



Integrated Automation Software Development Environment

Guide Form Specification





1 Overview4

1.1 Purpose..... 4

1.2 Definitions 4

2 Technical Requirements5

2.1 System Architecture 5

2.2 Licensing 5

2.3 Documentation 6

2.4 Sample System 6

3 Common Components7

3.1 Single Database..... 7

3.2 Object Re-use 7

3.3 User Interface..... 7

3.4 Project based Cross-referencing 7

3.5 Transparent Data Exchange 7

4 Operator Interface & MMI Development8

4.1 Graphics Capabilities 8

4.2 Sources of Data for Object Animation..... 10

4.3 Networking 10

4.4 Input Devices 11

4.5 Item Selection and Data Entry 11

4.6 Command/Scripting Language 11

4.7 Web Publishing (Support for Web Clients) 12

4.8 Scalability 12

4.9 Multi-language Support..... 12

4.10 Development and Run Time system..... 12

4.11 Alarm and Message Handling..... 12

5 Control Programming (PLC, CNC, and PC Control)14

5.1 Logic Editors 14



- 5.2 PC Control - Run Time Engine 14
- 5.3 PC Control - I/O Drivers 14
- 5.4 PC Control - Web Publishing (Support for Web Clients) 14
- 6 Motion Programming..... 15**
- 6.1 Motion Editor 15
- 6.2 Network Support 15
- 6.3 Diagnostic Tools..... 15
- 7 System Management..... 16**
- 7.1 Security 16
- 7.2 Version Management..... 16
- 7.3 Audit Trails 16
- 7.4 Administration 17
- 8 Service and Support..... 18**
- 8.1 Training 18
- 8.2 Installation Services 18
- 8.3 Support..... 18
- 8.4 Software Warranty 18
- 9 Vendor Requirements 19**
- 9.1 Development Life Cycle 19
- 9.2 Product Life Cycle 19
- 9.3 Manufacturer Qualifications 19



1 Overview

1.1 Purpose

The purpose of this paper is to capture the requirements for an integrated automation software development environment. The software development environment must be able to support development of individual applications as well as complete systems that might include any combination of the following:

- Operator Interface (OI) and MMI
- Control (PLC, CNC and PC Control) and
- Motion Control

1.2 Definitions

- Communications Protocol: A formal set of conventions governing the control of Inputs and Outputs between the two communicating processes.
- Ethernet: A very high performance local area network standard providing the two lower levels of the ISO/OSI seven layer reference model, the physical layer and the data link layer.
- MMI: Man Machine Interface. Used to gather data from a wide variety of devices, animate the process/machine graphically based on the data, accept user requests and publish the data to a wide variety of other automation software. Also commonly referred to as HMI or Human Machine Interface.
- Network: An interconnected group of nodes, a series of devices, nodes or stations connected by communications channels.
- OI: Operator Interface. Used to provide a graphic representation of data from a process and to accept user commands to be fed back to the process.
- Operating System: A program that controls the entire overall operation of the system hardware / software.
- PC Control: A control system, similar to a PLC, except that it runs on a standard Windows based industrial computer.
- PLC: Programmable Logic Controller. A device used to automate monitoring and control of an industrial plant. Can be used stand-alone or networked and used in conjunction with an MMI or SCADA system.
- Project: Includes all of the hardware and software components needed for the application. This may include PLCs, Computers, operator interface panels, motion controllers, etc.
- SCADA: Supervisor Control and Data Acquisition. Used in industry to monitor and control plant status and provide logging facilities. SCADA systems are highly configurable, and usually interface to the plant via PLCs.
- TCP/IP: a protocol widely used across Ethernet networks for connecting computers and programmable controllers.
- Web Browser: A client application that provides a user interface via the World Wide Web. Netscape and Microsoft Internet Explorer are two popular examples.





