IPSES
scientific electronics

IPSES S.r.l.

CUSTOM STEP TYPES DESCRIPTION FOR EQ METERS

Rel 01.01.000

NATIONAL INSTRUMENTS Silver Alliance Partner
AUTHORIZED MICROCHIP DESIGN PARTNER
XJTAG TECHNOLOGY PARTNER
6TL Engineering
ACCREDIA
SISTEMA DI GESTIONE CERTIFICATO ITALCERT UNI EN ISO 9001

The information contained in this document is the property of IPSES S.r.l. and must be considered and treated as confidential.

This publication may be reproduced, transmitted, transcribed or translated into any human or electronic language only with the written permission of IPSES S.r.l..

The information contained herein has been carefully verified and is considered valid as of the date of publication of this document.

The information contained in this document is subject to change without notice and does not represent a commitment on the part of IPSES. The design of this equipment undergoes continuous development and improvement. As a result, the equipment associated with this document may contain minor differences in detail from the information provided in this manual.

Printed in Italy

Copyright © 2020-21 IPSES S.r.l.

All rights reserved.

IPSES s.r.l.
Registered office : Piazzale Giulio Cesare, 9 - 20145 Milan ITALY
Research and development office: Viale Carlo Forlanini, 44A - 20024 Garbagnate Milanese (MI) ITALY
VAT code: 03999740966 - Tel. (+39) 02 39449519 – Fax (+39) 02 700403170 e-mail: info@ipses.com – https://www.ipses.com

CONTENTS

CUSTOM STEP TYPES DESCRIPTION FOR EQ METERS 0
1 REVISIONS 4
1.1 Manual revisions 4
2 CUSTOM STEP TYPES 5
3 CUSTOM STEP TYPES EQ METERS HMI (ImageTesting) 6
3.1 Image Grabber Init 6
3.2 Image Grabber Close 8
3.3 Image Grab 9
3.4 Image Save 10
3.5 Image ROI Matching 12
3.6 Image Number Recognition 14
3.7 Image Monitor Init 16
3.8 Image Monitor Close 17
4 CUSTOM STEP TYPES EQ METERS HMI (SimulTyping) 18
4.1 Typing Init 18
4.2 Typing Execute 20
4.3 Typing Close 21
5 USE OF STEP TYPES IN TESTSTAND 22
6.1 Custom Step Types Introduction 22
6 SYSTEMS REQUIREMENTS 23
6.1 Software Requirements 23
6.1.1 Requirements for LabVIEW Run-Time Engine 23
6.2 Hardware Requirements 23
6.2.1 Connections 24
7 CONTACTS 28
8 ENGINEERING PROBLEM REPORT 31

1 REVISIONS

1.1 Manual revisions

Revision/ Date	Revision Description	Author
19/09/2022	First release	Matteo Molteni
19/07/2023	"Number Recognition" update and connections descriptions	Matteo Molteni

2 CUSTOM STEP TYPES

This chapter describes shared commands and behavior for custom Step Types. These Step Types are created for performing tests within NI-TestStand test sequencer environment.

Each step is designed with an "Edit" panel devoted to Step configuration. Specific functionalities of Step Types are explained in following sections. The general structure of "Edit" panels is the same for every Step, therefore, common commands are described below.

RUN: Executes the custom step, the function that is executed when the sequence is running.

SAVE: Saves the properties changes made in the Edit panel, which will be used during sequence running. The Edit panel is then closed.

CANCEL: Ignores the properties changes made in the Edit panel and keeps the parameters from the previous configuration. The Edit panel is then closed.

REVERT: Ignore and erase the properties changes made in the Edit panel recovering the initial ones (at the start of the Edit panel)

The Step Properties are described or each Step Types. Such properties are step specific and allow the configuration of the same.

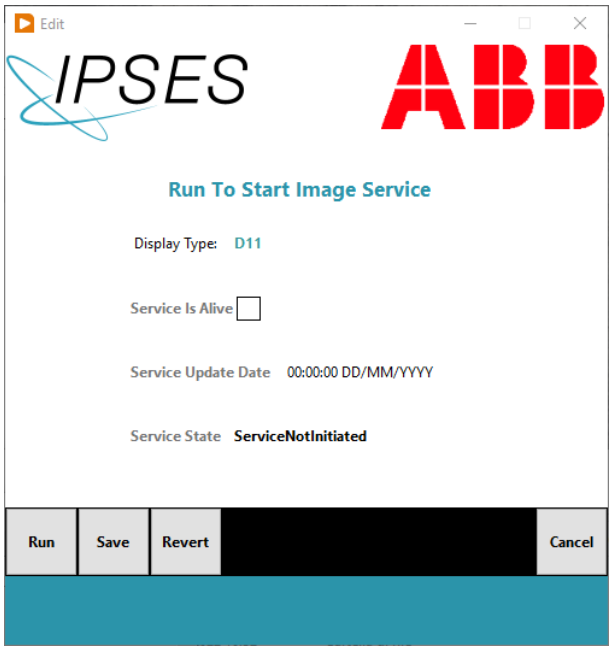
Before installing Step Types all software requirements must be installed on the target system (LabView Runtime, TestStand, VISA and Vision), see 6.1 for more details.

3 CUSTOM STEP TYPES EQ METERS HMI (ImageTesting)

In this chapter the step types for acquiring and analyzing the HMI screens of the EQ METERS are briefly described.

The acquisition system is based on a background service that acquires (or "sniffs") data processed by the display to reconstruct images, emulating the graphics chipset. At low level a collection of digital lines is monitored and analyzed in order to reproduce the image proposed by physical display.

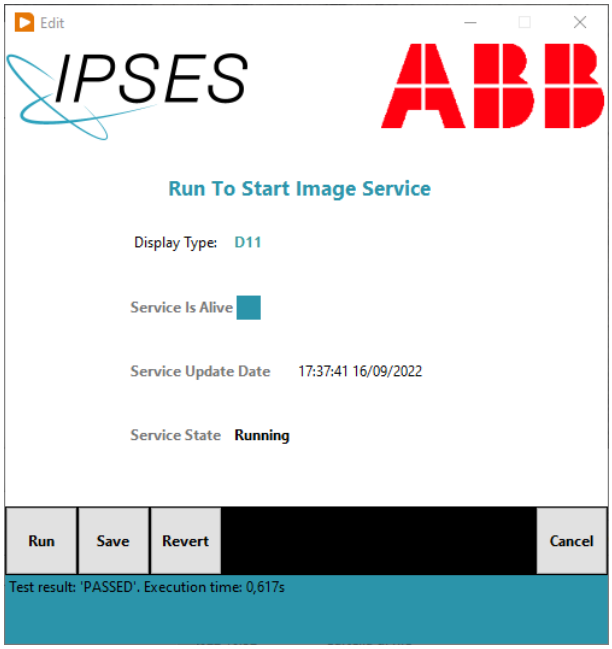
3.1 Image Grabber Init



This Step Type allows to configure which kind of display is installed in system setup:

Display Type allows to define display model from a drop-down menu.

