Quick Note 059

Use Digi ESP for Python on a TransPort router to upload PPP Stats to Digi Remote Manager as Data Points.

19 September 2017

Contents

1 INTRODUCTION

1.1 Outline

This document will describe how to push mobile statistics (ppp stats) as Data Points to Remote Manager using Digi ESP for Python. The example will use the PPP statistics but most other values can be used by modifying the driver. This will be described in the document.

The PPP statistics consist of the cellular mobile data IN and OUT combined. This is useful to show the amount of cellular data used by a device.

Please note: The document will assume that a Remote Manager account has previously been created and a Digi TransPort router has been added to this account.

To create a developer test account on Remote Manager, please use the following URL:
http://myacct.digi.com/

For help on configuring a Digi TransPort router for Remote Manager, please visit the following page:
http://knowledge.digi.com/articles/Knowledge_Base_Article/Configuring-a-Digi-TransPort-router-for-Remote-Manager-connectivity-Web-User-Interface-WebUI-method

1.2 Assumptions

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product and of the requirements for their specific application.

This quick note applies only to:

Model: Digi TransPort WR11, WR21, WR31, WR44

1.3 Corrections

Requests for corrections or amendments to this documentation are welcome and should be addressed to: tech.support@digi.com

Requests for new quick notes can be sent to the same address.

1.4 Version

Version Number	Status
1.0	Completed 13.06.2017

2 DIGI ESP FOR PYTHON INSTALLATION

Download Digi ESP for Python from the Digi Support Web site:
https://www.digi.com/support/productdetail?pid=3632&type=drivers

Start the installation and follow the instructions on the screen.

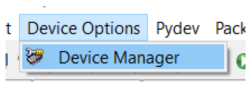
When requested to select the “**workspace**” make sure to note the location as it will be required to navigate to that directory in the next steps.

3 DIGI ESP CONFIGURATION

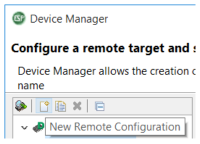
3.1 Device Manager Configuration

Start Digi ESP for Python

On the top toolbar, click on **Device Options** and click **Device Manager**

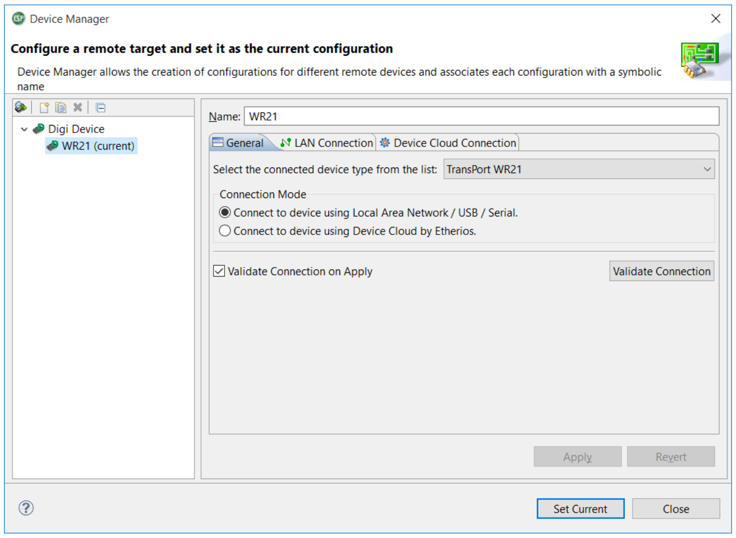


Click on **New Remote Configuration**


Device Manager
Configure a remote target and s
Device Manager allows the creation c
name
New Remote Configuration

Enter a name for this configuration, in this example, a TransPort WR21 will be used so “WR21” is used for Name.

Chose TransPort WR21 for the **device type** and **Local Area Network** for the Connection Mode



Device Manager

Configure a remote target and set it as the current configuration

Device Manager allows the creation of configurations for different remote devices and associates each configuration with a symbolic name

Digi Device
WR21 (current)

Name: WR21

General | LAN Connection | Device Cloud Connection

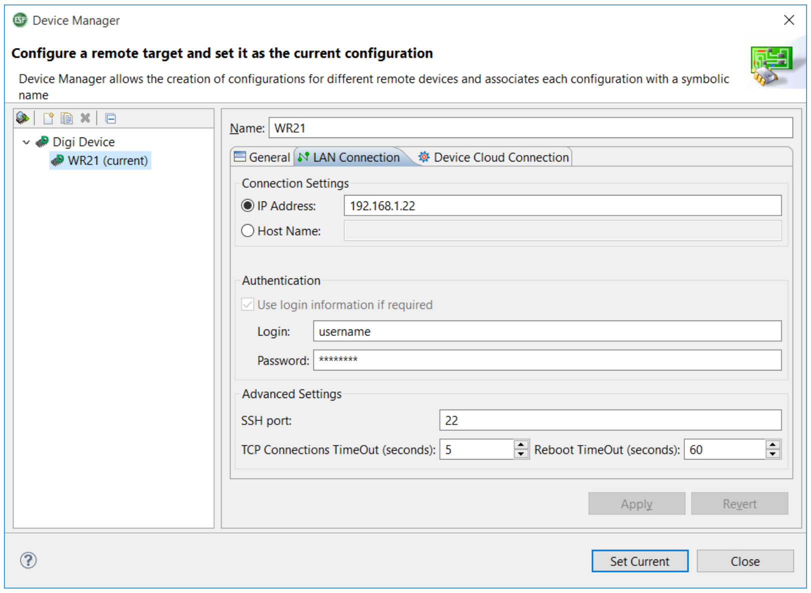
Select the connected device type from the list: TransPort WR21

Connection Mode
- (●) Connect to device using Local Area Network / USB / Serial.
- ( ) Connect to device using Device Cloud by Etherios.

