

Introduction

What is SILECS?

Silecs is a framework for Ethernet communication among an heterogeneous set of controllers by providing to the user a configuration tool, and communication library with hardware independent API.

Silecs configuration tool provides users the possibility to generate PLC and different controllers code (PXI platform & uController) using a graphical user interface. It is designed to allow users to easily set-up the communication with a controller without having to physically write any source code, thus saving time in the long term.

The configuration tool is based around an XML Editor that allow users to design the SILECS 'design' and 'deploy' documents.

The design document describes the sets of data that the user wants to exchange with the controller in a completely hardware independent manner. The deploy document is responsible of binding instances of different designs on a controller. This document therefore introduces the dependency with the hardware.

SILECS configuration work flow

The following workflow defines a sequence of step to guide the user in setting up communication within his controller and his client application (Fesa class, C++ program, Python script, ...).

Some steps may be skipped but the configuration of a new system may follow the following steps:

1. DESIGN

- a. Define one or more **design documents**. These will define the structure of the exchanged data;
- b. Define a **deploy document**. This will define the hardware on which the class(es) will be deployed together with the controller memory mapping strategy and the communication parameters.
- c. **Generation of controller resources**. This will generate the controller sources and the client configuration documents. This last one will be automatically delivered on the operational database (TN network)
- d. **Generate FESA class** (optional). Starting from a SILECS Design document the user can generate a default FESA3 class design and a c++ wrapper to the SILECS communication library customized for user design.

2. CONFIGURATION

The user will have to manually **upload** the generated resources (PLC data-blocks, Variables & Symbols definition, ...) on his controller. Each controller requires different techniques/tools in order to load sources within the controller configuration memory and therefore this operation cannot be automatized.

3. VALIDATION

Test the controller via the **diagnostic tool** (optional). Silecs provides a powerful diagnostic tool that allows the user to communicate with their controller only using the resources delivered on the operational database without writing any code. This tool is dedicated for the experts and shall never be used for operation.

4. INTEGRATION

The user creates a new FESA class (C++ client or other) using the generated design as template and integrates the generated c++ code. Compile and start the class which can communicate with the controller via the SILECS communication library.

