# Instructions for Python call of DERMSInterface.dll

This document contains example code for calling the DERMSInterface from python scripts. The code provided is for example purposes only. The test python code uses ‘ctypes’ to interface with the dlls.

# Fast Start

You can skip this document and do the following instead:

* Start up a DER SOAP server (note, need customer server at this point, awaiting latest client changes to formats)
* Edit app.config (set endpoints appropriately)
* Z:\opendms\test> cd opendms\test
* Z:\opendms\test> python createDERGroup.py
* Z:\opendms\test> python requestDERGroupMembers.py
* Z:\opendms\test> python dispatchDER.py
* Z:\opendms\test> python getDERGroupStatus.py

# Files

* DERMS\* - dll, lib, etc…
* Instructions.docs – this document
* app.config – configuration file. Edit and set with your endpoints
* enterpriseConfig.xml – configuration file created by Enterprise Config application. Is used and REQUIRED by createDERGroup.py and dispatchDER.py
* createDERGroup.py – test python script that calls dll and creates a DER Group
* dispatchDER.py – test python script that calls dll and dispatches a DER
* getDERGroupStatus.py – test python script that calls dll and get DER Group Status
* requestDERGroupMembers.py – test python script that calls dll and gets names of DERs in DERGroup

# Python Setup

## Import ctypes, required for dll interface

from sys import exit

from ctypes import \*

## Load the dll

If the python script is in a different directory, use a qualified path name.

The variable ‘a’ can be thought of as an object that points to the methods that are to be called.

All access to the DLL must be done through dot notation, ie: a.getDERRequest(args…)

a = ctypes.cdll.LoadLibrary("DERMSInterface.dll")

## Load dotNet Configuration file

dotNet requires a configuration file before the external methods can be called. This configuration file specifically points to the SOAP endpoints. Modify this file such that the endpoints identify the DER server of choice.

The loadAppconfig method loads the app config file into memory.

a.loadAppConfig('Z:/git/opendms/test/app.config');

## Declaring/Using Python Variables

* foo = c\_wchar\_p() # string
* foo = c\_double(71.5) # double/float
* byref(foo) # pass reference of value to function b
* s = cast(foo, c\_wchar\_p).value # convert unicode to ascii

# Create DER Group

int createDERGroup(path, group, ders, xml)

* path : name of DER file. See appendix A for example. May be file name if in same directory as the dll, or fully qualified path name
* group : name of a DER group within the DER file ( See appendix A). The DER group name must exist within the DER file or the operation will fail
* ders : the ders to be created within the DER group. currently supports a maximum of 9 ders. The ders must exist within the DER group (see above).
* xml : response (see appendix B)

## Code sample

The example below will create a new DERGroup called ‘dergroup1’, containing two ders, ‘der1’, ‘der2’.

Note: the DER Group ‘dergroup1’ must exist in the DER file, and ‘der1’, ‘der2’ must be ders within the DERGroup.

x = (ctypes.c\_wchar\_p \* 10)()

x[0] = c\_wchar\_p('der1')

x[1] = c\_wchar\_p('der2')

xml = c\_wchar\_p()

rc = a.createDERGroup("enterpriseConfig.xml", "dergroup1", byref(x), byref(xml))

## Excerpt from enterpriseCongfg.xml (from appendix A)

…

<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>

## Code Sample

The example below, by not defining any ders, will copy all the ders in the DER Group to the newly created DER group.

x = (ctypes.c\_wchar\_p \* 10)()

xml = c\_wchar\_p()

rc = a.createDERGroup("enterpriseConfig.xml", "dergroup1", byref(x), byref(xml))

## Working Code

opendms/test/createDERGroup.py

# Request DER Group Members

int requestDERGroupmembers(path, mrid, ders, xml)

* path : name of DER file. See appendix A for example. May be file name if in same directory as the dll, or fully qualified path name
* mrid : MRID of a DER group within the DER database (retrievable from the DER SOAP server) note: does not use the enterpriseConfig.xml file
* xml : response (see appendix B)

## Code sample

The example below will return (if they exist) dernames for the specified DERGroup.

