

**SCADAPack E ISaGRAF
Technical**



Documentation

Table of Contents

Part I ISaGRAF Technical	4
1 Technical Support.....	4
2 Safety Information.....	5
3 Preface.....	7
4 Overview & Terminology.....	9
4.1 ISaGRAF Workbench Software	11
4.1.1 Suggested ISaGRAF Options.....	12
4.1.2 ISaGRAF Serial Communications	13
4.1.3 ISaGRAF Port Command Line Shell.....	14
4.1.4 ISaGRAF Remote Access.....	15
4.2 SCADAPack EISaGRAF Target & Target Memory Usage	16
5 Target Scanning Cycle.....	17
5.1 Continuous Scanning	18
5.2 Cycle Time Scanning	19
5.2.1 Permanent & Temporary Cycle Timing	20
5.3 High Priority Scanning	21
6 RTU Logic Functionality.....	22
6.1 Input Scanning	23
6.2 Output Updates	24
6.3 Language Types & Conversion Tables	25
6.4 Data Types	26
6.5 Operators	27
6.6 I/O Locking	28
6.7 Message Variables & Conversion	30
6.8 Retained Variables	31
6.9 Array Variables	32
6.10 Second ISaGRAF Kernel Features	34
7 Application Management.....	35
7.1 On-line Modification	36
7.2 Application Storage	37
7.3 Storing a Project on the RTU	38
8 I/O Interfaces.....	40
8.1 Physical I/O	41
8.2 Derived Data	43
8.3 ISaGRAF I/O Boards	44
8.3.1 Digital Input & Output Boards, & Counter Input Boards.....	46
8.3.2 Analog Input & Output Boards.....	47
8.3.3 SCADAPack ER I/O Boards.....	48
8.3.3.1 Digital Output (er16do) & Input (er32di) Board.....	49
8.3.3.2 Analog Input (er16ai) & Output (er4ao) Board.....	50
8.4 Remote Control Interlock & Hold On Stop	51
8.5 PLC Device I/O Boards	52
8.5.1 Input Boards.....	54
8.5.2 Output Boards.....	55
8.5.3 Board Status.....	56

8.6	RTU Data via Function Blocks	63
9	ISaGRAF Variable / Point Interaction.....	64
10	ISaGRAF Analog I/O Boards / Representation & Conversion.....	65
11	ISaGRAF OPERATE Function.....	66
11.1	Using ISaGRAF Operate Functions with Conversion Tables	68
12	Remote ISaGRAF Access.....	69
12.1	Application File Transfer	70
12.2	DNP3 Communications	71
12.3	TCP/IP Communications	74
12.3.1	ISaGRAF Workbench Ethernet Settings.....	75
12.3.2	ISaGRAF TCP/IP Communications Server.....	76
13	ISaGRAF Troubleshooting & Error Codes.....	77
13.1	Error Types	78
13.2	Error Summary	79
13.3	Error Descriptions	82
14	ISaGRAF Unsupported Features.....	88

I ISaGRAF Technical



Documentation

©2013 Control Microsystems Inc.
All rights reserved.
Printed in Canada.

Version: 8.05.4

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

1 Technical Support

Support related to any part of this documentation can be directed to one of the following support centers.

Technical Support: The Americas

Available Monday to Friday 8:00am – 6:30pm Eastern Time

Toll free within North America 1-888-226-6876

Direct Worldwide +1-613-591-1943

Email TechnicalSupport@controlmicrosystems.com

Technical Support: Europe

Available Monday to Friday 8:30am – 5:30pm Central European Time

Direct Worldwide +31 (71) 597-1655

Email euro-support@controlmicrosystems.com

Technical Support: Asia

Available Monday to Friday 8:00am – 6:30pm Eastern Time (North America)

Direct Worldwide +1-613-591-1943

Email TechnicalSupport@controlmicrosystems.com

Technical Support: Australia

Inside Australia 1300 369 233

Email au.help@schneider-electric.com

2 Safety Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

	The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.
	This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

CAUTION

CAUTION used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage..

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and the installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ CAUTION**EQUIPMENT OPERATION HAZARD**

- Verify that all installation and set up procedures have been completed.
 - Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
-

- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and grounds, except those grounds installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove ground from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

3 Preface

Purpose

The purpose of this document is to describe in detail the interface between SCADAPack E RTU's and the ISaGRAF target software.

It is assumed that the reader is familiar with the ISaGRAF workbench software. For more information

about the ISaGRAF program environment, the see the *ISaGRAF Workbench Users Guide* which contains the *ISaGRAF Language Reference and ISaGRAF Target User's Guide*.

Additional programming options specific to the SCADAPack E controllers can be found in the *SCADAPack E ISaGRAF Function Blocks Reference* manual.

Note that the ICS / Triplex ISaGRAF Target User's Guide included with the ISaGRAF Workbench manual is not dedicated to any particular ISaGRAF target and describes general concepts. The material covered in this manual together with the *SCADAPack E ISaGRAF Function Blocks Reference* manual therefore supplements the *ISaGRAF Target User's Guide* for it deals exclusively with the implementation of the ISaGRAF Target kernel on a Schneider Electric SCADAPack E processor.

Specific technical reference information pertaining to ISaGRAF Function Blocks used with SCADAPack E RTU's can be found in the *SCADAPack E ISaGRAF Function Blocks Reference* manual.

Assumed Knowledge

Familiarity with the ISaGRAF Workbench is strongly recommended.

Target Audience

- Systems Engineers
- Commissioning Engineers
- Maintenance Technicians

References

- SCADAPack E Configuration Reference Manual
 - ICS / Triplex ISaGRAF Manual set
-

4 Overview & Terminology

Overview

This document describes the interface between SCADAPack E RTU operating and ISaGRAF target software.

ISaGRAF provides IEC61131-3 international standard PLC programming capability to the SCADAPack E RTU.

The ISaGRAF target kernels execute on the RTU and are built in to the operating system firmware.

The ISaGRAF Workbench executes on a PC and provides application generation, transferring and debugging through connection to the target by serial or network communications.

Highlights

By way of summary, the following facilities are notable ISaGRAF facilities, or are provided in addition to the standard ISaGRAF facilities when used on SCADAPack E controllers:

- Load applications to RTU via ISaGRAF Workbench, via File Transfer, remotely via DNP3, remotely via IEC60870-5-101 & -104 protocol
- Remote debugging of ISaGRAF applications via DNP3 links, TCP/IP
- Dual ISaGRAF target kernels for executing 2 separate ISaGRAF projects simultaneously
- Download ISaGRAF application project source to the RTU, for later upload
- Online application logic modification
- Scan control prioritization for improved performance or high accuracy scan intervals
- Interaction between ISaGRAF applications and RTU operating system facilities
- Interaction between ISaGRAF applications and RTU point database attributes
- ISaGRAF interaction with RTU database floating point numeric and string data types
- ISaGRAF application control over data exchange with peer RTU nodes
- ISaGRAF application control over trend sampling
- ISaGRAF application file management
- ISaGRAF application custom communication protocol management
- ISaGRAF interface to common PLC and peripheral device protocols

Terminology

The SCADAPack E RTU uses ISaGRAF target software within the RTU's multi-tasking environment.

- [ISaGRAF Workbench Software](#)^[11]
- [SCADAPack E ISaGRAF Target & Target Memory Usage](#)^[16]

- [Target Scanning Cycle](#)¹⁷⁾
-

4.1 ISaGRAF Workbench Software

An ISaGRAF application can be loaded, configured, debugged on the SCADAPack E RTU from a standard PC running the ISaGRAF Workbench if there is a connection between the PC and the RTU. Serial and Ethernet / TCP/IP communication facilities are provided by the RTU to allow different options of connection from a standard PC. The SCADAPack E controllers also support connection of ISaGRAF remotely via a DNP3 communication channel.

The ISaGRAF Workbench software is used to create, manage and simulate IEC61131-3 ISaGRAF applications.

The ISaGRAF Workbench debugger transfers ISaGRAF application programs to the target and provides application-debugging facilities.

ISaGRAF Workbench communication with SCADAPack E RTUs is supported for Windows 2000 and XP. For more information on the ISaGRAF Workbench, refer to the *ISaGRAF Workbench User's Guide*.

ISaGRAF Workbench can connect to an SCADAPack E RTU in a number of ways:

- ISaGRAF local serial communications - up to 2 serial port workbench connections are simultaneously supported
- ISaGRAF Ethernet communications - a single TCP/IP workbench connection is supported
- ISaGRAF - DNP3 communications - a single connection via any DNP3 interface is supported (serial, Ethernet, USB)

The above connections are supported in any combination to both RTU ISaGRAF target kernels. Only one active connection is available to a particular target at a time.

The following sections describe the various connection options that are available between the ISaGRAF Workbench and SCADAPack E RTU's:

- [ISaGRAF Serial Communications](#)^[13]
- [ISaGRAF Port Command Line Shell](#)^[14]
- [ISaGRAF Remote Access](#)^[15]

4.1.1 Suggested ISaGRAF Options

The ISaGRAF Workbench provides a number of options that it is suggested be enabled for convenience and improved operation with SCADAPack E RTUs.

The options are defined in the **ISA.INI** file, typically located in the ISaGRAF installation folder:

```
C:\ISAWIN\ESeries\EXE
```

It is suggested that the following parameters be updated in the **ISA.INI** file. In general, if the [section-name] already exists in the file, replace a similar entry with the one shown here, or add the section name and/or entry if they aren't present.

```
[ WSARK]
Path=C:\ISAWIN\ESeries\ARK
```

```
[ WSLB1EDT]
Comments=1
```

```
[ WSCN1EDT]
ChannelNumber=0
```

```
[ DEBUG]
ARRAYS=1
```

The [WSARK] path parameter sets the default path for the ISaGRAF Library archive facility used to backup and restore library components. Using this or another similar path is more convenient than the default ISaGRAF path of A:\ (floppy disk drive)

The [WSLB1EDT] comments parameter causes the ISaGRAF Library Editor to show comments for the various library elements simplifying management.

The [WSCN1EDT] channel number parameter causes the ISaGRAF Connection Editor to start its channel numbering at 0 by default instead of 1. This is useful if RTU point numbering starting at zero is preferred (e.g. when connecting ISaGRAF to point addresses used for DNP3 protocol).

The [DEBUG] arrays parameter allows the ISaGRAF workbench ladder and connection editors to support Arrays. For more information on using arrays see section [Array Variables](#)^[32].

Restart the ISaGRAF Workbench software, if necessary, for the changes to take effect.

4.1.2 ISaGRAF Serial Communications

The ISaGRAF Workbench debugger operates with the SCADAPack E RTU's on an RTU serial port configured as "ISaGRAF". Operation is supported for RS232, RS422 and RS485.

TIP: A maximum of one RTU serial port may be configured as "ISaGRAF" for local connection to an ISaGRAF Workbench debugger. If a second ISaGRAF Workbench connection is desired, configure a separate RTU serial port as "ISaGRAF 2" or use ISaGRAF via Ethernet.

An ISaGRAF application previously loaded into the RTU (via Workbench or SCADAPack E Configurator) will execute on the RTU regardless of whether an "ISaGRAF" port is configured.

The "ISaGRAF" port on an RTU can also be used for other purposes:

- ISaGRAF "Diagnosis/Monitoring Tool"
- RTU Command Line and Diagnostics "Shell".

4.1.3 ISaGRAF Port Command Line Shell

ISaGRAF Port Command Line Shell

Operation of the Command Line Shell allows an ASCII terminal to be connected to the RTU "ISaGRAF" port.

Press <Enter> <Enter> <Enter> from an ASCII terminal on the ISaGRAF port to activate the command line shell. The terminal communications settings need to match those on the RTU's "ISaGRAF" port. Once activated, the RTU Command Line prompt is displayed, and ISaGRAF Workbench Communications are disabled on this port. RTU diagnostics and command line facilities are available. This functionality is the same as that provided by a configured "Cmd Line" port, without re-configuring the RTU.

To return from Command Line Shell back to ISaGRAF Workbench communications, enter the command line "**BYE**" command and disconnect the ASCII terminal.

For more information on the Command Line RTU facilities see the *SCADAPack E Operational Reference* manual.

4.1.4 ISaGRAF Remote Access

The SCADAPack E RTU provides various remote access facilities for ISaGRAF. These are:

- ISaGRAF application file transfer
- ISaGRAF application file transfer via FTP
- ISaGRAF workbench communication via DNP3
- ISaGRAF workbench communication via TCP/IP

Each of these facilities is further described in Section [Remote ISaGRAF Access](#)⁶⁹.

4.2 SCADAPack E ISaGRAF Target & Target Memory Usage

SCADAPack E ISaGRAF Target

The SCADAPack E RTU is equipped with ISaGRAF target kernel software. This allows the RTU to perform PLC control functions using the IEC 61131-3 international standard. The control functions provided by ISaGRAF targets are completely autonomous of any supervisory (SCADA Master) system or communications network (e.g. DNP3). The ISaGRAF application operates on the RTU regardless of the state of remote communications. The SCADAPack E ISaGRAF target version is 3.40 and supports ISaGRAF Workbench versions 3.20 and later.

The SCADAPack E RTU firmware supports the simultaneous execution of up to **two ISaGRAF target kernels** on the same RTU. This allows up to two independent ISaGRAF applications to execute simultaneously on the same RTU. The two ISaGRAF targets within the RTU have fixed ISaGRAF Slave addresses of 1 and 2 respectively. The Workbench "Slave Number" communications parameter needs to be set to match the SCADAPack E RTU's ISaGRAF Slave address for the appropriate target kernel prior to connecting the ISaGRAF Workbench debugger to the RTU. Both ISaGRAF targets are activated by default, i.e. there is no additional configuration required to activate the second ISaGRAF target.

The status (running or halted) of an ISaGRAF application on each target kernel can be obtained from the SCADAPack E Configurator software or via separate *ISaGRAF Application Halted* system binary points 50100 and 50101. These binary points are part of the system points dedicated to providing the status of the RTU. . For more information see the *SCADAPack E Operational Reference Manual*.

The SCADAPack E RTUs are based on Intel and ARM processors. The ISaGRAF Workbench software generates target independent code (TIC) for Intel, Motorola and PC Simulation. To use ISaGRAF applications with the SCADAPack E RTU, ISaGRAF Workbench Make-Compile Options should be selected for "**ISA86M: TIC Code for Intel**".

ISaGRAF Workbench Application Symbols are not required for the SCADAPack E RTU.

Target Memory Usage

The SCADAPack E RTU is equipped with FLASH memory, which contains the RTU operating system, ISaGRAF target kernel engine and telemetry communications software. The RTU is also equipped with battery-backed System Static for storage and execution of user ISaGRAF applications.

See Section [Application Storage](#)^[37] for details on application storage.

RTU system RAM is also available for facilities such as Online ISaGRAF application modification, RTU communication buffer storage, file system support, RTU SCADA event data and historical sample storage, etc.

The that 4K of static RAM is also available for ISaGRAF retained variables. See Section [Retained Variables](#)^[31] for details.

5 Target Scanning Cycle

The ISaGRAF target executes user applications in a cyclic fashion. As such, user applications are also designed to execute in a cyclic fashion. As shown in [Figure 4.1](#)^[17], the target cycle has distinct phases. These include obtaining inputs to the ISaGRAF application (via Input Boards), execution of various parts of the ISaGRAF user application, and updating of outputs (via Output Boards).