[✓] Validate Connection on Apply

Under **LAN Connection**, enter the IP Address of the TransPort router.

Under **Authentication,** enter the username and password for this device. By default, “**username/password**”


Device Manager

Configure a remote target and set it as the current configuration

Device Manager allows the creation of configurations for different remote devices and associates each configuration with a symbolic name

Digi Device
WR21 (current)

Name: WR21

General | LAN Connection | Device Cloud Connection

Connection Settings
- (●) IP Address: 192.168.1.22
- ( ) Host Name:

Authentication
- [✓] Use login information if required
- Login: username
- Password: ********

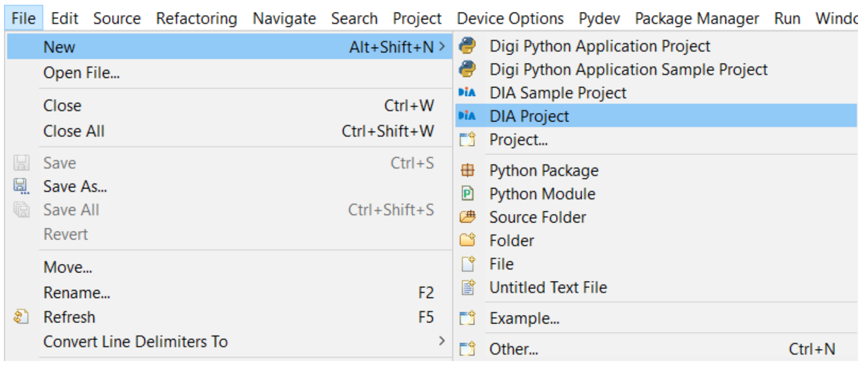
Advanced Settings
- SSH port: 22
- TCP Connections TimeOut (seconds): 5
- Reboot TimeOut (seconds): 60

Please note: It is possible to do the above steps via Remote Manager by selecting “**Connect to device using Device Cloud by Etherios**” under the General tab.

Click on **Set Current**

3.2 Create a new Project

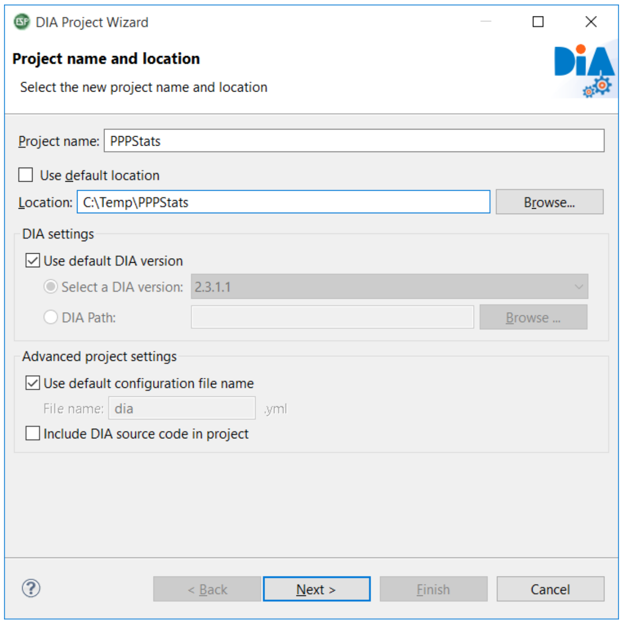
Click on **File > New > DIA Project**



Give the project a name, in this example: **PPPstats**

Select the location where this project will be saved (typically the default workspace, please note this path as it will be needed in the next step)

Make sure to check the box "**Include DIA source code in project**"


DIA Project Wizard

Project name and location

Select the new project name and location

Project name: PPPStats

☐ Use default location

Location: C:\Temp\PPPStats

DIA settings

☑ Use default DIA version

Select a DIA version: 2.3.1.1

DIA Path:

