

Sample Client Programs to Use Web Services

This appendix describes the sample client programs that use Cisco DCNM-SAN Web Services. Cisco DCNM-SAN supports the following web services.

- [SOAP Web Services](#)
- [REST Web Services](#)

SOAP Web Services

The following client programs use SOAP web services in Cisco DCNM.

- [Java Client](#)
- [Perl Client, page A-16](#)

Java Client

This section provides an example of the Java client that uses the SOAP web services.

Example A-1 Zone Manager Web Services API

```
package com.cisco.dcbu.smis.jaxws.test;

import static com.cisco.dcbu.smis.jaxws.test.ZoneWsUtil.*;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import com.cisco.dcbu.smis.jaxws.ep.Fabric;
import com.cisco.dcbu.smis.jaxws.ep.OperationStatus;
import com.cisco.dcbu.smis.jaxws.ep.Vsan;
import com.cisco.dcbu.smis.jaxws.ep.WwnKey;
import com.cisco.dcbu.smis.jaxws.ep.Zone;
import com.cisco.dcbu.smis.jaxws.ep.ZoneMember;
import com.cisco.dcbu.smis.jaxws.ep.ZoneSet;
import com.cisco.dcbu.smis.jaxws.ep._switch;
public class ZoneWsClient {

    /**
     * Default constructor
     */
    public ZoneWsClient() {
```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        input_ = new BufferedReader(new InputStreamReader(System.in));
    }

    /**
     * Main method is the starting point of execution.
     * @param args
     * @throws Exception
     */
    public static void main(String[] args) throws Exception {
        menu();
    }

    /**
     * This method show all available zone webservice as user menu.
     * @throws Exception
     */
    public static void menu() throws Exception {
        ZoneWsClient zwsClient = new ZoneWsClient();
        int choice = 0;

        for (;;) {
            header("=====Zone WS
Client=====");
            System.out.println("Available Zone services:\n\n " + "\t1:Discover Zones in
fabric\n " + "\t2:Discover Zonesets in fabric\n " + "\t3:Discover Zonemenbers in fabric\n
"
                + "\t4:Create new Zone in fabric\n " + "\t5:Create new ZoneSet in
fabric\n " + "\t6:Add Zonemember to existing Zone in fabric\n " + "\t7:Add existing Zone
to existing Zoneset in fabric\n "
                + "\t8:Activate/Deactivate a ZoneSet in fabric\n " + "\t9:Manual Zone
Cache Invalidation\n " + "\t10:Exit ");
            System.out.println("\n\tPlease enter your Choice: ");
            try {
                choice = Integer.parseInt(input_.readLine());
            } catch (NumberFormatException e) {
                System.out.println("Enter Integer choices only...");
                continue;
            }

            switch (choice) {
            case 1:
                zwsClient.displayZone();
                break;
            case 2:
                zwsClient.displayZonesets();
                break;
            case 3:
                zwsClient.displayZonemembers();
                break;
            case 4:
                zwsClient.createZone();
                break;
            case 5:
                zwsClient.createZoneSet();
                break;
            case 6:
                zwsClient.addZoneMember();
                break;
            case 7:
                zwsClient.addZone();
                break;
            case 8:
                zwsClient.activateAndDeactivateZoneSet();
                trailer("ACTIVATE/DEACTIVATE ZONESET");
            }
        }
    }

```

```

        break;
    case 9:
        zwsClient.zoneCacheInvalidation();
        break;
    case 10:
        System.out.println("Exiting ZoneWSClient..");
        trailer("Zone WS CLIENT");
        System.exit(1);
    default:
        System.out.println("Wrong Choice entered.");
    }
}

}

/**
 * This method invalidates zonedata cache by making use of
 * ZoneManagerWS: invalidateZoneDataCache()
 */
public void zoneCacheInvalidation() {
    try {
        System.out.println("Invalidation of Zone data cache started");
        zoneManager.invalidateZoneDataCache();
        System.out.println("Succesfully Invalidated Zone data cache");
    } catch (Exception e) {
        exception(e);
    } finally {
        trailer("Zone Cache Invalidation");
    }
}

/**
 * This method takes prompts user to enter already existing zoneset name to
 * activate/deactivate.
 * ZoneManagerWS: ActivateZoneset()
 * ZoneManagerWS: deActivateZoneset()
 */
public void activateAndDeactivateZoneSet() throws Exception {
    int flag = 0, choice = 0;
    do {
        try {
            System.out.println("Enter your choice\n 1:Activate a ZoneSet\n 2:Deactivate a
ZoneSet\n 3:Main menu");
            choice = Integer.parseInt(input_.readLine());
            OperationStatus oper = null;
            if (choice == 3) return;
            if (choice != 1 && choice != 2 && choice != 3) {
                System.out.println("Enter right Choice: 1 or 2 or 3");
                flag = 0;
                continue;
            }
            oper = activation(choice);
            if (oper != null) {
                System.out.println("Zoneset activation/deactivation is done successfully.\n
Please wait for 1-5 minutes to get reflected.");
                flag = 1;
            }
            else if (oper == null) {
                System.out.println("Zoneset activation/deactivation is not successfull");
                flag = 1;
            }
        } catch (NumberFormatException ne) {
            break;
        }
    } while (flag == 0);
}

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        System.out.println("NumberFormatException: Integer type expected.");
        continue;
    }
    catch (Exception e) {
        exception(e);
        flag=0;
        continue;
    }
    }while(choice !=3||flag==0);
}

private OperationStatus activation(int choice) throws Exception {
    OperationStatus oper = null;
    System.out.println("Enter the ZoneSet name to be activated/deactivated ");
    String zName = input_.readLine();
    Fabric[] fabrics = getFabrics();
    if(fabrics!=null){
        for (Fabric fab : fabrics) {
            _switch[] sws = getSwitchesByFabric(fab);
            Vsan[] vsans = getVsans(fab);
            if(sws!=null){
                for (_switch sw : sws) {
                    if(vsans!=null){
                        for (Vsan vsan : vsans) {
                            ZoneSet enfZoneset = zoneManager.getEnfZoneSetDO(vsan.getKey());
                            ZoneSet[] zoneSets = zoneManager.getZoneSets(vsan.getKey(),new
WwnKey(sw.getWwn()));
                            if(zoneSets==null){
                                System.out.println("Zoneset is null for VSan
"+vsan.getKey().getVsanID());
                                continue;
                            }
                            for (ZoneSet zs : zoneSets) {
                                if (enfZoneset != null) {
                                    if (enfZoneset.getName().equals(zName) && choice == 1 &&
enfZoneset.isActive()) {
                                        System.out.println("Selected ZoneSet is already activated");
                                        return null;
                                    }
                                    else if (!enfZoneset.getName().equals(zName) && choice == 2 &&
zs.getName().equals(zName)) {
                                        System.out.println("Selected ZoneSet is already
deactivated");
                                        return null;
                                    }
                                } else if (enfZoneset == null && choice == 2) {
                                    System.out.println("No ZoneSet is active to deactivate");
                                    return null;
                                } else if (zs.getName().equals(zName) && choice == 2 &&
!zs.isActive()) {
                                    System.out.println("Selected ZoneSet is already deactivated");
                                    return null;
                                }
                                if (zs.getName().equals(zName)) {
                                    if (choice == 1)
                                        oper = zoneManager.activateZoneset(vsan.getKey(), new
WwnKey(sw.getWwn()), zs.getName());
                                    else if (choice == 2)
                                        oper = zoneManager.deActivateZoneset(vsan.getKey(), new
WwnKey(sw.getWwn()), zs.getName());
                                    return oper;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
}

    System.out.println("Invalid Zoneset is selected");
}
}}
return oper;

}

/**
 * This method prompts user to enter a unique zone name and creates the
 * zone in default vsan.
 * ZoneManagerWS:createZone()
 */
public void createZone() {
    try {
        String zName=null;
        do{
            System.out.println("Enter the new Zone name");
            zName= input_.readLine();
        }while(zName.length()<1);
        Fabric[] fabrics = getFabrics();
        if(fabrics!=null){
            for (Fabric fab : fabrics) {
                _switch[] sws = getSwitchesByFabric(fab);
                Vsan[] vsans = getVsans(fab);
                if(sws!=null){
                    for (_switch sw : sws) {
                        if(vsans!=null){
                            for (Vsan vsan : vsans) {
                                int qosPriority = -1;
                                boolean readOnly = false;
                                boolean broadcast = false;
                                boolean qos = false;
                                Zone zone1 = zoneManager.createZone(vsan.getKey(),new
WwnKey(sw.getWwn()), zName, readOnly, broadcast, qos, qosPriority);
                                if (zone1 == null)
                                    System.out.println("Failed to create Zone: " + zName);
                                else
                                    System.out.println("Zone " + zName + " created successfully in
Vsan:" + vsan.getKey().getVsanID() + ",Switch: " + sw.getName());
                                return;
                            }
                        }
                    }
                }
            }
        }
    } catch (Exception e) {
        exception(e);
    }finally{
        trailer("CREATE ZONE");
    }
}

/**
 * This method prompts user to enter an existing zone,zonememberid and zonemembertype
 and creates a zonemember in zone provided.
 * ZoneManagerWS:addZoneMemberToZone
 */
public void addZoneMember() {
    try {

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        for (;;) {
            System.out.println("\nEnter existing zonename to which member has to be
added\n");
            String zoneName = input_.readLine();
            Fabric[] fabrics = getFabrics();
            if(fabrics!=null){
                for (Fabric fab : fabrics) {
                    _switch[] sws = getSwitchesByFabric(fab);
                    Vsan[] vsans = getVsans(fab);
                    if(sws!=null){
                        for (_switch sw : sws) {
                            if(vsans!=null){
                                for (Vsan vsan : vsans) {
                                    Zone existingzones[] = zoneManager.getZones(vsan.getKey(), new
WwnKey(sw.getWwn()));
                                    if(existingzones==null){
                                        continue;
                                    }
                                    boolean flag=false;
                                    for (Zone zs : existingzones) {
                                        if (zs.getName().equals(zoneName)) {
                                            flag=true;
                                            System.out.println("Enter zonemember type which is in
number format. \n 1. For WWN ");
                                            int zonememtype = Integer.parseInt(input_.readLine());
                                            String hexstr = null;
                                            byte[] zonememberid = null;
                                            if(zonememtype==1){
                                                System.out.println("Enter zonemember id which is in
hexa format Eg: 1122334455667788 or 11:22:33:44:55:66:77:88");
                                                hexstr=input_.readLine();
                                                if (hexstr.contains(":"))
                                                    zonememberid = ZoneWsUtil.fromHexString(hexstr,
true);
                                                else
                                                    zonememberid = ZoneWsUtil.fromHexString(hexstr,
false);
                                            }
                                            else{
                                                System.out.println("Demo ZoneWsClient do not support
other member types\n");
                                                continue;
                                            }
                                            ZoneMember zoneMemberAdded =
zoneManager.addZoneMemberToZone(vsan.getKey(),new WwnKey(sw.getWwn()),zoneName,
zonememtype, -1,-1, -1, zonememberid, null);
                                            if (zoneMemberAdded == null)
                                                System.out.println("Failed to add Zonemember to
specified zone");
                                            else
                                                System.out.println("ZoneMember added succesfully to
zone "+ zoneName);
                                            return;
                                        }
                                    }
                                }
                            if(flag==false)
                                System.out.println("Zone " + zoneName + " not found. Please
enter an existing zone");
                            }}
                        }}
                    }
                }
            }
        }
    }
}

```