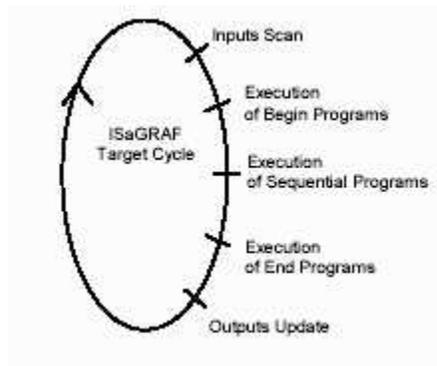


Figure 4.1: Target Scanning Cycle

The ISaGRAF target executing on an SCADAPack E RTU can be configured to scan in various ways:

- [Continuous Scanning](#)^[18]
- [Cycle Time Scanning](#)^[19]
- [High Priority Scanning](#)^[21]

5.1 Continuous Scanning

This is the default-scanning mode that is used by ISaGRAF applications. At the end of an ISaGRAF target scan, the target simply restarts the scanning cycle. Typically, the SCADAPack E RTU has many activities it needs to perform in addition to scanning the user's ISaGRAF application. As such, the ISaGRAF application may be interrupted for short periods throughout the scan cycle. This can result in some variations in the time it takes ISaGRAF to perform each scan. If the RTU becomes unusually busy performing these activities, this can result in slower execution of the user's ISaGRAF application.

The ISaGRAF Debugger indicates "**allowed=0**" for the Continuous Scanning mode of operation.



Figure 4.2: ISaGRAF Debugger Window

It is recommended that, where possible, ISaGRAF applications use continuous scanning mode and be designed to take into account variations in the timing of the scan rate.

5.2 Cycle Time Scanning

This scanning mode can be either permanently set for an ISaGRAF application or temporarily set. Instead of the ISaGRAF target restarting the scanning cycle immediately after updating the outputs, as in the default continuous mode, the scanning process is stopped for a certain time interval. This time interval, which makes up the difference between the actual time of a complete scan and a fixed cycle time, is now available for the RTU to perform activities that previously may have interrupted the scanning cycle. As such, the ISaGRAF application will tend to have more consistent cycle scan times.

The SCADAPack E RTU can still schedule other system operations in the middle of an ISaGRAF scan if necessary. This is particularly the case for larger ISaGRAF applications. Smaller ISaGRAF applications will not normally be interrupted by routine RTU activities.

The ISaGRAF Debugger indicates a non-zero “**allowed**” time when *Cycle Time Scanning* mode is in use as shown in the figure below.



Figure 4.3: ISaGRAF Debugger Window

The ISaGRAF Debugger will report “overflow” errors if the time taken to perform an individual scan is higher than the fixed (allowed) time. In this case the fixed cycle time should be increased.

The following sections describe methods for permanently or temporarily setting Cycle Timing scanning for an ISaGRAF application. ISaGRAF also supports changing this mode from within a user application. See the “SYSTEM” function within the *ISaGRAF Language Reference* manual for this implementation. Using this facility overrides both Permanent Cycle Timing and Temporary Cycle Timing as described in [Permanent & Temporary Cycle Timing](#)^[20].

5.2.1 Permanent & Temporary Cycle Timing

Permanent Cycle Timing

To permanently set an ISaGRAF application cycle time from the Workbench, enable *Trigger Cycles* from the *Application Run Time Option dialog* under the *Make* menu and enter a value in the *Cycle Timing* field. This needs to be done before the application is compiled.

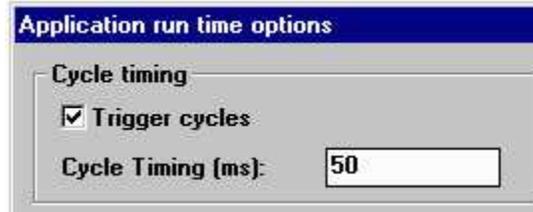


Figure 4.4: Application Run Time Option Dialog

To change the ISaGRAF scanning mode from cycle timing back to continuous scanning, disable *Trigger Cycles* and recompile the ISaGRAF user application.

Temporary Cycle Timing

To temporarily set or change an ISaGRAF application cycle time, select the ISaGRAF Debugger menu option *Control | Change Cycle Timing* menu or “Set Cycle Timing” icon .



Figure 4.5: Application Run Time Option Dialog

The Cycle Timing value entered is in milliseconds (ms).

To change the ISaGRAF scanning mode from cycle timing back to continuous scanning, set *Cycle Timing* to zero (0).

If the ISaGRAF application or the SCADAPack E RTU is restarted, a temporary cycle timing setting is lost. The cycle time will revert to the setting as compiled by the Workbench.

5.3 High Priority Scanning

The SCADAPack E RTU supports increasing the priority of an ISaGRAF target kernel task to achieve improved accuracy in the application cycle timing.

This is achieved by setting the “ISA_TASK_PRI” parameter using the *RTUPARAM* function block within the ISaGRAF application.

An ISaGRAF application can raise its own priority, subject to the following criteria:

- The user ISaGRAF application needs to be in “Cycle Timing” scanning mode. (See Section [Cycle Time Scanning](#)^[19] above), and
- Each ISaGRAF application cycle needs to scan in less time than the fixed *Cycle Timing* setting.

High priority scanning sets the ISaGRAF target kernel task to run at a higher priority than routine RTU operating system tasks. This is useful where highly accurate execution rates are required.

An ISaGRAF task executing with standard priority and Cycle Time Scanning will still achieve relatively accurate execution timing, suitable for typical applications. Schneider Electric recommends that High Priority Scanning only be used where absolutely necessary for the application as it can adversely affect the RTU's capacity to perform other tasks.

If multiple ISaGRAF target kernels are executing on the same RTU, it is not recommended for more than one of the tasks to operate in high priority scanning mode.

Where high priority scanning is required, it is suggested that user ISaGRAF applications be split into a high priority and low priority component. This is to minimize the amount of ISaGRAF code that runs at high priority and the otherwise resultant impact on other RTU operations, such as communication. Two ISaGRAF target kernels could be used to separately execute the low and high priority components in separate tasks. One of the tasks could then use the *RTUPARAM* function block to increase its priority.

For more information on the *RTUPARAM* function block see the *SCADAPack E ISaGRAF Function Block Reference* manual.

6 RTU Logic Functionality

- [Input Scanning](#)^[23]
 - [Output Updates](#)^[24]
 - [Language Types & Conversion Tables](#)^[25]
 - [Data Types](#)^[26]
 - [Operators](#)^[27]
 - [I/O Locking](#)^[28]
 - [Message Variables & Conversion](#)^[30]
 - [Retained Variables](#)^[31]
 - [Array Variables](#)^[32]
 - [Second ISaGRAF Kernel Features](#)^[34]
-

6.1 Input Scanning

Binary Inputs

The update of binary input states to an ISaGRAF application, either from RTU local I/O, remote I/O or other RTU database data, is synchronized with the scanning of the ISaGRAF application. ISaGRAF updates its input boards at the start of the scan cycle as described in Section [Target Scanning Cycle](#) [17]. The input variables that are updated from the I/O remain constant for the duration of one program scan. Physical RTU Digital inputs are represented as Binary Input Points in the RTU address space and are represented as *Boolean* variables in an ISaGRAF application.

Analog Inputs

The update of analog input values to an ISaGRAF application is similar to binary inputs. The updates are synchronized with the scanning of the ISaGRAF application. The SCADAPack E RTU's database supports 16-bit Signed Integer, 32-bit Signed Integer as well as 32-bit Floating-Point data points. Analog variables in ISaGRAF are represented as 32-bit Signed *Integer* or 32-bit Floating Point *Real* within the ISaGRAF environment. The RTU supports data conversion between the 16-bit and 32-bit analog data types. For more information see [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#) [65].

6.2 Output Updates

Binary Outputs

RTU binary output data is updated at the end of the scan cycle following changes made by the ISaGRAF application to its Boolean variables. See [Target Scanning Cycle](#)^[17] for details. RTU Digital outputs and RTU derived database points are represented as Binary Points in the RTU address space and Boolean variables in an ISaGRAF application.

By default, an executing ISaGRAF application with output variables connected to the physical I/O boards has control of Physical RTU Binary Outputs and Derived RTU points, unless a “Remote Interlock” is active for individual binary points. See [Remote Control Interlock](#)^[51] for more information.

See [Stopping an ISaGRAF Application](#)^[41] for a description of the effect on Binary Outputs when the ISaGRAF application is stopped.

Analog Outputs

The analog output data is updated at the end of an ISaGRAF scan similar to Binary Outputs. The RTU supports 16-bit Signed Integer, 32-bit Signed Integer and Floating Point data objects. Analog variables are represented as 32-bit Signed *Integer* or 32-bit Floating Point *Real* within the ISaGRAF environment.

The RTU also supports data conversion between the 16-bit and 32-bit analog data types. For more information see Section [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#)^[65].

By default an executing ISaGRAF application, with analog points on ISaGRAF Output Boards, has control of those Physical RTU Analog Outputs and Derived RTU points, unless a “Remote Interlock” is active for individual analog points. See [Remote Control Interlock](#)^[51] for more information.

See [Stopping an ISaGRAF Application](#)^[41] for a description of the effect on Analog Outputs when the ISaGRAF application is stopped.

6.3 Language Types & Conversion Tables

Language Types

The ISaGRAF target kernel tasks on the SCADAPack E RTU support the five IEC 61131-3 (1993) international standard sequencing languages. These are:

- SFC - Sequential Function Chart
- FBD - Function Block Diagram
- LD - Ladder Diagram
- ST - Structured Text
- IL - Instruction List

Control applications may be written in ISaGRAF using any combination of the above languages, and can be executed on either, or both, ISaGRAF target kernel tasks. In addition, ISaGRAF Workbench versions 3.30 and above support the 'FC – Flow Chart'.

Conversion Tables

ISaGRAF allows up to 127 look-up Conversion Tables, each containing up to 32 look-up points. An interpolated value is used for points not specifically defined within the lookup table. Conversion Tables can be applied to RTU physical Analog Input channels or derived RTU Analog points. The Conversion Table "curve" needs to be monotonically increasing or monotonically decreasing.

The SCADAPack E RTU allows combinations of integer and floating point values to be applied to conversion tables. The "Electrical" values of the conversion tables apply to the RTU I/O or Derived points, and the "Physical" values apply to internal ISaGRAF variable values.

6.4 Data Types

The SCADAPack E RTU supports the following ISaGRAF data types and corresponding Point values:

Table 5.1: Relationship between ISaGRAF and SCADAPack E Data

ISaGRAF Data	Database point representation	Data Format
Integer Analog	Analog point 16-bit or 32-bit analog value	Signed integers
Real Analog	Short floating point (Engineering) value	32-bit IEEE-754 standard
Boolean	Binary Point state	ON or OFF
Timer		1 ms counts
Message	Octet string points	Up to 255 characters strings

ISaGRAF integer analog variables can correspond to RTU 16-bit signed integers or 32-bit signed integer values of analog points. Real ISaGRAF analog variables correspond to 'short floating point' Engineering values of analog points (stored as standard IEEE-754 32-bit single precision floating point values). See Section [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#) ^[65] for information on data type conversion used by the RTU.

6.5 Operators

ISaGRAF application software supports a comprehensive array of built-in operators including functions for:

Table 5.2: ISaGRAF Supported Built in Operators and Functions

boolean operations	&, =1, >=1, and, or, f_trig, r_trig, rs, sema, sr, xor
analog operations	*, +, -, /, and_mask, ArCreate, ArRead, Arwrite, cmp, limit, max, min, mod, neg, not_mask, odd, or_mask, rand, rol, ror, shl, shr, stackint, xor_mask
data manipulation	1 gain, mux4, mux8, sel
type conversion	ana, ascii, boo, char, msg, real, tmr
comparison	<, <=, <>, =, >, >=
maths	abs, acos, asin, atan, cos, expt, log, pow, sin, sqrt, tan, trunc
string management	cat, day_time, delete, find, insert, left, mid, mlen, replace, right
timer control	tof, ton, tp,
control/signal handling	average, blink, derivate, integral, hyster, lim_alm, sig_gen
Counting	ctd, ctu, ctud
system access	operate, system

6.6 I/O Locking

ISaGRAF I/O locking provides the ability to lock inputs and outputs into a certain state, regardless of their true state. This provides a mechanism for freezing I/O in a fixed state, allowing maintenance to be performed while the ISaGRAF application is still executing.

WARNING

UNEXPECTED EQUIPMENT OPERATION

Evaluate the operational state of the equipment monitored and controlled by the SCADAPack E RTU prior to initializing the SCADAPack E RTU.

Failure to follow these instructions can result in death, serious injury or equipment damage.

SCADAPack 300E RTU's "**FORCE**" LED indicates that an ISaGRAF application has at least one I/O variable in a LOCK state. When no variables remain locked, the FORCE LED will turn off.

ISaGRAF I/O variables can be locked and unlocked individually through the ISaGRAF Workbench when connected online with the ISaGRAF debugger to the SCADAPack E RTU's target kernel.

The ISaGRAF debugger also provides a mechanism to unlock locked variables. From the ISaGRAF debugger window use the **Control | Unlock all IO variables** menu.

WARNING

Online firmware update will NOT preserve the controller's operational state prior to the upgrade, including Controller output states, ISaGRAF application state information, ISaGRAF I/O Lock states, etc.

Consider the impact on operating equipment when upgrading firmware.

[Figure 5.1](#)^[29] shows diagrammatically how inputs and outputs are isolated from the ISaGRAF variable database and ISaGRAF debugger when locked.

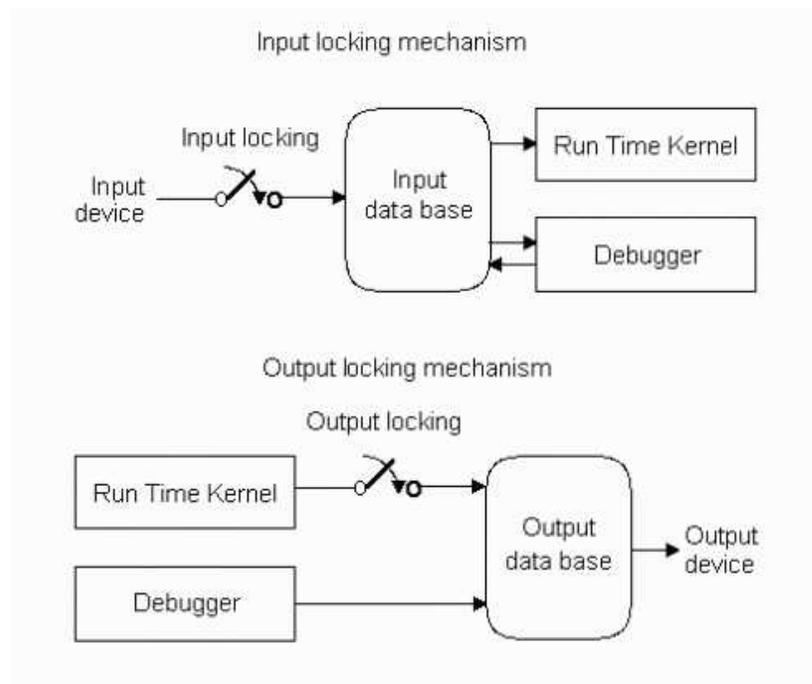


Figure 5.1: I/O Variable Locking

6.7 Message Variables & Conversion

Message Variables

Message variables provide a mechanism for manipulating groups of ASCII characters (strings). For example, ASCII characters can be read in from external equipment, processed by an ISaGRAF program, which may then build and output an appropriate ASCII reply.

Message variables also interface with the SCADAPack E RTU system string points. ISaGRAF message variables are limited to 255 characters in length.

Converting a Message to Integer Values

The **ANA** native ISaGRAF function can convert other ISaGRAF data types to an analog value. When **ANA** is used to convert a message data type to an integer, ISaGRAF does not automatically perform bounds checking on the input message value. Where a MSG message string has a value greater than 2^{31} , unexpected results may occur.

It is recommended that the user perform their own bounds checking on the message value if it is required to handle numbers above 2^{31} .

