HMI / SCADA
Proficy* HMI/SCADA - iFIX

Guide Form Specification

GFT-521a
February 2010
# HMI / SCADA – Guideform Specification

## 1 General
1.1 Operating System  
1.2 Configuration  
1.3 On-line Operation  
1.4 Hardware Requirements  
1.5 Documentation  
1.6 On-line Help  
1.7 Sizes

## 2 Data Handling Capabilities
2.1 Data Integrity  
2.2 Database Tag Configuration  
2.3 Database Tag Types  
2.4 Tag Attributes  
2.5 Counting of Tags in Sized Systems  
2.6 Double Precision data and calculations  
2.7 Change Management and Deployment

## 3 I/O Device Communications
3.1 Driver Configuration  
3.2 Display-only Communications  
3.3 Error Detection and Recovery  
3.4 OPC Server Connection  
3.5 Diagnostics  
3.6 I/O & OPC Server Driver Toolkit

## 4 Stand-alone Operation
4.1 Without Networking
4.2 Adding a Network

5 Networking and Distributed Operation
  5.1 Configuration and Expansion
  5.2 Local Area Network (LAN) Architecture and Use
  5.3 Error Detection, Recovery, and Diagnostics
  5.4 Data Handling without duplication of data
  5.5 Remote Access
  5.6 OPC and Active X Support
  5.7 Open Database Connectivity (ODBC) Support

6 Microsoft Windows Terminal Server Support
  6.1 Terminal Services Server
  6.2 CITRIX Support

7 Electronic Signatures and Electronic Records
  7.1 Biometric Devices

8 Graphics Capabilities
  8.1 Graphic Creation
  8.2 Graphics Pre-Loading
  8.3 Color Support
  8.4 Graphic Toolbox
  8.5 Reusing Graphic Objects
  8.6 Tag Aliasing and Tag Group support
  8.7 Support Microsoft Excel and Word Documents
  8.8 Documenting Graphic Displays
  8.9 Operator Entry Methods
  8.10 Command/Scripting Language
  8.11 Multimedia Capability
### 8.12 Sample System

### 8.13 Web SCADA Clients

### 8.14 Change Management and Deployment

<table>
<thead>
<tr>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>Alarm and Message Handling</strong></td>
<td>43</td>
</tr>
<tr>
<td>9.1</td>
<td>Alarm Types</td>
<td>43</td>
</tr>
<tr>
<td>9.2</td>
<td>Alarm Priorities and Filters</td>
<td>44</td>
</tr>
<tr>
<td>9.3</td>
<td>Messages</td>
<td>44</td>
</tr>
<tr>
<td>9.4</td>
<td>Alarm Areas</td>
<td>45</td>
</tr>
<tr>
<td>9.5</td>
<td>Alarm Counters</td>
<td>45</td>
</tr>
<tr>
<td>9.6</td>
<td>Alarm Destinations</td>
<td>46</td>
</tr>
<tr>
<td>9.7</td>
<td>Time Stamping</td>
<td>46</td>
</tr>
<tr>
<td>9.8</td>
<td>Auto Alarm Manager</td>
<td>46</td>
</tr>
<tr>
<td>9.9</td>
<td>Alarm Notification and Acknowledgment</td>
<td>47</td>
</tr>
<tr>
<td>9.10</td>
<td>Alarm Summary Display</td>
<td>48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td><strong>Archiving and Reporting</strong></td>
<td>49</td>
</tr>
<tr>
<td>10.1</td>
<td>Data File Handling</td>
<td>49</td>
</tr>
<tr>
<td>10.2</td>
<td>Archive Configuration</td>
<td>49</td>
</tr>
<tr>
<td>10.3</td>
<td>Displaying Archived Data</td>
<td>49</td>
</tr>
<tr>
<td>10.4</td>
<td>Reporting Options</td>
<td>50</td>
</tr>
<tr>
<td>10.5</td>
<td>Display Output</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td><strong>Event Scheduling</strong></td>
<td>51</td>
</tr>
<tr>
<td>11.1</td>
<td>Event-Based Events</td>
<td>51</td>
</tr>
<tr>
<td>11.2</td>
<td>Time-Based Events</td>
<td>51</td>
</tr>
<tr>
<td>11.3</td>
<td>Events with VBA scripts</td>
<td>51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td><strong>Security Management</strong></td>
<td>52</td>
</tr>
<tr>
<td>12.1</td>
<td>Security Areas</td>
<td>52</td>
</tr>
</tbody>
</table>
12.2 Security Manager
12.3 Securing the Windows GUI Environment
12.4 Limiting Login Time
12.5 Automatic and Manual Logging In and Out
13 Recipe Management
13.1 Recipe Configuration
13.2 Recipe Management Functionality
13.3 Recipe Security and Tracking
14 Redundancy
14.1 SCADA server Failover
14.2 LAN Redundancy
15 Software Tools
15.1 Language
15.2 Data Access
15.3 Change Control
16 Languages Supported
16.1 Dynamic Language Switching
17 Services
17.1 Training
17.2 Warranty and Customer Support
17.3 Software Improvements & “Bug” Fixes
17.4 Quality Assurance
18 Vendor Requirements
18.1 Development Life Cycle
18.2 ISO 9001 certified
18.3 Preferred Vendor / Manufacturer
1 General

The vendor is to provide a highly functional, generally available (that is, non-custom) software system for data acquisition, monitoring and control. The following specifications apply:

1.1 Operating System

The operating system shall be Windows XP Embedded, Windows XP Professional, Windows 2003 or Windows Vista. The software shall exhibit strong compliance with Microsoft’s Windows Open Systems Architecture (WOSA) standards, such as in its use of dialog boxes and menus. The system must support running as a service under Windows XP/2003/Vista/XPe, making it independent of the XP/2003/Vista/XPe user login limitations.

1.2 Configuration

The software shall be configurable by any user on the system (from technician to process engineer to manager)

The system shall provide a mechanism for accepting configuration input either directly from the keyboard, via a mouse, or as appropriate, indirectly through ASCII files that are created by an external text editor, spreadsheet or relational database program.

Source code modifications, re-assembly or recompilation must not be required for implementing user-level system changes.

The entire system should be able to be generated automatically through the use of its scripting language.

1.3 On-line Operation

All configuration changes shall be capable of being made on-line, while the system is operating. Data definitions, operator displays, etc. shall be capable of being modified, added or deleted without having to interrupt the data acquisition.

1.4 Hardware Requirements

- The system shall be capable of operating on a minimum configuration of:
- IBM Pentium 3-based 733 MHz Personal or Industrial Computer that runs Windows XP/2003/Vista
- RAM Memory size:
  - Server -- 512 MB RAM
  - Client -- 512 MB RAM
- SVGA and a 100% IBM-compatible, 24-bit graphics card capable of 800x600 resolution and at least 16 million colors.
- 4.0 GB hard disk, with a minimum of 1 GB free hard disk space.
- Parallel printer port
- Serial port
- Mouse or trackball
• DVD drive

Optionally, the system shall also be capable of supporting:

• Network adapter(s): NETBOIS-compatible or TCP/IP-compatible network interface adapter. (For stand alone application no network adapter required).

• Additional main memory

• Larger hard disk

• Additional printer ports

• Support of multiple port serial add-on boards

• High speed PLC communication cards.

• Touch-screen monitor

• Windows XPe based Panels and Tablets

1.5 Documentation

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.

1.6 On-line Help

1.6.1 Program Help

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, add custom comments, bookmark topics, copy and pasting into another application, printing, and use of system fonts and colors.

1.6.2 Custom Help

The system should be capable of incorporating a user-built Windows standard help file (*.hlp) into the graphic displays without any scripting or code. For example, by pressing the Shif+F1 key the mouse will change to a pointer with a question mark (with no scripting). When the user clicks on an object, e.g., a tank, a window displays a corresponding entry in a Windows help file – again with no scripting or custom code.

The system should provide the tools to the developer to convert standard documents and procedures written in any popular word processing program (e.g. Microsoft Word or WordPerfect) into a compiled Windows help file with a table of contents, index and search capabilities.

1.7 Sizes

The system software shall be provided in different sizes to meet various application needs. Ideally, the sizes would be for tag counts of 150, 300, 900 and no limit for systems with I/O attached. These systems shall be able to display data from and write values directly to other nodes without the need to duplicate the data in the local system.
Configurations that are not attached to I/O and do not contain tags shall also be available. These systems shall be available with two types of rights to a tag:

- With rights to read and write to tags in other nodes.
- With rights to only read tags from other nodes.

These rights can be controlled by the developer using security (covered in Section 12).
2 Data Handling Capabilities

No programming, compiling or linking shall be required to configure the system. The database tags must be configurable on-line. That is, new function and database tag assignments can be added while the system is performing data acquisition and control operations.

The process database containing the current value of the data, or tag list, shall be memory-resident and of a design that is appropriate for real-time monitoring and control functions. Its design shall be optimized for speed, memory usage, data integrity and system security. Floating-point arithmetic shall be used in all calculations. This database shall be stored as a standard Windows file on the local or network hard disk and, upon starting the system, this database is loaded into the computer's memory. Only computers physically connected to process equipment shall require a database.

The actual application of monitoring and control functions will be dictated by the definition of the contents of this memory-resident database. Configuration of this database shall be done by a user familiar with automation or instrumentation terms.

