DERMS Interface API Document

# Summary

This document describes the API and how to use the DERMS Interface DLL. The key to understanding this interface is the CIMData object. It contains the information necessary to send the requests. The CIMData object can be loaded from an xml configuration file, or ‘hand’ packed by the the application that calls it.

# XML Configuration File Format

This DLL uses the standard C# xml serializer to both save to the configuration file and read from it. The DERMSInterface dll supports loading the CIMData object from the configuration file. See Appendix A for example configuration file

## Headers

The file has four headers, one each for : createDER, dispatchDER, getDER, and getDERStatus

Each header contains a specific endpoint for that call, along with a custom verb. Other information, if not provided (but required by SOAP call) will be filled in.

## Groups

The configuration file stores DER Group and DER Member (children of DER Groups), including MRID, real and apparent power, feeders, substations, and so on.

## Purpose

The key to using the interface is to understand the architecture. The DLL’s methods depend on the CIMData object. The object’s headers contain the endpoint and header data structure. The object’s group list contains pre-defined (by the user) DER groups, and DER members.

The arguments passed to the DLL methods are for the purpose of find the correct DER group within the CIMData object, which is then used to make the SOAP call.

Note: The dispatchDERGroup method violates the above contract, specifically for the EnterpriseConfig application.

# DLL Method Calls

There are two ways to use the DLL.

1. Static method calls : The four method calls can be called singly, by passing the name of the config.xml file (CIMData), along with the other parameters.
2. Create a CIM object, load a configuration file, and then call any of the four methods.

## Methods

### Create DER Group

Parameters:

* (Static call only) Path – fully qualified file name (c:\foo\bar\config.xml)
* DERGroupName – name of a DERGroup that exists within the config.xml file
* (Static call only) Ref String SOAPMessage – returns the xml for the SOAP call and response
* return code – 0 for success, 1 for failure
* exceptions – will throw exceptions for misconfigure CIMData

SOAP Call : changeDERGroup -> CreateDERGroup

public static int CreateDERGroup(string path, String DERGroupName, ref String SOAPMessage)

public int CreateDERGroup(String DERGroupName);

### Dispatch DER Group

Parameters:

* (Static call only) Path – fully qualified file name (c:\foo\bar\config.xml)
* DERGroupName – name of a DERGroup that exists within the config.xml file
* Quantity (enum) – realPower or ApparentPower
* (Static call only) Ref String SOAPMessage – returns the xml for the SOAP call and response
* isOverride – if set true, the value dispatched is overrideValue, instead of the value from the config.xml
* overrideValue – value sent to server if isOverride is true.
* Return code – 0 for success, 1 for failure
* Exceptions – will throw exceptison for misconfigured CIMData

SOAP Call : changeDERGroupDispatch -> CreateDERGroupDispatch

public static int DispatchDERGroup(String path, String DERGroupName, quantity q, ref String SOAPMessage, Boolean isOverride = false, double overrideValue = 0.0)

public int DispatchDERGroup(String DERGroupName, quantity q, Boolean isOverride = false, double overrideValue = 0.0)

### Get DER Group Status

Parameters :

* (Static call only) Path – fully qualified file name (c:\foo\bar\config.xml)
* MRID – the MRID of the DER Group for which status is desired
* Quantity (enum) – realPower or ApparentPower
* (Static call only) Ref String SOAPMessage – returns the xml for the SOAP call and response
* Returns DERStatus – see appendix B

SOAP Call : GETDERGroupStatus -> GETDERGroupStatus

public static DERStatus getDERGroupStatus(String path, String mrid, quantity q, ref String SOAPMessage)

public DERStatus getDERGroupStatus(String mrid, quantity q)

### Request DER Group Members

Parameters:

* (Static call only) Path – fully qualified file name (c:\foo\bar\config.xml)
* MRID – the MRID (unique ID) of the DER Group containing the DER members to be returned
* (Static call only) Ref String SOAPMessage – returns the xml for the SOAP call and response
* Returns array of strings containing names of DER members

SOAP Call : GetDERGroup -> GetDERGroup

public static String[] requestDERGroupMembers(String path, String mrid, ref String SOAPMessage)

public String[] requestDERGroupMembers(String mrid);

### Load config file

This is a convenience method. It loads data from a config file and returns a CIM object. The CIM object can then call the four SOAP methods.