2 Technical Requirements

2.1 System Architecture

The integrated automation software development environment shall have the ability to implement the following:

- A full-featured MMI application
- A full featured Control application
- A full featured Motion application

2.1.1 Operating System

The operating system shall be Microsoft Windows NT v4.0, Windows 2000 or Windows XP Professional.

2.1.2 Hardware Requirements

The system shall be capable of operating on a minimum configuration of:

- 500 MHz Pentium-based workstation
- 128 MB RAM
- 750 MB hard disk space
- TCP/IP Network protocol-based workstation

Optionally, the system shall also be capable of supporting:

- Additional RAM memory
- Larger hard drive
- Support of multiple port serial add-on boards
- High-speed PLC communication cards
- Touch-screen monitor

2.2 Licensing

2.2.1 It shall be possible to license all components in the integrated development environment with one serial number and through one authorization event, whether that be with a hardware key or software authorization.

2.2.2 The licensing mechanism shall support single user licenses as well as site licenses. The site licenses must not require any web or telephone authorization process.





2.3 Documentation

- 2.3.1 The system shall provide complete user documentation, including examples of how to operate the various modules within the system. The documentation must be in electronic format, HTML based with the ability to search for topics by keyword or search for specific text.
- 2.3.2 The online help system in the product needs to be context sensitive such that immediate help is available for the selected functionality.
- 2.3.3 An on-line "help" facility, based upon Windows standard Hypertext, shall provide useful, context-sensitive information on the operation of the package. This help facility shall be capable of being invoked on-line through a point-and-click operation. The "help" facility must also support the ability to perform full text word search, copy and pasting into another applications, print, and use of system fonts and colors.
- 2.3.4 Product Documentation must be available on the vendor's web site for users to download.

2.4 Sample System

The vendor shall provide a set of samples to assist a user in using the software for the first time. Samples shall be available for OI/MMI, Control, and Motion and shall demonstrate most of the features available in the software.





3 Common Components

3.1 Single Database

- 3.1.1 The integrated development environment must support a centralized data manager that will support all of the variables used in a project, so as to avoid data entry duplication. Automatic import/export is not acceptable.
- 3.1.2 If a variable is renamed in one location, all other references to that variable must automatically be updated to reflect the new name of the variable.

3.2 Object Re-use

- 3.2.1 The development system shall provide a repository of preconfigured object templates, which can be dragged-and-dropped into an automation application. These objects can be as simple as a single ladder logic instruction, or as complex as a robotic arm with fully configured ladder logic and HMI animation.
- 3.2.2 The object repository shall be pre-populated with logic instructions, scripting commands, and graphical objects and can be extended by the user by selecting the object in logic or on a graphics panel and dragging it into the repository.
- 3.2.3 The object repository must allow objects to be shared across applications and support a mechanism to distribute the objects to various stations
- 3.2.4 If an object in the repository is modified, all applications that made reference to that object must automatically be updated the next time the application is used.

3.3 User Interface

- 3.3.1 All editors within the development environment must share a single workspace and a common set of tools.
- 3.3.2 The look-and-feel of all editors within the environment must share a similar design philosophy so as to ensure a standardized user interface across the application.

3.4 Project based Cross-referencing

- 3.4.1 The development environment must provide a list of all locations in the project, whether they are in Control, HMI, or Motion, where a particular variable is being used.
- 3.4.2 The environment must maintain this list automatically for all variables in the project so as to eliminate the need to initiate individual searches through the application when this information is required.

3.5 Transparent Data Exchange

- 3.5.1 The software must provide automatic routing of data between all applications in the project, whether they reside on the same hardware (computer, PLC, etc.) or across a network.





4 Operator Interface & MMI Development

4.1 Graphics Capabilities

The development environment must provide a means of creating and displaying color object-oriented graphic displays that will be used by the operator to monitor and control the process.

Real-time values being read from field devices shall be capable of being displayed in a variety of user-configurable formats.

Support for displays larger than the size of the monitor shall be provided. If used, scroll bars shall be provided to allow the user to move to other areas of the display.

