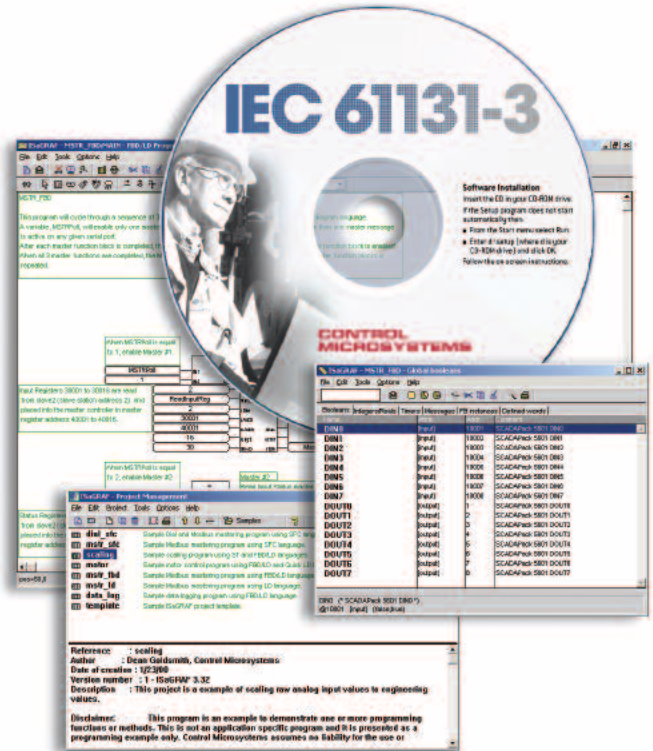# IEC 61131-3

Features:

- Flexible application development environment
- Support for five standard IEC 61131-3 languages:
  - Sequential Function Chart (SFC)
  - Functional Block Diagram (FBD)
  - Ladder Diagram (LD)
  - Structured Text (ST)
  - Instruction List (IL)
- Support for additional Flow Chart (FC) language
- Custom functions for use with SCADAPack and SCADAPack E-series controllers
- Diagnostics and configuration functions for SCADAPack controller serial ports and communication protocols
- On-line debugging and monitoring
- C/C++ applications run concurrently with IEC 61131-3
- Controller-embedded source code
- I/O count-based and Technician Licences

Overview

Control Microsystems unites its industry-standard family of SCADAPack controllers with the world-renowned ISaGRAF IEC 61131-3 automation software; creating a powerful solution for your SCADA system automation needs.

Flexible Development Environment

ISaGRAF provides a user-friendly, flexible environment for developing, debugging and downloading SCADAPack controller logic code; and provides local and remote access to your process by utilizing Intranet and Internet technologies.

Control Microsystems implementation of ISaGRAF is comprised of three main components: the Application Workbench, the Runtime Target (or "Virtual Machine") and the Extensions.

The Application Workbench is a complete programming environment used to develop highly portable applications. It fully supports six automation languages: the five IEC61131-3 languages plus Flow Chart.

Language	Type of Application
Sequential Function Chart (SFC)	Sequential Process
Functional Block Diagram (FBD)	Process Flow
Ladder Diagram (LD)	Electrical Flow
Structured Text (ST)	Textual, Calculative
Instruction List (IL)	Boolean, Simple, Textual
Flow Chart (FC)	Logical, Decision-Based Operations

Language types can be mixed within the same project or in some cases, within the same program. This flexibility enables developers to choose the language that best suits their level of programming proficiency, and programming style, and the nature of the application itself. For example, in a typical application, a custom function could be programmed using ST and then made accessible, as a function block, in a FBD or LD program.

The Workbench also provides tools for editing, debugging, code generation, documentation, library management, archiving, on-line monitoring, off-line simulation and on-line changes.

The Extensions are custom functions for ISaGRAF that have been specifically designed by Control Microsystems for use with our SCADAPack controllers. These functions, which bring SCADA-specific operability to the ISaGRAF workbench, include:

- *master* and *masterip* for sending Modbus protocol master commands over serial and TCP/IP connections
- *dial* and *inimodem* for dial-up modem connections
- Various *Hart* functions for communicating with Hart devices
- *sleep* for low-power mode
- *pida* and *pidd* for analog and digital output-based PID control
- *total* for analog input-based non-volatile totalizing
- *flow* for pulse input (turbine meter)-based non-volatile flow totalizing
- Various *dlog* functions for data logging and retrieving
- *rxstring* and *txstring* for retrieving and transmitting text strings on serial com port

In addition, the Extensions provide SCADAPack diagnostics and configuration functions for serial ports and communication protocols. This allows the application to dynamically monitor comport and protocol statistics, (useful in communication-management algorithms); and configure communication parameters, (ideal for systems that use a variety of protocols and communication parameters).

The Runtime Target is the portable execution engine that runs the developed applications. The portability of the applications generated by the Workbench enables the Runtime Target to execute them without modification; and because only one development environment is required, users leverage the full benefits of a true automation approach.

On-line Debugging and Monitoring

ISaGRAF fully supports on-line debugging and monitoring of program logic. While in the debug mode, code written in any of the graphic-based languages can be viewed in real time, with logic power flow and variable values appearing as they do in the offline mode. Variable values are also available in the dictionary and I/O connection list views.

Two additional features make on-line debugging even more powerful:

- *Spy Lists* permit the user to create, display and save any combination of variables, thereby allowing the focus to be placed on a group of related variables.
- *Spotlight*, ISaGRAF's built-in mini HMI package. Variable values can be displayed in a number of graphical representations including text, bar graph, curve, Boolean icon and bit field. External bitmap files can be imported to furnish a background on which these graphics can be placed; and the entire display can be saved and retrieved at any time.

Concurrent C/C++ Code Execution

For additional programming power, an ISaGRAF application can execute in the SCADAPack controller concurrently with up to 32 independent C/C++ programs. Process data generated in either application can be passed to the other. This means, for example, that the results of complex mathematical algorithms, implemented using C/C++, can be used by the ISaGRAF application to control the process.

In a typical oil and gas application, Control Microsystems RealFLO gas flow computer C/C++ code handles the gas flow calculation and logging functions while a concurrently-executing ISaGRAF application handles the remainder of the well site automation duties; which could include the control of equipment such as pumps and motor starters.

Embedded Source Code

ISaGRAF provides the facility to compress and zip all files associated with a project into a single file and then download this file to the SCADAPack controller. This means that project-descriptive text, comments, modification history and diary files, variable lists and graphic display elements can be kept with the controller. Code developed on one computer can be recovered from a controller and restored to another computer.