6.8 Retained Variables

Retained Variables

The SCADAPack E RTU provides a facility for storing ISaGRAF application variable values that are required to retain their value when power is lost, or when an ISaGRAF application is restarted. A copy of every ISaGRAF Boolean, analog, timer and message variable with the retain attribute is stored by the RTU, at the end of every ISaGRAF application scan (for the two ISaGRAF target kernel tasks), in non-volatile areas of RTU memory. After an ISaGRAF application restarts, retained variables for that application are refreshed with their last saved value. 4K of RTU system NV-RAM is set aside for retained variables, which is shared by the four standard variable types. The table below shows the memory used for each of the standard types.

Table 5.3: Memory used by standard ISaGRAF variable types

ISaGRAF Variable Type	RTU Memory Usage
Boolean	1 byte
Analog	4 bytes
Timers	4 bytes
Messages	256 bytes

The amount of memory allocated to each variable type is automatically allocated by the SCADAPack E RTU depending on the application requirements and the number of active applications in the RTU. The ISaGRAF Workbench *Application run time options* memory list is not required for the SCADAPack E to use *retained variables*. Non-volatile memory storage for retained variables is allocated at a fixed memory segment within the RTU. The total retained memory space requirements of an application needs to be less than 4KB or an ISaGRAF startup status code will occur (Status Code 8).

The SCADAPack E RTU clears retained variable values when an ISaGRAF application or RTU NV-RAM is cleared.

E.g. when a Cold Boot factory-defaults initialization occurs, if RTU firmware is changed through a local, manual firmware upgrade, or when using the command line "CLEAR ISaGRAF" command.

The ISaGRAF target kernels in the RTU automatically clear retained variables for the appropriate target kernel task when a new ISaGRAF application is loaded.

6.9 Array Variables

ISaGRAF support for Array Variables is not enabled on the ISaGRAF Workbench by default. See section [Suggested ISaGRAF Options](#)^[12] for information on enabling ISaGRAF Array variables.

Array variables are supported by SCADAPack E RTUs from firmware version 7.70, and ISaGRAF Workbench version 3.55.

Single Dimension Arrays

Arrays are created in the ISaGRAF dictionary. Arrays are available for Boolean, Integer, Real or Timer variables.

When Arrays are enabled on the ISaGRAF workbench, ISaGRAF dictionary variable declarations include a **Dim** field indicating the Dimension of the array (total number of elements).

If the Dim field is blank, single variables are declared. When a number is entered, this represents the number of elements in the array. The elements are numbered from 0 to Dim-1.

The maximum array dimension size that can be entered is 255, being array elements [0] to [254].

The screenshot shows a dialog box titled "Integer/Real Variable". It has several input fields and buttons. The "Name" field is filled with "myArrayName". The "Network Address" field is empty. The "Comment" field is empty. The "Unit" field is empty. The "Conversion" dropdown is set to "[none]". In the "Attributes" section, the "Internal" radio button is selected. In the "Format" section, the "Integer" radio button is selected, and the format dropdown is set to "[standard]". The "Initial value" field is filled with "0". The "Retain" checkbox is unchecked. The "Dim" field is circled in red and contains the value "25". On the right side, there are buttons for "Store", "Cancel", "Next", "Previous", and "Extended".

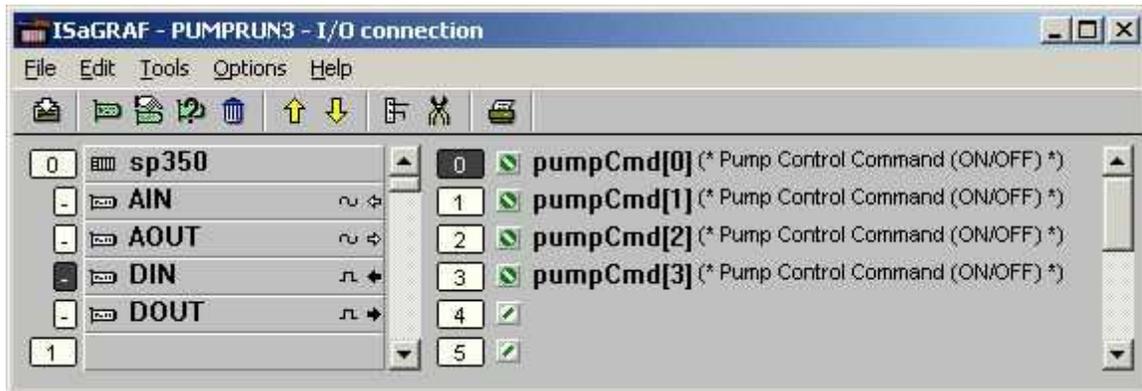
In this example myArrayName is an array with elements myArrayName[0] .. myArrayName[24]

CAUTION

Like many other Programming Languages, access of an ISaGRAF array variable in a program using an index number beyond the configured dimension can lead to serious unpredictable results including overwriting of data, reset of the RTU, etc.

Once an array is defined, the IEC notation `ArrayName[index]` may be used in any program or I/O board connection.

When arrays are input or output variables, the elements need to be connected on contiguous channels of the same I/O board.



When using ISaGRAF arrays, keep in mind:

- Functions and Function Blocks cannot have parameters of array type, but can have an element of the exact element (e.g. using notation `ArrayName[7]`)
- **TIP:** if the value of an array element indexed by a variable is required to be passed in to a function or function block, assign it to a local variable and pass the local variable in to the function or function block
- Array elements starts from element index 0
- Pass an array index number in to user functions and function blocks to access arrays internally
- ISaGRAF Debug mode only monitors the first array element (index 0)
- Arrays can't be initialized in the ISaGRAF Dictionary. **TIP:** write initial values using a FOR loop in a startup program
- Retained attributes only apply to the first array element. Copy the array elements to RTU database points to retain their values. The use of `getpntxx` and `setpntxx` functions and function blocks can be useful when used with arrays
- If Array elements are connected to an ISaGRAF I/O board, use the define "**Z_ArrayName**" as the I/O address of the first element of the array. Use the define "**ZHI_Array_Name**" as the value of the highest element index to avoid going off the end of the array

Sample Structured Text Code (assuming array x is connected to an I/O Board):

```
FOR i := 0 TO ZHI_Ains BY 1 DO
    IF xValue = Ains[ i ] THEN
        yValue := y[ i ];
        EXIT;
    END_IF
END_FOR;
```

6.10 Second ISaGRAF Kernel Features

A user ISaGRAF application executing on the RTU's second ISaGRAF kernel may open any RTU points on Input Boards, but cannot open output boards with points that are already on an output board used by the First ISaGRAF kernel.

Each of the RTU's ISaGRAF communication mechanisms may only communicate with one ISaGRAF Kernel task at a time, but will dynamically switch between them as requests are received from each. The address of the ISaGRAF Kernel to communicate with is configured in the Workbench *Debug Link-setup* menu.

It is recommended that if ISaGRAF PLC I/O boards are to be used in both kernels simultaneously, then they should be communicating to PLC devices through different RTU ports. It is possible for both ISaGRAF kernels to use the same port thereby trying to communicate out a single port to an RTU; however the ISaGRAF communications task may watchdog when attempting to download the application from the ISaGRAF workbench. A work-around for this is to use DNP3 communications to transfer and start the application rather than the ISaGRAF workbench. After the application has been restarted, it can be debugged through the Workbench.

ISaGRAF target addresses are fixed as "1" and "2" for the first and second kernels, respectively.

7 Application Management

The following sections describes facilities provided by ISaGRAF for application management

- [On-line Modification](#)^[36]
- [Application Storage](#)^[37]
- [Storing a Project on the RTU](#)^[38]

7.1 On-line Modification

On-line modification allows an ISaGRAF application running on the SCADAPack E RTU to be modified, without any discontinuity of execution. Use of this feature is subject to available RAM memory on the RTU and similarity of the new application with the executing application.

On-line application modification is achieved via the following mechanism. Firstly the modified ISaGRAF code is loaded into a new memory area and consistency checking is performed between the new applications data requirements and the original data. On completion of the current scan, control is then handed over to the modified code and execution continues. If further modifications are made, the code is loaded to another new memory area, and execution is handed to the new code on completion of the current scan.

Upon successful loading and activation, the new application is saved by permanently storing it in the RTU file system, making it the current ISaGRAF application for that ISaGRAF kernel task.

ISaGRAF will only accept a newer version of the running application code if the database is identical. This means that new variables, function block instances and changes to the IO board mapping are not allowed. While the ISaGRAF workbench does permit the application's name to be changed, the ISaGRAF debugger will not show any information about the renamed application. Changing the application name is not recommended.

7.2 Application Storage

The ISaGRAF application(s) loaded into the SCADAPack E RTU is stored in a file system partition on the RTU that is non-volatile. When power is restored after an outage, each stored application is validated by the RTU and a separate copy is made in non-volatile memory (NV-RAM) (for execution).

If the file system copy is successfully validated, this is copied to the NV-RAM area where execution commences, otherwise the NV-RAM area copy is checked for validity. If the NV-RAM copy is valid, the ISaGRAF application is copied back to the file system, then execution of the NV-RAM copy begins. If both copies are not valid, the ISaGRAF kernel task waits for a new application to be loaded and the *ISaGRAF Application Halted* system binary point for that ISaGRAF kernel target is activated (RTU system binary point 50100 or 50101).

Variables, other than those specified as ISaGRAF *retained variables*, are stored in volatile (non-saved) memory. ISaGRAF initializes their values when the application restarts.

The ISaGRAF applications are stored in the RTU file system in files called:

- **isa11**, and
- **isa21**

These are used for the **first** and **second** ISaGRAF target kernel tasks, respectively. These correspond to the “Slave” number of the ISaGRAF target kernel; address 1 and 2 respectively.

7.3 Storing a Project on the RTU

ISaGRAF provides the facility for storing the IEC61131-3 "source" on the target kernel along with the executing TIC code.

This facility allows the original code that generated the ISaGRAF application to be stored on the SCADAPack E RTU, and later uploaded to the ISaGRAF Workbench.

The ISaGRAF upload facility is supported through the following SCADAPack E ISaGRAF interfaces:

- via an RTU serial port configured as *ISaGRAF*
- via Ethernet when RTU has ISaGRAF/TCP service enabled (see SCADAPack E Configurator **TCP/IP** page)

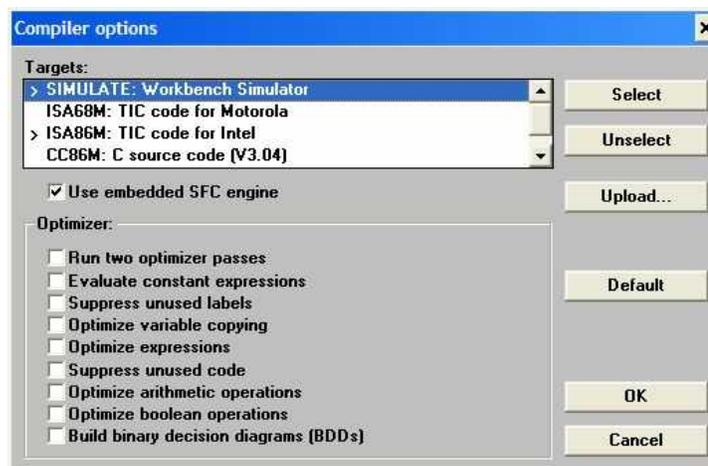
ISaGRAF project Upload facilities are not supported via DNP3 communications using SCADAPack E Configurator.

Downloading an ISaGRAF Project

The ISaGRAF project "source" can be downloaded along with the RTU TIC code. The ISaGRAF project is compressed by the ISaGRAF workbench and forms part of the ISaGRAF application downloaded to the RTU.

This applies to applications downloaded via the Workbench Debugger or via file transfer in an ISA11 or ISA21 file. (See [Remote ISaGRAF Access](#)⁶⁹).

To include ISaGRAF Project "source" with the compiled ISaGRAF application, set the following ISaGRAF compiler options. Open the ISaGRAF **Programs** window for the application project. Open **Make | Compiler Options** from the menu. Click on the **Upload** button as shown in the following figure.



In the **Prepare project for upload** dialog, tick **Embed source code for upload** as shown in the following figure and press **OK**.



When the ISaGRAF application is next compiled by the Workbench and downloaded to the RTU, it will store the project on the RTU for later retrieval by [Uploading](#)³⁹.

Uploading an ISaGRAF Project

The ISaGRAF **Project Management** window's **File** menu provides an **Upload project** item. It shows the dialog in the following figure.



Press the **Setup** button to choose the appropriate SCADAPack E ISaGRAF **Target Slave Number** and PC serial port or Ethernet connection for the Upload.

The **Configurator** option is not supported for ISaGRAF project upload.

When the **Run** button is pressed, the ISaGRAF workbench will upload the project and add it to the Workbench project list. The user may be prompted to optionally overwrite an existing project, or rename to a new project name.

Once uploaded, the project can be compiled by the ISaGRAF Workbench and a debugger connection made to the RTU.

8 I/O Interfaces

ISaGRAF provides several interfaces for associating RTU data with ISaGRAF variables. The following sections describe the methods that can be used.

- [Physical I/O](#)^[41]
 - [Derived Data](#)^[43]
 - [ISaGRAF I/O Boards](#)^[44]
 - [PLC Device I/O Boards](#)^[52]
 - [RTU Data via Function Blocks](#)^[63]
-

8.1 Physical I/O

Physical inputs and outputs on the SCADAPack E RTU can be accessed by the ISaGRAF application via the ISaGRAF I/O Board mechanism. RTU internal data points may also be accessed via I/O boards (or via other C Function Blocks) as will be described later in this section. Each ISaGRAF I/O board needs to be supplied with an address that specifies the RTU starting point index or offset when reading from inputs or writing to outputs. This address is entered into the **first_point_num** field of the particular ISaGRAF I/O board using the ISaGRAF Workbench I/O Connections editor.

For information on the SCADAPack E RTU physical point index mapping see the following manuals, where relevant:

- *SCADAPack E 5000 Series I/O Expansion Reference manual*
- *SCADAPack E Configurator User Manual*
- *SCADAPack E DNP3 Technical Reference manual*

Stopping an ISaGRAF Application

The effect of stopping an ISaGRAF application on Physical I/O varies between SCADAPack E controllers.

The following table defines the effects.

When ISaGRAF application is Stopped:	SCADAPack ES local I/O	SCADAPack ER I/O cards	SCADAPack k 300E I/O	SCADAPack ES Remote I/O	5000 Series I/O	Remote RTU & IED Devices *
Binary Inputs updated in database?	Yes	Yes	Yes	Yes	Yes	Yes
Analog Inputs updated in database?	Yes	Yes	Yes	Yes	Yes	Yes
Binary Output & database state if ISaGRAF I/O board not using "Hold on Stop" parameter	Reset Off	Reset Off	Unchanged	Unchanged	Unchanged	Unchanged
Binary Output & database state if ISaGRAF I/O board is using "Hold on Stop" parameter	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged
Analog Output & database value if ISaGRAF I/O board not using "Hold on Stop" parameter	Reset to 0	Reset to 0	Unchanged	Unchanged	Unchanged	Unchanged
Analog Output & database value if ISaGRAF I/O board is using "Hold on Stop" parameter	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged
Binary Output changes if database value updated after ISaGRAF Stop	Yes	Yes	Yes	Yes	Yes	Yes

Analog Output changes if database value updated after ISaGRAF Stop	Yes	Yes	Yes	Yes	Yes	Yes
--	-----	-----	-----	-----	-----	-----

*Applies to DNP3 Master and IEC60870-5-103 Master communications to remote devices

See [Hold On Stop](#)^[51] for ISaGRAF output board parameter information.

The above table does not include data to/from PLC devices such as Modbus, DF1, etc. ISaGRAF manages the data transfer for PLC protocol communications and so when an ISaGRAF application is stopped, PLC communications and data transfer ceases.

