EcoStruxure™

Geo SCADA Expert

SE Datalogger

Simple Driver Development Kit Sample Driver

www.schneider-electric.com

Issue Details

Issue	Date	Author	Comments
1	07.04.21	S. Beadle	New.

References

Datalogger Web Site:

https://ecostruxure-process-instrument-datalogger.se.app/

MIT License

Table of Contents

1 Introduction

This document describes the code for a driver for Geo SCADA Expert, written in C# for the Simple Driver Framework using the Driver Development Kit (DDK).

The driver is offered as source code which you can build with Visual Studio. It includes the two parts of the driver and an installer, enabling you to build a package to deploy to Geo SCADA Expert servers. It is not supported.

The source code is available for you to freely use, modify and extend to suit your requirements or that of your clients. It is not the most optimized, efficient or elegant code, and the functionality is not assured in the way the core product is, but we hope that its simplicity will encourage engagement with the Geo SCADA driver development process and explore SE Datalogger implementation with Geo SCADA. We encourage you to add to the code by submitting 'pull requests' on GitHub.

The functionality in the driver includes basic data processing, and the feature which we hope will stimulate innovation and conversation is the built-in automatic configuration facility. This integrates well with the Geo SCADA template and instance features.

To implement and deploy this example you will need to verify functionality and add appropriate security measures for your environment.

You can discuss these features, driver development, MQTT and the Datalogger in the SE Exchange forums:

https://community.exchange.se.com/t5/Geo-SCADA-Expert-Forum/bd-p/ecostruxure-geo-scada-expert-forum

1.1 This Driver's Source Code

You will find source code for this driver within the GitHub system. The project name is Geo SCADA / Driver-SELogger:

URL: https://github.com/GeoSCADA/Driver-SELogger

1.2 The Datalogger Web API

The API specifies how to connect and retrieve data messages. For API detail, refer to the documentation on the web site https://ecostruxure-process-instrument-datalogger.se.app/

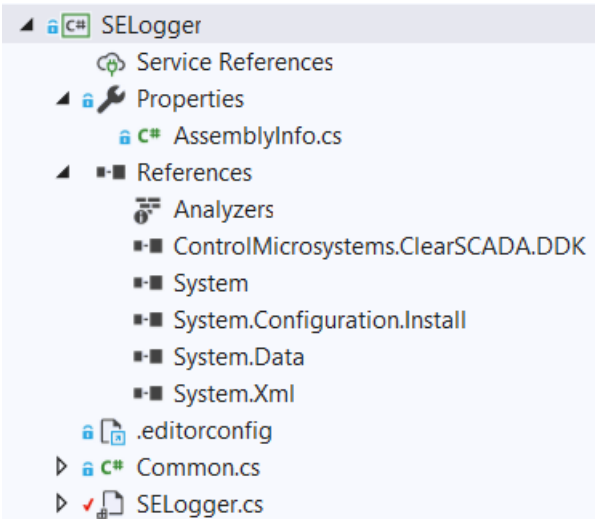
2 Building the Driver

This section lists the projects within the solution and notes on how to build them.

2.1 SELogger

This is the .dll module which defines in-database objects, properties and behaviors. The module is loaded at startup by the Geo SCADA DBServer process.

2.1.1 References



DDK

The module refers to the Geo SCADA dll 'ControlMicrosystems.ClearSCADA.DDK.dll'. You should find this reference in the project, remove it and then reinstate it from the location used on your build computer.

If you are developing on a computer which has Geo SCADA installed, find this in c:\Program Files\Schneider Electric\ClearSCADA

Alternatively, you can copy this .dll to your own location from a Geo SCADA installation. In this case you may set up multiple builds for different Geo SCADA versions if you wish. Note that you need the major version number of this .dll (e.g. 83) to match the major version of the Geo SCADA target.