2.1 Data Integrity

The software shall provide pre-emptive multitasking to ensure that common Windows actions do not interfere with I/O communications, processing of data, alarming, and the integrity of the real-time and historical data. These common Windows actions include moving a window with a mouse, opening a file, accessing the hard disk, or printing a graphic display. The software shall be written fully 32-bit so that it runs native in the Windows XP/2003/Vista operating system. Emulation using 16-bit software code is not permitted.

2.2 Database Tag Configuration

Various input/output hardware assignments, as well as processing functions, shall be assigned to named tags or “function blocks”. Multiple tags can be tied together to perform more complex functions. During the configuration process, the program shall be capable of checking the tag structures for correct linkages, appropriate names, and so on. The scan processing program shall also be capable of detecting and handling configuration errors at run-time. Any errors encountered shall generate messages to the user.

The user shall be able to perform tag configuration (adding, modifying, deleting, viewing) in several ways, as follows:

- Directly from the graphics editor, so that tags can be configured as graphics are developed.
- Via an interactive spreadsheet-style database builder program that uses a fill-in-the-blank menu methodology. The database builder program shall provide a Microsoft Fluent User Interface for the following editing functions:
  1. Cut/Copy/Paste tags
  2. Duplicate tags
  3. Generate multiple tags from a given pattern
  4. Sort tags
  5. Query tags
6. Display tags in user-configurable formats
7. Via the importation of a CSV text file developed in another program as input for tag creation. The database builder program shall also be able to export the current tag listing for modification by the external program.

For methods 2 and 3 above, the development of the database tags shall be completely independent of the creation of graphics displays. Use of a programming language, such as Visual Basic, C or a C-like language, shall not be required.

Database verification. The package must also allow for database configuration verification. This task will allow for verification of configuration errors on a local database or a database on another node. Errors shall be reported in a dialog box and a user must be able to make the corrections from this dialog box.

Tag Database Editing. The database has to allow for editing from a graphic editor, from within the building of a graphic operator screen, or from within a VBA script. The database editing must be able to be accessed locally or across the network. A node can edit a database on another node while online.

2.3 Database Tag Types
Functions shall be available in the database to support the following tag types:

- Analog Input
- Analog Alarm
- Analog Output
- Analog Register
- Boolean Logic
- Calculation
- Device Control
- Digital Input
- Digital Alarm
- Digital Output
- Digital Register
- Multi-State Digital Input
- Event Action
- Fanout
- Program
- Real-Time Trend
- Text
- Timer
- Totalizer
- PID
- Lead/Lag
• Dead Time
• Ratio/Bios
• Ramp
• On-Off Control
• Signal Selector
• SQL Data
• SQL Trigger
• Pareto
• Statistical Data
• Statistical Control
• User-Created custom

A description of each tag type is given below:

**Analog Input**

This reads an analog value (time, temperature, speed, pressure, level, etc.) either directly from an A/D converter or from a register within an I/O device such as a programmable controller, and automatically scales the raw data to engineering units (seconds, revolutions/minute, pounds/sq. in., degrees, etc.) An Analog Input tag must also support write outs.

**Analog Alarm**

This provides enhanced alarming capabilities to the Analog Input, including alarm suspension, remote acknowledgment, delay time, re-alarm time, close a contact based on an alarm condition, etc. An Analog Alarm tag must also support write outs.

**Analog Output**

This writes an analog value (set point, output, speed, etc.) either directly to a D/A converter or to a register within an I/O device, such as a programmable controller.

**Analog Register**

This accesses multiple analog I/O points for read or write functions with a single I/O tag. The number of addresses is dependent upon the I/O device used.

**Boolean Logic**

This tag-type takes up to eight (8) inputs, typically logical or digital values, and performs Boolean arithmetic on them. The result can then be passed to or used by other tags or applications within the system. The operators must include:

• OR
• AND
• EQUAL
• NOT EQUAL
• NOT
• XOR
• NAND
• Parentheses

**Calculation**

This tag-type takes up to 8 variables or constants and performs an arithmetic calculation on them. The result can then be passed to or used by other tags or applications within the system. The operators must include:

• Add
• Subtract
• Multiply
• Divide
• Parentheses
• Absolute value
• Square root
• Exponentiation
• Natural log
• Base-10 log
• Relational operations (greater than, less than)
• Change floating point values to integers

**Device Control**

This supports 16 digital inputs and 8 digital outputs, each with random (i.e. not necessarily contiguous) addresses. A set of statements within the tag shall provide the ability to perform pattern checking, interlocking, and timing.

**Digital Input**

This senses the logical on/off state of a switch, relay, pushbutton, etc. either directly from the I/O hardware or from a bit within the memory of an I/O device, such as a programmable controller. The value shall be displayed in a user-selected format (0/1, open/close, false/true, etc.) A Digital Input tag must also support write outs.

**Digital Alarm**

This provides enhanced alarming capabilities to the Digital Input, including alarm suspension, remote alarm acknowledgment, delay time, re-alarm time, close a contact based on an alarm condition, etc. A Digital Alarm tag must also support write outs.

**Digital Output**
This sets a logical on/off state in an output relay either directly in the I/O hardware or in a bit within the memory of an I/O device, such as a programmable controller. The value shall be accepted in a user-selected format (0/1, open/close, false/true, etc.) The Digital Output tag must also provide output-reverse handling and the ability to specify an initial cold-start position.

**Digital Register**

This accesses multiple digital I/O points for read or write functions with a single I/O tag. The number of addresses is dependent upon the I/O device used.

**Multi-state Digital Input**

This tag combines up to three (3) digital values and outputs a value of 0-7 based upon the status of the values in sequence (000 = 0, 001 = 1, 010 = 2, 011 = 3, 100 = 4, 101 = 5, 110 = 6, 111 = 7).

**Event Action**

This tag-type monitors the value or alarm conditions of an I/O point and, based upon TRUE/FALSE conditions, performs one of two operations. These operations include starting or stopping the processing of another tag or opening or closing a digital point.

**Fanout**

This tag passes the value it receives to up to five tags within the same database.

**Program**

This function provides the user with a procedural language for sequencing, monitoring, and controlling typical process operations. The programming function shall execute within the database's scan cycle and is separate from the HMI scripting language. Program functions shall include:

- If/Then go to another step
- Wait until a process condition occurs
- If time-out go to another step
- Go to another step
- Set a tag to a value or the value of another tag
- Open/Close a digital tag
- Set Auto/Manual status of a tag
- Set On Scan/Off Scan status of a tag
- Add/Subtract a value to/from a tag
- Print a message
- Call other program blocks as subroutines
- Run other program blocks in parallel
- Stop other program blocks
• Pause or delay a number of seconds
• Play a .WAV sound file
• Run an executable program

The current step being processed must be capable of being displayed on the operator's CRT. In addition, a debug mode shall be provided to facilitate program checkout.

**Real-time Trend**

This tag shall take as its input an analog or calculated value and save up to six hundred (600) values. Averaging of accumulated input values will be provided to extend the amount of time represented by the trend block. The data within the tag can be graphically depicted on operator displays.

**Text**

This function reads or writes text of up to eighty (80) characters from or to a device.

**Timer**

This tag performs a counting operation. It counts in either the up or down direction, from a pre-set value to a target value. Upon reaching the target or time-out condition, a contact may be closed. This tag also supports conditional next block processing. It shall time up to one (1) year. The timer may be started, stopped, reset or resumed based on a sensed condition or operator command.

**Totalizer**

This tag-type maintains a floating point total for values passed to it from other database tags.

**PID (3-term) control**

This tag will provide a modulated output based on the standard PID algorithm. The output will be positional and will be capable of being used as the setpoint term for cascaded loops. Setpoints can be provided from the CRT keyboard or from other tags or programs. The input to this tag can be from an analog input, a calculated value from another tag, or from a program.

**Lead/lag**

This tag will provide the transfer function necessary for feed-forward control.

**Dead time**

This will delay a signal by a selectable time period from 1-255 seconds to compensate for transportation delay.

**Ratio/Bias**

The output of this tag will be a linear function \(y = mx + b\) of the input.

**Ramp**

The output of this tag is a linear increase/ decrease to a selected target value over a selected time period. This function will support up to three (3) individual ramps to targets with user-defined hold times between the ramping action.
On-Off Control
This function provides a convenient means of receiving an analog value and sending a digital value (ON or OFF) corresponding to the analog (HI/LO) limits. Two (2) contact outputs are allowed per analog input.

Signal Selector
This can take up to six (6) input signals (sensor-based, calculated or numeric constants) and output a single signal as selected by the user:  
- High, Low, Average, or Sum of the signals  
- First good signal  
- First, Second, Third, Fourth, Fifth, or Sixth signal

**SQL (Structured Query Language) Tags.**

This shall consist of two (2) database tag-types, the SQL Data and SQL Trigger, which facilitate the transfer of information bi-directionally between the data acquisition system and a relational (or other) database management system, based on event, time or both. It shall utilize Microsoft’s 32-bit Open Database Connectivity (ODBC) for connection to the external ODBC-compliant database. The capability to transact among multiple databases and multiple rows within a database is required. Also, ”Stored Procedures” shall be able to be invoked in the relational database.

In the event that the database server is “down” (not available) at the moment that a query is executed, the system shall provide a built-in capability to back up the SQL commands and data to user-definable primary or secondary hard drives. The system shall automatically detect when the database server comes back on-line, and, at that time, shall execute all of the transactions that it has backed up.

Support of communication to multiple relational databases simultaneously is also required.

**SQL Data**
This tag defines the data which is to be transferred between the data acquisition system and the relational database, and also, the direction of information flow.