8.2 Derived Data

Derived RTU data (e.g. calculations to be sent to a SCADA master, data for peer-to-peer communications, etc.) can be accessed by the ISaGRAF application in exactly the same way as physical I/O, i.e. using the I/O board mechanism (or function block mechanism). Each board type used to access physical I/O may be used for manipulating derived RTU data, by simply specifying an appropriate *first_point_num*. Derived RTU points can be created using SCADAPack E Configurator. Once created, a user ISaGRAF application can open I/O boards attached to the derived points.

Analog I/O boards can receive data from and send data to ISaGRAF in integer or real (floating point) format from pseudo or physical I/O points (e.g. for SCADA). Floating point RTU data is related to integer data through Raw and Engineering scaling parameters configured for each point. Analog points on I/O boards may also use ISaGRAF Conversion Tables. For more information on data conversion, see Section [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#) [65].

ISaGRAF output boards used for derived outputs *export data* from an ISaGRAF application to the RTU database. This data is then available to the SCADA master system or other peer RTUs. Similarly ISaGRAF input boards used for derived points *import data* from the RTU database into an ISaGRAF application. Control data from a SCADA master, or peer RTU for example is then available to the ISaGRAF application.

ISaGRAF derived outputs = SCADA master inputs, Peer RTU input data

ISaGRAF derived inputs = SCADA master outputs, Peer RTU output data

ISaGRAF input variables attached to input boards from RTU derived data may be initialized to a predefined state via the ISaGRAF *OPERATE* command. See Section [ISaGRAF OPERATE Function](#) [66]. [Figure 6.1](#) [43] shows an example of the ISaGRAF I/O board interface, with inputs attached to the currently selected board (rtu16di) displayed on the right.

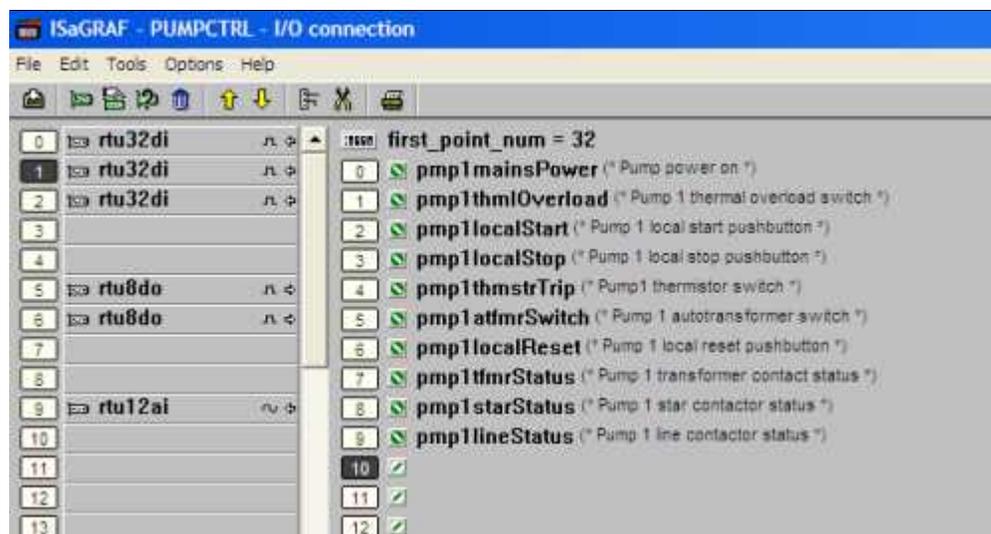


Figure 6.1: ISaGRAF I/O Boards

8.3 ISaGRAF I/O Boards

The list below contains I/O Boards available in the Schneider Electric ISaGRAF library for the SCADAPack E RTU.

The ISaGRAF I/O board types access database points configured in the RTU.

Both physical I/O points and derived points are accessed using an I/O board.

ISaGRAF I/O boards need not necessarily correspond to the RTU I/O card arrangements.

Table 6.1: SCADAPack E ISaGRAF Library I/O Boards

Board Name	ref [†]	ISaGRAF Data Type	RTU Points supported
rtu16di	0001	16 Boolean Inputs	Physical Input, Derived & System Binary Objects.
rtu32di	0001	32 Boolean Inputs	Physical Input, Derived & System Binary Objects
rtu4do	0002	4 Boolean Outputs	Physical Output, Derived & System Binary Objects
rtu8do	0002	8 Boolean Outputs	Physical Output, Derived & System Binary Objects
rtu16do	0002	16 Boolean Outputs	Physical Output, Derived & System Binary Objects
rtu8dos	000B	8 Boolean Outputs	Physical Onput, Derived & System Binary Objects with Status
rtu16dos	000B	16 Boolean Outputs	Physical Onput, Derived & System Binary Objects with Status
rtu6ai	0003	6 Analog Inputs* ¹	Physical Input &, Derived Integer & Floating Point Objects.
rtu2ao	0004	2 Analog Output* ¹	Physical Output &, Derived Integer & Floating Point Objects
rtu4ao	0004	4 Analog Outputs* ¹	Physical Output &, Derived Integer & Floating Point Objects
rtu2aos	000C	2 Analog Output* ¹	Physical Output &, Derived Integer & Floating Point Object with Status
rtu4aos	000C	4 Analog Outputs* ¹	Physical Output &, Derived Integer & Floating Point Object with Status

rtu16ctr	0005	16 Counter Inputs	Counter Input Objects
rtu32ctr	0005	32 Counter Inputs	Counter Input Objects.

+ For advanced ISaGRAF users: other I/O Boards, I/O Configurations or Complex Equipment types based on these reference numbers shown are possible. *ref* is the ISaGRAF target Board Reference parameter shown in HEX format. The SCADAPack E RTU has no limit on the number of I/O channels per board for board types 0001, 0002, 0003, 0004, 0005, 000B & 000C. PLC / Peripheral boolean I/O board types are limited to 32 channels per board.

*1 See Analog Input and Output Board representation and conversion rules in [ISaGRAF Variable / DNP3 Point Interaction](#)^[64] and [ISaGRAF Analog I/O Boards / DNP3 Representation & Conversion](#)^[65].

8.3.1 Digital Input & Output Boards, & Counter Input Boards

Digital Input Boards

Physical RTU digital points may be imported into ISaGRAF through Digital Input (di) boards (ISaGRAF *Boolean Input Board* types). Where an ISaGRAF application attaches a Boolean variable to a Digital Input Board, the *Current State Property* of the digital point will be read into the ISaGRAF variable. If the digital point is a Physical Binary I/O address, the physical digital input channel corresponding to that address is read.

Digital Output Boards

Physical RTU digital outputs have two sets of ISaGRAF interfaces. The state of a digital output is controlled through the Digital Output (do) boards (ISaGRAF *Boolean Output Board* types). The feedback status of digital outputs are read through Digital Output Status (dos) boards (ISaGRAF *Digital Input Board* types).

Derived RTU Digital points are controlled through Digital Output (do) boards (ISaGRAF *Boolean Output Board* types). The feedback status of derived digital points is read into Digital Input (di) boards (ISaGRAF *Boolean Input Board* types).

Where an ISaGRAF application attaches a Boolean variable to a Digital Output Board, the *Current State Property* of the digital point will be controlled from the ISaGRAF variable.

Counter Input Boards

ISaGRAF application *Counter Input Boards* (ctr) support only ISaGRAF “Integer” analog variables. The *Current Integer Value* property of the physical counter input will be read into the ISaGRAF variable.

ISaGRAF analog integer variables contain signed 32-bit numbers, however Counter Inputs are 32-bit unsigned values. For counter values less than 2147483648, the counter value and ISaGRAF variable value are the same. For counter values above 2147483647, the ISaGRAF variable indicates a negative value. The user ISaGRAF application needs to handle the case where counter input numbers greater than 2147483647 are indicated as negative ISaGRAF numbers. This may be necessary, for example, where a comparison or subtraction of counter values occurs in the user ISaGRAF application. (e.g. Preset counters or reset counter prior to value exceeding 2147483647).

8.3.2 Analog Input & Output Boards

Analog Input Boards

Physical RTU Analog points may be imported into ISaGRAF through Analog Input (ai) boards (ISaGRAF *Analog Input Board* types). Where an ISaGRAF application attaches an “Integer” analog variable to an Analog Input Board, the *Current Integer Value* property of the analog point will be read into the ISaGRAF variable. Where an ISaGRAF application attaches a “Real” (floating point) analog variable to an Analog Input Board, the *Current Eng. Value* property of the analog point will be read into the ISaGRAF variable.

Where the analog point is a Physical Analog I/O address, the Physical Analog Input channel corresponding to that address is read. See the following section regarding reading the value of a Physical Analog Output channels.

Both “Integer” and “Real” ISaGRAF analog variables may be mixed on the same ISaGRAF Analog Input Board.

ISaGRAF analog integer variables contain signed 32-bit numbers. The value of an “Integer” analog variable will be the physical analog input variable in the range MIN-RAW to MAX-RAW as configured in the point’s attributes.

ISaGRAF analog real variables contain 32-bit floating point numbers. For a physical analog input variable, variables will be in the range MIN-ENG to MAX-ENG as configured in the point’s attributes.

Analog Output Boards

Physical RTU Analog Outputs have two sets of ISaGRAF interfaces. The value of physical analog outputs is controlled through Analog Output (ao) boards (ISaGRAF *Analog Output Board* types). The feedback status of analog outputs are read into ISaGRAF through Analog Output Status (aos) boards (ISaGRAF *Analog Input Board* types).

Derived RTU Analog points are controlled through Analog Output (ao) boards (ISaGRAF *Analog Output Board* types). The feedback status of derived analog points is read into ISaGRAF through Analog Input (ai) boards (ISaGRAF *Analog Input Board* types).

Where an ISaGRAF application attaches an “Integer” analog variable to an Analog Output Board, the *Current Integer Value* property of the analog point will be controlled from the ISaGRAF variable. The analog point’s *Current Integer Value* property, MIN-RAW, MAX-RAW, MIN-ENG & MAX-ENG attributes will be used to automatically calculate the *Current Eng. Value* property of the point.

Where an ISaGRAF application attaches a “Real” (floating point) analog variable to an Analog Output Board, the *Current Eng. Value* property of the analog point will be controlled from the ISaGRAF variable. The analog point’s *Current Eng. Value* property MIN-RAW, MAX-RAW, MIN-ENG & MAX-ENG attributes will be used to automatically calculate the *Current Integer Value* property of the analog point.

Both “Integer” and “Real” ISaGRAF analog variables may be mixed on the same ISaGRAF Analog Output Board.

8.3.3 SCADAPack ER I/O Boards

These I/O boards are only supported on SCADAPack ER RTU's. Applications on other SCADAPack E Smart RTUs that reference these I/O boards will not start.

SCADAPack ER I/O boards reference physical channels directly, as opposed to referencing a specific I/O channel by point number. The SCADAPack ER I/O boards provided in the SCADAPack E ISaGRAF Library are listed in the following table

Table 6.2: SCADAPack ER I/O Boards

Board Name	Board Reference (hex)	ISaGRAF Data Type
er16do	0x18	16 Boolean Outputs
er16ai	0x19	16 Analog Inputs *
er32di	0x1A	32 Boolean Inputs
er4ao	0x1B	4 Analog Outputs *

* See Analog conversion rules in [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#) ^[65].

The SCADAPack ER I/O boards reference the respective physical I/O cards by specifying a *Slot_Num* field. The *Slot_Num* field is set via user configuration through the I/O board parameters. These are set as part of the ISaGRAF application and are entered into the I/O board parameter fields within the ISaGRAF Workbench I/O Connections editor.

The required fields are described as follows

Slot_Num: specifies the I/O card slot on the SCADAPack ER rack
 1 = I/O Card Slot 1
 2 = I/O Card Slot 2, etc.

The default value is 1 (i.e. I/O Card Slot 1).

A valid I/O card configuration needs to be loaded into the SCADAPack ER RTU prior to loading an ISaGRAF application that references SCADAPack ER I/O boards, otherwise the I/O board will not be opened and the **ISaGRAF application WILL NOT START**. This is done using SCADAPack E Configurator by assigning an I/O card and writing the Configurator file changes onto the RTU. A RTU restart is required after these configuration details have been written to the RTU. See *SCADAPack E Configurator User Manual* for details.

8.3.3.1 Digital Output (er16do) & Input (er32di) Board

Digital Output Board: er16do

The **er16do** output board references a physical relay output card by specifying a *Slot_Num* field (see [SCADAPack ER I/O Boards](#)^[48] for detailed descriptions of these fields). The channel number in the ISaGRAF I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Where an ISaGRAF application attaches a Boolean variable to an **er16do** output board, the state of the corresponding digital relay will be controlled from the ISaGRAF variable. If there is a physical digital output configuration point associated with this physical channel, the *Current State* of this configuration point will be updated after the successful control of the relay output.

Controls issued to SCADAPack ER relay output cards resulting from **attached** variables changing state, are issued as complete I/O card controls. This allows that any simultaneous state changes at the ISaGRAF output board level, are executed simultaneously at the SCADAPack ER relay output card.

The **er16do** output board can be opened only if there is valid I/O card configuration loaded into the SCADAPack ER controller. Unlike the standard output boards, it is NOT necessary that there are physical digital output configurations points associated with the physical channels referenced by the **er16do** output board.

Digital Input Board: er32di

The **er32di** input board references a physical binary input card by specifying a *Slot_Num* field (see Section [SCADAPack ER I/O Boards](#)^[48] for detailed descriptions of these fields). The channel number in the ISaGRAF I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Unlike the **er16do** output board, there needs to be point database configuration points associated with the physical channels referenced by the **er32di** input board *Slot_Num* fields for proper operation.

Where an ISaGRAF application attaches a Boolean variable to a **er32di** input board, the *Current State Property* of the digital point will be read into the ISaGRAF variable. The **er32di** input board may be successfully opened if there is a valid I/O card configuration loaded into the SCADAPack ER controller, and there is at least 1 physical binary input configuration point associated with the given I/O card.

8.3.3.2 Analog Input (er16ai) & Output (er4ao) Board

Analog Input Board: er16ai

The **er16ai** input board references a physical analog input card by specifying a *Slot_Num* field (see Section [SCADAPack ER I/O Boards](#)^[48] for detailed descriptions of these fields). The channel number in the ISaGRAF I/O Connection window corresponds to the physical channel number on the SCADAPack ER I/O card.

Unlike the **er16do** output board, there needs to be point database configuration points associated with the physical channels referenced by the **er16ai** input board *Slot_Num* fields for proper operation.

Where an ISaGRAF application attaches an “Integer” analog variable to a **er16ai** input board, the *Current Value Integer Property* of the associated analog point will be read into the ISaGRAF variable. Where an ISaGRAF application attaches a “Real” (floating point) analog variable to an **er16ai** input Board, the *Current Value Engineering Property* of the associated analog point will be read into the ISaGRAF variable. Both “Integer” and “Real” ISaGRAF analog variables may be mixed on the same ISaGRAF **er16ai** input Board.

The **er16ai** input board can be opened only if there is a valid I/O card configuration loaded into the SCADAPack ER controller, and there is at least 1 physical analog input configuration point associated with the given I/O card.

Analog Output Board: er4ao

The **er4ao** output board references a physical analog output card by specifying a *Slot_Num* field (see [SCADAPack ER I/O Boards](#)^[48] for detailed descriptions of these fields). The channel number in the ISaGRAF I/O Connection window corresponds to the physical channel number on the I/O card.

Unlike the **er16do** output board, there needs to be point database configuration points associated with the physical channels referenced by the **er4ao** output board *Slot_Num* fields for proper operation.

Where an ISaGRAF application attaches an “Integer” analog variable to the **er4ao** Output Board, the *Current Value Integer Property* of the analog point will be controlled from the ISaGRAF variable. The analog point’s *Current Value Integer Property*, MIN-RAW, MAX-RAW, MIN-ENG & MAX-ENG attributes will be used to automatically calculate the *Current Value Engineering Property* of the point.

