Features of IEC 61850 Source Code Libraries

What is IEC 61850?

IEC 61850 is an international standard originally designed for the integration of electric utility substation devices. It is also being extended for use in wind power generation and for the management of Distributed Energy Resources (DER). It includes several Ethernet-based communications protocols, along with standardized naming and object modeling. It also includes an XML-based Substation Configuration Language (SCL), which was developed to allow for the exchange of configuration data between tools. SCL is used to design, document and exchange both device level and substation level configurations. It is a much more comprehensive approach to integration standards than previous efforts in substation integration. IEC 61850 uses advanced communications techniques to address data management and simplify integration of applications.

What are Triangle MicroWorks, Inc. Software Libraries?

Triangle MicroWorks' Software Libraries provide a cost-effective means of supporting industry-standard protocols in your device. Incorporating our royalty-free Software Libraries in your products will shorten development time, freeing internal resources to work on company proprietary aspects of your products.

Several library configurations are available:

- The basic **IEC 61850 Server Source Code Library** meets the needs of most device-level implementations.
- The **GOOSE Publisher/Subscriber** (Generic Object Oriented Substation Event) optional component implements an additional IEC 61850 protocol used for protection functions and other applications requiring high speed multicast peer-to-peer communications capabilities.
- The **Client** optional component can be used in a variety of embedded and non-embedded environments to provide very efficient high-speed client services. Many device level implementations use GOOSE services for peer-to-peer communications, and therefore do not require the Client optional component.
- **IEC 61850-7-410 (Hydro)**, **-7-420 (DER)** and **IEC 61400-25 (Wind)** Object Models are supported.
- Support for **IEC 61850-9-2 (Sampled Values)** is optionally available.
- The **IEC 61850 Server Front End Toolkit** provides an implementation of the Source Code Library on a separate Linux coprocessor board and obtains data through existing DNP3, Modbus or IEC 60870-5 communication ports. It requires no programming and uses an IED Capability Description (ICD) file to define the object model and mapping information. For more information, see product literature on *IEC 61850 Server Front End Toolkit Design Details for Implementation*.

Features Common to All of Our IEC 61850 ANSI-Standard C Source Code Libraries

- Written in ANSI-Standard C Source Code.
- Designed to be processor and operating system independent, using any ANSI-Standard C compiler.
- Simple configuration for big-endian or little-endian byte order.
- Can be used with or without a Real Time Operating System (RTOS).
- Highly efficient code and advanced protocol parsing techniques designed to work well in embedded and workstation environments.
- Typical product integration times depend highly on the database model complexity.

— Continued on back of this page —

IEC 61850 Server Source Code Library Features

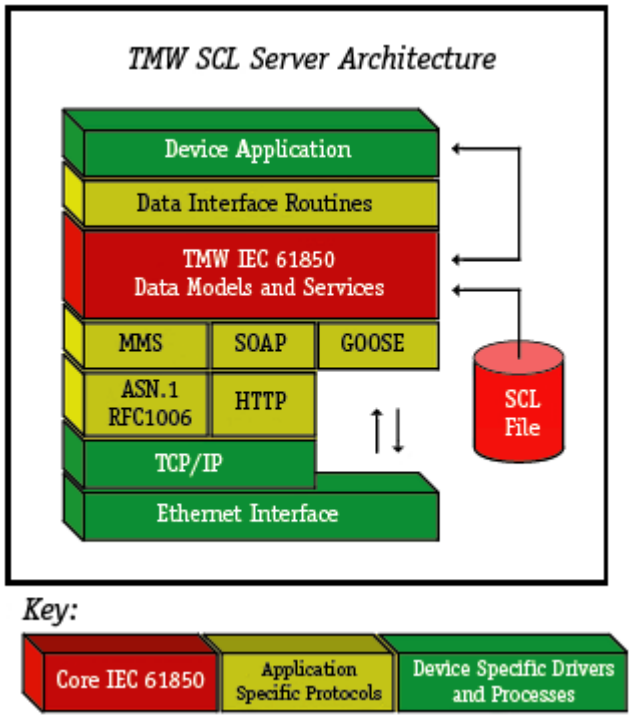
- Support for total runtime configuration via local SCL file or API.
- Additional lower level interfaces available for specialized device integration requirements.
- Advanced data modeling techniques simplify generation of conformant servers.
- Implements IEC 61850 services within the library – not in user code.
- Integrates with application at object level – not transaction level.
- Efficient memory management; malloc() and free() only used at system startup.
- Hardware requirements: TCP/IP port, ½ Megabyte RAM, ½ Megabyte ROM. If your device does not meet these requirements, see the *IEC 61850 Server Front End Toolkit* datasheet for alternative solutions.