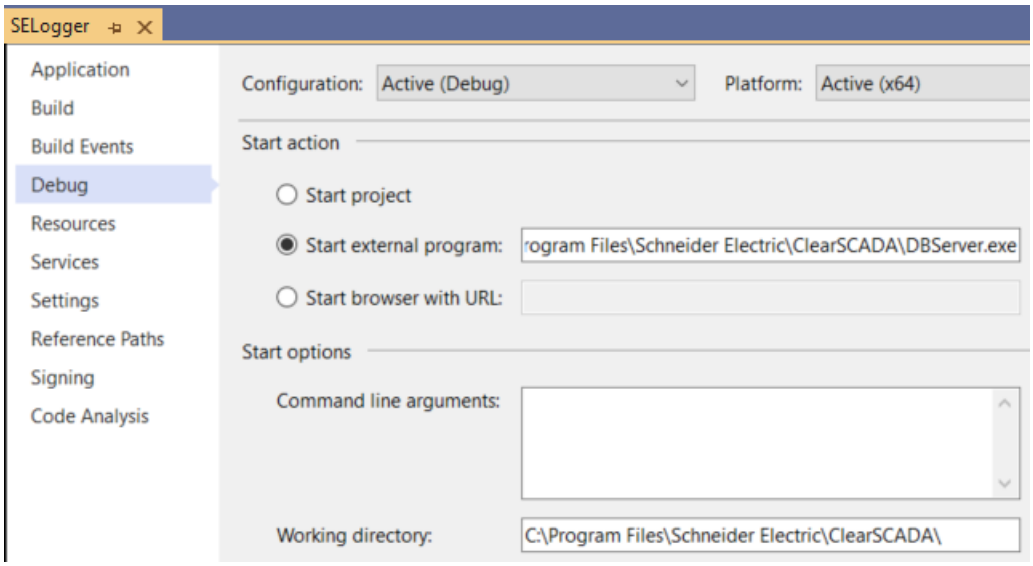
2.1.2 Files

In the file Common.cs the base number for OPC IDs is defined as `0x04691000`. Keep this unchanged, and if you add new fields please remain within the range of `0xFFF`.

The file SELogger.cs contains the object definitions and will be of most interest.

2.1.3 Debugging

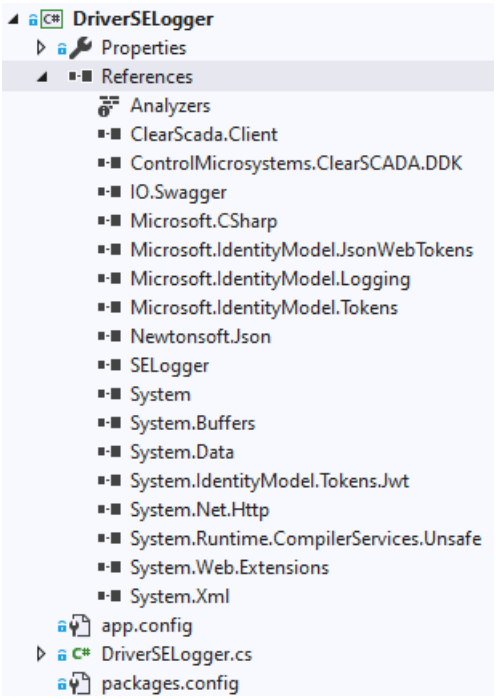
If you wish to debug this module, set the external program and working directory as shown here:



2.2 DriverSELogger

This is the .exe file which runs independently of the database and contains the functionality necessary to interact with the API and to interpret data messages.

2.2.1 References



DDK and Client

The module refers to the Geo SCADA dll 'ControlMicrosystems.ClearSCADA.DDK.dll', AND the dll 'ClearSCADA.Client'. You should find these references in the project, remove them and then reinstate them from the location used on your build computer.

If you are developing on a computer which has Geo SCADA installed, find them in c:\Program Files\Schneider Electric\ClearSCADA

Alternatively, you can copy these .dll to your own location from a Geo SCADA installation. In this case you may set up multiple builds for different Geo SCADA versions if you wish. Note that you need the major version number of these .dll (e.g. 83) to match the major version of the Geo SCADA target.

It is not usual for a driver to use the Client dll. In this case it allows database configuration as described later. By requiring this .dll the driver must be built specifically for the Geo SCADA version of the server, and not just the major version number (e.g. v83) as required for most drivers.

SELogger

This driver module requires a reference to the dll built above. You will therefore first build the SELogger project, then remove and re-add the reference to the .dll just built (default <path>\Driver-SELogger\SELogger\bin\x64\Debug\SELogger.dll).

IO.Swagger

This project has been code-generated by the Swagger web API tools specifically for the SE Logger web API version 2.0

Other NuGet Packages

Various references from Microsoft, System and Newtonsoft are included using the NuGet package manager.

2.2.2 Post-Build Operations

The driver build properties includes a copy operation at the end of build. It will copy the .exe and .dll files to the target folder. You may need to change these. The default program files location for Geo SCADA is: c:\Program Files\Schneider Electric\ClearSCADA

2.3 DriverSELoggerInstaller

This project is a basic WIX installer for the driver. It produces a .msi file for execution on a target computer. There is no upgrade capability – just remove and reinstall. Version numbers, as for the projects, are fixed and could be changed by you.

The installer places the driver .exe and .dll files into the correct location and inserts the registry entries required for the driver to run.

The installer, like the driver, is unsigned.

2.4 IO.Swagger

This project is necessary for the driver and has been created once automatically. Modifications to the driver should not require any change to this project.

3 Using the Driver

This section describes how to get started with the driver.

3.1 Installation

3.1.1 Automated Installation

Automated installation can be achieved with the installer within the development project. The installer project will copy the required files into the right place and register the DLLs.

3.1.2 Manual Installation

If you wish to install manually, this driver is installed in the same way as other DDK drivers.

