



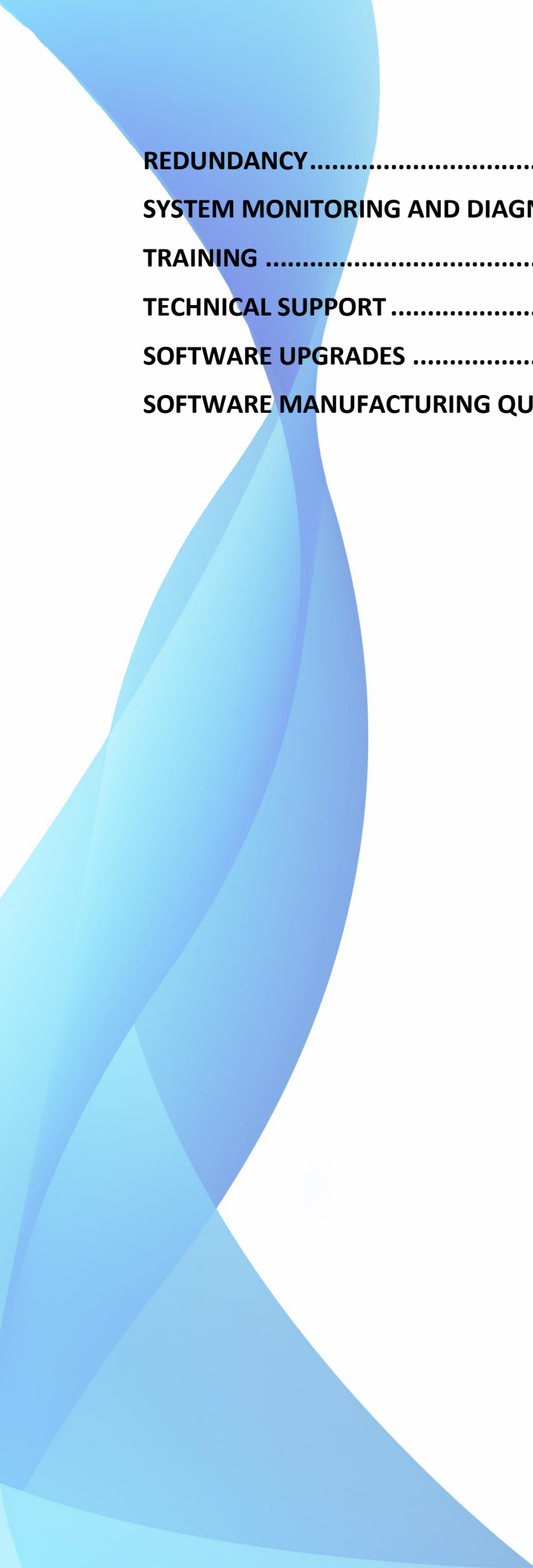
HMI/SCADA Suite Product Specification

CONTENTS

INTRODUCTION	1
SYSTEM OPERATING SOFTWARE	1
SOFTWARE ARCHITECTURE	2
General Design	2
Data Connectivity	3
Networking.....	3
Miscellaneous	4
SYSTEM HARDWARE	4
Hardware Unit.....	4
Display	5
Keyboard and Mouse	5
Printer and Peripherals	5
CENTRAL CONFIGURATION AND PROJECT MANAGEMENT MODULE	5
General.....	5
Shared Services	7
GRAPHIC DISPLAY SOFTWARE MODULE.....	8
General.....	8
Configuration Workspace	9
Graphical Capabilities	10
Graphical Configuration – General	10
Graphical Visualization.....	11
Graphical Configuration – 2D Graphics.....	11
Graphical Configuration – 3D Graphics.....	13
Drawing Utility Tools.....	15
Animation Tools	17
Dynamic Connections	17
General.....	17
Size Dynamic	18

Rotation Dynamic.....	19
Digital Location Dynamic	19
Dial Dynamic	20
Hide Dynamic	20
Enabled/Disable Dynamic	20
Digital Color Dynamic.....	21
Analog Color Dynamic.....	21
Button Action Dynamic	21
Value Dynamic	23
Data Entry Dynamic	24
Time and Date Dynamic.....	25
State Field Dynamic.....	25
State Field Data Entry Dynamic.....	26
State Dynamic	26
Range Dynamic.....	26
Module Controls	27
Symbol Storage	28
Expressions.....	29
Layers	30
Scripting	30
Other Features	30
GEO Spatial Mapping Module.....	32
ASSET MANAGEMENT MODULE.....	32
General.....	32
Communication.....	33
TRENDING AND DATA LOGGING SOFTWARE MODULE.....	34
General.....	34
Historical Data Logging	34
Data Trending/Charting	36
Data Trending/Charting Configuration	36
Data Trending/Charting Runtime.....	38
ALARM MONITORING MODULE	39

General.....	39
Alarm Server	39
Alarm Server – Digital Alarms	40
Alarm Server – Analog Limit Alarms	41
Alarm Server – Deviation Alarms	41
Alarm Server – Rate of Change Alarms	41
Alarm Server – Alarm Toggle Count.....	41
Alarm Server – Time in Alarm	41
Multimedia Alarm Notification	42
Alarm Logger	42
Alarm Live/Active Alarm Viewing Configuration	43
Alarm Live/Active Alarm Viewing Runtime	44
Historical Alarm Reporting.....	45
GRID VISUALIZATION MODULE	46
General.....	46
Grid Visualization Configuration	46
Grid Visualization Runtime	46
FAULT DETECTION AND DIAGNOSTICS MODULE	47
General.....	47
Fault Detection and Diagnostic Engine	47
Fault Detection and Diagnostic Visualization Runtime	47
SCHEDULING MODULE	48
SECURITY	48
WEB AND INTERNET CONNECTIVITY	50
DATA SOURCES	51
Data Manager	51
OPC Connectivity.....	51
BACnet	52
SNMP Compliant Managed Device Connectivity	52
ALIASING	53
LANGUAGE SWITCHING	53



REDUNDANCY.....	54
SYSTEM MONITORING AND DIAGNOSTICS.....	55
TRAINING	55
TECHNICAL SUPPORT	55
SOFTWARE UPGRADES	56
SOFTWARE MANUFACTURING QUALIFICATIONS	56

HMI/SCADA Suite Product Specification

INTRODUCTION

The 64-Bit HMI/SCADA Automation Solution (referenced within this document as "The SOLUTION") will manage data and provide a graphical interface for the monitoring, supervisory control, data and alarm archiving and management of all equipment, processes and systems identified herein.

The Solution will exchange data with all control equipment, plant equipment, Manufacturing Intelligence and Business Intelligence systems. These will include as specified herein, all PLC's, Distributed Control Systems, embedded equipment controls, networked instruments, electric metering systems, production planning systems, Quality Systems, ERP systems and accounting systems.

The SOLUTION will exchange data with building sub-systems, which may include, but is not limited to: HVAC monitoring/control, lighting control, elevator monitoring, direct digital control (DDC), electronic access control, alarm monitoring, lab systems, electric power metering/control, sun louver/window shade control, wastewater/water harvesting control, fire alarm status control, conveying equipment, exterior building maintenance, communications, digital video surveillance, solar energy collection, metering, seismic/weather station monitoring and/or other building management-related applications.

The SOLUTION will also exchange data with existing enterprise and business-level software systems. The SOLUTION will have the capabilities to read (monitor) or write (modify) to each of the above sub-systems using open communication protocol standards via the building's IT network, including but not limited to OPC UA and/or OPC Classic, BACnet/IP, LON/IP, Modbus/IP, TCP/IP, and SNMP.

SYSTEM OPERATING SOFTWARE

The SOLUTION shall be compatible and function fully with Microsoft Windows operating systems including the following:

- Windows Server 2008 x64
- Windows Vista x64
- Windows Server 2008 R2 x64
- Windows Server 2012
- Windows 7 x64
- Windows 8 x64

The SOLUTION's vendor should have a strategy for upward compatibility with future operating systems and technologies.

SOFTWARE ARCHITECTURE

General Design

The SOLUTION shall be based on a modular architecture and be inherently based on a distributed architecture that supports Microsoft Windows networking as well as OPC-based communication technology. The SOLUTION should leverage technologies such as Microsoft Silverlight, Windows Communication Foundation (WCF) and Web Services to allow for enhanced security, flexibility and extensibility.

The SOLUTION must be based in 64-bit computing in order to leverage the advantages of larger memory capacity (greater than 4 GB of RAM), hardware acceleration, multi-core processors and multi-threading.

The SOLUTION must be capable of managing large scale SCADA systems in excess of 1 million tags (supported by properly scaled hardware or server-client architectures). It should also provide the flexibility to purchase and implement only the modules and components needed for the application. For example, if only alarming is required on one station, then only the alarming module should have to be purchased (and not the graphics, trending, etc. for that node). With that in mind, the SOLUTION vendor should be able to provide separate stand-alone modules for Dynamic Graphic Displays; Asset Management, Real-Time Trending and Historical Data Logging; Alarm Management; Security; GEO-SCADA; Data Management; Fault Detection and Diagnostics, Real-time OPC and OPC UA Servers; Multi-level Redundancy; on-demand Reporting; and OPC Data Bridging. The SOLUTION should have a scalable architecture, such that the application can grow as required.

The SOLUTION should ensure that when any application module needs to be accessed by an Operator-level User, it should be accessible with user-level security appropriately providing access to the specific display, process values, and commands. The SOLUTION should offer the means to customize security policies, groups and users.

The SOLUTION should make use of, and spread, application loads across multiple processors on a single machine, as well as support multi-threading, Web-based thin client visualization, local and remote server access, Open Standards, client/server implementation, IPv4/IPv6, redundant servers, VPN tunneling/IPSec and multiple protocols on the same server.

The SOLUTION should include a legacy support path via an HMI Display Translator from older versions.

The SOLUTION should be compliant with virtualization tools and should take steps to verify and prove compatibility with virtualization solutions such as VMWare.

Data Connectivity

Connectivity to Data Sources shall be through a universal data browser that provides the ability to specify the tag or point name using point and click selection with a mouse. The software package should support and be certified for OPC UA connectivity across all modules.

The modules should support Classic OPC, OPC DA, OPC HDA, OPC A/E, OPC .NET (formally Xi) and all OPC UA types. Support for BACnet, SNMP, Modbus IP and Web Services (SOAP/XML) should be considered "native" to the application.

The universal data browser shall serve as a single tag browser across all types of OPC tags (real-time and historical), global aliases, local aliases, language aliases, and local variables. Data Sources shall be divided into tabs and sub menus including OPC Data Access (including specification versions 1.0, 2.05 and 3.0 and OPC XML DA), My Computer, Network, Global Aliases, Language Aliases or conversion, and Expression-based editing of any of the other types.

The universal data browser tool shall be persistent, resizable and multi-threaded, as well as context-sensitive or "smart" by limiting the options to those available within the launching module. All modules and applications in the HMI/SCADA system should utilize the universal data browser for data connectivity.

Networking

The SOLUTION shall implement Client/Server Distributed Networking, as well as support industry standard network protocols such as TCP/IP. It should also support OPC over IP and/or SOAP/XML (configurable on a per-node basis). The SOLUTION should also support IPv4 or IPv6 addressing.

The SOLUTION should operate on a network with multiple protocol stacks and "Plug and Play" with standard routers, network intranets and the Internet. It should offer a Web-based Browser Plug-in so that alarms, graphics and trends can be viewed from an Internet Explorer station. Also, it must permit interaction (e.g., changing set point, switching screens, etc.) from these Internet Explorer stations.

An option for Microsoft Silverlight graphic publishing should allow the user to utilize the system with cross-browser and cross-operating system functionality. The SOLUTION should permit users to perform any function in the SCADA software from any node on the network and it should be configurable as an independent distributed database, regardless of the number of nodes in the system. It should allow configuration changes to the distributed database from any node on the system, as well as connect to a large number of I/O Device Interfaces via OPC Servers across the network.

An optional add-on product should be available to provide HMI visualization on all major tablet and phone providers. This visualization should be based on the pervasive technology of HTML (specifically HTML5 for the latest advances).

The SOLUTION should include an option to network nodes to create "load balancing." This option should be configurable for multiple server nodes to maintain larger applications. Operation of this "load balancing" should be transparent to the users.

Miscellaneous

The SOLUTION should provide software-based licensing capability that does not require a hardware "dongle" key for licensing. It should provide an option for hardware "dongle" keys for licensing (like those plugged into the USB port) as an alternative to any software-based licensing. A demo or trial mode should be provided for users to configure and view their system without purchasing a license, for testing purposes.

The SOLUTION should allow such licensing to reside on each individual node in the system, or be served up from one central server (whereby all other nodes point back to that one station for its licensing). A combination should also be allowed. The SOLUTION should support Tool Tips, dockable toolbars, an intuitive ribbon menu, context sensitive help and right-click mouse function for easy context menu commands.

The SOLUTION should be supplied as a complete package. No additional software should be required to configure or run the features of the system.

SYSTEM HARDWARE

Hardware Unit

The hardware running the SOLUTION should have at least a dual core 64-bit processor, at least 4 GB of physical memory (RAM) and the ability to support up to 2 terabytes (TB) of RAM for scalability purposes. The SOLUTION itself should

be under 10 GB of disk space, while the machine capacity should be at least 6 GB on installation.

The hardware running the SOLUTION should have at least an 8x speed DVD-ROM Drive. All Communication Port Hardware necessary to interface to the field devices shall be included. Communication cards and interconnecting hardware shall be used, if applicable, to connect the SCADA/HMI computers to a Local Area Network.

The graphics of the machine should be supported by a dedicated graphics card with over 500MB of onboard memory.

The hardware running the SOLUTION should contain one of the following operating systems: Windows 8 x64, Windows 7 x64, Windows Server 2012, Windows Server 2008 R2 x64, Windows Vista x64 or Windows Server 2008 x64.

Display

The hardware running the SOLUTION should include an LCD/LED display that has a minimum resolution of 1024 x 768, supports at least 32-bit color, integrates with an included video card with DirectX 9 or 10 (or better), is Energy Star-compliant, and is touch-enabled (Optional based on application).