The graphic screens need to be based on objects and not individual pixels. The object graphics will consist of an image and image attributes, such as size, color, and position that will define the properties of the object. The user will use a common window or dialog box to change object properties. An object is defined as anything that can be created with drawing tools from within the package, an image imported into the package, or any 3rd party ActiveX control.

4.1.1 Properties Window

A properties window, exposing all properties for an object must live on the workspace. The properties window must support edit functions for any object selected.

4.1.2 Color Support

The development environment shall provide support for all colors supported by the hardware platform being used. For Windows NT/2000/XP based systems, this includes an unlimited choice of colors with 256 colors supported at any one time.

4.1.2.1 Color Palettes

The system must come with a standard, rainbow color palette as well as the ability to define custom colors from the full color spectrum.

4.1.2.2 Color Changes

Color changes must be selectable from editing the individual foreground, background, or edge color property for each object. In addition the user must be able to select as many objects as they wish and choose which property of an object or objects they wish to change. All selected objects with that property must change at one time.

4.1.3 Graphic Objects

At a minimum, the following graphic objects must be supported:

4.1.3.1 Object drawing tools:

- Rectangle/Square
- Rounded Rectangle/Rounded Square
- Circle /Ellipse
- Polygon
- Pie Wedge (wedges of a circle)
- Arc Wedge
- Bezier Curve
- Straight Line
- Polyline (two or more connected line segments)





- BitMap
- Text
- Button

4.1.3.2 Dynamic Tools:

- Alarm
- Trend
- Bar Graph
- Pilot Light
- Time/Data Display
- Indexed Image Display
- Indexed Message Display
- Triggered Message Display
- Numeric Data Display
- Text Data Display

4.1.3.3 Interactive Tools

- Numeric Data Entry
- Text Data Entry
- Combo Box
- List Box
- Push Button
- Word Button
- Illuminated Push Button
- Selector Switch
- Goto Button
- Print Button

4.1.4 Graphic Development Operations

Operations that may be performed on objects or groups of objects must include the following:

- Select/Select All
- Deselect/Deselect All
- Change Color
- Move
- Nudge (move a single pixel at a time)
- Cut
- Copy
- Paste
- Delete
- Duplicate
- Group/Ungroup
- Align (left, center, right and top, middle, bottom)
- Rotate / Flip (right and left, and Horizontal and vertical)
- Send to Back/Bring to Front
- Snap to grid
- Choice of line and fill styles
- Undo
- Space objects evenly (vertical & horizontal)





4.1.5 Multiple-pen chart link

- Must support a minimum of 10 pens per chart
- Displays run time and historical data
- Configurable time span
- Configurable trend direction (left to right and right to left)

4.2 Sources of Data for Object Animation

The animation of the graphics and objects with dynamic properties shall be able to be linked to any of these types of data:

- Data acquired by the system and stored in its memory resident database
- Data acquired by another networked system and stored in its memory resident database
- OPC Data Source
- Variables declared in the command language scripts

4.3 Networking

4.3.1 The system shall be capable of supporting the following Ethernet based Protocols:

Allen-Bradley TCP/IP
GE Fanuc SRTP
Modbus TCP

4.3.2 The system shall be capable of expansion to support Fieldbus communication configurations:

DeviceNet Slave
GE Genius
PROFIBUS Slave

4.3.3 The system shall be capable of supporting the following network Protocols:

Allen-Bradley DF1 & DH
Allen-Bradley SLC500
CTC Serial Driver
GE Fanuc SNP, SNP-X
Honeywell UMC Series
Modbus RTU
Siemens PPI
Siemens MPI
Siemens D SY/MAX

4.3.4 The system shall support OPC (OLE for Process Control), allowing the automation control application to communicate to IO data locally on the same computer or remotely on a different computer. The OLE for Process Control (OPC) shall support all required OPC interfaces in versions 1.0a, 2.0, and 3.0 of the OPC specifications.