IEC 61850 Client Option Features

- Application interface designed for ease of use, even in embedded environments.
- Supports all services required for IEC 61850.
- Highly efficient operation.

GOOSE Option Features

- Application interface designed to free application developers from protocol details.
- Integrated with data model features, including control blocks.
- Includes GSSE (Generic Substation Status Event) support.
- Implements IEC 61850 GOOSE protocol very efficiently.



This figure shows the major components of an IEC 61850 server implementation. The **Device Application** (process logic) is largely independent of the transaction sequences, etc. – integration is done mostly through the data interface routines. There is, however, an interface capability that allows the application code to get access to service and indications when needed.

The **Data Interface Routines** allow the 61850 Data Object Models to be interfaced with the application process data of the device. The IEC **61850 Data Models and Services** are implemented directly within the library, and generally do not require custom code.

The **MMS (Manufacturing Message Specification, or ISO 9506)** stack implements the actual IEC 61850 transactions over TCP/IP connections. The **SOAP (Simple Object Access Protocol)** package implements IEC 61400-25 typically used in windmill applications. The optional **GOOSE** package implements a simple but reliable multicast data exchange protocol directly over the Ethernet.

The architecture assumes that **TCP/IP** and **Ethernet** interfaces are available on the target platform.

2840 PLAZA PLACE, SUITE 205 • RALEIGH, NC 27612 • PH: 919.870.5101
FAX: 919.870.6692 • WEB SITE: www.TriangleMicroWorks.com

IEC 61850 Server Source Code Libraries

Design Objective:

Our primary design objective is to provide our customers with an ANSI Standard C Source Code Library with a **T**arget **A**pplication (**TA**) Interface that can be implemented in less than three man weeks. To accomplish this, our design divides the interface into "entry-points" from TA-to-Source Code Library, and "calls" back into the TA from the Source Code Library.

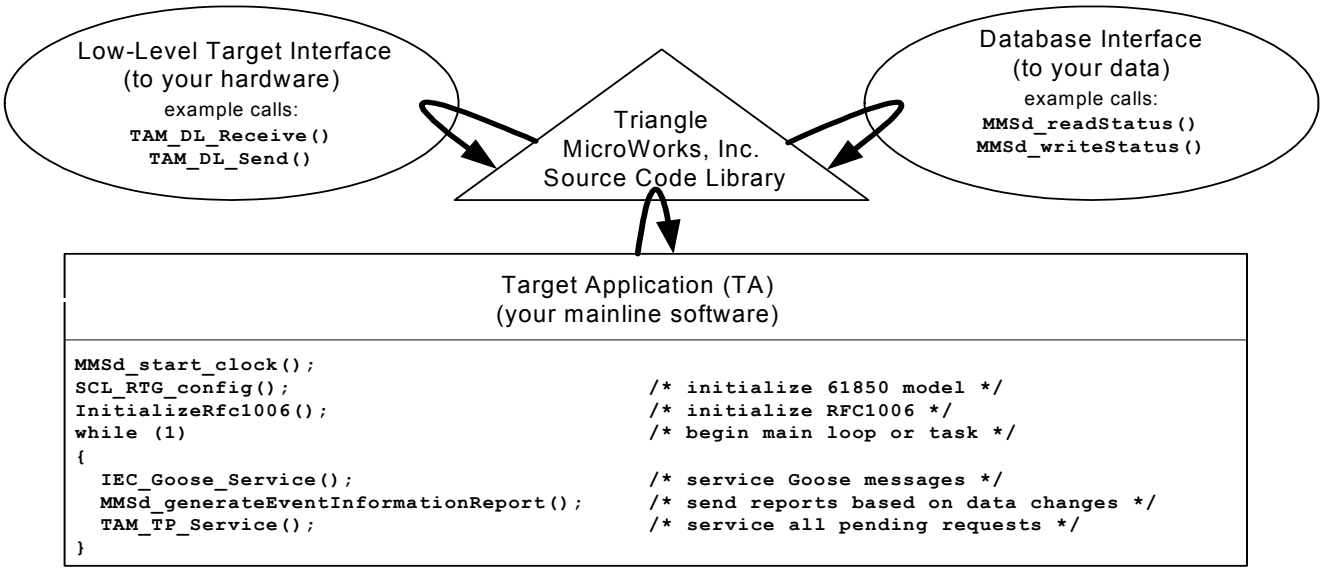
The "Triangle" Approach to Interfacing between the Source Code Library and TA:

The interface between the Source Code Library and TA can be viewed as a three-sided, or "triangle" interface. Two sides represent calls back into the TA from the Source Code Library Interface. Each of these sides organized into individual, well-documented modules or header files. These files are the only recommended customer-editable (or platform-specific) files. All other files are protocol-specific and should not need to be modified by the customer.

The three sides of the interface are shown in the diagram below:

1) TA-to-Source Code Library Entry Points: The entry points include a few Source Code Library initialization functions and process functions for each protocol supported (MMS, GOOSE, GSE, etc.); the process functions can be called regularly as part of a Target Application main loop, or as event-driven tasks in a real-time operating system environment.
2) Low-Level Target Interface: Provides access to hardware components such as TCP/IP, Ethernet adapters, and timers.
3) Data Interface: Provides a simple callback mechanism for associating TA data with IEC 61850 modeling data.

Example flow diagram for an installation of the IEC 61850 Server Source Code Library



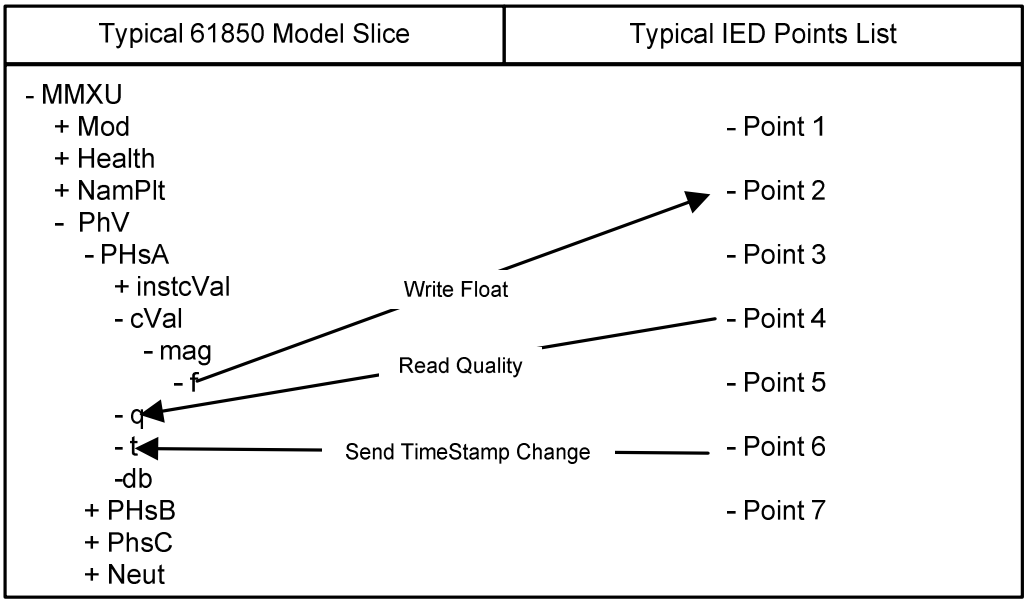
— Continued on back of this page —

Typical Installation Sequence for IEC 61850 Server Source Code Libraries

We strongly recommend that before you develop any communications protocol, whether or not you use our Libraries, you should create a **Protocol Implementation Conformance Statement (PICS)** that conforms to standard "device profile" templates contained in the IEC 61850 documents. The PICS describes which optional features of IEC 61850 are implemented, and specifies details such as timer accuracy and various ranges supported by the device. Our Source Code Libraries come with PICS templates, with much of the information already filled in. The IEC 61850 standard also requires each vendor to supply an IED Capability Description (ICD), written in an XML-based language called the Substation Configuration Language (SCL). The ICD file specifies all of the capabilities of the server implementation, including detailed descriptions of all object models. This file is used by IEC 61850 clients to configure databases and engineering tools. Server implementations based on our Source Code Libraries can also use the ICD file for configuration of their object models directly.