```

        } catch (NumberFormatException ne) {
            System.out.println("NumberFormatException: Please check the input
type/format");
        }
        catch (Exception e) {
            exception(e);
        }finally{
            trailer("ADD_ZONE");
        }
    }

    /**
     * This method prompts user to enter existing zoneset, existing zone
     * and adds zone to zoneset.
     * ZoneManagerWS:addZoneToZoneset()
     */
    public void addZone() {
        try {
            for (;;) {
                System.out.println("Enter already existing zoneset to which the zone has to
be added: ");
                String zonesetName = input_.readLine();
                Fabric[] fabrics = getFabrics();
                if(fabrics!=null){
                    for (Fabric fab : fabrics) {
                        _switch[] sws = getSwitchesByFabric(fab);
                        Vsan[] vsans = getVsans(fab);
                        if(sws!=null){
                            for (_switch sw : sws) {
                                if(vsans!=null){
                                    for (Vsan vsan : vsans) {
                                        ZoneSet existingzonesets[] =
zoneManager.getZoneSets(vsan.getKey(), new WwnKey(sw.getWwn()));
                                        for (ZoneSet zset : existingzonesets) {
                                            if (zset.getName().equals(zonesetName)) {
                                                System.out.println("Enter existing zone to be added to
zoneset: ");
                                                String zonename = input_.readLine();
                                                Zone existingzones[] =
zoneManager.getZones(vsan.getKey(),new WwnKey(sw.getWwn()));
                                                for (Zone zone : existingzones) {
                                                    if (zone.getName().equals(zonename)) {
                                                        ZoneSet znst =
zoneManager.addZoneToZoneset(vsan.getKey(),new WwnKey(sw.getWwn()),zset, zone);
                                                        if (znst == null)
                                                            System.out.println("Failed to add Zone to
specified Zoneset");
                                                        else
                                                            System.out.println("Zone "+ zonename+ " added
succesfully to zoneset "+ zonesetName);
                                                        return;
                                                    }
                                                }
                                            }
                                        }
                                        System.out.println("Zone "+ zonename+ " not found.
Please enter an existing zone");
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        System.out.println("ZoneSet " + zonesetName+ " not found. Please enter an
existing zoneset");
    }
    } catch (Exception e) {
        exception(e);
    }finally{
        trailer("ADD ZONE");
    }

}

/**
 * This method prompts user to enter zoneset name which will be created
 * in default vsan.
 * ZoneManagerWS:createZoneSet()
 */
public void createZoneSet() {
    try {
        System.out.println("Enter the new unique Zoneset name");
        String zonsetName =input_.readLine();
        Fabric[] fabrics = getFabrics();
        if(fabrics!=null){
            for (Fabric fab : fabrics) {
                _switch[] sws = getSwitchesByFabric(fab);
                Vsan[] vsans = getVsans(fab);
                if(sws!=null){
                    for (_switch sw : sws) {
                        if(vsans!=null){
                            for (Vsan vsan : vsans) {

                                ZoneSet newzoneset = zoneManager.createZoneSet(vsan.getKey(), new
WwnKey(sw.getWwn()), zonsetName);
                                if (newzoneset == null)
                                    System.out.println("Failed to create Zoneset: " + zonsetName);
                                else
                                    System.out.println("ZoneSet " + zonsetName+ " created in
Vsan:"+ vsan.getKey().getVsanID() + ",Switch: " + sw.getName());
                                return;
                            }
                        }
                    }
                }
            }
        }
    } catch (Exception e) {
        exception(e);
    }finally{
        trailer("CREATE ZONESET");
    }
}

/**
 * This method displays the Zone member setting data in the fabric.
 * ZoneManagerWS:getZoneMembers()
 */
public void displayZonemembers() {
    try {
        Fabric[] fabrics = getFabrics();
        if(fabrics!=null){
            for (Fabric fab : fabrics) {
                _switch[] sws = getSwitchesByFabric(fab);
                Vsan[] vsans = getVsans(fab);
                if(vsans!=null){
                    for (Vsan vsan : vsans) {
                        Zone[] enfZones = getEnfZones(vsan);
                        if (enfZones != null) {
                            for (Zone ezone : enfZones) {

```

```

        ZoneMember[] zms = getZoneMembers(ezone);
        for (ZoneMember zm : zms)
            System.out.println("vsan id:" + vsan.getKey().getVsanID() +
",vsan wwn:" + vsan.getKey().getPWwn().getValue() + ",Zone Name:" + ezone.getName()
+ ", ZoneMemberType: " + zm.getType() + ",
ZoneMemberId: " + ZoneWsUtil.toHexString(zm.getId()) + ",Status:" + ezone.isActive());
        }
    }
}
}
if(sws!=null){
for (_switch sw : sws) {
    if(vsans!=null){
        for (Vsan vsan : vsans) {
            Zone[] zones = getZones(vsan, sw);
            if (zones == null)
                return;
            for (Zone zone : zones) {
                ZoneMember[] zms = getZoneMembers(zone);
                for (ZoneMember zm : zms)
                    System.out.println("vsan id:" + vsan.getKey().getVsanID() +
",vsan wwn:" + vsan.getKey().getPWwn().getValue() + ",Zone Name:" + zone.getName()
+ ", ZoneMemberType: " + zm.getType() + ",
ZoneMemberId: " + ZoneWsUtil.toHexString(zm.getId()) + ",Status:" + zone.isActive());
            }
        }
    }
}
} catch (Exception e) {
    exception(e);
}finally{
    trailer("Zone Members");
}

}

private ZoneMember[] getZoneMembers(Zone zone) {
    header("Now getting ZoneMembers for Zone " + zone.getName());
    ZoneMember[] zoneMemz = null;
    try {
        zoneMemz = zone.getMembers();
        if (zoneMemz != null)
            System.out.println(zoneMemz.length+ " ZoneMembers found for the Zone " +
zone.getName());
        else
            System.out.println("No ZoneMembers found for the Zone " + zone.getName());
        return zoneMemz;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

/**
 * This method displays the active and local zonesets in the fabric.
 * ZoneManagerWS:getZoneSets()
 */
public void displayZonesets() {
    try {
        Fabric[] fabrics = getFabrics();
        if(fabrics!=null){
            for (Fabric fab : fabrics) {
                _switch[] sws = getSwitchesByFabric(fab);
                Vsan[] vsans = getVsans(fab);

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        if(vsans!=null){
            for (Vsan vsan : vsans) {
                getEnfZonesets(vsan);
            }
        }
        if(sws!=null){
            for (_switch sw : sws) {
                if(vsans!=null){
                    for (Vsan vsan : vsans) {
                        ZoneSet[] zoneSets = zoneManager.getZoneSets(vsan.getKey(), new
WwnKey(sw.getWwn()));
                        header("Now getting local ZoneSets for Vsan "+
vsan.getKey().getVsanID());
                        if (zoneSets == null) {
                            System.out.println("No Zoneset found for switch"+ sw.getName()
+ " Vsan "+ vsan.getKey().getVsanID());

                                }
                                else{
                                    System.out.println(zoneSets.length+ " Zonesets found for the switch
"+ sw.getName() + " Vsan "+ vsan.getKey().getVsanID());
                                    for (ZoneSet zoneset : zoneSets) {
                                        System.out.println("VsanId:"+ vsan.getKey().getVsanID() +
",Vsan WWN:"+ vsan.getKey().getPWwn().getValue()+ ",ZonesetName:" + zoneset.getName()+
",Status:" + zoneset.isActive());
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    } catch (Exception e) {
        exception(e);
    }finally{
        trailer("ZONESETS");
    }
}

private static Fabric[] getFabrics() {
    try {
        Fabric[] fabrics = san.getFabrics();
        if (fabrics == null) {
            System.out.println("fabrics is null" + fabrics.length);
            trailer("No Fabric - Return to Main Menu");
            menu();
        }
        return fabrics;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

private static _switch[] getSwitchesByFabric(Fabric fabric) {
    try {
        _switch[] sws = san.getSwitchesByFabric(fabric.getFabricKey());
        if (sws == null)
            System.out.println("No switches found for fabric");
        else {
            header(sws.length + " switch(es) found for fabric");
            for (_switch sw : sws) {
                System.out.println("Name:" + sw.getName() + ",IP:"+ sw.getIpAddress() +
",WWN:"+ sw.getWwn().getValue());
            }
        }
    }
}

```

```

        }
    }
    return sws;
} catch (Exception e) {
    exception(e);
}
return null;
}

private static Vsan[] getVsans(Fabric fabric) {
    try {
        Vsan[] vsans = san.getVsans(fabric.getFabricKey());
        if (vsans == null)
            System.out.println("No vsans found for fabric");
        else {
            header(vsans.length + " vsan(s) found for fabric");
            for (Vsan vsan : vsans) {
                System.out.println("VanID:" + vsan.getKey().getVsanID() + ",VsanWWN:" +
                    vsan.getKey().getPWwn().getValue());
            }
        }
        return vsans;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

/**
 * This method displays all the active and local zones in the fabric.
 * ZoneManagerWS:getEnfZones()
 * ZoneManagerWS:getZones()
 */
public void displayZone() throws Exception {

    Fabric[] fabrics=getFabrics();
    if(fabrics!=null){
        for (Fabric fabric : fabrics) {
            System.out.println("Discovered the Fabric with WWN:" +
                fabric.getSeedSwWwn().getValue());
            _switch[] sws = getSwitchesByFabric(fabric);
            Vsan[] vsans = getVsans(fabric);
            if(vsans!=null){
                for (Vsan vsan : vsans) {
                    Zone[] enfZones = getEnfZones(vsan);
                    if (enfZones != null) {
                        for (Zone ezone : enfZones) {
                            System.out.println("vsan id:"+ vsan.getKey().getVsanID() + ",vsan
                                wwn:"+ vsan.getKey().getPWwn().getValue() + ",Zone Name:" + ezone.getName() + ",Status:"+
                                    ezone.isActive());
                        }
                    }
                }
            }
            if(sws!=null){
                for (_switch sw : sws) {
                    if(vsans!=null){
                        for (Vsan vsan : vsans) {
                            Zone[] zones = getZones(vsan, sw);
                            if (zones == null) continue;
                            for (Zone zone : zones) {