It is possible to execute the step by clicking on the **RUN** button during the Edit phase.

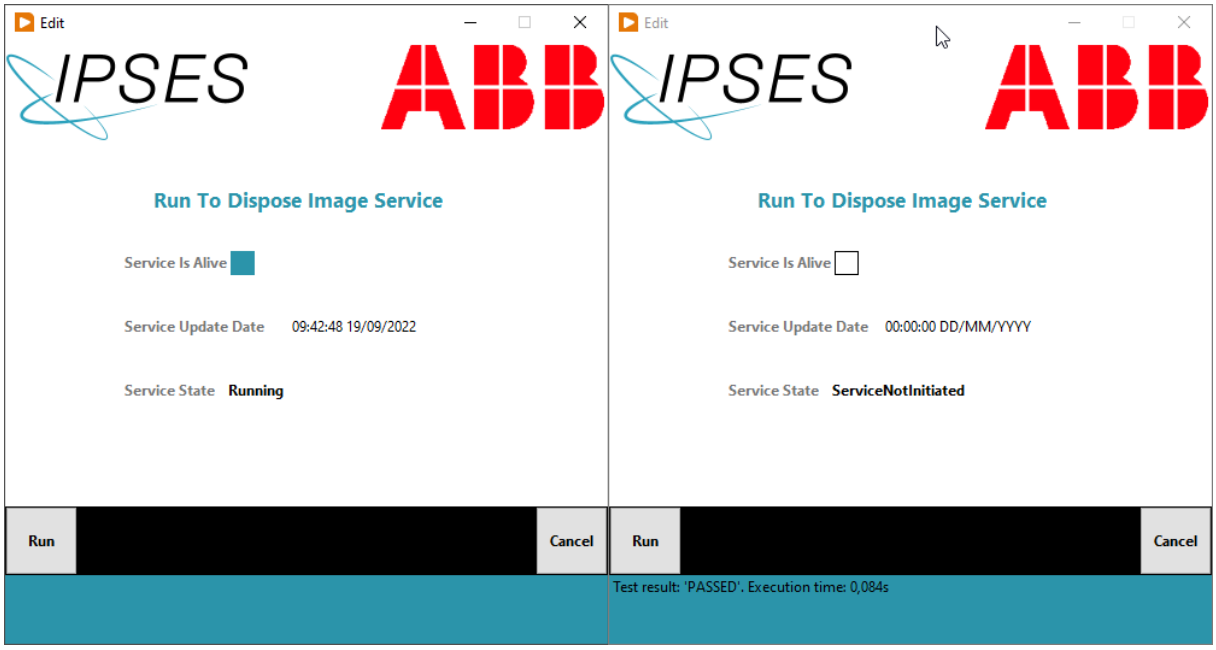


Service Is Alive indicates whether the Image Grabber service is operational (water green LED).

Service Update Date indicates the date of the last status update of the "Image Grabber" service.

Service State indicates the status of the Image Grabber service.

3.2 Image Grabber Close



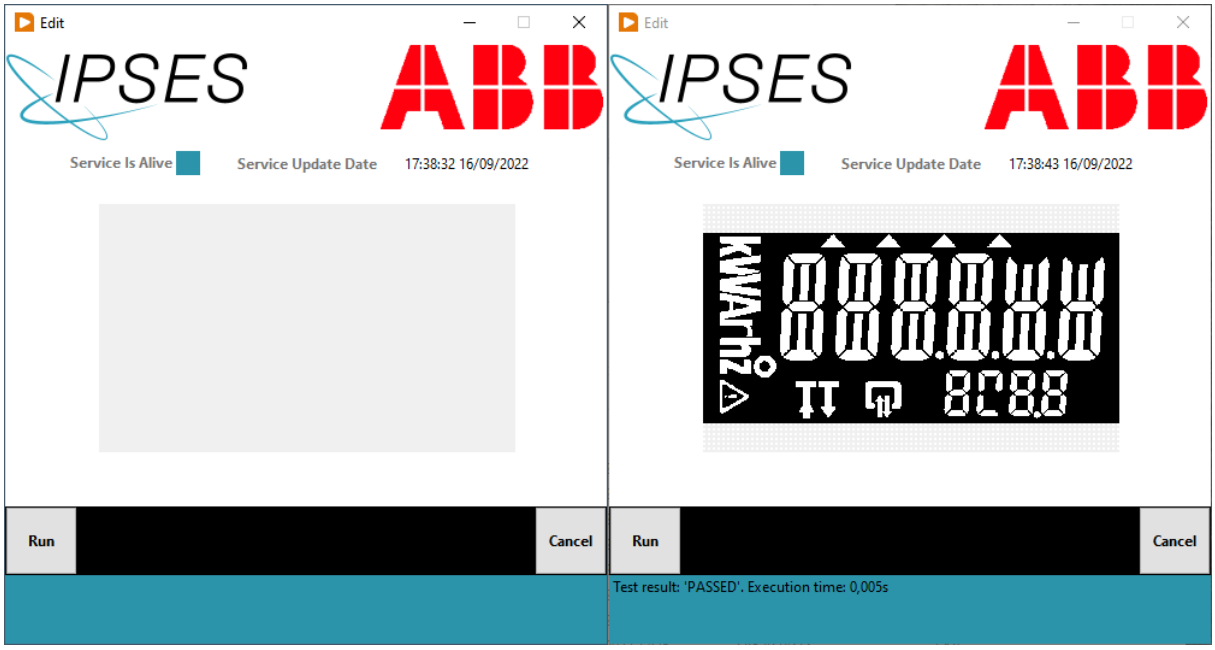
This Step Type has no configurable parameters in the Edit panel as it is used to close the data acquisition and processing session towards the display. However, it is possible to perform this operation by clicking on the **RUN** button during the Edit phase.

Service Is Alive indicates whether the Image Grabber service is operational (water green LED).

Service Update Date indicates the date of the last status update of the "Image Grabber" service.

Service State indicates the status of the Image Grabber service.