Keyboard and Mouse

The hardware running the SOLUTION should include a full 80-101 key ASCII keyboard or access to one through OS-provided software. It should have the standard QWERTY-style keyboard with cursor control, numeric keypad and at least 10 special function keys. It should also have a standard Microsoft mouse for point and click system configuration.

Printer and Peripherals

The hardware running the SOLUTION should have the ability to utilize a color ink jet printer for documentation of graphic displays.

CENTRAL CONFIGURATION AND PROJECT MANAGEMENT MODULE

General

The SOLUTION should include a centralized configuration workspace that can function as a center point for development. This module should include a categorization for shared services across the SOLUTION.

The SOLUTION should support both a web and desktop based configuration workspace. Both should offer most if not all of the functionality available from the combined workspace.

The SOLUTION's central interface should include project management tools including import and export tools to common formats for mass or batch editing of configurations. Standard formats such as .CSV, Microsoft Excel and .XML should be supported. Whole or partial configurations should be allowed for import/export tasks.

The SOLUTION's central interface should allow for web or remote configuration. The SOLUTION should thus be configurable from anywhere.

Deployment should be supported within the central interface through a simple wizard for packing up configuration databases. Operations such as find and replace should be supported.

The SOLUTION's central interface should also allow support for runtime operations including display of multiple HMI screens, trending screens, alarming screens and layout management capabilities.

Data connection within the SOLUTION's central interface should be through a standard form or dialog. Although the application should be able to see third party data, a wrapper for the application's data should also provide data from within the application to outside sources. Enterprise wide scripting should also be available in the SOLUTION to tie values together during runtime.

The SOLUTION's Project Management feature should integrate all of the applications outlined in this specification into a single format or container, embedded in Microsoft Management Console. It should support a unified configuration of all modules, allow for third party integration of modules, be able to manage more than one project at a time, and have the ability to launch and make a project active with one command.

The SOLUTION's Project Management feature should permit association of at least documents/files of the type provided by the HMI/SCADA software. It should also support project deployment.

The SOLUTION's Project Management feature should support the ability to Pack/Zip up all of the files required for a project into a single, compressed file format for easy deployment from one machine to another. It should also be able to Unpack/Unzip a project back into the proper locations on the same or other PC.

The SOLUTION's Project Management feature should permit "versioning" of the each of the files used in the project.

The SOLUTION's Project Management feature should manage the publishing of Trend and Alarm configurations as well as entire projects to the Web (Intranet/Internet). It should generate reports on the resources used for each project (e.g., listing the files, types, etc.). Such reports shall be able to be created on either the entire project, or just the files associated with a particular application used within the project (e.g., Graphics Module). The reports shall list all parts of the project, giving details as to when each file was created, modified, its size, as well as giving an summary of entire project disk space required.

The SOLUTION's Project Management feature should have an option to list all OPC Tag names used in a project, and flagging any points that cannot be verified. It should have optional project summary information field for such items as: Project Name; Project Version Number; Author Name(s); Last Modified By; and Comments.

The SOLUTION's Project Management feature should permit locking a packed project via a password (so it cannot be unpacked without proper authorization). It should also serve as a location from where users can launch each application associated with a project, as well as be able to "Start/Activate" the entire project (putting the various configured components into "runtime" mode, for example).

The SOLUTION's Project Management feature should offer links to tools for trouble-shooting a project (e.g., network connection diagnostics, OPC Server diagnostics, etc.). It should also provide a utility to globally search and/or replace text throughout a project and all of its sub-documents including individual files, as well as support Application configurations in SQL.

Shared Services

The SOLUTION should offer some level of global communication for issuing commands or directions from one module to another. These "commands" should be executed from the asset management module or from any graphic as defined in the graphics module.

"Commands" should include, but are not limited to options for opening new HMI screens, creating/visualizing trends of an asset or filtering alarms on a specific asset or parent level (i.e., building, production line).

The SOLUTION's central configuration module should support a number of shared services including Web Services and Database Services.

Web Services should be configurable and extendable by allowing users to add their own to the configuration. The SOLUTION should provide a simple method for doing this. Web Services support shall include private web services and as well

as public internet webs services such as weather, electricity prices, oil prices, exchange rates and more.

One or more Web Services should come pre-configured with the initial installation of the SOLUTION.

The SOLUTION should allow multiple types of database connectivity for use in the software. Database connectivity should include support for SQL, SAP, ODBC, OLEDB, DB, DB2, SQL Query, MySQL and Datasets as standard connections.

Database Services should be central to the SOLUTION and accessible through the other modules in the SOLUTION.

The SOLUTION should be certified for the highest level of BACnet communication.

The SOLUTION should offer SNMP (at least Version 1 and 2) support for monitoring computer and networking equipment.

GRAPHIC DISPLAY SOFTWARE MODULE

General

The SOLUTION should not require compilation before use. This should be accomplished by a "save" function or, in Web-related cases, a "publish" function.

The SOLUTION should auto-generate an HTML reference page for use in the web server module when appropriate. Users should not be required to create this reference.

The SOLUTION should permit having multiple instances of the development environment open at the same time within the same PC. It should also permit drag-n-drop of static and dynamic objects between the multiple instances of the development system for moving and/or copying of parts of the display.

The SOLUTION should allow multiple graphic displays to be viewed on the screen simultaneously. All graphic displays shall be able to be configured as either "Fixed scale" or as "Scalable", which would allow automatic scaling of the display so as to maintain a full view of the graphics regardless of the window size and/or screen resolution. Graphical displays shall have no hard coded limitations for display size.

The SOLUTION should have multi-touch capabilities allowing for single tap, double tap and swipe type gestures. These gestures or touches should be integrated as options for buttons or click actions.

The SOLUTION should provide the capability to contain any number of objects (no hard coded maximum). Graphic displays shall have the ability to turn off their Menu and/or Title bar during runtime, as well as make their window non-re-sizable during runtime. An option shall be provided to enable a display (on a per-display basis) to always be "On Top".

The SOLUTION should support networked "Portable Displays", whereby completed graphical displays can be copied to any other node without the need to edit and/or re-establish dynamic connections. The module should support "compressed displays" by using a compression algorithm to allow easier transfer of displays over the Web.

An option shall be provided to show tool tips during runtime (and configuration mode) that can be customized to show any combination of the following: Tag used (for dynamic objects), Process Value (for dynamic objects), and User defined text for help/description.

The SOLUTION should provide the capability to develop and alter graphics from Internet Explorer. It should take advantage of new multi-touch and Operating System integration (e.g., Jump Lists in Windows 7) and include an API for the module scripting language to allow users to extend the functionality of the software package. It should also include an API for symbols within the graphic display module.

The SOLUTION should allow for Password Protection of graphical displays. For security purposes a pre-configured "Access Denied" screen should be available. The "Access Denied" display should be customizable.

Configuration Workspace

The SOLUTION should provide a simple and intuitive configuration interface for creating and editing displays. It should allow configuration of complex screens with animations and customization of the configuration workspace through dockable tool windows.

The SOLUTION should allow easy access to object and display properties, application preferences, templates, Trend and Alarm controls, the scripting interface, and pre-configured or user symbols.

The SOLUTION should allow the creation of user symbols or "smart" symbols, and provide standard operating commands (such as Save, Save As, Print, etc.). It should utilize the latest Microsoft Office ribbon technology for ease of use and familiarity. The module should allow easy Web publishing, which should be allowed for both Windows Presentation Foundation (WPF) and

Silverlight displays in the same manner and which should be a simple step-by-step process.

Graphical Capabilities

The SOLUTION should utilize hardware acceleration for graphics, be capable of creating 2D vector graphics, as well as be capable of creating and integrating 3D vector graphics.

The SOLUTION should allow for multiple client types for displays allowing for Microsoft Windows Presentation Foundation (WPF) and Microsoft Silverlight displays for cross browser support.

The SOLUTION should allow displays and other "controls" to be used in a dashboard type frame-based visualization area.

The SOLUTION should include a simple method of incorporating assets, trending, alarming, faults and mapping to any display. This can be done through .NET "controls" or similar method.

The SOLUTION should support video (embedded and streaming). Standard static protocols such as WMV, AVI, MPEG and MP4 as well as streaming protocols such as HTTP should also be supported.

Graphical Configuration – General

The SOLUTION should permit the use of a mouse for creating, editing and placing objects. It should have the following options: to utilize a grid and "snap to" feature to assist configuration, to customize grid size and spacing; to create primitive shapes; to align/combine/arrange the depth of primitives; to create runtime enabled/disabled layers to organize displays; to create groups or symbols to organize primitives; to resize primitives using the standard bounding box handles; to rotate primitives using the standard bounding box rotation; to maintain aspect ratio during resize or rotate using a modifier key; and to copy formatting between objects.

The SOLUTION should have pre-configured styles as well as configurable fill colors, fill patterns, line thicknesses, line styles and line colors. It should have a fill gradient tool that includes a configurable style, number of steps and color for each step. It should have configurable effects such as Opacity, Shadow, Blur, Bevel and Glow. It should have preset styles with customization (add, delete or reset) and a way to show customizable properties for selected objects or controls.

The SOLUTION should provide a dialog or wizard for setting up the file type, resolution, compression and name. This dialog or wizard should provide template capabilities relating to predefined screen sizes to accommodate for desktop, web and mobile displays.

Graphical Visualization

The SOLUTION should provide multiple formats for viewing HMI screens including, but not limited to: Desktop; Web; Dashboard; and Mobile (smartphones and tablets).

The SOLUTION's desktop functionality should be fully featured and allow all types of visualization "controls" and objects. The viewer should allow functionality for "full-screen" operations forcing users to only use the software.

The SOLUTION's web functionality can be a subset of the fully featured software, but should offer a "thin" and "thick" option.

The SOLUTION's dashboard technology should allow for a structured approach using "frame" based visualization. The "frames" should be capable of containing any other module viewer. Communication such as data reads and writes between "frames" must be supported.

Graphical Configuration – 2D Graphics

The SOLUTION should utilize a 2D graphical configuration that includes the following "primitive" objects: Line, Arc, Polyline, Rectangle, Ellipse, Polygon, Text, and Picture (embedded or referenced). It should allow primitives to be drawn to any length or size.

The SOLUTION should allow the following functions for a Line: to snap to certain degrees of inclination (e.g., 45 degrees) using a modifier key (e.g., SHIFT); joining of lines into a polygon which can be filled; adding/deleting/editing anchor points of a line; and to use a property panel to change color/size/center/style.

The SOLUTION should allow the following functions for an Arc: to be drawn as segments of an ellipse or circle; to be changed by grabbing the end points; to be drawn with at least single degree increments; to have a fill color; to change into a Pie (drawing a line from the center of an ellipse to create a "slice" of an ellipse or circle); to change into a Chord (connecting a straight line between endpoints; to be converted into a polyline equivalent so that the arc can be modified and/or merged with others; to be converted into a Path/Polygon/Image to be converted into a Polygon; and to use a property panel to change color/size/center/style.

The SOLUTION should allow the following functions for a Spline: to be drawn via consecutive mouse clicks until complete; to allow properties to define the spline style/width/color/cap type; to have an option to "close" the spline automatically; to be converted into a Polygon/Polyline/Image; and to use a property panel to change color/size/center/style.

The SOLUTION should allow the following functions for a Polyline: to be drawn via consecutive mouse clicks until complete; to be drawn at specific angles using a modifier key (e.g., SHIFT); to have a way to edit the points via text to modify the polyline after creation; and to allow properties to define the polyline style/width/color/ cap type.

The SOLUTION should allow the following functions for a Rectangle: to allow a modifier key (e.g., SHIFT) to be used to create a square; allow the mouse to define the corner locations; to allow rounded corners to be applied after rectangle creation; to allow axis independent rounding (separate values for x and y axis); to allow a rectangle to be converted into a Polygon; to allow a rectangle to be converted into a Path; to allow a rectangle to be converted into an Image; and to use a property panel to change color/size/center/style.

The SOLUTION should allow the following functions for an Ellipse: to allow a modifier key (e.g., SHIFT) to be used to create a circle; to allow an ellipse to be converted into a Polygon; to allow an ellipse to be converted into a Path; to allow an ellipse to be converted into an Image; and to use a property panel to change color/size/center/style.

The SOLUTION should allow the following functions for a Polygon: to be formed with joined "multi-segmented" lines; editing after completion via point handles; use a property panel to change color/size/center/style; and to allow lines to snap to certain degrees of inclination (e.g. 45 degrees).

The SOLUTION should allow the following functions for Text: to allow selection of a font via an easy to use list with samples; configurable font size on any object; configurable font style on any object; to allow multi-line text to remain as one object; allow text alignment within the boundaries of the text object; allow an option to "fit" the text size to its boundaries; allow an option to maintain the text's size while resizing the boundaries; allow a background color to be chosen that fills the rectangular boundary; allow rotation of the text object; and allow decorations such as Strikethrough and Baseline.

The SOLUTION should allow the following functions for a Picture: allow the image to be resized with or without maintaining the aspect ratio; support compression of files; allow embedded pictures or referenced pictures in displays; have an option to reset to the default image size and shape; allow replacement of an image with another; select a "transparent color" for an image (e.g.,

selecting blue would eliminate the display of that shade of blue within the image, thereby revealing any object behind the image; allow insertion of any of the following file types: Windows bitmaps (*.BMP), GIF (*.GIF), JPEG (*.JPG;*.JPEG), Icon (*.ICO), Portable Network Graphics (*.PNG), Tag Image File Format (*.TIF;*.TIFF), Microsoft Visio Files (*.EMF), Autodesk AutoCAD Files (*.DWG) Metafiles (*.WMF, *.EMF); Allow insertion of either .wmf or .emf metafiles; allow resizing of metafiles with or without maintaining the aspect ratio; and allow a metafile to be converted to constituent shapes and grouped as a symbol.

The SOLUTION should allow the following functions for WPF and WinForms Controls: ability to insert and run multiple WPF/WinForms Controls from third party vendors; bring up the WPF/WinForms Property Inspector for the embedded controls so that it can be customized for each instance; and provide links to the scripting environment to directly manipulate the WPF/WinForms Control and capture any Events it may trigger.

It should be possible to configure "safe zones" for multi-touch interaction where a button must be "depressed" for others to be "enabled".