After completing the PICS and ICD files, a typical sequence to install one of our Source Code Libraries is:

1) Edit low-level target interface files to interface TCP/IP and Ethernet adaptors (Ethernet adaptors only required for Goose protocol support), add access to a free-running millisecond timer/counter, and to configure byte-order and floating point formats.
2) Add TA-to-Source Code Library entry point functions in Target Application for initialization.
3) Conduct initial testing, which verifies operation of the Source Code Library on the target hardware. The initial testing can be done using one of the sample applications delivered with the Source Code Library.
4) Attach Source Code Library-to-TA calls to your data by adding custom callbacks for read, write, and change notification. Change notification can be event or poll based.



5) Make final adjustments of the configuration interface to ensure that all user settable configuration parameters are mapped to Source Code Library configuration structures.
6) Conduct final testing.

Testing and Diagnostics

For testing Source Code Library installations, we provide the following tools:

- *SCL Forge* with the ability to create and edit ICD files is included.
- Optional 61850 Test Suite with *Anvil* and *Hammer* is available to facilitate testing and provide reference implementation.
- A small HTTP server is included that will display XML/HTML pages for the IEC 61850 models on the device via a web browser.

IEC 61850 Client Source Code Libraries

Design Objective:

Our primary design objective is to provide our customers with an ANSI Standard C **Source Code Library** with a **T**arget **A**pplication (**TA**) Interface that can be implemented in less than three man weeks. To accomplish this, our design divides the interface into "entry-points" from TA-to-Source Code Library, and "calls" back into the TA from the Source Code Library.

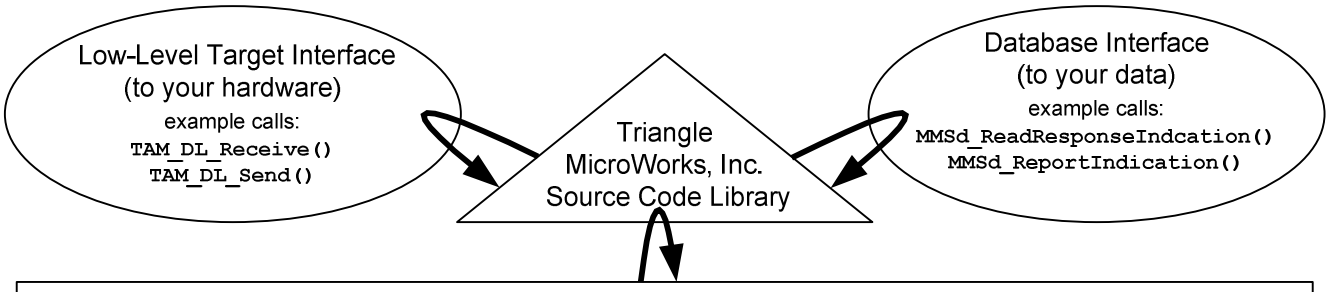
The "Triangle" Approach to Interfacing between the Source Code Library and TA:

The interface between the Source Code Library and TA can be viewed as a three-sided, or "triangle" interface. Two sides represent calls back into the TA from the Source Code Library Interface. Each of these sides organized into individual, well-documented modules or header files. These files are the only recommended customer-editable (or platform-specific) files. All other files are protocol-specific and should not need to be modified by the customer.

The three sides of the interface are shown in the diagram below:

1) TA-to-Source Code Library Entry Points: The entry points include a few Source Code Library initialization functions and process functions for each protocol supported (MMS, GOOSE, GSE, etc.); the process functions can be called regularly as part of a Target Application main loop, or as event-driven tasks in a real-time operating system environment. Routines corresponding to the MMS services used in IEC 61850 are provided, such as Initiate, Read, Write, Conclude, as well as Goose Subscribe.
2) Low-Level Target Interface: Provides access to hardware components such as TCP/IP, Ethernet adapters, and timers.
3) Database Interface: Callback functions are invoked when responses to service requests are received. The callback functions use simple utility routines to retrieve data from the responses.

