

μServer

User Manual



TABLE OF CONTENTS

I. INTRODUCTION.....	10
II. SOFTWARE OVERVIEW.....	11
1. Advantages of µServer.....	11
2. Architecture of the software.....	12
III. USE OF µSERVER.....	13
1. Getting started.....	13
1.1. Connection to the web interface of µServer.....	13
2. User rights to the web interface.....	13
2.1. Administration of user profiles.....	14
2.2. Administration of user accounts.....	18
3. Devices.....	20
3.1. Communication.....	20
3.1.1. Communication in "passive" mode.....	21
3.1.2. Communication in "active" mode.....	24
3.2. Administration of equipment.....	26
3.3. Device operation.....	32
4. Variables.....	34
4.1. Creation of variables.....	34
4.2. Identification of variables.....	35
4.2.1. Variables address.....	35
4.2.2. Variable type.....	35
4.2.3. Cross reference per mnemonic.....	37
4.2.4. Particular case: "general" variables.....	37
4.2.5. Particular case for the "passive" mode communication.....	37
4.2.6. Particular case for the "active" communication mode.....	39

4.3. Diagnosis variables for µServer.....	41
4.4. Administration of variables.....	44
4.5. Visualization/forcing of variables.....	50
4.5.1. Visualization.....	50
4.5.2. Forcing variables.....	50
4.6. Access rights to variables.....	51
5. Alarms.....	51
5.1. Definition of alarms.....	51
5.1.1. Communication in “passive” mode:.....	52
5.1.2. Communication in “active” mode:.....	52
5.2. Highlight of variables in state of alarm.....	52
5.3. Alarms views.....	52
5.3.1. View of active alarms.....	53
5.3.2. View of alarm log.....	55
5.4. Alarm acknowledgement.....	58
5.5. Alarm inhibition.....	58
5.6. Access rights to alarms.....	59
6. Events.....	60
6.1. Definition of events.....	60
6.1.1. Alarm events.....	60
6.1.2. Forcing-related events.....	60
6.1.3. Variable value change event.....	60
6.2. Events log.....	61
6.2.1. Definition of the period to be displayed.....	62
6.2.2. Event visualization.....	62
6.2.3. Display filter by variable category.....	64

6.2.4. Csv Export.....	64
6.3. Access rights to events.....	64
7. Log of values.....	64
7.1. Definition of log.....	64
7.1.1. Communication in “passive” mode.....	65
7.1.2. Communication in “active” mode.....	65
7.2. Plot charts/Histograms.....	65
7.2.1. Log mode.....	65
7.2.2. Real time mode.....	68
7.3. Bar chart.....	68
7.3.1. Definition of the period to be displayed.....	69
7.4. Value tables.....	71
7.4.1. Definition of periods.....	71
7.4.3. Display of several variables.....	72
7.5. Log filters.....	72
8. Alert reports.....	74
8.1. Administration of reports.....	74
8.2. Special fields.....	78
9. Synoptic views.....	80
9.1. Presentation of synoptic views.....	80
9.2. Administration of synoptic views.....	80
9.3. Edition of synoptic views.....	82
9.3.1. Edition via µIHM.....	82
9.3.2. Edition via the online editor.....	82
9.3.3. Generalities regarding the online editor.....	84
9.3.4. Background of synoptic views.....	86

- 9.3.5. Synoptic Objects..... 87
- 9.3.6. Synoptic layers..... 103
- 9.3.7. Dynamic properties of objects..... 104
- 9.3.8. Special controls..... 106
- 9.3.9. Global variables of a synoptic..... 109
- 9.4. Synoptic visualization..... 110
 - 9.4.1. Associating a synoptic to a device..... 110
 - 9.4.2. Associate a synoptic to a user account..... 110
 - 9.4.3. Default synoptic..... 111
- 10. Scripts..... 111**
 - 10.1. Administration of scripts..... 112
 - 10.2. Programming..... 114
 - 10.2.1. Principle..... 114
 - 10.2.2. µServer functions..... 114
 - 10.2.3. Debug console..... 119
- 11. Languages..... 120**
- 12. Hour and date..... 121**
 - 12.1. Timestamp of the data..... 121
 - 12.2. Timezones..... 121
- 13. Generalities of µServer interface..... 122**
 - 13.1. Search filters..... 122
 - 13.1.1. Edition of elements..... 124
 - 13.1.2. Multiple editions..... 124
 - 13.1.3. Previous button..... 125
- IV. ANNEXES..... 126**
 - 1. Annex A. Revision history..... 126**

INDEX OF EXAMPLES

Example 1. Variable reference in µServer.....	36
Example 2.Variable reference to another µServer.....	37
Example 3.Reference to word bits.....	38
Example 4.Virtual variables.....	39
Example 5.Associating several events to a same variable.....	39
Example 6. Managing events.....	60
Example 7.Filters.....	72
Example 9.Property replaced by a variable.....	103
Example 10.Animated property on the main variable.....	104
Example 11.Display text list.....	104
Example 12. Javascript instructions.....	107
Example 13.Timezones.....	118

INDEX OF FIGURES

Figure 1. Architecture of the software.....	11
Figure 2. Login in µServer.....	12
Figure 3. µServer header: user accounts and user profiles on fourth and fifth positions from the left.....	13
Figure 4. Active alarms submenu options.....	54
Figure 5. Alarm log submenu options.....	57
Figure 6. Form to set the Y-axis scale of a variable shown in a plot graph.....	66
Figure 7. Log and real time visualization modes and time period form for curve visualization.....	67
Figure 8. Form to set the Y-axis scale of a variable shown in a bar chart.....	69
Figure 9. µServer edition tool bar.....	81
Figure 10. Object showing the resizing points on the contour.....	84
Figure 11. µServer variable search filter.....	119
Figure 12. µServer variable search filter with device field list active.....	119
Figure 13. Set of colors: red (left) and orange (right).....	120

INDEX OF TABLES

Table 1. Features available in the µServer user profile.....	16
Table 2. Available features for the user account.....	19
Table 3. Equipment features available in µServer.....	28
Table 4. Sub-menu options once entered the Devices view of mServer.....	33
Table 5. Type of standard variables for µServer.....	36
Table 6. µServer diagnosis variables.....	42
Table 7. List of variable features.....	46
Table 8. Fields shown in the active alarms table.....	54
Table 9. Fields shown in the alarm log table.....	57
Table 10. Fields shown on the event table.....	63
Table 11. Available description fields in the report section.....	76
Table 12. Special fields available.....	79
Table 13. Properties available on the synoptic background.....	87
Table 14. Properties available on Image objects.....	88
Table 15. Properties available on Push-button objects.....	89
Table 16. Properties available on Toggle-button objects.....	90
Table 17. Properties available on Text objects.....	92
Table 18. Properties available on Select box objects.....	93
Table 19. Properties available on Text entry objects.....	95
Table 20. Properties available on Text area objects.....	97
Table 21. Properties available on Gauge objects.....	98
Table 22. Properties available on VU meter objects.....	100
Table 23. Properties available on Rectangle objects.....	103
Table 24. Properties available on Ellipse objects.....	103

Table 25. μServer functions available for synoptics..... 108

Table 26. Configurable properties on μServer scripts..... 113

I. INTRODUCTION

This document provides a guide the use of µServer V8 software tool dedicated to the teleprocessing and proper operating of all kinds of installations and equipment.

Document version	08/08/14
-------------------------	----------

©SIREA

All Rights Reserved

No part of this document or any of its contents may be reproduced, copied, modified or adapted, without the prior written consent of the author, unless otherwise indicated for stand-alone materials.

©SIREA - 69 Rue de l'Industrie - ZI DE MELOU - 81100 CASTRES-FRANCE

Tél. +33 (0)5 63 72 93 92 Fax: + 33 (0)5 63 72 93 19

II. SOFTWARE OVERVIEW

μServer is a software tool dedicated to the teleprocessing and proper operating of all kinds of installations and equipment.

It enables the local or remote control of technical installations (sensors networks, centralized technical management) in order to provide specifically-thought out answers to all types of supervision and teleprocessing needs (technical installations or instrumented systems).

μServer is a platform dedicated to the operating and graphic visualization of data, leaning on the advantages of a web server improved with SCADA-related functions. It is the ideal solution for a simple and quick operating from any internet-enabled client station.

μServer features many possibilities, among which these 3 types of standard applications:

- The central server, for the teleprocessing of numerous remote equipment
- The on-site SCADA software, for the control of PLCs (PanelPC or lone web server)
- The communication gateway towards a central server

1. Advantages of μServer

Among the advantages derived from the use of the μServer software, the following can be cited:

- Full-web solution
- Real time SCADA
- Animated graphic views
- Visualization/ forcing of variables
- Mysql database
- Alarms administrator
- Events administrator
- Sending of alarms via email/sms
- Display of curves and charts
- Export in .csv file format
- Data printing

- Embedded development tool
- Graphic views editor
- Unlimited number of variables

2. Architecture of the software

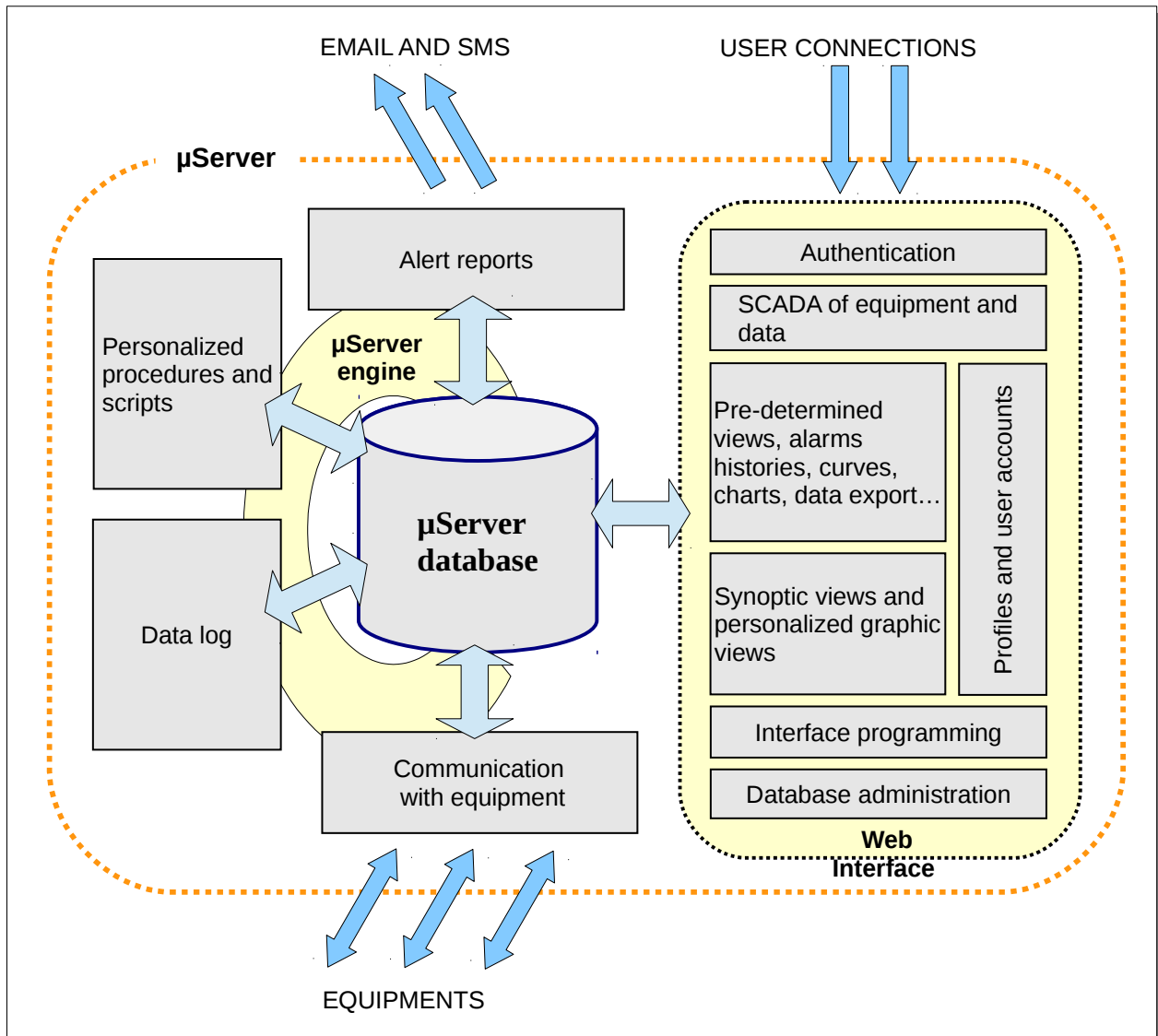


Figure 1. Architecture of the software.

III. USE OF μSERVER

1. Getting started

1.1. Connection to the web interface of μServer

μServer can be accessed and used from an Internet browser.

In order to be connected to μServer, you only need to type the IP address or the server hostname in the browser, or you can use a bookmark created on a previous connection.

Note:

For a local SCADA-type application, its use also requires an Internet browser, but launched on the engine itself. It is possible to configure the engine so that the browser automatically opens itself on the μServer interface, or you can simply create a keyboard shortcut. Besides, Sirea proposes a “full screen” mode for PanelPC-based installations.

2. User rights to the web interface

While connecting to μServer, the first view is an authentication request.

Vous devez vous identifier pour accéder à cette page

Utilisateur:

Mot de passe:

Se connecter

[← Précédent](#) [Impression](#)

Figure 2. Login in μServer.

However, on the first access to μServer after installation, user access is not configured, which enables to access the interface without having to log in. In this case, any user is considered as the administrator.

The management of access rights-related procedures is based on the combination of “[User accounts](#)” and “[User profiles](#)”. A user account must be associated to only one user profile. Several user accounts can be associated to a single user profile.

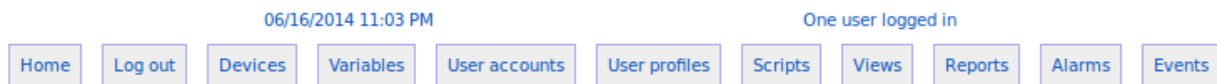


Figure 3. µServer header: user accounts and user profiles on fourth and fifth positions from the left.

Once connected, the name of the user account appears on the upper-right corner of the µServer header. This header also features the number of connected users left to the active user account. Clicking on this last option allows to access to the connections history. The access to that history is submitted to a specific user right (see section “3.2.1 Administration of user rights”).

The authentication session is temporarily stored by the web browser. The following actions enable to close that session:

- Click on the “[Log out](#)” button
- Close the browser
- Delete all sessions in the browser menu
- Delete the user account under way

2.1. Administration of user profiles

To create a new profile:

[User profiles](#) → Then click on [Add](#) button

To visualize a profile:

[User profiles](#) → Then click on [Search](#) button. A list of profiles will appear. Click next on the profile you may want to visualize.

To modify one or several profiles:

[User profiles](#) → Click on the [Search](#) button, then on the desired profile and then on the “Modify” button.

You can also modify different profiles at the same time by doing:

[User profiles](#) → Click on the [Search](#) button, then select with a tic the desired profiles to be modified and then on the [Edit](#) button.

To copy a profile:

[User profiles](#) → [Search](#) → Click on the profile → [Copy](#)

To delete one or several profiles:

[User profiles](#) → [Search](#) → Click on the profile → [Delete](#)

Or to delete several profiles at the same time:

[User profiles](#) → [Search](#) → Select which profiles to delete → [Delete](#)

Note:
All these actions require administrator rights.

Table 1. Features available in the µServer user profile.

User Profile	
Features	Description
Name	Name of the user profile.
Can administrate the base	This right includes : <ul style="list-style-type: none"> - Access to the following tabs: "Variables", "User accounts", "User profiles", "procedures", "synoptic views", "reports", "outbox" from the main menu. - Possibility to add or copy a device - Possibility to copy, modify, delete a variable (status or measurement)
Can view connected users	Enables to consult the connections history by clicking on the number of persons connected, on the upper header.
Can activate objects	Access to "clickable" items on synoptic views (buttons, entry areas, drop-down lists...).
Can set variables values	Access to the forcing function from the visualization of statuses or measurements, or from the visualization of a variable.

User Profile	
Features	Description
Can view logs	<p>–Access to the tab " Events", either from the main menu or from a variable.</p> <p>–Access to the link "see alarms history" thanks to the visualization of alarms from the main menu or from a variable .</p> <p>–Access to links "histogram", "curves", "bar charts", "value tables" from the visualization of statuses and measurements, or from the visualization of a variable.</p>
Log accessibility period (in days)	<p>Limitation of the number of days available in histories of alarms and events, along with the following views: "histograms", "curves", "bar charts", "value tables".</p> <p>No limitation if the field is left empty.</p>
Can view device information	Enables to mask most data related to the view " Information" of a device.
Can edit device information	This right grants a user the possibility to modify or delete a device, even if this user is not the administrator.
Can view boolean variables	Access to the tab "statuses" of all equipment.
Can view numeric variables	Access to the tab "measurements" of all equipment.