Graphical Configuration – 3D Graphics

The SOLUTION should utilize a graphics package that offers an extensive 3D configuration environment, meant to allow users to configure complex 3D environments that accurately emulate real world conditions. It should include the following 3D primitives: Sphere; Cube; Cone; Cylinder; Pipes; Torus; Terrain; and 2D planes. A pipe primitive should be able to be moved, extended or changed by a simple user manipulation.

The SOLUTION should include Import functionality that can be implemented to include the following standard formats: *.DWG, *.3DS, *.DAE (COLLADA), *.OBJ, and *.XAML.

The SOLUTION should allow Users to select objects in the following ways: Select an object directly; Select all objects of a certain type directly (e.g., all cubes); Select all objects in a 3D view; Select all objects with configured dynamic properties; and Select a group of objects instead of the individual objects within a group.

The SOLUTION should allow certain sub-level selection on primitives, including the selection of: the Vertices of an object individually; the Edges of an object individually; and the Faces of an object individually. It should also allow for selection of the smallest subset of an object (e.g., the planer triangles that are often used to create 3D objects).

The SOLUTION should provide Users with the ability to Move objects on the screen by different methods including: On-screen mouse manipulation using three dimensional handles; Nudge buttons in all three dimensions that move the object using a variable input that is configurable by the user; and Arrow keys that move the selected object(s) using a variable input that is configurable by the user.

The SOLUTION should provide Users with the ability to Scale objects on the screen by different methods including: On screen mouse manipulation using three dimensional handles; Nudge buttons in all three dimensions that scale the object using a variable input that is configurable by the user; and Arrow keys that scale the selected object(s) using a variable input that is configurable by the user.

The SOLUTION should provide Users with the ability to Rotate objects on the screen by different methods including: On screen mouse manipulation using three dimensional handles; Nudge buttons in all three dimensions that rotate the object using a variable input that is configurable by the user; and Arrow keys that rotate the selected object(s) using a variable input that is configurable by the user.

The SOLUTION should allow transformation manipulation and should include the following operations: Reflect over the x, y, or z-axis; Integer based scale, rotation and translation; and Extrude face.

The SOLUTION should allow duplication while using a transform in the following ways: Duplicate in a grid; Duplicate in a line; Duplicate and rotate; and Duplicate and scatter.

The SOLUTION should allow for Grouping of 3D objects. It should also allow for alignment of 3D objects in the following manners: Align Left (x-axis); Align Right (x-axis); Align Top (y-axis); Align Bottom (y-axis); Align Front (z-axis); Align Back (z-axis); Align Center (all axes); Stack in depth (all axes); Make the same size (all axes); Center All; and Make all the same size.

The SOLUTION should include materials to “map” onto 3D surfaces to allow greater flexibility in design. Material configuration should include: An extensive range of over 256 distinct colors to apply to 3D objects; 3D objects should allow image textures to be applied; and Image textures should be allowed as embedded or referenced images. It should have some preconfigured colors or styles that are easy to apply, have the ability to map different colors or styles to the front and back of an object, and have the ability to map two dimensional entities such as process points to a three dimensional object. The type of mapping should be configurable and include: planar mapping; spherical mapping; cylindrical mapping; and Automatic mapping.

The SOLUTION should allow animation based on changes in values or a timer in the following ways: Direct Variable Display; Text; Fill Color; Visibility/Invisibility; Movement along a path; Rotation; and Resizing (must include option to maintain aspect ratio).

The SOLUTION's graphics package should allow users to switch between a few pre-configured views like Top, Side and Perspective, as well as allow for user configured views.

The SOLUTION's 3D entity of the graphics package should allow for complex animations based on user configured views. The "camera" animation should be able to move about these views with options including a user configured list of views to include in the animation and user configured duration to stay on each view. The animation should be able to be triggered by an inactivity time.

The SOLUTION should allow configuration with and without scripting. Also, an advanced option of Collision Detection should be available. Collision Detection should allow the user to choose what objects are included in analysis to avoid calculation for objects the user knows will never touch. Collision Detection should also have optional visual feedback, such as a color change to indicate collisions, as well as a configurable quality and should allow Alarm logging based on events.

The 3D graphic configuration of imported objects should allow for a "polygon reduction" option that will allow imported items to reduce the size of displays and the number for polygons or triangles that need to be rendered in real-time improving performance.

The SOLUTION's Defaults should also be configurable. All properties should be able to be changed and set as default through a preferences menu.

Drawing Utility Tools

The SOLUTION should include a Color Palette that supports at least 256 colors and allows custom colors via direct editing of RGB values. It should also include an Eye Dropper tool to select a color from a picture to use and a Group/Ungroup Tool to define if selected objects are grouped into one symbol or not. The Grouped symbols must still be able to be edited individually without destruction of the grouping or any dynamics associated with that group. For example, it should be possible to edit a pump symbol having color and pick dynamics on the pump as a whole (changing perhaps one of the individual line or arc objects used in drawing the pump) without first having to ungroup, and thereby losing the dynamics. All must be preserved.

The SOLUTION should provide a method for naming colors for reuse. "Named Colors" should be referenced and part of a global Color Palette for each application. Colors should support referenced updating to prevent users from having to update every instance of the Named Color manually.

The SOLUTION should include: Bring-to-Front/Send-To-Back Tools to move the Z-order of any object(s) forward or backward and a Delete Tool to delete selected object(s). It should also include Alignment Tools to line up multiple selected objects based on: Tops; Left Sides; Right Sides; Bottoms; Middles in the Vertical Direction; Centers in the Horizontal Direction; Evenly Distributed Across (with boundaries defined by left-most and right-most selected objects); and Evenly Distributed Down (with boundaries defined by top-most and bottom-most selected objects).

The SOLUTION should include: a Make Same Height Tool to make all selected objects the same vertical length as one of the selected objects; a Make Same Width Tool to make all selected objects the same horizontal size as one of the selected objects; a Make Same Size Tool to make all selected objects the same size (both height and width) as one of the selected objects; and a Make Same Style tool. This can be accomplished with a "Copy, Paste" action through a context menu.

The SOLUTION should include a Search/Replace Tool to allow search and replace editing of: Dynamic connection tag names (described later); Filenames used in various "Pick Action" dynamics (described later); Text used in Text Object labels; and Text in Referenced images. The "Scope" of the Search/Replace shall be user selectable as: Only those objects selected; The Entire Display; and Multiple Displays as selected by the user (so an entire project can be updated at once).

The SOLUTION should include Zoom options for finer control of drawing and editing of objects, including: Predefined percentages for easy access (e.g., 50%, 75%, 100%, 200%); Custom Percentage (user enters a number between 10% and 1000%); Show the Entire Display; Zoom to the Selected Objects; and "Rubber band" or "Box" Zoom (whereby the user defines a "rectangle" defining the area to zoom to by dragging the mouse). This can be done using a modifier key (e.g., Alt); Return to the "Home" view, the default used for the display; Create editable "Custom Views"; Be able to add delete or modify "Custom Views"; and Zoom in/out around the mouse pointer by using a modifier key (e.g., Alt) and rolling the center Mouse Wheel (for mice so equipped).

The SOLUTION should include an Auto-Panning feature, so that when objects are drawn or moved beyond the zoomed section of the display, the area auto-scrolls in the direction of the mouse drag.

Animation Tools

The SOLUTION should include Animations, which should be configurable with a point and click interface or "Browser", which should support local and network point configuration without additional user configuration. Animations should be configurable with or without scripting.

The SOLUTION should allow for multiple dynamics of differing type to be applied to the same object (e.g., a size dynamic, and a color dynamic both added to a rectangle). It should also allow for multiple dynamics of the SAME type to be applied to objects (e.g., one size dynamic tied to one variable, and another size dynamic tied to another variable both added to a rectangle).

Only applied dynamics associated with the object shall be listed. The option shall be available to only copy dynamic connections from one object to others.

The SOLUTION should include a Format Painter (similar to MS Office Format Painter) for transferring objects' visual style.

Dynamic Connections

General

The SOLUTION should allow for Dynamic connections that should be able to be applied to a 2D or 3D object. Some "dynamics" should work with groups and symbols specifically. Dynamic connections shall include the ability to browse for tags, eliminating the need to type an OPC Tag name. Wherever a tag name can be used, an "Expression" shall also be permitted (Expressions are outlined elsewhere in this specification). Wherever a tag name can be used, a "Local Variable" shall also be permitted. These local variables are unique for each display file, and shall NOT count against any "tag count" limit imposed by the license. Wherever a tag name can be used, an "Alias" shall also be permitted (Aliases are outlined elsewhere in this specification). Wherever a tag name can be used, a "Language Alias" shall also be permitted. The Language Aliases shall allow the user to substitute various language phrases based on a substitution table enabling an operator to read the item in his native language.

The SOLUTION should allow that, should a dynamic connection fail in getting a value from I/O, the dynamic must indicate such failure visually on the screen. Visual Dynamics on graphic objects (size, color, flash, location, etc.) must change the color of the bound object (color user-definable). Data Values shown numerically must have an option to show a user-defined character (question marks or asterisks, for example) instead

of the last known value. Dynamics can be applied to either individual objects, or a group of objects.

The SOLUTION should allow for an optional "tool tip" to display description, data source information and/or error information upon "mouse-over" of the dynamic object.

The SOLUTION should allow that all dynamics respect an optional Application mode to simplify the configuration. They should be able to be configured and viewed through a "Basic," "Standard," and "Advanced" mode. "Custom Data" should be configurable for all dynamics.

The SOLUTION should allow for dynamics to allow an "Analog" and "Discrete" mode to work with analog and digital tags. In "Discrete" mode, a properties menu should be available to configure a delay in the animation start and duration of animation, as well as acceleration or deceleration of the animation over time. In "Discrete" mode, a properties menu should be available to configure a "repeat count" from 1 to infinite. In "Discrete" mode, a properties menu should be available to configure a flash rate or toggle rate.

The user shall be able to select whether it flashes on the "True" or on the "False" condition. The object will flash by hiding the object or by changing the Object's Colors (either its Line color, or Fill color, or both). This should be configurable by the user. The rate at which the object will flash shall be configurable by the user on a per object basis (range at least 50ms - 2 minutes). In "Discrete" mode, a properties menu should be available to configure whether or not the animation will "auto-reverse" or animate in reverse, when the configured condition is not met.

"Analog" data sources should have a configurable start and end percentage to allow customization.

Size Dynamic

The Solution should allow for a Size Dynamic to be applied to Line, Polygon, Arc, Ellipse, Circle, Rectangle, Image, Text, and Metafile 2D or 3D objects, as well as grouped combinations of these objects. The object shall grow in size in one of two methods (selectable on a per-object basis): Scale (the entire object grows/shrinks relative to the analog signal it's connected to); or Clip or Reveal (the dimensions of the overall object remain the same, but the object is revealed relative to the analog signal it's connected to).

The range of the analog connection through which the object can grow shall be configurable by the user. The starting and ending size, as a

percentage, shall be configurable by the user. The size of the object(s) shall be able to be configured to change in the Left, Right, Top and Bottom directions. The size of the object(s) shall be able to be configured to change in any combination of 2 directions simultaneously (e.g., left and right growing from the center, top and right growing from the lower left corner, etc.). The size of the object(s) shall be able to be configured to change in any combination of 3 directions simultaneously (e.g., left, right, and top, etc.). The size of the object(s) shall be able to be configured to change in all four directions simultaneously. The size of the object(s) shall be able to be configured to change in the Left or Right, with a "bias" in the middle (starting in the middle and either going to the left or to the right, but not both at the same time). The size of the object(s) shall be able to be configured to change in the Top or Bottom with a "bias" in the middle (starting in the middle and either going to the top or bottom, but not both at the same time). Horizontal or vertical "anchors" should allow the dynamic to grow or shrink with regard to a certain location.

Rotation Dynamic

The SOLUTION shall allow for a Rotation Dynamic to be applied to a 2D or 3D object, as well as grouped combinations of these objects. The object shall rotate based on the value of a digital or analog signal as it goes through its range. The range of the connection through which the object will rotate color shall be configurable by the user. The starting angle and ending angle shall be configurable by the user, in one-degree increments. The pivot point around which the object rotates shall be configurable by the user on the screen or by the x and y coordinates off of center. Whether the object rotates clockwise or counterclockwise shall be configurable by the user.

Digital Location Dynamic

The SOLUTION shall allow a 2D or 3D object to change its location (move across the screen) based on the value of a digital signal as it goes through its range. The range of the digital connection through which the object will move shall be configurable by the user. The location dynamic should be configurable on the screen with vertices to define the path. The location dynamic should have a secondary editing option that should allow editing of the value of the vertices. Vertices should be able to be configured as a curved or straight vertex. Users should be able to edit, add or remove vertices along the path at any time during configuration. A slider dynamic should be easily created through the location dynamic. Editing the path on the screen should take advantage of configuration tools such as using a modifier key (e.g., Shift) to draw straight or angled

lines as well auto-panning. Digital location dynamics should follow all rules of discrete dynamics listed above.

Dial Dynamic

The SOLUTION shall allow for a dial dynamic to extend the properties of the rotate dynamic. The dial dynamic should allow user interaction. The user will be able to click and drag the object, rotating it around the pivot point defined through the angles defined. As this rotation occurs, it will write values to the connected Analog tag, with the low range corresponding to the start angle, and the high range corresponding to the end angle. The user shall be able to select when the analog values are updated, either: as quickly as possible while the "dial" action is taking place (continuous updates); or only when the user releases the mouse click. The user shall be able to configure whether the rotation of the dial is smooth through the range, or uses just a number of discrete steps (detents) through the range.

Hide Dynamic

The SOLUTION shall allow for a 2D or 3D object to hide based on the status of a Digital Signal (or the Boolean result of a logical expression). The user shall be able to select whether it hides on the "True" or on the "False" condition. The user shall be able to select what values of an analog or digital data source will hide or show the object. The user should follow the guidelines of discrete and analog data sources listed above. The dynamic should allow for animated translucency.