Advanced project settings

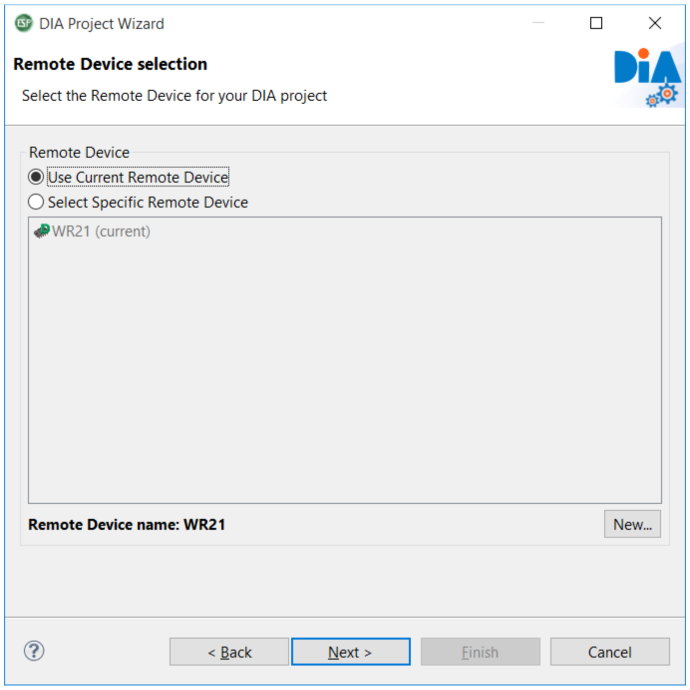
☑ Use default configuration file name

File name: dia .yml

☐ Include DIA source code in project

Click **Next**

In the next screen, select “**Use Current Remote Device**”


DIA Project Wizard

Remote Device selection

Select the Remote Device for your DIA project

Remote Device

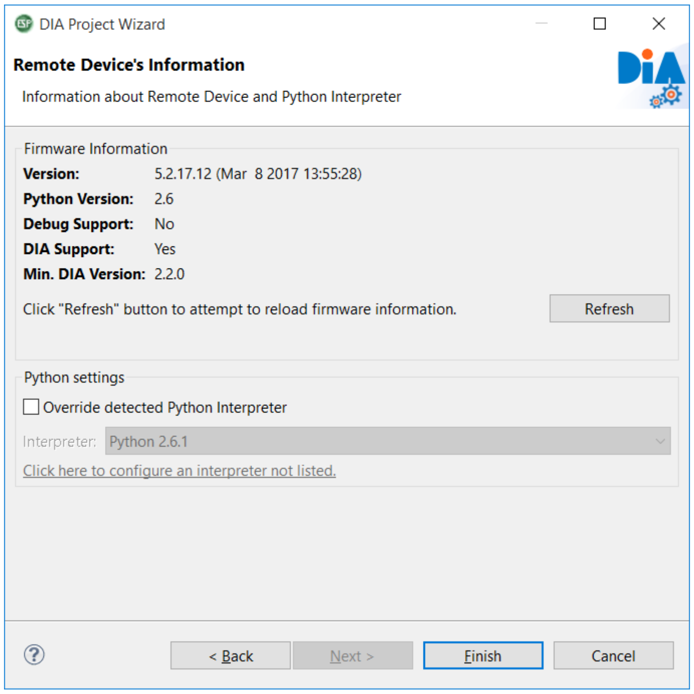
● Use Current Remote Device

○ Select Specific Remote Device

WR21 (current)

Remote Device name: WR21

Click **Finish**


DIA Project Wizard

Remote Device's Information

Information about Remote Device and Python Interpreter

Firmware Information

Version:	5.2.17.12 (Mar 8 2017 13:55:28)
Python Version:	2.6
Debug Support:	No
DIA Support:	Yes
Min. DIA Version:	2.2.0

Click "Refresh" button to attempt to reload firmware information.

Python settings

☐ Override detected Python Interpreter

Interpreter: Python 2.6.1

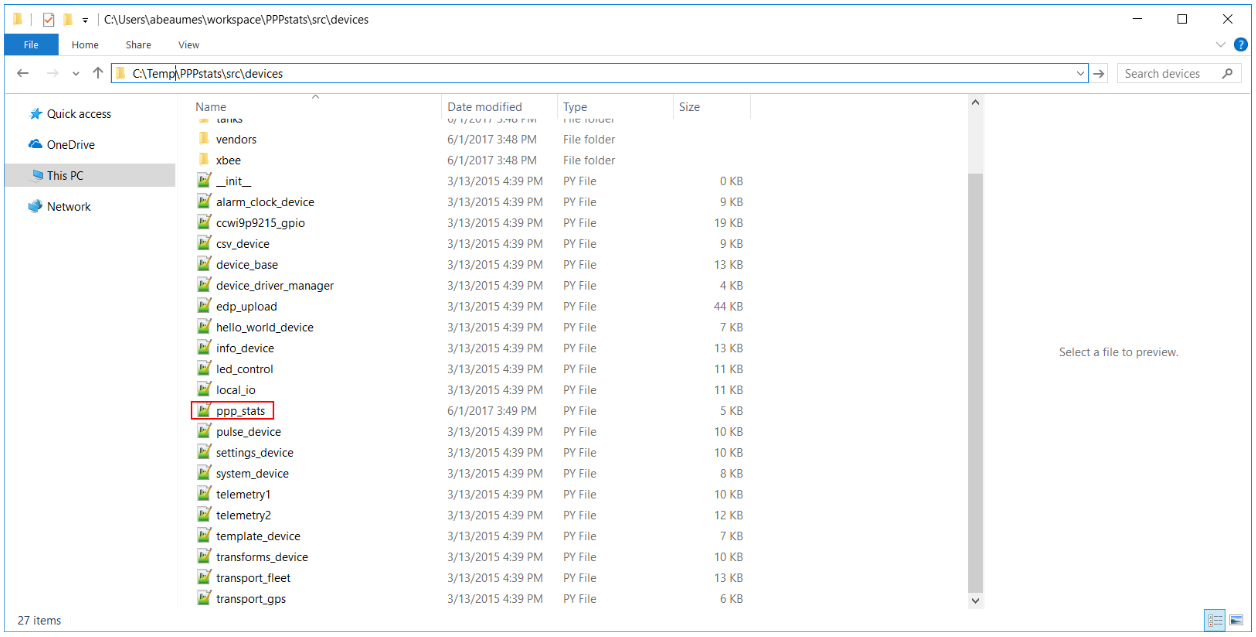
Click here to configure an interpreter not listed.