User Profile	
Features	Description
Category filter	<p>Access filter to variables per category</p> <p>If the field is empty, there is no filter on variables.</p> <p>If the field is a text, or list of texts separated by “,”, then the only accessible variables are those whose category is included in the list.</p> <p>If the field begins with “/”, then the logic is inverted: the user will have access to all categories excepting the specified category or categories after “/”.</p> <p>Example: “/Masked, Admin” allows the user access to all the variables but those inside the “Masked” and/or “Admin” categories.</p> <p>Access to variables can be applied to boolean variables, numeric variables, alarms and events.</p> <p>Categories of a variable are defined on the variable properties (see section <i>4.4 Administration of variables</i>)</p>
Can view alarms	Access to alarms views (both current and logged) from the main menu, a device menu or a variable view.
Can acknowledge alarms	Right to acknowledge alarms from alarm views. (see section <i>5.4 Alarm acknowledgement</i>)
Can inhibit alarms	Right to mask an alarm by clicking on it. (see section <i>5.5 Alarm inhibition</i>)
Can call devices	Access to the button "Update this equipment" from the tab "information" of a device.
Access to layers 1 to 10	This right enables to define visible/masked layers on synoptic views. Indeed, each synoptic element can be visible only on certain layers among the 10 existing, a synoptic actually being the superposition of 10 transparent layers. (see section <i>9.3.6 Synoptic layers</i>)

2.2. Administration of user accounts

To create a new account:

[User accounts](#) → [Add](#)

To visualize an account:

[User accounts](#) → [Search](#) → Click on the account

To modify one or several accounts:

[User accounts](#) → [Search](#) → Click on the account → [Edit](#)

Or:

[User accounts](#) → [Search](#) → Select which accounts to modify → [Edit](#)

To copy an account:

[User accounts](#) → [Search](#) → Click on the account → [Copy](#)

To delete one or several accounts:

[User accounts](#) → [Search](#) → Click on the account → [Delete](#)

Or:

[User accounts](#) → [Search](#) → Select which accounts to modify → [Delete](#)

Note:

All these actions require administrator rights.

Table 2. Available features for the user account.

User Account	
Feature	Description
User Name	Name of the user account, which also serves as the authentication login. This account name is the one used to gather alert reports (see section <i>8.1 Administration of reports</i>).

User Account	
Feature	Description
Password	Password necessary to the authentication procedure.
Profile	Selection of the profile associated to this user account. Note: If there is only one profile, this field does not appear and the account is automatically associated to that profile.
Regional format	Linguistic parameter of the account. (see section <i>11 Languages</i>)
Email address	Email address to send alert reports. (see section <i>8.1 Administration of reports</i>) It is possible to specify several addresses by separating them with commas.
Phone number	Telephone number to send alert reports. (see section <i>8.1 Administration of reports</i>) It is possible to specify several numbers by separating them with commas.
Zone filter	Access filter to equipment, per area It enables to limit the access to all equipment belonging to a same area. If the field is empty, there will not be any filter on the area. If the field is a text or a list of texts separated by characters such as “,” then only the equipment which area is on the list will be accessible. The equipment area (also called maintenance area), can be modified from the tab ‘information’ of this equipment.
Device filter	Access filter to equipment, per identification number (identifier). If the field is empty, there will not be any filter on the equipment

User Account	
Feature	Description
	number. If the field is a number or a list of numbers separated by characters “,” then only equipment which identifiers are on the list will be accessible.
View	Welcome synoptic view. If the name typed in corresponds to an existing synoptic, the latter appears on the home page of the user account.
Always show main menu	Choose “Yes” to show the main menu on the header or “No” to hide it.

Note:

A user who does not have the required administrator right to access the database cannot access the [User accounts](#) menu, but he can still modify certain characteristics of his own account via the [User accounts](#) menu.

3. Devices

3.1. Communication

µServer ensures communication in real time or off-line, locally or remotely, either with a whole set of devices or a single one.

Devices can be:

- SIREA automation platforms such as µFox
- Any automation system module including the Modbus protocol, through a µNPE-type gateway
- Any automation system module including the Modbus protocol, and directly linked to the µServer engine.
- Other µServer SCADA installed on industrial computers for example

Equipment can indirectly communicate with μServer via a hierarchic communication system. For example:

- 2 μFox manage several PLCs and report all information to a global SCADA system.
- 10 μNPE serve as gateways between several sensors and a central one.
- ...

In this case, we consider as an “equipment” within μServer, the module which directly communicates with μServer. In the case where a μNPE gateway is used, the equipment within μServer is μNPE, unlike the modules connected to μNPE.

Each equipment within a μServer application features a unique equipment number. This number (also called index or identifier) enables to distinguish all equipment within the database, to address variables, to filter equipment per users, to manage communication with several equipment...

There are two main communication modes between equipment and μServer:

- “passive” mode
- “active” mode

3.1.1. Communication in "passive" mode

3.1.1.1. Principle

This mode is used for central server-type applications.

The communication is called “passive” given that equipment connect themselves to μServer, as opposed to the “active” communication, where μServer connects to equipment to interrogate them.

This communication mode can as well be called “client-server”, which differs from the “master-slave” of active communication. It can also be considered as “remote” mode, as opposed to active communication, mostly used locally.

The communication is based on TCP-IP protocols. It is independent from the network access support (Ethernet, GPRS/3G, ADSL...).

Each equipment is configured with the following elements:

- its own equipment number
- its name and type

- the IP address (or hostname) of the server
 - a reconnection timeout
 - a list of variables to make accessible from the server
- 1) As soon as a device starts, it connects itself to the server and indicates its identification number, name and type. If this equipment does not exist on the server, it is automatically created. If it already exists, its features are updated if necessary (for example: change of name).
 - 2) Then the equipment sends its list of variables, along with their characteristics.
 - 3) In the case where a variable does not exist, it is automatically created. Likewise, if its characteristics have been modified (comment of a variable for example), they are updated on the server.
 - 4) Once the connection established, the equipment switches to a listening mode and awaits the server requests. Therefore, the connection remains effective between the equipment and the server. Among those requests, the following can be found:
 - Update requests (see section [3.1.1.2 Data updates](#))
 - Forcing (or writings) generated by µServer (see section [4.5 Visualization/forcing of variables](#))

In sum, the connection is always initialized by a device, but requests are initialized by the server.

All concerned equipment are permanently connected to the server but the related data are not necessarily exchanged permanently.

3.1.1.2. Data updates

By default, the server permanently updates the whole set of connected equipment. However, you can pre-set an update period or even forbid any update (see “**Refresh delay (in seconds)**” in the [3.2 Administration of equipment](#)) . This is particularly useful when data exchange must be limited, like in the case of a 3G/GPRS subscription, usually charged on a data basis.

As an answer to the update request, the equipment sends:

- Current values related to all variables which value has changed since the last update. It is to be noted that the first update consequent to the equipment starting process causes it to send the values of all its variables.
- The whole set of history points (measurements, alarms, events) that have been generated since the last update.

The server requests a data update:

- Cyclically when the update period is reached, or permanently if such a period has not been defined.
- Occasionally when a user “calls” the equipment (button on the [Devices](#))
- Occasionally when the equipment reinitializes the connection in order to force an update request (if an alarm has been reported for example).
- Permanently when a user visualizes a device (if the AUTO REFRESH mode is activated).

The equipment can however force an update request. It is the case when an alarm is being reported.

3.1.1.3. Definition of equipment and their variables

In this communication mode, it is not necessary to create equipment and variables on the server, since everything is automatically done during the first connection of the equipment.

Once a device or a variable is identified by the server, you can complete its description on the server. Indeed, certain characteristics of equipment and variables are defined on the equipment while others are defined on the server.

For instance, the “**Name**” of the equipment is defined on the equipment. You can modify it from the server, but it will always be reset each time the equipment will reconnect.

On the other hand, characteristics like “**Address**” and “**Postal code**” are defined on the server. Please consult the characteristics lists of equipment and variables for more details.

Likewise, if you delete a device or a variable from the server, it is re-created when the equipment is reconnected and sends its characteristics.

In particular, once the equipment has been created on the server, it is important to fix the “**Refresh delay**” and “**Timeout**” options if the default values are not appropriate.

3.1.1.4. Management of communication errors

As long as a device is not able to connect to the server, it re-tries.

If a device is connected to the server, but does not receive any order for some time, it resets its connection. This equipment “**timeout**” is configured within the equipment and not within the server, but it should be in line with the “**Refresh delay**” configured in the server.

If the server does not receive any answer from the equipment for some time, it will report a communication default (see section [4.3 Diagnosis variables for μServer](#)). This “server timeout” is configured within the server (see section [3.2 Administration of equipment](#)).

3.1.2. Communication in "active" mode

3.1.2.1. Principle

It is the "classic" communication mode between a SCADA and a local equipment. μServer masters the system and permanently interrogates slave devices.

The MODBUS protocol is very used, but other protocols are available (see "**Communication protocol**" property at equipment characteristics in [3.2 Administration of equipment](#)).

The communication is called "active" because μServer connects to equipment in order to interrogate them, as opposed to "passive" communication, where equipment connect to μServer.

We can consider this communication mode as "master-slave", as opposed to the "client-server" mode of passive communication. We might as well call it "local" mode, which differs from "passive" communication, broadly used via Internet.

This mode functions for equipment that are directly connected to the μServer engine via a serial port (RS232, RS485) or an Ethernet port, or thanks to remote equipment that feature an accessible IP address.

Note:

The communication in "active" mode is powered by the μDriver software, which must be installed in addition to μServer. If some equipment are defined in the active communication mode, μServer automatically starts μDriver.

3.1.2.2. Definition of equipment and their variables

In this mode, you have to create equipment and their variables from the μServer interface. It is the contrary of the "passive" mode, where the characteristics of equipment and variables are defined (partly) within the equipment.

Please see section [3.2 Administration of equipment](#) and [4.4 Administration of variables](#) for all information regarding the creation of equipment and variables.

3.1.2.3. Management of communication errors

The μDriver software manages the communication process with devices.

It indicates μServer the time period since the last "OK" exchange with the equipment. If this period is too long in comparison with the equipment **timeout** set ([3.2 Administration of equipment](#)), μServer reports the communication error.

Communication errors are reported by the variables of the diagnosis (see [4.3 Diagnosis variables](#)

for μServer).

3.2. Administration of equipment

To create manually a new device:

[Devices](#) → [Add](#)

To access a device:

[Devices](#) → [Search](#) → Click on the device

To modify one or several equipment:

[Devices](#) → [Search](#) → Click on the device → [Edit](#)

Or:

[Devices](#) → [Search](#) → Select which devices to modify → [Edit](#)

To copy a device:

[Devices](#) → [Search](#) → Click on the device → [Copy](#)

To suppress one or several equipment:

[Devices](#) → [Search](#) → Click on the device → [Delete](#)

Or :

[Devices](#) → [Search](#) → Select which devices to modify → [Delete](#)

Note:

All these actions require administrator rights.

Table 3. Equipment features available in µServer.

Equipment	
Feature	Description
Communication	<p>Status of the communication with the equipment</p> <p>(see sections <i>3.1.1.4 Management of communication errors</i> and <i>3.1.2.3 Management of communication errors</i>)</p> <p>Communication in "passive"mode</p> <p>OK: Last exchange with the equipment AND no control under way.</p> <p>ERR: Defect reported on the last exchange with the equipment, or reached timeout AND no control under way.</p> <p>In progress: Last exchange with the correct equipment AND control under way.</p> <p>In progress: Defect on the last exchange with the equipment or reached timeout, AND control under way.</p> <p>(See sections <i>3.1.1.4 Management of communication errors</i> and <i>3.1.2.3 Management of communication errors</i>)</p> <p>Communication in "active mode"</p> <p>OK: Last exchange OK< equipment timeout</p> <p>ERR: Last exchange OK> equipment timeout</p>

Equipment	
Feature	Description
Index	<p>Unique identification number of the equipment (EQT)</p> <p>The equipment number constitutes its reference within the application.</p> <p>It enables to address the variables of a device, to filter access to equipment per user accounts, to link alert reports to an installation...</p> <p>In the case of a "passive" communication, this feature is defined within the equipment.</p> <p>In the case of a manual creation, it is advised to process manually with the attribution of a number to the new equipment. Nonetheless, if the field is left empty, µServer will automatically attribute an index to the equipment.</p>
Type	<p>Type of equipment(EQT)</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p> <p>Note: In the case of a "passive" communication, this feature is defined within the equipment.</p>
Name	<p>Name of the equipment (EQT)</p> <p>The name has to be as explicit as possible to describe the installation.</p> <p>Information regarding the equipment description (management of equipment, search filters...);</p> <p>Note: In the case of a "passive" communication, this feature is defined within the equipment.</p>
Enabled	<p>Activate/Deactivate a device.</p> <p>When the equipment is deactivated, the communication is ignored and all variables associated to this communication are no longer administrated.</p>

Equipment	
Feature	Description
Refresh delay (in seconds)	<p>Duration in seconds between two update requests for equipment data. (see section <i>3.1.1.2 Data updates</i>)</p> <p>Particular values:</p> <p>0: Permanent update</p> <p>-1: No update</p> <p>This parameter allows to limit data exchanges when such exchanges are charged, like 3G subscriptions for instance.</p>
Timeout (in seconds)	<p>Timeout of communication defect (in seconds)</p> <p>If there has been no correct exchange at the end of this duration, a communication defect is reported.</p>
Address	<p>Location address of the equipment</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>
Postal Code	<p>Location zip code of the equipment</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>
City	<p>Location city of the equipment</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>
Country	<p>Location country of the equipment</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>

Equipment	
Feature	Description
Maintenance zone	<p>Location area of the equipment</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p> <p>This feature enables to filter access rights to equipment. (see section 2.2 Administration of user accounts) and alert reports (see section 8.1 Administration of reports).</p>
Contact name	<p>Name of the person in charge of the equipment.</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>
Installation Date	<p>Installation date of the equipment (format "DD/MM/YYYY")</p> <p>Information regarding the equipment description (management of equipment, search filters...).</p>
View	<p>Name of the synoptic view associated to the equipment</p> <p>If this feature is defined, a link towards the synoptic is added to the equipment tabs (provided that the synoptic exists).</p>
Log purge delay (in days)	<p>Number of days to keep in store (in the equipment history)</p> <p>All anterior data are automatically deleted from the database.</p> <p>If this field is empty or equals to 0, no automatic deletion process will be implemented on this equipment.</p>

Equipment	
Feature	Description
Communication Protocol	<p>Parameter regarding "active"communication</p> <p>Communication protocol</p> <p>Modbus: standard modbus</p> <p>Sirea: extended modbus for sirea PLCs (distinct address areas for %MW, %MD and %MF, management of %S et %SW...).</p> <p>Ext: extended Modbus for standard PLCs (confused address areas for %MW, %MD and %MF).</p> <p>The RTU or TCP mode is automatically defined, depending on if the equipment is connected via a serial port or an IP address (communication channel).</p>
Communication channel	<p>Parameter regarding "active"communication</p> <p>Number of µDriver channel</p> <p>For a TCP/IP communication: IP address/ host name and TCP port of the slave, in "host:port"format.</p> <p>For a serial communication: number of µDriver communication ports.</p>
Slave number	<p>Parameter regarding "active"communication: Slave number of the equipment</p>
SNMP address	<p>SNMP address of the device</p> <p>Parameter for the SNMP communication.</p>

Note:

In the case of a "passive" communication, the features that have been defined within the equipment are labeled with a (*EQT*). The other parameters are defined on the server.

3.3. Device operation

To access a device view:

[Devices](#) → [Search](#) → Click on the device

Once entered the device, the following options are available in the sub-menu located just below the main menu.

Table 4. Sub-menu options once entered the Devices view of µServer.

Sub-menu options of the Devices view	
Sub-menu	Description
Information	<p>Visualization of the equipment information</p> <p>It is the list of all equipment features (see section “ Administration of equipment”).</p> <p>From this view the user can, depending on his rights (see section <u>2.1 Administration of user profiles</u>):</p> <ul style="list-style-type: none"> -Visualize only the main features of the equipment. -Visualize all features of the equipment. -Modify certain features of the equipment, copy or delete the equipment. -“Call” the equipment. It equates to updating the communication with the equipment without having to wait for the pre-determined update period.
Boolean variables	<p>Visualization of the equipment boolean variables, that is those which value can only be 0 or 1. Practically speaking, it concerns variables which type only comprises on letter (%M, %P, %D, %S...).</p> <p>From that view, the user can, depending on his rights (see section <u>2.1 Administration of user profiles</u>):</p> <ul style="list-style-type: none"> -Visualize the ongoing value of variables. -Visualize the safety status of a variable (displayed in red if an alarm is reported). -Force the value of a variable (colored button which frames the value if the variable can be forced).

Sub-menu options of the Devices view	
Sub-menu	Description
	<p>-Access the view of a variable (by clicking on it).</p> <p>-Copy/modify/delete one or several variables (thanks to administration buttons, once you have selected the variables concerned).</p> <p>-Access the history of one or several variables thanks to the Histograms button.</p> <p>-Filter the display of variables per category. The list of categories does only appear if one or several variables of the list are associated to categories.</p> <p>Note: Only the variables characterized by the feature “visible” are displayed on that view. To access non-visible variables, please go to the Variables menu.</p>
Numeric variables	<p>Visualization of equipment numeric variables:</p> <p>As opposed to the "visualization of boolean variables", this option displays all non-binary variables, that is those which value can differ from 0 or 1 (whole and decimal numbers, text...). Practically speaking, it concerns the variables which type comprises several letters (%MW,%PW,%DW ,%SW ...).</p> <p>From that view, the user can, depending on his rights (see section 2.1 Administration of user profiles):</p> <p>-Visualize the ongoing value of variables.</p> <p>-Visualize the safety status of a variable (displayed in red if an alarm is reported).</p> <p>-Force the value of a variable (colored button which frames the value if the variable can be forced).</p> <p>-Access the view of a variable (by clicking on it).</p> <p>-Copy/modify/delete one or several variables (thanks to administration buttons, once you have selected the variables concerned).</p> <p>-Access the history of one or several variables thanks to Plot charts, Bar charts, Value table buttons.</p> <p>-Filter the display of variables per category. The list of categories does only appear if one or several variables of the list are associated to categories.</p>

Sub-menu options of the Devices view	
Sub-menu	Description
	Note: Only the variables characterized by the feature “visible” are displayed on that view. To access non-visible variables, please go to the Variables menu.
Alarms	Visualization of device alarms That view enables the user to manage equipment alarms. The administration of alarms is detailed in section 5 Alarms .
Events	Visualization of device events That view enables the user to manage equipment events. The administration of events is detailed in the following section 6 Events .