Manual installation consists of copying the files into the Geo SCADA executable folder. Obtain the list of files needed from the driver .exe build command's post-build actions.

There are some registry settings needed to allow the driver to work. You can import these manually or paste this into a .reg file and open it.

```
Windows Registry Editor Version 5.00


[HKEY_LOCAL_MACHINE\SOFTWARE\Schneider Electric\ClearSCADA\DriverSELogger]
@="SELogger"
"AssemblyName"="C:\\Program Files\\Schneider Electric\\ClearSCADA\\SELogger.dll"
"DebugMode"="False"
"TaskName"="DriverSELogger.exe"
"LogEnable"="True"
```

If creating registry entries manually, use single '\' characters in the path to the AssemblyName.

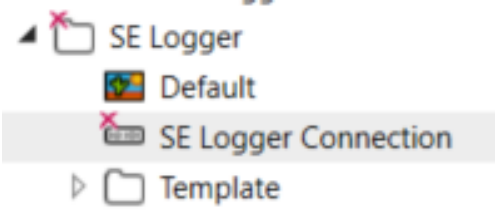
Alternatively, you can call the .Net installation command as follows:

1. Open a command prompt with administrative permissions and set the current directory to the ClearSCADA directory (usually c:\Program Files\Schneider Electric\ClearSCADA).
2. Run the Microsoft .NET Framework `InstallUtil.exe` on the DLL:

```
%WindDir%\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe SELogger.dll
```

3.2 Configure Objects

To get started it is recommended to create a structure like this:



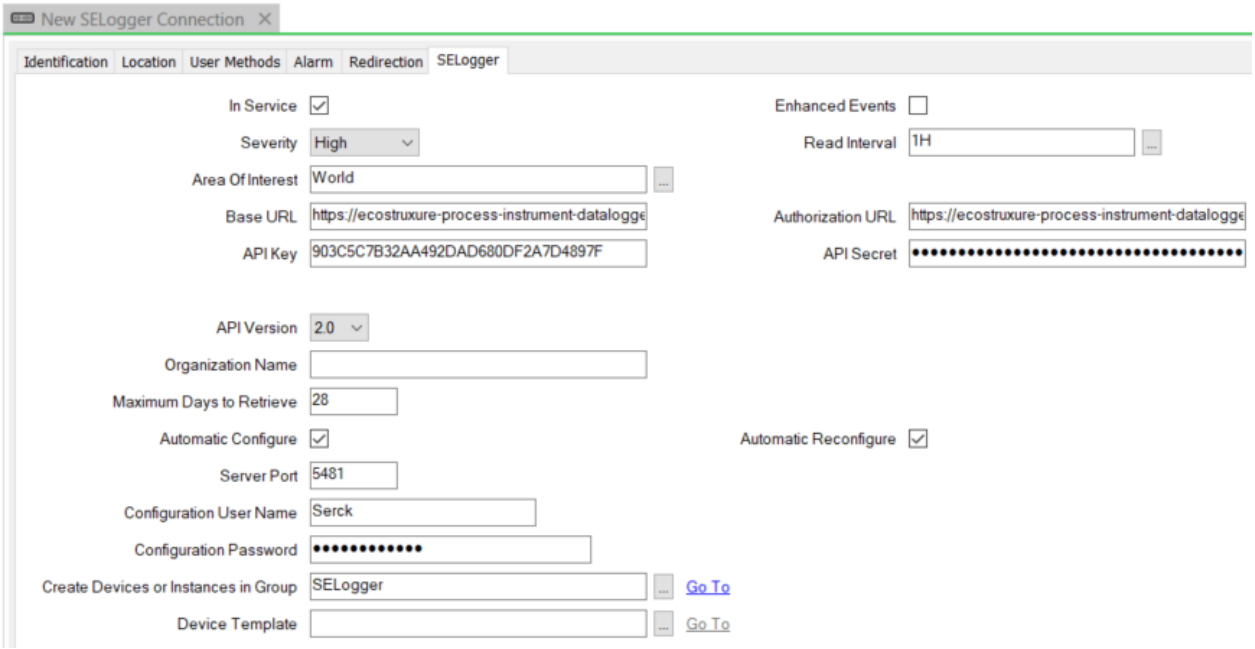
Descriptions:

- SE Logger – create a group for configuration objects
    - SE Logger Connection – create the SE Logger channel
    - Template – create a group for the common template (start with this empty).

3.3 Configure the Channel

An SE Logger Connection object defines the connection to the web service, the Group Id of the devices communicating and other characteristics of the group such as the automatic configuration features.

Example:



The top section of this object is concerned with the web service connection. Sections lower down relate to the automatic configuration of device and point objects in the database.

The fields of interest are:

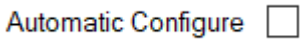
- Enhanced Events – check this to make the driver log event text to both its log file and the Geo SCADA Event log. Only enable this for short periods to diagnose issues, as leaving it enabled for long periods could cause performance issues.
- Read Interval – used to retrieve new logger and point configuration data.