Example flow diagram for an installation of the IEC 61850 Client Source Code Library



Target Application (TA)
(your mainline software)

```
MMSd_start_clock();
IEC_Goose_Initialize();                    /* initialize Goose system */
TAM_TP_InitializeContext();                /* initialize transport layer (RFC1006) */
MMSd_ClientInitialize();                   /* initialize client system */
MMSd_InitializeRemote();                   /* prepare connection (per server) */
MMSd_InitializeRequest();                  /* connect to server (per server) */
while (1)                                  /* begin main loop or task */
{
  MMSd_openReadVariableRequest();          /* perform polling, for example */
  MMSd_addReadVariableRequest();
  MMSd_closeReadVariableRequest();
  IED_Goose_Service                        /* service Goose messages */
  TAM_TP_Service();                        /* service all pending requests */
}
```

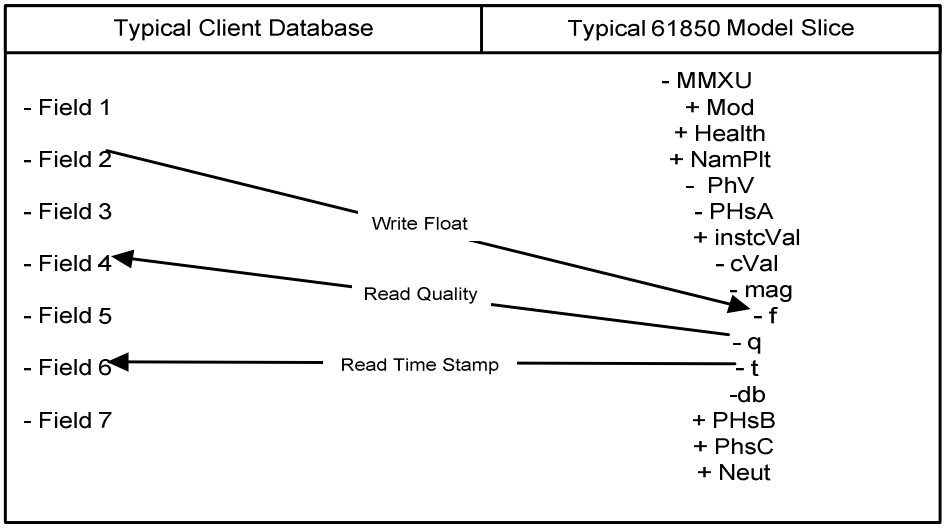
— Continued on back of this page —

Typical Installation Sequence for IEC 61850 Client Source Code Libraries

We strongly recommend that before you develop any communications protocol, whether or not you use our Libraries, you should create a **Protocol Implementation Conformance Statement (PICS)** that conforms to standard "device profile" templates contained in the IEC 61850 documents. The PICS describes which optional features of IEC 61850 are implemented, and specifies details such as timer accuracy and various ranges supported by the device. Our Source Code Libraries come with PICS templates, with much of the information already filled in.

After completing the PICS, a typical sequence to install one of our Source Code Libraries is:

1) Edit low-level target interface files to interface TCP/IP and Ethernet adapters (Ethernet adaptors only required for Goose protocol support), add access to a free-running millisecond timer/counter, and to configure byte-order & floating point formats.
2) Add TA-to-Source Code Library entry point functions in Target Application for initialization.
3) Conduct initial testing, using a simple IEC 61850 server provided with Source Code Library which verifies operation of the Source Code Library on the target hardware. The initial test consists of establishing connections and reading values from the sample server executable, which is provided for operation on Windows or Linux platforms.
4) Add TA-to-Source Code Library entry point functions in Target Application for service requests (for example, read dataset or configure control blocks for Goose, Logs, or Reports). The strategy for using the services depends on the TA's use of IEC 61850.
5) Attach Source Code Library-to-TA calls to your database design implementing the service callback functions to store received data.



6) Make final adjustments of the configuration interface to ensure that all user settable configuration parameters are mapped to Source Code Library configuration structures.
7) Conduct final testing.

Testing and Diagnostics

For testing Source Code Library installations, we provide the following tools:

- 61850 Test Suite, consisting of:
  - SCL Forge: provides the ability to create and edit ICD files (license included with library purchase)
  - Hammer: Optional test client to test servers and provide reference implementation
  - Anvil: Optional test server to validate clients and provide reference implementation.
- Sample TA source code for a client application (Windows or Linux) which will connect a server, perform some basic data access and exercise the IEC 61850 reporting mechanism. This example is suitable for use as a starting point for client implementation or as a test bed for rapid prototyping portions of the entire system.