Enabled/Disable Dynamic

The SOLUTION shall allow for a 2D or 3D object to become enabled or disabled based on the status of a Digital Signal (or the Boolean result of a logical expression). The user shall be able to select whether it becomes disabled on the "True" or on the "False" condition. If this Disable Dynamic becomes active for any type of user data-entry dynamic (like data entry fields, sliders, dials, check boxes, radio buttons, push buttons, etc.), then that dynamic will no longer accept input from the user while it remains disabled. The fill and line colors for objects that become disabled shall change to reflect that the object is now disabled. The default "Disabled colors" for all objects on the display shall be light gray for Fill, and dark gray for Line, but these shall be user configurable.

Digital Color Dynamic

The SOLUTION shall allow for the dynamic to change the color of a 2D or 3D object (either Line color, Fill color or both) based on the status of a digital signal. The user shall be able to select whether it changes color on the "True" or on the "False" condition. More than one Color Dynamic shall be permitted to be configured per object, each with its own independent Boolean expression to control the color dynamic. For example, one digital signal would change the object's fill color to red, while a different digital signal would change the fill color to yellow and the line color to black, while a third digital expression evaluates an analog signal to go above a threshold to change the fill color to green. If more than one digital color dynamic is assigned to the same object, the user shall be able to configure their order of precedence. Digital color dynamics should follow all rules of discrete dynamics listed above.

Analog Color Dynamic

The SOLUTION shall allow for the dynamic to change the color of a 2D or 3D object (either Line color, Fill color or both) based on the status of an analog signal. The range of the analog connection through which the object will change color shall be configurable by the user. The user shall be able to configure the Start color (corresponding to the low range) and the End Color (corresponding to the high range). As the Analog signal goes through this range, the object will change its color from the Start to End colors. The user shall be able to configure if the object shall revert to the default color assigned to the object if the analog signal falls below or goes above the specified range.

Button Action Dynamic

The SOLUTION should allow for a button action to be able to be applied to a 2D or 3D object. A button action should allow a shortcut key to be set to perform any command that is configured by the user. The "trigger" for the pick action command should be configurable between the different combinations of mouse buttons or mouse buttons with modifier keys pressed down (e.g., Shift + Click). The "trigger" should also allow mouse-over, execute while down, execute on release and execute on press events. The standard options for "tool tips" should apply (i.e., show description, data source information and error information). There should be an option to have a configurable pop-up dialog box for users to confirm data entry prior to changing a value.

The Solution should provide the ability to create an "Enabler" Button. This button must be held in place, via a continuous touch on a touchscreen or

mouse, or a selection (with a forced time out) via a mouse, before other button actions on the display will be enabled for execution. This feature forces 2 simultaneous touches to be required for an operation of equipment from a touchscreen.

The SOLUTION should allow for the Button Action Dynamic to configure the following:

Display Navigation - the user should be able to navigate forward and backward between recent displays via a button action dynamic "command."

Display Navigation - the user should be able to close the window via a button action dynamic "command."

Navigation -the user should be able to load a display via a button action dynamic "command."

Login/Logout Dialog - the user should be able to launch a security login/logout dialog for operators to login through;

Menu - the user should be able to launch a configurable menu on mouse event;

Popup Menu - the user should be able to add, edit and delete items from the menu during configuration or change colors;

Reports - the user shall be allowed to input information to enable reports on configured data;

Scripts - the user should be able to have a pick action start a script to perform some advanced or other feature;

Scripts - the user should be able to "bind" and "edit" scripts (covered in this document) using a convenient method such as a popup "browser" or "dockable" toolbar;

Global Aliases - the user should be able to configure one or more global aliases to be set. The global aliases should be created in a central application module accessible by the whole of the application and stored in a database;

Language - the user should be able to select a language to use once the mouse event has occurred. This list of languages should

be configured in a central application module accessible by the whole of the application;

Local Aliases - the user should be able to set a local alias and send the information to a contained window, the current display and dynamic objects within those displays.

Object Visibility - the user should be able to use a mouse event to set or change another objects visibility, including the object that holds the pick action.

View - the user should be able to use a mouse event to change the view of the current display.

Applications - the user should be able to launch a third-party application. Parameters include any EXE file name (or .BAT or .COM file) and optional command line strings for it. The user should also be able to specify a filename to launch.

Values - the user should be able to write a value to a variable or data source (given it is writable). The user should also be given the option to toggle between a predetermined set of values (configured by the user) during a live data mode.

Update Database - the user should be able to trigger a database update in a data mining control that is connected to a database.

Value Dynamic

The SOLUTION shall allow for a "value" Dynamic to be configured, with its "visible" properties matching those of Text objects described earlier. The data connection shall support (and be configurable as) reading 8-, 16-, 32-, and 64-bit integers and unsigned integers. The data connection shall support (and be configurable as) reading Boolean data. The user shall be able to specify whether leading zeros are shown (e.g., 23 vs. 0023). The user shall be able to specify whether thousands-separators are shown (e.g., 1000000.00 vs. 1,000,000.00). The thousands -separator, if shown, will conform to the local symbol (e.g., comma or period for US, Europe, etc.). The data connection shall support (and be configurable as) reading Single and Double-Precision Float data. For Floating Point and Double Precision data types, the user shall be able to configure how many decimal places are shown and if it appears in scientific notation. For all data types, the user shall be able to configure prefixes and postfixes. The data connection shall support (and be configurable as) reading String

data. For String data types, the user shall be able to configure the maximum number of characters to display.

The user shall be able to specify a text string to show for the units of the data being shown, and whether it is shown to the right or to the left of the number. The text string for units shall be able to be tied to a global alias or a language string (instead of just a constant text string). The data connection shall support (and be configurable as) reading Object (native data type) as the default.

The SOLUTION should offer a form of global communication in the form of commands that can be sent to other product modules from the graphics module.

The global commands for alarming should include, but are not limited to: Acknowledge Alarm(s); Load Configuration; and Set Filter.

The global commands for trending should include, but are not limited to: Set Time Range; Load Configuration; Set Time Period; Create Trend; and Delete Trend.

The global commands for graphics should include, but are not limited to: Open HMI Display; Write Value; Set Visibility; Navigate; and Print/Export.

Data Entry Dynamic

The SOLUTION shall allow for a Data Entry Dynamic to be configured, with its "visible" properties matching those of Text objects described earlier. The requirements for Value Dynamic shall apply to the Data Entry Dynamic as well. The user shall be able to override the High and Low ranges used to define the values allowed to be entered into the field. The user shall have the option to specify an Initial Value. The user shall have the option to pop-up a dialog box confirming the user's data entry prior to downloading the Value (i.e., and "are you sure you want to change Tag1 to 100" type of message with OK/Cancel buttons). The confirm dialogue box must include functionality for customizable text. The user shall have the option to pop-up a keypad for the data entry (thereby facilitating touch screen installations or other such pointing devices without requiring an actual keyboard). The pop-up keypad shall be a numeric pad for all number data types, and automatically switch to being a QWERTY keyboard for String data types. The pop-up keypad shall be user configurable for scalable sizing. During runtime, the user shall be able to increase/decrease the value by pressing the up/down arrow keys, respectively. The user shall be able to hit the ESC key, or simply clicking onto another object without first pressing the Enter key

as the confirmation of the entry. Doing so shall cause the data entry field to revert to the value it contained prior to the user typing in values. Confirm dialogue boxes must include functionality for customizable descriptive text.

Time and Date Dynamic

The SOLUTION shall allow for a Time/Date Dynamic to be configured, with its "visual" properties matching those of Text objects described earlier. It shall be configurable to show the current system date/time with the Current Date or Current Time Only. It shall be configurable to show the current system date/time with Current Date and Time (in that order). It shall be configurable to show the current system date/time with Current Time and Date (in that order).

The Time format shall be able to be configured in any combination of Hours, Minutes, Seconds (e.g., just the hour, or just the minutes, or just seconds, or hour and minute but not seconds, all three, etc.). The Time format shall be able to be configured in Military Time (24hour) or AM/PM format. The Time format shall be able to be configured in AM/PM indicator shown or not shown. The Date format shall be able to be configured in any combination of Day, Month, Year (e.g., just the month, just the day, just the year, all three, etc.), and in any order (e.g., month/day/year, or day/month/year, or year/month/day, etc.).

The Date format shall be able to be configured with a separator character between fields shall be user defined. The Date format shall be able to be configured so that the Day shall either be numeric, abbreviated string, or full name (e.g., 06, or Wed, or Wednesday). The Date format shall be able to be configured so that the Month shall either be numeric, abbreviated string, or full name (e.g., 12, or Dec, or December). The Date format shall be able to be configured so that the Year shall be configured to show either the last two numbers in the year, or the entire 4 numbers (e.g., 01, or 2001). The dynamic should also be able to display date for a Data Source of the type Date Time in addition to the current system time.

State Field Dynamic

The SOLUTION shall allow for a State Field Dynamic to be configured, with its "visual" properties matching those of Text objects described earlier. The text displayed shall be based on the value of the Analog Signal it is tied to. The User shall be able to build a table with the state definitions. For example, a value of 0 corresponds to the string "Auto", a value of 5 corresponds to the string "Manual Mode", a value of 15 corresponds to the string "Clean Cycle Beginning", and so on. A "default" string shall be able to be defined, which is shown in the event that the

analog signal has a value not defined in the string definition table. The maximum number of states defined per dynamic shall be 100 at a minimum. The dynamic should be able to define a different color for each state text. The dynamic should also be able to associate a text string with a range of values.

State Field Data Entry Dynamic

The SOLUTION shall allow for a State Field Dynamic (described earlier) to be turned into a data entry field. When clicking on such a field, the user shall be presented with a drop-down list of the states defined in the dynamic. Selection of an entry in the list shall be by mouse, up/down arrow keys or by typing in the first letter of an entry. Once selected, the corresponding numerical value shall be downloaded to the Analog Tag connected to the dynamic.

State Dynamic

The SOLUTION shall allow for the ability to have only one of the objects within the grouping to be shown at a time, with each object in the group having its visibility tied to the status of a Digital Signal (or the Boolean result of a logical expression).

For example, one digital signal would make just object one be shown, while a different digital signal would make the just second object shown, while perhaps a third digital expression evaluates an analog signal to go above a threshold to make just the third object shown. The user shall be able to select whether each object hides on the "True" or on the "False" condition of its digital connection. The user shall be able to configure the order of precedence (in case two conditions occur simultaneously, the higher priority object would be shown). The user shall have access to the properties of each of the objects in the dynamic so changes can be easily made. The state dynamic should animate between all the images or objects selected by the user. The state dynamic should deal with analog or digital data. Animation of the "states" can be linked to a data source in the normal manner for the application. A configurable order of images or objects can be edited. The objects shall become animated based on the status of a Digital Signal (or the Boolean result of a logical expression). The user shall be able to change the order in which the objects are shown (re-sequence the animation).

Range Dynamic

The SOLUTION shall allow for the range dynamic to be like a state dynamic, but work with one data source and switch images or objects

based on a percentage of the data range. The range dynamic should animate between all the images or objects selected by the user. Only one of the objects within the grouping shall be shown at a time, with each object in the group having its visibility tied to a portion of a range of an analog signal. The range dynamic should deal with digital or analog data. Animation of the "states" can be linked to a data source in the normal manner for the application. A configurable order of images or objects can be edited. The range Dynamic will cycle through all of the images selected at a user-specified "frame rate" (range at least 50ms to 2 minutes), thereby creating a picture slide-show animation. The objects shall become animated based on the status of a digital or analog signal (or the Boolean result of a logical expression).

The user shall be able to select whether the animation occurs on the "True" or on the "False" condition. The user shall be able to change the order in which the objects are shown (re-sequence the animation). The state Dynamic will cycle through all of the images selected at a user-specified "frame rate" (range at least 50ms to 2 minutes), thereby creating a picture slide-show animation.

Module Controls

The SOLUTION shall allow it to be possible to include embedded Trend Views (both Real-time and Historical) from the Trending Module of the HMI/SCADA package.

The SOLUTION shall allow users to include embedded Alarm Views (both Live and Historical) from the Alarming Module of the HMI/SCADA package.

It shall be possible to include more than one of Trend and/or Alarm can be included in any given display. The user shall be able to configure the placement (Top, Left) and size (Width, Height) of each Alarm/Trend/Data-Mining window. All of the features enjoyed by such windows with their respective Module shall also be available when embedded within the Graphics Module. Icons to insert Trend and/or Alarm views to any given display shall be through a control which is accessible through the main menu tool bar.

The SOLUTION should have a grid visualization control that can be embedded into an HMI screen for viewing of database data. The control should allow for writing, reading and deleting entries of data.

The SOLUTION should provide a Tile control with intelligent data connections. These Tiles should be able to provide navigation by user click and provide dynamic information within them for visualizing Key Performance Indicators.

The SOLUTION should provide a Menu control that can use gesture recognition to slide left and right on-screen. Swiping should switch pages of information configured by the user.

The SOLUTION should provide miscellaneous controls to allow for faster configuration. Controls can include radial, horizontal or vertical “scales” for gauges, a control to allow for easy configuration of piping diagrams and/or a simple control to make simple menus.

The SOLUTION should provide a control for viewing Video. This Video Control should provide at the very least Motion JPEG (MJPEG) viewing of streaming video. The video control should offer Pan, Tilt and Zoom (PTZ) controls within the module itself.

Symbol Storage

The SOLUTION shall allow for a graphics module to include a symbol storage capability that supports both static and dynamic symbols. The system shall be supplied with a standard set of graphic symbols in the library. Symbols in the library shall be able to be dragged into a graphic display for use. User created Symbols shall be able to be dragged into the library for storage. The library shall include the ability to create multiple categories in which to store the symbols. This will facilitate management of many symbols. A preview of the stored symbols shall be provided to allow the user to view a symbol before it is selected from the library. It shall be possible to make the preview thumbnail large (for seeing more detail), or small (for fitting more of them on a single page). The symbol library shall be able to “Float” above the Graphics workspace, thus enabling the drag-n-drop of symbols with minimal screen switching. The symbol library shall also be able to “dock” to any side of the graphics development environment as a toolbar. Symbols shall have the ability to be linked together (by some shared keyword or the like).

Objects sharing the same keyword shall be able to be updated on a global basis (whereby user selects the scope of the changes via a display list). This permits objects to have their look-n-feel visual properties, as well as their associated dynamics, updated throughout the project.

Aliasing for display properties should be allowed in “Smart” symbols for generic symbols to be saved and reused.