- Base URL and Authorization URL – these default to the correct addresses for the internet hosted Schneider Electric datalogger web site: https://ecostruxure-process-instrument-datalogger.se.app/ If you are using a locally hosted data gathering service then these would need to be changed.
- API Key and API Secret – These are authorization fields needed to retrieve data. Get these by selecting the 'API' menu of the data logger web site and click 'Generate API Key'.
- API Version – Defaults to 2.0, and this is the only version supported.
- Organization Name – leave this blank. You may wish to use it if your account has multiple organizations and wish to filter only those loggers matching this name.
- Maximum Days to Retrieve – the driver will retrieve all data since last retrieved, but in the case of the first retrieval this sets the maximum period the driver requests, and this is therefore the oldest time for backfilled data.
- Automatic Configure – as will be described later, self-configuration is a key feature of this driver. The feature enabled by this checkbox will cause self-configuration to happen when any new device is seen. When unchecked, new nodes and devices will be placed in a queue, pending an action on the broker to cause configuration. (Right-click menu).
- Automatic Reconfigure – like the above feature, this will cause the driver to modify device and point properties if the configuration of an existing device has changed. (And new points will be created if they do not exist).
- Server Port, User Name, Password – these fields are a valid Geo SCADA user credentials which will execute the configuration actions. When not configured, the driver will be unable to create or modify devices.
- Create Devices or Instances in Group – Enter a group name. e.g. "SE Logger". New loggers will be created within this group.
- Device Template – Enter a template name. This template should contain a Logger object and zero or more points, plus any displays, logic, trends or other objects as required. When a new logger is configured by the driver, this template will be instanced. (Leave blank for now. More about this later).

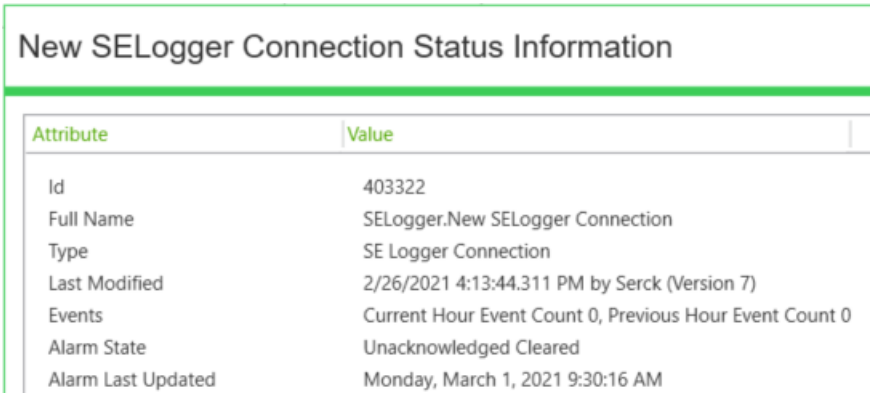
When a channel is configured correctly and is connected to the web service, the broker state will be Healthy.

3.4 Connection to the API

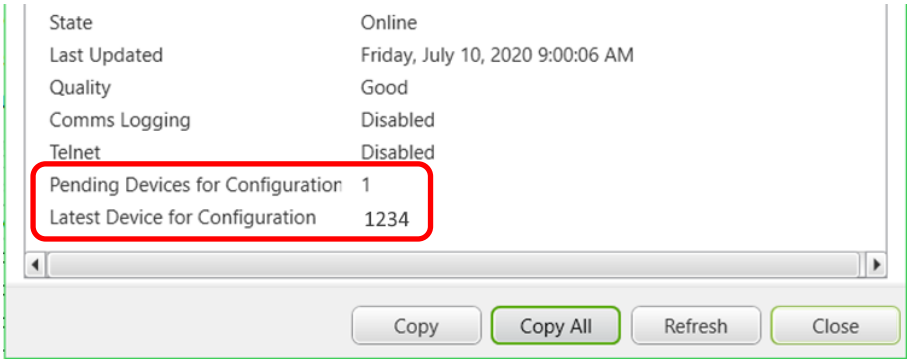
On connecting the driver will attempt to list all loggers and points (streams) on the web service. What happens next will depend on how the 'Automatic Configure' checkbox is set on your connection object.