The system must support OLE automation of 3rd party applications, allowing, for example, an Excel spread sheet to be created and fully controlled from the MMI application.





4.4 Input Devices

The system shall support a variety of input devices, including:

- Mouse or Trackball
- Touch-screen
- Keyboard (standard or function-style)

The support for and use of any of these entry devices shall be as provided within Microsoft Windows or by the manufacturer of the input device. Support for multiple simultaneous input devices must also be supported.

4.5 Item Selection and Data Entry

Items on a display shall be available to have their values changed by the operator, as appropriate. Selecting an item for data entry shall be done with the use of a pointing device or keyboard. The selected item will be highlighted by a box surrounding it.

The system must support the following data entry methods:

- Numeric Data Entry
- Slider
- Pushbutton
- Alphanumeric

4.6 Command/Scripting Language

The scripting language used by the system must be Microsoft's VBScript, not Visual Basic "like", C or C "like". Scripts can be simple or complex and allow users to automate operator tasks, and create automations solutions.

4.6.1 Scripting language requirements:

- Application, Panel, and Button scripts
- String manipulation
- COM Port Functions
- SQL Commands
- Extended Math
- SPC support
- Multi-Language Functions
- Robust String Support with overloaded operators
- User Defined Functions
- Extendable via COM objects
- Local variables
- Incorporate and communicate with third party and custom Active X controls.

4.6.2 Multimedia Capability

Support for standard Windows NT/2000 multimedia capabilities, including audio and video, shall be provided.





4.7 Web Publishing (Support for Web Clients)

- 4.7.1 The system shall have the capability to enable viewing application data, and application visualization screens using a web browser such as Microsoft Internet Explore 5.5 or later.
- 4.7.2 In addition to viewing the data and screens, with proper access privileges, the web clients should have full control over the application, including the ability to edit data and move between panels using the graphic representations.

4.8 Scalability

- 4.8.1 The same development environment shall be used for developing Operator Interface and MMI applications.

4.9 Multi-language Support

- 4.9.1 The HMI system shall allow the configuration of multi-language support. The system shall dynamically translate languages at run time. The system shall allow the current language to be change based on an input condition.

4.10 Development and Run Time system

The system shall include a run time engine for execution of automation applications.

The development system is a tool, which fully integrates control and visualization. This tool shall be a Windows base environment. The development environment must share the same set of tools, providing a consistent user interface throughout the development process. The system shall provide drag-and-drop capabilities between tools and editors.

4.11 Alarm and Message Handling

The system shall be capable of detecting alarm conditions based on the states and values of the various sensed variables. The alarm conditions shall be detected even if the variables causing alarms are not currently on the display. Alarms can be filtered based upon location, priority and other user-selectable criteria. When a new alarm is detected, all client machines will immediately see the new alarm. If the alarm is acknowledged on one machine then all machines will see that it was acknowledged.

Alarms will be used to report potentially harmful process conditions requiring a response. Typically when a process value exceeds the pre-defined limits. An example would be a tank's level that is too high is an alarm condition to which an operator must respond. Messages are to report non-critical information that does not reacquire a response. The user must be able to enter alarm limits at configuration time. Alarm limits are expressed in engineering units.

4.11.1 Alarm Types

Analog input or alarm variables shall have the following alarm types:

- High High
- High
- Normal
- Low
- Low Low
- Minor % deviation
- Major % deviation
- Minor value deviation





- Major value deviation
- Time rate-of- change
- Event

Digital input variables shall have the following alarm types:

- None
- When on

4.11.2 Alarm Groups

In order to logically divide an application into smaller units, the system shall allow for unlimited, named individual alarm groups to be defined. All alarm groups must be accessible by each tag.

Each alarm group must be able to support an area where you can associate a graphic screen for the alarm.





