ActionNET: Scripts Languages (Program Sequencing)

The ActionNET user can write computer programs (scripts) while creating a supervisory project. These programs will be embedded in the system, making possible to particularize or create behaviors and actions on objects (tags, screens, reports) that are not natively present in SCADA.

The supported programs are C# and VB.NET and differently of others SCADAs, the Scripts are compiled, generating machine code, what make them 20-100 times faster than interpreted scripts. More than this, they are Microsoft´s standard languages and not a proprietary Script from SCADA Company. They support dotNET debugger tools and are compiled before run, identifying syntax and semantic errors and generating objects uses count, cross reference tables of all objects used as well as objects declared but not used.

The Scripts can be created as tasks, classes, expressions, screens' code behind and inside symbols draws.

The Scripts are .NET natives and support interllicense.

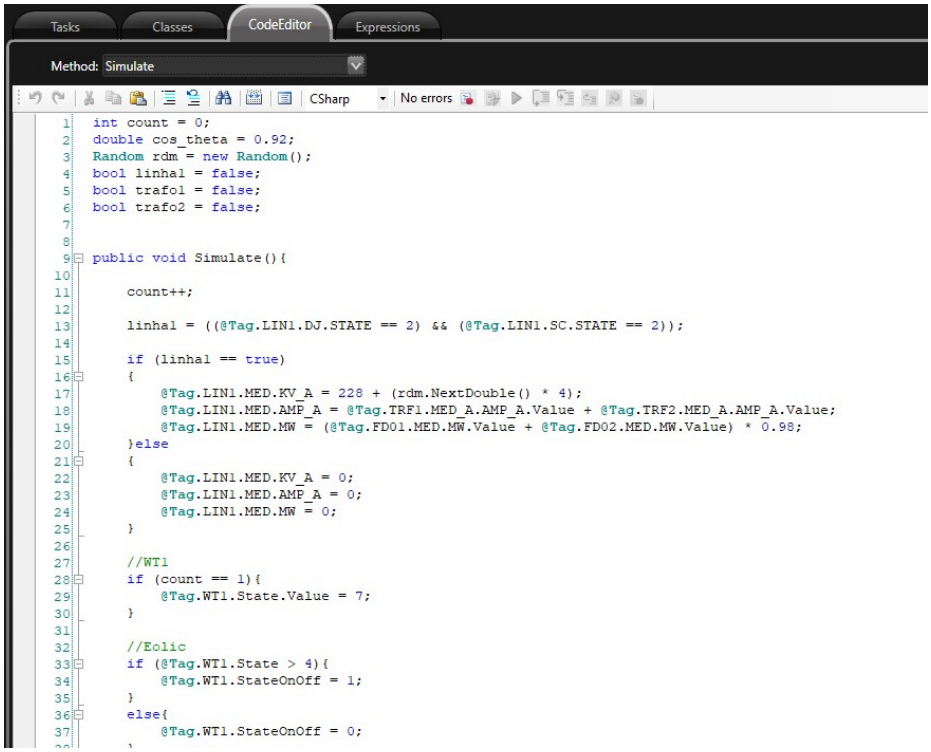


Figure 1 – CSharp Script Example

ActionNET: Disturbance Capture

A SCADA normally does not function to read disturbance records (DR) because the reading frequency is well above the reading supported by SCADA applications. On the other hand, relays using the IEC61850 protocol can automatically capture these files through protocol´s directives, once the protocol supports the file transfer function. In this way, through the IEC61850 it is possible to read these files and record in a disturbance file area, independent of the relay´s manufacturer.

The reading may be programmed to periodically look for disturbance files or spontaneously, when a relay´s "report" is programmed to send a message whenever a disturbance happened. Through scripts, the disturbance files will be recorded in a certain directory.