3.3 Image Grab

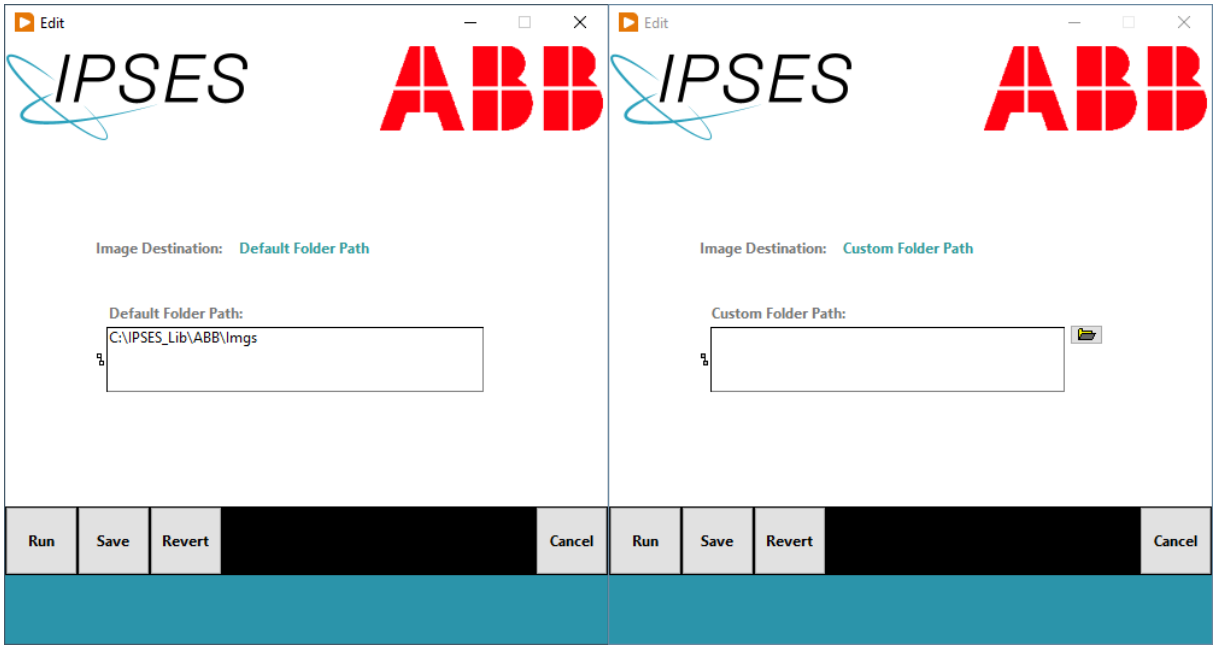


This Step Type has no configurable parameters in the Edit panel as it serves to bring into memory the last image sniffed by the service. It is possible to perform this operation by clicking on the **RUN** button during the Edit phase. The last image brought in memory is reported in the dedicated box.

Service Is Alive indicates whether the Image Grabber service is operational (water green LED).

Service Update Date indicates the date of the last status update of the "Image Grabber" service.

3.4 Image Save



This Step Type allows to configure the saving of image acquired with the Image Grabber step type. You can choose between the default path (defined by a configuration file located in the packed library installation folder, *i.e.*: "C:\IPSES_Lib\ABB\EqMeters\HMI\Config_ImageTesting.txt"), a path selected by the

operator or a path relative to the folder hosting the sequence that is running the step. The image is saved in bitmap format with a file name encoding the date and time of the image (e.g.:" 20211126_163240_945.bmp").

Image Destination allows to define the path type where to save the images during the RUN. (*Default Folder Path*, *Custom Folder Path* or *Custom Relative*)

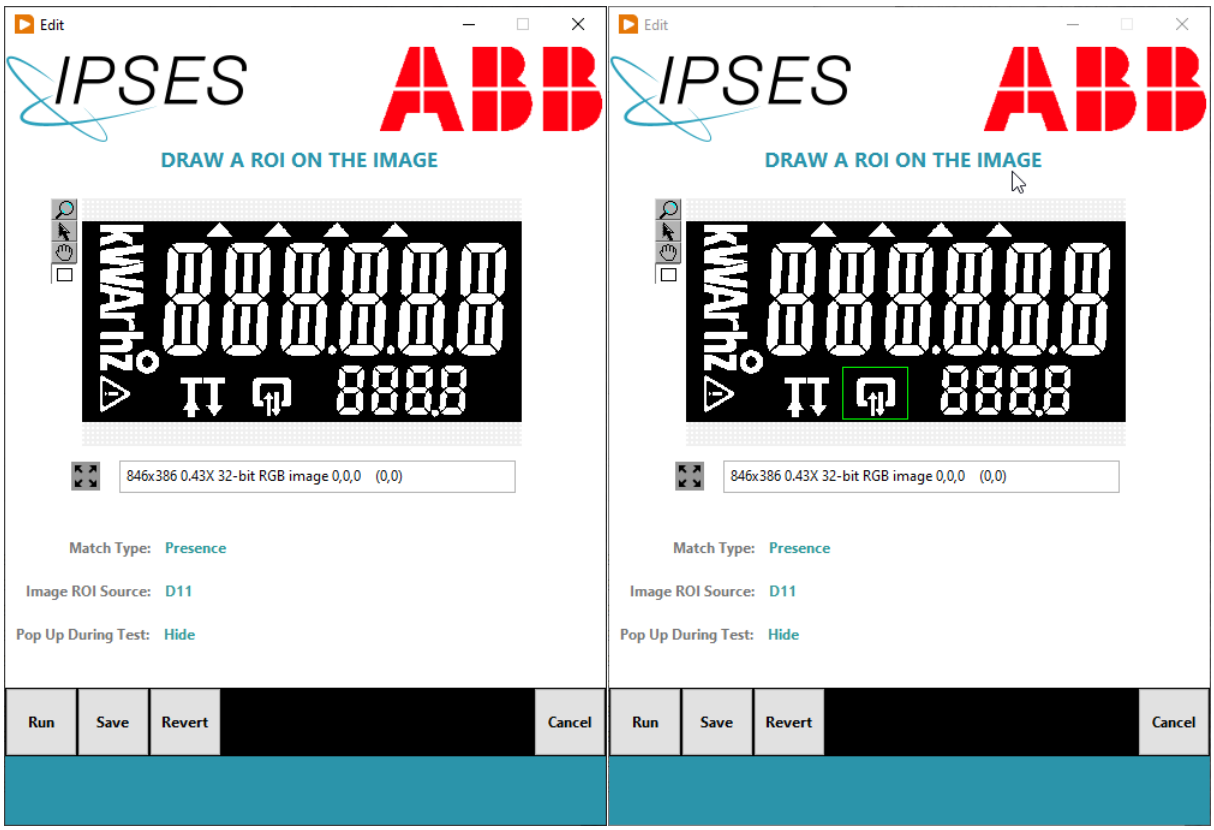
Default Folder Path indicates the default path defined in the configuration file "Config_ImageTesting.txt".

