CORRECTIONAL SERVICES CANADA
FACILITIES BRANCH
ELECTRONIC SECURITY SYSTEMS

---

22 October 2014

DESIGN REQUIREMENTS

FOR THE COMMAND AND CONTROL PLATFORM INCLUDING AN APPLICATION DEVELOPMENT AND SERVICE DELIVERY ENVIRONMENT

Table of Contents

Table of Abbreviations and Acronyms

The following abbreviations are used in this specification:

Abbreviation	Expansion
API	Application Programming Interface
ATP	Acceptance Test Procedure
BIFMA	Business & Industrial Furniture Manufacturers Association
CCDA	Command Control and Data Acquisition
CCTV	Closed Circuit Television
CMMS	Computerized Maintenance Management Systems
CD	Commissioner's Directive
CER	Common Equipment Room
CIP	Critical Infrastructure Protection
COTS	Commercial-Off-The- Shelf
CPU	Central Processing Unit
CSA	Canadian Standards Association
CSC	Correctional Service Canada
CSV	Comma Separated Value
DCMS	Door Control and Monitoring System
DES	Director Engineering Services
EIA	Electronic Industries Association
EAM	Enterprise Asset Management
FAAS	Facility Alarm Annunciation System
FAR	False Alarm Rate
FDS	Fence Disturbance Detection System
FIU	FAAS Interface Unit
GFE	Government Furnished Equipment
GUI	Graphical User Interface
HMI	Human Machine Interface
IDE	Integrated Development Environment
IP	Internet Protocol
I/O	Input / Output
MCCP	Main Communications and Control Post
MSDE	Microsoft Database Engine
MDS	Motion Detection System
NAR	Nuisance Alarm Rate
NTP	Network Time Protocol
OLE	Object Linking and Embedding
ONVIF	Open Network Video Interface Forum
OPC	OLE for Process Control
PA	Public Address
Pd	Probability of Detection
PIDS	Perimeter Intrusion Detection System
PIU	Perimeter Intrusion Detection System Integration Unit
RFP	Request for Proposal
PLC	Programmable Logic Controller

PPA	Portable Personal Alarm
PPAL	Portable Personal Alarm Locatable
PSIM	Physical Security Information Management
RDP	Remote Desktop Protocol
REST	Representational State Transfer (web service)
RFID	Radio-Frequency Identification
ROC	Rate of Change
RTU	Remote Terminal Unit
SIO	Security Intelligence Officer
SOAP	Simple Object Access Protocol
SOW	Statement of Work
SQL	Structured Query Language
STR	Statement of Technical Requirements
TCP/IP	Transport Control Protocol/Internet Protocol
TER	Telecommunications Equipment Room
UPS	Uninterruptible Power Supply
V&C	Visits and Correspondence
VIRS	Visits Intercept and Recording System
VMS	Video Management System

Table of Definitions

The following definitions are used in this specification:

#	Term	Example(s)	Description	Function
1	Administrative User Interface		Monitor and Software that supports task specific User Interaction for System Administrators, located in a secure area	Provides Administrative Personnel with the ability to map enrolled users to the functional domains that they are allowed to access and change
2	Application	Cell Call Management, PA Management	Software that is used to deliver Application Support functionality for a sub-system	Software that provides the Operator Interface and supporting logic that allows a sub-system (Control Domain) to be managed
3	CCTV Monitor	PIDS or Range CCTV Monitor	Computer Monitor Hardware	Displays CCTV images for Operator viewing
4	Client		Rack mounted computer located in a secure area away from a Control Post or Control Desk.	Runs software and supports one or more Application
5	Configuration Data	Site floor plans showing quantity of cameras, doors, cells etc. Camera locations. Number of User Interfaces required in a Post.	Site and System specific information typically supplied by CSC that defines how a sub-system Application is to be set-up for a site, location within a site, or post.	The configuration data provides the information that a sub-system application requires to tailor it to meet site, location within a site, or post user requirements.
6	Configuration User Interface		Monitor and Software that supports task specific User Interaction, located in a secure area	Allows suppliers or qualified personnel to add, delete and modify Application Configuration
7	Contract Authority		Public Works and Government Services Canada (PW&GSC) is responsible for all contractual matters associated with the system design and implementation.	
8	Contractor		The company selected as the successful bidder.	
9	Control Console	MCCP Console, Living Unit Control Post Console	Console, typically located in a Control Post. Serves as the physical support infrastructure for Operator User Interfaces	Contains User Interfaces or Control Panels used by staff to execute their management responsibilities and interact with the Domains over which they have Control
10	Control Desk	Living Unit Control Desk	Desk, typically located in a Control Post or Office. Serves as the physical support infrastructure for Operator User Interfaces	Equipped with User interfaces used by staff to execute their management responsibilities and interact with the Domains over which they have Control
11	Control Domain	Cell Call, Guard Tour, Public Address	A group of Physical and Virtual devices or objects, often supported by specialized hardware and software, that performs a set of related functions	Collect information, or activate capabilities in their operational domain

