dd-opcda - Simple OPC DA data one way replicator

cyops-se: *This application is part of the cyops.se community and use the same language and terminology. If there are acronyms or descriptions here that are unknown or ambiguous, please visit the documentations (https://github.com/cyops-se/docs) site to see if it is explained there. You are welcome to help us improve the content regardless if you find what you are looking for or not.*

Table of Contents

Introduction

This very basic program collects all tags from an OPC DA source once a second and sends them in JSON format over UDP to a single-cast receiver IP.

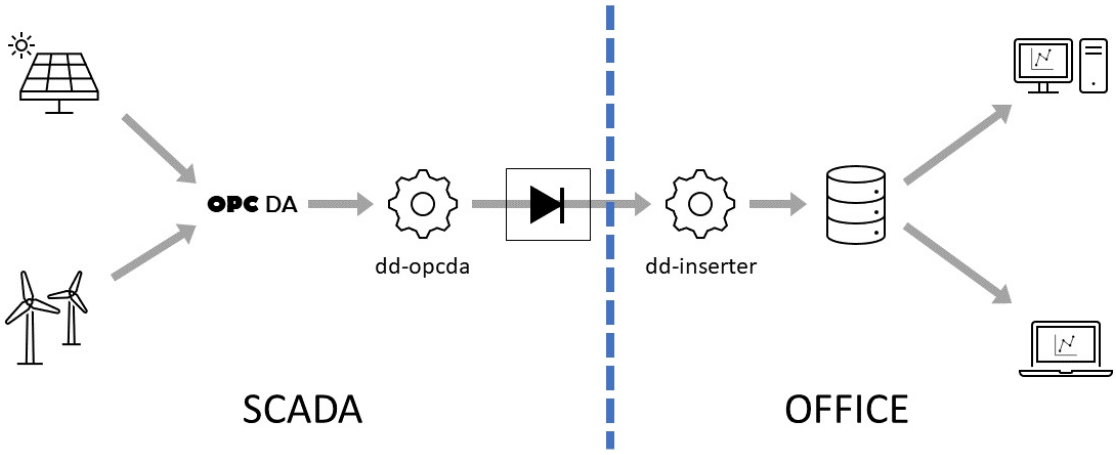
The primary use of this program is for replicating real time data from sensitive systems through a data diode to a potentially hostile network in order to achieve complete isolation of the sensitive network.

On the outer side of the data diode, use dd-inserter (https://github.com/cyops-se/dd-inserter) to store the data in a Timescale database. More one way data receivers will be added in the future.

IMPORTANT NOTE!
`dd-opcda` is dependent on OPC core components (usually provided by the local OPC server) and the Grabox OPC wrapper. Please refer to the CLI Examples section for more information! Without these pre-requisites, the application will fail!

Overview

Today it is almost impossible to keep an IACS isolated over time as the businesses operating them find the information in them valuable and even critical to run the business efficiently. Fortunately, there are ways to meet that need without compromising the security architecture. One simple way is to use a data diode as illustrated below.



Each dd-opcda process can connect to one OPC DA server and extract either all or a configured set of tags that is then sent over UDP to the receiver on the other end of the data diode. It is possible to control the following parameters using either command line arguments or a configuration file:

- OPC server prog id
- Sampling interval (default 1s)
- Branch of tags to sample
- List of tags to sample (configuration file)
- Target IP address (and port, default 4357)

It is also possible to:

- List OPC servers
- Browse specific branches of an OPC server
- Create a configuration file with tags from a specific branch
- Define a number of iterations to read the specified tags (for performance testing) in each
- Install it as a Windows service

dd-opcda is currently only able to connect to local OPC servers

Tag values are collected together with current time and quality and are sent in batches of 10 to avoid risking packet fragmentation.

Forward Error Correction

There is currently no support for forward error correction, but it is a reasonable request for future versions of this program for those that have links prone to bit errors.

Packet loss

One of the real challenges with data diodes is handling lost packets as there is no way to automatically detect and feedback to the sender for retransmission.

CLI Examples

These examples use the IntegrationObjects OPC server simulator available for free at: https://integrationobjects.com/sioth-opc/sioth-opc-servers/opc-server-simulators/.

Pre-requisites: The OPC core components (provided by the local OPC server) must be installed and the Graybox OPC Wrapper must be registered for `dd-opcda` to work properly.