**SQL Trigger**
This tag defines the conditions (triggers) under which the transfer of data will take place.

**Pareto**
This tag-type will accept up to eight (8) inputs and will sort these inputs and their associated descriptors in one of the following ways: ascending, descending, none.

**Statistical Data**
This tag will take its input either manually (from the operator’s keyboard), automatically (from a sensed input elsewhere in the system) or externally (from another program). It must provide the following types of alarms whenever:
- A user-configurable number of consecutive X-bar values is outside the control limits.  
- A user-configurable number of X-bar values is outside the warning limits (67% of the control limits).
• A user-configurable number of range values is outside the control limits.
• A user-configurable number of consecutive ascending or descending X-bar values occurs.
• A user-configurable number of consecutive X-bar values is above or below the control limits.
• Not enough (user-configurable) X-bar values cross the X-axis (non-randomness).

**Statistical Control**

This takes the output of a statistical data tag (the X-bar value) and applies rate and deviation factors to calculate an adjustment to a specific process variable. In this mode, the block performs control based on statistical inputs.

**Batch Support**

The system will provide blocks for state-driven, sequenced, interlocked, batch control operations. The software development company meeting this specification must be able to offer a fully integrated Batch package for future considerations. The functions shall include.

• Device Control. For coordination of opening and closing digital devices based upon user-defined
conditions.
• Program. For running short programs for batch operations or to increase the degree of automation in an application.

Custom features. The ability to add optional function blocks that will co-exist with the standard function blocks and have all the same configuration access as the standards block. The optional blocks are to be built by a toolkit offered by the software development company.

### 2.4 Tag Attributes

Each tag will have an instrumentation tag name of up to 30 characters. The name shall be alphanumeric. All other application programs will use this tag name as their sole reference to the data element assigned.

For tags assigned to actual hardware points, they shall also contain fields for:

• Hardware device name
• Hardware address
• Hardware specific parameters
• Signal conditioning requirements

Simulation tags can also be created that receive their values from the operator’s keyboard, other internal calculations, or other programs.

Tags shall be processed periodically, with the fastest scan rate being fifty (50) milliseconds. Scan rates shall be able to be set independently for each appropriate tag. Longer scan rates of up to once per twenty-four (24) hours shall also be supported. A mechanism for load-leveling, or phasing, the time-based processing of tags is also required.

Alternatively, the user may elect to have tags processed on an exception basis. This choice shall be allowed on a tag-by-tag basis.
Also, any input tag may be configured as a "one-shot". This will cause it to fetch the value from the I/O device just one time when the system is started or the tag is brought on scan.

All tags must have a description field.

Each tag associated with a hardware address or capable of causing an alarm condition shall have a means of displaying a descriptive message on the alarm printer. The descriptor shall be at least 40 characters in length.

The system shall provide built-in signal generators, including sine waves, ramps and pseudo-random random numbers, that can be used for process simulation purposes.

Any output or control block must be able to log a "time stamp" when an operator changes a value.

2.5 Counting of Tags in Sized Systems

For the smaller systems with a limited number of tags (reference Section 1.7), only the following primary blocks are counted towards the I/O limit:

- Analog Input
- Analog Alarm
- Analog Output
- Analog Register
- On-Off Control
- Text Block
- Custom written Tags that address I/O points

Only the blocks listed above are to be counted. There shall be no need to include tags located in other nodes/PCs nor tags that are calculated points.

2.6 Double Precision data and calculations

All the Analog tag types shall support reading and storing double precision data types (as supported by the I/O driver). The Calculation block shall also provide capability to execute double precision calculations.

2.7 Change Management and Deployment

The Tag Database configuration of the system should support Versioning of the SCADA Database. It should support a secure Check In/Check Out concept of Change control, which allows for tracking any changes made to the application and also enables Disaster Recovery mechanisms if required. The Change Management functionality should also support Difference Reporting features for Database Tags with respect to Addition/Deletion of tags and Tag properties.
3 I/O Device Communications

The system must support communication with a variety of external input/output (I/O) devices. The devices that can be interfaced to the system must include:

- Programmable logic controllers (PLCs -- interfaced via serial communications or PLC vendor-supplied interface cards)
- Intelligent single-loop controllers
- Distributed control systems (DCS)
- Bar code readers and scales
- Analog-to-digital converters
- Remote I/O
- OPC Servers

The system must be capable of supporting up to eight (8) different types of device communications drivers and up to eight (8) serial ports simultaneously. For multiple device and/or for multiple serial port configurations, a communications co-processor, such as a Digiboard, can be employed.

3.1 Driver Configuration

The communications driver shall be configurable on-line

When supported by the I/O device (typically programmable controllers), block transfers (the ability to acquire multiple variables in one communications request) must be provided. The block sizes and poll times must be individually adjustable by the user. Supported block transfer times vary depending upon the I/O device, but shall be able to run as fast as the I/O device can transfer data. Alternatively, exception-based processing may be optionally selected. In some cases, the vendor may provide support for the use of unsolicited messages, if supported by the I/O device.

For serial or complex devices, the configuration process shall be interactive and use menus to select (depending on the device):

- Baud rate (if applicable)
- Reply time-out time (with 0.1 second resolution)
- Station address of device (if applicable)
- Block transfer base address and size (if applicable)
- Block interrogation rate

The hardware address format that the user enters at configuration time will correspond with the address format employed by the I/O hardware vendor.

The use of Microsoft Windows Dynamic Data Exchange (DDE) for device communications shall be supported by the vendor. However, the use of DDE shall be limited to communications with low-throughput devices, such as barcode readers and scales. It shall not be permitted for use with PLCs or other similar devices. A DLL or OPC Server will be used for PLC communication.
3.2 Display-only Communications

To facilitate more efficient communications with programmable controllers, the system must be able to provide display-only communications. Communications to read or write tags (Analog Registers and Digital Registers) will only be established when a graphics display containing these tags is open. When the display is closed, communications will cease.

3.3 Error Detection and Recovery

Wherever possible, the device communications program will perform error checking on messages. These error checks will include lost response (time-out) and data error (checksum, LRC, CRC, etc.).

Should communications errors be detected, the software shall automatically indicate that the data (on graphic displays, in historical files, etc.) is no longer valid. The invalid data should be replaced with any used defined characters or have the ability to alter the color, or font to let the operator know the data may be invalid. In the latter case the system shall display the last known value. The system shall automatically attempt to re-establish communications, and, if successful, shall then replace the characters with valid data. These capabilities shall be built-in to the software and shall not require any user programming or other actions to implement.

Failover to a user-configurable back-up port shall be provided as a standard function of the driver. This capability shall, for example, cause the software to automatically attempt to establish communications between the computer and the I/O device via COM2 when it has determined that the connection via COM1 has failed.

3.4 OPC Server Connection

In addition to I/O drivers, the process database must be able to send and receive data with an OPC (OLE for Process Control) server. Any database block should be able to receive or send OPC data by supplying an OPC address. The OPC address should be the following syntax:

ProgID;GroupName;ItemID;AccessPath

**ProgID** is the name of the OPC server, **GroupName** is the name of the OPC group, and **ItemID** is the name of the OPC item. The **AccessPath** is optional, however must be support, and will instruct the server how to access the data.

The system must also be capable of serving OPC information to any OPC complaint database.

The system will support connectivity to Remote OPC Servers through Microsoft DCOM (Distributed COM).

3.5 Diagnostics

The system will provide a diagnostic program capable of running on-line or off-line that can monitor message rates from the communication program. The diagnostic will display the number of new messages, retries, time-outs, and any occurrences of error.

For serial drivers, a built-in datascope shall be provided. This datascope function shall allow the user to observe the messages being sent between the computer and the I/O device.
3.6 I/O & OPC Server Driver Toolkit

A toolkit must be available which will allow the creation of I/O drivers by end users, system integrators, and others. This toolkit must be able to develop DLL I/O drivers as well as OPC Servers through easy to use development wizards. The Toolkit must be able to auto generate documentation.
4 Stand-alone Operation

4.1 Without Networking
In non-networked applications, the software shall be capable of performing all desired functions, data acquisition, graphics, trending, reporting, etc. within a single computer.

4.2 Adding a Network
Should the user elect to network the computers at a later time, all that shall be required is to connect each computer to the local area network, and enable the networking functions within the software. Neither reconfiguration nor duplication of database tags shall be required to make data available to other nodes that require access to it.
5 Networking and Distributed Operation

The system must have a distributed, client/server system architecture based on OPC and Component Object Model Technology (COM). This architecture will employ a local area network (LAN) as the method for communicating among stations. Each computer may be assigned one or more tasks. For example, computer “A” may be used simply for graphic display, computer “B” may be connected to a programmable controller and used for data collection, and computer “C” may be used both for display and data collection.

Configurations shall be available that provide complete functionality, others that provide read/write access to the data (but do not perform I/O communications themselves), and still other information nodes that provide read-only access to all data on the network.

Data shall be available to all computers and individuals on the network that have been provided access. Real-time data shall be available directly across the network from the computer that acquired it from the process hardware. *Configurations that require each computer to contain copies of database tags it needs to access are not acceptable.*

The system shall be configured such that the failure of any one computer will not affect the operation of others on the network. It is recognized that data contained in a failed machine will be unavailable to other machines requesting it. However, the system shall offer the provision for re-starting or re-configuring other stations to take over.