Custom Folder Path allows to define a custom path where to save the images during the RUN.

Sequence Folder Path indicates the folder hosting the sequence currently calling the step.

Custom Relative Path allows to define a path relative to the Sequence Folder Path where to save the images during the RUN.

3.5 Image ROI Matching

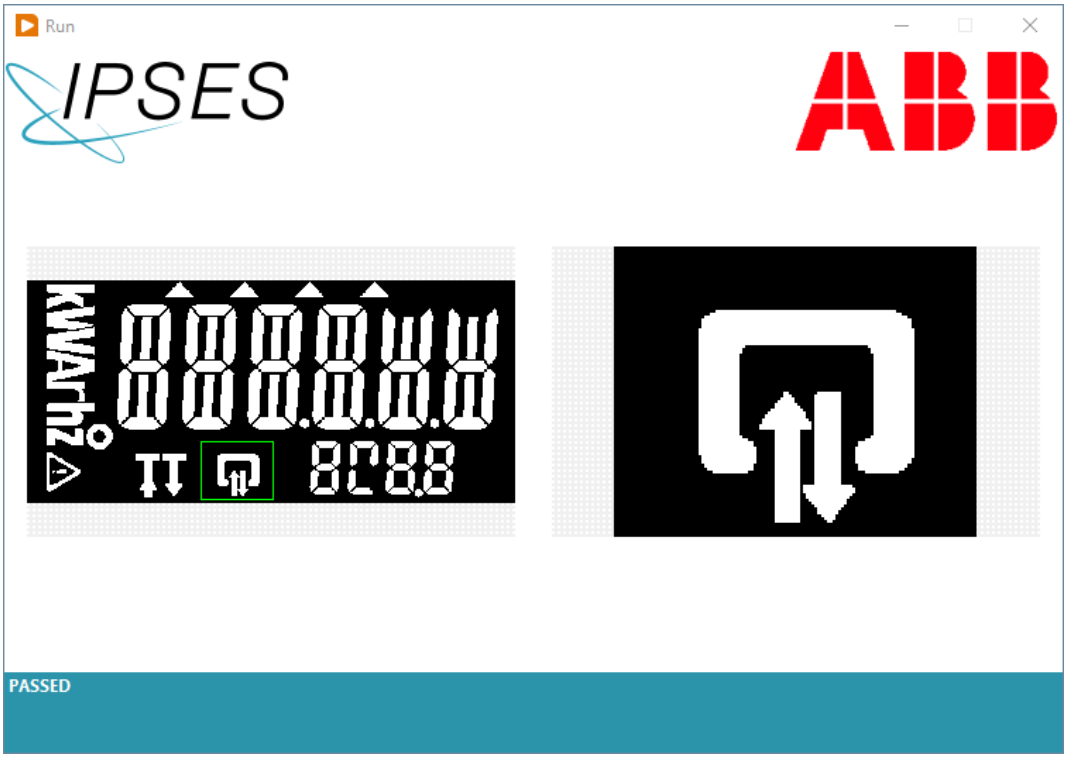


This Step Type allows to compare pixel by pixel a ROI of a grabbed image to a previously defined matrix of pixels. In Edit phase, "***Image ROI Source***" sets the reference image to define the matrix of pixels: when "File" is selected, an existing image file path must be indicated; when "Buffer" is selected, the reference image is the image fixed by the step type "Image Grab"; when a display specific tag is selected, such as "D11", the image presents all the segments of the display model. Once a valid image source is set, the image will populate the indicator and a ROI could be defined through mouse interaction. "***Pop Up During Test***" configure the behavior of the step type during execution, if "Show" is selected, a Popup will appear showing the comparison between the Image grabbed and the reference matrix of pixels, while, if "Hide" is selected, the comparison will be performed without such indication. "***Match Type***" configures step working mode: when "Presence" is selected, perfect pixel by pixel match search is executed, while when "Absence" is selected, the ROI must be empty to have a PASSED step. The duration of the comparison

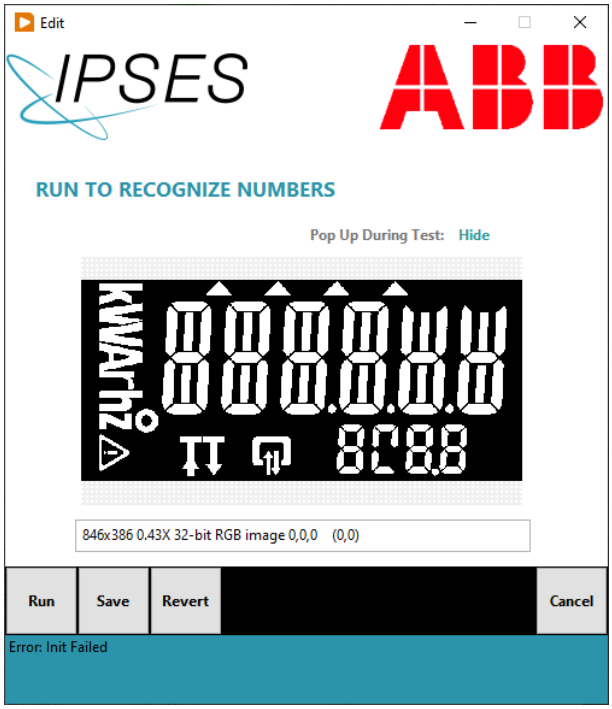
Popup is defined in the Configuration File at "C:\IPSES_Lib\ABB\EqMeters \HMI\Config_ImageTesting.txt".

The test will PASS only if all the pixels of the selected ROI match the reference matrix of pixels (if "Presence" working mode is selected).

The comparison Popup that could be shown during execution has the following appearance:



3.6 Image Number Recognition



This Step Type allows to identify numbers present in a selected ROI. In Edit phase, the image display is populated by the currently grabbed image as reference. "***Pop Up During Test***" configures the behavior of the step type during execution, if "Show" is selected, a Popup will appear showing the recognition process. The duration of the Popup is defined in the Configuration File at same level of Packed Library (*i.e.*: C:\IPSES_Lib\ABB\EqMeters\HMI\Config_ImageTesting.txt").