4. Variables

4.1. Creation of variables

Creation depends on communication mode (see section [3.1 Communication](#)):

1) Communication in "passive" mode

Variables are defined within the device. The list is sent to the server when communication is initialized. Each variable which doesn't exist within µServer is automatically created.

2) Communication in "active" mode

User creates manually the variables within µServer (see section [4.4 Administration of variables](#))

Note:

When a new device is created on µServer, µServer creates some diagnosis variables associated to this device. These variables are automatically managed by µServer (see section [4.4 Administration of variables](#)).

4.2. Identification of variables

4.2.1. Variables address

Each variable is assigned to a single device.

To search a particular variable within µServer, use a reference address, which is the combination of 3 features: its type (or prefix), its address (or offset) and its device number (or device index).

The syntax is the succession of the following elements, put together without space:

- The character "%"
 - The type
 - The address
 - The character
 - The device number

For example, "%MW10.3" refers to the word register n°10 of the device 3.

4.2.2. Variable type

The following table show the type of variables available in µServer.

Table 5. Type of standard variables for µServer.

Types of standard variables for µServer		
Prefix	Description	Possible values
"PLC"Variables		
M	PLC bit	0 or 1
MW	PLC 16 bits word	0 to 65535 or -32768 to 32768 depending on the configuration
I	PLC TOR input	0 or 1
IW	PLC analog input	0 to 65535
IF	PLC floating analog input	Signed decimal value
Q	PLC TOR output	0 or 1

QW	PLC analog output	0 to 65535
MD	PLC 32 bits PLC	-2147483648 to 2147483648
MF	PLC floating word	Signed decimal value
S	PLC system bit	0 or 1
SW	PLC system word	0 to 65535
"SCADA"variables		
P	SCADA bits	0 or 1
PW	SCADA word	All numeric types or text
"Diagnosis"variables		
D	Diagnosis bits	0 or 1
DW	Diagnosis word	All numeric types or text

These variables are the generally used, however, variables with non-standard types can also be declared. It is not necessary to declare the type for using them. For example, a variable "%KWO" or "%LocalWO" can be created.

It is necessary to remark that certain types determine specific functionalities of µServer and are reserved:

- PLC variables (MW, Q...) are automatically managed through communication if the equipment is in "active" communication mode. (see section [4.4 Administration of variables](#))
- Diagnostic variables (D and DW) are automatically created and updated by µServer.

When some devices indirectly communicate with µServer (see section [3.1 Communication](#)), the variable address is automatically modified by the communication protocol, in order to interlock equipment numbers by separating them with the character "_" and "." which enable to determine the equipment number, within µServer only.

Example 1. Variable reference in μServer.

A μFox administrates 2 PLCs (identified as 1 and 2 within the μFox). This μFox returns its data itself on a central server, known as μFox n°120.

The word MW10 of the PLC n°1 will feature, within μServer, the following reference: `"%MW10_1.120"`

4.2.3. Cross reference per mnemonic

You can refer to a variable via its mnemonic (symbolic name) instead of its address.

In this case, the mnemonic must be unique within μServer variables. Therefore, this kind of references should be avoided in central server-type applications, where several equipment generally feature the same variables and mnemonics.

4.2.4. Particular case: "general"variables

You can define "general"variables within μServer, which will not be assigned to a particular equipment. In this case, you have to create manually the equipment-1 which is the server itself. "General" variables can be associated to that equipment-1.

In this case, the reference to variables is simplified: the equipment number preceded by "." is not specified.

For instance, after the creation of the variable %PW0, associated to the equipment-1, this variable is referred to via the syntax "%PW0" instead of "%PW0.-1".

4.2.5. Particular case for the "passive" mode communication**4.2.5.1. Device hierarchy**

When devices are indirectly communicating with μServer (see section [3.1 Communication](#)), the variable address is automatically modified by the communication protocol, so as to nest equipment numbers separated by character "_", character "." with the aim of defining a unique number of equipment inside μServer.

For example, a μFox can manage 2 PLCs identified inside the μFox as 1 and 2. This μFox would send by itself its data to a server identified as μFox n°120. In this case, the word MW10 of the PLC n°1 would have, inside μServer, the following reference: "%MW10_1.120" (Type = 'MW', Address = '10_1', Device = '120')

In particular, μServer can communicate its data to another μServer by itself. This would be the

case, for example, of several sites equipped with a μServer local supervision, all of them uploading to a remote management central server

For addressing variables inside the μServer at a higher level, a chain of “_” inside the field of the variables' address is applied.

Example 2. Variable reference to another μServer.

Variable MW0 of device n°12 is referenced as %MW0.12 (Type = 'MW', Address = '0', Device= '12') inside a μServer. If this μServer uploads itself its data, as device n°30 to another μServer at higher level, this variable will be then referenced inside this last master μServer as: %MW0_12.30 (Type = 'MW', Address = '0_12', Device = '30'). And so on.

4.2.5.2. Remote variables / local variables

When a device is connected to μServer for the first time, it transfers its variable list. This variables are then automatically created. They are then considered as **remote variables** as they are defined inside the device.

Once the device is created in μServer, it is possible as well to add supervision variables, which are associated to the device inside μServer, but which did not exist inside the device. These variables are **local variables** inside μServer.

That means, the type is then defined depending on the way that this variable has been created:

- If the variable has been created by the device at the moment of the connection, the variable will be **remote**.
- If the variable has been created by an administrator in μServer, the variable will be **local**.

The main difference between a local variable and a remote variable is the writing management:

- If μServer writes on a remote variable, this variable is not immediately updated. It is first transferred to the device via communication protocol. If the communication is active, the device writes on the variable and then transfers the new value to the server, which is then updated. If communication is not active, the writing demand remains in standby and the value in the server remains unchanged.
- If μServer writes on a local variable, the value is updated immediately.

Note:

If the device is “deactivated” (see property “active” in section [4.4 Administration of variables](#)) the communication is ignored and the writing on a remote variable is immediate.

4.2.6. Particular case for the “active” communication mode

4.2.6.1. Word bits and double words

Communication in active mode allows to directly access a bit or a group of bits of a variable.

To refer to a word bit, the following syntax is used: “%WORD:xN” where WORD is the word address and N the bit number.

Example 3. Reference to word bits.

%MW5:x3.1"(Type = 'MW', Address = '5:x3', Device = '1') expresses the bit of number 3 (that is the forth bit, as the identification number starts in 0) of word MW5 of device number 1.

Similarly, for long words, the syntax %DOUBLE:wN is used, where DOUBLE is the double address and N is the word number.

%MD0:w0 expresses the MD0 double low word while %MD0:w1 expresses the high word.

Note:

When the writing of a word bit (or double word) is carried out, µServer takes the last known value of the word, applies the modification of the bit (or word) and then transfers the writing to the device.

It is thus necessary to avoid the use of words (or double) able to be written simultaneously by µServer and device.

For example, let's imagine that the bit 0 of a word is a µServer command while bit 1 is a default generated by an automate. In a first state there is no command or default, so values in both bits are 0 and thus the word value is 0.

µServer sends then a command thus turning the bit 0 into 1. This value is first sent to the PLC. But in the meanwhile (some milliseconds), the PLC has detected a default and turns its bit 1 into value 1. The value on the PLC has changed into 2 and the writing action, which arrives later, overwrites this value writing 1. At the end of this process, the value is 1, so the command has been correctly transferred, but the default would not be visible.

4.2.6.2. PLC virtual variables

It is possible to declare several variables in µServer referencing a same variable inside a PLC (or device). To do this, the syntax “#x” should be added at the end of the variable address, where x

corresponds to the virtual variable number.

Example 4.Virtual variables.

%MW10#0.1 (Type = 'MW', Address = '10#0', Device = '1') is the virtual variable 0 referencing the word 10 of device 1.

If the word changes its value inside the PLC, µServer automatically modifies all virtual variables associated to this word.

Similarly, if a virtual variable is written, µServer modifies the word on the PLC, thus updating as well all the other virtual variables associated to this word.

This is useful, for example, when wanting to associate several event texts to a same variable.

Example 5.Associating several events to a same variable.

A word MW30 inside the device number 1 determines a machine status:

- 0: machine off
- 1: machine on
- 2: machine to sleep

Let's imagine we want to follow the evolution on the log by showing clear texts for each change in status. We would create 3 variables:

- "%MW30.1" with the event condition "v==0" and with event label "Machine stopped"
- "%MW30#0.1" with the event condition "v==1" and with event label "Machine started"
- "%MW30#1.1" with the event condition "v==2" and with event label "Machine to sleep"

Thank to these two virtual variables, it is possible to manage three different texts for three different events associated to a same variable inside a same device.

4.2.6.3. PLC variables / supervision variables

When creating a variable into a device which communicates in "active" mode, two cases can be distinguished:

- The variable type is managed by communication (examples: "MW", "M"). This is the case of a

PLC variable.

- The variable type is not managed by communication (examples: "PW" or any other type non PLC). This is the case of a supervision variable.

See section [4.2.2 Variable type](#) for further information.

The main difference between a PLC variable and a supervision variable is the writing management.

- If µServer writes on a PLC variable, this variable is not immediately updated. It is first transferred to the device via communication protocol. If the communication is functional, the device writes on the variable, the new value is then read by µServer and updated. If communication is not functional, the writing demand remains in standby and the value in the server remains unchanged.
- If µServer writes on a supervision variable, the value is updated immediately.

Note:

If the device is "deactivated" (see property "active" in section [4.4 Administration of variables](#)) the communication is ignored and the writing on a remote variable is immediate.

4.3. Diagnosis variables for µServer

The next table shows

Table 6. µServer diagnosis variables.

µServer diagnosis variables	
Variable	Description
%D1	<p>Communication error with the equipment – " passive" Mode</p> <p>0 : No defect</p> <p>1 : defect</p> <p>If the equipment has been automatically created via a connection in " passive" mode, the defect is reported when this equipment does not proceed with a request beyond the pre-set timeout.</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer.</p> <p>By default, , this variable is determined as an alarm</p>
%D2	<p>Status of the communication with the equipment – " active" Mode</p> <p>0: no request under way</p> <p>1: request under way (currently emitting or awaiting a reply)</p> <p>If the equipment has been automatically created via a connection in "passive" mode, this variable is automatically updated by µServer.</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer.</p>
%D3	<p>Communication defect with the equipment – " active" Mode</p> <p>0 : No defect</p> <p>1 : defect</p> <p>If the equipment communicates in "active" mode, the defect is managed via the comparison between the last frame OK (sent by µDriver) and the timeout (configured on µServer).</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer.</p> <p>By default, this variable is determined as an alarm.</p>

µServer diagnosis variables	
Variable	Description
%DW1	<p>Request-reply duration(in seconds) – " passive"Mode</p> <p>It enables µServer to detect timeout overruns.</p> <p>If the equipment has been automatically created via a connection in “passive” mode, this variable is automatically updated by µServer.</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer</p>
%DW2	<p>Duration (in seconds) since the last correctly-communicated request with the equipment – " passive"Mode</p> <p>If the equipment has been automatically created via a connection in “passive” mode, this variable is automatically updated by µServer.</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer</p>
%DW3	<p>Duration (in seconds) since the last correctly-communicated frame with the equipment – " active"Mode</p> <p>If the equipment communicates in “active” mode, this variable is automatically updated by µServer.</p> <p>Otherwise, this variable is not created, nor is it administrated by µServer.</p>

Once created by µServer, these diagnosis variables can be manually modified. For instance, it can be interesting to modify the comment of the variable or the parameters log to generate curves showing the evolution of %DW2.

4.4. Administration of variables

To manually create a new variable:

[Variables](#) → [Add](#)

To access a variable:

[Variables](#) → [Search](#) → Click on the variable

To modify one or several variables:

[Variables](#) → [Search](#) → Click on the variable → [Modify](#)

Or:

[Variables](#) → [Search](#) → Select which variables to modify → [Modify](#)

To copy one or several variables:

[Variables](#) → [Search](#) → Click on the variable → [Copy](#)

Or:

[Variables](#) → [Search](#) → Select which variables to copy → [Copy](#)

To delete one or several variables:

[Variables](#) → [Search](#) → Click on the variable → [Delete](#)

Or:

[Variables](#) → [Search](#) → Select which variables to delete → [Delete](#)

Note:

All these actions require administrator rights.

Table 7. List of variable features.

Variables	
Feature	Description
Value	<p>Current value of the variable</p> <p>According to the type of variable, the value can be a whole or a decimal number, a text...</p> <p>The value of such a variable can be updated in different ways:</p> <ul style="list-style-type: none"> -Communication with equipment -Forcing from the interface -Scripts -Automatic management by µServer (concerning diagnosis variables) -Etc...
Last change	<p>Date and time of the last change of value</p> <p>This feature is automatically updated by µServer.</p>
Device	<p>Equipment number upon which depends this variable(EQT)</p> <p>This feature forms part of the address identification of the variable (see section "<i>3.4.1 Identification of variables</i>").</p>
Type	<p>Type of variable</p> <p>This feature forms part of the address identification of the variable (see section "<i>3.4.1 Identification of variables</i>").</p>
Address	<p>Address (or offset) of the variable (EQT)</p> <p>This feature forms part of the address identification of the variable (see section "<i>3.4.1 Identification of variables</i>").</p>
Label	<p>Symbolic name of the variable. (EQT)</p> <p>This feature enables to identify the variable by mnemonic (see section "<i>3.4.1 Identification of variables</i>").</p>

Variables	
Feature	Description
Comment	<p>Comment (or wording) of the variable(EQT)</p> <p>This text appears on the operator pages of the interface (statuses, measures, curves, exportations...).</p>
Public	<p>Display/mask variables on the equipment interface (EQT)</p> <p>If the variable is not visible, it does not appear in the list of statuses/measures of the equipment.</p>
Writable	<p>Authorize the writing of the variable from µServer(EQT)</p> <p>“Forced” variables appear on the interface along with a modification button in the list of statuses/measures of the equipment.</p> <p>The variable must be “forced” if the writing is done from a synoptic button, from the variables properties or the list of statuses/measures... However, a variable can be modified by a script (procedure) even if it is not defined as “forced”.</p>
Min. write value	<p>Configuration of the forcing range (EQT)</p> <p>Leave this field empty if there is no forcing limitation.</p>
Max. write value	
List of categories	<p>Categories of variables</p> <p>This parameter enables to classify variables in categories, which helps to :</p> <ul style="list-style-type: none"> -Configure a better display filter in the list of statuses/measures of a device. -Filter alarm reports by categories of variables. A variable can belong to several categories, if you separate their names with commas. <p>Available categories do not require to be created, you only need to add a name in this field to define a new category.</p>

Variables	
Feature	Description
Format	<p>Format for the display of the variable</p> <p>You can precede or follow the value of a variable with texts. This is particularly useful if you need to display variable units.</p> <p>The channel "%v" refers to the value of a variable.</p> <p>For instance, if you want to display a value in bar, you then have to define the following format: "%v bar".</p>
Float precision	Number of decimals to display after the comma (EQT)
Min. display value	<p>Configuration of the display range of the variable (EQT)</p> <ul style="list-style-type: none"> -Concerning curves: the default range -Concerning graphic templates such as gauges or dials: the reference variation range -Concerning certain properties of graphic templates: possibility to determine a scalable value
Max. display value	
Min. raw value	<p>Configuration of a scaling process, between the real value within the equipment and the value calculated within µServer</p> <p>For example, if a PLC features a temperature measurement in tenth of a degree whereas you need to display it in degree, you will have to type:</p> <p>Min. raw value = 0</p> <p>Max. raw value = 100</p> <p>Min. scaled value = 0</p> <p>Max. scaled value = 1</p> <p>Thus, 2530 tenths of a degree on the PLC will naturally correspond to 25.30 degrees within µServer.</p> <p>Leave empty if there is no scaling process.</p>
Max. raw value	
Min. scaled value	
Max. scaled value	

Variables	
Feature	Description
Alarm inhibited by	Identification of the user that has inhibited an alarm.
Inhibition start date	Alarm inhibition-related parameters are managed by µServer when a user wishes to ignore an alarm during a certain period.
Inhibition end date	
Manage alarms	<p>Define the variable as an alarm</p> <p>Note: In the “passive” communication mode, do not re-define the alarm here if it has been defined within the equipment.</p>
Alarm label	Text associated to the alarm
Alarm condition	<p>Activation condition of the alarm</p> <p>The value is represented by the character ‘v’.</p> <p>For instance, “ V >10” means that the alarm is activated as long as the value of the variable remains superior to 10.</p> <p>Available operators are the comparing operators of the Python language (>,<,>=,<=,==,...).</p> <p>You can combine several comparisons according to the rules of the Python language (example : "v>10 and v<20")</p> <p>If this field is empty, the alarm is activated as soon as the value differs from 0.</p>
Manage events	<p>Define the variable as an event</p> <p>NB: In the “passive” communication mode, do not re-define the alarm here if it has been defined within the equipment.</p>
Event label	Text associated to the event

Variables	
Feature	Description
Event condition	<p>Activation condition of the event</p> <p>The value is represented by the character 'v'.</p> <p>For example " V > 10" means that an event is triggered as soon as the variable overruns the threshold "10".</p> <p>If the field is empty, the event is triggered each time the value of the variable changes.</p>
Log values	<p>Define the variable as a historized value (curves...).</p> <p>NB: In the "passive" communication mode, do not re-define the historization here if it has been defined within the equipment.</p>
Average measure	<p>Log parameters</p> <p>By default, µServer measures a point in the log each time the value of the variables changes.</p> <p>If you want to lighten the database, it is highly recommended to defines log filters:</p> <ul style="list-style-type: none"> -The period between two points specifies a minimum duration between 2 measurements. -The hysteresis specifies a minimum difference of value between 2 measurements. -If you need more realistic curves, you can establish an average calculation between 2 measurements thanks to the parameter Average the curve.
Delay between two points (in seconds)	
Hysteresis	
SNMP address	<p>SNMP address of the variable.</p> <p>Parameter needed for the SNMP communication.</p>

Note:

In the case of a communication in "passive" mode, the features that have been defined within the device are labeled with a (EQT). The other parameters are defined on the server.