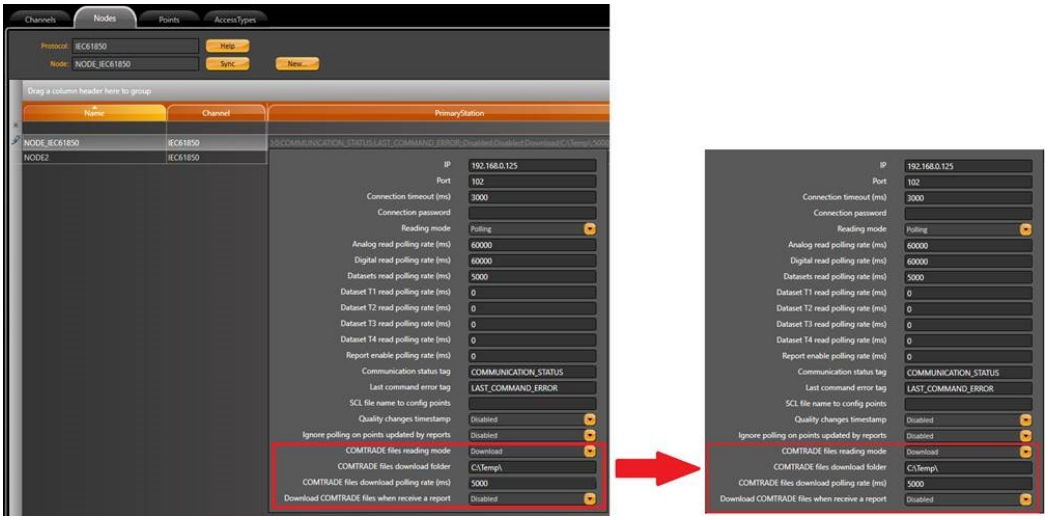


Figure 1 – 61850 attributes to read disturbance files

ActionNET – Load Monitoring

In order to run a load-monitoring module, initially the historical demand of the system must be registered, every few minutes. To do this, during an initial period, the load of all the transformers of the system and its sum, corresponding to the load of the system, will be written to the defined frequency. In this record will also be identified the day of the week, the time, whether or not holiday, in the case of working days, and whether or not daylight saving time is in effect.

In these records will also be added:

- Average time zone temperature;
- Hours of major events, that occurred on the day (if occurred one), such as world cup matches, superball match, etc.

With this data, the existing load is monitored, based on the load passed on similar days. The module will have a routine that treats several records generating a typical day´s curve for that period.

In the figure below, in red a typical curve for the day and in blue the current load. The difference between the two is shown in a percent window.

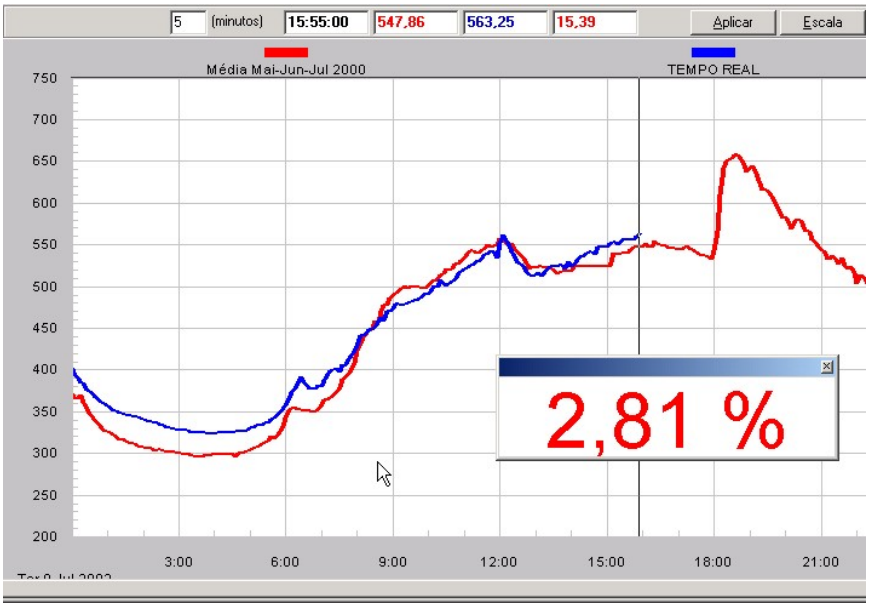


Figure 1 – Load Monitoring

To analyze the load profile the load is grouped considering its tariff classification as "industrial", "commercial", "govern", "residential class A, B, C", etc.