Note: the DER Group ‘dergroup1’ must exist in the DER file, and ‘der1’, ‘der2’ must be ders within the DERGroup. Note: in the example code, I used a different calling method for this method than for createDERGroup, just because.

xml = c\_wchar\_p()

func = a.requestDERGroupMembers

func.argtypes = [c\_wchar\_p,c\_wchar\_p, POINTER(c\_char\_p)]

func.restype = c\_int

xml = c\_char\_p()

rc = func('enterpriseConfig.xml', '1234-5678', byref(xml))

## Working Code

opendms/test/requestDERGroupMembers.py

# Dispatch DER

int dispatchDERGroup(path, mrid, ders, xml)

* path : name of DER file. See appendix A for example. May be file name if in same directory as the dll, or fully qualified path name
* mrid : MRID of DER GROUP within the DER file ( See appendix A). The DER group name must exist within the DER file or the operation will fail
* quantity : RealPower or ApparentPower
* xml : response (see appendix B)
* isOverride : True, False – if False, the value from the der will be used
* overrideValue : float value, will only be used if isOverride = True.

Note : This method is currently not properly implemented. A DERGroup supports multiple DERS, which have their own values. This method does not include the DER, so what exactly is updated on server side? Unknown.

## Code sample

The example below will set a new value Real or Apparent for the DER Group identified by the MRID

Note: the DER Group ‘dergroup1’ must exist in the DER file, and ‘der1’, ‘der2’ must be ders within the DERGroup.

xml = c\_wchar\_p()

val = c\_double(240.0)

rc = a.dispatchDERGroup("enterpriseConfig.xml", "128", "RealPower", byref(xml), False, val)

## Excerpt from enterpriseCongfg.xml (from appendix A)

…

<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>

## Working Code

opendms/test/dispatchDER.py

# GET DER Group Status

Int getDERGroupStatus(configFile, mrid, quantity, xml)

* path : name of DER file. See appendix A for example. May be file name if in same directory as the dll, or fully qualified path name
* mrid : MRID group currently resident in the server
* quantity : RealPower or ApparentPower
* xml : response (see appendix B)

Note : This method is currently not properly implemented. A DERGroup supports multiple DERS, which have their own values. This method does not include the DER, so what exactly is updated on server side? Unknown.

## Code sample

The example below return information for the DERGroupName given. See Appendix A – Status tab.

Note: the DER Group ‘dergroup1’ must exist in the DER file, and ‘der1’, ‘der2’ must be ders within the DERGroup.

xml = c\_wchar\_p() **rc**= a.getDERGroupStatus("enterpriseConfig.xml", "dergroup1", "RealPower", byref(xml))

## Excerpt from enterpriseCongfg.xml (from appendix A)

…

<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>

## Working Code

opendms/test/dispatchDER.py

# Appendix A - XML Result Data

## XML Tags

* Status – fields returned from the getDERGroupStatus.py method
* Members – returned by getDER.py
* Returncode – returned by all, 0 means success
* ErrorMessage – if exceptions were thrown, they are contained here
* SOAPMessage – Message sent to SOAP server
* SOAPResponse – SOAP message response from server

<?xml version="1.0" encoding="utf-16"?>

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

<Status>

<PresentMaxCapability>12</PresentMaxCapability>

<PresentValue>11.1</PresentValue>

<PresentMinCapability>10.1</PresentMinCapability>

<Quantity>0</Quantity>

<Mrid>1234</Mrid>

</Status>

<Members>

<string>name1</string>

</Members>

<Returncode>0</Returncode>

<ErrorMessage>…</ErrorMessage>

<SOAPMessage>&lt;?xml version="1.0" encoding="utf-16"?&gt;

&lt;DERGroupRequestMessageType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;

&lt;Header xmlns="http://iec.ch/TC57/2011/DERGroupMessage"&gt;

&lt;Verb xmlns="http://iec.ch/TC57/2011/schema/message"&gt;create&lt;/Verb&gt;

&lt;Noun xmlns="http://iec.ch/TC57/2011/schema/message"&gt;DER Request&lt;/Noun&gt;

&lt;Context xmlns="http://iec.ch/TC57/2011/schema/message"&gt;Testing&lt;/Context&gt;