4.5. Visualization/forcing of variables

Variables are automatically updated by μServer once they have been associated to an "active" or "passive" mode communication.

4.5.1. Visualization

A variable value can be visualized:

- From the [Boolean variables](#)/[Numeric variables](#) options inside the concerned device ([Devices](#)).
- From the property "Value" inside the concerned variable ([Variables](#))
- From a synoptic thanks to objects included: texts, images, etc. (see section [9 Synoptic views](#))
- From the [Debug console](#), showing messages from a procedure, thanks to functions `get()` and `log()`. (See section [10.2.2 μServer functions](#))

The variable view can be personalized. For each variable, it is possible to determine the number of digits after the comma or even decide the unit to be displayed (see attributes "Format" and "Precision" in section [4.4 Administration of variables](#))

4.5.2. Forcing variables

To write on a variable, the variable must have been previously declared as "Writable" and the value to be written must be contained in the write value range (see section [2.4.4. Administration of variables](#))

A variable value can be written:

- By clicking on the variable from [Boolean variables](#)/[Numeric variables](#) or from the view of a variable (see section [3.3 Device operation](#)). The user must have the rights to write variables (see section [2.1 Administration of user profiles](#)). To force a text, the text must be enclosed in double quotes "".
- From a synoptic thanks to objects included: buttons, texts, zones of input entry, etc. (see section [9 Synoptic views](#))
- From a procedure, thanks to function `set()`. (See section [10.2.2 μServer functions](#))

4.6. Access rights to variables

A variable is not visible in the following cases:

- When this variable is associated to a device not accessible for the user (“Device filter” or “Zone filter” to be defined in [User accounts](#))
- If this variable has a category not accessible for the user (“Category filter” to be defined in [User profiles](#))

Variable writing depend on the rights defined inside the user profile: **Can set variable values**. Access to views of Boolean variables/Numeric variables depend on the rights: **Can view boolean variables** and **Can view numeric variables** also determined in the user profile.

Note:

If the user is administrator (right “Can administrate the database” selected), the rest of restrictions are ignored

For further information see sections [2.1 Administration of user profiles](#) and [2.2 Administration of user accounts](#)

5. Alarms

The administration of alarms is an embedded function of μServer.

It enables to:

- Display current alarms
- Display the alarms history
- Filter alarms per equipment, variable and variable category
- Acknowledge alarms
- Mask alarms
- Report alarms via sms/email
- Highlight all variables in state of alarm on a device interface
- Export all data in .csv file format

5.1. Definition of alarms

An alarm is always associated to a variable; however, the definition of alarms differs according to

the type of communication used.

5.1.1. Communication in “passive” mode:

Alarms are not defined within µServer, they are defined within the equipment.

The equipment sends all information related to the appearance or disappearance of alarms via the communication protocol.

5.1.2. Communication in “active” mode:

All alarms are created and administrated by µServer.

µServer supervises variables and detects the appearance and disappearance of alarms.

Variables only need to be defined as alarms and their status must be specified. Please read the section *4.4 Administration of variables* for further details.

5.2. Highlight of variables in state of alarm

Variables in state of alarm appear in red on [Boolean variables](#) and [Numeric variables](#) options of a device.

5.3. Alarms views

Alarm views are hierarchic according to 3 levels:

- 1) **General view of alarms.** It gathers all equipment alarms of the server which are accessible by the user. It can be accessed via the general menu [Alarms](#).
- 2) **View of equipment alarms.** It only gathers alarms associated to a certain device. It can be accessed from:
 - a device view, by clicking on the [Alarms](#) tab.
 - The general view of [Alarms](#), by clicking on a device name in the table of displayed alarms.
- 3) **View of a variable alarms.** It only gathers alarms associated to a certain variable. It can be accessed from:
 - The [Variables](#) view, by clicking on the [Alarms](#) tab.
 - The general view of [Alarms](#), by clicking on the alarm name in the table of displayed alarms.

→ The view of a device alarms, by clicking on the alarm name in the table of displayed alarms.

Whatever the visualization level of alarms, 2 types of views can be distinguished:

- Active alarms
- Alarms log

5.3.1. View of active alarms

It is the default type of view on an alarm display screen, is a first option inside [Alarms](#) → [Active alarms](#).

This type of view only displays active alarms of the present moment.

5.3.1.1. Change of the type of visualization

In case you desire to see to the alarms log, you only need to click on the sub-menu [Alarms log](#).

5.3.1.2. Visualization of alarms

All active alarms are listed in a table including a detailed description. A color caption enables to visualize the acknowledgement status of each equipment. The acknowledgement date does only appear if the alarm is effectively acknowledged. In this case, moving the cursor to the date will display the name of the user account which has implemented the acknowledgement and written its comment.

Table 8. Fields shown in the active alarms table.

Active alarms table	
Column	Description
Appearance	Alarm appearance date
Device	Name of device implied in the alarm This column is only available on the general view. A click on any device of this column allows to access the Active alarms of this device. If the device does not have a name, the device number is shown instead.

Active alarms table	
Column	Description
Alarm	<p>Alarm label</p> <p>This column is only available on the general view. A click on any device of this column allows to access the Active alarms of this device.</p> <p>If the variable does not have a label, its comment or its address are shown instead.</p>
Acknowledge	<p>Acknowledge information of the alarm.</p> <p>If the alarm has been acknowledged, the acknowledge date is shown.</p> <p>When passing over the cursor of this date, the name of the user account that the user that has acknowledge the alarm, as well as the acknowledge alarm, are shown.</p>
Sel.	<p>Alarm selection button.</p> <p>For select or unselect the alarm, just check or uncheck the selection box. This allows to do some actions as acknowledging alarms.</p> <p>A click on the Sel. title of the column, allows to select or unselect all.</p>

The [Active alarms](#) section offers the different actions and options explained next.

5.3.1.3. Display filter by acknowledgement and active statuses

A set of ticking boxes enables to filter the display according to the acknowledgement and active statuses of the alarms.

5.3.1.4. Display filter by category of variables

A drop-down list enables to filter categories of variables to display.

This list only appears if categories of variables have been defined and if variables of these categories are displayed (see section [4.4 Administration of variables](#)).

5.3.1.5. Acknowledgement of alarms

See section [5.4 Alarm acknowledgement](#).

5.3.1.6. Inhibition of alarms

This action is only available in the Variables menu, inside the inhibition of alarms related property (see section [5.5 Alarm inhibition](#)).

5.3.1.7. Csv Export

Click on the [Export](#) button to download the alarms table. The file is a table of the displayed data, in .csv format.

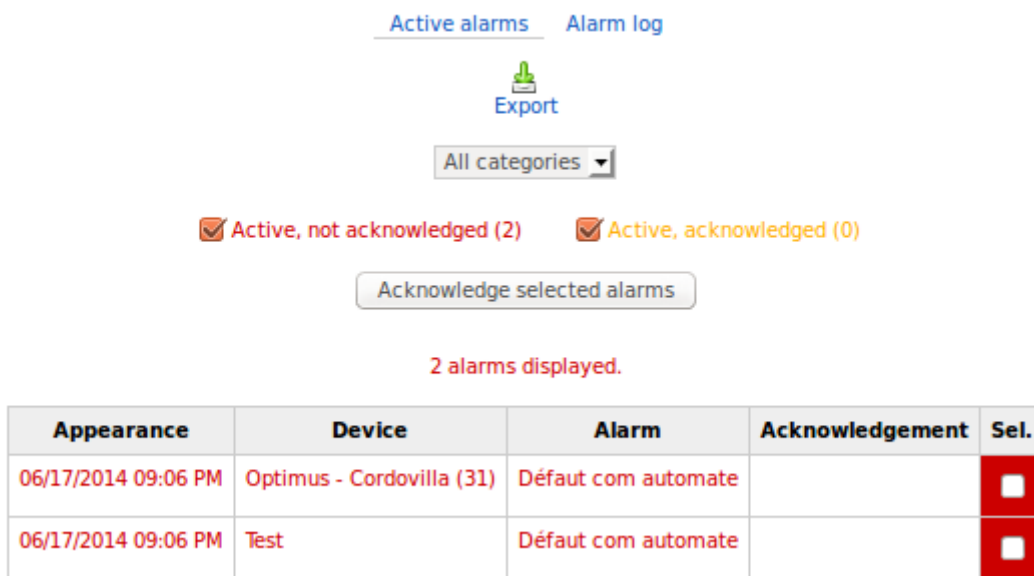


Figure 4. Active alarms submenu options.

5.3.2. View of alarm log

The alarm log can be accessed through [Alarms](#) → [Alarms log](#).

This type of view displays not only Active alarms of the present moment, but also inactive alarms.

5.3.2.1. Change of the type of visualization

In case you desire to see to the active alarms, you only need to click on the sub-menu [Active alarms](#).

5.3.2.2. Visualization of alarm log

All alarms are listed in a table including a detailed description, similarly as it was done in the [Active alarms](#) section. A color caption enables to visualize the acknowledgement status of each

equipment. The acknowledgement date does only appear if the alarm is effectively acknowledged. In this case, moving the cursor to the date will display the name of the user account which has implemented the acknowledgement and written its comment.

Table 9. Fields shown in the alarm log table.

Alarm log table	
Column	Description
Appearance	Alarm appearance date
Device	<p>Name of device implied in the alarm</p> <p>This column is only available on the general view. A click on any device of this column allows to access all the Alarm log of this device.</p> <p>If the device does not have a name, the device number is shown instead.</p>
Alarm	<p>Alarm label</p> <p>This column is only available on the general view. A click on any device of this column allows to access the Alarm log of this device.</p> <p>If the variable does not have a label, its comment or its address are shown instead.</p>
Disappearance	<p>Disappearance date of the alarm</p> <p>If the alarm is still active, this column is empty.</p>
Acknowledge	<p>Acknowledge information of the alarm.</p> <p>If the alarm has been acknowledged, the acknowledge date is shown.</p> <p>When passing over the cursor of this date, the name of the user account that the user that has acknowledge the alarm, as well as the acknowledge alarm, are shown.</p>

Alarm log table	
Column	Description
Sel.	<p>Alarm selection button.</p> <p>For select or unselect the alarm, just check or uncheck the selection box. This allows to do some actions as acknowledging alarms.</p> <p>A click on the Sel. title of the column, allows to select or unselect all.</p>

The [Alarms log](#) section offers the different actions and options explained next.

5.3.2.3. Display filter by acknowledgement and active statuses

A set of ticking boxes enables to filter the display according to the acknowledgement and statuses of the alarms.

5.3.2.4. Display filter by category of variables

A drop-down list enables to filter categories of variables to display.

This list only appears if categories of variables have been defined and if variables of these categories are displayed (see section [4.4 Administration of variables](#)).

5.3.2.5. Acknowledgement of alarms

See section [5.4 Alarm acknowledgement](#).

5.3.2.6. Inhibition of alarms

This action is only available in the Variables menu, inside the inhibition of alarms related property (see section [5.5 Alarm inhibition](#))

5.3.2.7. Csv Export


Click on the [Export](#) button to download the alarms table. The file is a table of the displayed data, in .csv format.

5.3.2.8. Definition of the period to be displayed

The alarm log shows alarms with appearance or disappearance occurred between the present date and a starting date, stated by default in 7 days.

The default period can be modified thanks to the period form.

[Active alarms](#) [Alarm log](#)


 Export

All categories ▾

Period: Days

Active, not acknowledged (2)
 Inactive, not acknowledged (16)

Active, acknowledged (0)
 Inactive, acknowledged (0)

2 alarms displayed.

Appearance	Device	Alarm	Disappearance	Acknowledgement	Sel.
06/17/2014 09:06 PM	Optimus - Cordovilla (31)	Défaut com automate			☐
06/17/2014 09:06 PM	Test	Défaut com automate			☐


[← Previous](#)  Print

Figure 5. Alarm log submenu options.

5.4. Alarm acknowledgement

The alarm acknowledgement allows to report that a certain alarm has been effectively taken into account by a user.

The acknowledgement is done from either of the two alarm views ([Active alarms](#) and [Alarm log](#)), by selecting one or several alarms, and then by clicking on [Acknowledge selected alarms](#). During an acknowledgement, µServer prompts the user to type a comment.

Once the alarm is acknowledged, it appears in orange in the alarm views. The acknowledgement comment, along with the user account name from which the user acknowledged the alarm, are visible when passing over the cursor on the acknowledgement date, inside the alarms table.

5.5. Alarm inhibition

Inhibiting an alarm allows to ignore it temporarily.

This is particularly interesting when you execute maintenance operations on an installation, for instance. In this case, the operator knows that such operations may trigger false alarms, which he can ignore during the concerned period.

An alarm inhibition results in:

- The alarm inhibition on all alarms views (both active and log)
- The deactivation of all Email/sms reports associated to that alarm.

The inhibition/dis-inhibition of an alarm is done from the visualization of a variable alarms. If the alarm is not masked, a button on the upper side of the screen enables to do it.

μServer automatically proposes to inhibit the alarm for a whole week, but this duration is configurable.

Once the alarm is inhibited, μServer reports it and features a button [Disinhibe](#).

Inhibited alarms are masked on alarms views, but a message appears to report that some of the alarms are inhibited. A click on that link enables to open the list of inhibited alarms. This view displays a detailed description of these inhibited alarms and also features the possibility to disinhibit the alarms.

The report of inhibited alarms on an Alarm view, along with the access to the list of inhibited alarms, depend on the level of visualization. In other words, if you visualize the alarms of a certain device, only the inhibited alarms related to that device will be reported and visible on that list.

5.6. Access rights to alarms

An alarm is not visible in the following cases:

- When this alarm is associated to a device not accessible for the user (**Device filter** or **Zone filter** to be defined in [User accounts](#))
- If this variable has a category not accessible for the user (**Category filter** to be defined in [User profiles](#))

Visualization and access to alarm log depend on the rights defined inside the user profile. For visualization: **Can view alarms**, and for access to the alarm log: **Can view logs** and **Log accessibility period (in days)**.

Similarly, alarm acknowledge and inhibition depend on the rights defined inside the user profile: **Can acknowledge alarms** and **Can inhibit alarms**.

For further information see sections [2.1 Administration of user profiles](#) and [2.2 Administration of user accounts](#).

6. Events

The management of events is an embedded function of µServer. It allows the following functions:

- Monitoring of the application behavior and users' actions
- Display of events log
- Event filtering by device, by variable and category of variable
- Data export in .csv file format

6.1. Definition of events

An event is always associated to a variable; however, the definition of events differs according to the type of communication used:

6.1.1. Alarm events

All appearance and disappearance of alarms is automatically considered as an event and added to the log.

6.1.2. Forcing-related events

Every forcing action carried out by a user **from a standard view of µServer** is automatically considered as an event and is added to the event log.

It is possible as well to log some user actions carried out **through synoptics** by activating the property **Log event on action** inside the object properties of the synoptic (for further information see section *9.3.5 Synoptic Objects*). Only objects that can be activated, such as buttons, lists, input zones, etc. can log an event. Furthermore, this type of log only works with actions that modify the value of the main variable associated to the object (for example, the call of a page cannot be logged).

In both cases, the user name who cause the forcing action is recorded and appears inside the event log.

6.1.3. Variable value change event

This type of event allows to generate a message on the log according to a variable value.

Every time a variable changes its value, if this variable property **Manage events** has been activated, µServer verifies whether its event conditions are fulfilled from a state where they were not. If it's the case, µServer generates a event message in the event log.

Manage events property, **Event condition** and **Event label** properties are defined inside variable properties (see section [4.4 Administration of variables](#))

Example 6. Managing events.

Let's imagine that we have a facility, where we would like to monitor the emergency stops and the overheating of an engine.

Variable %M0.1 represents the emergency stop (0: emergency stop off, 1: emergency stop on)

Variable %MF0.1 represents the engine temperature in °C.

Properties for %M0.1:

- Manage events: Yes
- Event label: Emergency stop
- Event condition: v!=0

This way, every time %M0.1. switches from 0 to 1, an event is added to the log with the message "Emergency stop".

Properties for %MF0.1:

- Manage events: Yes
- Event label: Engine overheating
- Event condition: v>60

This way, every time %MF0.1. changes and reaches a value higher than 60, an event is added to the log with the message "Engine overheating".

Note:

- In the case of "passive" mode communication, events are not defined by μServer, but inside the device. In this case, it would be the device that would send event activation warning via communication protocol.
- To define several event labels on a variable, it is necessary to use virtual variables (see section [4.2.6.2 PLC virtual variables](#)) or create several different variables.

6.2. Events log

Events log follow the following hierarchy according to three levels:

1) *General event log*

It gathers the events of all server equipment accessible by the user.

You can access it via the general menu [Events](#).

2) *Event log of a device*

It gathers all events associated to a given device. You can access it from two different ways:

- ➔ From the visualization of a device, by clicking on the tab [Events](#).
- ➔ From the general event log accessed through [Events](#) on the main menu, by clicking on the name of a device in the table of displayed events.

3) *Event log of a variable*

It only gathers events associated to a certain variable.

You can access it from three different ways:

- ➔ From the visualization of a variable, by clicking on the tab [Events](#).
- ➔ From the general event log accessed through [Events](#), by clicking on the name of a device in the table of displayed events.
- ➔ From the event visualization of a device, by clicking on the name of the variable in the table of displayed events.