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        System.out.println("vsan id:"+ vsan.getKey().getVsanID() + ",vsan
wwn:"+ vsan.getKey().getPwwn().getValue()+ ",Zone Name:" + zone.getName() + ",Status:"+
zone.isActive());
    }
    }}
    }}
    trailer("ZONES");
}

private Zone[] getEnfZones(Vsan vsan) throws Exception {
    header(" Getting active zones for Vsan " + vsan.getKey().getVsanID());
    Zone[] enfZones = null;
    try {
        enfZones = zoneManager.getEnfZoneSet(vsan.getKey());
        if (enfZones != null)
            System.out.println(enfZones.length+ " Active zones found for the Vsan "+
vsan.getKey().getVsanID());
        else
            System.out.println("No Active zones found");
        return enfZones;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

private ZoneSet getEnfZonesets(Vsan vsan) throws Exception {
    header("Now getting active ZoneSets for Vsan " + vsan.getKey().getVsanID());
    ZoneSet enfZoneset = null;
    try {
        enfZoneset = zoneManager.getEnfZoneSetDO(vsan.getKey());
        if (enfZoneset != null)
            System.out.println("Active zoneset for the VsanId: "+
vsan.getKey().getVsanID() + ", Pwwn: " + vsan.getKey().getPwwn().getValue()
+ ", ZonesetName: " + enfZoneset.getName()+ ", Status " +
enfZoneset.isActive());
        else
            System.out.println("Active zoneset is null");
        return enfZoneset;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

private static Zone[] getZones(Vsan vsan, _switch sw) {
    header("Now getting Local Zones for switch " + sw.getName() + " vsan "+
vsan.getKey().getVsanID());
    try {
        Zone[] zones = zoneManager.getZones(vsan.getKey(), new WwnKey(sw.getWwn()));
        if (zones != null)
            System.out.println("Found " + zones.length + " zones");
        else
            System.out.println("No zones found ");
        return zones;
    } catch (Exception e) {
        exception(e);
    }
    return null;
}

static {
    getCredentials();
}

```

```

        getLoginService();
        getLogin();
        getZoneManagerService();
        getSanService();
    }
}

public class ZoneWsUtil {

    public static String url = null;

    public static String username = null;

    public static String password = null;

    public static String token = null;

    public static Logon logon = null;

    public static San san = null;

    public static ZoneManager zoneManager = null;

    static BufferedReader input_ = null;

    static byte[] fromHexString(String hexStr, boolean hasColon)
        throws NumberFormatException {
        hexStr = hexStr.toLowerCase();
        int len = hexStr.length();
        byte bytes[] = new byte[hasColon ? ((len / 3) + 1) : (len / 2)];
        int sPos = 0; // position in hexStr
        int bPos = 0; // position in bytes
        try {
            while (sPos < len) {
                char a = hexStr.charAt(sPos);
                char b = hexStr.charAt(sPos + 1);
                if (hasColon && (a == ':' || b == ':'))
                    throw new NumberFormatException("bad Hex format");
                int v1 = Character.digit(a, 16);
                int v2 = Character.digit(b, 16);
                if (v1 < 0 || v2 < 0)
                    throw new NumberFormatException("bad Hex format");
                int v3 = (int) (v1 * 16 + v2);
                bytes[bPos] = (byte) v3;
                sPos += hasColon ? 3 : 2;
                bPos++;
            }
        } catch (Exception ex) {
            throw new NumberFormatException("bad Hex format");
        }

        if (bPos < bytes.length)
            throw new NumberFormatException("bad Hex format");
        return bytes;
    }

    static String toHexString(byte[] inputByte) {
        char[] HEX_DIGIT = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
            'a', 'b', 'c', 'd', 'e', 'f' };
        if (inputByte == null || inputByte.length == 0)
            return null;
        StringBuffer outputstr = new StringBuffer(inputByte.length * 3);
        for (int i = 0; i < inputByte.length; i++) {
            int n = (int) (inputByte[i] & 0xFF);
            outputstr.append(HEX_DIGIT[(n >> 4) & 0x0F]);

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        outputstr.append(HEX_DIGIT[n & 0x0F]);
        if (i + 1 < inputByte.length)
            outputstr.append(':');
    }
    return outputstr.toString();
}

static void header(String text) {
    System.out.println(" ");
    System.out.println(text);
}

static void trailer(String string) {
    System.out.println("\n***** END OF "
        + string + " ***** ");
}

/**
 * This static method is used read the login credentials from the properties
 * file named "configuration.properties"
 */
public static void getCredentials() {
    Properties props = new Properties();
    try {
        if(props!=null){
            System.out.println("Reading credentials from the 'configuration.properties'
file");
            props.load(new FileInputStream("configuration.properties"));
            url = props.getProperty("FMServer-ipadress");
            username = props.getProperty("UserName");
            password = props.getProperty("Password");
        }
    } catch (FileNotFoundException fe) {
        System.out.println("ZoneWsClient: 'configuration.properties' File Not
found!!");
        url="localhost";
        username="admin";
        password="cisco123";
    }
    catch (IOException e) {
        exception(e);
    }
}

/**
 * This method is to check LogonWSService is available or not. If available
 * bind to the stub.
 */
public static void getLoginService() {
    try {
        String loginServiceAddr = "http://" + url
            + "/LogonWSService/LogonWS";

        LogonServiceLocator logonService = new LogonServiceLocator();
        logonService.setEndpointAddress("LogonPort", loginServiceAddr);

        logon = logonService.getLogonPort();

        if (logon == null)
            System.out.println("login service not available" + logon);
        else
            System.out.println("Logon success.");
    }
}

```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * This method is to get login token.
     */
    public static void getLogin() {

        try {
            if(logon!=null){
                token = logon.requestToken(username, password, 1000000000L);
                if (token == null)
                    System.out.println("Token not found" + token);
                else {
                    System.out.println("Token found for configured user. ");
                }
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * This method is to check SanWSService is available or not. If available
     * bind to the stub.
     */
    public static void getZoneManagerService() {
        try {

            String zmServiceAddr = "http://" + url
                + "/ZoneManagerWSService/ZoneManagerWS";
            ZoneManagerServiceLocator zmService = new ZoneManagerServiceLocator();
            zmService.setEndpointAddress("ZoneManagerPort", zmServiceAddr);
            zoneManager = zmService.getZoneManagerPort();

            if (zoneManager == null) {
                System.out.println("zonemanager service not available");
                return;
            }
            SOAPHeaderElement hdrElement = setSoapHeader(token);
            ZoneManagerBindingStub zmStub = (ZoneManagerBindingStub) zoneManager;
            zmStub.setHeader(hdrElement);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * This method is to check ZoneManagerWSService is available or not. If
     * available bind to the stub.
     */
    public static void getSanService() {
        try {

            String sanServiceAddr = "http://" + url + "/SanWSService/SanWS";
            SanServiceLocator sanService = new SanServiceLocator();
            sanService.setEndpointAddress("SanPort", sanServiceAddr);
            san = sanService.getSanPort();
            if (san == null) {
                System.out.println("service not available");
            }
        }
    }

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        return;
    }
    SOAPHeaderElement hdrElement = setSoapHeader(token);
    SanBindingStub sStub = (SanBindingStub) san;
    sStub.setHeader(hdrElement);
} catch (Exception e) {
    e.printStackTrace();
}
}

private static SOAPHeaderElement setSoapHeader(String token) {

    SOAPHeaderElement hdrElement = new SOAPHeaderElement(
        "http://ep.cisco.dcbu.cisco.com", "token");

    hdrElement.setPrefix("m");

    hdrElement.setMustUnderstand(false);

    hdrElement.setValue(token);
    return hdrElement;
}

static void exception(Exception e){
    header("ZonwWSWclient: Some exception occurred!! \n"+e.toString()+"\nEnter [y] to
view the stack trace or any key to continue...");
    try {
        String view=input_.readLine();
        if(view.equalsIgnoreCase("y")){
            e.printStackTrace();

        }
    } catch (IOException e1) {
        exception(e);
    }
}
}
}

```

Perl Client

This section describes the Perl client that use the SOAP web services, installing ActiveState Perl, and the SOAP:Lite package. Also, this section provides information about installing ActiveState Perl, the SOAP:Lite package and the Perl client to consume the exposed SOAP web services along with the sample code.

This section contains the following sections:

- [Installing Perl](#)
- [Installing SOAP:Lite](#)
- [Sample Code for Perl Client](#)
- [Running the Zone Client](#)
- [Running the Statistics Client](#)

Before executing the Perl script, install ActiveState Perl and the SOAP:Lite package.

Installing Perl

Step 1 Download Perl from <http://www.perl.org/get.html>



Note Choose and download the appropriate Perl version for your operating system.

Step 2 Run the downloaded executable file to install Perl.

Step 3 To verify whether Perl has been installed properly, use the **perl -v** CLI command in the command prompt.
If the installation is proper, the details of the installed version of Perl are displayed.

Installing SOAP:Lite

Download SOAP:Lite from <http://search.cpan.org/dist/SOAP-Lite/>.



Note Choose and download the appropriate SOAP:Lite version for your operating system.

To install SOAP:lite, follow these steps:

Step 1 Using the command prompt, browse to the directory where the Makefile.PL is located in the downloaded SOAP:Lite package.

Step 2 Use the **perl Makefile.PL -noprompt** CLI command to install the SOAP:Lite package.

Step 3 Download nmake utility for Windows from <ftp://ftp.microsoft.com/Softlib/MSLFILES/Nmake15.exe>.

Step 4 Extract the downloaded Nmake15.exe
You will find two files, NMAKE.exe and NMAKE.ERR.
Copy both the NMAKE.exe and NMAKE.ERR files to the installed Perl bin directory.

Step 5 Complete the SOAP:Lite installation by entering the following commands for Windows and Linux machines:

```
nmake
nmake test
nmake install
```

For more information on installing SOAP:Lite, see <http://soaplite.com/install.html>.

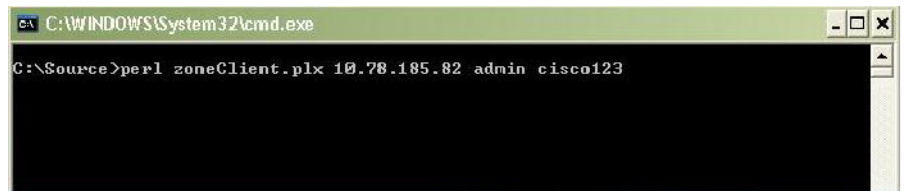
Running the Perl Clients



Note To run a Perl client, the client file and its module file need to be in the same Perl folder.

From the Perl folder where the source files exist, enter the following command:

```
perl Client-File-Name FM-Server-IP Username password
```

REVIEW DRAFT—CISCO CONFIDENTIAL**Figure A-1** Running a Perl Client**Running the Zone Client**

To run the zone client, the files `zoneClient.plx` and `zoneModule.plx` need to be in same directory.

From the Perl folder where the source files exist, enter the following command:

```
perl zoneClient.plx FM-Server-IP Username password
```

Running the Statistics Client

To run the statistics client, the files `statistics.plx` and `statisticsModule.plx` need to be in same directory.

From the Perl folder where the source files exist, enter the following command:

```
perl statistics.plx FM-Server-IP Username password
```

Sample Code for Perl Client**Note**

- The sample code provided here works only with DCNM release 6.1(x) and later. For prior DCNM versions, you need to make changes to the namespace base on the WSDL.
- To collect statistics related data, you need to turn on the PM.