Parameters:

* Path – fully qualified file name (c:\foo\bar\config.xml)
* Returns CIM object

public static CIM loadConfigFile(string path)

### Example Code

#### Example 1 : Static Call

int rc = 0;

String log;

rc = CIM.CreateDERGroup(“c:\foo\bar\config.xml”, “DERGroup1”, null, log);

Console.Out.WriteLine(“SOAP Messages : “ + log);

#### Example 2 : Object Call

int rc = 0;

CIM c = CIM.loadConfig(“c:\foo\bar\config.xml”);

rc = c.CreateDERGroup(“DERGroup1”, null);

Console.Out.WriteLine(“Soap Call : “ + c. LastMessageSent);

Console.Out.WriteLine(“Soap Response : “ + c.LastMessageReceived);

#### More Code Examples

See **Appendix C** for Unit Tests that further demonstrate how to call the methods.

# Appendix A

Example configuration file:

<?xml version="1.0"?>

<CIMData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Headers>

<header>

<Name>createDER</Name>

<EndPoint>http://10.3.253.100:8080/DERConnect/cim/changeDERGroup</EndPoint>

<Verb>create</Verb>

<Noun>DER Request</Noun>

<ReplyAddress>http://localhost:8080/DERConnect/cim/Forecasts</ReplyAddress>

<Context>Testing</Context>

<Comment>general comments</Comment>

<AckRequired>true</AckRequired>

<UserOrganization>Epri</UserOrganization>

<UserID>epri1</UserID>

</header>

<header>

<Name>dispatchDER</Name>

<EndPoint>http://10.3.253.100:8080/DERConnect/cim/changeDERGroupDispatch</EndPoint>

<Verb>create</Verb>

<Noun>DER Request</Noun>

<ReplyAddress>http://10.3.253.100:8080/DERConnect/cim/requestChangeDERGroupDispatch</ReplyAddress>

<Context>dispatch message</Context>

<Comment>dispatch test comment</Comment>

<AckRequired>false</AckRequired>

<UserOrganization>Epri</UserOrganization>

<UserID>epri</UserID>

</header>

<header>

<Name>getDER</Name>

<EndPoint>http://10.3.253.100:8080/DERConnect/cim/getDERGroup</EndPoint>

<Verb>get</Verb>

<Noun>DER Get</Noun>

<ReplyAddress>http://localhost:8080/DERConnect/cim/RequestGetDERGroup</ReplyAddress>

<Context>get DER</Context>

<Comment>test get der</Comment>

<AckRequired>false</AckRequired>

<UserOrganization>Epri</UserOrganization>

<UserID>epriUser</UserID>

</header>

<header>

<Name>getDERStatus</Name>

<EndPoint>http://10.3.253.100:8080/DERConnect/cim/getDERGroupStatus</EndPoint>

<Verb>get</Verb>

<Noun>DER Request</Noun>

<ReplyAddress>http://localhost:8080/DERConnect/cim/requestGetDERGroupStatus</ReplyAddress>

<Context>status get</Context>

<Comment>status request</Comment>

<AckRequired>false</AckRequired>

<UserOrganization>epri</UserOrganization>

<UserID>epriUser</UserID>

</header>

</Headers>

<Groups>

<DERGroup>

<GroupName>dergroup1</GroupName>

<Mrid>123</Mrid>

<Revision>0.1a</Revision>

<Substation>hawk</Substation>

<Feeder>grifin</Feeder>

<Segment>23a</Segment>

<Devices>

<device>

<Name>der1</Name>

<WattCapacity>75.77</WattCapacity>

<VarCapacity>82.01</VarCapacity>

</device>

<device>

<Name>der2</Name>

<WattCapacity>167.22</WattCapacity>

<VarCapacity>100</VarCapacity>

</device>

<device>

<Name>abc</Name>

<WattCapacity>12.5</WattCapacity>

<VarCapacity>15.77</VarCapacity>

</device>

</Devices>

</DERGroup>

<DERGroup>

<GroupName>dergroup2</GroupName>

<Mrid>128</Mrid>

<Revision>0.1a</Revision>

<Substation>hawk</Substation>

<Feeder>grifin</Feeder>

<Segment>23b</Segment>

<Devices>

<device>

<Name>h1</Name>

<WattCapacity>73</WattCapacity>

<VarCapacity>174.8</VarCapacity>

</device>

<device>

<Name>h2</Name>

<WattCapacity>83.5</WattCapacity>

<VarCapacity>180.2</VarCapacity>

</device>

</Devices>

</DERGroup>

</Groups>

<Version>

<Major>1</Major>

<Minor>1</Minor>

<Revision>1</Revision>

<Date>2015-11-08T14:27:25.0745814-05:00</Date>

</Version>

</CIMData>

# Appendix B

public class DERStatus

{

private float presentMaxCapability = 0;

public float PresentMaxCapability

{

get { return presentMaxCapability; }

set { presentMaxCapability = value; }

}

private float presentValue = 0;