&lt;ReplyAddress xmlns="http://iec.ch/TC57/2011/schema/message"&gt;http://localhost:8080/DERConnect/cim/Forecasts&lt;/ReplyAddress&gt;

&lt;User xmlns="http://iec.ch/TC57/2011/schema/message"&gt;

&lt;UserID&gt;epri1&lt;/UserID&gt;

&lt;Organization&gt;Epri&lt;/Organization&gt;

&lt;/User&gt;

&lt;Comment xmlns="http://iec.ch/TC57/2011/schema/message"&gt;general comments&lt;/Comment&gt;

&lt;/Header&gt;

&lt;Payload xmlns="http://iec.ch/TC57/2011/DERGroupMessage"&gt;

&lt;DERGroups xmlns="http://www.epri.com/2013/DERGroup#"&gt;

&lt;EndDeviceGroup&gt;

&lt;mRID&gt;123&lt;/mRID&gt;

&lt;Version&gt;

&lt;major&gt;1&lt;/major&gt;

&lt;minor&gt;1&lt;/minor&gt;

&lt;revision&gt;1&lt;/revision&gt;

&lt;versionDate&gt;0001-01-01T00:00:00&lt;/versionDate&gt;

&lt;/Version&gt;

&lt;name&gt;dergroup1&lt;/name&gt;

&lt;/EndDeviceGroup&gt;

&lt;/DERGroups&gt;

&lt;/Payload&gt;

&lt;/DERGroupRequestMessageType&gt;</SOAPMessage>

<SOAPResponse>&lt;?xml version="1.0" encoding="utf-16"?&gt;

&lt;DERGroupResponseMessageType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;

&lt;Header xmlns="http://iec.ch/TC57/2011/DERGroupMessage"&gt;

&lt;Verb xmlns="http://iec.ch/TC57/2011/schema/message"&gt;create&lt;/Verb&gt;

&lt;Noun xmlns="http://iec.ch/TC57/2011/schema/message"&gt;DER Request&lt;/Noun&gt;

&lt;Context xmlns="http://iec.ch/TC57/2011/schema/message"&gt;Testing&lt;/Context&gt;

&lt;ReplyAddress xmlns="http://iec.ch/TC57/2011/schema/message"&gt;http://localhost:8080/DERConnect/cim/Forecasts&lt;/ReplyAddress&gt;

&lt;User xmlns="http://iec.ch/TC57/2011/schema/message"&gt;

&lt;UserID&gt;epri1&lt;/UserID&gt;

&lt;Organization&gt;Epri&lt;/Organization&gt;

&lt;/User&gt;

&lt;Comment xmlns="http://iec.ch/TC57/2011/schema/message"&gt;general comments&lt;/Comment&gt;

&lt;/Header&gt;

&lt;Reply xmlns="http://iec.ch/TC57/2011/DERGroupMessage"&gt;

&lt;Result xmlns="http://iec.ch/TC57/2011/schema/message"&gt;OK&lt;/Result&gt;

&lt;operationId xmlns="http://iec.ch/TC57/2011/schema/message"&gt;102799&lt;/operationId&gt;

&lt;/Reply&gt;

&lt;Payload xmlns="http://iec.ch/TC57/2011/DERGroupMessage"&gt;

&lt;DERGroups xmlns="http://www.epri.com/2013/DERGroup#"&gt;

# Appendix B – EnterpriseConfig.xml

## XML Tags

* Headers – Currently not used by this application
* Groups – DER Groups and DERS defined by user

&lt;EndDeviceGroup comment="epri cim test"&gt;

&lt;Version&gt;

&lt;major&gt;1&lt;/major&gt;

&lt;minor&gt;72&lt;/minor&gt;

&lt;revision&gt;2&lt;/revision&gt;

&lt;versionDate&gt;2015-12-17T23:30:25.169-05:00&lt;/versionDate&gt;

&lt;/Version&gt;

&lt;name&gt;testing123&lt;/name&gt;

&lt;/EndDeviceGroup&gt;

&lt;/DERGroups&gt;

&lt;/Payload&gt;

&lt;/DERGroupResponseMessageType&gt;</SOAPResponse>

</DERResult>

Appendix B – EnterpriseConfig.xml

<?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>false</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>

</CIMData>