Where an ISaGRAF application attaches a “Real” (floating point) analog variable to the **er4ao** Output Board, the *Current Value Engineering Property* of the analog point will be controlled from the ISaGRAF variable. The analog point’s *Current Value Engineering Property* MIN-RAW, MAX-RAW, MIN-ENG & MAX-ENG attributes will be used to automatically calculate the *Current Value Integer Property* of the analog point.

Both “Integer” and “Real” ISaGRAF analog variables may be mixed on the same ISaGRAF **er4ao** Output Board.

8.4 Remote Control Interlock & Hold On Stop

Remote Control Interlock

By default, RTU output points (Binary & Analog, Physical & Derived) attached to an ISaGRAF Output Board are under ISaGRAF control while a user ISaGRAF application is executing. Attempts to control an output point that is under ISaGRAF control will normally lead to a control error. However, a “Remote Interlock” point may be associated with each output point.

If a “Remote Interlock” point is defined and configured for an output point, but is *inactive*, ISaGRAF retains control of the output point. Communication requests to the output point (e.g. from DNP3, IEC60870-5 protocols, Modbus, etc) is rejected when ISaGRAF is controlling that output point.

If a “Remote Interlock” point is defined and configured for an output point, but is *active*, then ISaGRAF does not have control of the physical output point. Instead, communication requests to physical outputs are accepted to the point.

Hold On Stop

The SCADAPack E ISaGRAF output boards support a **hold_on_stop** parameter.

This BOOLEAN parameter allows the user to identify how output boards are processed on a shutdown (or restart) of the ISaGRAF application.

This applies for local RTU physical output points on the SCADAPack ES and SCADAPack ER RTUs.

The hold on stop parameter is not used on the SCADAPack 300E RTUs even if it is available on an ISaGRAF output board.

See [Stopping an ISaGRAF Application](#)^[41] for a description of the effect of stopping an ISaGRAF application.

This Hold On Stop ISaGRAF output board parameter defaults to *false* (meaning outputs will be reset off or reset to 0 when the ISaGRAF application stops).

When set to true, the local physical outputs of SCADAPack ES and SCADAPack ER RTUs are held at their current values until the database point is updated to some other value (e.g. by a new ISaGRAF application, DNP3 / IEC60870-5-101 / Modbus control, etc).

8.5 PLC Device I/O Boards

As an extension of the data interface provided by the SCADAPack E RTU, access by ISaGRAF applications to external PLC or peripheral device data is supported.

Standard ISaGRAF I/O boards can access RTU I/O and the point database, while PLC Device ISaGRAF boards allow data to be extracted from external devices such as PLCs,

External peripheral data (e.g. from PLCs) is cached internally by the RTU to maximize ISaGRAF application performance. Access to this cached device data is restricted to ISaGRAF and is termed PLC Device data. Direct access to PLC device data through DNP3 / IEC60870 or other protocols requires additional ISaGRAF logic to copy the peripheral data to/from the RTU point database.

The PLC device that can be accessed via the ISaGRAF PLC Device I/O boards depends on the PLC or peripheral device drivers supported by the RTU Operating System firmware.

LED(s) on the SCADAPack E RTU may indicate communication activity with external peripheral device(s). For more information see the relevant RTU *Hardware User Manual*.

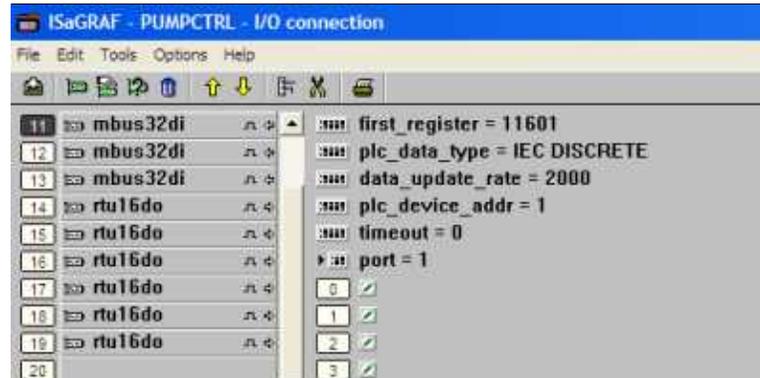


Figure 6.2: Example ISaGRAF PLC Device I/O boards

TIP:

When connecting the ISaGRAF Workbench Debugger to a SCADAPack E RTU with an application using PLC Device I/O boards, the Debugger may indicate "DISCONNECTED" for a period of time, particularly if there is a large number of PLC Device I/O boards, or if a PLC device is not responding. Try increasing the ISaGRAF Workbench timeout to correct this.

Different PLC Device I/O boards are provided for different types of PLC data.

For example: read PLC value registers (analog input board), write PLC coils (boolean output board), read PLC accumulated data (analog input board). The different types of I/O boards available and ranges of PLC data that can be accessed depend on the individual PLC driver. The following section details a summary of the SCADAPack E RTU's MODBUS PLC driver. For detailed information on Modbus and other drivers, see the relevant *SCADAPack E PLC Device Interface* manual.

ISaGRAF PLC Device I/O boards access data in the following way:

- a PLC Device input board normally corresponds to a read access to PLC data
- a PLC Device output board corresponds to a write-only access to PLC data

-
- an ISaGRAF OPERATE function call may be performed on an input variable and write to a PLC (in this case the input board variable is read-write)
 - Serial communication with external devices, such as PLCs, is made through the SCADAPack E RTU port(s) configured as "PLC Device".
 - ISaGRAF function blocks: **mbusctrl**, **mtcpctrl**, **df1ctrl**, **koyoctrl** can be used to control read & write communications to the PLC device whose communication is defined by I/O boards. Disabling read operations affects PLC device Input Board data. Disabling write operations affects PLC device Output Board data and "OPERATE" function data.
 - Up to a total of 100 PLC Device I/O boards can be defined in total for "PLC Device" communication ports and ISaGRAF kernels. Multiple "PLC Device" serial ports, as well as TCP/IP channels for some PLC devices, can be used for PLC peripheral communication.
-

8.5.1 Input Boards

ISaGRAF PLC Device input boards typically require user configuration through the I/O board parameters. These are set as part of the ISaGRAF application and are entered into the I/O board parameter fields within the ISaGRAF Workbench I/O Connections editor.

The ISaGRAF “OPERATE” function may be used on PLC Device Input Boards where the PLC register read by the input board is also writeable. This permits PLC registers to be inputs into ISaGRAF, but have them “Presetable” in the PLC by ISaGRAF.

Typical fields are:

first_register: (or similar name depending on the particular PLC terminology) specifies the PLC Device data registers to access when reading from PLC data into ISaGRAF variables. The PLC data type accessed is specific to the PLC Device I/O board and board address. This value is usually the PLC’s data (or register) address.

plc_data_type: specifies the PLC data register type. Currently *IEC UINT* type is supported for analog boards and *IEC DISCRETE* type is supported for Boolean boards. Other data types may be supported in the future. See specific PLC driver interface manuals for more information.

data_update_rate: The units for this parameter vary depending on the type of PLC device. For example this may be a setting in milliseconds for a directly connected device, or in minutes for a low power type device (see the *SCADAPack E Modbus Communication Interfaces* manual). As the SCADAPack E RTU needs to extract the data for the I/O board from the PLC or peripheral device, this sets the rate at which the data is extracted. Individual I/O boards may have different data update rates allowing prioritization of data extracted from a PLC Device. The SCADAPack E RTU may not be able to read requested PLC data within the time set by the data update rate depending on the quantity of data to be read, rate of write requests and PLC communication speed. In this case the update rates will be slower.

plc_device_addr: Some PLC device drivers support multi-drop PLC devices on the same communication channel, or have unique addressing identifiers. Where the SCADAPack E RTU driver provides multi-drop support, ISaGRAF may access data from any of the locally multi-dropped devices. A separate I/O board will be required for each device.

timeout: PLC device drivers with comprehensive I/O board interfaces may provide a parameter for specifying the communications timeout on an individual I/O board (i.e. the timeout applies to communications associated with that board). Where this value is “0”, the PLC device driver will use a default timeout. The units for this field are dependent upon the PLC device driver. Units may be, for example, milliseconds, seconds, minutes, etc.

port: this parameter may be on a PLC Device I/O board for a device driver. Where present, it defines which of the multiple SCADAPack E RTU “PLC Device” ports will be used to communicate with the PLC or peripheral device. ISaGRAF PLC Device I/O boards not including this parameter can only be used when a single “PLC Device” port is configured on the SCADAPack E RTU.

8.5.2 Output Boards

ISaGRAF PLC Device output boards typically require user configuration through the I/O board parameters. These are set as part of the ISaGRAF application and are entered into the I/O board parameter fields within the ISaGRAF Workbench I/O Connections editor.

Typical fields are:

first_register: (or similar name depending on the particular PLC terminology) specifies the PLC Device data registers to access when writing from ISaGRAF variables to PLC data. The PLC data type accessed is specific to the PLC Device I/O board and board address. This value is usually the PLC's data (or register) address.

plc_data_type: specifies the PLC data register type. Currently *IEC UINT* type is supported for analog boards and *IEC DISCRETE* type is supported for Boolean boards. Other data types may be supported in the future. See specific PLC driver interface manuals for more information.

plc_device_addr: Some PLC device drivers support multi-drop PLC devices on the same communication channel, or have unique addressing identifiers. Where the RTU driver provides multi-drop support, ISaGRAF may access data from any of the locally multi-dropped devices. A separate I/O board will be required for each device.

must_write_rate: The unit for this parameter is driver specific, and configures the rate at which the data for the Output board is written to the PLC. Between "*must_write_rate*" periods, data is written to the PLC only when the ISaGRAF output variable values change. Individual I/O boards may have different must write rates allowing prioritization of data sent to a PLC Device.

timeout: PLC device drivers with comprehensive I/O board interfaces may provide a parameter for specifying the communications timeout on an individual I/O board (i.e. the timeout applies to communications associated with that board). Where this value is "0", the PLC device driver will use a default timeout. The units for this field are dependent upon the PLC device driver. Units may be, for example, milliseconds, seconds, minutes, etc.

port: this parameter may be on a PLC Device I/O board for a device driver. Where present, it defines which of multiple RTU "PLC Device" ports will be used to communicate with the PLC or peripheral device. If only one "PLC Device" port is configured, this field is ignored. ISaGRAF PLC Device I/O boards not including this parameter can only be used when a single "PLC Device" port is configured on the SCADAPack E RTU.

8.5.3 Board Status

The SCADAPack E RTU checks for data being written to the PLC by ISaGRAF, before the PLC Device input board data is retrieved.

Communication requests made by the SCADAPack E RTU to the PLC are asynchronous to the scanning of the ISaGRAF application, but data within ISaGRAF remains consistent duration of a scan cycle.

To assist with debugging of ISaGRAF PLC Device I/O board communication, the SCADAPack E RTU provides two types of *analog* system points which provide useful information:

1. PLC Communication Status

Status available for the first 60 ISaGRAF PLC Device I/O Boards used in ISaGRAF kernel 1, and the first 14 ISaGRAF PLC Device I/O Boards used in ISaGRAF kernel 2.

2. Cache Time (seconds)

- for ISaGRAF PLC Device Input Boards these system points represents the age of the data since the last update
- these system points are not updated for ISaGRAF PLC Device Output Boards (0)

SCADAPack E RTU analog system points 53300 to 53419 are set aside for the PLC Device communication status information for up to 60 PLC Device I/O boards in kernel 1. Points 53422 to 53449 are set aside for the PLC Device communication status information for up to 14 PLC Device I/O boards in kernel 2. A pair of consecutive points represent the PLC Communication status ('COMMS_STATUS') and the age of the data ('UPDATE_TIME') for each PLC Device I/O board (refer to table below). Non 'PLC Device' ISaGRAF I/O boards, including blank ISaGRAF I/O slots, are not included in the consecutive count.

This data is accessible externally to the RTU (via a communications protocol accessing these points) or using ISaGRAF (through *rtu* input boards or table read/write function blocks to these point addresses). These points are only accessible if PLC Device functionality is enabled.

Table 6.3: 'Comms Status' & 'Cache Time' System Point Numbers for ISaGRAF kernel 1

Consecutive ISaGRAF PLC Device I/O board	Comms_Status Analog System Point No.	Cache_Time Analog System Point No.
1	53300	53301
2	53302	53303
3	53304	53305
4	53306	53307
5	53308	53309
6	53310	53311
7	53312	53313
8	53314	53315

Consecutive ISaGRAF PLC Device I/O board	Comms_Status Analog System Point No.	Cache_Time Analog System Point No.
9	53316	53317
10	53318	53319
11	53320	53321
12	53322	53323
13	53324	53325
14	53326	53327
15	53328	53329
16	53330	53331
17	53332	53333
18	53334	53335
19	53336	53337
20	53338	53339
21	53340	53341
22	53342	53343
23	53344	53345
24	53346	53347
25	53348	53349
26	53350	53351
27	53352	53353
28	53354	53355
29	53356	53357
30	53358	53359
31	53360	53361
32	53362	53363
33	53364	53365
34	53366	53367
35	53368	53369
36	53370	53371

Consecutive ISaGRAF PLC Device I/O board	Comms_Status Analog System Point No.	Cache_Time Analog System Point No.
37	53372	53373
38	53374	53375
39	53376	53377
40	53378	53379
41	53380	53381
42	53382	53383
43	53384	53385
44	53386	53387
45	53388	53389
46	53390	53391
47	53392	53393
48	53394	53395
49	53396	53397
50	53398	53399
51	53400	53401
52	53402	53403
53	53404	53405
54	53406	53407
55	53408	53409
56	53410	53411
57	53412	53413
58	53414	53415
59	53416	53417
60	53418	53419

Table 6.4: 'Comms Status' & 'Cache Time' System Point Numbers for ISaGRAF kernel 2

Consecutive ISaGRAF PLC Device I/O board	Comms_Status Analog System Point No.	Cache_Time Analog System Point No.
1	53422	53423
2	53424	53425
3	53426	53427
4	53428	53429
5	53430	53431
6	53432	53433
7	53434	53435
8	53436	53437
9	53438	53439
10	53440	53441
11	53442	53443
12	53444	53445
13	53446	53447
14	53448	53449

These system points are only accessible if PLC Device functionality is enabled through a *PLC Device* port setting or *Modbus/IP (Client)* setting.

Table 6.5: PLC Comms Status values

- 0** = Normal Operation
- 101** = Unknown device status
- 102** = Illegal Data Count
- 103** = Illegal Data Address
- 104** = Device Timeout
- 105** = Read/Write lock failed
- 106** = Invalid message
- 107** = Device Busy
- 108** = Data value out of range

PLC Device Input Board data is updated by the SCADAPack E RTU when it communicates with the PLC device. The PLC communication status is updated if there is an error returned from the PLC or no response from the PLC after a data request by the RTU. Error codes are presented in Table 44. The value in each status register is cleared by the RTU upon successful communication sessions. Variables within an ISaGRAF program can be used to log transient errors.

PLC Device Output Board data is written to the PLC when an ISaGRAF application changes the value of a variable attached to the Output Board. In addition, output board data is written to the PLC under the following conditions:

- When the ISaGRAF application starts, output board data is written
- If the PLC does not respond to a control, it is resent until it is responded
- Boards with a “must_write_rate” parameter output data at this rate
- Output board data is rewritten at a background update rate

SCADAPack E RTU system point “53420” controls the background update rate of all PLC Device Output Boards on the RTU. Its default value is 60 seconds. It may be adjusted by the user dynamically or specified in an RTU configuration, and is a non-volatile RTU system point that is retained by the RTU. Note that the background updates are disabled by setting the system point “53420” to 0 (zero). This may be used to optimize the PLC Device communications bandwidth where background writes are not appropriate or not necessary.