“Smart” symbols should be configurable with the user able to “expose” certain properties of an object or a group of objects as top level properties of the symbol itself.

"Smart" symbols should allow users to "bind" properties of objects within the symbol to the same property, data source or value. This bound property should give the user the ability to update multiple objects simultaneously using the top level property of the symbol.

"Smart" symbols should provide the ability to use expressions to determine the binding of a property. This will allow the properties to modify or use logic to determine what value to be used.

Expressions

The SOLUTION should allow for a built-in Expression editor, provided by the Graphics Package. Expressions shall be able to be used anywhere a Tag Name can be used. Direct access shall be provided to connect to OPC Tags. Numeric and String constants shall be able to be used in the expressions. Wherever a tag name can be used, an "Alias" shall also be permitted (Aliases are outlined elsewhere in this specification). The Editor shall parse the entered expression to check for syntax errors. "White space" (i.e., tabs, spaces, CR/LF) shall be ignored, permitting better formatting for readability of the expressions. A list of the expressions used in the current display shall be available (e.g., perhaps via a drop-down list) to aid in expression development and re-use.

The SOLUTION should allow for Functions provided by the Expression editor to include: Open and Closed Parenthesis (for defining order of operation); Addition, Subtraction, Multiplication, Division, Modulus Division; Relational expressions of greater than, less than, equal to, not equal to, less than or equal, greater than or equal; Logical AND, Logical OR, Logical NOT operations; and Bit-wise manipulation including AND, OR, NOT, XOR, Bit Shift Left, Bit Shift Right, and Bit Test.

The SOLUTION should allow for Mathematical functions provided by the Expression editor to include: Conditional Function - if(BooleanCondition, ValueIfTrue, ValueIfFalse) Condition; Arithmetic Functions - Sine, Cosine, Tangent, Arc-sine, Arc-cosine, Arc-tangent, Square Root, Raised to Power of, Logarithm, Natural Logarithm, Exponential, Absolute Value, Integer Ceiling, Integer Floor, Minimum and Maximum; String functions - Wildcard String Compare, String Length, Substring Extraction, Left Substring, Right Substring, String Concatenation, String Search, Trim Left and Right, Trim Left, Trim Right, Lower Case Cast, Upper Case Cast and Type Conversion; Constant Functions - Math Constant PI, Math Constant E, Hexadecimal Constant, Octal Constant, Binary Constant; Conditional If, Then, Else statement; and OPC Quality Testing (e.g., 192 - Quality GOOD) of values of OPC Tags.

Layers

The SOLUTION shall allow for the Graphics Package to offer Layering. The user shall be able to create an unlimited number of layers. The user shall be able to change the order of the layers. Layers above and/or below the currently active layer during configuration mode shall be able to be turned on/off to assist in display creation. A document explorer tool should provide easy access to this feature during configuration. The ability to duplicate an active layer during configuration shall be provided. The visibility of a layer during runtime shall be able to be tied to the status of a Digital Signal (or the Boolean result of a logical expression).

The solution should support graphic decluttering. The visibility of an object or a layer shall have the ability to be tied to the zoom factor used during runtime. For example, a layer which is only visible for zoom factors of 300% and greater would not be shown under normal view, but would rather only appear once the user has zoomed into a section of the display. Dynamics on a hidden layer shall have the choice of being configured to either remain on scan (so their values are cached and ready to be shown when the layer is once again shown), or taken off scan (to reduce I/O overhead, etc.). The visibility of a layer shall be easily tied to a Security Group or Level in the HMI software.

Scripting

The SOLUTION shall allow for a graphics module to create buttons and pick actions that, when activated (clicked-on by the user), will create an event that can run a script. The graphics module should utilize JScript scripting and provide intelligent prediction of user typing. The scripts will be "associated" with the object containing the pick action, such that if it is copied from one screen to another, or stored in the Symbol Library, the scripts will be stored along with the object. Scripting should be done inside the application and not require 3rd party software. Scripting should allow the use of "forms" that can be used to provide a better user experience. The "forms" should be easily configured through a design mode. A "toolbox" of standard items for forms shall be provided in the scripting interface. The scripting interface should be customizable in size and location within the graphics application for user preferences. An "auto complete" functionality shall be provided to aid in scripting. Displays with scripting should be able to be run in a Web environment.

Other Features

The SOLUTION shall allow for a display that supports the ability to enter the Runtime Environment from the Menu bar of the configurator. During a runtime session, Security Login can be required with Security Groups and Levels to determine how much accessibility operators have during Runtime.

During a runtime session, the user shall have the ability to select a display file from the Menu bar, based on Security Group or Level. The display Menu bar shall be user configurable to be hidden based on Security Group or level during a runtime session.

The SOLUTION shall allow that, during a runtime session, the graphical interface shall maintain a history of the last 50 display files (similar to Internet Explorer's back/forward feature), allowing the user to navigate through the display file history via "Display Forward" or "Display Backward" commands.

The display file history feature (forward/back commands) shall remember the initial alias settings specified when a display was opened (see description of the aliasing feature), and will reset those aliases when you go back to that display in the file history.

The SOLUTION shall allow that, during runtime, the user shall be able to move among the various data entry dynamics (sliders, buttons, dials, data entry fields, state field entries, pick actions) by either pointing to it with the mouse (or touch screen, etc.), or by tabbing to the field to enter data.

The SOLUTION shall allow that, during runtime, the graphics package should allow for "caching" to enable faster load times after the preliminary load of a display. The "caching" should be configurable (change to ON or OFF state) through both a "configuration" mode and a "runtime" mode.

The SOLUTION shall allow that the order in which the objects are tabbed to shall be configurable by the user. The mouse cursor shall change whenever the user hovers over an object with a dynamic for user input. A keyboard focus highlight will be shown.

The SOLUTION shall allow that the graphics editor shall have capability to create/edit templates and apply these on various displays. These templates can be saved without any related script code.

The SOLUTION shall allow that the graphics module shall offer a Print Preview option to show what the printout will look like prior to sending it on to the printer. The graphics module shall offer a Print feature to specify on a per display basis if it is to be printed with a white background in order to save ink. The graphics module shall offer several print regions.

The SOLUTION shall allow that the graphics editor shall maintain a list of at least the last 5 displays that were edited (making it easier to switch between them).

GEO Spatial Mapping Module

The SOLUTION should include a mapping module or integrated mapping within the graphics module. The mapping module should be able to be placed in a display by use of a layer and should be based on a tile system that can utilize multiple inputs.

The SOLUTION should be capable of using multiple tile systems provided by third parties (e.g., Microsoft Bing, Google, and Esri). These should be configurable in the live preview mode and operators should be given the option to change the tile system dynamically.

The SOLUTION should allow for multiple layers of mapping tiles. Operators must be able to show and hide layers dynamically. Layers should be independent and able to accommodate any mapping provider and map type. Layers should also have transparency options to partially show a layer.

The SOLUTION should be able to cache the mapping “tiles” for quicker loading of recently loaded tiles. The mapping module should allow demo configuration on a demo map and should allow the import of GPS Exchange Format (GPX). Quick configuration should be available through “pins” or symbols that can be dropped to specific locations. These “pins” should permit dynamics within configuration.

The SOLUTION should provide a method to have animation of objects based on Latitude and Longitudinal coordinates.

The SOLUTION’s mapping module should allow the user to “zoom to” a latitude and longitude by typing, as well as choose the type of map to view in configuration and runtime modes. The types should include Satellite; Road; Road (Topology shading); and Hybrid.

The SOLUTION’s mapping module’s Runtime manipulation should support the following options via the mouse or for multi-touch on support displays: Pan; Drag; Zoom; and Zoom To.

ASSET MANAGEMENT MODULE

General

The SOLUTION should provide an asset management module where “assets” can be configured for any real-world object, equipment or location as part of a hierarchical navigation tree. The hierarchical structure shall support enterprise organization as defined by the ISA 95 standard.

The Asset Management Module should load dynamically to afford an unlimited number of assets. The loading of "child assets" should not exceed a few seconds (number dependent).

The Asset Management Module should provide an organization of "levels" to be customized for each application. There should be a default configuration, but the list should be easily customizable to indicate what the levels represent. An icon can be used to signify each level.

The Asset Management Module should be designed with reusable classes or templates that can be configured and then "added" to the asset navigational tree. These templates should be customizable to represent specific equipment or locations. These templates shall support the use of "aliased" variables, which are resolved at the time of each unique instantiation of a template to create an asset. The aliased value shall assure that the user only need to enter those parts of tag definitions, or expressions, or text that are unique for the specific asset.

The Asset Management Module should support the definition of unlimited "properties" associated with each asset. Properties of an asset can include real-time tags, expressions, constants, historical values or aggregations.

Communication

The SOLUTION should utilize a global communication mechanism as a primary function of sending "commands" for execution in other product modules. This communication should serve to allow real-time changes to be made to the visualization by operators.

The SOLUTION should offer a form of global communication in the form of commands that can be sent to other product modules from the asset management module.

The global commands for alarming should include, but are not limited to: Acknowledge Alarm(s); Load Configuration; and Set Filter.

The global commands for trending should include, but are not limited to: Set Time Range; Load Configuration; Set Time Period; Create Trend; and Delete Trend.

The global commands for graphics should include, but are not limited to: Open HMI Display; Write Value; Set Visibility; Navigate; and Print/Export.

TRENDING AND DATA LOGGING SOFTWARE MODULE

General

The SOLUTION should be based on OPC specifications by the OPC Foundation, being compliant not only with the OPC-Data Access (DA) spec as a client, but also OPC-Historical Data Access (HDA) as a server (as of this writing, the latest HDA specification is Version 1.2). It should also support Windows Presentation Foundation (WPF) communication and be documented and accessible through an extensive API.

The SOLUTION should be of modular design, with separate components for: Logging Configuration; Data Logging to a Database; Data Trending/Charting; and Historical Reporting. It should support Silverlight communication including Silverlight HDA and Silverlight viewers for the HMI/SCADA package. It should also support exporting of trend statistics and export of runtime data.

Historical Data Logging

The SOLUTION should provide an OPC-HDA Compliant historical data logging tool, which should log OPC Data and/or "Expressions" (Expressions are outlined earlier in this specification) as well as collect the data at a user-defined frequency, ranging from 10 milliseconds to once every 24 hours.

The SOLUTION's Historical Data Logging should first cache a user-defined amount of such collected data into RAM before writing the information to disk. This "block writing" to disk will permit disk-write optimizations, data compression, filtering, etc. It should also define when the block writing to disk occurs, choosing from: Time Interval (range at least 10ms to 24 hours); Specific Time (e.g., every day at 8:30am); Number of samples collected (range 1 to 10,000); or Condition/Event tied to an OPC Tag and/or an Expression (e.g., flush the buffer when a new part is made).

The SOLUTION's Historical Data Logging should apply optional data compression filters on the data as it is written to disk as follows: The Maximum value; The Minimum value; The Average value; The Standard Deviation; A Running Totalizer; The Running Maximum value (the maximum value over the entire logging period); The Running Minimum value (the minimum value over the entire logging period); The Running Average value (using an exponentially weighted moving average filter, giving more "weight" to newer samples); The Moving Maximum value (the maximum value over a user-defined "sliding window" of time); The Moving Minimum value (the minimum value over a user-defined "sliding window" of time); or The Moving Average value (the average value over a user-defined "sliding window" of time).

The SOLUTION's Historical Data Logging should use either the user-defined logging interval or a user-defined calculation period (range 10ms to 24 hours) for the above mentioned data compression filters. It should offer the choice of logging data only when it changes by a user-defined percentage (e.g., 1.3% of the value), or by a user-defined absolute threshold (e.g., 4 PSI). It should also provide support for user configurable logging intervals for data logging at fixed intervals on a per group basis, as well as offer the user a choice of OPC DA refresh rates.

The SOLUTION's Historical Data Logging should start the logging process (collecting to RAM, applying filters, writing to disk, etc.) based on: Time Interval (range at least 10ms to 24 hours); Specific Time (e.g., every day at 8:30am); Condition/Event tied to an OPC Tag and/or an Expression (e.g., begin data monitoring/logging whenever the machine is turned on.); or as soon as the Trending Software Module starts up.

The SOLUTION's Historical Data Logging should stop the logging process (collecting to RAM, applying filters, writing to disk, etc.) based on different set of conditions, choosing from: Time Interval (range at least 10ms to 24 hours); Specific Time (e.g., every day at 8:30am); or Condition/Event tied to an OPC Tag and/or an Expression (e.g., begin data monitoring/logging whenever the machine is turned on.).

The SOLUTION's Historical Data Logging should log the data to the following open databases: Microsoft SQL Server 2005 SP4; SQL Server 2008 SP2; SQL Server 2008 R2 SP1; Microsoft SQL 2012 including all Express Editions. It should also have the ability to log data to more than one of the above-mentioned databases simultaneously.

The SOLUTION's Historical Data Logging should store the database on either the local machine, or on a remote file server, or other such networked computer. It should define more than one group of data, at minimum permitting a dozen such data groups, each with its own logging start/stop times, calculation periods, filters, frequencies, and so on. The storage of the configuration data itself shall also be stored in an open database (e.g., SQL). The timestamps used for the logged data shall have the option of using Universal Time Code (UTC), as well as have an option for correcting for Daylight Savings Time.

The SOLUTION's Historical Data Logger should support "Store and Forward" capabilities, which ensure data logging integrity even when the database server fails or communications fail. The configuration capabilities should be separated sufficiently from the Logger allowing the Historical Data Logger configuration to be changed without disturbing the Historical Data Access (HDA) Replay. It should also support a "Monitor View" which will report key runtime information.

Data Trending/Charting

The SOLUTION should include a Data Trending/Charting component (viewer) that is capable of being embedded within a graphic display file and configured through that graphic display to view data. The viewer should be able to contain multiple configurable charts or trends. The trends within the viewer should be configurable as to the: Data Source; Colors and line thicknesses; Plot type; and Scale.

The charts within the viewer should be configurable as to the: Data shown and order of the data; Legend; Time period; Easy access to change the Time Period; and Data collection and refresh rates.

