAXBElement<VidmonPollingListType>

- *VidmonPollingListType*—A JAXB-generated class that encapsulates a domain name and an array list of *VidmonPollingType*.
- *VidmonPollingType*—A JAXB-generated wrapper class that encapsulates a device name, VidMon type, delay factor (DF) in milliseconds, a Media Loss Rate (MLR) threshold value (number of packets), a milli-percentage value that specifies a MRV maximum threshold., and a milli-percentage value that specifies a MRV minimum threshold.

**Note**

The *setVidmonType* field in the *VidmonPollingType* class indicates the type of VidMon device for which you are adding polling. If the device is a Cisco 76xx series router, specify 1. If the device is an ASR 9000 series device, specify 0. Only 76xx devices support MLR monitoring; therefore you should only specify an MLR value in the setMlr call if the device is a 76xx device.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample

public void getAllVidmonPollingConfigs() {
    String dt = "test-01";
    try {
        JAXBElement<VidmonPollingListType> sgresp =
        (JAXBElement<VidmonPollingListType>)
        getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastGetAllVidmonPollingRequest(dt));

        VidmonPollingListType sgt = sgresp.getValue();
        System.out.println("=====");
        System.out.println("Getting vidmon polling configurations for domain " + dt);
        System.out.println("=====");

        for (int i = 0; i < sgt.getVidmons().size(); i++) {
            VidmonPollingType sg = sgt.getVidmons().get(i);
            System.out.println(sg.getVidmonName() + "\t" + sg.getVidmonType() + "\t" +
            sg.getDf() + "\t" + sg.getMlr() + "\t" + sg.getMrvMax() + "\t" + sg.getMrvMin());
        }

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}
}
```

Set Global Polling Configuration

Use this operation to set the global polling configuration for a specified domain.

Request Message

JAXBElement<*GlobalPollingListType*>

- *GlobalPollingListType*—A JAXB-generated class that encapsulates an array list of *GlobalPollingTypeWS*.
- *GlobalPollingTypeWS*—A JAXB-generated wrapper class that encapsulates global polling configuration settings: including
 - Arguments that specify the polling time and interval, such as the days and times when polling is to be active.
 - Arguments to set designated router (DR) polling, threshold polling, and heartbeat polling.

- An array list that specifies global default e-mail addresses for event notification
- An array list of *GlobalPollingTrap* elements that specifies the IP addresses to poll for SNMP traps.
- Values to set the trap repeat value for Rising and Falling traps and enable them
- A value to enable forwarding of Mixed Signals traps
- A value to set the Video Clear Timer

Response Message:

The response message contains an integer indicating the result of the operation.

SOAP Errors

A SOAP fault describing the error.

Example Code

```
// Code Sample
public void setGlobalPollingConfigs() {
    try {
        GlobalPollingListType gplt = new GlobalPollingListType();

        GlobalPollingTypeWS gpt = new GlobalPollingTypeWS();
        gpt.setDays("M-F"); // Sun, Mon, Tue, Wed, Thu, Fri, Sat, M-F, Everyday
        gpt.setHurMinSec("60"); // HR:3600, MIN:60, SEC:1
        gpt.setInterval(301);
        gpt.setStartTimeHur(2);
        gpt.setStartTimeMin(3);
        gpt.setStopTimeHur(22);
        gpt.setStopTimeMin(40);

        gplt.setDrPolling(gpt);
        gplt.setThresholdPolling(gpt);
        gplt.setHeartBeatPolling(gpt);

        List<String> emails = new ArrayList();
        emails.add("test@test.com");
        gplt.getEmailAddress().addAll(emails);

        List<GlobalPollingTrap> traps = new ArrayList();
        GlobalPollingTrap trap = new GlobalPollingTrap();

        trap = new GlobalPollingTrap();
        trap.setPort(0);
        trap.setTrapReceiverAddress("172.20.111.240");
        traps.add(trap);

        gplt.getTrapReceivers().addAll(traps);

        gplt.setTrapRepeat(1);
        gplt.setIsRisingFalling(false);
        gplt.setIsFwdMixedSignalTraps(false);
        gplt.setVideoProbeClearTimerHur(0);

        JAXBElement<Integer> ret = (JAXBElement<Integer>)
getWebServiceTemplate().marshalSendAndReceive(objf.createMulticastSetGlobalPollingRequest(
gplt));

        int val = ret.getValue();

        System.out.println("Global Polling status " + val);
    }
}
```

```

    } catch (SoapFaultClientException ex) {
        System.out.println("SOAP Fault Code:" + ex.getFaultCode().getLocalPart());
        System.out.println("SOAP Fault String:" + ex.getFaultStringOrReason());
    }
}

```

Spring Framework Beans

The CMM SDK includes a number of Java beans that are generated using the Spring Framework:

- [Message Factory Bean, page 2-46](#)
- [JAXB Marshaller Bean, page 2-46](#)
- [Security Interceptor Bean, page 2-47](#)

Message Factory Bean

This bean is used by superclasses of *CmmWebServicesClient* to initialize the message factory object for using SAAJ SOAP messages.

```
<bean id="messagFactory" class="org.springframework.ws.soap.saaj.SaajSoapMessageFactory"/>
```

JAXB Marshaller Bean

This bean is used for marshalling and un-marshalling SOAP messages from XML format to Java objects or from Java objects to XML format.

```

<bean id="marshaller" class="org.springframework.oxm.jaxb.Jaxb2Marshaller">
    <property name="contextPath" value="com.cisco.nm.cmm.ws.client.jaxws"/>
    <property name="mtomEnabled" value="true"/>
</bean>

```

This is the main bean in the application. The main method gets an instance of this bean and starts issuing API operations. This bean takes the message factory bean, the marshaller bean, and the default URI as parameters. In the sample code, the service URL is supplied through the *build.xml* file.

```

<bean id="httpClient" class="com.cisco.nm.cmm.ws.client.jaxws.CmmWebServicesClient">
    <constructor-arg ref="messagFactory"/>
    <property name="marshaller" ref="marshaller"/>
    <property name="unmarshaller" ref="marshaller"/>
    <property name="defaultUri" value="http://localhost:8082/cmm/service/cmmNB.wsdl"/>
    <property name="interceptors" ref="securityInterceptor"/>
</bean>

```

Security Interceptor Bean

This bean lets you define WS-Security policies. In the current implementation of the SDK, we use the user name *admin* and the password *rsmmt* for authenticating WS messages.

```
<bean id="securityInterceptor"
class="org.springframework.ws.soap.security.wss4j.Wss4jSecurityInterceptor">
  <property name="securementActions" value="UsernameToken"/>
  <property name="securementUsername" value="admin"/>
  <property name="securementPassword" value="rsmmt"/>
  <property name="securementPasswordType" value="PasswordText"/>
</bean>
```