#	Term	Example(s)	Description	Function
12	Control Panel	PACP, Fire Alarm	Hardware and Software device that provides an Operator Interface (I/O device), located in a Control Post	Allows Operators to manage one or more Domain
13	Control Post	Living Unit Control Post/MCCP	Room or area, typically located in a secure area in an institution	Room used by staff to execute their management responsibilities and interact with the Domains over which they have Control
14	Custom Equipment		Equipment designed and/or manufactured specifically for a specific contract.	
15	Design Authority		Director, Electronic Security Systems (DES) Correctional Service of Canada (CSC) is responsible for all technical aspects of the system design and implementation.	
16	Device	CCTV Camera, Managed Door, Call Origination Device	A specialized device, typically consisting of hardware and software	Provides data collection or activate functions associated with a specific system or sub-system
17	Enrolment User Interface		Monitor and Software that supports task specific User Interaction, located in a secure area	Allows Designated Personnel to enroll and delete Users from the Command, Control and Data Acquisition System.
18	Maintenance User Interface		Monitor and Software that supports task specific User Interaction, located in the CER or Maintenance Service Provider Office	Provides Maintenance Personnel with the ability to interact with one or more Systems to carry out their day to day tasks to troubleshoot and maintain Systems and Subsystems
19	Notification	Notification that a door is opened, or a door is closed, or a sensor is in alarm	A notification is a message that can be shown on a User Interface and/or logged in a database that represents a change in state or a command initiated by an operator.	
20	Off-the Shelf		Equipment currently on the market with available field reliability data, manuals, engineering drawings and parts price list.	
21	Operator User Interface	PIDS Display, Door Control and Monitoring System Display	Computer Monitor and Software that supports User Interaction (I/O device)	Provides an Operator with the ability to interact with one or more Systems to carry out their day to day tasks at a Control Console or Control Desk
22	Project Officer		A CSC employee or a contracted person designated by DES to be responsible for the implementation of the project.	
23	Reporting User Interface		Monitor and Software that supports task specific User Interaction, located in a secure area	Provides Management Personnel with the ability to access preconfigured reports and to create custom reports

#	Term	Example(s)	Description	Function
24	Server	Network Video Recorder	Rack mounted computer that runs software and is located in an equipment room such as a CER or TER	Runs software that is used to deliver services that support Command and Control Applications to connect to sub-systems
25	State		The state of a device as reported to a sub-system or system	This is a logical representation of the state of a device that is being monitored or managed
26	Sub-system	Cell Call, Guard Tour	A group of Physical and Virtual devices or objects, often supported by specialized hardware and software, that perform a specific set of related functions	Collects information, or activates capabilities in their operational domain
27	System	PIDS	A group of Physical and Virtual devices or objects, often supported by specialized hardware and software, including devices from sub-systems that perform a more general set of related functions	Collects information, or activates capabilities in their operational domain
28	Touch Screen User Interface	Door Control and Monitoring System User Interface	Typically an LCD Monitor with touch screen technology	Allows an Operator to view and interact with the Systems presented on the Monitor
29	Workstation		Rack mounted computer located in a secure area away from a Control Post or Control Desk	Runs software that is used to deliver Command and Control Capabilities

1 Scope

This standard defines the essential design and functional requirements of the Correctional Service of Canada (CSC) for the Command, Control and Data Acquisition (CCDA) Platform, including the Application Development and Service Delivery Environment that will be deployed to support the applications and services needed to deploy “Next Generation” Command and Control Systems at all Control Posts in Federal Correctional Institutions. The Platform must include:

- An integrated development environment (IDE) to build custom, domain-specific applications using a consistent and standardized graphical user interface layout provided by CSC
- A runtime environment where the applications will run when in-service
- A service and application delivery environment for configuring and deploying applications in a distributed network
- A security infrastructure that defines user access and scope of control to the system platform, its applications and managed security systems and devices.
- A high availability, reliable, redundant and robust processing environment such that the failure of any one hardware or software element will not disable the Platform and running applications
- A software framework with tools and libraries that simplifies application development and communication with local and remote security devices and mediation systems distributed within a correction institution,
- A software framework with the ability to create response plans for events or alarms triggered by devices or users.
- A structured and defined database or file that contains the complete configuration of the system as implemented for a specific location that can be exported for backup, restoration and training purposes,
- A centrally managed and accessible data logger or historical database for real-time data collection, management, storage and retrieval of the system’s events and alarms

2 General CCDA Software Requirements

The primary purpose of the Command, Control and Data Acquisition (CCDA) software is to provide a development framework or “integrated development environment” (IDE) that will support CSC’s requirement to replace disparate legacy Security Management, Operational Management Applications and User Interfaces. The next generation technology must provide intuitive, easy to operate applications with common and consistent “Look and Feel” colour graphical user interface (GUI) deployed across multiple subsystems and institutions, supporting a “national” design standard. The GUI must also be displayed on touch screen monitors which will serve as the primary user input device.

The CCDA software is intended to create a uniform software development environment and component model across all layers of the device-to-presentation solution. This must allow developers to work in the same programming environment with the same tools.

The CCDA IDE must consist of a human machine interface (HMI) system with support for situational awareness, supervisory and process control, real-time data acquisition, alarm and event management, historical data collection, report generation, local or remote telemetry communications to any mediation sub-system or security device and internet/intranet access. The software must be easy-to-use, with an object-oriented development environment and must support the .NET software framework developed by and integrated into Microsoft’s Windows operating system platforms.

The system must have the built-in flexibility to permit easy configuration of the system in accordance with the specific end-user requirements as well as quick and easy modification by on-site systems integrators, systems engineers and maintenance service providers, with appropriate software access control protocols to eliminate uncontrolled access.

The software must consist of a suite of off-the-shelf modular components from a single software manufacturer that are tightly integrated together to perform all CCDA system functions. The suite must contain an interactive system (HMI) for presenting, monitoring alarms events and controlling processes and managing devices and systems. The CCDA must also provide tools to perform administrative, enrolment, configuration, maintenance and reporting activities.

The software must also have the ability to easily interface with external databases, modeling and simulation, such as Enterprise Asset Management (EAM) and Maintenance Management Systems (CMMS).

2.1 Support for Global Standards and Protocols

The CCDA system must facilitate integration of edge devices, mediation systems or servers (RTU’s, PLC's or Door Control systems, CCTV systems, PA systems, etc.) using common and global standards and protocols such as Modbus, OPC and CIP for the interface to physical security products.

The CCDA must have the ability to connect to the devices using web services such as ONVIF (SOAP) and PSIM (REST).

2.2 Extensibility

Recognizing that the specific functionality of the supplied software is unlikely to meet 100% of the requirements for the CCDA system, the software provided must be extensible by a competent system integrator or systems engineer.

Advanced toolkits must be available to enable custom software components to be developed and integrated with the CCDA software.

The software architecture must be capable of supporting five hundred thousand (500,000) I/O and fifty (50) nodes in a distributed network such as the kind that will be deployed in many of CSC's Correctional Facilities.

2.3 Operating System Support

The software specified must be licensed to the latest releases of any of the following operating systems supported by Microsoft on appropriate hardware in any combination of:

Microsoft Windows 7.x or Windows 8.x (Professional versions)

Microsoft Windows Server 2012 or later

32-bit and 64-bit versions of the operating system depending on the hardware architecture.

Support for tablets running any of the above-mentioned operating systems must be supported.

In the case of a later release of operating system the manufacturer must have a defined statement and roadmap for supporting that operating system.

The CCDA software must also be supported in virtual machine environments including Microsoft Hyper-V and VMware environments.

Hardware must support multi-core and multi-processor computers.

3 Integrated Development Environment (IDE) Software Requirements

This section describes the integrated development environment (IDE) requirements to develop and customize domain-specific software for the CSC's next generation of the command and control.

3.1 Development Environment

The IDE must consist of a single, integrated application which must be able to manage all aspects of development and testing of a CCDA application.

The IDE must provide simultaneous multi-user capabilities, where application developers are subject to security permissions based on predefined system-wide roles.

3.2 User Authentication

The IDE must be configurable to support Microsoft operating system authentication, for example Active Directory Domains, to allow application developer access to the CCDA to view, configure, or modify applications.