The event log displays all events which have appeared/disappeared during a given period.

6.2.1. Definition of the period to be displayed

The log displays events which have occurred between the current date and a starting date. The default period covers one week but it can be modified by filling a period form.

6.2.2. Event visualization

Events are listed in a table which features a detailed description of each event.

Table 10. Fields shown on the event table.

Event table	
Column	Description
Appearance	Event appearance date.

Event table	
Column	Description
Device	<p>Name of the device implied in the event.</p> <p>This column is just visible when viewing the general event log. Clicking on this field allows to access to the event log associated to this device.</p> <p>If the device has not been named, its device name will be displayed instead.</p>
Variable	<p>Comment on the variable implied in the event.</p> <p>This column is just visible when viewing the general event log or a device event log. Clicking on this field allows to access to the event log associated to this variable.</p> <p>If the variable does not have a comment, its label or address will be displayed instead.</p>
Action	<p>Event description.</p> <p>This text depends on the event type.</p> <ul style="list-style-type: none"> • Alarm event: the text indicates whether it is an alarm appearance of disappearance. • Forcing-action event: the text indicates the forcing value and the name of the user that did it. • Value change event: the text indicates the new value of the variable.
Comment	<p>Event comment.</p> <p>This text depends on the event type:</p> <ul style="list-style-type: none"> • Alarm event: it is the alarm label. • Value change event: it is the event label. • Other: this column is empty.

6.2.3. Display filter by variable category

A drop-down list allows to filter variables to be displayed by categories.

This list is only available if categories of variables have been defined and if other categories are displayed (see section [4.4 Administration of variables](#)).

6.2.4. Csv Export

Click on the [Export](#) button to download the events table. The file is a table of the displayed data, in .csv format.

6.3. Access rights to events

An event is not visible in the following cases:

- When this alarm is associated to a device not accessible for the user (“Device filter” or “Zone filter” to be defined in [User accounts](#))
- If this variable has a category not accessible for the user (“Category filter” to be defined in [User profiles](#))

Access to event log depend on the rights defined inside the user profile: **Can view logs** and **Log accessibility period (in days)**.

For further information see sections [2.1 Administration of user profiles](#) and [2.2 Administration of user accounts](#).

7. Log of values

Log of values is an embedded function of μServer. It allows to follow the evolution of a variable in real time and to display that variable log in different ways:

- Plot charts / Histograms
- Bar chart
- Value tables
- csv format export

7.1. Definition of log

The log definition differs according to the type of communication used:

7.1.1. Communication in “passive” mode

Logs are not defined within μServer, but within the equipment.

The equipment sends logged values via the communication protocol.

7.1.2. Communication in “active” mode

Logs are defined and administrated by μServer.

μServer monitors variables and starts the log when the values of these variables change.

Log parameters of these variables are defined by the variable properties (see section [4.4 Administration of variables](#))

In order to lighten the database, and to consequently speed up the log display, the definition of log filters is recommended.

7.2. Plot charts/Histograms

Plot charts and histograms views are similar, but plot charts are assigned to numeric variables and histograms are assigned to boolean variables (see section [3.3 Device operation](#)). Therefore, histograms feature the particularity of having a fixed scale, between 0 and 1.

Broadly speaking, the word “curves” is generally used to allude to plot charts and histograms.

Curves enable to display simultaneously the evolution of one or several variables. A curve can be displayed in several ways:

- From the visualization of [Numeric variables](#) / [Boolean variables](#) of a device, select the variable(s) to visualize and click on [Plot chart](#) / [Histograms](#).
- From the visualization of a [Variables](#), from the main menu, click on the tab [Plot chart](#) / [Histograms](#).

Two visualization modes must be distinguished:

- “Log” mode
- “Real time” mode

7.2.1. Log mode

The log mode enables to visualize data from a defined starting date to a defined end date.

It is the default mode configured on all curves.

7.2.1.1. Definition of the period to be displayed.

The period form fixes the starting and ending dates to be considered for the variable display. Default values consider the last week period display.

In order to facilitate the period definition, these dates can be modified in different ways:

1. Manual Method:

Dates can be directly modified in the form. Then click on [Apply](#).

Note:

- An empty starting date box corresponds to a data display from the very beginning of log (most ancient point in the database). The button |< allows to obtain the same result.
- An empty end date box corresponds to a data display to the present moment. The button |> enables to obtain the same result.

2. Cursors method

This method consists in the modification of starting and end dates by selecting them graphically. It is particularly useful when you need to zoom in on a part of the curve. To apply this method follow the following steps:

1. Position the starting cursor by clicking on the curve
2. Update the starting date by clicking on the [Cursor](#) button located at the right side of the period form, at the starting date row..
3. Position the ending cursor by clicking on the curve.
4. Update the ending date by clicking on [Cursor](#) button located at the right side of the period form, at the end date row.
5. Update the curve by clicking on [Apply](#).

3. Zoom method

Use buttons [Zoom in](#) / [Zoom out](#) after clicking on the curve to position the reference point.

Note:

The zoom only modifies the display period, not the scale.

4. Slide method

Use buttons [Go forward](#) / [Go back](#) to position the period to be considered on the variable display.

Note:

Only variables with the property Log values set to “Yes” will be represented in curves. If curves related to the variable you may want to visualize are empty, please check that this property has been activated.

7.2.1.2. Scale definition

The scale on the Y-axis is configured via the form located just below the curve. The new scale is applied as soon as you click out of the entry field (elsewhere on the interface).


Variable	Value	Scale
 Timer Value	--	<input type="text"/> - <input type="text"/>

Figure 6. Form to set the Y-axis scale of a variable shown in a plot graph.

For an automatic scaling, leave the form empty. This can be done by leaving empty just the low value, the high value or both.

By default, the scale is pre-set with values defined in the properties of the variable (see section [4.4 Administration of variables](#)). Nonetheless, the last values typed in during the visualization are saved for an adapted pre-setting.

7.2.1.3. Value display

You only need to click on the curve to display the details of a point, date and value.

In order to also display the units associated to the value, you have to define the format in the properties of the variable (see section [4.4 Administration of variables](#)).

7.2.1.4. Display of several curves

When several curves are displayed on a same graph chart, a color caption enables to distinguish curves, scales and values related to a same point.

A different scale can be specified for each variable if desired.

7.2.1.5. Data export

For each variable, you can export data from the curve by clicking on the **export button** (a small green arrow, see Figure 6), which is located on the Y-axis scale form, just before the variable identification (comment or label).

The exported file will include the whole set of measurements available on the total defined

period, in .csv format.

7.2.2. Real time mode

In order to switch to the "real time" mode, click on the tab [Real time](#) located above the time period form.

The screenshot shows a user interface for selecting a time period. At the top, there are two tabs: 'Log' and 'Real time'. The 'Real time' tab is currently selected. Below the tabs, there are two rows of date and time pickers. The first row is labeled 'Start:' and contains a month dropdown (set to 'June'), a day dropdown (set to '17'), a year input field (set to '2014'), a time dropdown (set to '11'), a colon separator, another time dropdown (set to '22'), a colon separator, and a third time dropdown (set to '31'). To the right of these are two buttons: '<' and 'Cursor'. The second row is labeled 'End:' and contains a month dropdown (set to '--'), a day dropdown, a year input field, a time dropdown, a colon separator, another time dropdown, a colon separator, and a third time dropdown. To the right of these are two buttons: '>' and 'Cursor'. Below the 'End:' row is a single 'Apply' button.

Figure 7. Log and real time visualization modes and time period form for curve visualization.

The real time mode executes an automatic update of the visualization period, which gives a dynamic aspect to the display.

This mode is similar to the history mode, except for the definition of the displayed period.

7.2.2.1. Definition of the period to display

Unlike history mode, the period is not fixed but gets updated in order to display only the latest data.

By default, µServer displays the past week but a form allows to modify this period.

7.3. Bar chart

Bar charts are only used for the display of counters which increment over time.

They are particularly adapted for energy production/consumption assessments for instance.

You can display a bar chart in various ways:

- From the view of a device [Variables](#), select the variable to visualize on the check box and click on [Bar chart](#).
- From the view of a variable, click on the tab [Bar chart](#).

The bar chart displays data over a total period by gathering them in sub-periods represented as bars. For each bar, the bar chart displays the difference between the values of the counter at the end and at the beginning of the sub-period.

7.3.1. Definition of the period to be displayed

The user only defines the total period to display. Sub-periods, that is to say the “duration” of bars, are automatically calculated in a way that enables to obtain whole units (months, weeks, days, hours, tenths of minutes, minutes).

The period form fixes the starting and ending dates to be considered for the variable display. Default values consider the last week period display.

In order to facilitate the period definition, these dates can be modified in different ways:

1. Manual Method:

Dates can be directly modified in the form. Then click on [Apply](#).

Note:

- An empty starting date box corresponds to a data display from the very beginning of log (most ancient point in the database). The button |< allows to obtain the same result.
- An empty end date box corresponds to a data display to the present moment. The button |> enables to obtain the same result.

2. Cursors method

This method consists in the modification of starting and end dates by selecting them graphically. It is particularly useful when you need to zoom in on a part of the curve. To apply this method follow the following steps:

1. Position the starting cursor by clicking on the curve
2. Update the starting date by clicking on the [Cursor](#) button located at the right side of the period form, at the starting date row..
3. Position the ending cursor by clicking on the curve.
4. Update the ending date by clicking on [Cursor](#) button located at the right side of the period form, at the end date row.
5. Update the curve by clicking on [Apply](#).

3. Zoom method

Use buttons [Zoom in](#) / [Zoom out](#) after clicking on the curve to position the reference point. The zoom enables a “clever” browsing, with whole units periods (months, weeks, days, hours, tenths of minutes, minutes...)

Note:

The zoom only modifies the display period, not the scale.

4. Slide method

Use buttons [Go forward](#) / [Go back](#) to position the period to be considered on the variable display.

Note:

Only variables with the property Log values set to “Yes” will be represented in curves. If curves related to the variable you may want to visualize are empty, please check that this property has been activated.

7.3.2. Scale definition

The scale on the Y-axis is configured via the form located just below the curve. The new scale is applied as soon as you click out of the entry field (elsewhere on the interface).


Variable	Value	Scale
 Timer Value	--	<input type="text"/>

Figure 8. Form to set the Y-axis scale of a variable shown in a bar chart.

For an automatic scaling, leave the form empty. This can be done by leaving empty just the low value, the high value or both.

By default, the scale is pre-set with values defined in the properties of the variable (see section [4.4 Administration of variables](#)). Nonetheless, the last values typed in during the visualization are memorized for an adapted pre-setting.

7.3.3. Value display

You only need to click on the bar chart to display the details of a point, date and value.

In order to display also units associated to that value, you have to define the format in the properties of the variable (see section [4.4 Administration of variables](#)).

7.3.4. Display of several bar charts

When several variables are displayed on a same graph view, a color caption enables to distinguish curves, scales and values related to a same point.

A different scale can be specified for each variable if desired.

7.3.5. Data export

For each variable, you can export data from the bar chart by clicking on the **export button**, a small green arrow, see Figure 8), which is located on the Y-axis scale form, just before the variable identification (comment or label).

The exported file will include the whole set of measurements available on the total defined period, in .csv format.

7.4. Value tables

Value tables allow to visualize the evolution of one or several variables in the form of a table.

They also represent a simple and efficient way to export data in .csv file format, since they allow to define a fixed step between each variable point, in addition to the possibility to simultaneously export data proceeding from several variables.

You can display a [Value table](#) in different ways:

- From the view of a device [Variables](#), select which variables to visualize through the check box and click on [Value table](#).
- From the view of a variable, click on the tab [Value table](#).

The table displays data of a total period by gathering them in sub-periods represented as lines in the table. For each line, the table displays the average value of the variable on the whole sub-period.

7.4.1. Definition of periods

Two periods can be defined on the form located at the top of the page:

- The total period represented by a starting and an ending date.
- The period of a line, merely called "period".

Once recorded, these periods are defined in such a way that the whole logged data are displayed, with one data group per day.

In order to modify these periods, fill in the form and click on [Apply](#) to generate the table.

Note:

- An empty starting date box corresponds to a data display from the very beginning of logged operations (most ancient point in the database).
- An empty ending date box corresponds to a data display until the current day.

7.4.2. Value display

Each line of the table represents the average value of each variable on the whole sub-period.

In order to display units, you only need to define the format in the properties of the variable (see section [4.4 Administration of variables](#)).

7.4.3. Display of several variables

When several variables are displayed on a same table, it will simply create an additional column per variable, along with a color set for a better visibility.

7.4.4. Data export

This view features 2 types of data export:

- A global export of all displayed variables (button [Export](#) in the upper side of the view)
- An individual export for each one of the displayed variables (a small green arrow, which is located just before the variable identification)

The exported file will include all the lines as they are displayed in the table, in a .csv file format.

7.5. Log filters

Each new measure is registered on the database so it is important to filter the logging. An overcharged database has as consequence:

- Increase in the space used on the disk
- The variable access time to show curves and graphs can become slower
- Increase in the data amount and thus in the time, and possibly in the communication costs in case where µServer sends its data to a central server. In this case, the central server database is also heavier.

By default, µServer registers a new point any time a logged variable changes its value. This is not adapted to a variable that constantly changes and, in particular, if it is the case of a decimal variation, as the minimal variation will be considered as a new value, and thus a new log.

Log parameters **Hysteresis**, **Delay between two points** and **Average measure**, described on section [4.4 Administration of variables](#) allow to register just significant data.

A well configured filter allows to analyze the evolution of an application without lack of information, just by saving a minimum of logged data. It is usually not necessary to have a great precision in order to analyze the behavior of an application.

It is necessary to know well the measurement in order to adjust the hysteresis. The minimum amplitude below which the change is not significant for the operation with µServer, needs to be determined.

That is, it is suitable to adapt the filtering according to:

- The number of logged variables.
- It is advisable to prioritize those variables that require the most precise tracking, but also to filter those from which we just want to follow the trend.
- The hardware performance
- Embedded products such as picoFox are not suitable for a too smooth logging. Ask Sirea for a product adapted to your needs.
- The system architecture for the data uploading
- In the case of a data upload to a higher level server, it is necessary to take into account the speed and cost of communications, as well as the number of devices which communicate with the server. In this case, all the logged points of each device are transferred to the central server

Example 7.Filters

1) To monitor a voltage that oscillates around 230V, a hysteresis type filter is recommended (10V, for example). This way, every anomalous voltage variation is registered, masking at the same time little variations around 230V.

- **Average measure:** No
- **Delay between two points (in seconds):**
- **Hysteresis:** 10

2) To monitor a temperature that varies progressively during the day, a delay (10 minutes, for example) with an **average measure** can be used. This way, every 10 minutes a new point is registered with the average temperature value of the last 10 minutes.

This type of filter allows to obtain average values representative to reality, but masks sporadic temperature peaks of short duration (<1 minute).

An additional filtering can be done by adding an hysteresis of some °C. In this case, the average will only be registered if it has sensibly changed. This would have the effect of lighten the database thus getting a curve display much faster, at the expense of the temporary resolution.

- **Average measure:** Yes
- **Delay between two points (in seconds):** 600
- **Hysteresis:** 2

3) In order to monitor a electric consumption counter which is increased with the time, a delay will be used (24 hours, for example) but no an average. A point is thus obtained every 24 hours, which is the exact value of the counter at this instant. This curve will have then a ladder aspect with a step every 24 hours. This way, consumption over a given month or year can be calculated by subtraction of end / starting period values (with a precision of 24 hours).

- **Average measure:** No
- **Delay between two points (in seconds):** 86400
- **Hysteresis:**

8. Alert reports

µServer can automatically generate alert reports via email and/or sms for any alarm appearance/disappearance. Recipients of such reports can be filtered by equipment, areas and/or categories of variables.

Concerning the sending of emails, µServer must have access to a SMTP server.

Concerning the sending of sms, µServer can use two methods:

- The operator opens an account on one of the sms-sending internet platforms (among the µServer-compatible ones)
- The machine is equipped with a GSM modem (among the µServer-compatible ones) connected via a serial port. The modem has to integrate a SIM card with a permanent service allowing to send sms.

Check it with Sirea for the parameter setting of the communication media.

8.1. Administration of reports

To create a new report:

[Reports](#) → [Add](#)

To visualize a new report:

[Reports](#) → [Search](#) → Click on the report

To modify one or several reports:

[Reports](#) → [Search](#) → Click on the report → [Edit](#)

Or:

[Reports](#) → [Search](#) → Select the report(s) to modify → [Edit](#)

To copy a report:

[Reports](#) → [Search](#) → Click on the report → [Copy](#)

To delete one or several reports:

[Reports](#) → [Search](#) → Click on the report → [Delete](#)

Or :

[Reports](#) → [Search](#) → Select the report(s) to delete → [Delete](#)

Note:

These actions require administrator rights.

Table 11. Available description fields in the report section.