There are individual PLC Device I/O Boards available in the Schneider Electric ISaGRAF library for different types of PLC devices and different types of data within the same PLC device. Not every PLC data type for a particular PLC device may be accessible from the PLC Device I/O boards. For more information see the relevant *SCADAPack E PLC Device ISaGRAF Interface* manual.

Board Name	Ref*	ISaGRAF Data Type
<i>ddd16di</i>	0007	16 Boolean Inputs
<i>ddd16do</i>	0008	16 Boolean Outputs
<i>ddd1ai</i>	0009	1 Analog Input* ¹
<i>ddd4ai</i>	0009	4 Analog Inputs* ¹
<i>ddd8ai</i>	0009	8 Analog Inputs* ¹
<i>ddd1ao</i>	000A	1 Analog Output* ¹
<i>ddd4ao</i>	000A	4 Analog Outputs* ¹
<i>ddd8ao</i>	000A	8 Analog Outputs * ¹
<i>pppp16di</i>	000E	16 Boolean Inputs
<i>pppp16do</i>	000F	16 Boolean Outputs
<i>pppp32di</i>	000E	32 Boolean Inputs
<i>pppp32do</i>	000F	32 Boolean Outputs
<i>pppp1ai</i>	0010	1 Analog Input* ¹
<i>pppp4ai</i>	0010	4 Analog Inputs* ¹

<i>pppp8ai</i>	0010	8 Analog Inputs* ¹
<i>pppp16ai</i>	0010	16 Analog Inputs* ¹
<i>pppp32ai</i>	0010	32 Analog Inputs* ¹
<i>pppp64ai</i>	0010	64 Analog Inputs* ¹
<i>pppp1ao</i>	0010	1 Analog Output* ¹
<i>pppp4ao</i>	0011	4 Analog Outputs* ¹
<i>pppp8ao</i>	0011	8 Analog Outputs * ¹
<i>pppp16ao</i>	0011	16 Analog Outputs * ¹
<i>pppp32ao</i>	0011	32 Analog Outputs * ¹
<i>pppp64ao</i>	0011	64 Analog Outputs * ¹

where:

ddd is the driver type for simpler peripheral device interfaces

pppp is the driver type for more comprehensive peripheral device interfaces

For example *mbus16di* reads 16 Boolean (digital) input points from a Modbus PLC, with settable data type and timeout parameters.

*¹ Analog input and output board conversion may be used as described in Section [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#)^[65]. The OPERATE function may also be used with Analog input boards as described in Section [ISaGRAF OPERATE Function](#)^[66]

+ For advanced ISaGRAF users: other I/O Boards, I/O Configurations or Complex Equipment types are possible based on the indicated reference numbers (hexadecimal format). Depending on the RTU's PLC driver, there is usually an upper limit of 32 I/O channels for boolean board types and 64 I/O channels for analog board types.

A *plc_data_type* user parameter is defined for some comprehensive PLC Device I/O boards. The value of this parameter field is a string field describing the data type of data being accessed (driver specific) – e. g. "IEC UINT".

An additional *plc_dev_type* hidden parameter string field value is driver specific, but generally has the form "xyz" where x = plc type, y = communication channel type, and z = special controls. "y" & "z" fields may be optional depending upon the specific driver.

E.g. "M" indicates simple Modbus board (M)

"ms" indicates advanced Modbus board (m), serial comms interface (s);

"mt" indicates advanced Modbus board (m), TCP socket interface (t);

"mtr" indicates advanced Modbus board (m), TCP socket interface (t), reset outputs on ISaGRAF application halted (r).

The "reset outputs on ISaGRAF application halted" function as set by the I/O board's hidden "*plc_dev_type*" parameter defaults to "Freeze" outputs unless an "r" is explicitly defined as a "special

controls” character.

Normally, the SCADAPack E RTU’s interface with intelligent PLC equipment would use ‘Freeze’ outputs to internal PLC data, so that a halted RTU ISaGRAF application would leave output data in the last written state. In the case where the RTU interfaces with I/O blocks for purposes of distributed I/O, for example, the “Reset” outputs option may be applicable. Complex equipment types describing I/O blocks, for example, may include an “r” special control field to force outputs to the “Reset” state after an ISaGRAF application is stopped.

8.6 RTU Data via Function Blocks

SCADAPack E provides a mechanism separate from ISaGRAF I/O Boards (described in preceding sections) for accessing RTU data.

In general the function blocks require more processing capacity in the RTU compared with ISaGRAF I/O Boards, but provide greater programming flexibility and access to more detailed RTU data.

Function blocks provide access to reading and writing current data values to/from the RTU point database, as well as access to point attributes not available via ISaGRAF I/O boards.

- The **getpntxx** functions and function blocks allow an ISaGRAF application to read point current value data from the database (applies to Physical I/O, Derived data and System Points)
- The **setpntxx** functions allow an ISaGRAF application to write point current value data to the database (applies to Physical Outputs, Derived data and System Points)
- The **rtupulsx** function blocks allow an ISaGRAF application to have accurate pulse control of binary points (applies to Physical Binary Outputs and Derived Binary points)
- The **rdfld_x** function blocks allow an ISaGRAF application to read attribute and property fields from points in the RTU database
- The **setatr_x** function blocks allow an ISaGRAF application to set attributes of points in the RTU database
- The **rdrec_x** function blocks provide ISaGRAF applications with a set of commonly used attribute and property fields for points in the RTU database

ISaGRAF Arrays can be useful when using the above Functions and Function Blocks. For more information see [Array Variables](#)^[32].

Details of these, and other SCADAPack E ISaGRAF functions blocks are described in the *SCADAPack E ISaGRAF Function Block Reference* manual.

9 ISaGRAF Variable / Point Interaction

ISaGRAF variables attached to the I/O boards described in Section [ISaGRAF I/O Boards](#)^[44] read data from or write data to the SCADAPack E RTU database points.

ISaGRAF *Input* boards read data from the RTU database into ISaGRAF input variables. ISaGRAF *output* boards write data to the RTU database from ISaGRAF output variables.

Each ISaGRAF I/O board is associated with a **first_point_num** corresponding to the number of the first point on the board. The following configuration concepts and rules apply:

- Variables attached to RTU point I/O boards correspond to consecutively numbered database points. The ISaGRAF I/O board address may be any valid RTU data point index (0-65535) corresponding to physical, derived or RTU system data of a compatible type (see I/O boards information in Section [ISaGRAF I/O Boards](#)^[44]).
- ISaGRAF Boolean I/O boards correspond to consecutive binary points starting at the board address of the I/O board. ISaGRAF Boolean Output boards cannot reference "read-only" RTU data points (e.g. physical digital inputs). Boolean I/O Boards support multiple channels.
- ISaGRAF Analog I/O boards support multiple channels; each corresponds to an RTU analog data point. The RTU point number of the first channel is specified by the board address. Subsequent channels correspond to consecutive RTU point numbers. ISaGRAF Analog Output boards cannot reference "read-only" data registers (e.g. physical analog inputs). See Section [ISaGRAF Analog I/O Boards / Point Representation & Conversion](#)^[65] below for analog conversion rules.
- ISaGRAF digital and analog output status Input boards (references 000B & 000C) support multiple channels; each corresponds to the status of a physical output point.
- ISaGRAF Counter Input boards (reference 000D) map to RTU counter input points or system counter points. SCADAPack E RTU counters are managed internally by 32-bit unsigned data types and are presented in 32-bit format to ISaGRAF analog integers. To reset or preset counters use the OPERATE command (see below).

If none of the RTU points referenced on the specific I/O board exist in the SCADAPack E RTU's database, the **ISaGRAF application WILL NOT START**. In order for a given board to be successfully opened, at least one of the RTU points referenced needs to exist.

For this reason, the SCADAPack E RTU configuration with defined points needs to be loaded prior to loading and executing the user ISaGRAF application(s).

10 ISaGRAF Analog I/O Boards / Representation & Conversion

Analog Input and Output Boards can have Integer or Real (floating point) ISaGRAF variables attached. Both integer and real ISaGRAF analog variables are represented in 32-bit format. The SCADAPack E RTU data interface to these boards is accomplished via point properties in the RTU point database. In addition to direct variable data mapping, ISaGRAF conversion tables may be attached to any ISaGRAF analog Input/Output variable. Conversion table functions are applied after the following conversion rules are applied:

- An ISaGRAF integer variable attached to an Analog Input board receives a 32-bit signed value from the point's Current Integer Value property. The type of DNP3 object selected for this point does not affect the value presented to ISaGRAF (i.e. An analog point's value may have a conversion applied to a 16-bit DNP3 analog object, but the conversion is not applied to the value reported to ISaGRAF).
- An ISaGRAF real analog (floating point) variable attached to an Analog Input board receives a 32-bit floating point value from the point's Current Engineering Value property. The type of DNP3 object selected for this point does not affect the value presented to ISaGRAF
- An ISaGRAF integer variable attached to an Analog Output board sends a 32-bit signed value to the point's Current Integer Value property. A conversion between integer and engineering value is also carried out according to an integer to engineering conversion formula. The type of DNP3 object selected for this point does not affect the value presented from ISaGRAF
- An ISaGRAF real analog (floating point) variable attached to an Analog Output board sends a 32-bit signed value to the point's Current Engineering Value property. A conversion between engineering and integer value is also carried out according to engineering to integer conversion formula. The type of DNP3 object selected for this point does not affect the value presented from ISaGRAF.
- An ISaGRAF integer variable attached to a Counter Input board receives a 32-bit signed integer value representing the 32-bit unsigned count value of an RTU counter point. As ISaGRAF does not handle unsigned integers, the user's ISaGRAF application needs to deal with the case of a negative Count value.
- An ISaGRAF real analog (floating point) variable attached to a Counter Input board receives an unsigned numeric value representing the 32-bit unsigned count value of an RTU counter point. The conversion applied may result in less accuracy of the count value as the ISaGRAF single precision (32-bit) floating point value will only provide 6 significant digit resolution.

Due to the arrangement of the SCADAPack E RTU data mapping for physical I/O, ISaGRAF input variables attached to physical I/O points on Digital Input and Analog Input I/O boards (e.g. `rtuNNdi`, `rtuNNai` where NN represents an integer) read the state or value of the physical Input points. ISaGRAF output variables attached to physical I/O points on Digital Output and Analog Output I/O boards (e.g. `rtuNNdo`, `rtuNNao`) control or write to the physical output points. To read the status of physical output points, attach ISaGRAF Input variables to Digital Output Status and Analog Output Status I/O boards (e.g. `rtuNNdos`, `rtuNNaos`).

11 ISaGRAF OPERATE Function

ISaGRAF provides a built-in *Operate* Function Block for manipulating I/O variables. Specifically, the SCADAPack E RTU provides an interface for initializing values of ISaGRAF Boolean input and analog input variables. This is typically used to load default values, or override variable values that are normally inputs into ISaGRAF from other sources (such as SCADA Master systems). The RTU also uses this mechanism to allow a user application to preset or reset Counter points.

The variable used in an *Operate* Function Block needs to be of ISaGRAF “Input” variable type, and needs to be attached to an ISaGRAF Input Board in order to override its value.

Furthermore, the operation simulates an external input change. That is **the input variable value does not update until the next I/O board update at the start of the ISaGRAF scan!** Consequently, access to the input variable after the OPERATE but before the end of scan will reflect the original value of the point (the variables value prior to the OPERATE). The input variable will have the “Operated” value following the start of the next ISaGRAF scan (assuming that the external source for the Input Board has not updated it again in the mean time).

The type of SCADAPack E RTU point manipulated by the OPERATE function is implied by the type of ISaGRAF Input Board it is attached to.

The Operate function block parameters are defined as follows:

- For BOOLEAN INPUT variables attached to Digital Input RTU boards:

Input parameters

IO name of boolean input variable
 Funct Not Used
 Arg state to initialize: 0=false, 1=true

Output parameter

Q 1=success, 0=fail

- For INTEGER ANALOG INPUT variables attached to Analog Input boards:

Input parameters

IO name of analog input variable
 Funct Not Used
 Arg value to initialize

Output parameter

Q 1=success, 0=fail

- For INTEGER ANALOG INPUT variables attached to Counter Input boards

Input parameters

IO name of counter input variable
 Funct Not Used
 Arg value to initialize

Output parameter

Q 1=success, 0=fail

- For REAL (FLOATING POINT) ANALOG variables:

Input parameters

IO name of real analog input variable
 Funct power for initialize calculation

Arg value to initialize calculation

Output parameter

Q 1=success, 0=fail

The Operate Function calculates the initialized value using the following equation:

$$\text{variable_initalised_value} = \text{Arg} / 10^{\text{Funcnt}}$$

For ISaGRAF real analog variables attached to an Analog Input Board, the value derived from the above calculation initializes the 32-bit floating Engineering Value of the RTU point corresponding to the board/channel address to which the variable is connected.

The ISaGRAF Boolean variable needs to be attached to a Boolean Input Board. The state with which the variable is to be initialized is set in the RTU binary object corresponding board/channel address at which the variable is connected. Similarly, the ISaGRAF integer variable needs to be attached to an Analog Input Board. The *OPERATE* function argument 'Arg' initializes the Current Value property of the point corresponding to the board/channel address to which the variable is connected. This also applies to ISaGRAF integer analog variables attached to Counter Input Boards. To reset a counter, use 'Arg' with a value of 0. [Figure 9.1](#)^[67] illustrates a sample use of the OPERATE function on a real analog input variable.

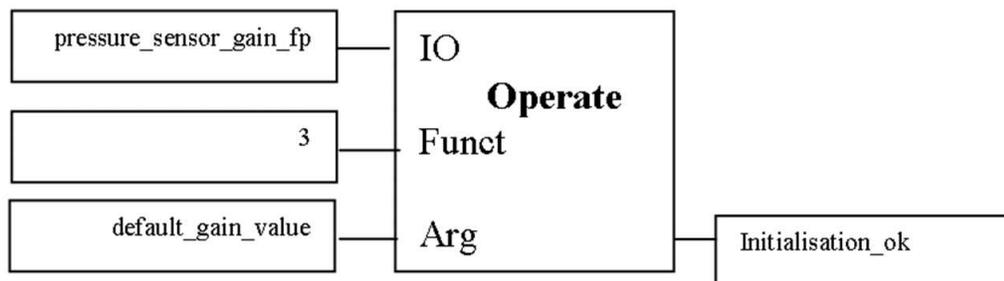


Figure 9.1: Example Operate command

The *Operate* function block can be used to manipulate RTU *physical* input point types providing that the input point is **not** associated with local physical I/O, or remote points that are mapped via the Data Concentrator.

11.1 Using ISaGRAF Operate Functions with Conversion Tables

Care should be taken when using *Operate* function block to set analog input variable values when the ISaGRAF analog variable has a conversion table attached to it.

Generally conversion tables cause the value held in the RTU point database, and the ISaGRAF internal value to be different. Conversion tables can be applied to both analog input and analog output variables, but *Operate* functions are only applicable for analog input variables.

The value passed to the *Operate* function, or resulting from the *Operate* function calculation in the case of Real analog input variable types, is the value set in the RTU point database. This value is then applied to a conversion table (if one is attached to the analog) resulting in a different value appearing in the ISaGRAF variable i.e. conversion table value adjustment is performed AFTER the Operate value is loaded into the RTU point.

Using Operate Functions with Conversion Tables should be avoided.

12 Remote ISaGRAF Access

The following section detail how ISaGRAF applications and debugging can be managed remotely from the SCADAPack E RTU.