3.3 *Download and Copy the PPP Stats device*

Please note: **Make sure to close Digi ESP for Python before proceeding below.**

Download the following python file (or copy the content shown below) and paste it into the "devices" folder on the previously created project. In this example: **C:\Temp\PPPstats\src\devices\**

http://ftp1.digi.com/support/documentation/ppp_stats.zip



C:\Users\abeaumes\workspace\PPPstats\src\devices
File
Home
Share
View
C:\Temp\PPPstats\src\devices
Search devices
Quick access
OneDrive
This PC
Network
Select a file to preview.
27 items

Name	Date modified	Type	Size
tanks		File folder	
vendors	6/1/2017 3:48 PM	File folder	
xbee	6/1/2017 3:48 PM	File folder	
__init__	3/13/2015 4:39 PM	PY File	0 KB
alarm_clock_device	3/13/2015 4:39 PM	PY File	9 KB
ccwi9p9215_gpio	3/13/2015 4:39 PM	PY File	19 KB
csv_device	3/13/2015 4:39 PM	PY File	9 KB
device_base	3/13/2015 4:39 PM	PY File	13 KB
device_driver_manager	3/13/2015 4:39 PM	PY File	4 KB
edp_upload	3/13/2015 4:39 PM	PY File	44 KB
hello_world_device	3/13/2015 4:39 PM	PY File	7 KB
info_device	3/13/2015 4:39 PM	PY File	13 KB
led_control	3/13/2015 4:39 PM	PY File	11 KB
local_io	3/13/2015 4:39 PM	PY File	11 KB
ppp_stats	6/1/2017 3:49 PM	PY File	5 KB
pulse_device	3/13/2015 4:39 PM	PY File	10 KB
settings_device	3/13/2015 4:39 PM	PY File	10 KB
system_device	3/13/2015 4:39 PM	PY File	8 KB
telemetry1	3/13/2015 4:39 PM	PY File	10 KB
telemetry2	3/13/2015 4:39 PM	PY File	12 KB
template_device	3/13/2015 4:39 PM	PY File	7 KB
transforms_device	3/13/2015 4:39 PM	PY File	10 KB
transport_fleet	3/13/2015 4:39 PM	PY File	13 KB
transport_gps	3/13/2015 4:39 PM	PY File	6 KB

Content of the python file below to create it manually:

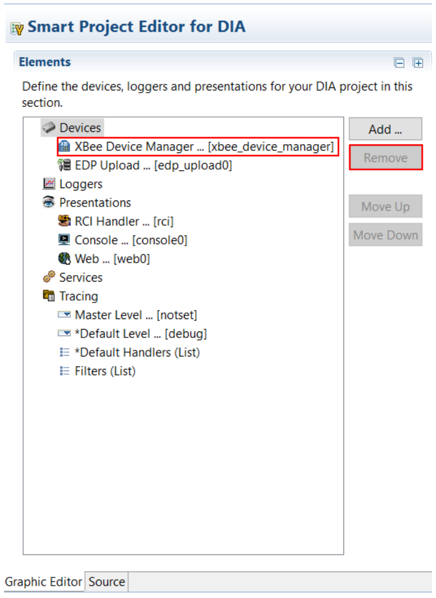
```
############################################################################
#                                                                          #
# Copyright (c)2008, 2009, Digi International (Digi). All Rights Reserved. #
#                                                                          #
# Permission to use, copy, modify, and distribute this software and its    #
# documentation, without fee and without a signed licensing agreement, is  #
# hereby granted, provided that the software is used on Digi products only #
# and that the software contain this copyright notice,  and the following  #
# two paragraphs appear in all copies, modifications, and distributions as #
# well. Contact Product Management, Digi International, Inc., 11001 Bren   #
# Road East, Minnetonka, MN, +1 952-912-3444, for commercial licensing     #
# opportunities for non-Digi products.                                     #
#                                                                          #
# DIGI SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED   #
# TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A          #
# PARTICULAR PURPOSE. THE SOFTWARE AND ACCOMPANYING DOCUMENTATION, IF ANY, #
# PROVIDED HEREUNDER IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND. #
# DIGI HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES,         #
# ENHANCEMENTS, OR MODIFICATIONS.                                          #
#                                                                          #
# IN NO EVENT SHALL DIGI BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT,      #
# SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS,   #
# ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF   #
# DIGI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.            #
#                                                                      #
########################################################################

# imports
from devices.device_base import DeviceBase
from settings.settings_base import SettingsBase, Setting
from channels.channel_source_device_property import *
from common.shutdown import SHUTDOWN_WAIT

import threading
import time
import sarcli

# constants

# exception classes

# interface functions

# classes

class PPPStatsDevice(DeviceBase, threading.Thread):
    """
    This class extends one of our base classes and is intended as an
    example of a concrete, example implementation, but it is not itself
    meant to be included as part of our developer API. Please consult the
    base class documentation for the API and the source code for this file
    for an example implementation.

    """

    def __init__(self, name, core_services):
        self.__name = name
        self.__core = core_services

        ## Settings Table Definition:
        settings_list = [
            Setting(
                name='update_rate', type=float, required=False,
default_value=1.0,
                  verify_function=lambda x: x > 0.0),
        ]

        ## Channel Properties Definition:
        property_list = [
            # gettable properties

                  ChannelSourceDeviceProperty(name="pppdata", type=str,
                initial=Sample(timestamp=0, value=""),
                perms_mask=DPROP_PERM_GET, options=DPROP_OPT_AUTOTIMESTAMP,),

        ]

        ## Initialize the DeviceBase interface:
        DeviceBase.__init__(self, self.__name, self.__core,
                                settings_list, property_list)

        ## Thread initialization:
        self.__stopevent = threading.Event()
        threading.Thread.__init__(self, name=name)
        threading.Thread.setDaemon(self, True)


    ## Functions which must be implemented to conform to the DeviceBase
    ## interface:

    def start(self):

        threading.Thread.start(self)

        return True

    def stop(self):
        self.__stopevent.set()

        self.join(SHUTDOWN_WAIT)
        if self.isAlive():
            raise RuntimeError("Could not stop %s" % self.__name)

        return True

    ## Locally defined functions:

    # Property callback functions:

    # Threading related functions:
    def run(self):

        while 1:
            if self.__stopevent.isSet():
                self.__stopevent.clear()
                break

            # increment counter property:

            pppStats = getpppStats()

            self.property_set("pppdata", Sample(0,pppStats))
            time.sleep(SettingsBase.get_setting(self,"update_rate"))

# internal functions & classes

def getpppStats():
    cli = sarcli.open()
    cli.write("at\mibs=ppp.1.dlim.totdata")
    ppp1totalstr = cli.read()
    beginningofppp1total = ppp1totalstr.find(".totdata = ")
    endofppp1total = ppp1totalstr.find("\n", (beginningofppp1total))
    ppp1total = ppp1totalstr[(beginningofppp1total+11):(endofppp1total-1)]
    cli.close()
    return ppp1total
```