Report	
Feature	Description
Comment	Description of the report.
Type	Type of report. Potential values : <i>sms, email</i>

Report	
Feature	Description
Recipient	<p>Address of the recipient.</p> <p>For a "sms" type, it concerns the phone number.</p> <p>For a "email" type, it concerns the email address.</p> <p>In both cases, you can specify the name of one or several user accounts (list of suggestions). This way, phone numbers and email addresses will belong to the user (see section 2.2 Administration of user accounts).</p> <p>Note: You can specify several recipients by separating them with ','.</p> <p>You cannot use the same process to mix phone numbers and email addresses, since the type of report is unique.</p>
Subject	<p>Subject of the email</p> <p>For a "sms"-type report, leave that field empty.</p> <p>The subject may include special fields which give a more dynamic aspect to the text (see section 8.2 Special fields).</p>
Message	<p>Content of the message.</p> <p>The subject may include special fields which give a more dynamic aspect to the text (see section 8.2 Special fields).</p>
Send on alarm appearance	<p>Activate for the reception of an alarm-appearance report</p>
Sending on alarm disappearance	<p>Activate for the reception of an alarm-disappearance report</p>
Zone filter	<p>Deactivate the report, in certain cases.</p>
Device filter	<p>Leave the field empty in order to deactivate the filter.</p>
Category filter	<p>For each filter, you can:</p>

Report	
Feature	Description
Variables type filters	<ul style="list-style-type: none"> - specify a unique value - specify a list of values separated by ',' <p>For the equipment filter, you can also:</p> <p><u>Area filter</u>: Only alarms related to variables that belong to the equipment of the filtered zone(s) will be sent.</p> <p><u>Equipment filter</u>: Only alarms related to variables that belong to filtered equipment will be sent.</p> <p><u>Category filter</u> : Only alarms related to variables that belong to filtered categories will be sent.</p> <p><u>Type of variable filter</u>: Only alarms related to variables that belong to filtered types will be sent.</p>

8.2. Special fields

Special fields are used to make dynamic the reports texts by associating them to the context of the alarm.

Table 12. Special fields available.

Special fields	
Syntax	Description
<ALM-STATE>	Event type "A" for appearance "D"for disappearance
<ALM-DATE>	Event date
<ALM-TIME>	Event time
<ALM>	Alarm label
<DEV-ID>	Index of the concerned device
<DEV-TYPE>	Type of concerned device
<DEV-NAME>	Name of the concerned device
<DEV-DEP>	Department (geogr.) of the concerned device
<DEV-CITY>	City of the concerned device

Example 8. Using special fields on Alarm message.

-Send on alarm appearance = Yes

-Sending on alarm disappearance= No

-Message= An alarm appeared on <ALM-DATE> at <ALM-TIME>

-Filters: None

Each time µServer detects an alarm appearance, it sends a message by replacing all special fields with current values. If the alarm appears on January, 1st 2013 at 2 p.m., the message sent will be *"An alarm appeared on the 01/01/2013 at 2 p.m."*

9. Synoptic views

9.1. Presentation of synoptic views

Synoptics views are graphic views, animated in function of the status of variables within μServer.

Based on the principle of a classic SCADA, synoptic views enable to create personalized views, representing for example an automation installation.

Synoptic views are “designed” by an administrator, but can be visualized by other users who do not have administrator rights (see sections [9.4.1 Associating a synoptic to a device](#) and [9.4.2 Associate a synoptic to a user account](#), and [9.4.3 Default synoptic](#)).

A synoptic can comprise one or several pages. Each page includes graphic objects. Properties of such objects can be fix or dynamic thanks to the association with μServer variables.

Due to different types of objects, synoptic views enable to:

- Display texts and values
- Display images
- Display geometrical forms
- Display gauges, dials
- Enter values
- Send orders to equipment
- Navigation between synoptic pages, towards other μServer views or other websites

9.2. Administration of synoptic views

To create a new synoptic view:

[Views](#) → [Add](#)

To access a synoptic view already existing:

[Views](#) → [Search](#) → Click on the synoptic

To modify one or several synoptic views:

[Views](#) → [Search](#) → Click on the synoptic → [Edit](#)

Or:

[Views](#) → [Search](#) → Select which synoptic views to modify → [Edit](#)

To copy a synoptic:

[Views](#) → [Search](#) → Click on the synoptic → [Copy](#)

To delete one or several synoptic views:

[Views](#) → [Search](#) → Click on the synoptic → [Delete](#)

Or:

[Views](#) → [Search](#) → Select which synoptic views to delete → [Delete](#)

To export a synoptic view into .vu format:

[Views](#) → [Search](#) → Click on the synoptic → [Export](#)

To import a synoptic view:

[Views](#) → [Search](#)

Note:

All these actions require administrator rights.

Synoptic view	
Feature	Description
Name	<p>Name of the synoptic view.</p> <p>This is the name used to associate synoptic views (see sections 9.4.1 Associating a synoptic to a device and 9.4.2 Associate a synoptic to a user account)</p> <p>If the synoptic has no name, it is considered as a default synoptic. (See chapter 9.4.3 Default synoptic)</p> <p>Many synoptics can have the same name to manage different screen sizes. (See chapter 9.4.4. Synoptic visualization)</p>
Comment	<p>Type of report.</p> <p>Possible values: <i>sms</i>, <i>email</i></p>

9.3. Edition of synoptic views

There are two ways of editing synoptic views:

- Via the online editor
- Via the software µIHM for edition of synoptic views

9.3.1. Edition via µIHM

Depending on whether you create a new synoptic or you modify an existing synoptic:

- open the µIHM software and create a new project
- or
- export the synoptic and open it with µIHM

Once the synoptic is modified on µIHM, import it from µServer.

9.3.2. Edition via the online editor

[Views](#) → [Search](#) → Click on the synoptic → [Modify](#)

Note:

When different synoptics have the same name (see chapter 9.2 Administration of synoptic views) a pull-down menu is shown on the title bar next to the synoptic's name. It allows to change the synoptic during edition, specially in the event of having the synoptic in different sizes.

9.3.2.1. Edition tool bar

Once entering the edition of a synoptic view, the following toolbar will appear:



Figure 9. µServer edition tool bar.

The following table describes the online editor tools, as they are shown in Figure 9 from left to right. Be aware that in µIHM software these menus and tools may slightly differ.

Edition Toolbar	
Tool	Description
<i>Edit global variables</i>	Execute the replacement of global variables. <i>(See section 9.3.9 Global variables of a synoptic)</i>
<i>Edit background</i>	Edit the background characteristics. <i>(See section 9.3.4 Background of synoptic views)</i>
<i>Show previous page</i>	Change the page to be edited while modifying/creating a synoptic.
<i>List of pages</i>	
<i>Show next page</i>	
<i>Create new page</i>	Add a new page after the already existing ones. When you create a new page, the editor automatically works on that page.
<i>Move page</i>	Move the current page to the desired position inside the synoptic. When you move a page among the others, their numbers are automatically re-organized. Therefore, you will need to update the links between the pages regularly.
<i>Delete page</i>	Delete the current page. When you delete a page among the others, their numbers are automatically re-organized. Therefore, you will need to update the links between the pages regularly.
<i>Display layer</i>	Display the layer (or all layers) that you want to see inside your synoptic.
<i>Add object</i>	Add an object in the center of the view. You can either double-click it or select it and click on <i>Edit selected objects</i> in order to modify its properties.

Edition Toolbar	
Tool	Description
<i>Duplicate selected objects</i>	Copy an object or selection of objects. A copy of each object is created and put right above the original object. You then have to shift the new object so that you can distinguish it from the original.
<i>Edit selected objects</i>	Edit an object or selection of objects. When clicking on this option, the object properties to be edited will be then shown on a new window.
<i>Move selected objects to foreground</i>	Define the objects position within the layer. Objects can overlap each other according to a certain hierarchy within the view plane. µServer manages the transparency of objects and images (if the image format allows it).
<i>Move selected object to background</i>	
<i>Move selected objects to another page</i>	Move the selected objects towards another page of the synoptic. When clicking you can choose the destination page.
<i>Delete selected objects</i>	Delete selected objects.

9.3.3. Generalities regarding the online editor

9.3.3.1. Selection of several objects

The selection of several objects is done via the “Ctrl” key.

9.3.3.2. Positioning of objects

The shifting of objects on a page is done by dragging-and-dropping.

Object can also be positioned by modifying their coordinates (see section [9.3.5 Synoptic Objects](#)).

The size of objects is expressed in pixels.

9.3.3.3. Object resizing

In order to resize an object, use the resizing points which appear on the contour of the object while being selected.



Figure 10. Object showing the resizing points on the contour.

Objects can as well be resized by modifying their property size (see section [9.3.5 Synoptic Objects](#)).

The size of the objects is expressed in pixels.

9.3.3.4. Edition of objects

For editing objects:

- Double-click on an object
- Selection of one or several objects and click on *Edit selected objects*.

The edition of an object clicking opens a new tab on the browser. If you modify several objects, it will open as many tabs as modified objects.

Colors

Colors are represented by their HTML notation, for example “#FFFFFF” for white.

When the color is not specified within the characteristics of an object, it means “transparent”.

Fonts

Fonts are based on the CSS notation.

In order to simplify it, you can define the following characteristics:

- Style
Examples: *"italic"*, **"bold"**
- Size
Examples: "12", "14px"

Properties are separated by a space, all properties are not specified, and some can be specified

several times.

Examples: `"italic 12 sans-serif", "italic bold 12px arial", "10"`

Images

µServer supports most of the standard images formats. It is better to use compressed formats and small images, especially for animations.

The transparency is managed, as long as the selected format supports it (example: png).

The background is to be defined in the first place, since it defines the size of the synoptic.

The background is common to all pages. As a result, it is not possible to specify a different size for each page.

- Family

Examples: `"serif", "sans-serif", "monospace" or "arial", "helvetica"`

9.3.4. Background of synoptic views

First, the background size must be defined, as it defines the synoptic size.

The background is common to all pages in the synoptic. Than means it is not possible to specify a different size for each page.

For an automatic adaptation of the synoptic size to the screen of the connected user, it is possible to create different synoptics with the same name but different size (see chapter 9.2 Administration of synoptic views)

In order to edit the properties of the background, use the button *Edit background* or double-click on the background of the synoptic.

Table 13. Properties available on the synoptic background.

Synoptic background	
Property	Description
Width	Size of the synoptic (<i>STAT</i>).
Height	See section <u>9.3.3 Generalities regarding the online editor</u> .

Synoptic background	
Property	Description
Image index	Background image of the synoptic (DYN). The image index is the image number within the list of images of the synoptic. The list of images is located right under the properties table. You can edit it via the following buttons: Add , Delete , Browse .
Image width	Size of the background image (DYN).
Image height	The background image is centered on the synoptic, but it does not necessarily take up all the room available in the synoptic. In order to automatically adapt the size of the image to the size of the synoptic, type 0 or leave this field empty.
Background color	Background color (DYN). See section 9.3.3 Generalities regarding the online editor

Legend:

(**STAT**): This property is static; the variable value cannot influence its animation.

(**DYN**): This property is dynamic; the variable value can influence its animation.

Note:
This previous legend will be applicable to all tables on synoptic objects shown next.

9.3.5. Synoptic Objects

Table 14. Properties available on Image objects.

Object: Image	
Display of an image in a list of images.	
Property	Description
X axis	Position of the object (STAT).

Object: Image	
Display of an image in a list of images.	
Property	Description
Y axis	See section <i>9.3.3 Generalities regarding the online editor</i>
Width	Size of the object (STAT).
Height	See section <i>9.3.3 Generalities regarding the online editor</i>
Main variable	Main variable for the animation of the object (STAT). See section <i>9.3.7 Dynamic properties of objects</i> .
Image index	Image to be displayed (DYN). The image index is the image number within the list of images of the synoptic. The list of images is located right under the properties table. You can edit it via the following buttons: Add , Delete , Browse .

Table 15. Properties available on Push-button objects.

Object: Push-button	
Button which returns to the OFF position when you release the click. For writing on a variable, change of page or other special control.	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section <i>9.3.3 Generalities regarding the online editor</i>
Width	Size of the object (STAT).
Height	See section <i>9.3.3 Generalities regarding the online editor</i>

Object: Push-button	
Button which returns to the OFF position when you release the click. For writing on a variable, change of page or other special control.	
Property	Description
Main variable	<p>Main variable associated to the button (STAT).</p> <p>It is the variable to which the ON and OFF values are written at each change of the button status.</p> <p>It is also used for the animation of the dynamic properties of the object. See section 9.3.7 Dynamic properties of objects.</p>
Log event of an action	Associate an event to the activation of this object.
Index of "OFF" image	Button associated images (DYN).
Index of "ON" image	<p>The button can have two different images according to its status: "released" or "pressed". It enables the user to visualize his action.</p> <p>The image index is the image number within the list of images of the synoptic. The list of images is located right under the properties table. You can edit it via the following buttons: Add, Delete, Browse.</p>
"OFF" value	Button controls (DYN).
"ON" value	<p>Commands are sent when the button status changes.</p> <p>When the button status changes from "released" to "pressed", the value ON is written in the main variable.</p> <p>When the button status changes from "pressed" to "released", the value OFF is written in the main variable.</p> <p>You can also define specific controls on a button (see section 9.3.8 Special controls).</p>

Table 16. Properties available on Toggle-button objects.

Object : Toggle button	
Button which remains in a position after releasing the click. Writing in a variable, change of page or other special control.	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor
Width	Size of the object (STAT).
Height	See section 9.3.3 Generalities regarding the online editor
Main variable	Main variable associated to the button (STAT). It is the variable to which the ON and OFF values are written at each change of the button status. It is also used for the animation of the dynamic properties of the object. See section 9.3.7 Dynamic properties of objects
Log event of an action	Associate an event to the activation of this object.
Index of the image OFF	Button associated images (DYN).
Index of the image ON	The button can have two different images according to its status: "ON" or "OFF". It enables the user to visualize his action. The image index is the image number within the list of images of the synoptic. The list of images is located right under the properties table. You can edit it via the following buttons: Add , Delete , Browse .
Value OFF	Button controls (DYN).
Value ON	The commands are sent when the button status changes. When the button status goes from "released" to "pressed" , the value ON is written in the main variable. When the button status goes from "pressed" to "released", the value

Object : Toggle button	
Button which remains in a position after releasing the click. Writing in a variable, change of page or other special control.	
Property	Description
	OFF is written in the main variable. You can also define specific controls on a button (see section 9.3.8 Special controls).

Table 17. Properties available on Text objects.

Object: Text	
Display of a text or a value	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor
Width	Size of the object (STAT).
Height	See section 9.3.3 Generalities regarding the online editor
Main variable	Main variable for the animation of the object (STAT). See section 9.3.7 Dynamic properties of objects .
Text color	Color selected for the text (DYN). See section 9.3.3 Generalities regarding the online editor .
Frame width	Width selected for the frame (DYN). See section 9.3.3 Generalities regarding the online editor .
Frame color	Color selected for the frame (DYN). See section 9.3.3 Generalities regarding the online editor .

Object: Text	
Display of a text or a value	
Property	Description
Background color	Color selected for the background (DYN). See section 9.3.3 Generalities regarding the online editor .
Horizontal alignment	Horizontal alignment of the text within the object. (STAT).
Vertical alignment	Vertical alignment of the text within the object. (STAT).
Font	Font of the text (DYN). See section 9.3.3 Generalities regarding the online editor .
Padding	The padding properties specify the width of the padding area of a box. It is a shorthand property for setting 'padding-top', 'padding-right', 'padding-bottom', and 'padding-left' at the same place in the style sheet. If there is only one component value, it applies to all sides. (STAT).
Text	Text to be displayed (DYN). The text can either be static or dynamic, to display the value of a variable for example. See section 9.3.7 Dynamic properties of objects .

Table 18. Properties available on Select box objects.

Object: Select box	
Selection list which enables to write a value within a variable.	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor .
Width	Size of the object (STAT).

Object: Select box	
Selection list which enables to write a value within a variable.	
Property	Description
Height	See section 9.3.3 Generalities regarding the online editor.
Main variable	Main variable associated to the select box (STAT). It is the variable to which all selected values are written. It is also used for the animation of the dynamic properties of the object. See section 9.3.7 Dynamic properties of objects.
Log event of an action	Associate an event to the activation of this object.
Text color	Color selected for the text (DYN). See section 9.3.3 Generalities regarding the online editor.
Frame width	Thickness selected for the frame (DYN). See section 9.3.3 Generalities regarding the online editor.
Frame color	Color selected for the frame (DYN). See section 9.3.3 Generalities regarding the online editor.
Background color	Color selected for the background (DYN). See section 9.3.3 Generalities regarding the online editor.
Horizontal alignment	Horizontal alignment of the text within the object. (STAT).
Vertical alignment	Vertical alignment of the text within the object. (STAT).
Font	Font of the text (DYN). See section 9.3.3 Generalities regarding the online editor.

Object: Select box	
Selection list which enables to write a value within a variable.	
Property	Description
Padding	The padding properties specify the width of the padding area of a box. It is a shorthand property for setting 'padding-top', 'padding-right', 'padding-bottom', and 'padding-left' at the same place in the style sheet. If there is only one component value, it applies to all sides. (<i>STAT</i>).
Text	List of options (<i>DYN</i>). The list of options is a set of associations between the variable value and the text to be displayed. The syntax is the following: <i>"value0:text0;value1:text1;value2:text2"</i> <i>Example: "0:No;1:Yes"</i>

Table 19. Properties available on Text entry objects.

Object: Text entry	
Entry field which enables to write a value on a variable. The entry field only comprises one line. Press the ENTER button to validate the entry.	
Property	Description
X axis	Position of the object (<i>STAT</i>).
Y axis	See section 9.3.3 Generalities regarding the online editor.
Width	Size of the object (<i>STAT</i>).
Height	See section 9.3.3 Generalities regarding the online editor.