In order to identify the consumption, by profile of the loads, it is necessary to analyze the commercial consumer´s registers to calculate the load by profile, for each standard interval of time. The idea here is to verify by time the load profile for, for example, the energetic planning expansion and the elaboration of public policies.

ActionNET – Lean Automation

This feature groups three features into one (IED Wizard, Creating your Own IED & Mimic Panel/Face plate of the IED).

With the Lean Automation (LA) methodology, developed by Spin and available in ActionNET, it is possible to configure complex automation applications for substations, power plants, wind farms, etc. in a few minutes, guaranteeing the quality and absence of errors in parameterization.
The purpose of the methodology is to automate an application such as several substations with the following differentials:

- Time: Minor ↓ (months are converted into weeks);
- Cost: low ↓ (reduction of the staff, their specialization and the time of implementation, with their unfolding in expenses with tickets, car rental, hotel, meals, etc.);
- Quality: high ↑ (methodology minimizes failures and guarantees quality and finishing);
- Flexibility: High ↑ (from a solution you can get others, easily).

The methodology is supported by the following five concepts and functionalities of ActionNET:

(1) **Template**: template will be the basic information cell. For example, if the objective is generate an application to create substations considering all typical bays of a utility, we will have a template for each typical bay: feeder, transformer, line, reactor, etc. If the objective is an application to generate wind farms, the templates will be of wind turbines, anemometrical towers, wind farms, etc.
One template is a record that contains all points of an information cell as a typical bay, a wind turbine, anemometrical tower, etc.
However, in LA, each template point will have all information necessary to treat this point as, for example, its address inside the IED, alarm and historical conditions, nomination rules, etc.

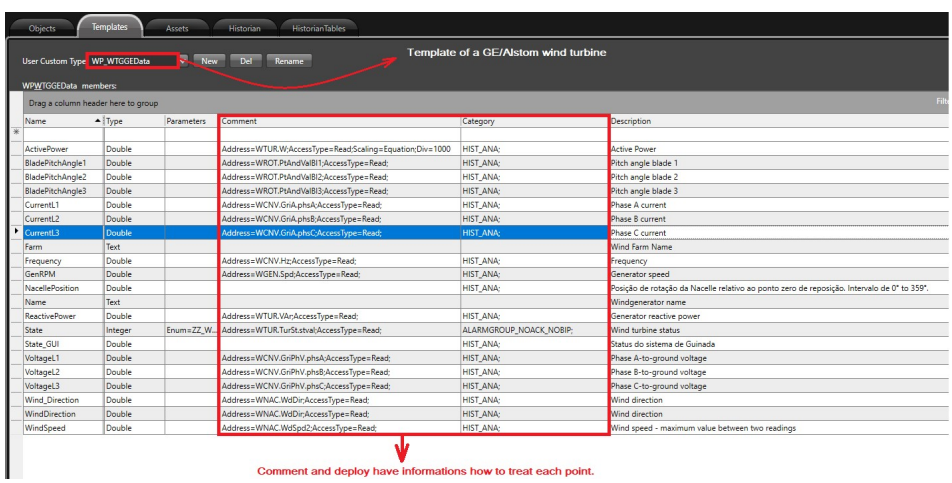


Figure 1 –Template of a Wind Turbine

(2) **Category:** the software has an array of properties, created at design time that can be used in the LA projects to associate the points to functionalities. So, we can associate several of these available properties to three different functionalities:

a) Alarm: sets the type of alarm associated with one point. If, for example, my application has ten different alarm types, 10 category properties will be consumed to represents these alarms;

b) History: defines the conditions that cause the recording of the current state/value of the point in a history record. Again, as many properties will be consumed as are the historical recording conditions;

c) Points in Devices: the user must manually create the channels and nodes of the Application, according to the characteristics of the protocols used. This parameterization should be done according to the IEDs with which the project will communicate.