Example A-2 *ZoneClient.plx*

```
# File: zoneClient.plx
# Name: Shailin Saraiya
# Description: zoneClient

# Variables

my $username;
my $password;
my $vsanId = 01;
my $vsan_wwn = '20:01:00:0d:ec:19:6a:81';
my $wwn = '20:00:00:0d:ec:19:6a:80';
my $zoneName = 'test5';
my $readOnly = 0;
my $broadcast = 0;
my $qos = 0;
my $qosPriority = -1;
my $zoneMemberType = 1;
my $zoneMemberFormat = 1;
my $zoneMemberIvrFabricIndex = -1;
my $zoneMemberIvrVsanIndex = -1;
my $zoneMemberId = 'b1a2334455667788';
my $zoneLastModtime = 0;
```

```

my $zoneMemberLunId = '';
my $zoneMemberDbId = 0;
my $zoneisIvr = 0;
my $zoneSetName = 'ZoneSet5';
my $active = 0;

require 'zoneModule.plx';
($url,$username,$password) = @ARGV;
# Getting Token
my $token = getToken($url,$username,$password);
if(!$token) {
    print "\nError in getToken call \n";
    goto exit;
}
# Creating Zone
my $zoneIndex =
createZone($token,$vsanId,$vsan_wwn,$wwn,$zoneName,$readOnly,$broadcast,$qos,$qosPriority)
;

if(!$zoneIndex) {
    print "\nError in createZone call \n";
    goto exit;
}
# Adding Member to Zone
my $status =
addZoneMemberToZone($token,$vsanId,$vsan_wwn,$wwn,$zoneName,$zoneMemberType,$zoneMemberFormat,
$zoneMemberIvrFabricIndex,$zoneMemberIvrVsanIndex,$zoneMemberId,$zoneMemberLunId);
if(!$status) {
    print "\nError in addZoneMemberToZone call \n";
    goto exit;
}
# Creating ZoneSet
my $zoneSetIndex = createZoneSet($token,$vsanId,$vsan_wwn,$wwn,$zoneSetName);
    if(!$zoneSetIndex) {
        print "\nError in createZoneSet call \n";
        goto exit;
    }
# Add Zone to ZoneSet
my $status =
addZoneToZoneset($token,$vsanId,$vsan_wwn,$wwn,$zoneIndex,$zoneName,$readOnly,$broadcast,$
qos,$qosPriority,$zoneMemberDbId,$zoneMemberType,$zoneLastModtime,$zoneMemberLunId,$zoneis
Ivr,$zoneSetIndex,$zoneSetName,$active);
if(!$status) {
    print "\nError in addZoneToZoneset call \n";
    goto exit;
}
# Activate ZoneSet
$status = activateZoneset($token,$vsanId,$vsan_wwn,$wwn,$zoneSetName);
if(!$status) {
    print "\nError in activateZoneset call \n";
    goto exit;
}
else {
    print "successful \n";
}
exit:

```

Example A-3 ZoneModule.plx

```

use SOAP::Lite;
# Get Token
sub getToken {

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

my $url;
my $username;
my $password;
($url,$username,$password) = @_;
# Setting uri and proxy

my $client = SOAP::Lite
    #->uri('http://ep.jaxws.dcbu.cisco.com/')
->proxy('https://'. $url .'/LogonWSService/LogonWS?wsdl')
->ns('http://ep.jaxws.dcbu.cisco.com/', 'namesp1');
my $username = SOAP::Data->type('string')->name('username')->value($username);
my $password = SOAP::Data->type('string')->name('password')->value($password);
my $expiration = SOAP::Data->type('long')->name('expiration')->value(100000000);
# Calling requestToken method
my $result = $client->requestToken($username,$password,$expiration);
# check for error
unless ($result->fault) {
my $token = $result->result();
return $token;
} else {
# error handling
print join ' ',
    $result->faultcode,
    $result->faultstring,
    $result->faultdetail;
return 0
}
}

# Create Zone

sub createZone {

my $token;
my $arg_vsanId;
my $arg_vsanwwn;
my $arg_wwn;
my $arg_zoneName;
my $arg_readOnly;
my $arg_broadcast;
my $arg_qos;
my $arg_qosPriority;

($token,$arg_vsanId,$arg_vsanwwn,$arg_wwn,$arg_zoneName,$arg_readOnly,$arg_broadcast,$arg_
qos,$arg_qosPriority) = @_;
my $VsanKey = SOAP::Data->name("vsanKey" => \SOAP::Data->value(
    SOAP::Data->name("vsanID" => $arg_vsanId),
    SOAP::Data->name("pWwn" => \SOAP::Data->value(SOAP::Data->name("value" =>
$arg_vsanwwn)))->type("Wwn")))->type("VsanKey");
my $Wwn = SOAP::Data->name("switchKey" => \SOAP::Data->value(
    SOAP::Data->name("wwn" =>
\SOAP::Data->value(SOAP::Data->name("value"=>$arg_wwn)))->type("Wwn")))->type("WwnKey");
my $zName = SOAP::Data->name("zoneName"=>$arg_zoneName);
my $readOnly = SOAP::Data->name("readOnly")->value($arg_readOnly);
my $broadcast = SOAP::Data->name("broadcast")->value($arg_broadcast);
my $qos = SOAP::Data->type('boolean')->name("qos")->value($arg_qos);
my $qosPriority =
SOAP::Data->type('int')->name("qosPriority")->value($arg_qosPriority);

# Setting header

```

```

        my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.c
om')->type('string');
        my $client = SOAP::Lite
            #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
            ->proxy('https://'. $url. '/ZoneManagerWSService/ZoneManagerWS?wsdl')
            ->ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp2');

        # Calling createZone method
my $result =
$client->createZone($header, $VsanKey, $Wwn, $zName, $readOnly, $broadcast, $qos, $qosPriority);
        unless ($result->fault) {
            return $result->valueof('//index');
        } else {
            print join ' ', ' ',
                $result->faultcode,
                $result->faultstring,
                $result->faultdetail;
            return 0;
        }
    }

# Adding Zone Member To Zone

sub addZoneMemberToZone {
    my $token;
    my $arg_vsanId;
    my $arg_wwn;
    my $arg_vsanwwn;
    my $arg_zoneName;
    my $arg_zoneMemberType;
    my $arg_zoneMemberFormat;
    my $arg_zoneMemberIvrFabricIndex;
    my $arg_zoneMemberIvrVsanIndex;
    my $arg_zoneMemberId;
    my $arg_zoneMemberLunId;

    ($token, $arg_vsanId, $arg_vsanwwn, $arg_wwn, $arg_zoneName, $arg_zoneMemberType, $arg_zoneMembe
rFormat, $arg_zoneMemberIvrFabricIndex, $arg_zoneMemberIvrVsanIndex, $arg_zoneMemberId, $arg_z
oneMemberLunId) = @_;
    my @hex      = ($arg_zoneMemberId =~ /(..)/g);
    my @dec      = map { hex($_) } @hex;
    my @zoneMemberId = map { pack('C', $_) } @dec;
    my $arg_zoneMemberId = join("", @zoneMemberId);

    my $client = SOAP::Lite
        #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
        ->proxy('https://'. $url. '/ZoneManagerWSService/ZoneManagerWS?wsdl')
        ->ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp3');
    my $VsanKey = SOAP::Data->name("vsanKey" => \SOAP::Data->value(
        SOAP::Data->name("vsanID" => $arg_vsanId),
        SOAP::Data->name("pWwn" => \SOAP::Data->value(SOAP::Data->name("value" =>
$arg_vsanwwn)))->type("Wwn")))->type("VsanKey");
    my $Wwn = SOAP::Data->name("switchKey" => \SOAP::Data->value(
        SOAP::Data->name("wwn" =>
\SOAP::Data->value(SOAP::Data->name("value"=>$arg_wwn))->type("Wwn")))->type("WwnKey");
    my $zName = SOAP::Data->name("zoneName"=>$arg_zoneName);
    my $zMemberType = SOAP::Data->name("zoneMemberType")->value($arg_zoneMemberType);
    my $zMemberFormat =
SOAP::Data->name("zoneMemberFormat")->value($arg_zoneMemberFormat);
    my $zMemberIvrFabricIndex =
SOAP::Data->name('zoneMemberIvrFabricIndex')->value($arg_zoneMemberIvrFabricIndex);

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

my $zMemberIvrVsanIndex =
SOAP::Data->type('int')->name("zoneMemberIvrVsanIndex")->value($arg_zoneMemberIvrVsanIndex
);
my $zMemberId = SOAP::Data->name("zoneMemberId")->value($arg_zoneMemberId);
my $zMemberLunId = SOAP::Data->name("zoneMemberLunId")->value($arg_zoneMemberLunId);

my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.c
om')->type('string');
my $result =
$client->addZoneMemberToZone($header,$VsanKey,$Wwn,$zName,$zMemberType,$zMemberFormat,$zMe
mberIvrFabricIndex,$zMemberIvrVsanIndex,$zMemberId,$zMemberLunId);
unless ($result->fault) {
    return 1;
}
else {
    print join ' ', ' ',
        $result->faultcode,
        $result->faultstring,
        $result->faultdetail;
    return 0;
}
}

# Creating ZoneSet

sub createZoneSet {

    my $token;
    my $arg_vsanId;
    my $arg_vsanwwn;
    my $arg_wwn;
    my $arg_zoneSetName;

    ($token,$arg_vsanId,$arg_vsanwwn,$arg_wwn,$arg_zoneSetName) = @_;
    my $client = SOAP::Lite
        #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
        ->proxy('https://'. $url. '/ZoneManagerWSService/ZoneManagerWS?wsdl')
        ->ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp4');
    # Creating complex data type

    my $VsanKey = SOAP::Data->name("vsanKey" => \SOAP::Data->value(
        SOAP::Data->name("vsanID" => $arg_vsanId),
        SOAP::Data->name("pWwn" => \SOAP::Data->value(SOAP::Data->name("value" =>
$arg_vsanwwn)))->type("Wwn"))->type("VsanKey");
    my $Wwn = SOAP::Data->name("switchKey" => \SOAP::Data->value(
        SOAP::Data->name("wwn" =>
\SOAP::Data->value(SOAP::Data->name("value"=>$arg_wwn))->type("Wwn"))->type("WwnKey");
    my $zSetName = SOAP::Data->name("zoneSetName"=>$arg_zoneSetName);

    # Setting header

    my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.c
om')->type('string');

    # Calling createZone method

    my $result = $client->createZoneSet($header,$VsanKey,$Wwn,$zSetName);

    unless ($result->fault) {
        return $result->valueof('//index')." \n";
    } else {