Object: Text entry	
<p>Entry field which enables to write a value on a variable. The entry field only comprises one line. Press the ENTER button to validate the entry.</p>	
Property	Description
Main variable	<p>Main variable associated to the entry field (STAT).</p> <p>It is the variable on which the entered value is written.</p> <p>The value is written as soon as the user presses ENTER or if the field loses its focus (press TAB, click out of the field).</p> <p>It is also used for the animation of the dynamic properties of the object. See section <i>9.3.7 Dynamic properties of objects</i>.</p>
Log event of an action	Associate an event to the activation of this object.
Text color	<p>Color selected for the text (DYN).</p> <p>See section <i>9.3.3 Generalities regarding the online editor</i>.</p>
Frame width	<p>Thickness selected for the frame (DYN).</p> <p>See section <i>9.3.3 Generalities regarding the online editor</i>.</p>
Frame color	<p>Color selected for the frame (DYN).</p> <p>See section <i>9.3.3 Generalities regarding the online editor</i>.</p>
Background color	<p>Color selected for the background (DYN).</p> <p>See section <i>9.3.3 Generalities regarding the online editor</i>.</p>
Horizontal alignment	Horizontal alignment of the text within the object. (STAT).
Vertical alignment	Vertical alignment of the text within the object. (STAT).
Font	<p>Font selected for the text (DYN).</p> <p>See section <i>9.3.3 Generalities regarding the online editor</i>.</p>

<p>Object: Text entry</p> <p>Entry field which enables to write a value on a variable. The entry field only comprises one line. Press the ENTER button to validate the entry.</p>	
Property	Description
Padding	The padding properties specify the width of the padding area of a box. It is a shorthand property for setting 'padding-top', 'padding-right', 'padding-bottom', and 'padding-left' at the same place in the style sheet. If there is only one component value, it applies to all sides. (STAT).

Table 20. Properties available on Text area objects.

<p>Object : Text area</p> <p>Entry field which allows to write a text within a variable. The multi-lines entry field can comprise several lines. Pressing ENTER will move the cursor to the next line but will not validate the entry.</p>	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section <i>9.3.3 Generalities regarding the online editor.</i>
Width	Size of the object (STAT).
Height	See section <i>9.3.3 Generalities regarding the online editor.</i>
Main variable	<p>Main variable associated to the entry field (STAT).</p> <p>It is the variable on which the entered text is written .</p> <p>The text is written as soon as the field loses its focus (press TAB, click out of the field).</p> <p>It is also used for the animation of the dynamic properties of the object. See section <i>9.3.7 Dynamic properties of objects.</i></p>

Object : Text area	
Entry field which allows to write a text within a variable. The multi-lines entry field can comprise several lines. Pressing ENTER will move the cursor to the next line but will not validate the entry.	
Property	Description
Log event of an action	Associate an event to the activation of this object.
Text color	Color selected for the text (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Frame width	Width selected for the frame (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Frame color	Color selected for the frame (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Background color	Color selected for the background (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Horizontal alignment	Horizontal alignment of the text within the object. (STAT).
Vertical alignment	Vertical alignment of the text within the object. (STAT).
Font	Font selected for the text (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Padding	The padding properties specify the width of the padding area of a box. It is a shorthand property for setting 'padding-top', 'padding-right', 'padding-bottom', and 'padding-left' at the same place in the style sheet. If there is only one component value, it applies to all sides. (STAT).

Table 21. Properties available on Gauge objects.

Object : Gauge	
Presentation change of a variable compared to its display scale, in form of bars.	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor.
Width	Size of the object (STAT).
Height	See section 9.3.3 Generalities regarding the online editor.
Main variable	<p>Main variable associated to the gauge (STAT).</p> <p>This variable determines the filling level of the gauge. The filling level depends on the display scale of the variable.</p> <p>When the variable is lower or equal to the minimum scale, the gauge is “empty”. When the variable is higher or equal to the maximum scale, the gauge is “full”. Between those values, the filling level is proportional.</p> <p>The display scale is defined in the properties of the variable (see section 4.4 Administration of variables.</p> <p>It is also used for the animation of the dynamic properties of the object. See section 9.3.7 Dynamic properties of objects.</p>
Border width	<p>Thickness selected for the contour (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The gauge contour delimits the whole object.</p>
Border color	<p>Color selected for the gauge contour (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The gauge contour delimits the whole object.</p>

Object : Gauge	
Presentation change of a variable compared to its display scale, in form of bars.	
Property	Description
Background color	<p>Color selected for the background (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The gauge background comprises any part of the object that is not masked by the filling area.</p>
Brush width	<p>Thickness selected for the outline (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The gauge outline is the line which demarcates the end of the filling area.</p>
Brush color	<p>Color selected for the outline (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The gauge outline is the line which demarcates the end of the filling area.</p>
Fill color	<p>Color selected for the filling (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The filling area is the area which evolves in function of the main variable value.</p>
Direction	<p>Orientation of the gauge filling (DYN). It can be either vertical or horizontal.</p>

Table 22. Properties available on VU meter objects.

Object: VU meter	
Presentation in the form of dials, variations of a variable compared to its display scale.	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor .
Width	Size of the object (STAT).
Height	See section 9.3.3 Generalities regarding the online editor .
Main variable	<p>Main variable associated to the VU meter (STAT).</p> <p>This variable is associated to the filling angle of the meter. The filling angle depends on the display scale of the variable.</p> <p>When the variable is lower or equal to the minimum scale the meter is "empty". When the variable is higher or equal to the maximum scale, the meter is "full". Between these two values, the filling angle is proportional.</p> <p>The display scale is defined in the properties of the variable (see section 4.4 Administration of variables).</p> <p>It is also used for the animation of the dynamic properties of the object. See section 9.3.7 Dynamic properties of objects.</p>
Border width	<p>Thickness selected for the border (DYN).</p> <p>See section 9.3.3 Generalities regarding the online editor.</p> <p>The contour of the meter demarcates the whole disk or only the portion of the disk comprised between the beginning angle and the final angle.</p>

Object: VU meter	
Presentation in the form of dials, variations of a variable compared to its display scale.	
Property	Description
Border color	Color selected for the meter border (DYN). See section <i>9.3.3 Generalities regarding the online editor</i> . The contour of the meter demarcates the whole disk or only the portion of the disk comprised between the beginning angle and the final angle.
Background color	Color selected for the background (DYN). See section <i>9.3.3 Generalities regarding the online editor</i> . The background of the meter comprises the whole disk or only the portion of the disk comprised between the beginning angle et the final angle, which are not masked by the filling area.
Brush width	Thickness selected for the brush (DYN). See section <i>9.3.3 Generalities regarding the online editor</i> . The brush of the meter is the line which demarcates the end of the filling area.
Brush color	Color selected for the brush (DYN). See section <i>9.3.3 Generalities regarding the online editor</i> . The brush of the meter is the line which demarcates the end of the filling area.
Fill color	Color selected for the filling area (DYN). See section <i>9.3.3 Generalities regarding the online editor</i> . The filling area is the area which evolves in function of the main variable value.
Start angle	Angle definition for the meter area (DYN).

Object: VU meter	
Presentation in the form of dials, variations of a variable compared to its display scale.	
Property	Description
Direction	The dial can be a full disk or only a portion. Angles are expressed in degree.

Table 23. Properties available on Rectangle objects.

Object : Rectangle	
Display of a geometrical form	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section <i>9.3.3 Generalities regarding the online editor.</i>
Width	Size of the object (STAT).
Height	See section <i>9.3.3 Generalities regarding the online editor.</i>
Main variable	Main variable for the animation of the object (STAT). See section <i>9.3.7 Dynamic properties of objects.</i>
Border color	Color of the border (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Fill color	Color of the filling area (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>
Border width	Thickness of the border (DYN). See section <i>9.3.3 Generalities regarding the online editor.</i>

Table 24. Properties available on Ellipse objects.

Object : Ellipse	
Display of a geometrical form	
Property	Description
X axis	Position of the object (STAT).
Y axis	See section 9.3.3 Generalities regarding the online editor.
Width	Size of the object (STAT).
Height	See section 9.3.3 Generalities regarding the online editor.
Main variable	Main variable for the animation of the object (STAT). See section 9.3.7 Dynamic properties of objects.
Border color	Color of the border (DYN). See section 9.3.3 Generalities regarding the online editor.
Fill color	Color of the filling area (DYN). See section 9.3.3 Generalities regarding the online editor.
Border width	Thickness of the border (DYN). See section 9.3.3 Generalities regarding the online editor.

9.3.6. Synoptic layers

On a same synoptic, it is possible to mask/hide certain objects according to the rights of the logged on user.

Indeed, a block can be viewed as the superposition of ten transparent layers.

By default, a new object is visible on all layers. This is editable in the properties of the object (see section [9.3.5 Synoptic Objects](#)).

The right of access to each layer is defined in the user profile (see section [2.1 Administration of user profiles](#))

Layers also allow to make it easier to edit synoptics when different objects are overlapped.

9.3.7. Dynamic properties of objects

In order to make an object property dynamic, you only need to associate it to a variable according to one of the methods explained next.

9.3.7.1. Property replaced by a variable

The property takes the value of the real time variable instead of being static.

The syntax is the following: `<=variable>`

A same object can feature different variables for each property. Such variables can differ from the main one.

Variables can also include text.

Example 9. Property replaced by a variable.

You wish to display an image representing an installation status.

The variable %MW0 in the PLC 1 represents that status. If it equals 0, the installation is not activated, if it equals 1, the installation is operating, if it equals 2, the installation has a default.

Steps to be followed would be the shown below:

- 1) You create an object image
- 2) In the list of images of the object, add the three images in order.
- 3) In the property **Image index**, specify: "`<=%MW0 . 1>`" (variable %MW0 of PLC 1)

The image is now animated.

- When %MW0 = 0, μServer displays the index image 0
- When %MW0 =1, μServer displays the index image 1
- When %MW0 =2, μServer displays the index image 2

If %MW0 =3, for instance, as index 3 does not exist; μServer would display nothing, thus making the area transparent.

9.3.7.2. Animated property on the main variable

A. List of values

The property picks up a value among a cross-referenced list between the value of the main

variable and the value of the property.

The syntax is the following: *Value1_variable:value1_property;value2_variable:value2_property...*

Example 10. Animated property on the main variable.

You wish to represent a motor with a color point.

The variable %MWO in the PLC 2 represents the motor status: if it equals 0, the motor is stopped and is displayed in white; if it equals 1, the motor is operating and is displayed in green; if it equals 10, the motor has a defect and is displayed in red.

- 1) Create an object Ellipse
- 2) In the property **Main variable**, specify "%MW0.1" (variable %MW0 of PLC 2)
- 3) In the property **Background color**, specify "0:#FFFFFF;1:#00FF00;10:#FF0000"

The color is now animated.

For each element on the list, the variable value can be either a unique value or a value range. For a value range, the syntax would be the following:

"Start 1->End1:property_value1;Start2->End2:property_value2;...;property_default_value"

Values *Start* and *End* are included in the range. In order to consider values from $-\infty$ or to $+\infty$, just leave blank the *Start* or *End* field concerned.

Example 11. Display text list.

To display a text list depending on the value of a variable, a text object associated to a main variable should be created, indicating the field "Text" with the following syntax:

"->0:Value out of range (negative);0->10:Weak value;10->20:Mean value;20->30: High value; 30->:Value out of range (positive);Error"

Note:

In any case µServer analyzes the conditions in the same order they have been defined and stops as soon as a condition is fulfilled.

For instance, if variable is 15 and the conditions are "10->:Higher than 10; 15:Equal to 15", the text will not be "Equal to 15" as the previous condition is first fulfilled.

B. Proportional variation

The property picks up a value proportional to the main variable, according to its display scale.

The display scale is defined in the properties of the variable (see section [4.4 Administration of variables](#)).

The syntax is the following: *Low value_property -> high value_property*

If the property only accepts integer values, the value is rounded to the nearest integer (for example, if the proportional value equals to 1,2, the image index is rounded to 1).

If the property is a color, a color gradient will be obtained.

Example:

You wish to adapt the font size of a text to the filling level of a tank. The variable %MWO of the PLC 1 represents this level, which varies from 0 to 100%. When the tank is empty, the font size of the text will be 10. When the tank is full, the font size of the text will be 18.

Steps to be followed are shown next:

- 1)** First, the display scales of the variable %MWO.1 (min=0; max=100) must be configured.
- 2)** Then, an object Text must be created.
- 3)** In the property "Main variable", specify "%MWO.1"
- 4)** In the property "Font", specify "10-18"

From now on, the font size of the text will evolve proportionally to the filling level of the tank.

9.3.8. Special controls

µServer allows to include special controls into the synoptic views.

For this purpose, it is necessary to create an object push button without a Main variable assigned but indicating "Value ON" on the property through one of the following syntaxes:

Page change

"%PAGE%=page_number"

This syntax allows to call a page on a active synoptic.

Call of a µServer function

```
"%SCRIPT%=µServer_function()"
```

This syntax allows to execute a function among the following available functions listed in 25, by using the following syntax.

Table 25.µServer functions available for synoptics.

µServer functions available for synoptics	
Function	Description
linkDevice(id)	A device view is opened.
popupDevice(id)	<p><i>linkDevice()</i> opens the view on the present tab</p> <p><i>popupDevice()</i> opens the view on a new tab</p> <p>Parameters:</p> <p><i>id</i>: device number</p> <p>Example:</p> <pre>%SCRIPT%=linkDevice(10)</pre> <p>This would open the view of device 10 on a new tab.</p>
link("view","params")	A generic view of µServer is opened.
popup("view","params")	<p><i>link()</i> opens the view on the present tab</p> <p><i>popup()</i> opens the view on the present tab</p> <p>Parameters:</p> <p>"<i>view</i>": name of the php page of µServer. Extension ".f php should not be specified".</p> <p>"<i>params</i>": supplementary parameters of the php page. Parameter "<i>db=nom_de_la_base</i>" should not be specified.</p> <p>Examples:</p>

µServer functions available for synoptics	
Function	Description
	<p><code>%SCRIPT%=link("home", "")</code></p> <p>Link to the user home page (home.php).</p> <p><code>%SCRIPT%=link("logtr", "&var1=%MW0.1&dynamic=1&pval=12&pmult=3600")</code></p> <p>Link to the curve of variable %MW0.1, in real time mode, display of the 12 last hours.</p> <p><code>%SCRIPT%=link("home", "&connect=-1")</code></p> <p>Disconnection of the user and link to the home page.</p> <p>Most of the µServer views are accessible and configurable this way. To know about the name of a µServer view and its parameter list, just go to this view and retrieve the address stated on the address bar in the navigator.</p>
call(id)	<p>Device refresh.</p> <p>It is similar to the button "Call" (see section 3.1.1.2 Data updates)</p> <p>Parameters:</p> <p><i>id: device number</i></p>
back()	<p>Back to the previous page.</p> <p>It's the equivalent to the return button on the navigation menu of µServer (see section 13 Generalities of µServer interface)</p>
logOut()	<p>Unlog user.</p> <p><i>This function exists from 27/08/2014.</i></p>
editAccount()	<p>Link to the account parameters of the connected user.</p> <p><i>This function exists from 27/08/2014.</i></p>

Javascript code execution

```
"%SCRIPT%=instruction1_javascript;instruction2_javascript"
```

This syntax allows to execute any javascript instruction.

Example 12. Javascript instructions.

```
%SCRIPT%=window.location.href='http://www.sireagroup.com'
```

Link to a web address.

```
%SCRIPT%=alert('Test button')
```

Message display.

Note:

The ON value of a button can be directly a special syntax or a reference to a variable containing the special syntax (see Section [9.3.7.1 Property replaced by a variable](#)), offering also a wide range of programming possibilities.

9.3.9. Global variables of a synoptic

In some cases it is interesting to create a synoptic model which could be duplicated just changing automatically properties on certain objects.

Example:

We want to assign an identical synoptic to each device. On this synoptic, the value of several variables of the device are shown as well as some buttons that force that variables. A synoptic can be developed for device 1, and then it can be duplicated (copy or export/import) for the device 2. For that, it is necessary to review all the objects to modify addresses making reference to device 1 in order to switch them to device 2. For instance, %MWO.1 turns into %MWO.2, etc.

With the aim of avoiding this tedious manipulation, it is possible to develop a generic synoptic on which the equipment number will be a global variable inside the synoptic.

%MWO.1 will be indicated as %MWO.{device}. Once the generic synoptic has been developed, it is duplicated and then a modification is done on its global variables (tool bar at the synoptic editor) just changing {device} by "1". μServer looks for all {device} occurrences inside the properties on all objects and change them to "1".

By duplicating again the generic synoptic and reiterating the same operation for the device 2,

and this way with every other device, the group of synoptics can be easily created, taking care of keeping always a copy of the generic synoptic.

Note:

- An unlimited number of global variables as desired can be defined thanks to the syntax `{variable_name}`
- Change on global variables directly affects the synoptic and cannot be canceled. It is necessary then to do a copy of the synoptic or a previous backup.

9.4. Synoptic visualization

From the synoptic editor, click on **Execute View** to switch to visualization and then to **Edit** to turn back to the editing mode.

This access is reserved to administrators having access to [Views](#) menu.

The following methods are available for the users:

- Synoptic associated to a device
- Synoptic associated to a user account
- Default synoptic

In these three cases, μServer proposes to automatically adapt the synoptic size to the screen of the connected user. For that purpose, it is necessary to create different synoptics with same name but different size. The web navigator sends the display size, thus allowing μServer to show the most suitable synoptic.

This size adaptation is available from 27/08/2014.

9.4.1. Associating a synoptic to a device

As administrator, modify the property **View** of the device (see section [3.2 Administration of equipment](#))

A “synoptic view” tab appears then on the device view and displays the synoptic in visualization mode. This tab becomes the default tab when opening the device view.

9.4.2. Associate a synoptic to a user account

As administrator, modify the property **View** of the user account (see section [2.2 Administration of user accounts](#)).

When a user connects to that µServer, its home page will be then this synoptic in visualization mode in full window. The µServer navigation menu is then hidden by the synoptic and any other view of µServer are called from the synoptic.

As the menu is hidden, it is a task of the creator to create the needed navigation buttons as “previous”, “unlog”, etc. Anyway, it is also possible to force the menu display on the user account properties, if desired.