For each existing node, the system will automatically create a category with the prefix NODE_ followed by the node name.

(3) **Category to Template Association**: as shown in the figure below, in the templates, double clicking on the column category of each point opens a window where all the categories created in this application are shown and the user associates one or more categories to each Point of the template. Thus, for each point its alarm type is defined, the condition for its recording in historical files and the information referring to the devices, nodes, type of point in the protocol and address 2 of the point (address 1 is associated with the device).

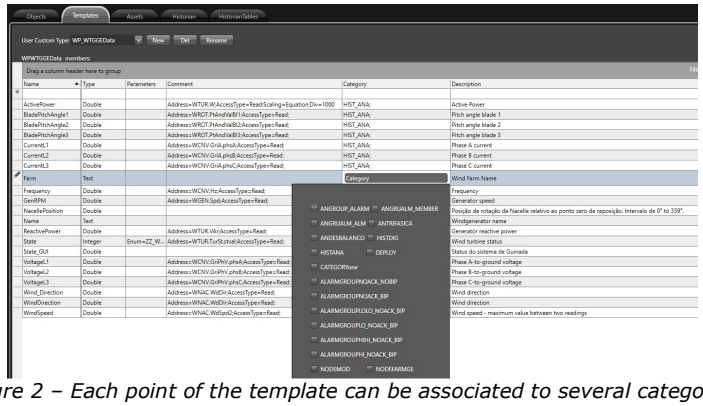


Figure 2 – Each point of the template can be associated to several categories

(4) **Linking the template name to the library symbol**: ActionNET has a functionality oriented to the development of the methodology that is, if a template has the name of a symbol from the symbol library, is created a link between them and the mapping of template variables is associated with symbol mapping. Thus, in the wind farm application, WP_GE01 to WP_GE06 is associated to the template WP_WTGGEData and corresponds to six GE´s wind turbines. If we copy one wind turbine register and paste in a screen draw, the symbol of a wind turbine, named "WP_WTGGEData" will be inserted in the screen and all points of it will be associated to that symbol.

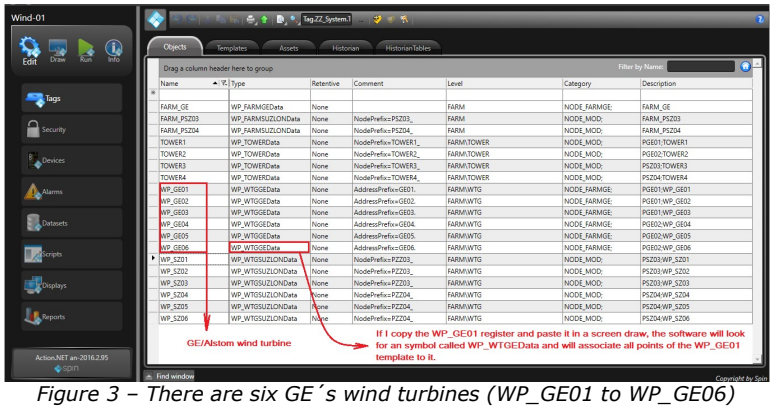


Figure 3 – There are six GE´s wind turbines (WP_GE01 to WP_GE06)

(5) **Extensions Deploy**: extensions are program libraries developed in the Dot NET environment that, at project time, allow you to import, export, and process data associated with an application. In the case of Deploy, when it is executed it scans the Tag/Object lines and, for each line that has objects to be exploded (associated category), it automatically generates the alarm tables, history and points of devices. After that, it deletes the categories associated with that line. This allows that in the future, when creating a new span, only it will be exploded, since the others do not have more associated categories.