Data Trending/Charting Configuration

The SOLUTION should include a Data Trending/Charting component that complies with the following configuration requirements. The trend module should be able to connect to data sources through traditional OPC or OPC UA communication. Multiple pens shall be able to be defined per window (over 100 should be allowed if necessary). Each pen shall have the following parameters be able to be configured independently: Color (supporting at least 256 colors); Solid Line Thickness; Name (either as a constant string or tied to a global alias); Auto-scale setting; High and Low Range (either as a constant value or tied to an OPC Tag or Expression, or to a global alias); Alarm-limit lines for High, Low, High-High, and Low-Low (either as a constant value or tied to an OPC Tag or Expression, or to a global alias); Whether the alarm limit lines should be plotted along with the pen line; and Connection of points method (straight line between samples, or step/stair plot connection).

The SOLUTION's Data Trending/Charting components should include Pen connections that can tie to: Live OPC Tags; Live Expressions; Data collected in a RAM buffer (user-defined timeline) for instant plotting; and Historical Data from the databases logged to by the Data Logger Module. Time saving features like automatically adding descriptions based on the Pen names should be included.

The SOLUTION's Data Trending/Charting components should include both Live and Historical data that can be plotted in the same trend window at the same time. Historical pens shall be able to be turned into an "Ideal Trend Pen", allowing the historical data from log files to be anchored on the screen, overlapped by current real-time data.

The SOLUTION's Data Trending/Charting components should include a trend viewer that is configurable to allow unlimited stacked plots within the same viewer. The trend window shall support multiple ranges displayed at the same time. The number of samples plotted shall be in the range of up to 10,000. It

should offer a user-defined plotting interval/rate (range from milliseconds to hours per data point).

The SOLUTION's Data Trending/Charting components should allow for the creation of trend display windows in the following formats: Time (Horizontal) - Data values are plotted horizontally against time. Time is on the X-axis and data range is on the Y-axis. The data scale shall be configured to be either linear or logarithmic; Time (Vertical) - Strip Chart Recording. Data values are plotted vertically against time. Time is on the Y-axis and data range is on the X-axis; Area Charts - Data values are plotted horizontally with the area below a line filled with a specified color; Circular Charts - Data values are plotted around a circular grid over time; X vs. Y - Data values are trended against a specified data pen, rather than time. The data range is on the Y-axis, with one of the pens having its values defining the X axis; Bar Plots - Each data value gets a bar distributed along the X-axis, with its value giving the bar height on the Y-axis; Pie Charts -Data is plotted as a percentage of a whole.

The SOLUTION's Data Trending/Charting components should allow for options shall to show in an area adjacent to the plot: 3D plot versions of the plots listed in the paragraph above; Ranges; Time stamp for each pen used (Note: historical pens will have differing times!); Pen data values; Pen data engineering units; Pen signal name (showing OPC name or expression used); Pen description (user-defined name); OPC Quality; User configurable major and minor tick marks on both X and Y axis; and Configurable chart titles and labels on the X and Y axis.

The SOLUTION's Data Trending/Charting components should allow for the user to specify a title for the trend view, defining either a constant string or tying it to a global alias, a language alias or OPC tag. The area used for the plot shall permit configuration of: Background Color and/or image; Background images types supported shall include: JPG (.jpg, .jpe, .jpeg), PNG (.png), GIF (.gif), TIFF (.tif, .tiff, .jfx), BMP (.bmp), Icon (.ico), Metafiles (.wmf, .emf); Printing of the active background image shall be configurable; Grid (grids with properties of: visibility, color, thickness and line style); and Pen Markers.

The SOLUTION's Data Trending/Charting components should allow for the order in which the pens appear in the list to be changeable. The user shall be able to define if each pen is plotted in the same range window, or if multiple "stacked" plots (with pens being plotted in sub-region windows) are desired. If stacked plots are used, then the assignment of each pen to a given sub-region shall be user selectable. The viewer shall be configurable to allow customizable background printing (i.e. a selection to include or exclude the background in printing).

Data Trending/Charting Runtime

The SOLUTION's Data Trending/Charting component shall comply with the following requirements while it is in "runtime" mode plotting the data. The plots should support multi-touch and conform to multi-touch standards such as zooming and panning. The plot shall be able to be "frozen" with the current data snapshot on the screen. During this time, the data is still being collected for plotting in the background, ready to be shown once "un-frozen". While frozen, the user shall be able to drag the mouse across the plot, with the data in the "detail table" section (if shown) updating to reflect the data under the mouse cursor. While frozen, if the user hovers over a particular data point, a pop-up window shall show the details of that data point. While frozen, the user shall be able to scroll forward/backward in time, seeing data that may have been scrolled out of the view window. While frozen, the user shall be able to zoom in (with the data re-scaled automatically to fill the defined trend area) by the mouse dragging a zoom box around a window of data.

The SOLUTION's Data Trending/Charting component should allow users to specify the initialization state in regards to polling data immediately on entering a "live" or "runtime" state and to start in a "freeze" mode where the most recent data is collected and then the trend is "paused" for viewing.

The SOLUTION's Data Trending/Charting component should allow for the trend to support on-line configuration of the following: Pen parameters including: color, width, style, range, engineering units, description, number formatting, and alarm line placement; Turning the Visibility of a pen on/off (if invisible, it is still collecting the data, just not showing it on the screen during that time, aiding in "de-cluttering" of the plot); and Number of grids used for the X-axis and Y-axis. The SOLUTION's Data Trending/Charting component should allow for the trend window to have the capability of adding new pens on the fly, while it is plotting existing data, by the following methods: a button click; and drag-n-drop a data point from the Graphic screen directly into the plot area.

The SOLUTION's Data Trending/Charting component shall support the realignment of historical data pens in a window (forward and/or backward in time specifying years, months, days, hours, minutes, seconds for the shift). Time shifting shall be accomplished by: Time entry into a pop-up dialog window; and dragging the pen with the mouse. The SOLUTION's Data Trending/Charting component should permit Multiple zoom levels (i.e., zoom in further into an already zoomed plot).

The SOLUTION's Data Trending/Charting component shall allow the user, when viewing a Historical Pen, to attach notes to a specific data value collected. Such notes shall be stored in the database along with the data value itself. A mechanism for viewing any existing notes already stored with the data values in the database shall be provided.

The SOLUTION's Data Trending/Charting component shall provide a single dockable toolbar for access to the following Runtime actions: Edit Pens; Edit Trend; Edit Period; Freeze Trend; Statistics; Zoom; Paging Forward and Back; Set Time; Show Comments; Print Trend; Save Trend; and Dock Toolbar. The user shall be able to obtain statistics of the data shown in the window (either the live view, or a frozen zoomed sub-window). Included for each pen shall be: Minimum Value; Maximum Value; Average Value; Standard Deviation; and Number of Samples. The statistics shall be able to be saved to a text file.

The SOLUTION should offer a configurable toolbar that provides a set of one touch buttons for easily switching to different trend periods. The solution shall include a pre-defined subset of period buttons such as one hour and one day. Any number of buttons shall be configurable to include shorter or longer time period.

ALARM MONITORING MODULE

General

THE SOLUTION shall provide an alarm module based on OPC specifications by the OPC Foundation, being compliant not only with the OPC-Data Access spec as a client, but also OPC-Alarms & Events being both a client and a server. The alarm module should comply with OPC A/E and OPC UA A/E, as well as be documented and accessible through an extensive API.

The SOLUTION should be able to integrate BACnet events into the alarm log. The ability to view and modify BACnet alarms should also be available.

The SOLUTION's alarm module should include a method to view real-time or OPC DA data from the server. This real-time data should be usable within the graphics module and any expressions to determine dynamic behavior.

THE SOLUTION's alarm module shall be of modular design, with separate components for: Alarm Configuration; Alarm Server; Alarm Logging to a Database and Printer; Live "Active" Alarm Viewing with Charting; and Historical Alarm Reporting. The Alarm module should be connected to the security module of the HMI/SCADA software package. The Alarm module should reset to a default configuration when users change to mitigate access problems for different users. The Alarm module should support Silverlight communication.

Alarm Server

The SOLUTION's Alarm Server Module shall be able to be configured to monitor OPC, OPC UA Data, BACnet and/or "Expressions" (Expressions are outlined

earlier in this specification) for alarm conditions, as well as contain help instructions on a per-alarm basis to assist operators in solving the alarms when viewing them. It should also associate a default graphic display with each alarm. That way, when alarms are viewed in runtime, an operator could quickly jump to the associated display to view details on the alarm.

The SOLUTION's Alarm Server Module shall enable monitoring of each individual alarm based on an OPC Data digital tag or the result of an expression (to aid in eliminating nuisance alarms, etc.) If the result is "true", then the alarm is monitored. If "false", then the alarm is ignored and won't even be generated. It should also offer grouping the alarms so configured into categories and sub-categories (also known as Alarm Areas). It should also have a delay time specified, on a per-alarm basis, that an alarm must be active before it is considered a true alarm. This delay can be defined as either a constant (e.g., 10 seconds), or tied to an OPC Tag or Expression.

The SOLUTION's Alarm Server Module shall be able to associate one or more (up to 10) OPC Tags/Expressions with each alarm as "supplemental" information to the alarm. For example, one could tie a "Batch ID" to a specific alarm. Then, when the alarm becomes active, a permanent record of the Batch ID is also stored along with the alarm information. It should also run locally or over the network, supplying alarms and events to multiple alarm clients on multiple machines.

The SOLUTION's Alarm Server Module shall enable storage of the alarm configuration data itself in an open database (e.g., Microsoft SQL Express, etc. and trouble shooting.

The SOLUTION's Alarm Server Module shall be able to send individual alarm acknowledgement bits to a PLC via the OPC server, as well as provide the following alarm types: Digital; Analog Limit; Deviation; Rate of Change; Alarm Toggle Count; and Time in Alarm.

Alarm Server – Digital Alarms

The SOLUTION's alarm server shall generate an alarm when a digital signal goes into alarm. The user shall be able to configure: Whether the Alarm State is 0 or 1, being either a constant value or tied to an OPC Tag and/or expression; The Priority of the alarm, being either a constant value or tied to an OPC Tag and/or expression; The message to display when the alarm occurs. This message can be either entered directly as a string constant, or tied to a global alias; the message to display when the alarm returns to normal; and whether the alarm requires acknowledgment from the operator.

Alarm Server – Analog Limit Alarms

The SOLUTION's alarm server shall monitor analog signals, generating an alarm when it passes a limit threshold. The user shall be able to configure: Individual alarm thresholds for four different alarm states: Low-Low, Low, High, High-High; The alarm thresholds, being either a constant value or tied to an OPC Tag and/or expression; A deadband through which the alarm must change before a change in state occurs, being either a constant value or tied to an OPC Tag and/or expression; The Priority of each alarm state, being either a constant value or tied to an OPC Tag and/or expression; Separate messages to display when each of the alarm states occur. These messages can be either entered directly as string constants, or tied to a global alias. The message to display when the alarm returns to normal. This message can be either entered directly as a string constant, or tied to a global alias; and whether each of the alarm states requires acknowledgment from the operator.

Alarm Server – Deviation Alarms

The SOLUTION's alarm server shall be able to monitor two analog signals, comparing them and generating an alarm when they deviate from one another. The user shall be able to configure the same parameters for Deviation alarms as those available for Limit Alarms.

Alarm Server – Rate of Change Alarms

The SOLUTION's alarm server shall monitor analog signals, generating an alarm when it changes faster than a defined rate. The user shall be able to configure: The Rate of Change Limit, being either a constant value or tied to an OPC Tag and/or expression; The Priority of the alarm, being either a constant value or tied to an OPC Tag and/or expression; the message to display when the alarm occurs. This message can be either entered directly as a string constant, or tied to a global alias; the message to display when the alarm returns to normal. This message can be either entered directly as a string constant, or tied to a language alias; and whether the alarm requires acknowledgment from the operator.

Alarm Server – Alarm Toggle Count

The SOLUTION's alarm server shall be able to count the times a point goes into and out of an alarm state.

Alarm Server – Time in Alarm

The SOLUTION's alarm server shall provide monitoring and display capabilities for the amount of time each alarm has been active.

Multimedia Alarm Notification

The SOLUTION shall have an OPC-compliant, distributed, enterprise-wide alarm notification system that delivers real-time alarm information via e-mail, pager, fax, voice, text-to-speech, phone and software and Ethernet-hardware marquees. The multimedia alarming shall be integrated like any other module.

The system shall include the following agents and methods for delivering remote alarm notification: Web-Enabled Alarming (Multimedia alarm notification system shall support Instant viewing and acknowledgment over the Internet and intranets); Voice over IP Support; Telephone Voice Call-In and Call-Out (Automatic notification or call-in using a standard or mobile phone.); E-mail Alerts (Send alarm messages and acknowledgements over standard business communication tools.); Personnel Scheduler and Video Alarms (Create schedules for escalating notification with live video); Scrolling Marquees & Pop-up Windows (Hardware and software scrolling marquees and attention- grabbing pop-up windows alert you wherever you may be); Wireless 2-Way Paging for GSM & SMS (Receive and acknowledge messages on a wide range of wireless devices including pagers, Blackberry devices and mobile phones); Support for SNMP (native Simple Network Management Protocol) and BACnet (the popular building control communication standard); Text to Speech; and Task Tray (Allow Task Tray notification balloons from the Windows task bar).

Alarm Logger

The SOLUTION's Alarm Logger Module shall be able to be configured to: Print alarms/events to a printer (either local or on a network); and Print alarms/events to an optional backup printer (either local or on a network).

The SOLUTION's Alarm Logger Module should be able to log alarms to the following types of open databases: Microsoft SQL Server 2008 R2; Microsoft SQL Server 2008; Microsoft SQL Server 2005; and Microsoft SQL Express Editions of the same Versions.

The SOLUTION's Alarm Logger Module should support redundant logging configurations with node pair database synchronization as well as "Store and Forward" capability, which ensures data logging integrity even when the database server fails or communications fail. It should log the alarms/events to a user-defined table name within the specified database.

The SOLUTION's Alarm Logger Module should allow for the selection of which fields/parameters of the alarms get logged/printed, selecting from: Date/Time; Tag Name; Priority; Alarm Area; Alarm Message; Alarm Type/Category; OPC

Quality; Node the Alarm was generated on; Acknowledgment Status; and An Operator-entered Acknowledgment comment.