Re-Open Digi ESP for Python

4 CONFIGURE DIGI ESP PROJECT

4.1 Add the new PPP stats device in the project

In the Smart Project Editor, click on the **Xbee Device Manager** and **Delete** as this will not be needed.

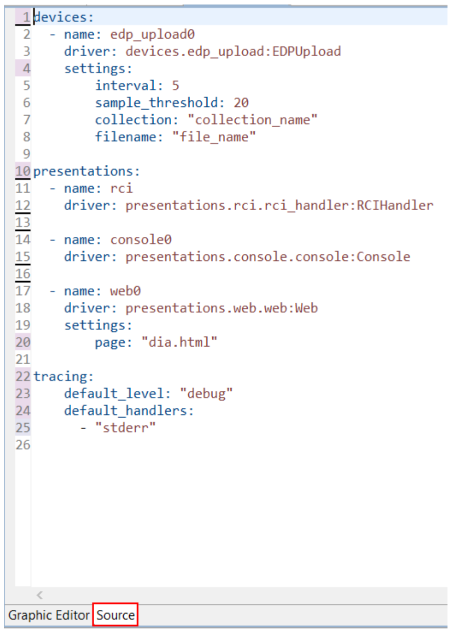


Define the devices, loggers and presentations for your DIA project in this section.

```
Devices
    XBee Device Manager ... [xbee_device_manager]
    EDP Upload ... [edp_upload0]
Loggers
Presentations
    RCI Handler ... [rci]
    Console ... [console0]
    Web ... [web0]
Services
Tracing
    Master Level ... [notset]
    *Default Level ... [debug]
    *Default Handlers (List)
    Filters (List)
```

By default, the copied device will not be available.

Now, click on **Source** to manually add the PPP stats device.

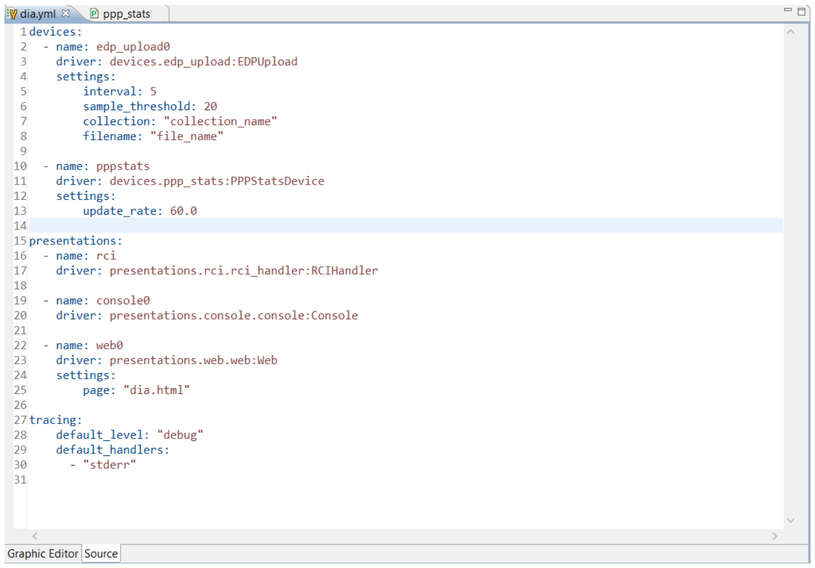


```
devices:
  - name: edp_upload0
    driver: devices.edp_upload:EDPUpload
    settings:
        interval: 5
        sample_threshold: 20
        collection: "collection_name"
        filename: "file_name"

presentations:
  - name: rci
    driver: presentations.rci.rci_handler:RCIHandler

  - name: console0
    driver: presentations.console.console:Console

  - name: web0
    driver: presentations.web.web:Web
    settings:
        page: "dia.html"

tracing:
    default_level: "debug"
    default_handlers:
      - "stderr"