Note:

- It is possible to force the menu display on the user account properties if desired.
- Do not assign a synoptic home view to an administrator account, as the access to the menu can be lost if the menu display is not activated on the account properties.
- µServer can be configured in such a way that the full window view can be ignored by displaying the menu in any case.
- If a super-administrator is defined, he ignores the full window mode by showing the menu in any case.

9.4.3. Default synoptic

If a synoptic without name is created, it is considered as default synoptic.

Default synoptic appears on the home page of every user without a specific synoptic associated to their account.

Default synoptics are managed from 27/08/2014.

10. Scripts

Scripts (procedures) are the programming tool in µServer. They enable to carry out all type of actions thanks to the following functionalities:

- Punctual call of a script
- Periodic call of a script
- Call of a script each time a variable or some variables suffer a value change
- Python language interpreter integrated
- Specific µServer functions
- Access to database

10.1. Administration of scripts

In order to create a new script:

[Scripts](#) → [Add](#)

In order to visualize a script:

[Scripts](#) → [Search](#) → Click on the script

To modify one or several scripts:

[Scripts](#) → [Search](#) → Click on the script → [Edit](#)

Or:

[Scripts](#) → [Search](#) → Select script/scripts to modify → [Edit](#)

To copy a script:

[Scripts](#) → [Search](#) → Click on the script → [Copy](#)

To delete one or several scripts:

[Scripts](#) → [Search](#) → Click on the script → [Delete](#)

Or:

[Scripts](#) → [Search](#) → Select script/scripts to delete → [Delete](#)

Note:

All these actions require administrator rights.

Properties which are configurable on scripts are shown on table.

Table 26. Configurable properties on µServer scripts.

Scripts properties	
Property	Description
Name	Script name.
Comment	Script description.
Trigger variables	Variables associated to script.

Scripts properties	
Property	Description
	The script is executed any time a value change occur at any of these variables. For specifying several values, they must be separated by “,”.
Run on startup	Execute the script on µServer startup.
Run periodically	Execute the script periodically.
Week days	Definition for the execution period.
Month days	The script is executed whenever the execution period is attained, that is when the conditions defined for the execution period in all the fields defined are met.
Month	
Hours	
Minutes	
Seconds	<p>For each of the fields, leave blank for specifying “any value”. Otherwise, specify the numeric value or several separated values separated by “,”.</p> <p>Example:</p> <p>Script to be executed any Sunday at 11pm.</p> <p>Week day = 7</p> <p>Month day =</p> <p>Month =</p> <p>Hours = 23</p> <p>Minutes = 0</p> <p>Seconds = 0</p> <p>Example:</p> <p>Script to be executed every 10 minutes.</p> <p>Week day =</p> <p>Month day =</p> <p>Month =</p>

Scripts properties	
Property	Description
	Hours = Minutes = 0,10,20,30,40,50 Seconds = 0
Code	Script python code. For further information see section 10.2 Programming .

10.2. Programming

10.2.1. Principle

Scripts are written in Python language. Programming syntax, rules and conventions are defined by the Python language standards.

Each script can be seen as a Python function which is automatically called by µServer depending on the execution conditions defined (see section [10.1 Administration of scripts](#)) or called by the program by means of the function `call()` (see section [10.2.2 µServer functions](#)). It is then possible to execute a script manually by clicking on the [Run](#) button on a script view.

The python code associated to a script is written into the **Code** property (see section [10.1 Administration of scripts](#)).

10.2.2. µServer functions

Besides the standard Python functions, µServer integrates a specific function library.

µServer specific functions available for script creation	
Function	Description
get(variable)	Read value of a µServer variable.
get()	<code>get()</code> can return different variable type in python: <ul style="list-style-type: none"> – If variable is <code>null</code> (variable never initialized) or if the variable

µServer specific functions available for script creation	
Function	Description
	<p>does not exist, <code>get()</code> returns <i>None</i> (Specific value in python)</p> <ul style="list-style-type: none"> – If variable has a numerical value, <code>get()</code> returns an integer (int) or float (float) – If variable has a text value, <code>get()</code> returns a string (str) – From 09/09/2014, µServer variables can also have complex types among the following: <i>dict, list, tuple, boolean</i>
geti()	<p><code>geti()</code> returns an integer.</p> <ul style="list-style-type: none"> – If variable is <i>null</i> (variable never initialized) or if the variable does not exist or its value is not integer, <code>geti()</code> returns <i>0</i>
getf()	<p><code>getf()</code> returns a float.</p> <ul style="list-style-type: none"> – If variable is <i>null</i> (variable never initialized) or if the variable does not exist or its value is not numerical, <code>getf()</code> returns <i>0.0</i>
getb()	<p><code>getb()</code> returns a boolean (<i>True</i> or <i>False</i>).</p> <ul style="list-style-type: none"> – If variable is <i>null</i> (variable never initialized) or if the variable does not exist, <code>getb()</code> returns <i>False</i>
gets()	<p><code>gets()</code> returns a string.</p> <ul style="list-style-type: none"> – <i>If variable is null (variable never initialized) or if the variable does not exist, gets() returns "" (empty string)</i> <p>Parameters</p> <p><i>variable</i> : variable address or mnemonic. String type.</p> <p>Example 1: read numerical value with geti()</p>

µServer specific functions available for script creation	
Function	Description
	<pre>variable_python=geti("%MW0.1")+10</pre> <p>Example 2: read numerical value with get()</p> <pre>variable_python=get("%MW0.1")</pre> <pre>try:</pre> <pre> variable_python=int(variable_python)</pre> <pre> variable_python+=10</pre> <pre>except:</pre> <pre> log("Variable is not numerical")</pre> <p>Example 3: read by mnemonic</p> <pre>variable_python=geti("MY_VAR")</pre> <p><i>Functions geti(), getb() and gets() exist from 27/08/2014.</i></p>
set(variable,value)	<p>Write value onto a µServer variable.</p> <p>Parameters</p> <p><i>variable</i> : variable address or mnemonic. String type.</p> <p><i>Value</i>: value to be written. It can be text or numerical. From 09/09/2014, µServer variables can also have complex types like: <i>dict</i>, <i>list</i>, <i>tuple</i>, <i>boolean</i>.</p> <p>Example 1: numerical value</p> <pre>set("%MW0.1",1)</pre> <p>Example 2: text value</p> <pre>set("%PW0","hello")</pre> <p>Example 3: list</p> <pre>set("%PW0",[1,2,3])</pre>

µServer specific functions available for script creation	
Function	Description
	<p>Example 4: use of mnemonic</p> <pre>set("MY_VAR",1)</pre>
log(text)	<p>Add a text to the debug log accessible through the Console, available inside a Script edition panel.</p> <p>Parameters</p> <p><i>text</i> : text to be written. String type.</p> <p>Example:</p> <pre>variable_python=10 log("The python variable value is"+str(variable_python))</pre>
call(script,[params],[sync])	<p>Call another µServer script.</p> <p>Parameters</p> <p><i>script</i> : name of the string to be called. String type.</p> <p><i>params</i> (optional): parameters to be transferred to the called script. "<i>params</i>" is a dictionary (type of python variable) in which each key is the name of the parameter to transfer and each value represents the value of the parameter to transfer.</p> <p>Parameters do not need to be defined in the called script. As long as the parameter is passed in the dictionary, it can be used in the called script as if it was a python variable already defined.</p> <p><i>sync</i> (optional): By default, calling a script through the <i>call()</i> function is blocking. This means that the calling procedure waits for the called procedure to be completed. It is possible to call a procedure in a nonblocking manner by adding the optional parameter "sync = False". In this case, the called procedure is executed in parallel with the calling procedure.</p> <p>Example: <code>call("ma_script",sync=False)</code></p>

µServer specific functions available for script creation	
Function	Description
	<p>Return</p> <p>Procedures can return values the same way as Python function. Just use the standard function <i>return</i> at the end of the procedure to be called. Retrieving returned values in the calling procedure is done simply by assigning the <i>call()</i> function to a variable.</p> <p>For example: <code>variable_python=call("script1")</code></p> <p>Example 1:</p> <p>Script "StopEngine1": code → <code>set("%MW10.1",0)</code></p> <p>Script "StopEngine2": code → <code>set("%MW11.1",0)</code></p> <p>Script "StopMachine": code:</p> <pre>if geti("%MW0.1")==1 : # If stop machine commanded call("StopEngine1") # Stop engine 1 call("StopEngine2") # Stop engine 2 set("%MW0.1",0) # Stop request acknowledgment</pre> <p>Example 2:</p> <p>Script "Sum": code → <code>s=a+b</code></p> <pre>return s</pre> <p>Script "PowerCalc": code:</p> <pre># Power sum calculation: engine 1 + engine 2 params={'a':geti("%MW20.1"),'b':geti("%MW21.1")} power=call("Sum",params)</pre>
query(request)	<p>Execute a SQL request inside the µServer database.</p> <p>Direct requests to the database are reserved to an advanced use of</p>

µServer specific functions available for script creation	
Function	Description
	<p>µServer. The structure of the database may change depending on the version of µServer.</p> <p>Parameters</p> <p><i>request</i> : request to be executed. String type.</p> <p>Requests are in standard MySQL format.</p> <p>Return</p> <p>Result of the request in a two-dimensions table format.</p> <p>It returns "None" in case of failure.</p> <p>Example:</p> <pre># Total number of present alarms in the server q="select * from alarm" tab=query(q) if tab!=None: nb_alarms=len(tab) log("Total number of alarms: "+str nb_alarms) else: log ("Error sql!")</pre>

10.2.3. Debug console

When Python finds an execution error, the script is interrupted and µServer logs the error (date, line number and error message) in the database.

For the debugging and the monitoring, µServer allows to display messages at every point of a script thanks to the function `log()` (see section [10.2.2 µServer functions](#)).

The [Console](#) enables to visualize/delete error and log messages and to filter them by script.

This view is dynamically updated, that means that the appearance of a error is automatically

shown, without need of a page refreshing. This way the developer can work on a script on a window/tab, having at the same time an additional window/tab where he can see the error and log messages in another window/tab. Just empty the [Console](#) ([Delete](#) button on the console), then execute manually the script ([Run](#) button on the Script view) for the visualization of possible messages.

To access the console for all procedures combined:

[Scripts](#) → [Console](#)

To access the console for just a given script:

[Scripts](#) → [Search](#) → Click on script → [Console](#)

Or:

[Scripts](#) → [Console](#) → Click on script name inside the message table

11. Languages

μServer enables to associate a linguistic parameter “Regional format” to each user account (see section [2.2 Administration of user accounts](#))

Available regional formats are the following:

- "FR": Language = "FR"; Date format= "dd/mm/yyyy"; Hour format = "hh:mm"
- "GB": Language = "EN"; Date format = "mm/dd/yyyy"; Hour forma = "hh:mm am/pm"
- "US": Language = "EN"; Date format = "mm/dd/yyyy"; Hour forma = "hh:mm am/pm"

Where the language codes make reference to available languages:

- "FR": French
- "EN": English

If the regional format of the account is not defined, μServer applies the regional format defined as default format.

Note:

The language is applied only to μServer interface (menus, buttons, links...), but in no case to the database.

In other words, if variables of a device have a French label, a user having associated the English language to his account will see the variable list in French too. It is the case of any text on a device, variable, synoptic, alert reports...

12. Hour and date

12.1. Timestamp of the data

The timestamp of the device data (measurements, alarms, events) depends on the type of communication.

Communication in “passive” mode

The log points are generated by the device and then sent to the server via the communication protocol. In this case, it is the time of the device which serves as reference to the timestamp. Thus it has to be ensured that all devices are on time.

Communication in “active” mode

Log points are generated by μServer it is then the server time which serves as reference for the timestamp.

12.2. Timezones

μServer manages internally all the time and date data as UTC.

Being the time uniformed this way, μServer can convert them in any timezone, because **any time shown from the μServer interface is shown in local time of the user connected.**

The time shown above the main menu is the server time (and not the connected PC time), but converted into local time of connected PC.

When a csv file is exported from the μServer interface, time are converted as well into local time of the connected PC.

In the case of a “passive” mode communication, equipments send their data converted into UTC time. We would remark then the importance of properly the timezones in the devices.

Example 13. Timezones.

A central server is installed on a country A and a device is installed in country B. Country B has 3h jet lag with country A. Another country C has 2h jet lag with country A.

An alarm appear at 9h, local time of the device, and the device sends this alarm to the server in

universal format.

If a user connects to the server from country B (the country where the device is located), he will see the alarm appearance at 9h.

If a user connects to the server from country A (the country where the server is located), he will see the alarm appearance at 6h.

If a user connects to the server from country C, he will see the alarm appearance at 4h.

13. Generalities of µServer interface

13.1. Search filters

On the [Search](#) views, filters allow to specify the search criteria. If a filter is left blank, it will not be applied. If a filter is filled, it will be underlined in blue and will be applied to the research.

For some textual properties, the research will include all the texts contained in the specified string. For instance, in the variable search menu, typing “voltage” into the comment filter will allow to look for all variables having a comment including the word “voltage”.

Device	<input type="text"/>
Type	<input type="text"/>
Address	<input type="text"/>
Label	<input type="text"/>
Comment	<input type="text"/>
Public	-- <input type="text"/> ▾
Writable	-- <input type="text"/> ▾
List of categories	<input type="text"/>

Figure 11. µServer variable search filter.

For some properties it is possible to specify several values separated by “,”. For instance, on the variable Search menu, it is possible to type “MW, MD” on the **Type** filter to find every variable of type “MW” or “MD”.

For some properties, a suggestion list allows to select a value among the already existent values

in the database. For example, on the variable Search menu, a single click on the **Device** filter field displays the list of the already existent devices.

Device	<input type="text"/>
Type	<input type="text" value="1"/>
Address	<input type="text" value="2"/>
Label	<input type="text"/>
Comment	<input type="text"/>
Public	<input type="text" value="--"/>
Writable	<input type="text" value="--"/>
List of categories	<input type="text"/>

Figure 12. µServer variable search filter with device field list active.

When the property authorizes to specify different values, the suggestion list ends by "+", allowing to introduce a new value.

When the suggestion list is too long, it is not displayed to avoid the slowing down of the interface.

For certain properties, a set of colors allows, while typing, to visualize the existence of this value on the database:

- Black: existent value
- Red: inexistent value
- Orange: one or some of the values on the list exist, but not all

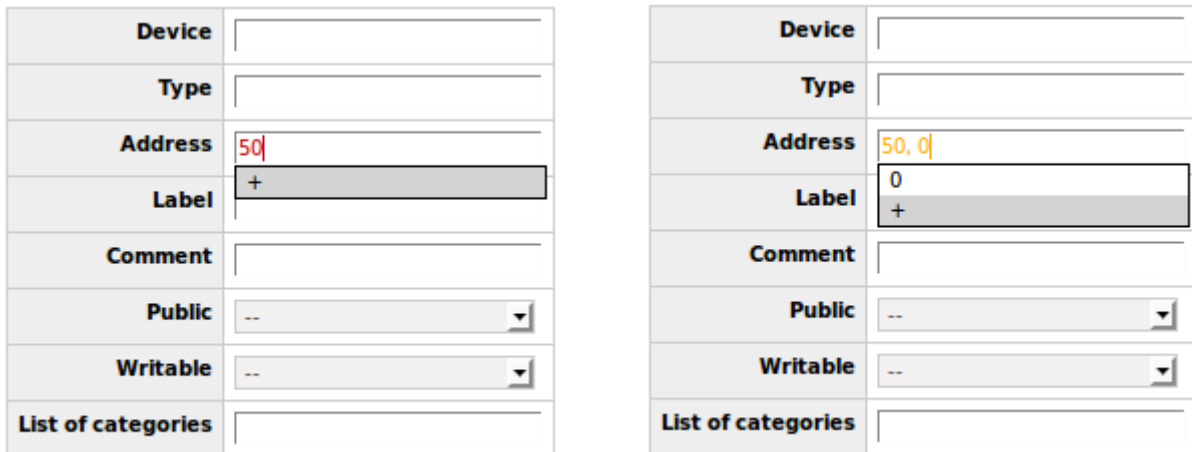


Figure 13. Set of colors: red (left) and orange (right).

13.1.1. Edition of elements

When editing an element, if a property of this element is linked to another type of element (for example, the property “profile” of a user account is linked to a user profile) µServer proposes a smart text input.

Depending on the properties, µServer proposes either a drop-down list, or a suggestion list. In this last case, a set of colors shows, while typing, the state of the link.

- Black: link correct
- Red: link incorrect
- Orange: one or some of the link on the list are correct, but not all

13.1.2. Multiple editions

It is also possible to modify simultaneously several elements from a Search view. Just follow the next steps:

- Select elements through the check boxes (the symbol Sel. Allows to select/deselect all the elements) and then click on Edit.
- This view makes it possible to see (remarked with blue frames) those properties which are already identical on all selected elements.
- Enter the property to be modified on all elements. If this property was not yet common on all selected elements, it will be then remarked with blue frames as well.

Every property remarked with the blue frame will be changed for all the elements. Double click on a property to add/remove it from list.

- Validate modification

13.1.3. Previous button

Previous button is located at the top left corner of most of the μServer views, symbolized with an arrow. This button is a little more complete than the “Previous” navigator button as it allows to:

- Call of a previous page of a synoptic, when navigating through the pages of a synoptic.
- Navigation standard return on a web, if navigation is not done through the pages of a synoptic.
- Closure of a tab if the tab has been opened by μServer and early log reached on this tab.

IV. ANNEXES

1. Annex A. Revision history

Date	Modification
08/08/2014	First official version of the µServer User Guide.
10/24/2014	Python functions geti(), getf(), getb(), gets() added. Specifications on set() function added. Synoptic functions logOut(), editAccount() added. Default synoptic management added. Option to create different synoptic sizes for its adaptation to user screen size added. Specifications on text forcing values added.