Register the Graybox OPC Wrapper by copying gbda_aut.dll (probably x86 version, but you need to figure that out yourself) to c:\windows\system32 and register it with the following command as administrator:

```
c:\windows\system32\regsvr32 gbda_aut.dll
```

Command line arguments can be listed by running: `dd-opcda -h`

```
.\dd-opcda.exe -h
Usage of .\dd-opcda.exe:
  -branch string
        Lists all tags at the specified branch tag
  -c string
        Configuration file, for example declaring tags to process. If not specified, all tags will be
processed
  -cmd string
        Windows service command (try 'usage' for more info) (default "debug")
  -create
        Use this parameter to create a config file with all tags found in the specified server
  -i int
        Number of times to get all specified tags (used to measure performance) (default 1)
  -list
        Lists the OPC DA servers available on the specified source
  -p int
        Read interval in seconds (default 1)
  -port int
        The UDP port of the outer inserter (default 4357)
  -progid string
        The OPC server prog id (default "IntegrationObjects.AdvancedSimulator.1")
  -source string
        The address of the OPC server (default "localhost")
  -target string
        The address of the outer inserter (default "172.26.8.243")
  -trace
        Prints traces of OCP data to the console
```

List local OPC servers

```
> dd-opcda -list
2020/11/16 11:05:33 Found 1 server(s) on 'localhost':
2020/11/16 11:05:33 IntegrationObjects.AdvancedSimulator.1
```

List tags on specified OPC server

```
> dd-opcda -progid IntegrationObjects.AdvancedSimulator.1 -branch root
2020/11/16 11:07:35 Available tags
root
   + Random
     - Random/Text
     - Random/Date
     - Random/Boolean
     - Random/Int1
     - Random/UInt1
     - Random/Int2
     - Random/UInt2
     - Random/Int4
     - Random/UInt4
     - Random/Real4
     - Random/Real8
   + Writable
     - Writable/Text
     - Writable/Date
     - Writable/Boolean
     - Writable/Int1
     - Writable/UInt1
     - Writable/Int2
     - Writable/UInt2
     - Writable/Int4
     - Writable/UInt4
     - Writable/Real4
     - Writable/Real8
```

Create a configuration file

```
> dd-opcda.exe -progid IntegrationObjects.AdvancedSimulator.1 -branch Random -create -c config.json
2020/11/16 11:28:08 Available tags
Random
   - Random/Text
   - Random/Date
   - Random/Boolean
   - Random/Int1
   - Random/UInt1
   - Random/Int2
   - Random/UInt2
   - Random/Int4
   - Random/UInt4
   - Random/Real4
   - Random/Real8
2020/11/16 11:28:08 New configuration file with all tags created: config.json
```

Sample and send using the configuration file

```
> dd-opcda.exe -progid IntegrationObjects.AdvancedSimulator.1 -c config.json -target 172.16.21.21
Connected...
It took 0.000000 seconds to read 11 tags
It took 0.000000 seconds to read 11 tags
It took 0.000000 seconds to read 11 tags
It took 0.001011 seconds to read 11 tags
It took 0.000996 seconds to read 11 tags
Exiting ...
```

Performance test

```
> dd-opcda.exe -progid IntegrationObjects.AdvancedSimulator.1 -i 1000 -target 172.16.21.21
2020/11/16 11:34:41 Available tags
root
   + Random
     - Random/Text
     - Random/Date
     - Random/Boolean
     - Random/Int1
     - Random/UInt1
     - Random/Int2
     - Random/UInt2
     - Random/Int4
     - Random/UInt4
     - Random/Real4
     - Random/Real8
   + Writable
     - Writable/Text
     - Writable/Date
     - Writable/Boolean
     - Writable/Int1
     - Writable/UInt1
     - Writable/Int2
     - Writable/UInt2
     - Writable/Int4
     - Writable/UInt4
     - Writable/Real4
     - Writable/Real8
Connected...
It took 1.190644 seconds to read 22000 tags
It took 1.196010 seconds to read 22000 tags
It took 1.203341 seconds to read 22000 tags
It took 1.209283 seconds to read 22000 tags
It took 1.198191 seconds to read 22000 tags
It took 1.194016 seconds to read 22000 tags
Exiting ...
```