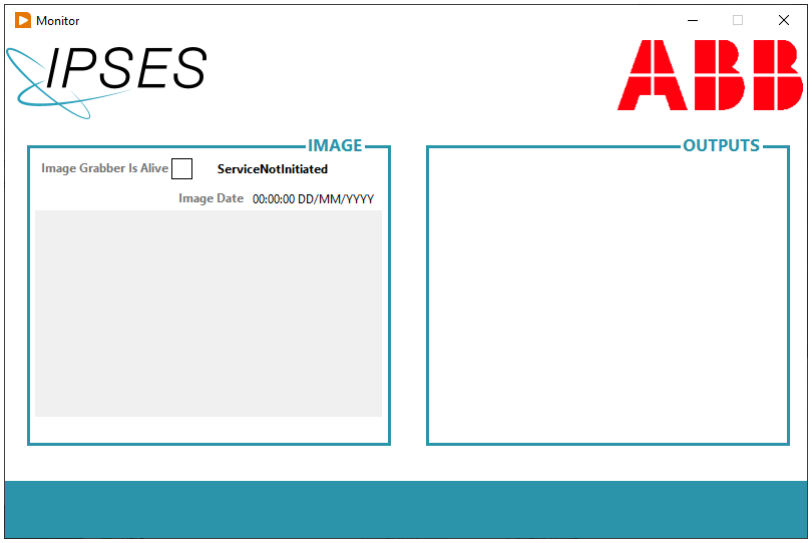
After execution, this step is marked as DONE and numbers recognized are reported in a dedicated string, under Step Type Result: string name is "NumberRecognized" and numbers format are "XXX.X.X. XXX.X" for D11 and "XXXX.X.X.X XX.X.X.X" for D13. Numbers are also reported in a dedicated array named "NumbersArray" of 2 elements. An array of booleans "NumbersFound" indicates if numbers recognized are complete and can be interpreted as actual numbers.

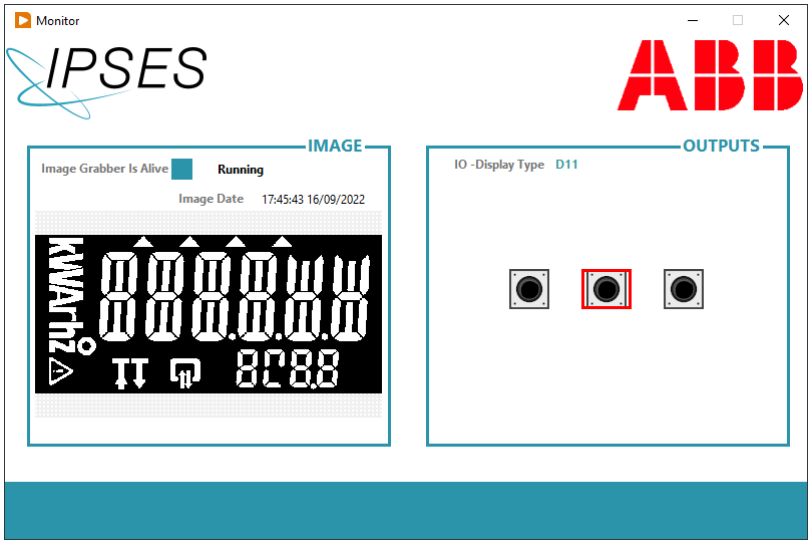
The Popup that could be shown during execution has the following appearance:

3.7 Image Monitor Init

This Step allows the operator to have general view of the executions on the system and it has no testing functionality.

It has no Edit phase. During execution it shows a Monitor Panel giving information about "Image Grabber" and "Simulation Typing" services. Water Green LEDs indicate that image service is correctly running. The picture shows in real time last complete image sniffed (Image Date is the related date). Red frames appearing on button pictures indicate in real time which kind of typing simulation is executed by the "Typing Simulation" service. The buttons are visible only when typing service is alive and "IO – Display Type" indicates device model configured for execution.





3.8 Image Monitor Close

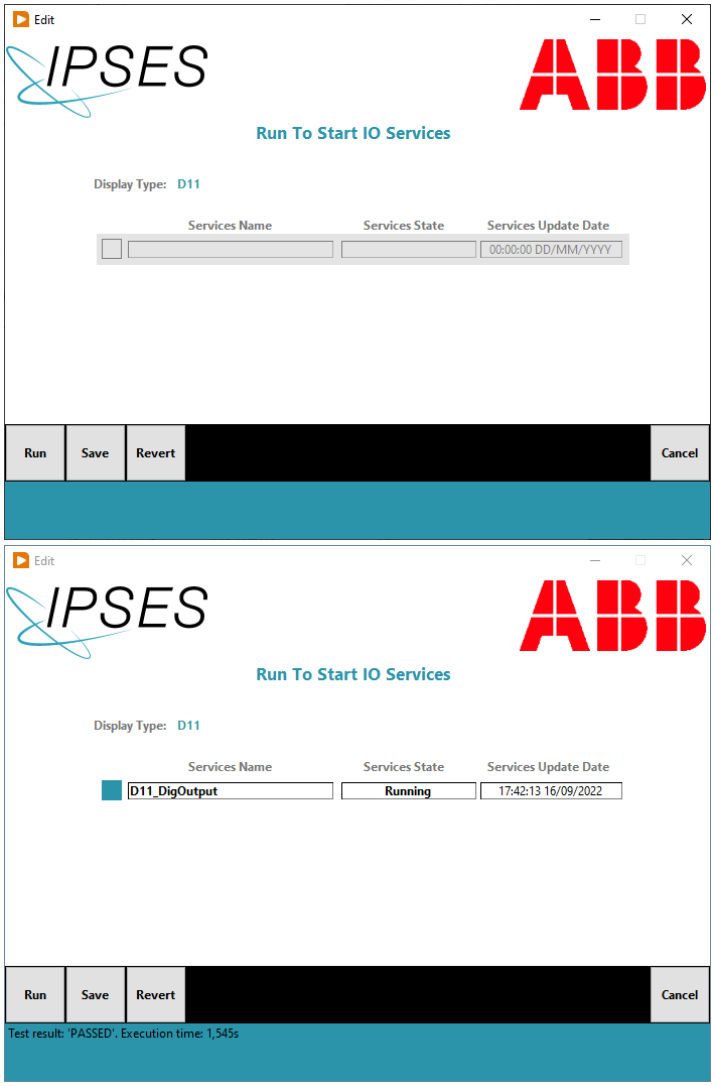
This Step Type has no Edit phase and its execution disposes the Monitor Panel.

4 CUSTOM STEP TYPES EQ METERS HMI (SimulTyping)

In this chapter the step types for simulating key typing on the physical buttons present on DUTs.

Simulation of physical keyboard carries software-defined digital signals to the chipset, emulating the signal generated by an operator when pressing a physical key.