```

In the “**devices:**” section, add the following:

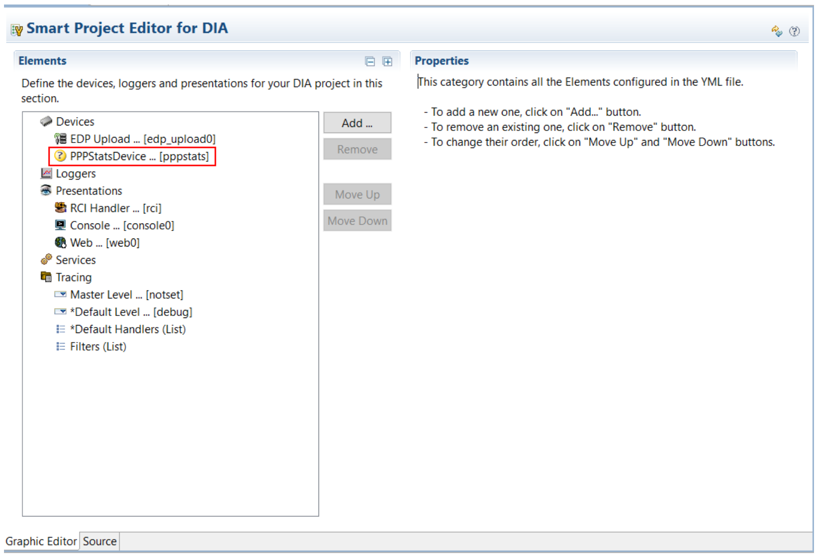
```
 - name: pppstats
   driver: devices.ppp_stats:PPPStatsDevice
   settings:
       update_rate: 60.0
```

Make sure to keep the same indenting.



```
devices:
  - name: edp_upload0
    driver: devices.edp_upload:EDPUpload
    settings:
        interval: 5
        sample_threshold: 20
        collection: "collection_name"
        filename: "file_name"

  - name: pppstats
    driver: devices.ppp_stats:PPPStatsDevice
    settings:
        update_rate: 60.0

presentations:
  - name: rci
    driver: presentations.rci.rci_handler:RCIHandler

  - name: console0
    driver: presentations.console.console:Console

  - name: web0
    driver: presentations.web.web:Web
    settings:
        page: "dia.html"

tracing:
    default_level: "debug"
    default_handlers:
      - "stderr"

```

Switch back to the “**Graphic Editor**”. A device named “**PPPStatsDevice”** should now be shown under **“Devices**”



Smart Project Editor for DIA

Elements

Define the devices, loggers and presentations for your DIA project in this section.

- Devices
  - EDP Upload ... [edp_upload0]
  - PPPStatsDevice ... [pppstats]
- Loggers
- Presentations
  - RCI Handler ... [rci]
  - Console ... [console0]
  - Web ... [web0]
- Services
- Tracing
  - Master Level ... [notset]
  - *Default Level ... [debug]
  - *Default Handlers (List)
  - Filters (List)

Properties

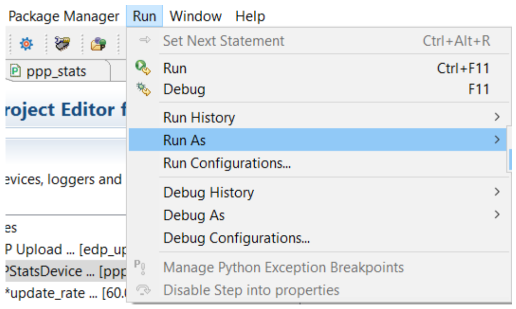
This category contains all the Elements configured in the YML file.

- To add a new one, click on "Add..." button.
- To remove an existing one, click on "Remove" button.
- To change their order, click on "Move Up" and "Move Down" buttons.

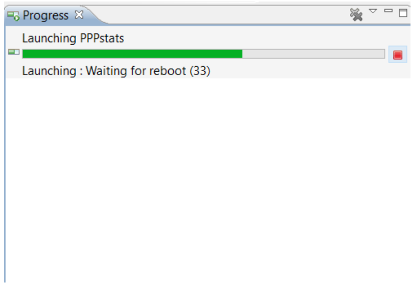
Please note: If Digi ESP shows an error message when switching back to the Graphic Editor, this is because the indenting has not been kept properly. Make sure to use only spaces and not tabs.