- [Application File Transfer](#)^[70]
- [DNP3 Communications](#)^[71]
- [TCP/IP Communications](#)^[74]

12.1 Application File Transfer

ISaGRAF applications (Intel format compiled TIC code) may be loaded from the Workbench compiled APPLI.X8M files, into the RTU file system. This file transfer may be done using:

- DNP3 protocol - File transfer
- FTP – TCP/IP File transfer
- IEC60870-5 - File transfer

The ISaGRAF workbench uses APPLI.X8M files to download to the SCADAPack E controller. These files can usually be found in the workstation's ISaGRAF workbench application directory tree.

E.g. c:\isawin\ESeries\apl\appl-name\appli.x8m

SCADAPack E Configurator ISaGRAF Application Transfer

SCADAPack E Configurator provides a simple method for downloading and activating ISaGRAF applications to the RTU.

Use the  icon on SCADAPack E Configurator and follow the dialog to select the appropriate APPLI.X8M file and desired ISaGRAF target kernel.

Manual ISaGRAF Application Transfer

When performing a manual transfer of the ISaGRAF application to the RTU, the APPLI.X8M file(s) needs to be renamed to ISA11 and/or ISA21 before restarting the ISaGRAF kernel target. For the SCADAPack E RTU, the ISaGRAF application files needs to reside in the root directory of the C: drive in order to be executed.

The first and second ISaGRAF Kernel Targets use filenames ISA11 and ISA21 respectively.

ISaGRAF Kernel	RTU ISaGRAF Application Filename
1	ISA11
2	ISA21

Details of performing this file transfer are provided in the *SCADAPack E Operational Reference* manual.

After transferring a new file, the appropriate ISaGRAF Kernel may be restarted by using the appropriate buttons on the SCADAPack E Configurator **Logic / ISaGRAF** page or through command-line commands:

- "RESTART ISAGRAF" - Restarts ISaGRAF Kernels 1 and 2
- "RESTART ISAGRAF 1" - Restarts ISaGRAF Kernel Task 1 only
- "RESTART ISAGRAF 2" - Restarts ISaGRAF Kernel Task 2 only

12.2 DNP3 Communications

The SCADAPack E RTU supports remote ISaGRAF Workbench communications via the DNP3 protocol.

This is achieved using DNP3 Virtual Terminal objects for transporting ISaGRAF Workbench communications via DNP3. See [Figure 11.2](#)^[73] for an illustration of the communication process between an ISaGRAF Workbench running an engineering terminal (e.g. a PC) and a remote RTU.

At the RTU, virtual terminal requests received for the ISaGRAF kernels are re-directed to an ISaGRAF communications task. Responses from the communications task are re-directed to DNP3 virtual terminal objects. Separate ISaGRAF communication tasks provide Local (serial) port and Remote (DNP3 Virtual Terminal) port connections to the ISaGRAF Kernels. The task responsible for handling a remote ISaGRAF communication is opened even if there is no local RTU serial port configured for ISaGRAF. However, if a local RTU serial port is not configured for ISaGRAF, the ISaGRAF local port communication tasks is not started.

A simultaneous connection from a local and remote ISaGRAF Workbench is possible on the same RTU. Each Workbench may connect to either of the RTU ISaGRAF Kernel tasks or both may connect to the same RTU ISaGRAF Kernel task if required.

SCADAPack E Configurator services its inter-task communications interface in order to receive a Workbench packet, encodes it into a DNP Object 112 "Virtual Terminal Output Block" and transmits the request to the remote RTU. Upon receipt of a DNP3 response from the RTU, the application reassembles the message for the Workbench and passes it on to the ISaGRAF task.

Setting up ISaGRAF communications via DNP3

SCADAPack E Configurator allows configuration of the physical communication medium interfaces which needs to be done prior to commencing workbench communications. E.g. serial communication, PSTN dial-up, GSM dial-up, radio, etc. In the case of PSTN or GSM Modem communications, the user should instruct SCADAPack E Configurator to set up the connection by communicating with the RTU (e.g. **General / Controller Status** refresh) before launching the Workbench application debug session.

A DNP3 communication channel from the ISaGRAF Workbench to the SCADAPack E RTU can be established via SCADAPack E Configurator as follows:

SCADAPack E Configurator initially should be launched and a connection established to the RTU. With an active SCADAPack E Configurator to RTU connection established, ISaGRAF should be launched from the SCADAPack E Configurator toolbar. This action will initiate the creation of an ISaGRAF Debugger / DNP3 logical link to the SCADAPack E Configurator software.

To finalize the communication channel from the ISaGRAF Workbench debugger to the RTU, the "Configurator" option should be selected from the ISaGRAF PC-PLC link parameters dialog as shown in the figure below. No additional setup is required within this dialog.

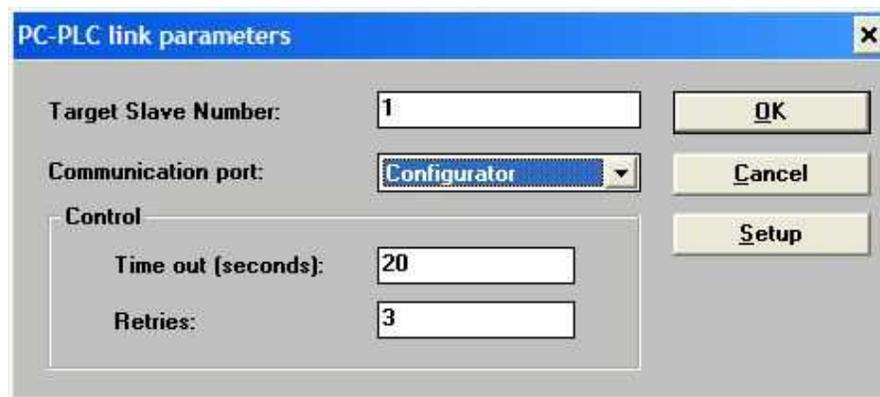


Figure 11.1: Configurator PC-PLC Link Option

Launching the ISaGRAF debugger at this point will connect the Workbench to the RTU via the 'SCADAPack E Configurator. – RTU' communication channel established earlier.

A connection may be established to an intermediate RTU as a means to communicate with one of many RTUs on an RTU sub-network (i.e. the connected RTU is used as a router), the ISaGRAF Workbench connection uses the currently active 'SCADAPack E Configurator - Router RTU' connection.

Due to the slower nature of remote ISaGRAF debug communications via Wide Area links (compared with local direct connections to an RTU) it may be necessary to increase the Workbench Debugger *Timeout and Retries* settings. From the Workbench select "Debug" "Link Setup". Suggested settings for these parameters are:

- Time out = 20 (seconds)
- Retries = 3

ISaGRAF Workbench / DNP3 Interaction

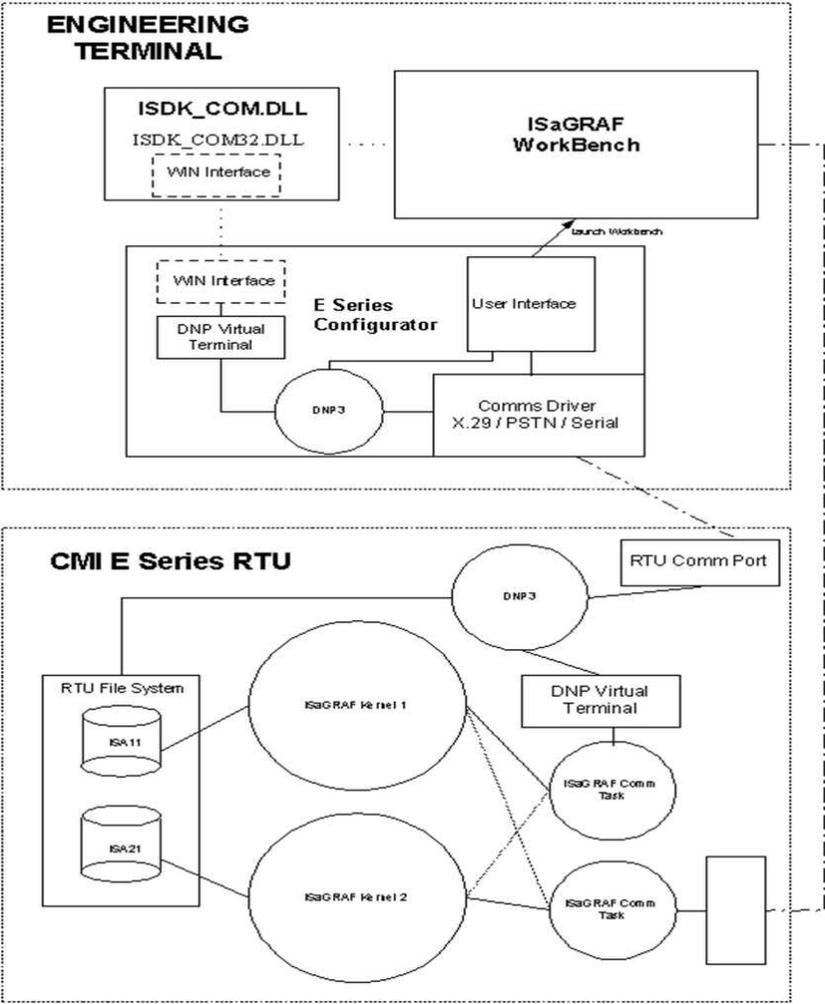


Figure 11.2: Remote ISaGRAF Communication

12.3 TCP/IP Communications

The SCADAPack E RTU provides an ISaGRAF Communications Server for TCP/IP.

Either the RTU Ethernet or a serial PPP interface may be used for ISaGRAF workbench communications using TCP/IP.

To establish a TCP/IP connection between the ISaGRAF Workbench and the RTU, the **ISaGRAF/TCP** service on the RTU needs to be enabled through the *TCP* page on SCADAPack E Configurator software. Refer to the *SCADAPack E Configurator User Manual* for details.

The ISaGRAF Workbench PC-PLC link parameters needs to be setup for an Ethernet connection. Setting up the Ethernet connection within the ISaGRAF Workbench is described below.

- [ISaGRAF Workbench Ethernet Settings](#)⁷⁵
 - [ISaGRAF TCP/IP Communications Server](#)⁷⁶
-

12.3.1 ISaGRAF Workbench Ethernet Settings

The ISaGRAF Workbench can be configured to connect to the RTU a TCP/IP communications by selecting the "ETHERNET" option from the PC-PLC link parameters dialog and entering the IP address of the remote RTU by selecting the 'Setup' button as shown in the figure below.

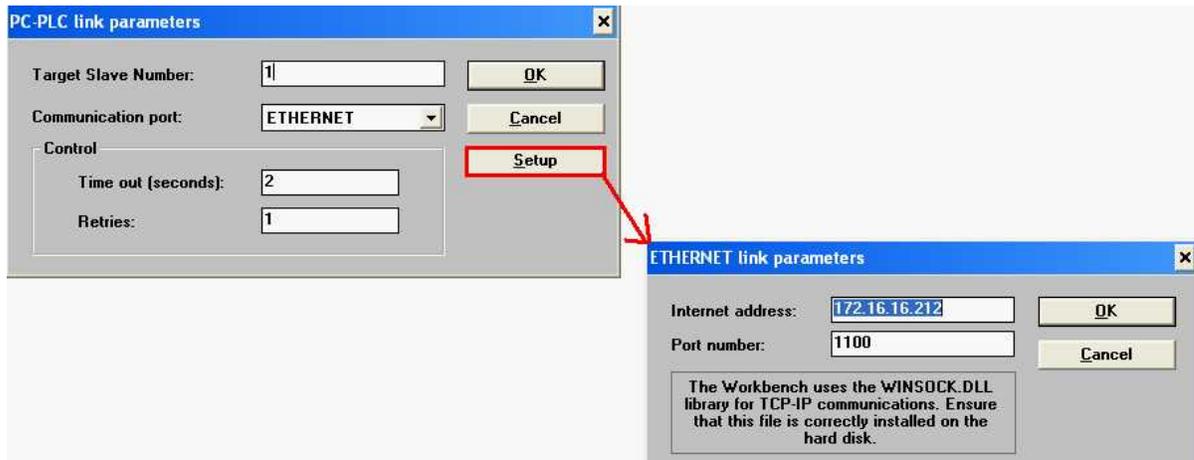


Figure 11.3: ISaGRAF Workbench Ethernet Settings

The function of each field in the ETHERNET link parameters dialog is described below:

Internet address IP address or HOST name of the SCADAPack E RTU

Port number **1100** for the SCADAPack E RTU. This is TCP Port number at which a socket to the ISaGRAF Communications Server will be open.

Using Ethernet communications for ISaGRAF connections requires the RTU has **ISaGRAF/TCP** service enabled (see SCADAPack E Configurator **TCP/IP** page)

At any given time, only a single ISaGRAF Workbench TCP/IP connection is permitted to be opened on the SCADAPack E RTU.

The "ETHERNET" setting may be used for any TCP/IP network connection to the SCADAPack E RTU. For example the RTU may use a PPP serial connection, mobile IP, etc. Use ISaGRAF "ETHERNET" communication port for every TCP/IP connection type to the RTU.

12.3.2 ISaGRAF TCP/IP Communications Server

Communications to the SCADAPack E RTU's ISaGRAF TCP/IP communications server (TCP ISaGRAF Comm Task shown in [Figure 11.4](#)^[76] below) may be established via the RTU's Ethernet interface or a serial PPP interface.

This requires the RTU has **ISaGRAF/TCP** service enabled (see SCADAPack E Configurator **TCP/IP** page)

Only a single ISaGRAF TCP/IP session may be opened at a time regardless of which RTU TCP/IP interface is being used. The ISaGRAF communications server may access either of the RTU's ISaGRAF Targets, set by the "Target Slave Number" as shown above in the "PC-PLC Link Parameters" dialog.

Other ISaGRAF communications link(s) may be used at the same time as one TCP/IP communications link.

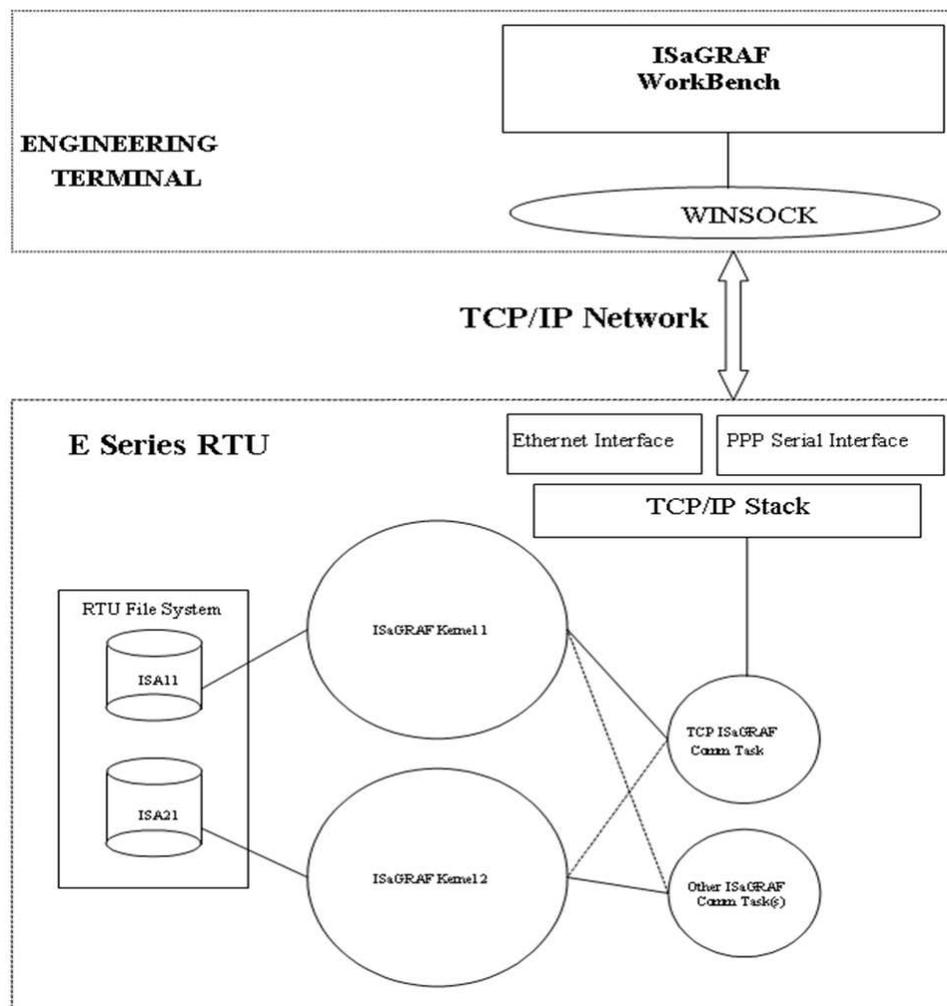


Figure 11.4: TCP/IP ISaGRAF Communication

13 ISaGRAF Troubleshooting & Error Codes

This section describes the different sources of ISaGRAF errors. These errors are accessible from the Workbench debugger or ISaGRAF *SYSTEM* call.