If this field is **unchecked**: 

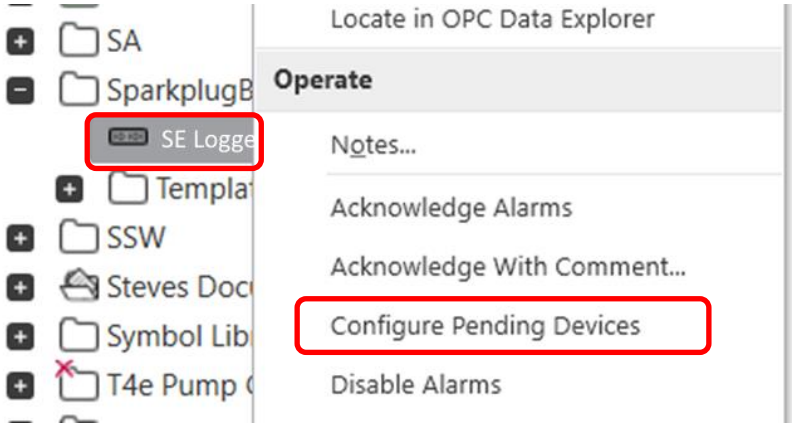
Nothing will appear to happen, but the object will cache what it has received. You can see this in the Status panel:



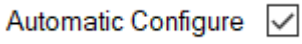
~~~~~~



The logger device is identified by its Device Id. You can now configure the logger by selecting the context menu item 'Configure Pending Devices':



Alternatively, if this field of the Broker is **checked**:

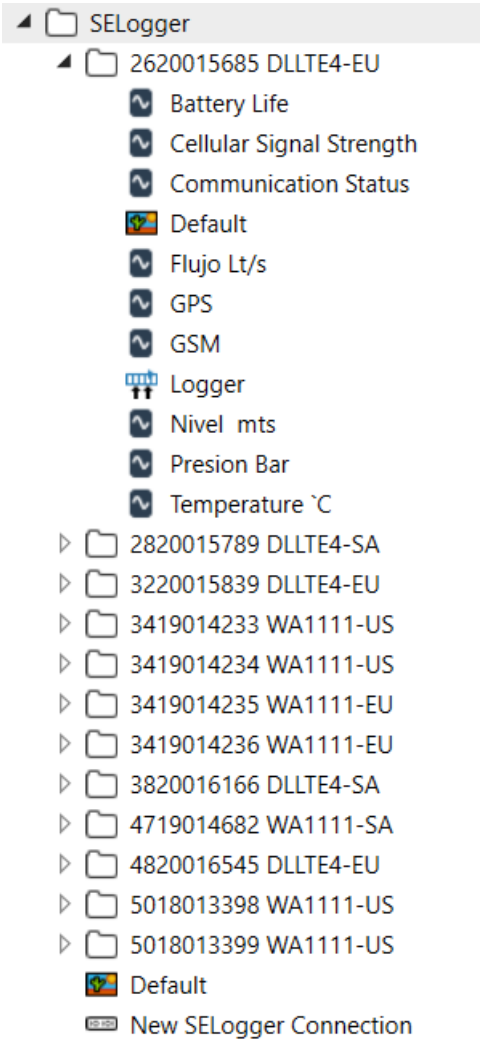


Then the configuration will be fully automatic, and all of the available the logger devices and points will be created with no user intervention.

The 'Configure Pending Devices' method should request all 'queued' configurations be action and then remove each of them from that queue. The code could be extended to allow individual node/device to be discovered and configured.
~~~~~~

3.5 Created Configuration

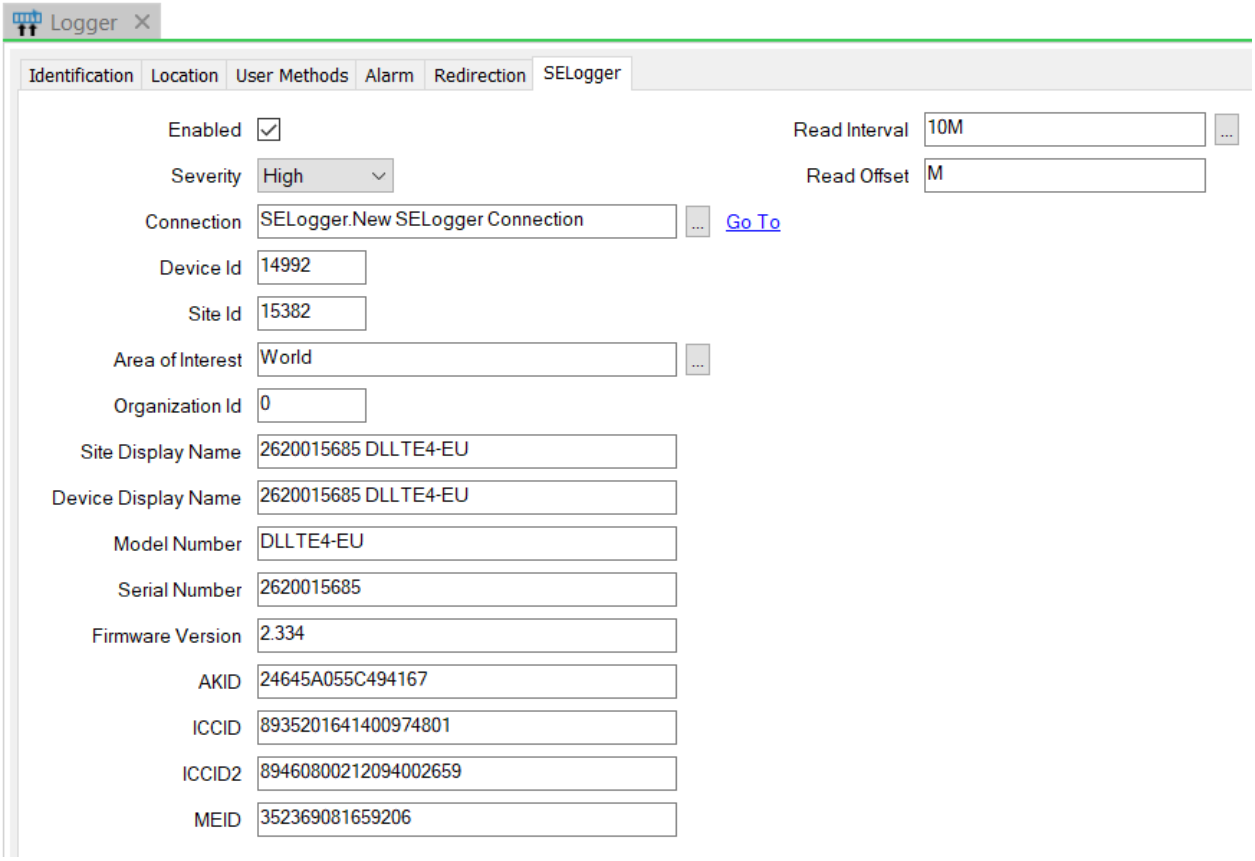
The driver will configure all database objects needed and set various properties as required for them to be valid and useful.