4.1 Typing Init



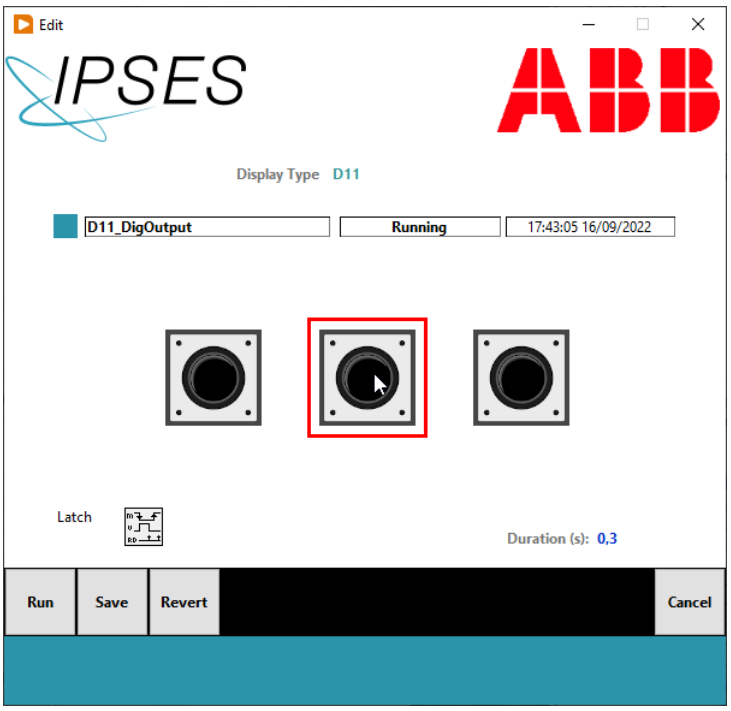
This Step Type allows to define DUT model and to generate digital signals that emulate the keypress. "***Display Type***" configures the DUT model from a drop-down menu.

Service Name indicates name of "Typing Simulation" service currently running (water green LED is lighted if so).

Service Update Date indicates the date of the last service status update.

Service State indicates the state of the service.

4.2 Typing Execute

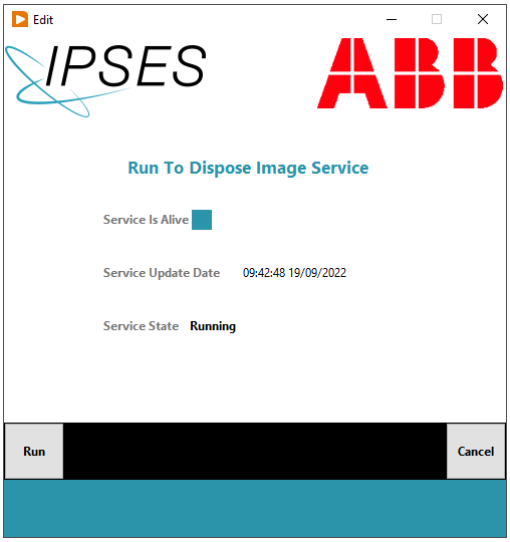


This Step Type allows to choose which key typing will be simulated at the RUN and the type of interaction ("Latch" with "Duration (s)" or "Switch" in which an uninterrupted button press is simulated until a different RUN is executed). The red frame visible on key of the physical keyboard, highlights on which key/s the simulated typing will take place

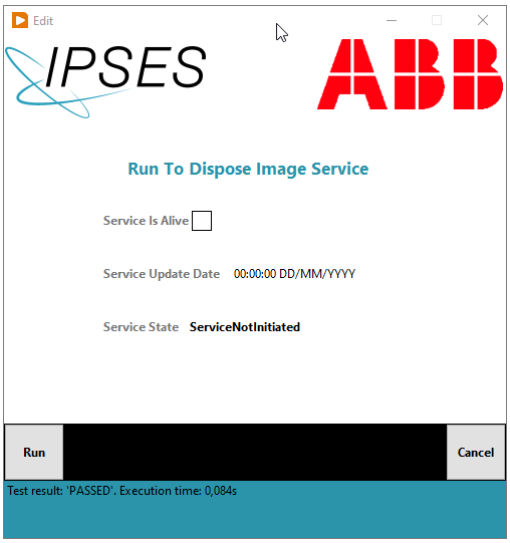
Action Type control for defining the type of interaction ("Latch" or "Switch") with the selected keys

Duration (s) control for defining the time in seconds of the "Latch" interaction if selected

4.3 Typing Close



This Step Type has no parameters in the Edit panel. It is used to dispose the "Typing Simulation" service. This action can be done during execution or by clicking on the RUN button during the Edit phase.

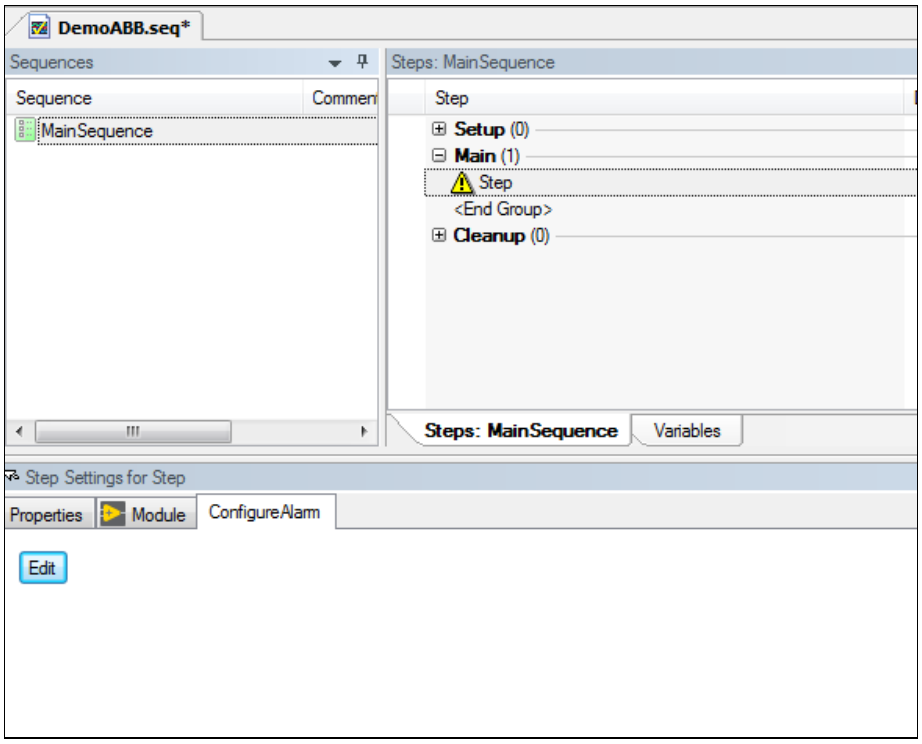


5 USE OF STEP TYPES IN TESTSTAND

6.1 Custom Step Types Introduction

The Step Types described above are usable in the TestStand development environment once imported through the configuration files (installation). They can be inserted in the sequence by drag and drop (just like the default TestStand steps) and they are placed in Insertion Palette under the respective folders

These Step Types also have an additional feature: the presence of the "Edit" panel. Such panel aiming to configure the execution of the step, is accessible by clicking on the Edit button (see figure).