3.3 Integrated Software Revision Control

The IDE must provide a management and control of changes made by developers to the source code of applications and graphical elements. The revision history for each change to the source code must include the User ID of the individual making the change(s), the time and date stamp, and a summary of the changes made.

The IDE must allow application developers with configuration rights to view a developed application and ensure that only one individual can perform source code changes to a specific application at any one time.

3.4 Managed Device Application Models and Templates

The IDE must use object modelling concepts to allow designing of applications which integrate the physical characteristics of the correctional institution, including geographical topology, physical equipment and computer locations. The system must provide the ability to develop applications, modules (stand-alone entities) which represent real world devices such as perimeter sensors, motion sensors, personal portable alarm devices, CCTV Cameras and associated control systems, door position switches, door controllers, power controllers, RFID card readers, intercoms, public address devices, push buttons etc.

The IDE must provide the ability to create application or module templates for the various domain applications and assign application logic, alarms, security and historical data to them. The templates must be reusable and extensible (refinement of parent template) to meet specific needs of a device or sub-system.

The CCDA must provide the ability to create applications or modules representing complex devices consisting of smaller and simpler devices by grouping or combining two or more application templates (e.g. doors configured in interlock series). Each templates adds specific attributes and logic to the top level application/module template.

The application/module template must allow for simple configuration for tracking changes to device's attributes values or states and writing the change to a data logger (historical event database).

The application/module template must allow configuration of attributes that will raise or generate an alarm upon a condition change in their value. For example, the IDE must support condition-oriented alarms (LoLo, Lo, Hi, HiHi, Rate-of-Change Deviation, etc.) and event-oriented alarms (True/False, Fail to Open, Fail to Close, Command Disagree, etc.) with predefined tools that will step the application developer through the process of defining the configuration.

3.4.1 Application/Modules Templates Security

The IDE must be able to configure security within the device application or module templates. At a minimum, runtime operational permissions must allow for:

The access or denial of the ability to view, modify or delete a device's application/module's attribute.

The access or denial to acknowledge an alarm in the runtime environment.

The modification of configuration attributes which allows application developers to configure the attribute's value (for example, a PLC register that defines a discrete input).

The modification of operational attributes which allows application developers with operational permissions to do certain day-to-day tasks like changing a setpoint, an output and control mode for a device, commanding a device or open and view a process or application window.

The modification of attributes which allows application developers to fine tune the attribute of an application/module in the runtime environment. Examples of tuning are attributes that adjust alarm setpoints and perimeter intrusion device sensitivity.

3.4.2 Device Application/Module Graphics and Animations

The application/module template must allow association and configuration of one or more graphical representations for visualization purposes. The visualization application itself must be able to be represented and contained by a module template. Please see Section 3.7.2 for additional details on adding graphics and animation to an application/module template.

3.4.3 Application Logic through Scripting

The IDE must allow integration of the application logic and behaviour by means of writing scripts associated to a change in an attribute value. When an application/module's value is modified during its operation, a logic script can be triggered to perform a specific action, process or generate an event or alarm.

Any scripting language must allow for integration of Microsoft's .NET (dot NET) software framework. .NET-based libraries are widely provided by security devices and mediation system manufacturers for easy integration of device-specific control logic into software.

The scripting language must be easy to program using common scripting syntax and statements requiring minimal knowledge of other programming languages. The application developer must be able to edit or modify the logic scripts while the system is monitoring the process (at runtime).

The scripting language must support the configuration of system logic into device application/module templates to monitor the status of each attribute in the system and perform specific functions based on the type and status of an alarm condition.

The scripting environment must support the configuration of device application/module that perform application control based upon an application developer definable state of a device and attribute or upon the result of an expression involving multiple object attribute names, including discrete attribute names, on/off state, alarm states such as Lo, LoLo, Hi, HiHi, or equivalence to a specific value.

The system must support the configuration of scripts that perform application control to change the state of discrete points, show windows, etc. This application logic must be able to also start and stop other application programs such as Microsoft Excel, Microsoft Word, Crystal Reports and other Windows based applications.

3.5 Device Application/Module Deployment

The IDE must provide the ability to deploy the device application/module templates from the development environment to the runtime environment. Deploying an instance of an application/module template makes the object active and functional.

All instantiated application/module components must be configured and deployed from the IDE to target workstations and servers.

3.6 Import/Export Utility

The IDE must support the import and export of the device applications' configuration data into a human readable file format such as CSV (comma separated file format) or other format for editing.

It must be possible to configure one (1) or more instances of application devices of the same type (based on the same template) from a CSV file or other format containing columns with values corresponding to object's attributes.