5.1 Configuration and Expansion

The system will provide an on-line installation and configuration program for configuring the various computers on the network. This configuration program will allow assigning unique node names to each computer as well as selecting the functions that the machine will perform.

The system will allow additional computers to be added to the network while on-line, without disrupting the operations of the other machines.

5.2 Local Area Network (LAN) Architecture and Use

The system shall be capable of supporting the following network configurations:

Network Adapters:
- Ethernet

Network Protocols:
- NetBIOS
- TCP/IP

The system must provide session-oriented communications for data transfer. Each computer station must be capable of establishing up to 100 sessions with other stations.

The system must be capable of distributing different files and functions on file servers and network resources other than a stand-alone computer to ease system management. For example, historical data may be saved on one server, graphic (“picture”) files on a different server and so forth.
The system must also be capable of running simultaneously with other LAN users who are not operating the data acquisition system software (but who might be using a LAN manager or file transfer software).

The application program interface used by all system program modules and by user-written applications shall make MMS-like (Manufacturing Message Specification) Variable Access Service function calls.

5.2.1 Network Security

The solution will offer encrypted networking to a level of 256 bits or better. In addition, it will be possible to explicitly specify remote nodes that will have access to a server node as opposed to leaving a system open to communication from any remote node.

5.3 Error Detection, Recovery, and Diagnostics

The system must provide on-line diagnostics that display the current status and operation of the local area network and its nodes. The diagnostic display must include the LAN adapter status for the machine showing the display, as well as the current number of messages, errors and retries.

An additional display will show the current session status (established, pending, off-line) of all stations on the network. A session monitor program that automatically monitors and recovers communications must be supplied with the system.

Should network communications errors be detected the software shall automatically indicate that the data (on graphic displays, in historical files, etc.) is no longer valid and shall replace the invalid data with any user defined characters or the ability to alter the color or font of the text. The system shall automatically attempt to re-establish communications, and, if successful, shall then replace the characters with valid data. This capability shall be built-in to the software and shall not require any user programming or other user-dependent actions to implement. It shall be customizable by the user whether the last known “good” value remains on the screen, with, or without different colors to indicate an error condition, or to have the user-defined characters discussed above used instead.

5.4 Data Handling without duplication of data

Tags shall not be duplicated among stations in order to provide access. Each data value shall be available to all authorized computers and users on the network. However, the data value shall reside only in the computer that is physically connected to the process equipment. Broadcasts of database tags for duplication among machines are not acceptable. Application programs, such as graphic displays, that need data in another station, must request that data over the LAN rather than access a copy in their local database.

Database tags shall be stored in files for each computer physically connected to process equipment. They shall be loaded into main memory during the startup sequence for that machine.

5.5 Remote Access

5.5.1 Via the Network

The system shall have the capability of creating and modifying tags in one node by operating the graphics editor or database builder program in another node. This operation shall be performed on-line, while the
destination node is operating. This new or modified tag data shall immediately be available to all other nodes on the network. The security system will restrict access to the database to authorized users.

5.5.2 Via Modems

Similarly, configuration shall also be available via a computer connected remotely via modems. When a remote computer is connected via a modem, the user shall have the same access as though they were at a computer directly attached to the network. The following functions shall be supported:

- Configuring the database tags throughout the network
- Viewing graphic displays being updated with real-time data
- Viewing historical trend data
- Copying files from/to the network
- Via an Internet Browser -- See section 5.9 of this document.

5.6 OPC and Active X Support

The system must support OLE automation of 3rd party applications, allowing, for example, an Excel spreadsheet to be created and fully controlled from the SCADA/HMI application. The system must support all events, including asynchronous events, of 3rd party ActiveX controls

The system must support OPC data to be directly accessed by the graphics application.

The system must not only allow 3rd party ActiveX controls to be inserted into the application, but for the controls to be able to directly access data from any networked machine without the creation of “dummy” or temporary tags. The system must also allow the ActiveX controls to share data with other objects on the same graphic, or on a different running graphic picture.

5.7 Open Database Connectivity (ODBC) Support

Support of accessing data to and from the process database and historical archive to another database using Structured Query Language (SQL) as a standard language.
6 Microsoft Windows Terminal Server Support

In addition to the client/server networking described in section 5, the system shall support the user of Microsoft Terminal Server. The user shall be able to have multiple configurations (view-only, view, and configure) without the need to purchase or install separate configurations. A configuration management tool will be provided to map remote users with appropriate SCADA/HMI applications to launch.

6.1 Terminal Services Server

The Terminal Services Server will be available in two forms, running along side the SCADA/HMI or running as a separate box from the SCADA/HMI. If along side the SCADA/HMI, the Server will support up to 3 remote clients. This will be limited to ensure proper operation of the SCADA/HMI.

If operating separate from the SCADA/HMI, the server shall be capable of up to 25 users.

6.2 CITRIX Support

The Microsoft Terminal Services Server will support operation in an enhanced CITRIX environment. This environment will enable the use of better session diagnostics and operation across multiple servers for load balancing.
7 Electronic Signatures and Electronic Records

The system should be able to handle the addition of electronic signatures to any point. Electronic signatures should include one or two signatures per any runtime change including alarm acknowledgement. User accounts should have the ability to be configured within Windows security and a period of continuous use should be configurable were one token (password) would need to be entered. There should be a provision to force 2 tokens even during this period of continuous use.

Users should have the ability to enter comments at time of signing and the system should allow for comments to be selected from a predefined list or entered in a free from text box.

No changes should occur until the authentication of the user is verified and an electronic record is created about the transactions. This record must contain the time, date, full name of the perform by, full name of the verified by, the original value and new value, the description of the action and any comments entered at time of signing.

7.1 Biometric Devices

The SCADA/HMI shall offer as an option, the ability to replace Password authentication with a Biometric Device (such as a thumb print reader). This will be a standard feature without the need for custom coding.
8 Graphics Capabilities

The graphics package 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 the field devices shall be capable of being displayed in a variety of user-configurable formats.

Graphic displays shall be standard Microsoft Windows files and shall be able to be stored on the system disk, a floppy diskette, virtual (RAM) disk or file server, based on user-entered selections. There shall be no limit (other than physical disk size) to the number of displays that can be developed and accessed on-line.

The development and runtime graphics packages must both be multi-document architecture applications.

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. Also, support for using Multi Display Screens for viewing Graphics should also be available without any programming or complex configurations.

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 tools menus and dialog boxes 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. All properties, events, and methods of the object — including 3rd party controls — must be exposed to the system and available for the developer.

The system must support changing individual graphic files (“pictures”) at any time with no additional configuration, or requiring the system to be stopped, or restarted.

8.1 Graphic Creation

The system shall provide an interactive object-oriented editor or workspace that allows creation of graphic displays using a pointing device (for example, a mouse).

The environment shall support the Microsoft Fluent UI, to arrange the options available for Graphics creation in a Ribbon toolbar format.

A facility shall be provided for the user to configure a Quick Access Toolbar, for faster access to frequently used Graphics development functions.

A facility shall be provided that quickly toggles, via a mouse click or hot-key, between the graphic building and graphic runtime modes to speed display animation verification during the development process.

The software must be designed with the ability to make changes to the graphics while the system is running. Shutting down the system shall not be required to make changes. If the graphics files are stored on a network file server then all machines will see the effect of any changes the next time the graphic is displayed.

8.1.1 Browser

Once an object is created, the object needs to automatically be placed on a tree similar to Microsoft’s Explorer program.

8.1.2 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. All properties are available in both the graphic workspace and the scripting engine.

8.1.3 Object Duplication

Object properties must be passed when an object is copied. Copying should be able to occur from the tree browser or workspace. All properties must be passed on to the duplicated object and the name properties must automatically get changed. Example: When an object with the following properties Name: OVAL; Foreground Color: RED 024 gets duplicated the new objects properties are Name: OVAL1; Foreground Color: RED 024.

If an object is copied all of its scripts are copied with it. If the script refers to a property of the object (or any item grouped within the object), the new copy will correctly refer to its own copy of the object.

8.1.4 Object Drill Down

The system will allow the developer to “drill down” into any sub or grouped objects within any other object and change any property, script, or animation without the need to ungroup or otherwise modify the main object.

8.1.5 Tile & Cascade

Graphic screens that are opened in configuration mode must support tiling and cascading. Tiling must have horizontal and vertical support and no overlapping when the graphic screens are viewed in this manor. The only limit on the number of graphic screens opened at one time is by the amount or Ram in the PC. Cascading is defined as a method to staggered pictures so they can be selected from their title bar.

8.1.6 Graphic Sizing

Size will be based on logical units; not pixels and any logical unit may be used. Graphic screen design at one resolution must be able to run at a different resolution. A full screen option as well as the ability to add sizing borders to any graphic screen must be supported. Also graphic screens must have an option to enable the screen to always be on top and a title bar enabled / disabled option.

8.1.7 Active X Support

The graphic screens must be an active X document and have the ability to have third party ActiveX OCX, controls dropped in. The system must be capable of containing any control that is placed onto a graphic screen. If a third party control crashes or misbehaves the system must be able to shut down the control while the graphic screen, system and PC remains running. Running third party controls out of the process is not a suitable requirement for the protection.

8.1.8 Relational Database Support

The graphics application will provide the ability to insert tables, queries, views and procedures from any ODBC or OLE-DB database (Access, SQL Server, Oracle, dBase, etc.) into a picture. The system will include listboxes, comboboxes, grids, and data controls to interact with and query the database. All of these controls should have the ability to be linked together so that, for example, the user can select a batch number from a listbox and the grid control automatically updates, with no scripting required of any kind.
The data control should have a “wizard” to automatically create SQL queries without the user knowing SQL. However, the user should have the ability to view and modify the SQL created by the wizard.