After the development of LA methodology, we developed the concept of component that is a project or one module of a project with some functionalities. At the end, each LA project were transformed in a component.
Bellow we point to some sample of LA components:

- Lean automation – Component concept;
- Lean Automation – Customization of a power utility substation;
- Lean Automation – Wind power component;
- Lean Automation – Asset control component;

ActionNET – Load Curtailment (Reduction)

Load Reduction is implemented through an algorithm that provides a set of loads, with priorities, and conditions for its cutting.

For example, in the figure below, there are four conditions of performance of the Load Curtailment Algorithm:

(1) TG2 source failure;

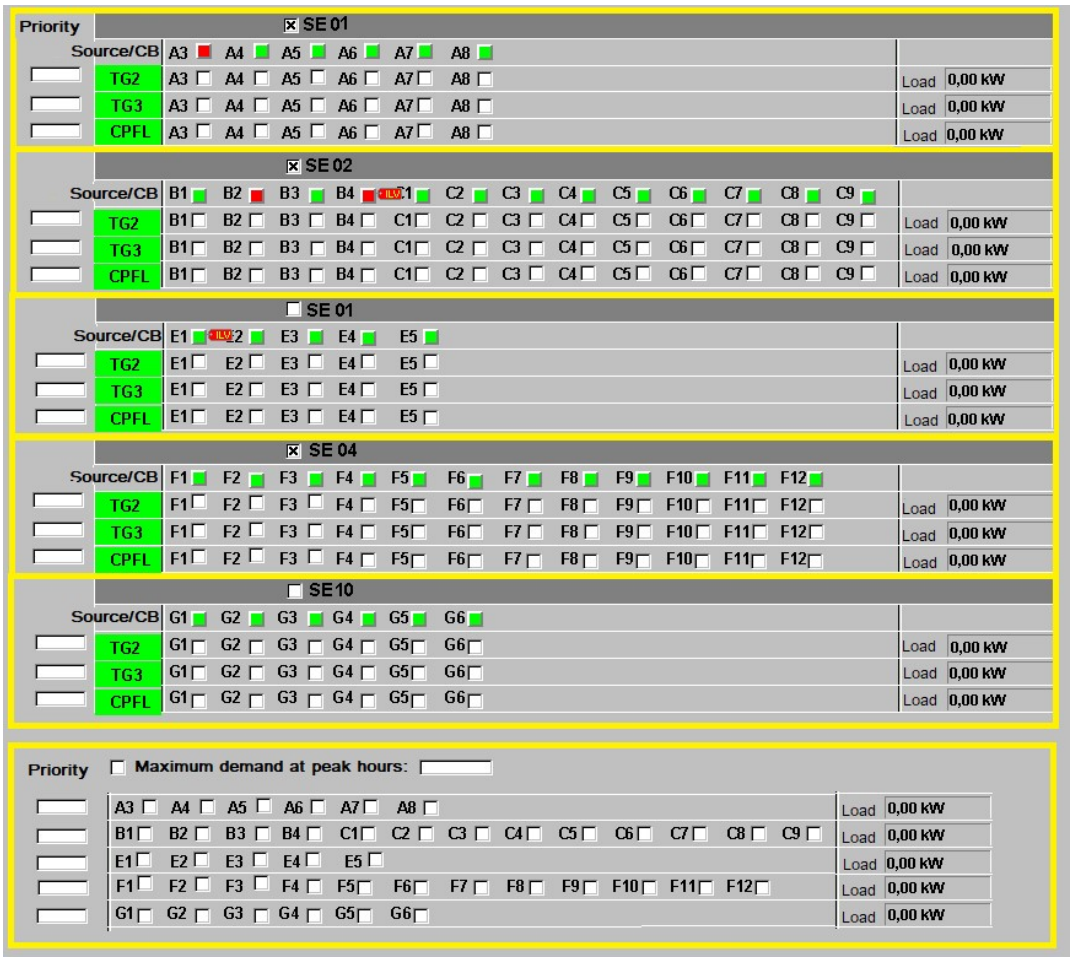
(2) TG3 source failure;

(3) CPFL source failure;

(4) Demand achieving the peak.