```

```

        print join ' ', '
            $result->faultcode,
            $result->faultstring,
            $result->faultdetail;
    return 0;
    }
}

# Adding Zone to ZoneSet
sub addZoneToZoneset {
    my $token;
    my $arg_vsanId;
    my $arg_vsanwwn;
    my $arg_wwn;
    my $arg_index;
    my $arg_zoneName;
    my $arg_readOnly;
    my $arg_broadcast;
    my $arg_qos;
    my $arg_qosPriority;
    my $arg_zoneMemberDbId;
    my $arg_zoneLastModtime;
    my $arg_memberType;
    my $arg_zoneMemberLunId;
    my $arg_isIvr;
    my $arg_zoneSetindex;
    my $arg_zoneSetName;
    my $active;

    ($token,$arg_vsanId,$arg_vsanwwn,$arg_wwn,$arg_index,$arg_zoneName,$arg_readOnly,$arg_broa
dcast,$arg_qos,$arg_qosPriority,$arg_zoneMemberDbId,$arg_memberType,$arg_zoneLastModtime,$
arg_zoneMemberLunId,$arg_isIvr,$arg_zoneSetindex,$arg_zoneSetName,$active) = @_;

    my $client = SOAP::Lite
        #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
        ->proxy('https://'. $url.' /ZoneManagerWSService/ZoneManagerWS?wsdl')
        ->ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp5');
    # Creating complex data type

    my $VsanKey = SOAP::Data->name("vsanKey" => \SOAP::Data->value(
        SOAP::Data->name("vsanID" => $arg_vsanId),
        SOAP::Data->name("pWwn" => \SOAP::Data->value(SOAP::Data->name("value" =>
$arg_vsanwwn))->type("Wwn"))->type("VsanKey"));
    my $Wwn = SOAP::Data->name("switchKey" => \SOAP::Data->value(
        SOAP::Data->name("wwn" =>
\SOAP::Data->value(SOAP::Data->name("value"=>$arg_wwn))->type("Wwn"))->type("WwnKey"));
    my $zoneDO = SOAP::Data->name("zoneDo"=> \SOAP::Data->value(
        SOAP::Data->name("index"=>$arg_index),
        SOAP::Data->name("name"=>$arg_zoneName),
        SOAP::Data->name("readonly"=>$arg_readonly),
        SOAP::Data->name("qos"=>$arg_qos),
        SOAP::Data->name("qosPriority"=>arg_qosPriority),
        SOAP::Data->name("broadcast"=>$arg_broadcast),
        SOAP::Data->name("active"=>$active),
        SOAP::Data->name("zoneLastModtime"=>arg_zoneLastModtime),
        SOAP::Data->name("zoneMemberDbId"=>$arg_zoneMemberDbId),
        SOAP::Data->name("vsanId"=>$arg_vsanId),
        SOAP::Data->name("isIvr"=>$arg_isIvr),
        SOAP::Data->name("memberType"=>$arg_memberType),
        SOAP::Data->name("lunId"=>$arg_zoneMemberLunId))->type("zone");
    my $zoneSetDO = SOAP::Data->name("zoneSetDo"=> \SOAP::Data->value(
        SOAP::Data->name("index"=>$arg_zoneSetindex),

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

SOAP::Data->name("name"=>$arg_zoneSetName),
SOAP::Data->name("active"=>$active),
SOAP::Data->name("zoneLastModified"=>$arg_zoneLastModtime))>type("ZoneSet");

# Setting header

my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.com')->type('string');

# Calling addZonetoZoneSet method

my $result = $client->addZoneToZoneset($header,$VsanKey,$Wwn,$zoneDO,$zoneSetDO);

unless ($result->fault) {
    return 1;
} else {
    print join ', ',
        $result->faultcode,
        $result->faultstring,
        $result->faultdetail;
    return 0;
}
}

#Activating ZoneSet

sub activateZoneset {

    my $token;
    my $arg_vsanId;
    my $arg_vsanwwn;
    my $arg_wwn;
    my $arg_zoneSetName;

    ($token,$arg_vsanId,$arg_vsanwwn,$arg_wwn,$arg_zoneSetName) = @_;

    my $client = SOAP::Lite
        #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
        ->proxy('https://'. $url.' /ZoneManagerWSService/ZoneManagerWS?wsdl')
        ->ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp6');
    # Creating complex data type

    my $VsanKey = SOAP::Data->name("vsanKey" => \SOAP::Data->value(
        SOAP::Data->name("vsanID" => $arg_vsanId),
        SOAP::Data->name("pWwn" => \SOAP::Data->value(SOAP::Data->name("value"
=> $arg_vsanwwn)))>type("Wwn"))>type("VsanKey");
    my $Wwn = SOAP::Data->name("switchKey" => \SOAP::Data->value(
        SOAP::Data->name("wwn" =>
\SOAP::Data->value(SOAP::Data->name("value"=>$arg_wwn))>type("Wwn"))>type("WwnKey");
    my $zSetName = SOAP::Data->name("zoneSetName"=>$arg_zoneSetName);

    # Setting header

    my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.com')->type('string');

    # Calling activateZoneset method

    my $result = $client->activateZoneset($header,$VsanKey,$Wwn,$zSetName);
    unless ($result->fault) {
        return 1;
    }
}

```

```

    } else {
        print join ' ', ' ',
            $result->faultcode,
            $result->faultstring,
            $result->faultdetail;
        return 0;
    }
}
1;

```

Example A-4 StatisticsClient.plx

```

# File: StatisticsClient.plx
# Name: Shailin Saraiya
# Description: Statistics Client

# Variables

my $username;
my $password;
my $url;
my $fabricDbId = -1;
my $switchDbId = -1;
my $vsanDbId = -1;
my $sortField = 'rxTxStr';
my $sortType = 'DESC';
my $limit = NaN;
my $groupId = -1;
my $isGroup = 0;
my $networkType = 'SAN';
my $filterStr = '';
my $temp = '24 Hours';
my $startIdx = 0;
my $recordSize = 45;
my $interval = 1;
require 'statisticsModule.plx';

($url,$username,$password) = @ARGV;

# Getting Token
my $token = getToken($url,$username,$password);
($rrdFile,$fid,$pmType) =
getIslStatList($token,$fabricDbId,$switchDbId,$vsanDbId,$sortField,$sortType,$limit,$groupId,
$isGroup,$networkType,$filterStr,$temp,$startIdx,$recordSize);
for (my $i=0;$i<1;$i++) {
    getPmChartData($token,@$rrdFile[$i],@$pmType[$i],@$fid[$i],$interval) . "\n";
}

```

Example A-5 StatisticsModule.plx

```

use SOAP::Lite;
use Time::Local;

# Get Token
$url = '';

sub getToken {
    my $username;

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

my $password;
($url,$username,$password) = @_;

# Setting uri and proxy

my $client = SOAP::Lite
#->uri('http://ep.jaxws.dcbu.cisco.com/')
->proxy('https://'. $url. '/LogonWSService/LogonWS?wsdl')
    -> ns('http://ep.jaxws.dcbu.cisco.com/', 'namesp1');
    #-> autotype(0)
    #-> readable(1);
my $username = SOAP::Data->type('string')->name('username')->value($username);
my $password = SOAP::Data->type('string')->name('password')->value($password);
my $expiration = SOAP::Data->type('long')->name('expiration')->value(100000000);

# Calling requestToken method

my $result = $client->requestToken($username,$password,$expiration);
    # check for error
unless ($result->fault) {
    my $token = $result->result();
    $token;
} else {

# error handling
print join ', ',
    $result->faultcode,
    $result->faultstring,
    $result->faultdetail;
}

}

sub getIslStatList {
    my $token;
    my $arg_fabricDbId;
    my $arg_switchDbId;
    my $arg_vsanDbId;
    my $arg_sortField;
    my $arg_sortType;
    my $arg_limit;
    my $arg_groupId;
    my $arg_isGroup;
    my $arg_networkType;
    my $arg_filterStr;
    my $arg_interval;
    my $arg_startIdx;
    my $arg_recordSize;

($token,$arg_fabricDbId,$arg_switchDbId,$arg_vsanDbId,$arg_sortField,$arg_sortType,$arg_limit,$arg_groupId,$arg_isGroup,$arg_networkType,$arg_filterStr,$arg_interval,$arg_startIdx,$arg_recordSize) = @_;

    my $DbFilter= SOAP::Data->name("arg0" => \SOAP::Data->value(
        SOAP::Data->name("fabricDbId" => $arg_fabricDbId),
        SOAP::Data->name("switchDbId" => $arg_switchDbId),
        SOAP::Data->name("vsanDbId" => $arg_vsanDbId),
        SOAP::Data->name("sortField" => $arg_sortField),
        SOAP::Data->name("sortType" => $arg_sortType),
        SOAP::Data->name("limit" => $arg_limit),
        SOAP::Data->name("groupId" => $arg_groupId),
        SOAP::Data->name("isGroup" => $arg_isGroup)->type("boolean"),
        SOAP::Data->name("networkType" => $arg_networkType),
        SOAP::Data->name("filterStr" => $arg_filterStr)))->type("DbFilter");

```

```

my $interval = SOAP::Data->type("string")->name("arg1")->value($arg_interval);
my $data_startIdx = SOAP::Data->name("arg2"=>$arg_startIdx);
my $data_recordSize = SOAP::Data->name("arg3"=>$arg_recordSize);

# Setting header

my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.com')->type('string');
my $header = SOAP::Header->name('token')->prefix('m')->value($token)->type('string');
my $client = SOAP::Lite
    #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
    ->proxy('https://'. $url. '/StatisticsWSService/StatisticsWS?wsdl')
        -> ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp2');
    #-> autotype(0)
    #-> readable(1);

# Calling getIslStatDataLength method
my $result =
$client->getIslStatList($header,$DbFilter,$interval,$data_startIdx,$data_recordSize);
unless ($result->fault) {
    print "\ngetIslStatList\n\n";
    my @fabric = $result->valueof('//item/fabric');
    my @title = $result->valueof('//item/title');
    my @maxTxStr = $result->valueof('//item/maxTxStr');
    my @maxRxStr = $result->valueof('//item/rxTxStr');
    my @rxTx = $result->valueof('//item/rxTx');

    my $count = @fabric;

    for(my $i=0;$i<10;$i++)
    {

        print "-----\n";
        print "Fabric -> " . $fabric[$i] . "\n";
        print "Title -> " . $title[$i] . "\n";
        print "maxTxStr -> " . $maxTxStr[$i] . "\n";
        print "maxRxStr -> " . $maxRxStr[$i] . "\n";
        print "rxTx -> " . $rxTx[$i] . "\n";
        print "-----\n";

    }

    my @rrdFile = $result->valueof('//item/rrdFile');
    my @fid = $result->valueof('//item/fid');
    my @pmType = $result->valueof('//item/pmtype');
    return (\@rrdFile,\@fid,\@pmType);

} else {
    print join ' ',
        $result->faultcode,
        $result->faultstring,
        $result->faultdetail;
}

}

sub getPmChartData {
    my $token;
    my $arg_rrd;
    my $arg_pmType;
    my $arg_fid;

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

my $arg_interval;
($token,$arg_rrd,$arg_pmType,$arg_fid,$arg_interval) = @_;
my $data_rrd = SOAP::Data->type("string")->name("arg0")->value($arg_rrd);
my $data_pmType = SOAP::Data->name("arg1")->value($arg_pmType);

my $data_fid = SOAP::Data->name("arg2"=>$arg_fid);
my $data_interval = SOAP::Data->name("arg3"=>$arg_interval);

# Setting header
my $header =
SOAP::Header->name('token')->prefix('m')->value($token)->uri('http://ep.cisco.dcbu.cisco.c
om')->type('string');
my $client = SOAP::Lite
    #->uri('http://ep.san.jaxws.dcbu.cisco.com/')
    ->proxy('https://'. $url.'/StatisticsWSService/StatisticsWS?wsdl')
        -> ns('http://ep.san.jaxws.dcbu.cisco.com/', 'namesp3');
        #-> autotype(0)
        #-> readable(1);

# Calling getPmChartData method
my $result =
$client->getPmChartData($header,$data_rrd,$data_pmType,$data_fid,$data_interval);
unless ($result->fault) {
    print "\n\nGetPmChartData\n\n";

    my @item = $result->valueof('//item/item');
    for(my $i=2;$i<30;$i=$i+3) {
        print "-----\n";
        print "Rx -> ".$item[$i-2]."\n";
        print "Tx -> ".$item[$i-1]."\n";
        print "Time -> ".scalar localtime($item[$i]) . "\n";
        print "-----\n";
    }
    } else {
    print join ', ',
        $result->faultcode,
        $result->faultstring,
        $result->faultdetail;
    }
}

1;

```

REST Web Services

The following client programs use REST web services in Cisco DCNM.

