

Remote OAC XML-RPC Interface

Beau Harrison

October 15, 2019

Abstract

The purpose of this document is to describe the process of creating an ACNET parameter and writing values to the parameter using the XML-RPC protocol. The Remote Open Access Client (OAC) serves as a simple setting interface to the Fermilab Controls System, ACNET. The creation of Remote was motivated by a need to integrate disparate experimental quality measures into the familiar Controls System for Operators and experts to monitor and respond. This has the added benefit of creating a parameter for directly communicating desired changes.

1 Introduction

XML-RPC uses HTTP to transmit an XML encoded message via the remote procedure call (RPC) protocol.¹ While XML-RPC messages could be written by hand, most major languages have a package that simplifies the message creation and transmission process.

2 Creating a Remote ACNET parameter

The easiest way to create a Remote parameter is to call and ask a Main Control Room Operator to create a copy of Z:REMOTE. Z:REMOTE is an existing Remote parameter. Parameter attributes should be changed as necessary, but a few that need to be changed are *Maintainer*, *Descriptive text*, and *Full text*. Features like status should be added or removed as needed. The new unique *Device Name* will be used when following the XML-RPC instructions.

2.1 Database application D80

To copy a device, if you have the appropriate permissions, you can enter Z:REMOTE into D80 and select Enable Edit from the bottom of the Pgm_Tools menu. Now when you overwrite the parameter name with the desired name of your new parameter a menu will appear and you select Create a new device. It will automatically populate the next entry for you. It will ask if you would like to retain the alarm blocks from Z:REMOTE. If you don't know, select No. Selecting Save Edit at the bottom of the Pgm_Tools menu will commit the new parameter to the database.

3 Using XML-RPC to write to an ACNET parameter

Find an example for your preferred language to make sure you are using the correct methods from the library. The server/client nomenclature can be confusing. The URL is the server, Remote OAC, location. The OAC has particular methods that it is able to execute. Currently, Remote.setting, Remote.status, Remote.setRaw, and setting array values are available. Remote.setting will change the setting of the given parameter and the reading will reflect the setting. Remote.status accepts a float representation of the 32-bit binary digital status. Remote.setRaw sets the raw value of a device to be scaled in ACNET. Setting array indexes and multiple device settings is also available.

¹<http://xmlrpc.scripting.com/>

3.1 Python example

```
from xmlrpc.client import ServerProxy # Import the server function from the XML-RPC library

# Setup server connection
SERVER = ServerProxy('https://www-bd.fnal.gov/xmlrpc/Remote') # This shouldn't change

# Using each method
SERVER.Remote.setting('Z:REMOTE', 10) # Sets device setting and reading to 10

SERVER.Remote.status('Z:REMOTE', float(0x1F)) # Sets the first five status bits to 1

SERVER.Remote.setRaw('Z:REMOTE', 20) # Sets the raw value to 20 to be scaled in ACNET

# Setting multiple values and array values with an array of devices
token = 10 # This is reflected in returned hash table
s = [
    # set Z:REMSE1
    {'name': 'Z:REMSE1', 'val': 10.0},
    # set Z:REMSE2 to array starting with element 0
    {'name': 'Z:REMSE2', 'val': [10.0, 20.0, 30.0, 40.0]},
    # set Z:REMOTE to array starting with element 2
    {'name': 'Z:REMOTE', 'index': 2, 'val': [10.0, 20.0]}
]

# The response from the server can be printed
print(SERVER.Remote.setting(token, s)) # Syntax for passing an array of settings
```