The data control must support asynchronous data transfer to/from the database so large queries can be executed without adversely impacting performance.

The data control must support reading and writing to a database using the built-in procedures of the database. In addition, if connecting to a procedure the control should indicate the number and type of variables that the procedure requires.

8.2 Graphics Pre-Loading

The system shall support the pre-loading (“caching”) of pictures to improve graphic performance. The use shall be able to set a list of pictures to be pre-loaded when the system starts, or select a maximum number of pictures to be automatically cached in memory as the system runs.

8.3 Color Support

The graphics package shall provide support for an unlimited choice of colors with at least 256 colors supported at any one time. The user shall have the ability to create, save, and restore custom color palettes. Each color must have an associated name so users can selected from the color from the name of the color. The graphics package will also provide facility to define gradients in the colors used on the graphics. Every color available in the palette should be available for gradient filling.

8.3.1 Color Palettes

The system must come with a standard, rainbow color palette as well as several standard, shades of color palettes. Each shade palette needs to have 256 shades. The following standard shade palettes must be supplied.

- Cyan
- Gray
- Green
- Magenta
- Red
- Yellow
- Blue

8.3.2 Color Changes

Color changes must be selectable from editing the individual foreground, background, or edge color property for each object. In addition color changes must be selected from a “Modeless” color box. The “modeless” color box must float on the workspace and allow the user to change color on as many objects as they wish and choose which property of an object or objects they wish to change.

8.3.3 Global Colors
The system must allow for a global or universal color table selections. This table is based on exact match, or range compression or a value. The colors in the tables will appear on any graphic screen when the value for the data source of the object matches the table. Changes to color tables must be independent of the graphic screens and not require the user to compile or pass the graphic screen through the graphic configuration program or mode for changes to take place. Changes to global color tables must be supported in runtime mode.

8.4 Graphic Toolbox

The system must provide configurable toolboxes that the graphics developer can customize as to what tools it contains and their position in the toolboxes.

The Toolboxes must be a Window where its shape, size and location can easily be changed with the mouse.

Toolboxes shall contain a method, like the ToolTips within Microsoft Word, to describe the function of each tool when the mouse cursor is positioned on a particular tool.

Once configured, the state of the toolboxes shall be automatically saved when the drawing session is completed. It shall be returned to that same condition when the next drawing session is started.

Users must have the ability to define their own buttons.

8.4.1 Graphic Objects

At a minimum, the following object drawing tools must be supported:

- Rectangle/Square
- Rounded Rectangle/Rounded Square
- Oval/Circle
- Straight Line
- Polylines (two or more connected line segments)
- Polygons
- Arcs (curved line segments)
- Chords (a curved line connecting a line segment)
- Pie Shapes (wedges of a circle)
- Text
- Pipes
- Connection Points
- Vertical and Horizontal connection lines

8.4.2 Graphic Development Operations

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

- Select/Select All
- Deselect/Deselect All
8.4.3 Graphic Animation

Each display must have the ability to dynamically update elements in the picture. Defining the method for dynamic update shall be determined by a point and click operation.
A pre-defined list of dynamic link elements that shall include the following:

- **Data Link**  Displays alphanumeric values (numeric values may be displayed in whole number, decimal, or scientific notation)
- **Historian Data Link**  Displays last stored value from Proficy Historian with a time stamp.
- **Time Link**  Displays current time
- **Date Link**  Displays current date
- **System Information Link**  Displays diagnostic information
- **Alarm Summary Link**  Displays current alarm information
- **Pushbutton Link**  Executes a Command Language script
- **Multi-pen Chart Link**  See below
- **OLE objects**  Display a third party OLE object
- **Enhanced Charts**  See below

### 8.4.4 Multiple-pen chart link

- Any number of pens
- Displays run time and historical data on same chart
- Allow the user to insert data (i.e., lab data) into a chart
- Configurable time span
- Configurable trend direction (left to right and right to left)
- Configurable Zoom
- Scrolling Grid
- Ability to invert Hi and Low limits
- Minimum of 5 line styles for pens
- Minimum of 3 pre built line makers and a customizable lime marker.

### 8.4.5 Enhanced Charts

- Displays Real time, Historical, X Bar, R Bar, S Bar and Histogram charts
- Allow the user to select real any number of pens from the database or Proficy Historian for the Real time and Historical charts
- Allows user to select the SPC and Histogram database blocks for the X bar, R Bar, S Bar and Histogram charts
- Allows users to toggle between Linear and Logarithmic scale in development and runtime modes
- Configurable time span
- Configurable Zoom
- Scrolling Grid
- Ability to invert Hi and Low limits
• Minimum of 10 line styles for pens
• Minimum of 5 plotting styles (Line, Bar, Points)

8.4.6 Dynamic Properties for Objects

Dynamic properties that may be assigned to an object or group of objects must include the following:

Color changes:
• Foreground Color
• Edge Color
• Background Color
• Gradient Fill Color

Fill Percentage:
• Horizontal (Left/Right)
• Vertical (Up/Down)

Position/animation:
• Horizontal (X)
• Vertical (Y)
• Rotate
• Scale

Script Language:
• Commands on down (mouse button or key)
• Commands on up (mouse button or key)
• Commands on mouse click
• Commands on mouse double click
• Commands on mouse move
• Commands on edit

Fill Style:
• Solid
• Hollow
• Horizontal
• Vertical
• Downward Diagonal
• Upper Diagonal
• Cross Hatch
• Diagonal Cross Hatch
• True Gradient Fill

Edge Style:
• Solid
• Hollow
• Dash
• Dot
• Dash Dot
• Dash Dot Dot
• Null
• Inside Frame

Objects may be assigned more than one (1) dynamic property. In addition, objects within groups may have individual dynamic properties in addition to those dynamic properties assigned to the overall group.

For properties other than “Commands”, configuration shall be by “point and click” operations; scripting or programming shall not be required.

When building object dynamics properties must support configuration from a dialog box, pop-up menu and user customizable dialog boxes or forms.

Positioning property changes must support a method to get screen coordinates and automatically fill in the required coordinates for positioning.

The user customizable dialog boxes or forms must be customizable through VBA. The system must supply the following pre built forms:

• Fill
• Rotate
• Position
• Scale
• Visibility
• Edge Color
• Foreground Color
• Background Color
• Data Entry
• Open Picture
• Close Picture
• Replace Picture
• Open Digital Tag
8.4.7 Graphic Refresh Rate

The refresh rate shall be user-definable on a per object basis with the fastest rate being fifty (50) milliseconds, although it is recognized that achieving this performance is dependent upon the overall system configuration.

8.4.8 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
- Local and networked relational databases using SQL/ODBC
- A 3rd party ActiveX control (without creating temporary or dummy tags)

The system shall provide a wildcard supported filter for assigning a data source.

The system must provide an expression builder that is accessible from the graphic workspace. The builder will allow an expression to be assigned to the data source. Supported functions of the builder are:

- Addition +
- Subtraction -
- Multiplication *
- Division /
- Left & Right Parenthesis ()
- Equal To =
- Not Equal To <>
- Greater Than >
- Less Than <
- Greater Than or Equal To >=
- Less Than or Equal To...<=

8.5 Reusing Graphic Objects

A method shall be provided for allowing graphics objects or groups of objects to be re-used easily. It shall allow the developer to insert native language prompts that request appropriate tag or other animation
information whenever the object or grouped object is reused in another graphic display. These objects, either single or grouped, shall be intelligent, Windows wizard-like objects, so that it is possible, for example, to have a single prompt request and substitute:

- A single tag name into multiple dynamic properties within the object
- Multiple attributes (current value, high alarm limit, tag name, etc.) from a single tag into multiple dynamic properties within the object
- Text into the object
- Parameters within command language sequences

A library of these objects shall be included with the standard product. At a minimum, this library shall include:

- Pipes
- Valves
- Pumps
- Motors
- Tanks
- Bitmaps

A method shall be provided to update all these wizards used in the application if the Master Wizard from the library is Updated.

The system will allow conversion of previously created graphic objects to the newer dynamically linked objects through a converter.

The system must allow for bitmaps created by other packages to be imported into the graphics, Bitmaps must support a transparent mode and Metafiles must import as objects not just bitmaps. At a minimum the system must support .bmp, .msp, .jpg, wmf, pcx, ico, cur, psd, epr, and wpg.

8.6 Tag Aliasing and Tag Group support

The system must support a means of creating single picture that can use any one of a number of different data sources. For example, the user can create a single PID faceplate “pop up” picture that when opened can display different data from different tags, based on which item PID the user selected. The system shall allow the user to create a picture by inserting tag substitution strings in any link in a picture. A separate tag group file will then be used to cross-reference the substitution strings with the data source. When the graphic is opened, the user can either specify the tag group file to use, or the operator can be prompted.

The tag substitution strings shall be insert able into any data link, variable, or animation in a picture and can be combined to create concatanable strings. (e.g., String1 + NODE + String2).

The tag substitution strings shall also be fully supported in the scripting environment.
A tag group editor shall be provided to assist in the creating of tag group files. The editor shall be able to browse any picture and automatically create a list of all unique tag substitution strings in a picture.