This section contains the following sections:

- [Python Client](#)
- [JavaScript Client](#)

Python Client

This section provides information about installing Python, and using the python client to consume the exposed REST web services along with the sample code.

This section contains the following sections:

- [Installing Perl](#)
- [Installing Pycharm \(IDE\)](#)
- [Running Python Client Using IDE](#)
- [Running Python Client Using Command Prompt](#)
- [Sample Code for Python Client](#)

Installing Python

-
- Step 1** You need to use the Python version 3.6.1.
- Step 2** Download Python from <https://www.python.org/downloads/release/python-361/>.
-

Installing Pycharm (IDE)

-
- Step 1** You need to use the Pycharm version 3.6.1.
- Step 2** Download Pycharm from <https://www.jetbrains.com/pycharm/download/#section=windows>.
-

Running Python Client Using IDE

-
- Step 1** Using Pycharm (Python IDE), set the interpreter to the location where python is installed. For example, C:\Users\Local\Programs\Python\Python36-32\python.exe.



Note In the *MyClient.py* file, set the queryKey, zone name, zoneset name, fabricDBID accordingly.

- Step 2** Go to **Run > Edit Configurations**. In the Script parameters tab provide the arguments such as FMServerIP, UserName, Password that need to be separated by blank spaces. For example, FMServerIP username password)
- Step 3** Right-click on the file, and select “Run FileName”.
-

Running Python Client Using Command Prompt

-
- Step 1** Using the command prompt, go to the location where python is installed.



Note In the *MyClient.py* file, set the queryKey, zone name, zoneset name, fabricDBID accordingly.

```
C:\users\exampleuser\AppData\Local\Programs\Python\Python36>python.exe
C:\Users\exampleuser\PycharmProjects\PythonRestClient\MyClient.py FMServerIP username
password
```

REVIEW DRAFT—CISCO CONFIDENTIAL

Step 2 Execute “`python.exe <complete path for MyClient.py file> FMServerIP username password`”. For example:

```
C:\Python\Python36-32>python.exe C:\Users\PycharmProjects\PythonRestClient\MyClient.py
FMServerIP username password
```

Sample Code for Python Client

Example A-6 MyClient.py

```
import sys
import time
from restfactory import RestFactory

try:
    fmserver = sys.argv[1]
    user = sys.argv[2]
    password = sys.argv[3]
    rf = RestFactory(fmserver, user, password)
    i_rest = rf.get_webzonemanager_rest();

    zonesetName = "PythonZoneSetTest"
    zoneName = "PythonTest"
    vsanID = 200
    switchName = "sw-9148-248"
    fabricName = "Fabric_sw-9250i-244"
    cacheBust = i_rest.getCacheBust()
    navId = "-1";
    fabricDBID = i_rest.get_fabrics(cacheBust, navId, fabricName)
    print("fabricDBID "+str(fabricDBID))
    vsanDBID = i_rest.get_vsanDBID(fabricDBID, cacheBust, navId, vsanID, switchName)
    print("VSANDBID "+str(vsanDBID))
    swwn =
    i_rest.get_switchesForVsan(fabricDBID, vsanDBID, cacheBust, navId, vsanID, switchName)
    print("SwitchWWN "+swwn)

    #queryKey is the combination of <FDBID>,<VSANIndex>,<SWWWN>,<client ID>,<otherclient
data>
    queryKey = i_rest.generate_queryKey(fabricDBID, vsanID, swwn, "-1", "-1")
    print(queryKey)

    time.sleep(2)
    createZoneSetResponse = i_rest.createZoneSet(queryKey, zonesetName, fabricDBID)

    # creatingZONE
    createZoneResponse = i_rest.create_zone(queryKey, zoneName, fabricDBID)
    #
    # # addingZoneToZoneSet
    addZoneResponse = i_rest.addZoneToZoneSet(queryKey, zoneName, zonesetName, fabricDBID)
    #
    # # deletingZoneFromZoneSet
    removeZoneFromZoneSet = i_rest.removeZoneToZoneSet(queryKey, zoneName, zonesetName,
fabricDBID)
    #
    # # deletingZone
    deleteZoneResponse = i_rest.deleteZone(queryKey, zoneName, fabricDBID)
    #
    # # deleting ZoneSet
```

```

        deleteZoneSet = i_rest.deleteZoneset(queryKey, zonesetName, fabricDBID)

except Exception as e:
    print(e)

```

Example A-7 *restfactory.py*

```

from san_rest_api import RestApi
from webzonemanagerrest import WebZoneManager

class RestFactory(object):
    _cookie = {}
    _dcnm_url = None
    _username = None
    _password = None
    _rest_api = None
    _is_remote = False
    _expiration_time = "9999999"

    def __init__(self, dcnm_url , username = None, password = None):
        if username is not None:
            self._username = username
            self._password = password
            print(dcnm_url)
            dcnm_url = "https://" + dcnm_url + "/"
            print(dcnm_url)
            self._rest_api = RestApi(dcnm_url,
                                    self._username,
                                    self._password,
                                    self._expiration_time)

            print(dcnm_url)
            print("here")
            print(self._rest_api)
            dcnm_token = self._rest_api.dcnm_token['Dcnm-Token']
            if dcnm_token is None:
                raise Exception("Failed to instantiate RestFactory as Dcnm-Token is :" +
                                dcnm_token)
            else:
                self._rest_api.headers['Dcnm-Token'] = dcnm_token
                print("headers with token")
                print(self._rest_api.headers)
            self._dcnm_url = dcnm_url

    def get_webzonemanager_rest(self):
        return WebZoneManager(self._dcnm_url, self._rest_api)

```

Example A-8 *san_rest_api.py*

```

"""This class handles all the REQUEST modules required for DCNM"""
import ast
import base64
import json
import requests

LOGON_URL = 'rest/logon'

class RestApi(object):

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

"""This handles all the request modules"""

def __init__(self, *args):
    self.headers = {'Content-Type': 'application/json; charset=utf8'}
    self.verify = False
    if len(args) != 0:
        self.dcnm_url = args[0]
        self.username = args[1]
        self.password = args[2]
        self.logon_url = self.dcnm_url + LOGON_URL
        self.expiration_time = args[3]
        self.dcnm_token = None
        self.fdbid = None
        self.get_dcnmtoken()

def get_dcnmtoken(self):
    self.credentials = self.generate_encoded_credentials(
        self.username, self.password)
    self.dcnm_token = self.generate_dcnmtoken(self.credentials)
    print("dcnm-token")
    print(self.dcnm_token)
    return self.dcnm_token

def generate_encoded_credentials(self, username, password):
    """This generates the credentials"""
    data = {'Content-Type': 'application/json; charset=utf8'}
    credentials = username + ':' + password
    encoded = base64.b64encode(credentials.encode())
    print("////////////////")
    print(encoded)
    print(encoded.decode('ascii'))
    print("////////////////")
    data.update({"Authorization": 'Basic ' + encoded.decode('ascii')})
    print(data)
    return data

def generate_dcnmtoken(
    self,
    headers,
    js_token=None):
    """This generates valid DCNM-Token for the encoded credentials"""
    dcnm_token = None
    payload = {'expirationTime': self.expiration_time}
    print("////////////////Header////////")
    print(headers)
    try:
        print(self.logon_url)
        response = requests.post(
            self.logon_url,
            data=json.dumps(payload),
            headers=headers,
            verify=self.verify)
        if response.status_code == 200: # Successful token generation
            dcnm_token = ast.literal_eval(response.text)
    except:
        return None
    return dcnm_token

```

Example A-9 webzonemanagerrest.py

```
import requests
```

```

import time
import json

class WebZoneManager(object):

    ZONE_REST_PATH = "fm/fmrest/WebZoneManager/"
    CREATE_ZONE = "createZoneMod"
    CREATE_ZONESET = "createZoneSetMod"
    ADD_ZONE_TO_ZONESET = "addZonesToZonesetMod"
    REGISTER_AND_POPULATE = "registerAndPopulateLocalZoneDBGet"
    DELETE_ZONE_FROM_ZONESET = "deleteZonesFromZonesetMod"
    DELETE_ZONESET = "deleteZonesetsMod"
    DELETE_ZONE = "deleteZonesMod"
    GET_FABRICS_URL = 'fm/fmrest/dcnm/fabrics'
    GET_VSANS_MOD_URL = "getVsansMod"
    GET_SWITCHES_IN_VSAN = "getSwitchesInVSANMod"

    def __init__(self, dcnm_url, restapi):
        self._dcnm_url = dcnm_url + self.ZONE_REST_PATH
        self._rest_api = restapi
        self.get_fabrics_url = dcnm_url + self.GET_FABRICS_URL
        self.verify = False
        self.headers = {'Content-Type': 'application/json; charset=utf8'}

    def createZoneSet(self, queryKey, zonesetName, fabricDBID):
        print("creating zoneset")
        zoneset_url = self._dcnm_url+self.CREATE_ZONESET;
        print(zoneset_url)
        self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
        payload = {}
        payload['queryKey'] = queryKey
        payload['zoneSetName'] = zonesetName
        payload['fabricDBID'] = fabricDBID
        response = requests.post(url=zoneset_url, data=payload,
headers=self._rest_api.headers, verify=self.verify)
        print(response.text)

    def create_zone(self, queryKey, zoneName, fabricDBID):
        print("creating zone")
        zone_url = self._dcnm_url + self.CREATE_ZONE
        print(zone_url)
        self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
        payload = {}
        payload['queryKey'] = queryKey
        payload['zoneName'] = zoneName
        payload['fabricDBID'] = fabricDBID
        response = requests.post(zone_url, data=payload, headers=self._rest_api.headers,
verify=self.verify)
        print("printing response")
        print(response.text)

    def addZoneToZoneSet(self, queryKey, zoneNames, zonesetName, fabricDBID):
        print("adding zone to zoneset")
        addzoneurl = self._dcnm_url + self.ADD_ZONE_TO_ZONESET
        print(addzoneurl)
        self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
        payload = {}
        payload['queryKey'] = queryKey
        payload['zoneNames'] = zoneNames
        payload['zonesetName'] = zonesetName
        payload['fabricDBID'] = fabricDBID
        response = requests.post(addzoneurl, data=payload, headers=self._rest_api.headers,
verify=self.verify)
        print("printing response")

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

print(response.text)

def removeZoneToZoneSet(self, queryKey, zoneNames, zonesetName, fabricDBID):
    print("deleting zone from zoneset")
    removezoneurl = self._dcnm_url + self.DELETE_ZONE_FROM_ZONESET
    print(removezoneurl)
    self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
    payload = {}
    payload['queryKey'] = queryKey
    payload['zoneName'] = zoneNames
    payload['zonesetName'] = zonesetName
    payload['fabricDBID'] = fabricDBID
    response = requests.post(url=removezoneurl, data=payload,
headers=self._rest_api.headers, verify=self.verify)
    print("printing response")
    print(response.text)

def deleteZone(self, queryKey, zoneName, fabricDBID):
    print("deleting zone")
    deletezoneurl = self._dcnm_url + self.DELETE_ZONE
    print(deletezoneurl)
    self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
    payload = {}
    payload['queryKey'] = queryKey
    payload['zoneName'] = zoneName
    payload['fabricDBID'] = fabricDBID
    response = requests.post(url=deletezoneurl, data=payload,
headers=self._rest_api.headers, verify=self.verify)
    print("printing response")
    print(response.text)

def deleteZoneset(self, queryKey, zonesetName, fabricDBID):
    print("deleting zoneset")
    delete_ZoneSet_url = self._dcnm_url+self.DELETE_ZONESET
    print(delete_ZoneSet_url)
    self._rest_api.headers['Content-Type'] = 'application/x-www-form-urlencoded'
    payload = {}
    payload['queryKey'] = queryKey
    payload['zoneSetName'] = zonesetName
    payload['fabricDBID'] = fabricDBID
    response = requests.post(url=delete_ZoneSet_url, data=payload,
headers=self._rest_api.headers, verify=self.verify)
    print(response.text)

def registerAndPopulate(self, queryKey, fabricDBID, clientRequestID, navId):
    registerAndPopulate_url = self._dcnm_url+self.REGISTER_AND_POPULATE
    registerAndPopulate_url += '/?queryKey=' + queryKey + '&fabricDBID=' + fabricDBID
    + '&clientRequestID=' + clientRequestID + \
        '&cacheBust=' + self.getCacheBust() + '&navId=' + navId
    response = requests.get(registerAndPopulate_url, params=None,
headers=self._rest_api.headers, verify=self.verify)
    resp_dict = json.loads(response.text)
    req_key = resp_dict['requestKey']
    return req_key

def getCacheBust(self):
    return str(int(time.time() * 1000))

def get_fabrics(self, cachebust, navId, fabricName):
    print("fetching fabrics")
    get_fabric_url = self.get_fabrics_url+"/?"+'&cacheBust=' + cachebust + '&navId=' +
navId
    print(get_fabric_url)
    self.headers['Content-Type'] = 'application/x-www-form-urlencoded'