In the Edit panel it is possible to configure the inputs of the code module that will be executed with the **RUN** command or during sequence running.

When parameters are configured and saved in the Edit panel, the property's default values are changed, in so doing new values are preserved even after the sequence is closed.

6 SYSTEMS REQUIREMENTS

The following system requirements are recommended. Failure to comply with these requirements may result in improper step types behavior.

6.1 Software Requirements

	LabVIEW 2020 32bit	**LabVIEW 2020** 64bit
Run-time LabVIEW	2020 SP1	2020 SP1
TestStand	2020	2020
VISA	NI – VISA version 18.5 o higher	
Vision	Vision Development Module Runtime	

6.1.1 Requirements for LabVIEW Run-Time Engine

Processor	Pentium 4M/Celeron 866 MHz (or equivalent) or later (32-bit) Pentium 4 G1 (or equivalent) or later (64-bit)
RAM	256 MB
Video card resolution	1024 x 768 pixels
Operating System	Windows 10/8.1/8/7 SP1 (32- and 64-bit) Windows Server 2012 R2 (64-bit) Windows Server 2008 R2 SP1 (64-bit)
Disk Space	620 MB

6.2 Hardware Requirements

Digital lines dedicated to segments activations are acquired by NI DIO device.

simulation will be controlled by NI DIO device.

6.2.1 Connections

Colored cables connections for D13:

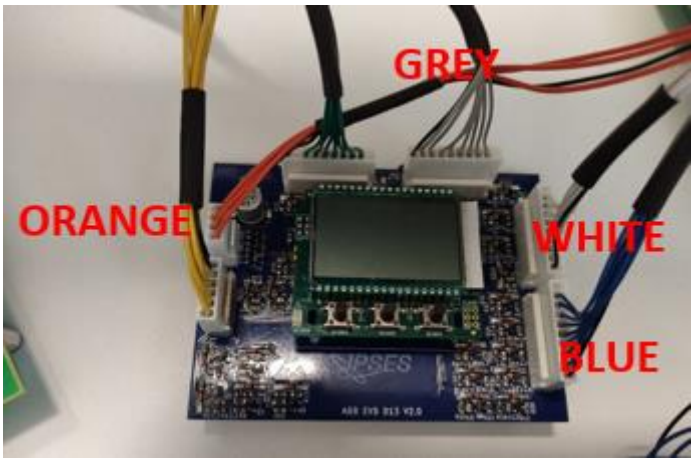


Figure 1: D13 Connections

Same cables (blue, white, grey and orange) are used for D11 as reported in picture:

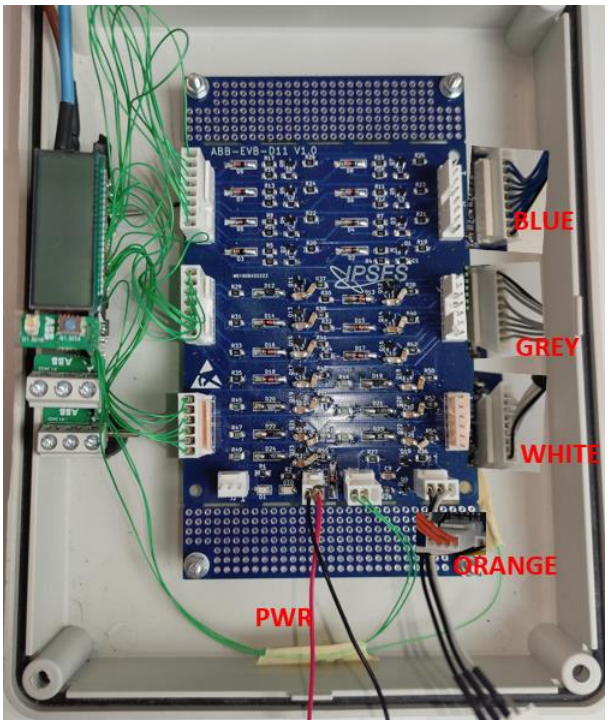


Figure 2: D11 Connections

Since number of pins does not match D11 male connector, use following table for right connections:

Table 1: D11 connections positions

	D13		D11	
DI	J4	blue	J4	mass in position 9 _not connected
	J5	white	J10	mass in position 9 _not connected
	J6	grey	J8	mass in position -1 _not connected
DO	J8	orange	J7	mass in position 3 _connected
PWR			J3	J3_1 -> PXI connector 0_ pin 14 (+5V)
				J3_2 -> PXI connector 0_ pin 15 (GND)

It is worth remember here that lines definition is statically configured in dedicated files:

- "Sniffer_DAQmx_Configuration.txt" as follow

  [ConfigData]

  D13_NumOfSamplesPerCh = "10000"
  D11_NumOfSamplesPerCh = "10000"

  D13_MinNumOverlapPnts = "60"
  D11_MinNumOverlapPnts = "60"

  PersistenceTime(s) = "0,000000"
  RefreshTime(s) = "0,000000"
  SampleClockSource = "/PXI-6289/100kHzTimebase"
  SampleClockRate = "0"

  D13_lines_Array.<size(s)> = "36"
  D13_lines_Array 0 = "PXI-6289/port0/line0"
  D13_lines_Array 1 = "PXI-6289/port0/line1"
  D13_lines_Array 2 = "PXI-6289/port0/line2"
  D13_lines_Array 3 = "PXI-6289/port0/line3"
  D13_lines_Array 4 = "PXI-6289/port0/line4"

D13_lines_Array 5 = "PXI-6289/port0/line8"
D13_lines_Array 6 = "PXI-6289/port0/line9"
D13_lines_Array 7 = "PXI-6289/port0/line10"
D13_lines_Array 8 = "PXI-6289/port0/line11"
D13_lines_Array 9 = "PXI-6289/port0/line12"
D13_lines_Array 10 = "PXI-6289/port0/line13"
D13_lines_Array 11 = "PXI-6289/port0/line14"
D13_lines_Array 12 = "PXI-6289/port0/line15"