A group has been created for each logger. Every item has an appropriate name taken from the existing configuration and has database fields set up correctly.

3.5.1 SE Logger

The logger is configured as follows:



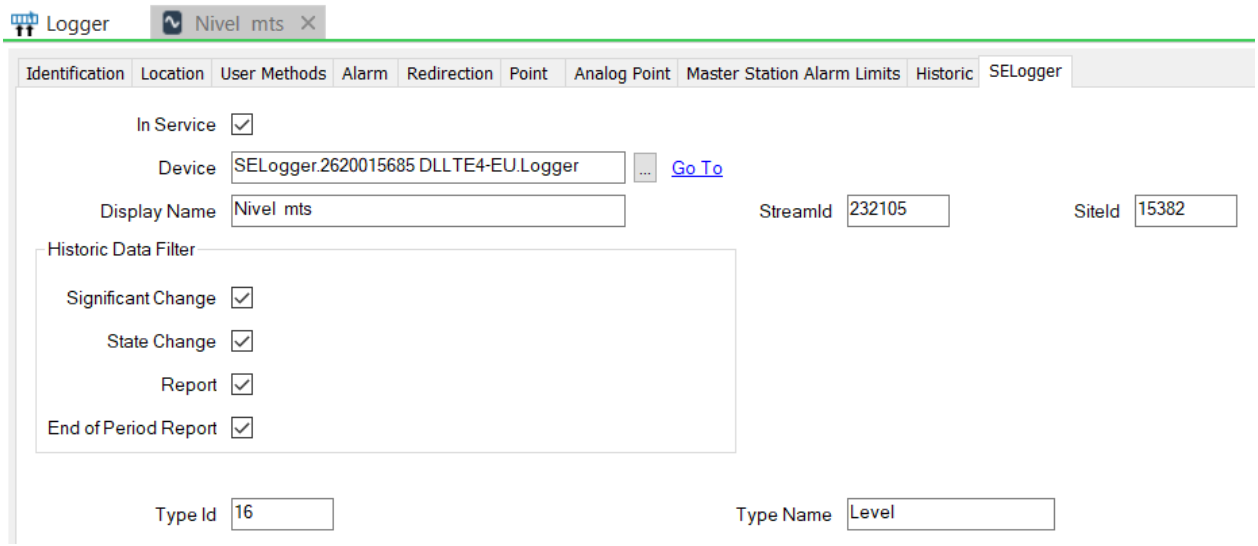
There is a link to the connection object. The specific fields here are:

- The read interval and offset are used to poll for new data from the logger.
- Device Id – the unique reference number used by the API to identify this logger. If the Geo SCADA logger name is changed, the driver will continue to read from the same logger using this Id.
- Site Id – unique site reference number used by the API to identify this loggers data streams.
- Site/Device display name, and all following fields. These fields are properties read and set from the API by the automatic configuration feature.

3.5.2 Points

Finally, points are created to match the streams in the list for this site. These are placed in a sub-group/folder path appropriate to their name, so for example if a metric is named "pump1.speed" then the point is named "speed" within a subfolder named "pump1".

Metrics have a Stream Id which is populated with a number. This field is used to identify the point, hence the other Name, FullName and Display Name properties are not used to retrieve data.



