1. Getting started

Requirements

In order to be able to run SILECS Diagnostic Tool all the following requirements must be fulfilled:

- The user shall be running the application from a Linux machine in the technical network or a trusted machine.
- The user must have a valid user login to the CERN middleware.
- As previously said SILECS Diagnostic Tool provides access only to hardware which has been configured by the SILECS framework. This framework count on the fact that the user is tightly following this 4 steps workflow:

    1. Design definition: to define the structure of the exchanged data
    2. Deploy definition: to define the hardware on which we will deploy the classes, to define the PLC memory mapping and the communication parameters.
    3. Client resources generation: to generate the controller sources and the client configuration documents.
    4. Controllers upload: manual upload over controller's symbols and code.

- SILECS Diagnostic tool is compiled for compiled for a 64 bit Linux SCL6 machine.

Starting the tool

If you are running a 64bit SLC6 Linux machine the tool can be started directly on your machine whereas if you are running an SLC6 32 bit or an SLC5 the tool will be launched automatically using an ssh tunnel toward a 64bit SLC6 Linux machine.

Application is however already installed in any of the distribution in order to be able to run with the command:

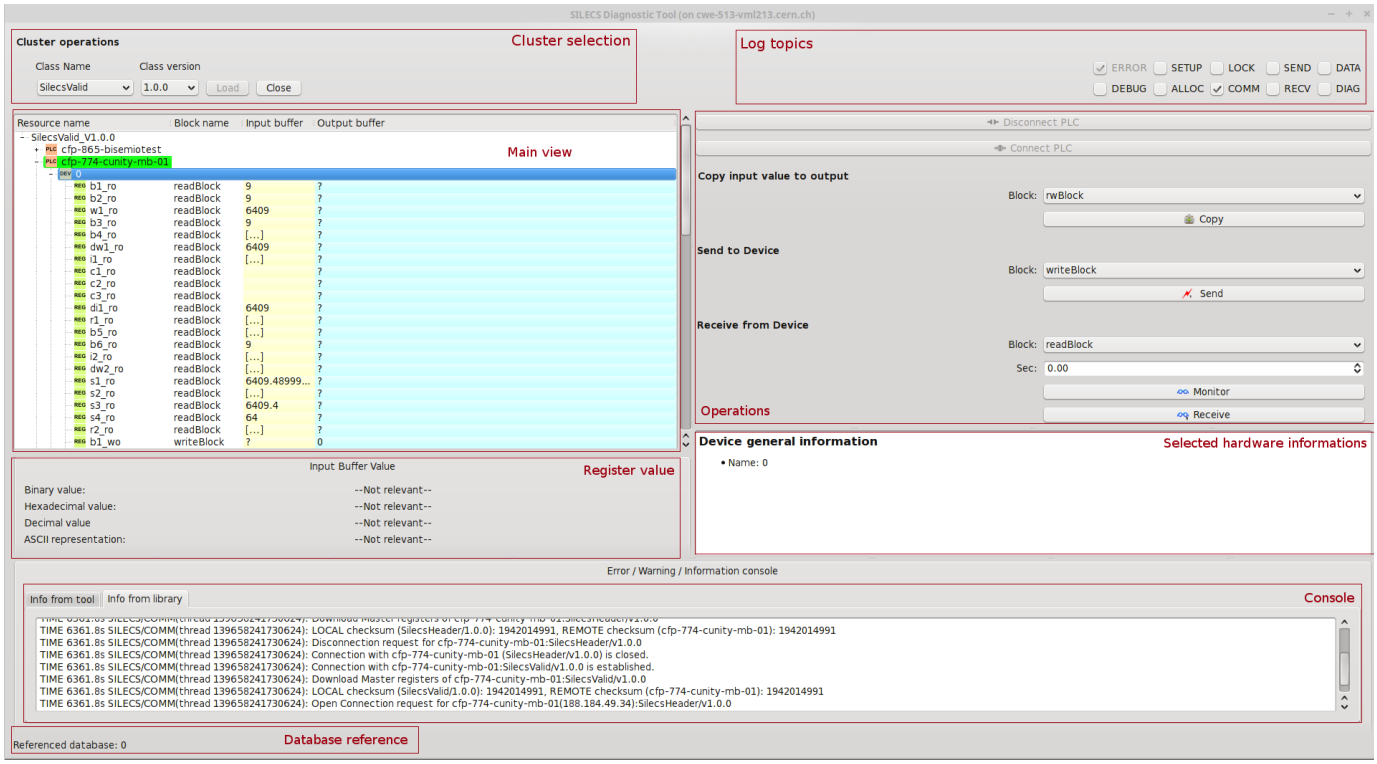
- **silecs -d**

The tool cannot be launched under Windows!

- You can also start the diagnostic tool directly from the JAVA configuration tool menu: **Silecs –> Diagnostic tool**

First view

As soon as the tool starts up, a log-in window will request the user for their CERN credential. If log-in procedure will succeed a windows similar to the one in the following figure shall be displayed.



The main window can be subdivided in 6 principal sectors:

1. **Cluster Section:** Responsible of displaying to the user the list of Cluster which they are allowed to open. The content of this sector strongly depend on the connected user. In order to be able to connect to a cluster the user needs to be defined as owner, editor or user in the class document.
   N.B: Being able to open one cluster does not imply being able to access all the PLCs which deploy the class. This is due to the fact that the tool constraints the access only to the PLCs whom the user is owner, or editor of.
2. **Main view**: this sector is the key sector of the whole application. Here the complete control architecture is represented in a tree structure. As in the SILECS communication library, we can consider that one design can be implemented in more than one PLC. A group of PLCs which map the same class shall be called “cluster”. For each PLC within the cluster a set of device is attached. Each device contains a list of register which are grouped into blocks.
3. **Log topics:** this sector allows to select which log topics the user what to be displayed within the "info from library tab" in the console section.
4. **Operations:** This sector is responsible if the whole operations available over the hardware. It gives indeed the possibility of connecting/disconnecting to/from PLCs and to send/receive/monitor blocks from/to the whole cluster, a single PLC or device.
5. **Selected hardware information**: This sector provides information about the selected resource within the centre main tree.
6. **Console:** This sector is responsible of informing the user with run time information and errors.
7. **Database reference:** Many different version of SILECS may be installed at the same time. The label is mostly for information for administrator users and does not affect standard users in any way.