5 Control Programming (PLC, CNC, and PC Control)

5.1 Logic Editors

- 5.1.1 The development system must provide the tools required to develop automation control projects that include PLC, CNC, and PC Control. The logic develop environment shall allow the creation of programs using standard IEC 61131-3 programming languages in a graphic environment.
The development system must fully integrate control and visualization. The development environment must share the same set of tools, providing a consistent user interface throughout the development process. The system shall provide drag-and-drop capabilities between tools and editors.
- 5.1.2 Both online and offline editors shall be provided.

5.2 PC Control - Run Time Engine

The system shall include a run time engine for execution of automation applications.

The runtime engine must support the 5 IEC 61131-3 languages (Ladder Diagram, Structured Text, Instruction List, Function Block Diagram, and Sequential Function Charts) and execute the same logic on a wide range of hardware platforms running Windows CE, Embedded NT, Windows NT/2000/XP, and VenturCom's Windows NT hard real-time extension (RTx).

5.3 PC Control - I/O Drivers

- 5.3.1 The system shall be capable of supporting the following Ethernet based I/O:
- GE Fanuc Ethernet Global Data
 - Modbus TCP
- 5.3.2 The system shall be capable of supporting Fieldbus communication configurations (with additional interface hardware):
- GE Fanuc Series 90-30 I/O
 - DeviceNet Master
 - PROFIBUS Master
- 5.3.3 The system shall be capable of supporting the following network Protocols:
- Modbus RTU Slave

5.4 PC Control - Web Publishing (Support for Web Clients)

- 5.4.1 The system shall have the capability to enable viewing application data, application visualization screens, and application logic using a web browser such as Microsoft Internet Explore 5.5 or later.
- 5.4.2 In addition to viewing the data and screens, with proper access privileges, the web clients should have full control over the application, including the ability to edit data and move between panels using the graphic representations.





6 Motion Programming

6.1 Motion Editor

The development environment must support all programming, monitoring, and debugging tools for the motion application. The motion editor must also support the ability to configure any local I/O associated with the Motion controller, if required.

The Motion Editor must support a series of application wizards to provide step-by-step instructions to guide first-time users through the configuration. The wizards must support:

- Motor and Controller selection and configuration
- Configuration and testing of communications
- Axis configuration
- Canned Homing and frequently used motions to simplify application development

The Motion Editor must also provide a mechanism for experienced motion developers to develop applications without the need for the application wizards.

6.2 Network Support

The following I/O networks must be supported in order to allow for I/O expansion and integration with other machine systems:

- PROFIBUS
- DeviceNet
- ASCII serial
- Modbus RTU

6.3 Diagnostic Tools

The motion programmer must support a series of debugging/diagnostic tools to assist with maintaining the application. At a minimum, these must include:

- Histogram for trending system variables such as torque, velocity and following error
- Data monitoring tools for watching non-sequential data or system values





7 System Management

7.1 Security

- 7.1.1 The security system shall control access to an automation application and control the operations that are permitted within that application.
- 7.1.2 Access to administrative functions needs to be limited to those with Administrative privileges.
- 7.1.3 The system should have an option for enforcing operations based on the highest level of permission for the specific user.
- 7.1.4 The system shall provide option for a unique combination of two distinct identification components (username and encrypted password).

7.2 Version Management

- 7.2.1 At run time an option for limiting user access to application project information will be provided.
- 7.2.2 During development a version control system shall manage up to 10 revisions of the application on the local computer or optionally an unlimited number of versions on a server
- 7.2.3 Automatic Version Numbering of User application Development: The version control system shall automatically increment version numbers as changes are checked into the system.
- 7.2.4 The system shall provide an interface for reporting on revision history of a specific user application.
- 7.2.5 The system shall allow the ability to revert to a previous version of a user application without deleting the newer versions in the system.
- 7.2.6 Version numbers shall not be editable
- 7.2.7 The system shall have the facility to add version comments as each version is saved.
- 7.2.8 It needs to be possible to retrieve an older version of a document in a read only format.