IEC 61850 Server Front End Toolkit

Design Objective:

Our primary design objective for this product is to provide a rapidly deployed solution for implementing IEC 61850 in a resource-limited device with a low overall cost of development. This is done by providing a Target Application for our IEC 61850 Server Source Code Library (TA – see product literature *Design Details for Implementation*) that runs on a separate front end processor and communicates with your device using Modbus, DNP3, or IEC 60870-5 (Master side). The default implementation requires no modification to the source code. Points received by the Master side are linked to the IEC 61850 Object Model in an IED Capability Description (ICD) file used to initialize the IEC 61850 Server Front End Toolkit.

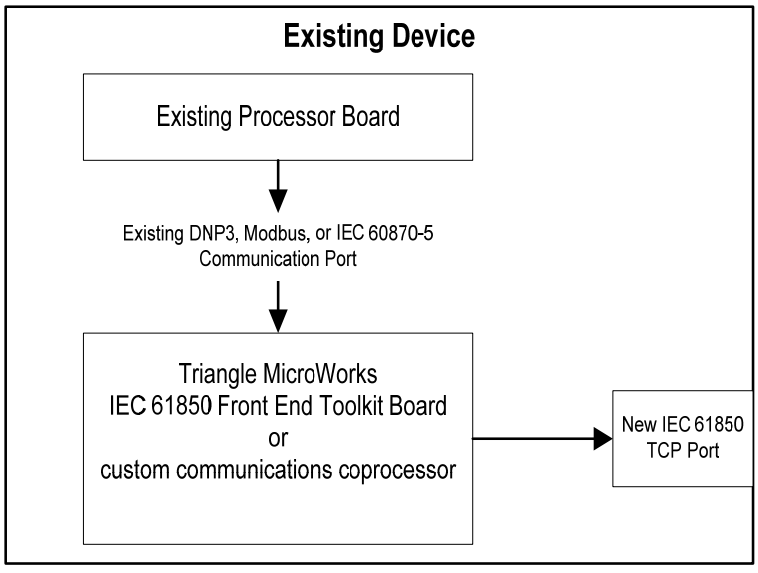
An ICD file is one of the IEC 61850 Substation Configuration Language (SCL) file formats based on XML that are used to exchange configuration data between different tools. The ICD file includes a standardized mechanism for mapping Basic Data Attributes in the object model to alternate protocols. The Server Front End TA makes use of these standardized features to support IEC 61850 access to small serial-based devices with little or no additional programming. Any IEC 61850 model can be supported by simply describing it in an ICD file, assigning the mapping information to the model elements, and installing the ICD file on the Front End device for use in boot-time configuration.

This product is also an excellent solution for target devices that lack the necessary resources to support IEC 61850 on existing hardware (TCP/IP port, ½ Megabyte RAM, ½ Megabyte ROM, spare processing power). Adding a separate communications coprocessor with TCP/IP support solves this problem without expensive redesign of the main processor board. IEC 61850 GOOSE communication is supported, but response time will be limited by the speed of the existing DNP3, Modbus, or IEC 60870-5 Communications port.

Typical Installation Sequence for IEC 61850 Server Front End Toolkit:

1. Customize the provided *Protocol Implementation Conformance Statement* (PICS) for the target device.
2. Create an ICD file representing the logical nodes supported by the target device using the provided ICD file editor or a 3rd party application.
3. Add Modbus/DNP3/IEC 60870-5 Point Numbers to ICD file along with formulas for any data translations required to match IEC 61850 data types.
4. Install Triangle MicroWorks Front End Toolkit Board in target device and load ICD file or download Linux executable and ICD file to custom hardware. For custom hardware not using Linux see product literature on *IEC 61850 Source Code Library Design Details for Implementation* for information on customizing the toolkit.
5. Conduct final testing.

Example diagram for an installation of the IEC 61850 Server Front End Toolkit



2840 PLAZA PLACE, SUITE 205 • RALEIGH, NC 27612 • PH: 919.870.5101
FAX: 919.870.6692 • WEB SITE: www.TriangleMicroWorks.com