8.7 **Support Microsoft Excel and Word Documents**

Microsoft Excel and Word documents must be able to live within a graphic screen. The documents will run within the graphic, not as an external call. The Microsoft Excel or Word toolbars will get inserted as part of the graphic toolbars for editing.

8.8 **Documenting Graphic Displays**

Printing of graphic displays in color and black and white shall be supported via the standard Microsoft Windows Print Manager in both the graphics development and runtime environments.

8.9 **Operator Entry Methods**

There shall be provided a flexible, Microsoft Windows standard methodology of operator interaction with the system.

8.9.1 **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. As appropriate, simultaneous support for multiple of these devices shall be provided.

8.9.2 **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
- Ramp Value Entry
- Alphanumeric

Each data entry type must be configurable to require confirmation if so desired. The use of third party active X controls may be used. Refer to the Active X section (6.1.6) of this document.
The system shall print a descriptive message with a time stamp and user ID on the alarm printer or to an alarm file (if so configured) whenever any of the following events occur:

- Alarm acknowledgment
- Data entry into a tag
- Reloading a database file
- Saving a database file
- Restarting the system

8.10 Command/Scripting Language

The scripting language used by the system must be Microsoft's Visual Basic for Applications (VBA) (at least version 6.4 or above) 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. Scripts must be capable of running in either the configure environment (“draw” or “edit”) or the run environment (“view”). The scripting language must use Microsoft’s IntelliSense feature, exposing all properties, methods and events of graphic objects. This includes 3rd party ActiveX controls. Editing will be with the Visual Basic Editor (VBE), which is part of VBA.

8.10.1 Scripting language requirements:

- Animation of objects in pictures.
- Automatic generation of pictures or objects.
- Read from, write to, and create database blocks.
- Automatically run other applications.
- Incorporate custom security features.
- Create custom prompts and messages for operators.
- Access ODBC or ADO data sources.
- Incorporate and communicate with third party and custom ActiveX controls.
- Trap bad or misbehaving ActiveX controls to prevent crashes.
- Write custom wizards for frequently performed tasks and offer as Toolbars
- Allow use of global scripts and global variables.
- Scripts become part of the graphic screen.
- The VBE must allow import and export capability.
- The must be a link from the graphic editor to the VBE.
- VBA or the VBE is launched from within the system, without any commands.
- All Properties, methods, and event of Graphic object created within the graphic editor or Third party ActiveX controls used in the graphic screen must be exposed to VBA.

8.10.2 Object Hierarchy Overview
8.10.3 Renaming, Duplicating and Moving Objects

If an object is renamed in a graphic screen the name must be changed automatically in the VBA script of the object.

When an object is duplicated in a graphic picture the script, form and module must also be duplicated and a new name reference must automatically happen to for the new object, form, module, and script.

When an object is moved between pictures the form, module and script must move with it. If the object is placed into a graphic screen that already is using the same name, the renaming mentioned above must take place for the object, form, script, and module.

8.10.4 Global Scripts and Variables

The ability to have scripts and variables available across all graphic screens. These global scripts and variables must get loaded when the system is started.

8.11 Multimedia Capability

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

8.12 Sample System

The vendor shall provide a “sample system” that assists a user in using the software for the first time.

- The sample system shall include several graphics mimicking two or more “real world” application (e.g., a water treatment facility and a discrete manufacturing facility) that shall demonstrate most of the features discussed in section 8. At a minimum, this shall include:

  Using tanks, pumps, valves, data links and pipes in graphics
  Using tag groups (Section 8.6)
  Using global color tables and string tables (Section 8.3.3)
  The alarming object (Section 9.10)
  Using the trending object to display real-time, historical and “lab data” (Section 8.4.4 & 10.3)
  Relational database integration (Section 8.1.8)
  Using the Microsoft web browser ActiveX control (Section 8.1.7) to display web pages within a graphic.
  Using and displaying alarm counters (Section 9.5)
  Picture layers (Section 8.4.2)
• Since the sample system is meant to be instructive, it shall use the graphic tools described in section 8.4 to create the displayed graphics.

• The system shall not use any scripting to “simulate” process data, but instead use the real-time database and the provided simulation signals.

• The sample system shall extensively use the context-sensitive help described in section 1.6.2 to assist the user in using the system, in addition to instructing the user in how particular features in the sample system were created.

8.13 Web SCADA Clients
The system should support rendering graphics already developed on the regular operator displays to the web through Internet Explorer and/or Mozilla Firefox browser. The system should not require any additional development or conversion to render the graphics on the web, nor, should it pose any limitation on functionality through the web client unless configured through security.

8.14 Change Management and Deployment
The Graphics configuration of the system should support Versioning of the graphics. It should support a secure Check In/Check Out concept of Change control which allows for tracking any changes made to the application and also enables Disaster Recovery mechanisms if required. The Change Management functionality should also support deploying Graphic changes to all the SCADA Clients participating in the system. The Change Management functionality should also support Difference Reporting features for Graphic Object Animations, Scripts, Global Variables and Graphics layouts in general.
9 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. Alarm limits can be entered by the user at configuration time or from the operator's display during run-time. Alarm limits are expressed in engineering units.

9.1 Alarm Types

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

- High High
- High
- Low
- Low Low
- Time rate-of-change
- Bad input from I/O
- Alarm Disable
- Off Scan
- Deadband

Digital input variables shall have the following alarm types:

- None
- Change of state
- Open
- Close

3-term (PID) control functions shall have the following alarm types (in addition to the alarm types associated with the analog input or alarm block providing the measurement):

- Deviation from Setpoint

Statistical data tags shall support “out of control” alarming on the following limits:

- $X_{\bar{b}}$, range and standard deviation control limits
- $\pm 2 \Sigma X_{\bar{b}}$
- $\pm 3 \Sigma X_{\bar{b}}$
- Range
- Trend of runs
- Length of runs
- Critical runs
- Standard Deviation (SBAR)

9.2 Alarm Priorities and Filters

The system shall support at least 7 alarm priorities for each alarm type: INFO, LOLO, LOW, MEDIUM, HIGH, HIHI, and CRITICAL. A filtering mechanism shall be provided so that the operator can adjust the system alarm priority. The priority should indicate the importance of a block’s alarms. The priority of the most critical blocks would be CRITICAL. This would distinguish the highest priority alarms from less-critical ones.

Special alarm messages (such as I/O failure) shall be non-maskable and shall always print.

9.3 Messages

Messaging enabling and disabling must be controlled at the block level. The system must be able to send messages based on the following events.

- An operator
- A Process Database Event Occurs
- A system-Level event occurs

In addition to alarms the following type of blocks must be able to generate messages that report to any transactions to and from the hardware:

- Digital Input
- Digital Output
- Digital Register
- Analog Output
- Analog Register
- Text

9.3.1 Application Messages

Application messages are considered messages that describe database-related activity or operator entry. Here is a list of the type of messages the system must produce:

- Operator – Change a process value, Load a process database, Log into the system
- Recipe – reports any recipe upload, download or save condition.
- Script - Send information from a VBA script to all enabled alarm destinations.
- Database Block – Send a message from the database to all alarm destinations.

All application messages are to be logged to alarm areas.
9.3.2 System Messages

System messages are messages that will provide information about completed tasks and errors. System Messages will occur on:

- A database finishes loading.
- The state of a network session changes.
- An I/O driver detects an error.
- Start ups.
- Database block errors.
- Run-time or system errors.

System errors will be viewed by a pop up message viewer. The viewer should allow users the following configuration:

- Show all entries or just new ones.
- Maximize on next new entry.
- Clear, and exit.
- Disable the viewer from popping up.

9.4 Alarm Areas

In order to logically divide a process into smaller units, the system shall allow for unlimited, named individual alarm areas to be defined. These alarm areas must be definable on an individual tag level. All alarm areas must be accessible by each tag and the system must support multiple alarm areas per tag.

Alarm areas are used to determine which destinations receive each alarm. The method of alarm distribution over a network must be session-based in order to guarantee alarm distribution and reception. Broadcasting of alarms on the network shall not be permitted.

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

9.5 Alarm Counters

The system shall provide “counters” to display the number, type, and priority of alarms in any alarm area. The system shall also provide Alarm Occurrence Counters to determine transitions in alarms between Acknowledgements. The use shall be able to use these counters in mathematical and Boolean expressions to display, for example, the number of high priority unacknowledged alarms in a user-specified number of alarm areas. The alarm counters must also be accessible as directly addressable I/O points in the tag database.

The alarm counters in any alarm area shall include:

- Total number of acknowledged alarms
- Total number of unacknowledged alarms
- Running total of alarms
- Number of acknowledged High, Medium, & Low priority alarms
• Number of unacknowledged High, Medium, & Low priority alarms
• Number of disabled High, Medium, & Low priority alarms

The system shall also be able to display a summary of all of the above alarm counters for all areas on a given node.

9.6 Alarm Destinations
The system shall provide a means for placing an alarm message in one or more of the following locations:

• Alarm summary display
• Alarm printer
• Alarm message file on disk
• Alarm history window (first-in, first-out scrolling window on the display)

The system shall allow either COM1, COM2, LPT1 or LPT2 and USB connected printers to be the alarm printer. The use of multiple alarm printers shall be supported to allow routing of alarms from different alarm areas to different printers.

Alarm messages shall be independently user-configurable as to what information is provided and its sequence within the message. The following shall be available choices:

• Time of the alarm
• Name of the tag causing the alarm
• Alarm condition code
• Engineering units value when the alarm occurred
• Descriptor text assigned to the tag
• Engineering units of the tag
• Directly to a relational database

Also, the user shall be able to specify the length of the alarm queue for each destination.