3.7 HMI Development Software Requirements

This section describes the engineering development requirements of all HMI system software functions. This includes development of colour graphic displays, configuration of the real-time and historical database, alarms, communications to field devices and application setup of clients and servers on the CCDA network.

3.7.1 Support for Multiple Languages

It must be possible, using the CCDA system's IDE, to construct an application which at runtime may be dynamically switched from one language to another.

The system should support any currently available language setting of the operating system.

At least two default languages must be able for selection (English and French).

Strings used in text objects with graphics must be able to be configured to be displayed in multiple languages and to be dynamically set by the system's local preferences when the operator logs into the application.

Alarm messages must be able to be configured to be displayed in the currently selected language.

Text with no equivalent translated string in the current language must be displayed in the default language of the HMI.

3.7.2 HMI Development Environment

The CCDA system's IDE must include software to create the layout of graphical user interfaces (GUI) or HMIs screens or windows. At runtime, the graphical symbols must be able to represent the current state of an application/module through animation linked to values of the device's attributes (operational data). The HMI must provide users with a realistic visualization of the CCDA system processes.

The CCDA IDE must allow for easy importing of floor plan images (vector-based format) representing the building layout of a correctional institution. The IDE must provide tools to animate the floor plan by including symbols representing doors, gates, video cameras, microphones and other devices. These symbols must provide visual cues (animation) depending on the real-time operational state of the device.

The application developers must be able to define graphic screens while the system is monitoring the process.

3.7.2.1 Integrated Graphics Editor

A graphics editor must be included with the (IDE) and must include a set of basic drawing tools to create simple or complex graphic objects. The toolkits must provide options to create lines, rectangles, polygons, ellipses, circles, open curves, closed curves, two point arcs and pies, three point arcs and pies, and filled shapes or text.

Any of these graphical objects can be assigned various attributes such as line colour, fill colour, size, and orientation and can be made static or dynamic. Text objects must be scalable and use true fonts in bold, italic or underline. All objects must be scalable and moved in any direction one pixel at a time or dragged with a mouse. A standard palette of 48 pre-defined colours must be provided.

A user defined colour palette can be created, exported and imported. The colour palette must be based on 16.7 million colours. The system must also support the user choosing transparent colours for all graphical objects and backgrounds. Possible colours must be solid, one, two and three colour gradients, patterns, textures, and no fill.

3.7.2.1.1 Editing and Manipulating Graphics

The graphics editor must support standard object manipulation functions such as cut, copy, paste and delete. Alignment tools must be included to simplify proper placement and arrangement of objects. Align commands must be included to align objects based on justification to the left, right, center, top or bottom. Object commands must also be included to space them vertically, horizontally, move to back, move to front, rotate or group and ungroup.

The graphics editor must be able to aggregate graphical elements together into a group as though they were one single graphic. Modifying a specific element of a group must be possible without breaking the group.

The graphics development environment must support the copy of single or multiple animated graphic objects and symbols from one window or display to another retaining all of the animation characteristics, links and attributes. In addition it must be possible to import windows from another application in this same fashion.

The graphics editor must provide the ability to import a suite of icons or graphics provided by CSC corresponding to complex security, operations and communications related such as Perimeter Sensors, Motion Sensors, Personal Portable Alarm Devices, CCTV Cameras and associated control systems, Door Position Switches,

Door Controllers, Power Controllers, RFID Card Readers, Intercoms, Public Address devices, Push Buttons etc.

All complex graphic objects must be scalable to any size and may include animation links to provide dynamic response based on real time data or user action.

The graphics editor must allow layering of objects to activate specific objects based upon conditions in the process. The graphics editor must be able to control and visually see the layering order of superimposed graphical elements.

3.7.2.2 Importing Image Files

The IDE's graphics editor must also allow the user to import drawings and images in BMP, JPEG, EMF, TIF, PNG and ICO file formats.

3.7.2.3 Embedding Logic into Graphics

The IDE's graphics editor must provide the ability to add logic and specific behaviour to the graphics with the use of scripting language.

It must be possible to use system calls from the .NET library within the scripting language.

Script writing through the graphics editor must provide the option to show or hide the graphic from user defined and named attributes (e.g. if a value is True or False, On False) or on changes to application object attribute values. The application developer must be able to define local and external device attributes of the following data types: Integer, String, Boolean, Float, Double, Time, and Elapsed Time.