5 RUN DIA PROJECT ON THE TRANSPORT ROUTER

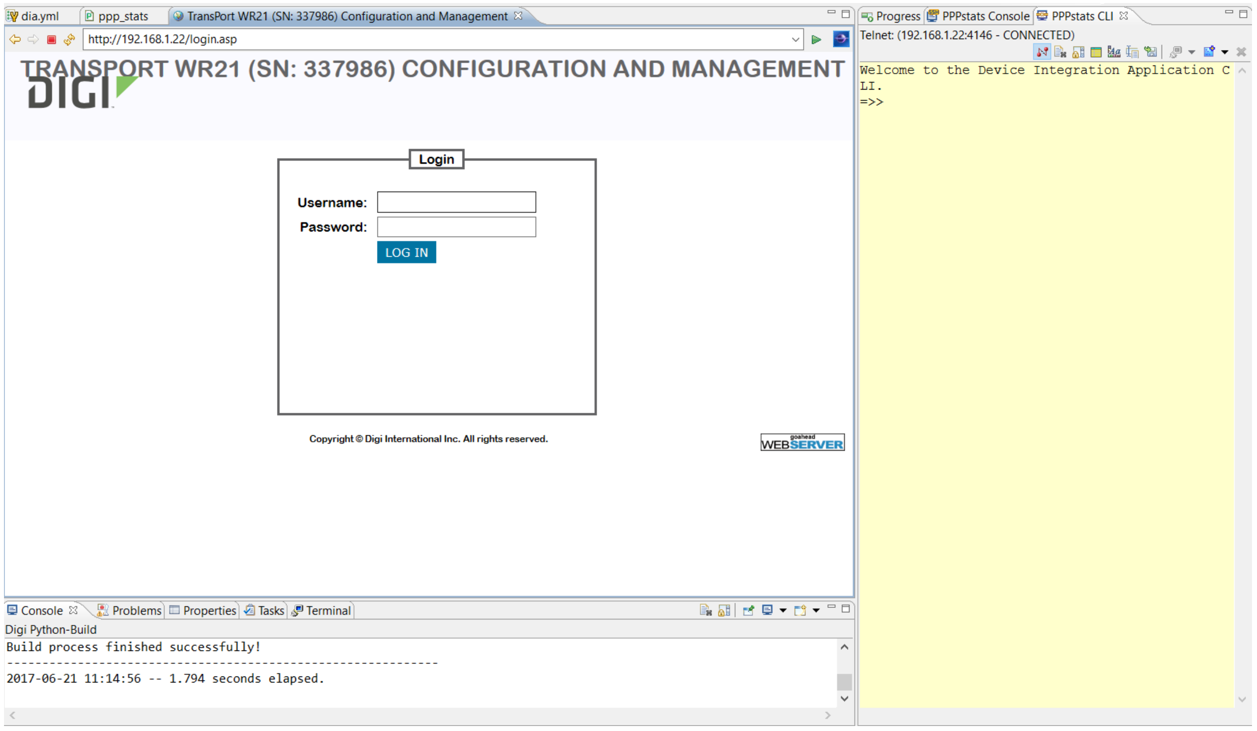
Click on **Run > Run As > Remote DIA**



The left panel will show a progress bar



Once the router has rebooted, the following will be displayed:


dia.yml
ppp_stats
TransPort WR21 (SN: 337986) Configuration and Management
http://192.168.1.22/login.asp
TRANSPORT WR21 (SN: 337986) CONFIGURATION AND MANAGEMENT
DIGI
Login
Username:
Password:
LOG IN
Copyright © Digi International Inc. All rights reserved.
WEBSERVER
Console
Problems
Properties
Tasks
Terminal
Progress
PPPstats Console
PPPstats CLI
Telnet: (192.168.1.22:4146 - CONNECTED)

```
Welcome to the Device Integration Application C
LI.
=>>
```

```
Digi Python-Build
Build process finished successfully!
--------------------------------------------------------------
2017-06-21 11:14:56 -- 1.794 seconds elapsed.
```

In the URL bar on the center page, change the url to point to:

/idigi_dia.html

In this example, http://192.168.1.22/idigi_dia.html

This will show the DIA Web presentation which is a simple web page showing the ppp stats value.

The Right panel shows a Telnet Command Line interface allowing the user to see the CLI output of the same information shown on the web page. This is done by issuing a “**channel_dump**” command:

```
Welcome to the Device Integration Application CLI.
=>> channel_dump

Device instance: edp_upload0


   Channel                     Value
Unit     Timestamp
   ------------------------ ------------------
-------- -------------------
   upload_samples            (N/A)

   upload_snapshot           (N/A)



Device instance: pppstats


   Channel                     Value
Unit     Timestamp
   ------------------------ ------------------
-------- -------------------
   pppdata                     0
         2017-06-21 09:20:40

=>>
```

Please note: For the value to increase and new data to be uploaded, the router must have a SIM card inserted and be configured to establish a cellular connection.