9.7 Time Stamping
A time stamp must be included with every alarm or message. This time stamp will indicate the time and date that the alarm or message was generated. Time stamping must be supported from the local computer time, OPC server time, or process hardware's clock.

9.8 Auto Alarm Manager
The system must support a method for alarms to be acknowledged from a central location, via modem. The connections from the remote node to the central node must only be live during an alarm condition. This connection must use Microsoft's Remote Access Service (RAS). The system must support a primary and a secondary selection for alarms reporting.
9.9 Alarm Notification and Acknowledgment

When a new alarm condition is detected, an alarm message will be generated.

If the alarm condition code text for the block is on the current display, then the text will flash until the alarm is acknowledged. Alarm acknowledgment will be performed from the operator's keyboard or with the mouse and shall require no more than one keystroke or mouse click.

The system must be capable of “freezing” the highest alarm status value on the display until acknowledgment is made. Once acknowledgment is made, the system will display the current alarm status text.

The software shall provide built-in capabilities to support the following:

- Remote acknowledgment. This shall allow, for example, a button to be depressed by the operator which closes a digital tag and acknowledges one or more alarm conditions, as configured by the user.
- Alarm suspension. This shall allow the user to specify digital tags, that when closed, cause alarms not to be generated for one or more alarm conditions. This is useful, for example, during the start-up phase of a project to avoid nuisance alarms.
- Re-alarm time. This shall allow the system to re-generate an alarm after a user-configurable amount of time, should the alarm condition still exist.
- Delay time. This shall allow the user to specify a period of time for which an alarm condition must remain before an alarm is generated. This is useful, for example, if certain actions may cause a temporary, but acceptable, fluctuation beyond alarm limits and the generation of alarms is not desired.
- Close contact on alarm. This shall allow the user to specify digital tags that become closed when certain alarm conditions occur. These contacts can then be used to take actions, such as sounding a horn or initiating a sequence of instructions. Also, the user can specify the conditions under which these digital tags are re-opened, including the following:
  - When the alarm is acknowledged
  - When the alarm is cleared
  - When the alarm is acknowledged and cleared
  - Never (it must be re-opened by a different function)

For analog values, re-alarm time, delay time, and close contact on alarm capabilities shall be supported, not just on a tag by tag basis, but for the following individual alarm types within a tag:

- High High
- High
- Low
- Low Low
- Rate of change
- Deviation from target value
- All other
When an alarm is acknowledged from any node on the network, the acknowledgment shall be made directly at the node from which the alarm was generated, and a message indicating that it has been acknowledged shall then be distributed to all alarm destinations.

Messages shall be able to be designated as "events-only". These will be distributed to alarm destinations, but shall not require acknowledgment.

9.10 Alarm Summary Display

The system must offer an alarm summary display as a pre-defined, customizable, ActiveX control within the graphics package. This alarm summary display must show a list of the pending alarms in the system. As new alarms are detected, entries are made to the display list. As the alarm conditions clear, the entries are removed from the list.

In addition to being able to configure the placement of the information (tag name, current value, descriptor, time of alarm, and alarm status), the user shall be able to specify the color codes to be used to indicate the various alarm conditions either by the Alarm status or the Alarm priority. The user shall also have the capability to reverse the row color upon acknowledgement of the alarm.

Alarms can be acknowledged from the alarm summary display either individually (by clicking on an alarm acknowledgment field) or for all alarms in the queue.

The alarm summary display must provide sorting and filtering capabilities. The user shall be able to filter on acknowledge status, node name, alarm area(s), alarm status and alarm priority. The user must be able to sort on time, tag, alarm area, alarm priority and alarm status. The user must be able to display field or fields about the alarm block in a column format and do complex filtering.

The alarm summary object must also be to fire a VBA "event" to indicate the presence of new alarms or that the list has changed. It should also have the capability to fire a event on alarm escalation of any alarm in the list.
10 Archiving and Reporting

The system must provide a facility for automatically collecting, storing and recalling data. Recalled data will be made available to a trend display program, a report generation program and to user-written programs.

10.1 Data File Handling

Data will be stored in Windows-compatible files in compressed format. Compression will be performed through a user-supplied dead band. Entries containing time, name, value and status will be made in the file whenever the real-time value exceeds the previously-stored value by the dead band limit. A dead band value of zero will cause an entry in the file each time the real-time value is examined.

Files shall be organized according to time and will contain values for multiple, named variables. The file can be placed on the hard disk or a floppy disk, and can be placed on a file server if LAN server software in installed.

A mechanism for on-line maintenance and automatic purging of files must also be provided.

The system must support third part applications for ODBC queries.

10.2 Archive Configuration

The data to be collected by the archiving program will be identified through an interactive, menu-based configurator. The user will enter the tag name, collection rate, and data compression dead band value.

Collection Rates are:
- 1 Second
- 2 Seconds
- 10 Seconds
- 20 Seconds
- 30 Seconds
- 1 Minute
- 2 Minutes
- 10 Minutes
- 20 Minutes
- 30 Minutes

The collection task can be run at any one (1) or more computers on the network. The task shall have the ability to access data from the memory-resident tag list in its own computer and/or any computer(s) on the network.

The task will allow the collection of groups of tags to be turned on and off automatically based on an alarm on discrete value in any networked database with no scripting required.

10.3 Displaying Archived Data

The operator shall be able to recall archived data from the disk to be displayed in graphic format along with Real-time data, OPC data, data from any object in a picture including 3rd party ActiveX controls, alarm counters and any data programmatically added to the chart.
The display of archived data shall be user-configurable. It shall be possible to configure objects in graphic displays that, when selected, fetch pre-defined historical trend data from disk and display it to the operator. The system must allow for users to edit a pen's attributes during runtime.

The display shall support unlimited variables to be displayed on the same time/value axis simultaneously. For each entry in the display list, the operator will be able to assign a given tag name and marker to a particular line color selected from palettes of unlimited colors. The operator may also enter display engineering units ranges to cause scaling of the display. Support shall be provided for multiple, different y-axis engineering units to be displayed, as appropriate.

The trend object will allow for bi-directional trending and scrolling. A user can select right to left or left to right.

A movable, vertical line will act as a time cursor on the display. This cursor can be moved by dragging it with the mouse. The date, time, and values of the trends corresponding to that time will be displayed in the bottom portion of the screen.

The grid of the trend object shall be scrollable.

The trend may be shifted forward or backward in time ("panning") by clicking on left/right buttons. New data will be fetched from the historical file as appropriate. Two sets of buttons shall be provided that cause shifting by different amounts of time. The amount of time shifting caused by these buttons shall be user-configurable.

The ability to display historical (pre collected) data with current (real time) data on the same chart must be supported.

A transparent option for the trend must be selectable.

The user shall be able to "zoom" in on any section of the trend display by "cutting" that section with a mouse. The software will automatically re-scale both the y-axis and time axis and will fetch the appropriate data for the time period selected.

The trend object must have a refresh rate selectable in .1 second increments from a minimum of .10 seconds to a maximum of 1800 seconds.

### 10.4 Reporting Options

The system shall make all of its historical and “real time” data available via ODBC drivers allowing the user to use the reporting tool of his or her choice.

### 10.5 Display Output

The trend graphic display must be printable to a black and white or color printer via the standard Microsoft Windows Print Manager.
11 Event Scheduling

The system shall support an event scheduler with event-based events, and time-based events. Any valid VBA code can be included in the scheduler. The scheduler must be able to run as a background NT "service" to allow scripts to run irrespective if an operator is logged in, or if the graphic application running.

11.1 Event-Based Events

Event based items are to be used by any data source available to the system to trigger events. The system must be able to trigger an event on the following conditions:

- When the data source resolves to a non-zero value (On True).
- When the data source resolves to a zero value (On False).
- At the specified interval as long as the data source evaluates to a non-zero value (While True).
- At the specified interval as long as the data source evaluates to zero (While False).
- When the data source changes (On Change).
- On any Boolean condition based on any number of data on any networked machine. E.g., if (Machine1.MOTOR_VOLTS <= 110) AND (Machine1.MOTOR_AMPS > 50) = True, False, While True, etc.

11.2 Time-Based Events

Time-based events are events that occur at a scheduled time and are not tied to a data source. The system must allow for scheduling of the following time-based events:

- One shot. Events that run once at the specified data and time.
- Continuous. Events that run continuously at the specified date and time.
- Daily. Events that run on the scheduled day of the week at a particular time.
- Monthly. Events that run on the scheduled day or the month at a particular time. “End of month” will be supported with no scripting to automatically take into account months with different lengths.

11.3 Events with VBA scripts

The system shall support a script-authoring wizard for events. The wizard will generate a VBA script tied to the appropriate event. Once this script is created, editing must be allowed from the wizard or the supplied Visual Basic Editor (VBE).
12 Security Management

The software shall provide a user-based security system. If enabled, the security system must allow for the creation of users with certain rights and/or privileges. These rights must include the ability to run any combination or all of the applications in the data acquisition system. The ability to allow or disallow users access to change values, such as set points and machine-setups, on an individual tag basis shall be supported.

The ability to prevent non-authorized users from modifying the value or set point of a tag must be part of the database design and must not require any scripting.

Groups of users, such as Operators or Supervisors, can be created and granted rights. All users assigned to a group obtain the rights of the group, although they are still tracked by the system by their individual ID. Individual members of a group may also be assigned additional rights.