```

```

        response = requests.get(url=get_fabric_url, params=None,
headers=self._rest_api.headers, verify=self.verify)
        print(response.text)
        resp = json.loads(response.text)
        for res in resp:
            if res['name'] == fabricName:
                self.fdbid = res['id']
        return self.fdbid

    def get_vsanDBID(self, fabricid, cachebust, navId, vsanID, switchName):
        print("fetching vsans")
        get_vsans_url = self._dcnm_url+self.GET_VSANS_MOD_URL+"/?"+'&fabricDBID=' +
str(fabricid) + '&cacheBust=' + cachebust + '&navId=' + navId
        print(get_vsans_url)
        self.headers['Content-Type'] = 'application/x-www-form-urlencoded'
        self.vsan_dbid = None
        response = requests.get(url=get_vsans_url, params=None,
headers=self._rest_api.headers, verify=self.verify)
        print(response.text)
        resp = json.loads(response.text)
        print(resp)
        for res in resp:
            if res['vsanIndex'] == vsanID:
                if res['prinswName'] == switchName:
                    self.vsan_dbid = (res['DBID'])
        return self.vsan_dbid

    def get_switchesForVsan(self, fabricid, vsanid, cachebust, navId, vsanID, switchName):
        print("fetching switches")
        get_switches_for_vsans_url = self._dcnm_url+self.GET_SWITCHES_IN_VSAN + "? " +
'&fabricDBID=' + str(
            fabricid) + '&vsanDBID=' + str(vsanid) + '&cacheBust=' + cachebust + '&navId='
+ navId
        print(get_switches_for_vsans_url)
        self.headers['Content-Type'] = 'application/x-www-form-urlencoded'
        self.switchWWN = None
        response = requests.get(url=get_switches_for_vsans_url, params=None,
headers=self._rest_api.headers, verify=self.verify)
        resp = json.loads(response.text)
        print(resp)
        for res in resp[1]:
            if res['sys_name'] == switchName:
                self.switchWWN = res['swWWN']
        return self.switchWWN

    def generate_queryKey(self, fDBID, vsanIndex, SWWN, clientID, otherClientData):
        key = str(fDBID)+","+str(vsanIndex)+","+str(SWWN)+","+str(clientID)+","+str(otherClientData);
        self.queryKey = self.registerAndPopulate(key, str(fDBID), clientID, "-1")
        print("query key "+self.queryKey)
        return self.queryKey

```

JavaScript Client

This section provides information about using the JavaScript client to access REST Web Services along with sample code and configuring the CORS plugin.

This section contains the following sections:

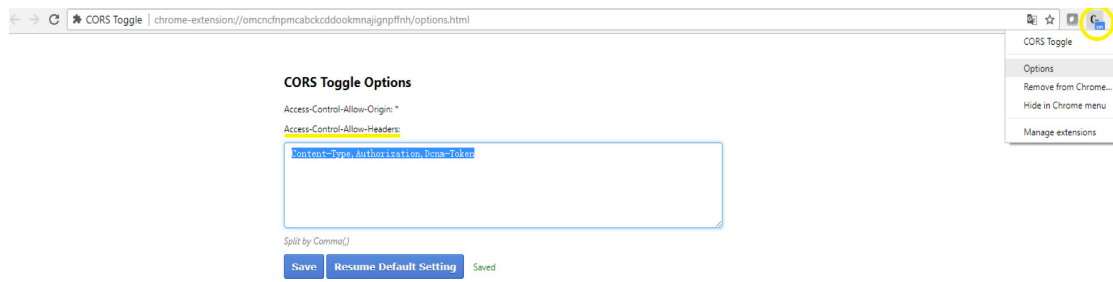
- [Downloading and Configuring the CORS Toggle Plug-in](#)

REVIEW DRAFT – CISCO CONFIDENTIAL

- [Running JavaScript REST Client](#)
- [Sample Code for JavaScript Client](#)

Downloading and Configuring the CORS Toggle Plug-in

- Step 1** Download the CORS Toggle plugin from Chrome web store using the following URL:
<https://chrome.google.com/webstore/search/CORSToggle?hl=en-US>
- Step 2** Add the Access control Allow-Headers in the space provided and save. (For example, Content-Type, Authorization, Dcnm-Token).



- Step 3** Ensure CORS status is 'ON'.
- Step 4** Click on the plug-in to change the status.

Running JavaScript REST Client

- Step 1** Double-click on *JavaScript_Client.html*, which will show the below UI in web browser.

Zone REST calls

DCNM Server IP:	
User Name:	
Password:	
Fabric Name:	
Switch Name:	
VSAN ID:	
ZoneSet Name:	
Zone Name:	

- Step 2** Fill the above fields, and click the **Submit** button. Cisco DCNM performs the supported zone-related operations and displays the status message.

Zone REST calls

DCNM Server IP:	10.100.11.123
User Name:	admin
Password:	*****
Fabric Name:	Fabric_SW-S2004-ET1
Switch Name:	SW-S2004-ET1
VSAN ID:	100
ZoneSet Name:	test_ZoneSet
Zone Name:	test_Zone

SUCCESS- Create Zoneset
SUCCESS- Create Zone
SUCCESS- Adding Zones to Zoneset
SUCCESS- Deleting Zones from Zoneset
SUCCESS- Delete Zone
SUCCESS- Delete ZoneSet

REVIEW DRAFT—CISCO CONFIDENTIAL

Step 3 After you get a success message, you can verify this in Cisco DCNM GUI (/SAN/Zoning).

Sample Code for JavaScript Client

Example A-10 *JavaScript_Client.html*

```
<!DOCTYPE html>
<html>
<body>
  <script
    src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.2.1.min.js"></script>
  <script type="text/javascript">
    function loadDoc() {
      debugger;
      var user = document.getElementById("un").value;
      var password = document.getElementById("pw").value;
      var serverIP = document.getElementById("ip").value;
      var fabricName = document.getElementById("fabricName").value;
      var switchName = document.getElementById("switchName").value;
      var zoneSetName = document.getElementById("zoneSetName").value;
      var zoneName = document.getElementById("zoneName").value;
      var vsanID = document.getElementById("vsanID").value;
      var cacheBust = new Date().getTime() * 1000;
      var navID = -1;
      var encodeUP = btoa(user + ':' + password);
      var authorization='Basic ' + encodeUP;
      var fabricDBID;
      var vsanDbId;
      var swWWN;
      var queryKey;
      var dcnm_Token;
      // Below ajax is for Getting Dcnm-token from serverIP
      $
        .ajax({
          type : "POST",
          headers : {
            "Authorization" : authorization,
          },
          url : "https://" + serverIP + "/rest/logon",
          data : JSON.stringify({
            expirationTime : '9999999'
          }),
          processData : false,
          async : true,
          httpNodeCors : {
            origin : "*",
            methods : "GET,PUT,POST,DELETE"
          },
          error : function(xhr, status, error) {
            alert('error !!');
            document.getElementById("result").innerHTML =
data.statusMessage
              + " " + status;
          },
          success : function(msg, status, xhr) {
            console.log(msg);
            var dcnmToken = xhr.responseText;
            dcnm_Token = dcnmToken.substring(15,
```

```

        dcnmToken.length - 2);
        getFabrics();
    }
});

// getting the fabricDBID
function getFabrics() {

$.ajax({
    url : 'https://' + serverIP
        + "/fm/fmrest/dcnm/fabrics/?&cacheBust="
        + cacheBust + "&navId=" + navID,
    type : 'GET',
    headers : {
        "Content-Type" : "application/x-www-form-urlencoded",
        "Authorization" : authorization,
        "Dcnm-Token" : dcnm_Token
    },
    success : function(data, textStatus, xhr) {
        console.log("data= " + fabrics);
        var fab = JSON.stringify(data);
        var fabrics = JSON.parse(fab);
        for (var i = 0; i < fabrics.length; i++) {
            if (fabricName == fabrics[i].name) {
                fabricDBID = fabrics[i].id;
            }
        }
        getVsanDbId(fabricDBID);
    },
    error : function(xhr, textStatus, errorThrown) {
        console.log(textStatus);
        document.getElementById("result").innerHTML = " "
            + textStatus;
    }
});
}

//getting the VsanDBID from the vsanID (vsanID is provided as the input)
function getVsanDbId(fabricDBID) {

$
    .ajax({
        url : 'https://'
            + serverIP
            + '/fm/fmrest/WebZoneManager/getVsansMod/?&fabricDBID='
            + fabricDBID + '&cacheBust=' + cacheBust
            + '&navId=' + navID,
        type : 'GET',
        headers : {
            "Content-Type" : "application/x-www-form-urlencoded",
            "Authorization" : authorization,
            "Dcnm-Token" : dcnm_Token
        },
        success : function(data, textStatus, xhr) {
            console.log("data= " + data);
            console.log(textStatus);
            var vsanSwitches = JSON.stringify(data);
            var vsanSwitch = JSON.parse(vsanSwitches);
            for (var i = 0; i < vsanSwitch.length; i++) {
                if (vsanID == vsanSwitch[i].vsanIndex) {
                    if (switchName == vsanSwitch[i].prinswName) {
                        vsanDbId = vsanSwitch[i].DBID;
                    }
                }
            }
        }
    })
}

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

    }
    getSwitchesInVSANMod(vsanDbId);
  },
  error : function(xhr, textStatus, errorThrown) {
    console.log(textStatus);
    document.getElementById("result").innerHTML =

data.statusMessage

    + " " + textStatus;
  }
});

}

//getting switch wwn
function getSwitchesInVSANMod(vsanDbId) {
  $.ajax({
    url : 'https://'
      + serverIP
      +
' /fm/fmrest/WebZoneManager/getSwitchesInVSANMod/?&fabricDBID='
      + fabricDBID + '&vsanDBID=' + vsanDbId
      + '&cacheBust=' + cacheBust + '&navId='
      + navID,
    type : 'GET',
    headers : {
      "Content-Type" : "application/x-www-form-urlencoded",
      "Authorization" : authorization,
      "Dcnm-Token" : dcnm_Token
    },
    success : function(data, textStatus, xhr) {
      console.log("data " + data);
      console.log(textStatus);
      var sw_wnn = JSON.stringify(data);
      var sw_WWN = JSON.parse(sw_wnn);
      for (var i = 0; i < sw_WWN[1].length; i++) {
        if (switchName == sw_WWN[1][i].sys_name) {
          swWWN = sw_WWN[1][i].swWWN;
          console.log(swWWN);
        }
      }
      generateQueryKey(fabricDBID, vsanID, swWWN, -1,
        -1);
    },
    error : function(xhr, textStatus, errorThrown) {
      console.log(textStatus);
      document.getElementById("result").innerHTML = " "
        + textStatus;
    }
  });
}

//generating QueryKey
function generateQueryKey(fabricId, vsanID, SWWN, clientRequestID,
  otherClientData) {
  var key = fabricId + ',' + vsanID + ',' + SWWN + ','
    + clientRequestID + ',' + otherClientData;

  $.ajax({
    url : 'https://'
      + serverIP