For each fault, we can select the Circuit Breakers that will be turned-off depending on their priority (1º - highest priority / 3° - lowest priority).

On the right side of the lines is shown the sum of CB loads selected. Therefore, we always know how many load will be rejected in each substation depending on the kind of fail. The operator can add / remove loads, just changing the selection.



ActionNET – Default Reports

The ActionNET empty project has a default layout with a header and eight reports, as shown below:

Figure 1 – Reports header and the currents alarms report

The header symbols are explained in the table below.

Back Go to previous screen	Tags Show selected tags (use filter)
Start Go to first screen	Events Measures Historian Show historic events / measures
Alarms Currents alarms (1 not recognized)	Trends Real time and historic trends
Events Events of the day	Filter Open a filter for reports: Selected tags, historian events, Historian measures, real time and historic trends.
Buzz Turn off audible alarm	

The screen below is a real time trend report with the filter window opened.

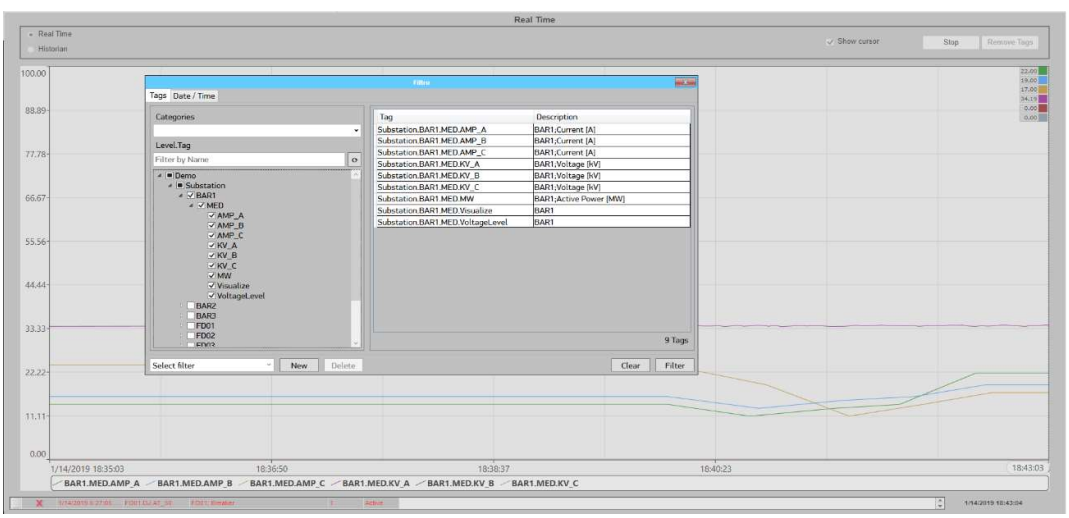


Figure 2 – Real time trend and Windows's filter

There is a component in the Default-New-Project called "Spin" that generated all these reports as well as have two windows that can be used to command switches and circuit breakers and insert operations annotations (see Annotations windows and list at online manual).