- [Error Types](#)⁷⁸
 - [Error Summary](#)⁷⁹
 - [Error Descriptions](#)⁸²
-

13.1 Error Types

The three different kinds of errors that can be detected by the RTU ISaGRAF target are:

SYSTEM Errors

These are likely due to target software or hardware, rather than due to application settings or to program execution. Try a hardware reset of the RTU and try to run other applications. Errors affecting system operation should be reported to Schneider Electric.

APPLICATION Errors

These are due to application parameters, size or content. These errors should disappear when loading a known and previously validated application. If the problem still appears, it becomes a system error as listed above.

PROGRAM Errors

These are due to a particular sequence of the loaded user application program. These kinds of errors should disappear when the application is started in cycle by cycle mode, or when the offending program is stopped.

Error code and Error argument are sent to the ISaGRAF error routine. The workbench will interpret errors and display text equivalent to that shown below.

A number or variable may appear in brackets [x] before the error text. Depending on the context of the application the debugger may be able to display the name of the object (variable or program) where the error comes from. In some cases a number may be displayed whose meaning is error dependent.

ISaGRAF target error information is stored and can be accessed later from the workbench debugger or the *SYSTEM* call from within an application.

13.2 Error Summary

The following table details the ISaGRAF errors that may be reported on SCADAPack E RTU's. For more information see the ISaGRAF Target User's Guide.

The Errors may be reported via a number of means: ISaGRAF Workbench message, RTU Diagnostic output, RTU System Error code, SCADAPack E Configurator Controller Status page.

Table 12.1: ISaGRAF Error List

Code	Message	Type
1	Cannot allocate memory for run time data base	system
2	Incorrect application data base (Motorola / Intel)	application
3	Cannot allocate communication mailbox	system
4	Cannot link kernel data base	system
5	Time-out sending question to kernel	system
6*	Time-out waiting answer from kernel	system
7	Cannot init communication	system
8	Cannot allocate memory for retained variables	application
9*	application stopped	application
10	too many simultaneous N or P actions	application
11	too many simultaneous setting actions	application
12	too many simultaneous resetting actions	application
13	unknown TIC instruction	application
14	invalid type cast	application
16	cannot answer read data request	system
17	cannot answer write data request	system
18	cannot answer debugger session request	system
19	cannot answer modbus request	system
20	cannot answer debugger application request	system
21	cannot answer debugger	system
23	unknown request code	system
24*	Ethernet communication error	system
25*	communication synchro error	system
28	cannot allocate memory for application	application
29	cannot allocate memory for application update	application
30	unknown OEM code	application
31	cannot init Boolean input board	application
32	cannot init analog input board	application

Code	Message	Type
33	cannot init message input board	application
34	cannot init Boolean output board	application
35	cannot init analog output board	application
36	cannot init message output board	application
37	cannot input Boolean board	application
38	cannot input analog board	application
39	cannot input message board	application
40	cannot output Boolean output variable	application
41	cannot output analog output variable	application
42	cannot output message output variable	application
43	cannot operate Boolean variable	application
44	cannot operate analog variable	application
45	cannot operate message variable	application
46	cannot open board	application
47	cannot close board	application
50	cannot overwrite Boolean output variable	program
51	cannot overwrite analog output variable	program
52	cannot overwrite message output variable	program
61	unknown system request code	program
62*	sampling period overflow	program
63	user function not implemented	application
64	integer divided by zero	program
65	conversion function not implemented	application
66	function block not implemented	application
67	standard function not implemented	application
68	real divided by zero	application
69	invalid operate parameters	application
72	application symbols cannot be modified	application
73	cannot update: different set of Boolean variables	application
74	cannot update: different set of analog variables	application
75	cannot update: different set of timer variables	application
76	cannot update: different set of message variables	application
77	cannot update: cannot find new application	application

* These ISaGRAF errors will NOT generate an RTU System Error Code in system analog point 50020.

Other listed ISaGRAF errors DO generate an RTU System Error.

13.3 Error Descriptions

1	Cannot allocate memory for run time data base	<i>system</i>
	No memory available. Check hardware configuration and application variable memory requirements.	
2	Incorrect application data base (Motorola/Intel)	<i>application</i>
	The application file transferred or backed up is not correct. This error appears if the application is generated for MOTOROLA and transferred to an RTU (which uses an Intel byte format processor).	
3	Cannot allocate communication mailbox	<i>system</i>
	This error is produced by the ISaGRAF communication tasks if it cannot allocate space for inter task communication.	
4	Cannot link kernel data base	<i>system</i>
	This error is produced by the communication task if it cannot find a kernel running with the slave number specified in the comm task startup.	
5	Time-out sending question to kernel	<i>system</i>
	The communication task cannot send a request to the ISaGRAF kernel. The kernel is probably not running.	
6*	Time-out waiting answer from kernel	<i>system</i>
	The communication task cannot receive an answer from the ISaGRAF kernel. The kernel is probably not running.	
7	Cannot init communication	<i>system</i>
	This warning is produced when the ISaGRAF communication layer cannot initialize the physical link. This does not stop the ISaGRAF application running, but it cannot communicate.	
8	Cannot allocate memory for retained variables	<i>application</i>
	ISaGRAF cannot manage retained variables. The reasons for such a problem may be: - the size of memory specified for each block is not sufficient Verify the syntax of the 'retain variable' parameter, or try with a reduced number of Retained Variables.	
9*	Application stopped	<i>application</i>
	This warning is produced every time the application is stopped from the debugger	
10	Too many simultaneous N or P actions	<i>application</i>
	This error occurs if one of the PLC cycles has to execute too many non stored, pulse actions or cyclic blocks. It is possible to locate this in CC mode. The maximum number of simultaneous actions is 2 + 4 per SFC program.	
11	Too many simultaneous setting actions	<i>application</i>
	This error occurs if one of the PLC cycles has to execute too many setting actions (executed when an SFC step becomes active). Proceed as for error 10.	
12	Too many simultaneous resetting actions	<i>application</i>
	This error occurs if one of the PLC cycles has to execute too many resetting actions (executed when an SFC step is de-activated). Proceed as for error 10	

13	Unknown TIC instruction	<i>application</i>
	The kernel has detected something wrong in the application TIC code (target independent code) of a program. There are two possible explanations: - the target only has a reduced set of instructions, and your application uses a non authorized instruction or variable type - another task has written into application code spare. Try to locate in CC mode & check that no I/O interface has incorrect parameters	
14	Invalid type cast	<i>system</i>
	The kernel has detected an invalid type casting operation in the application (eg. type casting a Boolean to a Boolean in ST program)	
16	Cannot answer read data request	<i>system</i>
	A communication error is detected answering specific ISaGRAF Modbus request function code 18 (file read). Check connection and system configuration on both target and master sides. Not supported by SCADAPack E RTUs	
17	Cannot answer write data request	<i>system</i>
	A communication error is detected answering specific ISaGRAF Modbus request function code 17 (file write). Check connection and system configuration on both target and master sides. Not supported by SCADAPack E RTUs	
18	Cannot answer debugger session request	<i>system</i>
	A communication error is detected answering a debugged request. Check connection and system configuration on both target and master sides.	
19	Cannot answer Modbus request	<i>system</i>
	A communication error is detected answering a Modbus request. Check connection and system configuration on both target and master sides	
20	Cannot answer debugger application request	<i>system</i>
	A communication error is detected answering a debugger request. Check connection and system configuration on both target and master sides.	
21	Cannot answer debugger	<i>system</i>
	A communication error is detected answering a debugger request. Check connection and system configuration on both target and master sides.	
23	Unknown request code	<i>system</i>
24*	Ethernet communication error	<i>system</i>
	This appears each time a debugger TCP/IP connection is closed. In this cast the system is working OK. Otherwise it means an Ethernet communication error is detected. Check connection and system configuration on both target and master sides. The following error may be given: 024:001 = error while sending or receiving 024:002 = error creating the socket 024:003 = error binding or listening on the socket 024:004 = error while accepting a new client	
25*	Communication synchro error	<i>system</i>
	A synchronization error is detected framing a debugger communication request, possibly due to missing bytes or CRC error. Check connection and	

	system configuration on both target and master sides.	
28	Cannot allocate memory for application	application
	No application memory available. Check the hardware memory configuration and size of the application.	
29	Cannot allocate memory for application update	application
	No working memory available. Check the hardware memory configuration and size of the application. Large applications may prohibit the use of on-line updating.	
30	Unknown OEM code	application
	The application is using a board whose manufacturer code is not recognized by the target. The OEM code for SCADAPack E RTUs is 90 (5A hex). Check the I/O connection in the workbench and use 'VIRTUAL' board attribute to locate the incorrect board. The workbench library may not correspond to the target versions.	
31	Cannot init Boolean input board	application
	A Boolean input board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	
32	Cannot init analog input board	application
	An analog input board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	
33	Cannot init message input board	application
	A message input board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	
34	Cannot init Boolean output board	application
	A Boolean output board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	
35	Cannot init analog output board	application
	An analog output board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	
36	Cannot init message output board	application
	A message output board init has failed. Check the I/O connection in the workbench and the parameter settings of the boards.	

37	Cannot input Boolean board	application
	An error has been detected while refreshing a Boolean input board. Check the I/O connection in the workbench and board parameters.	
38	Cannot input analog board	application
	An error has been detected while refreshing an analog input board. Check the I/O connection in the workbench and board parameters.	
39	Cannot input message board	application
	An error has been detected while refreshing a message input board. Check the I/OP connection in the workbench and board parameters.	

40	Cannot output boolean output variable	<i>application</i>
	An error has been detected while updating an output Boolean variable. Check the I/O connection in the workbench and board parameters.	
41	Cannot output analog output variable	<i>application</i>
	An error has been detected while updating an output analog variable. Check the I/O connection in the workbench and board parameters.	
42	Cannot output message output variable	<i>application</i>
	An error has been detected while updating an output message variable. Check the I/O connection in the workbench and board parameters.	
43	Cannot operate Boolean variable	<i>application</i>
	An error has been detected executing an OPERATE call to an analog variable. Verify the OPERATE parameters and I/P board user's note supports operate calls to analog input variables only.	
44	Cannot operate analog variable	<i>application</i>
	An error has been detected executing an OPERATE call to an analog variable. Verify the OPERATE parameters and I/O board user's note supports operate calls to analog input variables only.	
45	Cannot operate message variable	<i>application</i>
	An error has been detected executing an OPERATE call to a message variable. Verify the OPERATE parameters and I/O board user's note does not support operates to message variables	
46	Cannot open board	<i>application</i>
	The application is using a board reference which is unknown in the target. Check the I/O connection in the workbench. The workbench library may not correspond to the target version.	
47	Cannot close board	<i>application</i>
	The application is closing a board reference which is unknown in the target. Check the I/O connection in the workbench.	
50	Cannot overwrite Boolean output variable	<i>program</i>
	Two SFC sequences are writing to the same Boolean output variable in the same PLC cycle. This should be avoided. In case of such a conflict, the priority is given to the highest program in the hierarchy. If the two SFC programs are located at the same level, the result is unpredictable.	
51	Cannot overwrite analog output variable	<i>program</i>
	Two SFC programs are writing the same analog output variable in the same PLC cycle. See error 50 comments.	

52	Cannot overwrite message output variable	<i>program</i>
	Two SFC programs are writing the same message output variable in the same PLC cycle. See error 50 comments.	
61	unknown system request code	<i>program</i>
	A program is using the SYSTEM call with an invalid order code.	
62*	Sampling period overflow	<i>program</i>
	The PLC cycle is longer than specified in the workbench menu. As the RTU has a multitasking operating system, this means there is not enough	

	<p>CPU time to execute a cycle even if the 'current cycle duration' is less than the specified period. There are many possible ways to remove this warning:</p> <ul style="list-style-type: none"> - reduce the number of operations performed at the instant where the warning is detected - reduce number of tokens of valid transitions, optimize complex processing, etc. - reduce communication traffic to give more time to ISaGRAF - use dynamic cycle duration modification to adapt the cycle duration to different process stages - set cycle duration to ZERO to let the ISaGRAF kernel run as fast as possible without any overflow checking 	
63	User function not implemented	<i>application</i>
	A program is using a C function which is unknown in the target. The workbench library may not correspond to the target version.	
64	Integer divided by zero	<i>program</i>
	A program is trying to divide an integer analog by zero. The application should avoid such an event which may have unpredictable effects. When this occurs, ISaGRAF places the maximum analog value as the result. When the operand is negative, the result is inverted.	
65	Conversion function not implemented	<i>application</i>
	A program is using a conversion function which is unknown in the target. The workbench library may not correspond with the target version. When this occurs, ISaGRAF does not convert the value.	
66	Function block not implemented	<i>application</i>
	A program is using a function block which is unknown in the target. The workbench may not correspond to the target version.	
67	Standard function not implemented	<i>application</i>
	A program is using a function block which is unknown in the target although it is supposed to be available on target implementations. Contact Schneider Electric.	
68	Real divided by zero	<i>application</i>
	A program is trying to divide a real analog by zero. The application should avoid such an event.	
69	Invalid operate parameters	<i>application</i>
	A program is passing invalid parameters in an OPERATE call. Check application calling parameters and see I/ Board technical note.	
72	Application symbols cannot be modified	<i>application</i>
	Trying to make an application update, the modified application cannot be started because the symbols are different. One or more variables or instances of function blocks may have been added, removed or modified, compared to the current application.	
73	Cannot update: different set of boolean variables	<i>application</i>
	The modified application cannot be started because some Boolean variables have been added or removed, compared to the current application.	
74	Cannot update: different set of analog variables	<i>application</i>
	The modified application cannot be started because some analog variables have been added or removed, compared to the current application.	

75	Cannot update: different set of timer variables	<i>application</i>
	The modified application cannot be started because some timer variables have been added or removed, compared to the current application.	
76	Cannot update: different set of message variables	<i>application</i>
	The modified application cannot be started because some message variables have been added or removed, compared to the current application.	
77	Cannot update: cannot find new application	<i>application</i>
	The modified application cannot be found in memory, something may have gone wrong during the transfer to the target	

* These ISaGRAF errors indicated by an asterisk will NOT generate an RTU System Error Code in system analog point 50020. Other listed ISaGRAF errors DO generate an RTU System Error.

14 ISaGRAF Unsupported Features

This section outlines the features that are documented in the ISaGRAF Workbench User's Guide or ISaGRAF Target User's Guide that are not currently implemented in the SCADAPack E RTU ISaGRAF Target.

Most other major ISaGRAF features are implemented.

An ISaGRAF target error will usually be produced if an ISaGRAF application or ISaGRAF Workbench accesses an unimplemented feature.

- Mailbox, Task & Keyboard management
-