The driver automatically enables historic data storage and filtering so that all data is stored in the Geo SCADA historian. These configuration properties are set when new configuration is received, so if you do not wish for the driver to do this you could use templates and property overrides to control it.

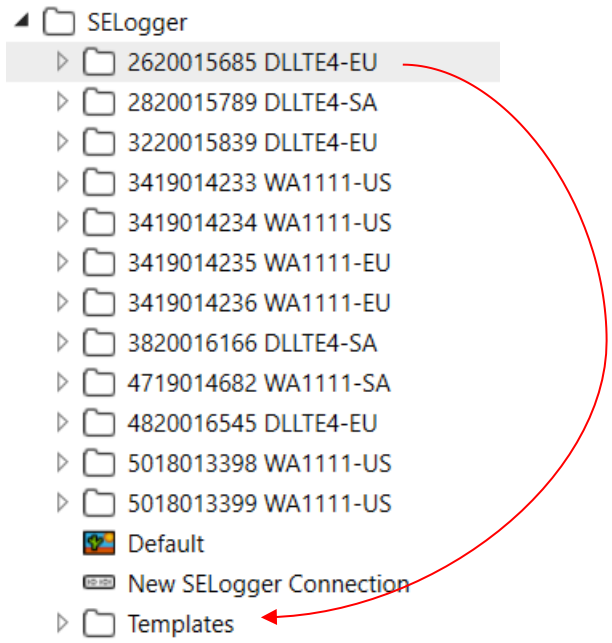
3.6 Receiving New Configuration

Points which are no longer within the API messages are not deleted by the driver and therefore will be retained until a user wishes them to be removed. It could be a useful enhancement for the configuration process to find and mark such points with a data or configuration field to identify them as unused.

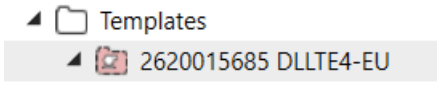
3.7 Making Templates of the Configuration

As well as automating the configuration process, this driver integrates with the Geo SCADA templates and instances feature. A good way to understand the features is to start from the configuration created above and then use that to define templates.

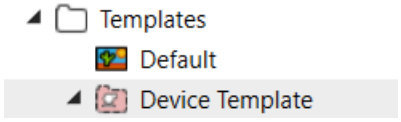
First, drag the configuration which the driver had made for a representative device into the Templates group you created:



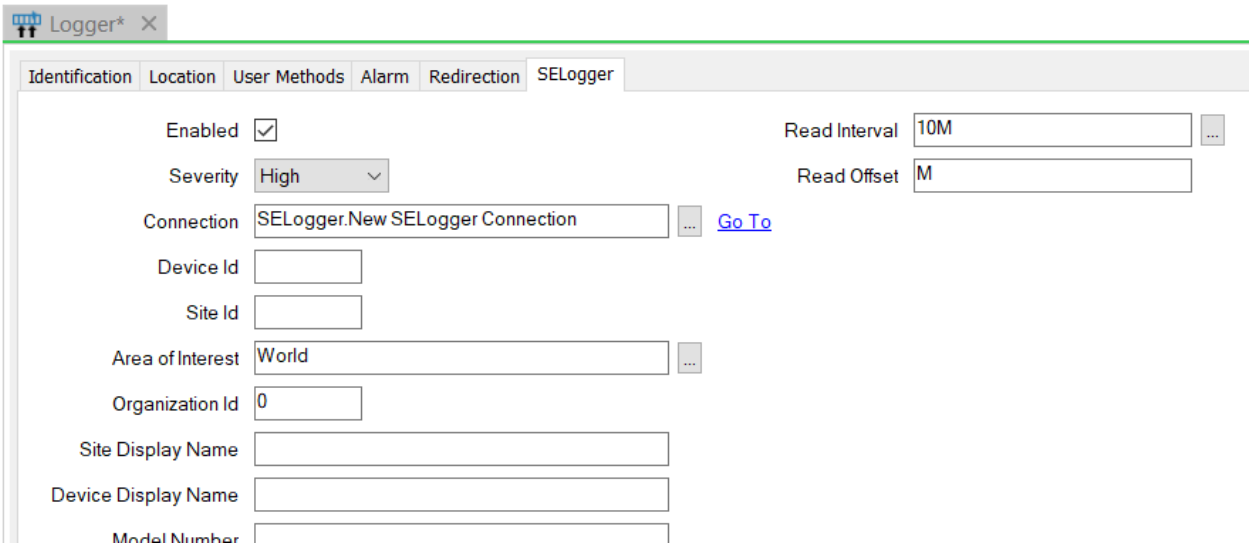
Then convert the device into a template with the 'Convert' to Template context menu.



Then rename to a more suitable device.