public float PresentValue

{

get { return presentValue; }

set { presentValue = value; }

}

private float presentMinCapability = 0;

public float PresentMinCapability

{

get { return presentMinCapability; }

set { presentMinCapability = value; }

}

private string DERGroupID;

public string DERGroupID1

{

get { return DERGroupID; }

set { DERGroupID = value; }

}

private int quantity = 0;

public int Quantity

{

get { return quantity; }

set { quantity = value; }

}

private string mrid;

public string Mrid

{

get { return mrid; }

set { mrid = value; }

}

}

# Appendix C

## Unit Test

These tests were specifically written to work with the responses from the test DER server. The examples will fail assertions with another server, but the example code is instructive.

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace DERMSTest

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

DERMSInterface.CIM c = DERMSInterface.CIM.loadConfigFile("testConfig.xml");

int rc = c.createDERGroup("dergroup1", null);

Console.WriteLine("xml : " + c.LastMessageSent + c.LastMessageReceived);

Assert.IsTrue(c.LastMessageSent.Length > 100);

Assert.IsTrue(c.LastMessageReceived.Length > 100);

Assert.AreEqual(rc, 0);

}

[TestMethod]

public void TestMethod2()

{

DERMSInterface.CIM c = DERMSInterface.CIM.loadConfigFile("testConfig.xml");

int rc = c.DispatchDERGroup("dergroup1", DERMSInterface.quantity.RealPower);

Console.WriteLine("xml : " + c.LastMessageSent + c.LastMessageReceived);

Assert.IsTrue(c.LastMessageSent.Length > 100);

Assert.IsTrue(c.LastMessageReceived.Length > 100);

Assert.AreEqual(rc, 0);

}

[TestMethod]

public void TestMethod3()

{

DERMSInterface.CIM c = DERMSInterface.CIM.loadConfigFile("testConfig.xml");

DERMSInterface.CIMData.DERStatus status = c.getDERGroupStatus("1234-5678", DERMSInterface.quantity.RealPower);

Console.WriteLine("xml : " + c.LastMessageSent + c.LastMessageReceived);

Assert.IsTrue(c.LastMessageSent.Length > 100);

Assert.IsTrue(c.LastMessageReceived.Length > 100);

Assert.AreEqual(status.PresentValue, (float)11.1);

}

[TestMethod]

public void TestMethod4()

{

DERMSInterface.CIM c = DERMSInterface.CIM.loadConfigFile("testConfig.xml");

String[] derms = c.requestDERGroupMembers("1234-5678");

Console.WriteLine("xml : " + c.LastMessageSent + c.LastMessageReceived);

Assert.IsTrue(c.LastMessageSent.Length > 100);

Assert.IsTrue(c.LastMessageReceived.Length > 100);

Assert.IsNotNull(derms);

Assert.AreEqual(derms.Length, 1);

Assert.IsTrue(derms[0].Equals("name1"));

}

[TestMethod]

public void TestMethod5()

{

String message = "";

int rc = DERMSInterface.CIM.CreateDERGroup("testConfig.xml", "dergroup1", null, ref message);

Console.WriteLine("xml : " + message);

Assert.IsTrue(message.Length > 100);

Assert.AreEqual(rc, 0);

}

[TestMethod]

public void TestMethod6()

{

String message = "";

int rc = DERMSInterface.CIM.DispatchDERGroup("testConfig.xml", "dergroup1", DERMSInterface.quantity.RealPower, ref message, false, 0.0);

Console.WriteLine("xml : " + message);

Assert.IsTrue(message.Length > 100);

Assert.AreEqual(rc, 0);

}

[TestMethod]

public void TestMethod7()

{

String message = "";

DERMSInterface.CIMData.DERStatus status = DERMSInterface.CIM.getDERGroupStatus("testConfig.xml", "1234-5678", DERMSInterface.quantity.RealPower, ref message);

Console.WriteLine("xml : " + message);

Assert.IsTrue(message.Length > 100);

Assert.AreEqual(status.PresentValue, (float)11.1);

}

[TestMethod]

public void TestMethod8()

{

String message = "";

DERMSInterface.CIM c = DERMSInterface.CIM.loadConfigFile("testConfig.xml");

String[] derms = DERMSInterface.CIM.requestDERGroupMembers("testConfig.xml", "1234-5678", ref message);

Console.WriteLine("xml : " + message);

Assert.IsTrue(message.Length > 100);

Assert.IsNotNull(derms);

Assert.AreEqual(derms.Length, 1);

Assert.IsTrue(derms[0].Equals("name1"));

}

}

}