The SOLUTION's Alarm Logger Module should allow for logging the Time Stamp to a precision of 1 millisecond (time stamp as supplied by the OPC Data). It should also filter which alarms are logged and/or printed. This alarm filtering shall be permitted on a per-alarm basis.

The SOLUTION's Alarm Logger shall provide the capability to log up to 20 OPC Tags or Expressions with each alarm as "supplemental" information to the alarm. For example, one could tie a "Batch ID" to a specific alarm. Then, when the alarm becomes active, a permanent record of the Batch ID is also stored along with the alarm info.

Alarm Live/Active Alarm Viewing Configuration

The SOLUTION's Alarm Viewer component shall comply with the following configuration requirements. View window dimensions (height, width, top, left) shall be defined by simply clicking/dragging with the mouse. The background color of the alarm window shall be defined from at least 256 colors. The lines (grids) separating the rows and/or columns shall be defined as to their color, width, line style, and visibility (shown or not). The number of rows visible shall be user defined (range at least 2 to 50). By default, each row shall contain one alarm message. A "multi-line" row shall be permitted, with the user specifying the number of lines (height) of each row (range at least 1-9 lines per alarm row).

The SOLUTION's Alarm Viewer component shall allow for Different Font Styles (font name, size, bold, italic, underline, color), which shall be defined for the following 5 categories: Active alarms not yet acknowledged; Active alarms which are acknowledged; Alarms which have returned to their normal state, but which have not yet been acknowledged; Simple "System" Events; and Tracking Events for following operator actions. Within each of the above five categories, additional font styles shall be able to be defined for different priorities. For example, unacknowledged alarms are white text on red background, Arial size 12; in addition, any of those which are also over 500 in priority should also be Bold, Underline, and overriding the size to 14).

The SOLUTION's Alarm Viewer component shall allow for Columns visible in the alarm window to be configurable to include any number of the following, or user defined names: Date/Time; Tag Name; Priority; Alarm Area; Alarm Message; Alarm Type/Category; OPC Quality; Node the Alarm was generated on; Acknowledgment Status; An Operator-entered Acknowledgment comment; Help Information; Time the Alarm has been Active; Additional Associated Values (e.g., other live data values) related to the alarm; or Alarm Server Name.

The SOLUTION's Alarm Viewer component shall allow for Column Text Alignment to be user-defined as Center, Left, or Right, on a per-column basis. The Alarm View shall be sorted by a user-selected column, either in ascending or descending order. Additional sub-sorting by another column shall be permitted. For example, first sort by alarm priority (highest at the top), then by area (alphabetical order), then by Tag name.

A column shall be able to be designated as the source for pop-up tooltips during runtime, and that column therefore may be made invisible. For example, the Help column could be used so operators get context sensitive pop-up assistance while hovering over an alarm line.

The SOLUTION's Alarm Viewer component shall be able to be configured to get its alarms from either a local alarm server, or one on the network. The Alarm Viewer shall be able subscribe to more than one such alarm server. Filters shall be able to be defined to select which types of alarms should be shown in a particular alarm view. For example, show only alarms with a priority greater than 100, or show only unacknowledged alarms, or only operator tracking events, etc. Alarm Filtering shall be configurable based on: Events vs. Alarms; Acknowledgment Status; Priority; Tag Name; Alarm Area; Alarm Type (limit, deviation, rate of change, digital); Alarm Sub-Type (High-High, High, Low, Low-Low); OPC Quality; Date/Time; and Node. Alarm Filtering shall also have the capacity to use aliasing for dynamic changes to the filter in runtime through button actions.

The SOLUTION's Alarm Viewer component shall allow for Multiple Alarm Filters to be configured for selection by the operator during runtime operation. Charting should also be configurable within the configuration of the Alarm Viewer. Alarm charts should consist of the same alarm subscription configuration to mimic the grid. Different chart types should be available including: Bar Plots (2D and 3D should be permitted); Pie Plots (2D and 3D should be permitted); Cone Plots; and Cylinder Plots.

Alarm Live/Active Alarm Viewing Runtime

The SOLUTION's Alarm Viewer shall comply with the following requirements while it is in "runtime" mode showing live/active alarms: If more alarms exist than there are rows visible, then a scroll bar shall be provided such that operators can scroll through the listing. If the operator scrolls down to another part of the listing and a new alarm comes into the system, a visual indicator shall be given (so the operator knows the view has changed beyond what he's looking at). The SOLUTION will provide messages for users for specific cases such as an attempt to acknowledge an alarm that has been previously acknowledged.

The user shall be able to select just one alarm in the list, or use a modifier key (e.g., shift) to select multiple alarm lines at the same time. The alarm selection will not affect the color of the alarm line, so that operators can still use color as an alarm indicator. Alarms acknowledged at any one client shall be reflected on all other subscribed alarm clients on the network.

When an alarm is acknowledged, the operator shall have the opportunity to enter in a comment to be stored in the log database. The capability shall be available to provide a pre-configured list of comments to select from. Operators shall have access to viewing the "Operator Comments", and append additional comments to the database.

The SOLUTION shall allow acknowledgment of alarms by: individual alarm, as selected by user; group of alarms, selected by the user; all alarms belonging to the same area; all alarms globally; all filtered alarms; all alarms currently visible in the alarm view window; all alarms in the system; and by alarm severity level.

The SOLUTION shall allow for Rows and/or Columns shall to be resized, if such option was configured to be available to the operator. It should also allow support for auto-sizing of columns, as well as the ability to change by which column is sorted during runtime. Also, more than one Alarm Viewer shall be permitted, each with its own set of configuration parameters (e.g., subscriptions, filters, styles, etc.). Individual alarms, configurable on a per alarm basis, shall be capable of acknowledgement on return to normal condition.

Historical Alarm Reporting

The SOLUTION's Historical Alarm Report component shall be capable of being embedded within a graphic display file. Alarms shall be gathered from the Alarm Log Database, as created by the Alarm Logger component. This shall be from either the local machine, a networked PC, or via a URL Web site. These historical alarms shall be presented in one of four main report styles: Tabular view; Bar Chart view; Pie Chart view; and Report view.

The SOLUTION's Historical Alarm Report component shall allow for the Font Style (font name, size, bold, italic, underline, color) used in the reports and the on-screen displays to be user-defined. Which fields (columns) are shown in the table shall be selected from those logged to the alarm database and the columns shall allow for user configuration, including renaming. The user shall be able to sort the information based on any of the fields in the report, choosing from both Ascending and Descending sorts. An option for "auto-refreshing" the information from the database shall be provided, with the user specifying the frequency of the report updates. The appearance of the Bar/Pie charts shall be configurable as to the size and plot area.

The SOLUTION's Historical Alarm Report component's Bar and Pie Charts shall include sections plotted as summaries of the number of alarms per either: Node; Alarm Area; Tag (Data Source); Priority (Severity); Date/Time; and Alarm Type. The SOLUTION's Historical Alarm Report component shall allow the ability to print the report during runtime with a simple push of a button.

GRID VISUALIZATION MODULE

General

THE SOLUTION shall provide a module for creating grid or "spreadsheet" type visualization of large amounts of data. The data should be of configurable update rate (or on request polling), but provide real-time data. Date source support for the module should include SQL Server Databases, OLEDB, ODBC, SAP, OPC DA and OPC UA DA data.

Grid Visualization Configuration

The SOLUTION's grid visualization module should be configured in a central interface in line with all modules. The module should include a method to allow reuse of the data configuration for multiple HMI displays in grid visualization viewing.

The SOLUTIONS's grid visualization data configuration should include creating data sets for use in the "control."

Grid Visualization Runtime

The SOLUTION should provide standard features for the "control" similar to the alarm control's grid layout. Items like color, font, size, and flashing should be available for text. Sorting, grouping and filtering should be provided for rows and columns.

The SOLUTION should provide a "control" or method of viewing for the grid visualization module within the graphics module. Any number of Grid views can be added to any graphic display. A convenient, easy to use dialog should allow users to customize the grid components and aesthetics.

The SOLUTION's grid visualization module should allow users to read and write to the data sources. During "runtime" or "live view" mode the grid visualization tool should allow live sorting of columns of data in Microsoft Office Excel type functionality. The module should also allow for filtering and grouping of data for customized viewing.

The SOLUTION should collect data requests when possible to reduce the load on the data sources utilizing aggregation or caching algorithms.

FAULT DETECTION AND DIAGNOSTICS MODULE

General

The SOLUTION shall provide a rule execution engine, which can monitor all available monitored data and alarms, and execute user defined logic to determine or predict equipment operational or efficiency faults.

The Fault Detection and Diagnostic Module should allow, but is not limited to integration of data from alarms, real-time data values, historical trends/data and manual cause/symptom entries.

Fault Detection and Diagnostic Engine

The SOLUTION should provide default conditional logic for fault detection. The user shall be able to modify the logic and inputs of any existing fault definition and shall be able to add any number of new faults, including all fault analysis logic, to any asset.

In addition to fault detection logic, the user shall be able to define a cause table that shall define all possible causes for a specific fault, as well as all inputs that can be indicative of a specific cause. The user shall be able to define the probabilities of specific causes, as influenced by the current state of the related inputs.

Fault Detection and Diagnostic Visualization Runtime

The SOLUTION shall continuously monitor all referenced input values, input states, historical derivations, and determine if a fault condition has occurred. On occurrence of a Fault condition, the system shall display, and issue alert notifications. On occurrence of a fault the system shall analyze all referenced inputs continuously and calculate all possible causes and calculate the probability of all possible causes.

THE SOLUTION shall provide a standard "control" for viewing current faults, symptoms and possible causes, sorted by probability to assist with the resolution of faults that have occurred.

SCHEDULING MODULE

THE SOLUTION shall provide a module for scheduling events that control data values or sets of data values. These schedules should be configurable centrally and allow for reuse.

The SOLUTION should provide the ability to view and modify BACnet Schedules as part of the native scheduling module.

The scheduling module should include the ability to create schedules with full or partial day events to drive an asset list of data values. The SOLUTION should be able to connect to OPC DA and OPC UA DA data while aggregating data into sets.

Scheduling shall allow the definition of an unlimited number of schedule definitions. Each schedule definition should allow for "normal" or "default" conditions as well as support for "exceptions" on specified days, seasonal changes, holidays and manual overrides. Each schedule shall allow for an unlimited number of scheduled actions in a specific 24 hour time frame, with a schedule resolution of xx seconds. The SOLUTION should manage these events within its central interface for reuse with multiple schedules automatically.

SECURITY

The SOLUTION should have mandatory driver signing to protect from 3rd party software interfering or replacing components. It should also be compatible with an operating system with protection from potentially malicious attacks via the Windows kernel (such as Kernel Patch Protection). In addition, it should be compatible with operating systems with protection from attacks originating from restricted or non-executable memory locations (software and hardware based Data Execution Protection - DEP).

The SOLUTION should provide a configurable security component that can be used to restrict access, application navigation and configuration of databases or displays. It should also support configuration of different sets (or policies) of individual users. It should support categorization (grouping) of those users and assignment of users to more than one category.

The SOLUTION should provide a method of Automatic Login to allow users to specify a default user to login on start up.

The SOLUTION's security component should allow for the names and groups to be obtained from a domain server, active directory group or local users. Synchronization should be supported for these users and the server and the system should be able to automatically populate. It should provide support for 256-bit encryption, as well as provide support for restricted user login access from any node on the system, including Remote, Networked, Web and Wireless nodes.

The SOLUTION's security component should provide individual users with the following specifications (including, but not limited to): Name; Password; Locked/Unlocked; Account policy; Allowed actions (by module); Allowed operations for data, alarms, files and login stations; Denied operations for data, alarms, files and login stations; and Custom allow/deny operations configuration.

The SOLUTION's security component should provide individual groups with the following specifications (including, but not limited to): Name; Allowed actions (by module); Allowed operations for data, alarms, files and login stations; Denied operations for data, alarms, files and login stations; and Custom allow/deny operations configuration.

The SOLUTION's security component should have a method to reserve users and/or user names for use. These users will always consume a user license, but assures them that access will not be denied due to user counts.

The SOLUTION's security component should provide account policies with the following specifications (including, but not limited to): Name; Limits on password editing, expiration, time before password change and complexity; Limits on login failure such as login failure attempt counts, locking, auto-logout; and Hours during days of the week the policy users are allowed to access stations.

The SOLUTION's security component should provide support for touch screens (for example, offering a pop-up keyboard for entering the information) when the user is prompted for Username/Password for login. If a user fails to successfully log in, it shall be possible to bring up a Graphics display offering user-configured options (e.g., re-attempt to login, go to a "home" screen, go back to previous display, etc.). If a user fails to successfully log in after a configurable number of times, it shall be possible to lock that account out until either an administrator clears it, or after a preset amount of time has elapsed.

The SOLUTION's security component should provide that, besides the per-user/group restrictions on writing or reading OPC Data Points, an additional list of OPC Data points are to be identified as "Critical" points.

The SOLUTION's security component shall provide for configuration of an Auto-Logout period, whereby the user is automatically logged out of the system after a specified amount of time has elapsed. Should Auto-Logout be in effect, there shall be an early warning message prior to the logout action (time remaining configurable by the user). It shall be possible to optionally require a password for logging out early (so someone doesn't log another user out against their wishes).

The SOLUTION's security component shall provide a utility to show which users are logged into the system (both on the local machine, and on networked machines). Audit Trails should be available to monitor all log-in and log-out activity (name, time, node,

etc.). This information should be available for viewing or printing from other nodes on the same system and by authorized users or groups.

The SOLUTION's security component shall make items available to lock out on a per group or per user basis including, but are not limited to: Individual modules of the product; Exiting the application; Utilizing templates; Runtime; Configuration; Mouse operations (zoom, drag and drop, mouse clicks); Menu operation; Starting/Stopping of alarms or data logging; and Filtering, editing, refreshing or exporting data (Filtering, acknowledging or changing alarms; Printing; Switching languages; Running a script; Accessing Windows desktop; and Opening/closing graphics displays).

The SOLUTION's security component should ensure that security extends to Silverlight controls and displays. Also, it should be possible to replicate security databases for easier deployment on multiple clients.