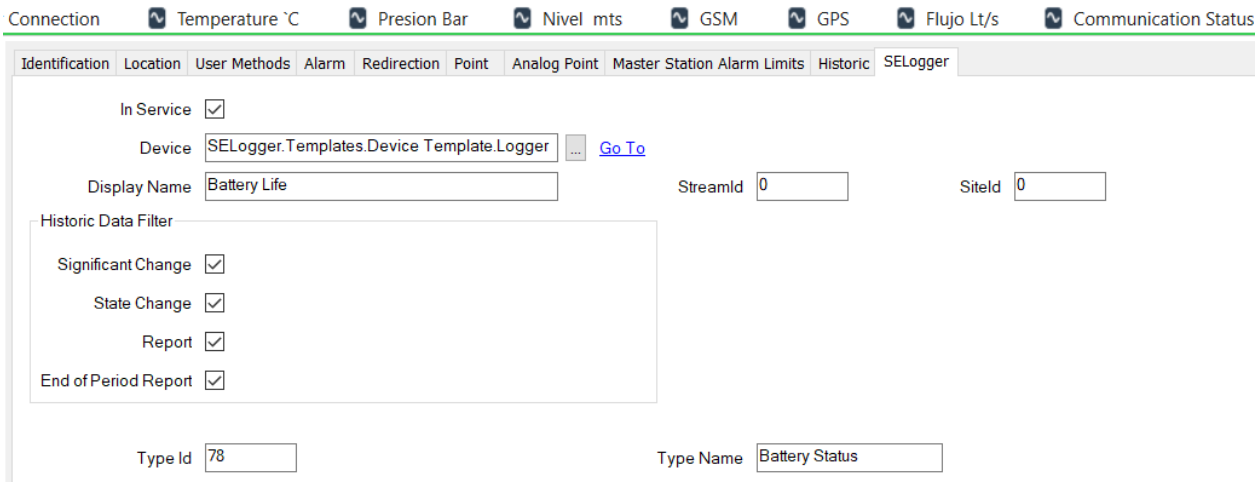


And remove the device, site, and other identification fields from the template's logger object:

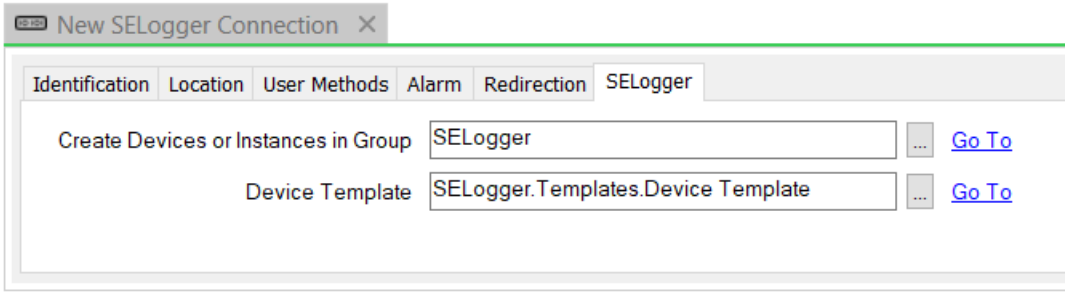


You can modify this template as needed, including the addition of mimics, trends, Logic and point properties such as ranges, limits and alarm settings.

At this point you might want to tidy the point configurations by removing the StreamId and SiteId values, because the points will be identified during the configuration process by their Display Name properties.



The logger connection object can now be configured to refer to the template. Please configure as in this screenshot:

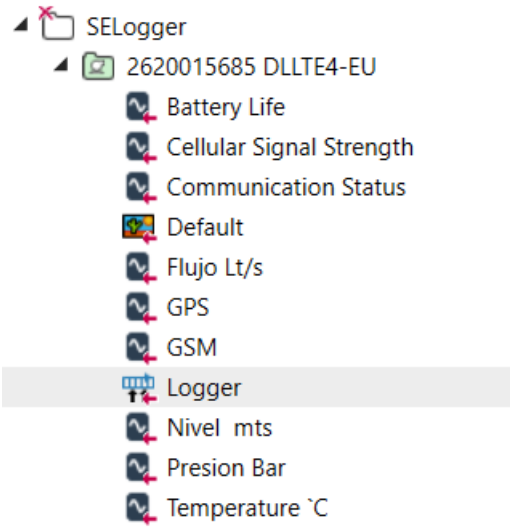


3.8 Automated Configuration of Instances

Now, go to your logger connection object and use the context menu to disable and re-enable it. This will force all configuration data to be re-read.

If you are not using automatic configuration then use the pick action to cause configuration.

Now the driver has created configuration based on the templates, and the instances are therefore controlled by the configuration in each template. Your mimics within the templates will be used automatically for new automatic configuration.



3.9 How to Diagnose any Problems

This driver logs information to the server and driver log files, similar to other drivers. Please see the product help on Logging, and the DDK driver guide's relevant sections on logging and debugging. Also see the Enhanced Events field description in the Connection section above.