The **PPPstats Console** is a debugging console showing the python's script activity. Every 60sec (default value) it will show a message that data has been uploaded to Remote Manager:

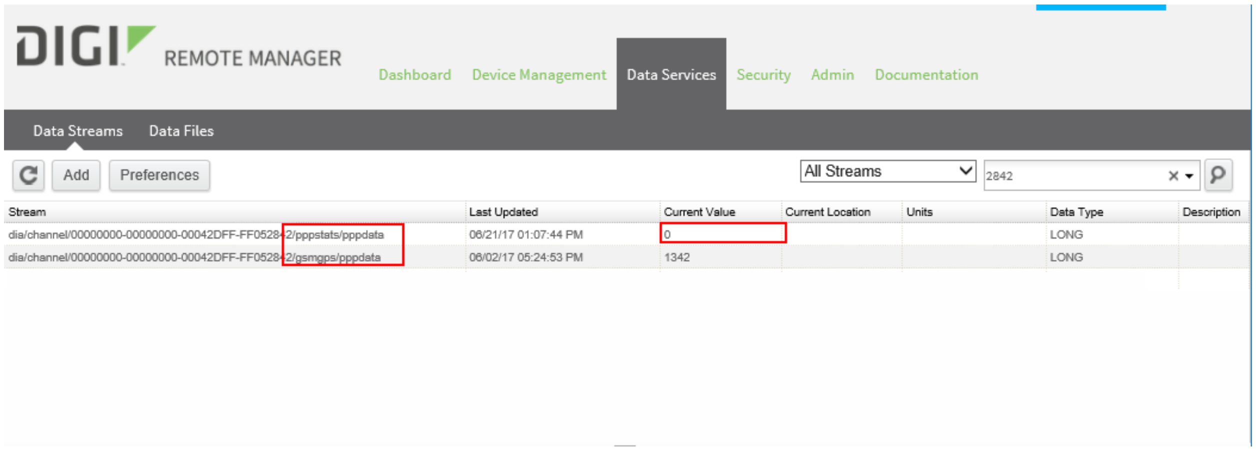
```
DEBUG:edp_upload0:Output List (1): {'pppstats.p
ppdata': (<type 'str'>, [<Sample: 0 at 2017-06-
21T09:23:40Z>])}
DEBUG:edp_upload0:<?xml version="1.0"?><idigi_d
ata compact="True" version="1.1"><sample name="
pppstats.pppdata" value="0" unit="" type="str"
timestamp="2017-06-21T09:23:40Z" />
</idigi_data>
DEBUG:edp_upload0:Starting upload to Device Clo
ud
DEBUG:edp_upload0:Attempting to upload file_nam
e19.xml to Device Cloud
DEBUG:edp_upload0:Successfully uploaded file_na
me19.xml to Device Cloud
```

6 VERIFY DATA STREAMS ON REMOTE MANAGER

Login to: https://remotemanager.digi.com/login.do using the credentials from the created account.

Navigate to **Data Services > Data Streams**

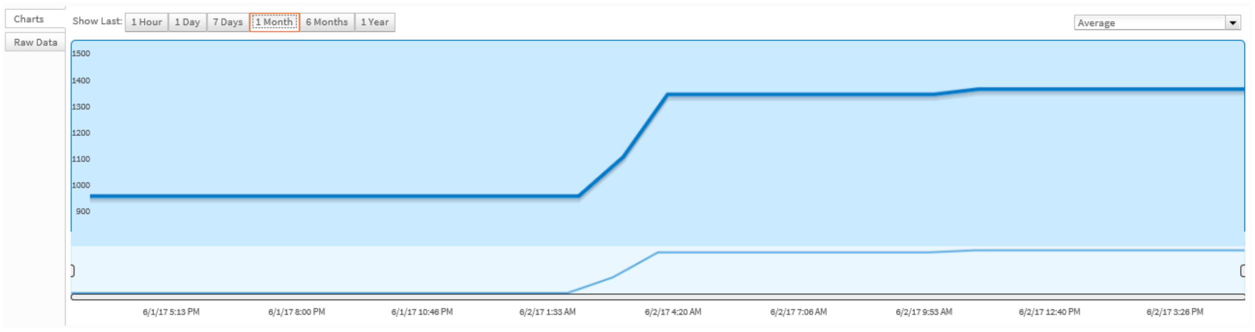
Data Streams starting with "**dia**" show be shown:



Stream	Last Updated	Current Value	Current Location	Units	Data Type	Description
dia/channel/00000000-00000000-00042DFF-FF052842/pppstats/pppdata	06/21/17 01:07:44 PM	0			LONG	
dia/channel/00000000-00000000-00042DFF-FF052842/gsmgps/pppdata	06/02/17 05:24:53 PM	1342			LONG	

The "**current value**" field is the pppdata from the router sent by the python script.

Clicking on one of the stream also allows showing this in a charts format:



7 NOTES

While this example shows the cellular data used, it is possible to modify it to use various other information.

The key part of the script that actually pulls the data from the Command Line interface of the router is found at the bottom of ppp_stats.py:

```
def getpppStats():
    cli = sarcli.open()
    cli.write("at\mibs=ppp.1.dlim.totdata")
    ppp1totalstr = cli.read()
    beginningofppp1total = ppp1totalstr.find(".totdata = ")
    endofppp1total = ppp1totalstr.find("\n", (beginningofppp1total))
    ppp1total = ppp1totalstr[(beginningofppp1total+11):(endofppp1total-1)]
    cli.close()
    return ppp1total
```

This line is the CLI command sent to the router:

```
    cli.write("at\mibs=ppp.1.dlim.totdata")
```

You can get the list of available commands on the router by issuing **at\mibs**

The number of character calculation as well as the characters to search for will need to be modified to suit the CLI command used.

Once modified, run the project as a Remote DIA like in section 5 and the new values will be available on the DIA page as well as on Remote Manager.