The security system will support either centralized or distributed security file management.

The system shall support using Microsoft Windows security in lieu of, or in addition to the package's own security system. If using Windows security, the system shall authenticate a user on any security domain without requiring the system to be re-started.

When user-based security is enabled, an audit trail will be generated in the system which will tag every operator action with a user identification (ID).

Systems that use a numeric level-based security methodology shall not be acceptable.

12.1 Security Areas

The system must support up to 254 separate security areas. Security areas shall be assignable on a per tag basis. Each tag can be assigned all of the available security areas, none of the available security areas, or up to three (3) individual security areas. Only users with clearance for those security areas shall have the ability to change parameters.

Security area names may be up to twenty (20) characters in length.

12.2 Security Manager

The following functions must be supported within the security manager application:

- Enable/Disable user-based security
- Define users, passwords and login names
- Define groups to which users may belong
- Define security path(s)
- Define user and/or group rights/privileges
- Define security area names
- Define system auto-start user
12.3 Securing the Windows GUI Environment

The ability to "lock" an operator or other user into the runtime graphics environment shall be provided. Specifically, disabling any combination of the following shall be supported, as configured by the user:

- Starting other applications.
- Switching to other applications that may be running.
- Exiting from the system.
- Restarting the computer using <Ctrl><Alt><Delete>.
- Opening unauthorized graphic screens.
- Closing the current graphic screens.
- Using the system menu.
- Switching to the configuration environment.
- Accessing the system tree.

12.4 Limiting Login Time

The system shall allow for a login timeout setting for each user account. This setting will log out an operator when the time interval expires.

12.5 Automatic and Manual Logging In and Out

The system shall support manual login and logout as well as automatic login. In addition, security information must be customizable through VBA scripting.
# 13 Recipe Management

A recipe management facility shall be provided that can be used for such functions as downloading set points or machine setup parameters to the process hardware.

## 13.1 Recipe Configuration

The recipe shall be configured via a spreadsheet-like application. Alternatively, recipe items may be imported from other applications, such as Microsoft Excel.

The user shall be able to develop recipes specific to a batch, line, or process or generic recipes that can be applicable to multiple batches, lines or processes.

## 13.2 Recipe Management Functionality

Both master (generic templates) and control (specific to a particular trial or batch) recipes shall be supported. There shall be the ability to use formulas, variables, and keywords within recipes in order to develop complex expressions.

Support for batch sizing shall be provided. This shall permit the master recipe to be developed for one quantity to be produced and to have the recipe automatically scale the recipe item each time that the recipe is run.

## 13.3 Recipe Security and Tracking

The recipe management system shall be integrated into the user-based security system with rights being assignable for such functions as upload, download, and accessing the recipe development facility.

Operators may be provided authority to override limits on some or all recipe items in order to compensate for such items as environmental conditions, potency of ingredients, and material thickness measurements.

Complete journal support of all actions shall be provided for validation and audit trail purposes.

The system must support a means of a generic recipe name for sharing recipes between different nodes with different names.
14 Redundancy

The system must support SCADA server failover and LAN redundancy. Both redundancy options are independent of each other. The system may be configured for SCADA server failover, LAN redundancy or both.

14.1 SCADA server Failover

The system shall have a failover option that allows a client to switch to a backup server in case the primary server is lost. The primary and the backup SCADA server will support the following tasks:

a) Database Synchronization – The system shall have the capability to run a SCADA synchronization task that will replicate the database from the Primary to the Backup SCADA server. Every aspect of the database tag, from alarm limits to alarm area configuration and from adding and deleting tags to changing egu limits in the tag, all will be synchronized and replicated from primary to secondary.

b) Real time Data synchronization – Any data which has been configured for animation or program execution purposes and which does not get updated from the process data, will also be synchronized from the Primary to Backup.

c) Alarm Synchronization – The alarm queues and alarm acknowledgment information shall be synchronized between the primary and backup SCADA. The backup SCADA shall, in a normal operation, only read alarms from the primary and will only have the capability to report its health status. In a failover situation, where the backup becomes the primary server, that server will start generating alarms.

d) Alarm ODBC failover – The Alarm ODBC task which stores alarms to a relational database shall also have the failover capability from Primary to Backup.

The SCADA synchronization task shall have the capability to execute on a dedicated LAN with an additional option of failover of that task to another set of LAN connections. The system shall also provide a SCADA synchronization monitor which will provide a running trail of all transactions executed between the primary and backup SCADA server. Both primary and backup server will be collecting information from the process hardware. When the connection from the client node to the primary server is lost, the system must allow for the client to automatically switch over to the backup. After a fail over has occurred the backup node will become that primary node and once the original primary node is fixed or replaced it will be the backup node.

14.2 LAN Redundancy

The system must support 2 physical network connections between a client node and a server node. Both network paths will be for the same process data and when the connection from one network path is lost the system will automatically fail over to the other path.
15 Software Tools

The system must be built on and use industry standard development tools.

15.1 Language

The underlying system must be written predominantly in the C and C++ languages (Microsoft compilers).

15.2 Data Access

The system shall provide an open architecture that allows interaction with other programs. It must provide a mechanism for other programs to access individual data elements and fields (such as the high alarm limit of an analog input) within data elements in real time. File transfer mechanisms are not acceptable; the access must be direct to the memory-resident database.

The following shall be supported:

- **ODBC**
  The system shall support Open Database Connectivity (ODBC) for sharing data from its database to any other ODBC complaint database through SQL queries, via an ODBC dynamic-link library (DLL) driver. At a minimum the database shall support communication to Microsoft Access, SQL Server, and Oracle.

- **OLE for Process Control (OPC).**
  The system should be an OPC-DA Client to access local and remote OPC DA Servers. The system should be an OPC DA Server to make data available to other OPC DA Clients. The system should be an OPC A&E Server to share alarm and event information with other OPC A&E Clients. The system should be an OPC A&E Client, to view alarms and Events from other OPC A&E Servers.

- **Visual Basic for Applications (VBA).**
  The system must have VBA embedded as part of the development environment. VBA support will be used for pre built scripts & custom scripts. It must also support search and replace and the ability to copy all forms modules and scripts from one object to the next.

- **OCX or Active X**
  The system must support the ability to have any third party OCX (Active X control) placed into its container. All third party controls must have the right to behave like any object created by the system. Also the system must contain any bad or misbehaving OCX or Active X control and be able to shutdown the control without shutting down the graphic picture, system, or Node.

15.3 Change Control

The System will offer an optional Change Management feature, tightly integrated with the system development menus. The ability to check out and check in files, complete with developer comments and Optional Electronic Signature authentication will be supported via right click menus in the system development environment. The user will not have to go to another application in order to manage application files from a remote Change Management server. Change Management will apply to all application files that make up a system.
16 Languages Supported

The software shall have been designed to be enabled for language translation.

16.1 Dynamic Language Switching

The software will support the exporting of Pictures for the purposes of translation. The resulting translation texts will be dynamically loaded as a system parameter at startup and can be loaded dynamically with the opening of a picture. This feature will facilitate use by multiple operators, enabling each operator to run the system in their native language.
17  Services

17.1  Training
An interactive on-line tutorial shall be provided as part of the software to teach the basic operations of the system, including graphics and tag development. The tutorial shall demonstrate the configuration operations using interactive on-screen instructions.

Standard classroom courses for engineers and operators of the system that cover the configuration and use of the system shall be available.

17.2  Warranty and Customer Support
The vendor must provide the capability of delivering extended warranty support, as well as after-hours warranty support. The vendor shall specify the length of the warranty (3 month minimum is required), availability and cost of extended support, and type of customer support provided (dedicated department or handled by programming staff or distributor, telephone hours, Email & Web support, FTP download area, Knowledge Base, bulletin board service, field service, etc.)

Twenty-four (24) hour, seven (7) day a week support shall be available from the vendor.

17.3  Software Improvements & “Bug” Fixes
The vendor shall have a location on their web site where users can download software improvements, bug fixes, add-ons, components and so forth. The vendor shall provide a mechanism to install software improvements and bug fixes without the need for uninstalling or reinstalling the software application – only the changed components need to be replaced or upgraded. The vendor shall also provide an easy mechanism for upgrading and installing software improvements and for allowing a user to quickly ascertain what improvements have been installed.

17.4  Quality Assurance
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.

The vendor must also demonstrate that its source code for the product is regularly archived both on-site and off-site in facilities suitable to withstand physical harm.
18 Vendor Requirements

18.1 Development Life Cycle
18.1.1 The vendor must have an established development life cycle that allows for traceability of features and functions throughout that life cycle.
18.1.2 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.
18.1.3 The vendor must also demonstrate that its source code for the product is regularly archived both on-site and off-site in facilities suitable to withstand physical harm.
18.1.4 The vendor shall allow for on-site auditing of the development life cycle to ensure good practice.

18.2 ISO 9001 certified
18.2.1 The vendor must be able to demonstrate that it has established procedures.
18.2.2 Vendor needs to be certified under the ISO 9001-2001 guidelines.

18.3 Preferred Vendor / Manufacturer
18.3.1 Pre-evaluation has identified that Proficy™ HMI / SCADA -- iFIX products from GE Fanuc Automation as the preferred software solution. Any proposed solution must include at a minimum the functionality contained in the current commercially available version of Proficy™ Change Management.
18.3.2 Licensing will be provided to support XX simultaneous users and will include an annualized contract for Proficy™ GlobalCare Support.