D13_lines_Array 13 = "PXI-6289/port0/line16"
D13_lines_Array 14 = "PXI-6289/port0/line17"
D13_lines_Array 15 = "PXI-6289/port0/line18"
D13_lines_Array 16 = "PXI-6289/port0/line19"
D13_lines_Array 17 = "PXI-6289/port0/line20"
D13_lines_Array 18 = "PXI-6289/port0/line21"
D13_lines_Array 19 = "PXI-6289/port0/line22"
D13_lines_Array 20 = "PXI-6289/port0/line23"

D13_lines_Array 21 = "PXI-6289/port1/line0"
D13_lines_Array 22 = "PXI-6289/port1/line1"
D13_lines_Array 23 = "PXI-6289/port1/line2"
D13_lines_Array 24 = "PXI-6289/port1/line3"
D13_lines_Array 25 = "PXI-6289/port1/line4"
D13_lines_Array 26 = "PXI-6289/port1/line5"
D13_lines_Array 27 = "PXI-6289/port1/line6"
D13_lines_Array 28 = "PXI-6289/port1/line7"

D13_lines_Array 29 = "PXI-6289/port2/line1"
D13_lines_Array 30 = "PXI-6289/port2/line2"
D13_lines_Array 31 = "PXI-6289/port2/line3"
D13_lines_Array 32 = "PXI-6289/port2/line4"
D13_lines_Array 33 = "PXI-6289/port2/line5"
D13_lines_Array 34 = "PXI-6289/port2/line6"
D13_lines_Array 35 = "PXI-6289/port2/line7"

D11_lines_Array.<size(s)> = "21"
D11_lines_Array 0 = "PXI-6289/port0/line15"
D11_lines_Array 1 = "PXI-6289/port0/line14"

D11_lines_Array 2 = "PXI-6289/port0/line13"
D11_lines_Array 3 = "PXI-6289/port0/line12"
D11_lines_Array 4 = "PXI-6289/port0/line11"
D11_lines_Array 5 = "PXI-6289/port0/line10"
D11_lines_Array 6 = "PXI-6289/port0/line9"
D11_lines_Array 7 = "PXI-6289/port0/line8"

D11_lines_Array 8 = "PXI-6289/port0/line21"
D11_lines_Array 9 = "PXI-6289/port0/line20"
D11_lines_Array 10 = "PXI-6289/port0/line19"
D11_lines_Array 11 = "PXI-6289/port0/line18"
D11_lines_Array 12 = "PXI-6289/port0/line17"
D11_lines_Array 13 = "PXI-6289/port0/line16"

D11_lines_Array 14 = "PXI-6289/port1/line0"
D11_lines_Array 15 = "PXI-6289/port1/line1"
D11_lines_Array 16 = "PXI-6289/port1/line2"
D11_lines_Array 17 = "PXI-6289/port1/line3"
D11_lines_Array 18 = "PXI-6289/port1/line4"
D11_lines_Array 19 = "PXI-6289/port1/line5"
D11_lines_Array 20 = "PXI-6289/port1/line6"

- "IO_Manager_Configuration.txt"

[D13_DigOutput]
MonitorTimeout(ms) = "1000"
DigOutputLines.Key0 = "PXI-6289/port0/line26"
DigOutputLines.Key1 = "PXI-6289/port0/line25"
DigOutputLines.Key2 = "PXI-6289/port0/line24"

[D11_DigOutput]
MonitorTimeout(ms) = "1000"
DigOutputLines.Key0 = "PXI-6289/port0/line26"

Additional info regarding D11 line connections are summarized in following table:

Table 2: D11 lines definition table

DISPLAY		sp	BOARD	pin	sp	PXI	line		pin	IDX
21	COM8		J1	8		port 0	8	connector 1	52	7
20	COM7			7			9		17	6
19	COM6			6			10		49	5
18	COM5			5			11		47	4
17	COM4			4			12		19	3
16	COM3			3			13		51	2
15	COM2			2			14		16	1
14	COM1			1			15		48	0
6			J9	6		port 0	16	connector 1	11	13
5				5			17		10	12
4				4			18		43	11
3				3			19		42	10
2				2			20		41	9
1				1			21		6	8
13			J6	7		port 1	6	connector 0	5	20
12				6			5		6	19
11				5			4		41	18
10				4			3		42	17
9				3			2		43	16
8				2			1		10	15
7				1			0		11	14

7 CONTACTS

IPSES *S.r.l.* deals with the design and marketing of electronic and scientific instruments. The **custom design** allows to respond to the different needs of those who search embedded systems dedicated to specific applications.

IPSES employs a staff with many years of experience in the field. The continuous updating and the constant evolution make **IPSES** a cutting-edge company, able to combine the dynamism of a young company with the professionalism and reliability of qualified personnel.

IPSES S.r.l. is SCAI Connect partner.

IPSES S.r.l.

Headquarters and development center:

Viale Carlo Forlanini, 44A

SISTEMA DI GESTIONE CERTIFICATO ITALCERT UNI EN ISO 9001

20024 Garbagnate Milanese (MI) Italy

tel. (+39) 02 39449519 - (+39) 02 320629547

fax (+39) 02 700403170

e-mail: info@ipses.com

http://www.ipses.com

SCAI CONNECT

SCAICONNECT S.r.l.

Via Francesco Gentile 135

00173 ROMA Italy

e-mail: info@scaiconnect.it

tel. (+39) 06 445 7180

TECHNICAL SUPPORT INFORMATION

Our technicians can be contacted at the following:

Telephone : (+39) 02 39449519

(+39) 02 320629547

Fax : (+39) 02 700403170

Email : support@ipses.com

PROBLEM REPORT

The form at following page allows to collect the necessary data for correct research of a problem eventually highlighted.

8 ENGINEERING PROBLEM REPORT

Problem describer

Name			IPSES S.r.l. Viale Carlo Forlanini, 44A Garbagnate Milanese (MI) Italy Fax ++39 02/700403170 e-mail *support@ipses.com*
Company			
Date	Tel.	Fax	

Product

Name	Version	Serial No.

Report Type (bug, change request or technical problem)

Major bug	☐	Urgency:	
Minor bug	☐	High	☐
Change request	☐	Medium	☐
Technical problem	☐	Low	☐

Problem Description

Reproduction of Problem

IPSES s.r.l. Action notes

Received by	Date	Report No.	Action

IPSES S.r.l.

Viale Carlo Forlanini, 44A

20024 Garbagnate Milanese (MI) - ITALY

Tel. (+39) 02 39449519 – (+39) 02 320629547

Fax (+39) 02 700403170

e-mail: info@ipses.com

support@ipses.com