WEB AND INTERNET CONNECTIVITY

The SOLUTION should provide a connection to a Web server integrated into the graphics module. It should be simple to "publish" to a Web server using the graphics module. A user shall only require Internet Explorer to be installed on their machine (Thin Client) to access the system, with any "Plug-ins" automatically installed for them by the Server when they first access a graphic, trend, or alarm screen.

The SOLUTION should allow for "publishing" so that displays can be viewed on the most popular internet browsers including Microsoft Internet Explorer, Mozilla Firefox, Google Chrome and Apple Safari.

The SOLUTION's Web and Internet Connectivity component shall make it possible to have "floating" Web licenses, whereby if one user logs off, that license is freed up so that another user can log in. The same security system employed by the HMI/SCADA software shall also manage users connecting via the Web.

The SOLUTION's Web and Internet Connectivity component shall utilize the product's language switching feature to switch between written languages through the Web interface.

The SOLUTION's Web and Internet Connectivity component shall allow the graphics files to be represented over the Web in the same manner as shown on a "standard" HMI/SCADA workstation (i.e., close-approximations via HTML are NOT acceptable: the graphics should look and interact in the same fashion as a regular station). The scope of this shall include: Visual objects drawn; and Visual Dynamics tied to these objects; User interactions (data entry fields, sliders, dials, etc.).

The SOLUTION's Web and Internet Connectivity component shall ensure that anywhere a file path name can be used within the graphics software, a URL can also be specified. For

example, the Working Directory can be either "C:\project1" or www.myWebserver.com, so a Display Load operation can go directly to a Web server for the display file.

The SOLUTION's Web and Internet Connectivity component shall provide Alarms on an IE Web station that shall look and provide user interaction in the same fashion as a standard workstation. The scope of this shall include: Visual alarm summary; Alarm Viewer features such as pop-up help, jumping to associated displays, etc.; Acknowledgment of Alarms (resulting in the same actions as if the alarm were acknowledged on any other workstation in the system); and The alarm configuration database shall be accessible over the Internet for configuration, diagnostics and troubleshooting.

The SOLUTION's Web and Internet Connectivity component shall provide Trend View windows on an IE Web station that shall look and interact the same as a standard workstation. It shall cost less per station to run via the Web than a standard workstation. A "standard" workstation user shall also be able to connect to a networked system of other "standard" workstations via the Internet.

DATA SOURCES

Data Manager

The SOLUTION shall provide an internal tool for creating registers, value sets and expressions to manipulate input data (e.g., through expressions) that is available to all modules of the software.

The SOLUTION should have read and write support for registers, value sets and expressions.

OPC Connectivity

The SOLUTION shall allow connectivity between the following communications protocol and the graphic display software module: OPC DA; OPC A/E; OPC HDA; OPC UA A/E; and OPC UA HDA. 9.1.2. The system will support OPC Data Access specifications version 1.0, 2.05 and 3.0 and OPC XML DA servers.

The SOLUTION shall support OPC Alarm and Events Specification Version 1.1 as well as OPC Historical Data Access Specification Version 1.2. The OPC Browser shall list all registered OPC Data Servers.

The SOLUTION shall support On-line OPC Data Configuration changes (e.g., adding a new tag on-line). OPC Data Ranges shall be read in and used to configure limits on graphic dynamics, trend limits, data entry field ranges, and so on. OPC Quality Flags shall be used in graphic dynamics (non-good quality indicated by change in color used for objects), data logging, and alarm presentation.

The SOLUTION shall provide a component for bridging/mapping OPC Data from one server to another shall be provided (e.g., map an AB data point to a Modbus register). This capability shall run as an NT Service; Configuration shall be stored in a database for easy access; Data should be allowed to pass through firewalls or over the Internet via native OPC UA capability.

BACnet

The SOLUTION shall be certified by the BACnet Testing Laboratories (BTL) to B-AWS profile. (BACnet Advanced Workstation), and shall support BACnet Revision 7 or greater.

The SOLUTION shall provide native BACnet connectivity. BACnet communication should be integrated into all modules of the HMI/SCADA system. The system should support BACnet data sharing and discovery services to allow BACnet communication to obtain information and properties of BACnet objects. The BACnet module should include user configured devices and ports.

The BACnet portion of the SOLUTION should include support for automatic retrieval of BACnet objects. The SOLUTION should be able to act as a BACnet Broadcast Management Device (BBMD).

The SOLUTION should include full support for BACnet Alarms. This includes integration to the alarm server and alarm logger.

SNMP Compliant Managed Device Connectivity

The SOLUTION shall provide the ability to communicate natively to any SNMP Compliant Managed Device on an Ethernet network. It should support both SNMP V1 and SNMP V2. It should also have an SNMP module that is part of the modular, but unified architecture.

The SOLUTION's SNMP module should allow tag import from local and remote devices. It should be able to return a binary value to show that a machine is "alive" or "off". A Full-featured SNMP data configurator must be included with: Auto Discovery of SNMP Compliant devices on the network; Auto-detection of Device Types with recognizable icons; Browsing for available objects within each managed device (OIDs); Human Readable names for the objects (OIDs); Importing of MIB files; Reception and handling of SNMP Traps; and Event messages based on SNMP Traps.

The SOLUTION shall provide functionality to use SNMP data as data tags in any dynamic element, expression, alarm, trend or report in the system as an OPC data point would be normally used. The SNMP trap (A/E) server should

allow native merging into the alarm module of the software package. The SNMP trap (A/E) server should allow native logging in the alarm module of the software package.

ALIASING

The SOLUTION shall, instead of connecting graphic dynamics, trend data, alarms, etc. directly to OPC Data tags, have a method of creating what some call "data pointers", "indirect tags", "data placeholders" or, as is designated in this document "aliases". The Graphics Module shall provide its own set of "local" aliases, which can be resolved within the graphic itself.

The SOLUTION shall provide "project-level" aliasing, which will be designated in this document as "Global Aliasing". It should also allow "object-level" aliasing within the graphic display module. It should be possible to import an alias configuration from a file. The global alias editor shall permit creating user-defined names for the aliases, permitting thousands of aliases per project.

The SOLUTION shall make it possible to define "themes" for global aliasing. (For example, themes could relate to a product being made, a batch ID, a building number, etc.). Each global alias will be tied to a theme, with each value for the alias being tied to a specific value/resolution of the theme. Therefore, simply changing the theme will change all of the associated aliases.

The SOLUTION shall make it possible, when setting a theme, to define the "scope" of the alias changes to the entire project, just the current application, or just the current document within that application.

The SOLUTION shall make it possible to import/export the alias definitions from/to a text file (e.g., comma separated variable CSV file). The configuration of the global aliases and themes shall be stored in an open database (e.g., Microsoft Access). The configuration of the global aliases and themes shall support "Find and Replace" editing of alias names.

LANGUAGE SWITCHING

The SOLUTION shall provide a native function for language-aliasing supported by a language configurator for multilingual language switching with automatic range and value scaling, as well as engineering unit conversions, formatting, and font selection. Language aliasing must apply to graphics displays, alarms and historical trends. Configurations data must be stored in an open database, such as MS-Access or MS-SQL Server.

The SOLUTION's Language Switching shall allow the user to configure the whole display or a specified object/symbol to switch. Language Switching shall be available through Pick action-Load display, Pick action-Drag/drop load display, Pick action-Popup window,

Pick action-Embedded window, Pick action-Set aliases, Pick action-Alias dialog, Tab Load Display, launching the display from the command-line and during a runtime session are modifiable via OLE automation.

The SOLUTION's Language Switching component shall allow that wherever a tag name can be used, a "Language Alias" can also be permitted. Confirmation Dialogue boxes must include the capacity to be language sensitive allowing the description of an object to utilize Language Switching. The language configurator must support Czech, Dutch, English, French, German, Italian, Russian and Spanish as default languages.

The SOLUTION's Language Switching component shall make it possible to add a new language to the configuration database, based on a Microsoft locale identifier (LCID), a 32-bit value that uniquely defines a locale. A locale is a set of user preference information related to the user's language. The locale determines how dates, times, currencies, and numbers are formatted, how items are alphabetically sorted, and how strings are compared.

The SOLUTION's Language Switching component shall make it possible that, throughout the graphics, trending and alarm applications, users can configure language aliases in data fields (e.g. process points and trend pens) with a simple browser methodology with a tree control to simply select any language alias by double-clicking the alias name.

The SOLUTION's Language Switching component must the definition of a custom dictionary. The custom dictionary must be exportable and importable to and from various applications in XML and CSV formats. The configuration of the language aliases and themes shall support "Find and Replace" editing of alias names.

REDUNDANCY

The SOLUTION must support OPC HDA and A/E Redundancy, as well as provide support for trend and alarm server redundancy including: Redundant Alarm Module AE Servers; and redundant logged data configurations (including: Historical Alarm Data Loggers; and Feature for automatic database synchronization; Automatic failover between node pairs).

The SOLUTION must support OPC DA Redundancy. It shall be possible to define which OPC DA node on the network is the "Primary" server station. It shall be possible to define one OPC DA node to serve as the "Backup" station. Settings for determining when an OPC DA node goes offline shall include: Frequency for server "alive" checks; Response failure time required for server to be considered down; and Reconnect attempt frequency.

The SOLUTION's Redundancy components shall allow that, when the Primary node goes off line, the Backup station will automatically take over serving the live and historical OPC Data. When the Primary node comes back on-line, there shall be an option to automatically switch back to that server, or keep the "Backup" server as the "new

Primary". The status of which OPC DA node is the active server shall be provided to the Graphics package (for use in dynamic connections for operator indication).

The SOLUTION must support OPC UA, OPC UA HDA, and OPC UA A/E Redundancy.

The SOLUTION must provide Store and Forward Technology for synchronizing node pairs and for synchronizing disconnected remote data servers including:
Alarm Logger Module Store and Forward; Trend Logger Module Store and Forward; and Global Diagnostics.

The SOLUTION shall provide a global diagnostic interface displaying the status of redundant system to users across the HMI/SCADA network.

SYSTEM MONITORING AND DIAGNOSTICS

The SOLUTION shall offer a "Monitor View" and a "Tracing View" (for diagnostics), which should include a centralized diagnostic area or form, and be offered to users with the minimum privileges. It should also support visualization of application version information; application Runtime status; and connections including OPC, Licensing and Runtime.

The SOLUTION's System Monitoring and Diagnostics component should support monitoring of redundancy and provide warnings and statistics. It should support notification when connected OPC servers become available and allow export of diagnostics to a usable format (e.g., .xml). The "Monitor View" shall support a System Tray icon which will provide a "visual balloon" interface.

TRAINING

The SOLUTION should be easy to learn, such that prerequisites for training do not extend beyond a basic understanding of computers and industrial measurement or control. It should be supported by monthly training courses provided by the vendor. It should also be supported by training courses offered in different difficulties (e.g., advanced and basic training).

The SOLUTION's training sessions should be taught by professional trainers within the vendor's company. These sessions should be taught optionally on site, with a customized curriculum (should the user request it) and in a hands-on manner, allowing students to spend more time on the software itself. These sessions should be accompanied by resources such as templates, displays and project files.

TECHNICAL SUPPORT

The SOLUTION should have support offered by a regional office or representative of the software supplier. It should have support offered by technicians or engineers

familiar with the software and industrial practices. Support should be offered through a variety of communications (including but not limited to: telephone, email, fax or Web). The support should be offered from at least 8:00 AM to 7:00 PM on weekdays, with optional 24-hour availability. The SOLUTION shall provide global support for international customers for appropriate knowledge and working hours.

The SOLUTION's technical support should include a support plan offered for purchase.

The SOLUTION's vendor should have in house support, if necessary, included in a support plan.

The SOLUTION should have a trouble-shooting and/or tracing ability.

SOFTWARE UPGRADES

The SOLUTION should have an offer for an annual software maintenance service for a nominal fee. Upgrades to the purchased software should be included in this service. The SOLUTION's vendor should have a record of providing an upgrade path for the product, as well as a method to upgrade or program licenses in the field.

SOFTWARE MANUFACTURING QUALIFICATIONS

The SOLUTION shall be provided by a software vendor which is an incorporated, established company in business for at least TWENTY-FIVE (25) years. It should be provided by a software vendor that has installed in excess of 125,000 systems.

The SOLUTION shall be provided with user references from current applications, as well as with system integrator references.



Founded in 1986, ICONICS is an award-winning independent software developer offering real-time visualization, HMI/SCADA, energy, fault detection, manufacturing intelligence, MES and a suite of analytics solutions for operational excellence. ICONICS solutions are installed in 70% of the Fortune 500 companies around the world, helping customers to be more profitable, agile and efficient, to improve quality and be more sustainable.

ICONICS products are used in building automation, oil & gas, renewable energy, utilities, water/wastewater, pharmaceuticals, automotive and many other industries. ICONICS' advanced visualization, productivity, and sustainability solutions are built on its flagship products: GENESIS64™ HMI/SCADA, Hyper Historian™ plant historian, AnalytiX® solution suite and MobileHMI™ mobile apps. Delivering information anytime, anywhere, ICONICS' solutions scale from the smallest standalone embedded projects to the largest enterprise applications.

ICONICS promotes an international culture of innovation, creativity and excellence in product design, development, technical support, training, sales and consulting services for end users, systems integrators, OEMs and Channel Partners. ICONICS has over 300,000 applications installed in multiple industries worldwide.

World Headquarters

100 Foxborough Blvd.
Foxborough, MA, USA, 02035
Tel: 508 543 8600
Email: us@iconics.com
Web: www.iconics.com

Czech Republic

Tel: 420 377 183 420
Email: czech@iconics.com

Italy

Tel: 39 010 46 0626
Email: italy@iconics.com

Germany

Tel: 49 2241 16 508 0
Email: germany@iconics.com

Netherlands

Tel: 31 252 228 588
Email: holland@iconics.com

France

Tel: 33 4 50 19 11 80
Email: france@iconics.com

UK

Tel: 44 1384 246 700
Email: uk@iconics.com

Australia

Tel: 61 2 9727 3411
Email: australia@iconics.com

China

Tel: 86 10 8494 2570
Email: china@iconics.com

India

Tel: 0091 22 67291029
Email: india@iconics.com

Microsoft Partner

Gold Application Development

2013 Microsoft
Partner of the Year
Award Finalist
Application Development



Further information:

www.iconics.com