```

```

        +
        '/fm/fmrest/WebZoneManager/registerAndPopulateLocalZoneDBGet/?queryKey='
        + key + '&fabricDBID=' + fabricDBID
        + '&clientRequestID=' + clientRequestID
        + '&cacheBust=' + cacheBust + '&navId='
        + navID,
    type : 'GET',
    headers : {
        "Content-Type" : "application/x-www-form-urlencoded",
        "Authorization" : authorization,
        "Dcnm-Token" : dcnm_Token
    },
    success : function(data, textStatus, xhr) {

        console.log("data " + data);
        console.log(textStatus);
        var query_Key = JSON.stringify(data);
        var query = JSON.parse(query_Key);
        queryKey = query.requestKey;
        createZoneset(queryKey, zoneSetName, fabricDBID)
    },
    error : function(xhr, textStatus, errorThrown) {
        console.log(textStatus);
        document.getElementById("result").innerHTML = " "
        + textStatus;
    }
    });
}

// 1. Creating Zoneset
function createZoneset(queryKey, zoneSetName, fabricDBID) {
    debugger;
    var payload = {};
    payload['queryKey'] = queryKey;
    payload['zoneSetName'] = zoneSetName;
    payload['fabricDBID'] = fabricDBID;
    $.ajax({
        url : 'https://'
            + serverIP
            + '/fm/fmrest/WebZoneManager/createZoneSetMod',
        type : 'POST',
        headers : {
            "Content-Type" : "application/x-www-form-urlencoded",
            "Authorization" : authorization,
            "Dcnm-Token" : dcnm_Token
        },
        data : payload,
        success : function(data, textStatus, xhr) {
            debugger;
            console.log(data);
            document.getElementById("result").innerHTML = data.status
                + "- Create Zoneset ";
            createZone(queryKey, zoneName, fabricDBID);
        },
        error : function(xhr, textStatus, errorThrown) {
            console.log(textStatus);
            document.getElementById("result").innerHTML = " "
                + textStatus;
        }
    });
}

// 2. Creating Zone
function createZone(queryKey, zoneName, fabricDBID) {

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

var payload = {};
payload['queryKey'] = queryKey;
payload['zoneName'] = zoneName;
payload['fabricDBID'] = fabricDBID;
$
    .ajax({
        url : 'https://' + serverIP
            + '/fm/fmrest/WebZoneManager/createZoneMod',
        type : 'POST',
        headers : {
            "Content-Type" : "application/x-www-form-urlencoded",
            "Authorization" : authorization,
            "Dcnm-Token" : dcnm_Token
        },
        data : payload,
        success : function(data, textStatus, xhr) {
            debugger;
            console.log(data);
            document.getElementById("result1").innerHTML = data.status
                + "- Create Zone";
            addZonesToZoneset(queryKey, zoneName,
                zoneSetName, fabricDBID);
        },
        error : function(xhr, textStatus, errorThrown) {
            console.log(textStatus);
            document.getElementById("result1").innerHTML = " "
                + textStatus;
        }
    });
}

// 3. Adding Zones to Zoneset
function addZonesToZoneset(queryKey, zoneNames, zonesetName,
    fabricDBID) {
    var payload = {};
    payload['queryKey'] = queryKey;
    payload['zoneNames'] = zoneNames;
    payload['zonesetName'] = zoneSetName;
    payload['fabricDBID'] = fabricDBID;
    $
        .ajax({
            url : 'https://'
                + serverIP
                + '/fm/fmrest/WebZoneManager/addZonesToZonesetMod',
            type : 'POST',
            headers : {
                "Content-Type" : "application/x-www-form-urlencoded",
                "Authorization" : authorization,
                "Dcnm-Token" : dcnm_Token
            },
            data : payload,
            success : function(data, textStatus, xhr) {
                debugger;
                console.log(data);
                document.getElementById("result2").innerHTML = data.status
                    + "- Adding Zones to Zoneset ";
                deleteZonesFromZoneset(queryKey, zoneName,
                    zoneSetName, fabricDBID);
            },
            error : function(xhr, textStatus, errorThrown) {
                console.log(textStatus);
                document.getElementById("result2").innerHTML = " "
                    + textStatus;
            }
        });
}

```

```

    }
// 4. Deleting Zones from zoneset
function deleteZonesFromZoneset(queryKey, zoneNames, zonesetName,
    fabricDBID) {
    debugger;
    var payload = {};
    payload['queryKey'] = queryKey;
    payload['zoneName'] = zoneNames;
    payload['zonesetName'] = zonesetName;
    payload['fabricDBID'] = fabricDBID;
    $
        .ajax({
            url : 'https://'
                + serverIP
                + '/fm/fmrest/WebZoneManager/deleteZonesFromZonesetMod',
            type : 'POST',
            headers : {
                "Content-Type" : "application/x-www-form-urlencoded",
                "Authorization" : authorization,
                "Dcnm-Token" : dcnm_Token
            },
            data : payload,
            success : function(data, textStatus, xhr) {
                debugger;
                console.log(data);
                document.getElementById("result3").innerHTML = data.status
                    + "- Deleting Zones from Zoneset ";
                deleteZones(queryKey, zoneName, fabricDBID);
            },
            error : function(xhr, textStatus, errorThrown) {
                console.log(textStatus);
                document.getElementById("result3").innerHTML = " "
                    + textStatus;
            }
        })
    }
// 5. Deleting Zones
function deleteZones(queryKey, zoneName, fabricDBID) {
    debugger;
    var payload = {};
    payload['queryKey'] = queryKey;
    payload['zoneName'] = zoneName;
    payload['fabricDBID'] = fabricDBID;
    $
        .ajax({
            url : 'https://'
                + serverIP
                + '/fm/fmrest/WebZoneManager/deleteZonesMod',
            type : 'POST',
            headers : {
                "Content-Type" : "application/x-www-form-urlencoded",
                "Authorization" : authorization,
                "Dcnm-Token" : dcnm_Token
            },
            data : payload,
            success : function(data, textStatus, xhr) {
                debugger;
                console.log(data);
                document.getElementById("result4").innerHTML = data.status
                    + "- Delete Zone ";
                deleteZoneset(queryKey, zoneSetName, fabricDBID);
            },
            error : function(xhr, textStatus, errorThrown) {
                console.log(textStatus);
            }
        })
    }

```

REVIEW DRAFT—CISCO CONFIDENTIAL

```

        document.getElementById("result4").innerHTML = " "
            + textStatus;
    }
    });
}

//6. Delete Zoneset
function deleteZoneset(queryKey, zonesetName, fabricDBID) {
    debugger;
    var payload = {};
    payload['queryKey'] = queryKey;
    payload['zonesetName'] = zonesetName;
    payload['fabricDBID'] = fabricDBID;
    $.ajax({
        url : 'https://'
            + serverIP
            + '/fm/fmrest/WebZoneManager/deleteZonesetsMod',
        type : 'POST',
        headers : {
            "Content-Type" : "application/x-www-form-urlencoded",
            "Authorization" : authorization,
            "Dcnm-Token" : dcnm_Token
        },
        data : payload,
        success : function(data, textStatus, xhr) {
            debugger;
            console.log(data);
            document.getElementById("result5").innerHTML = data.status
                + "- Delete ZoneSet ";
        },
        error : function(xhr, textStatus, errorThrown) {
            console.log(textStatus);
            document.getElementById("result5").innerHTML = " "
                + textStatus;
        }
    });
}

}
</script>
</body>
<body>
    <br>
    <br>

    <h3>Zone REST calls</h3>
    <table>
        <tr>
            <td>DCNM Server IP:</td>
            <td><input id="ip" type="text" value=""></td>
        </tr>
        <tr>
            <td>User Name:</td>
            <td><input id="un" type="text" value=""></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><input id="pw" type="password" value=""></td>
        </tr>
        <tr>
            <td>Fabric Name:</td>
            <td><input id="fabricName" type="text"
                value=""></td>
        </tr>
    </table>

```

```

        <tr>
            <td>Switch Name:</td>
            <td><input id="switchName" type="text" value=""></td>
        </tr>
        <tr>
            <td>VSAN ID:</td>
            <td><input id="vsanID" type="text" value=""></td>
        </tr>
        <tr>
            <td>ZoneSet Name:</td>
            <td><input id="zoneSetName" type="text" name="zoneSetName"
                value="" /></td>
        </tr>
        <tr>
            <td>Zone Name:</td>
            <td><input id="zoneName" type="text" name="zoneName" value="" /></td>
        </tr>
    </table>
    <br>
    <button type="button" onclick="loadDoc()">Submit</button>

<!-- below divs for displaying status messages -->
    <div id="result"></div>
    <div id="result1"></div>
    <div id="result2"></div>
    <div id="result3"></div>
    <div id="result4"></div>
    <div id="result5"></div>

</body>
</html>

```

REVIEW DRAFT – CISCO CONFIDENTIAL