7.3 Audit Trails

- 7.3.1 The Audit Trail database shall be configurable for one of Access, SQL Server, or Oracle.
- 7.3.2 The Audit trail database needs to be able to be secured if using SQL Server or Oracle, such that it cannot easily be tampered with.
- 7.3.3 The system shall record user and administrative actions in the system.
- 7.3.4 Changes made by the user to projects (i.e. Check in, Check out, Open, etc.) shall be logged in the audit trail database.
- 7.3.5 The database shall store as a minimum:
 - Date and Time in UTC format
 - User ID and Full user name
 - The action that was performed (Check in, Check out, etc)
 - User comment





- 7.3.6 Audit Reports - Audit reports need to be available in a human readable format, either through a built in reporting tool, or 3rd party tools. The delivery of the reports shall be in a format, such that they cannot be tampered with.
- 7.3.7 The actions that trigger an audit entry should be configurable based on the product type and general actions such as login/logout.
- 7.3.8 Changes to the audit configuration shall be recorded by the audit trail itself, such that if an audit is turned off, it is traceable to the individual that made the change.

7.4 Administration

- 7.4.1 Administration functions shall be available only to those users belonging to the administrator groups, or to those granted permission for the specific administrative function.
- 7.4.2 Administration of the system needs to be possible from any client. The user must have appropriate permissions to access these functions.





8 Service and Support

8.1 Training

8.1.1 Vendor needs to be able to provide both on-site and off-site training.

8.2 Installation Services

8.2.1 The vendor shall have the ability to provide qualified consulting and installation services. These services need to be provided either by the vendor or via certified partners. Services include:

- Site Assessment
- Consultation
- Implementation
- Assistance and turnkey
- Validation services

8.2.2 Global Services

Vendor needs to be able to provide service and support the installations locally worldwide.

8.3 Support

The manufacturer or its authorized representative shall provide complete technical support for all of the products. This shall include headquarters or local training, regional application centers, and local or headquarters technical assistance. A toll-free (800) number shall be available for technical support.

8.4 Software Warranty

8.4.1 Vendor shall provide the option to purchase support contract. This contract is to include:

- Telephone and Email support Mon-Fri 8:00AM to 8:00PM EST
- 24x7 Emergency Support
- Software Upgrades
- 24x7 Access to web based technical and support information

8.4.2 The vendor shall have a location on their web site where users can download software improvements, bug fixes, add-ons, components and so forth.

8.4.3 The vendor shall provide an easy mechanism for upgrading and installing software improvements and for allowing a user to quickly ascertain what improvements have been installed.





9 Vendor Requirements

9.1 Development Life Cycle

- 9.1.1 The vendor must have an established documented development procedure. This procedure must track customer requirements, development risk, and corrective action. The procedure must include a product validation cycle.
- 9.1.2 The vendor must have a formal and documented set of quality assurance procedures that are applied to the engineering design, development, and documentation of the software. The presence of a formal quality assurance department shall be required.
- 9.1.3 The vendor must also demonstrate that it's source code for the product is regularly archived with suitable backup.
- 9.1.4 The vendor must follow a documented configuration management system.

9.2 Product Life Cycle

- 9.2.1 The vendor must have an established product life cycle policy. Documentation of the product life cycle shall be available.

9.3 Manufacturer Qualifications

- 9.3.1 The manufacturer shall have shown a high commitment to product, manufacturing and design process quality. The manufacture shall be ISO9000 registered.
- 9.3.2 The manufacture shall have fully operational quality assurance and quality control programs in place. Complete documentation describing the quality assurance and quality plan shall be available.
- 9.3.3 The integrated software development environment, and all of the corresponding components including HMI, motion, and control, shall be offered by a company who regularly develops and supports this type of software.
- 9.3.4 The software must be generally available (that is, non-custom).
- 9.3.5 Vendor needs to be certified under the ISO 9001-2000 guidelines.