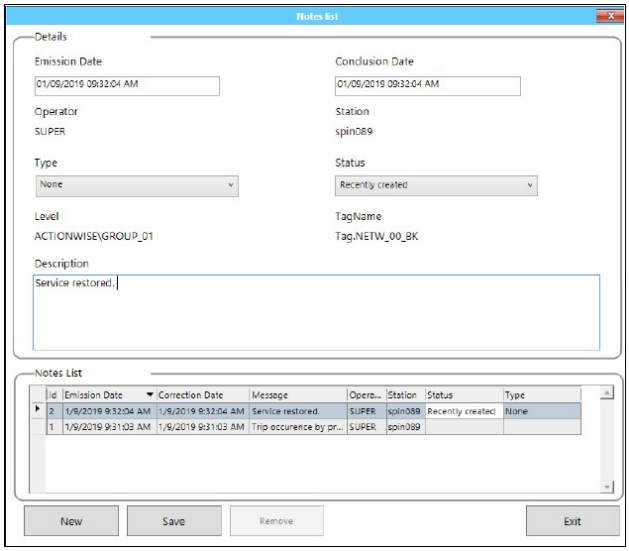


Figure 3 – Windows Annotation

ActionNET – Maneuvers

Maneuvers is a functionality available in the default-new-project whose the purpose is to define maneuvers, that is, sequential sets of remote commands that are normally used in the operation of electrical systems.

A window has been implemented in which you can define a new maneuver, change one already defined, exclude maneuvers and show an already registered maneuver. Finally, the window also allows the sending of a request to execute an already existing maneuver.

In the ActionNET online manual, in the section "Using the Maneuver Window" you will find detailed information how to use it in your project.

Below is showed an existing maneuver were in the upper left there is a box with the list of all the maneuvers registered in this project. The name that appears in the box is the name of the maneuver currently being displayed in the data grid, which is in the main part of the window.

In this mode of using only "**Show**" maneuver data, all grid fields with data are read-only type. They can not be changed.

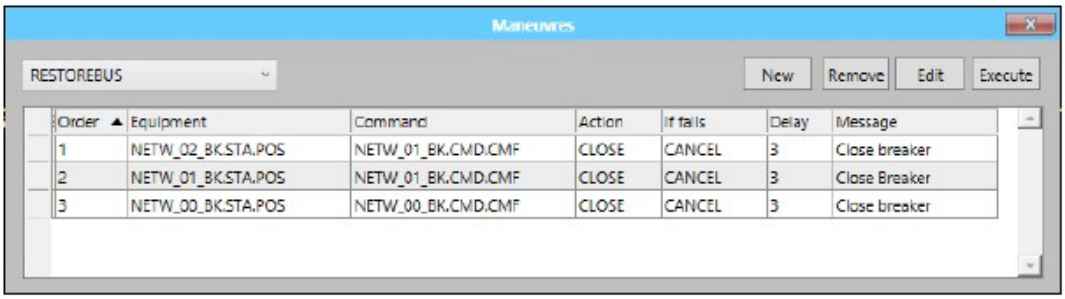


Figure 1 – Maneuvers windows

Clicking on the List-Box shows the list with the maneuvers. If an item is selected from the list by clicking with the mouse, this maneuver will have its data shown in place of the previous data:

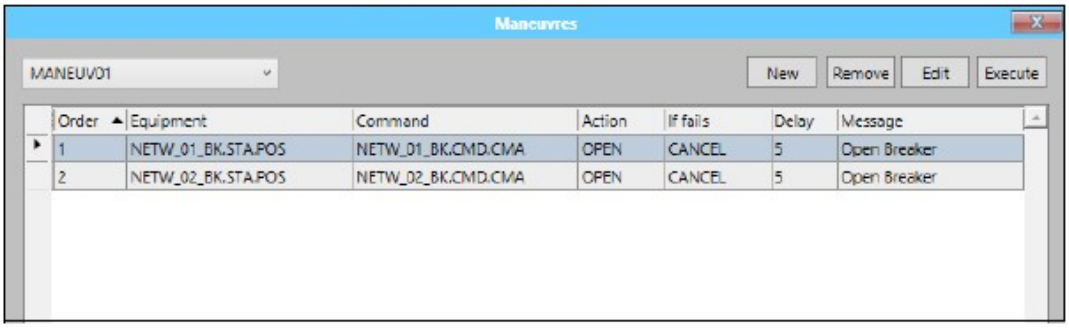


Figure 2 – Windows with another maneuvers

All substations maneuvers can be implemented using these functionalities. Using Scripts the user can, automatically, run a maneuver when some operations conditions occurs.