There must be no practical limits to the number of graphic symbols that can comprise a single compound symbol, up to, and including an entire window of symbols. Symbol must connect either to applications/modules, or reference other attributes of the HMI or script variables. Changes to the graphical symbol must propagate to all instances of the symbol.

3.7.2.4 Graphics Import and Export

The customized graphics toolkit created with the graphic editor must allow for the import and export of both the graphical symbols and the logic and behaviour assigned to them via scripting.

3.7.2.5 Graphical Animation

The graphics editor must support configuration of the following animations as a minimum. Graphical objects must be able to be animated based upon any user-defined criteria made up of device attribute names in the system. This includes the use of expressions containing mathematical functions and the status of analog and discrete values in the system.

Animation Through Usage of Colour:
The colour of graphical symbol may be changed dynamically by discrete, analog, or string references through either scripting or by association to application object attribute value changes. There must be no fixed limit on the number of colour changes a single symbol may have. Percentage of colour fill for symbols should be possible by associating the fill value to an application object's attribute value.

Animation using Blinking:
A graphical object may blink based upon any discrete expression, alarm, event, or upon a designated group of alarms. The blink must be adjustable to slow, medium or fast.

Visibility and Transparency:
Each symbol must have a visibility and a transparency attribute option allowing for visibility/transparency of the object based upon the status of an analog or discrete point, alarm, or operator security level.

Size, Location and Orientation:
The system must support animation of symbols via re-sizing, moving, and/or rotating based upon a change in a tag name or hierarchical name.

3.7.3 Alarm Summary/Alarm History

Alarms configuration must include parameters to colour code alarms according to their state and priority. The colour coding must also apply to various states of alarms (unacknowledged, acknowledge and returned to normal but is not yet acknowledged states). The user must be able to choose from 32 different colours for display of each of these alarm states.

Alarms must be able to be shown either as real-time alarms or as historical alarms. The same object must communicate with the live process or the alarm history database.

3.7.4 Embedded Help

Context sensitive help must support the building of the HMI application's screens and windows. Application developers must be able to obtain immediate help on all configuration subjects by pressing a single function key.

3.7.5 HMI Application Management

The CCDA system IDE must integrate an application management and delivery software to easily manage the HMI application(s) deployed to target workstations.

The HMI application management and delivery software must propagated any changes made to a visualization application to all host workstations through the entire CCDA system. The Application Manager must provide the capability to dynamically change the resolution of the application windows. This will allow graphic displays to be developed on workstations with different display resolutions and convert them to the desired resolution quickly so that they are all consistent in look and feel.

3.7.5.1 Distributed Network Application Management

The CCDA software must provide standard functionality that will simplify the configuration, operation, troubleshooting and maintenance of the application by providing means of easily distributing the application in network environments.

The management software must allow a single master application to be developed and maintained on the network. The IDE must allow automatic distribution of the master application to all nodes on the CCDA control network as well as propagation of changes to the master application to all its instances in the system.

3.7.5.2 Notification of Application Changes to Client

When a client node is deployed with an HMI application, the client must maintain a copy of the application on its local hard drive and become registered as an Application developer of that application. When a change to the template application is detected, each registered user node must be notified of the change. The IDE must allow the application developer to define how the client node is notified of the change in the application. The client node must either automatically load the new application, prompt the application developer to load changes or ignore, or automatically ignore such changes. If a network failure occurs between the repository and client, then the client must continue to run the last distributed application. When the network is restored and the application has changed, the system will then distribute the application to the client.

4 The Runtime Environment

This section describes the various user interface functions of the system in the runtime mode in any combination as follows:

4.1 Alarm Management

Alarms must be detected and reported by an Alarm Manager Service. The Alarm Manager Service must support no less than two hundred (200) simultaneous alarm client displays. In the event of an alarm storm (hundreds or thousands of alarms detected within one second), the Alarm Manager must report and the client must be capable of displaying up to one thousand (1000) new alarms within ten(10) seconds of the detection of the alarms.

The system must be able to collect, notify and record various types of alarms such as:
- System Resources such as CPU utilization, memory exhaustion, etc.
- Network Status such as IP connectivity, bottlenecks in message transmit/receive throughput, etc.
- Environmental conditions alarms that may impact the good function of systems or devices
- Domain-specific alarms (e.g. door access, door forces, video camera failures, control post shutdown, etc.)
- User activities (e.g. unusual user activities – excessive login attempts)

Alarms must be logged to a database (preferably a Microsoft SQL Server or MSDE-(Microsoft Database Engine). Alarm events to be recorded must include alarm instantiation, alarm return-to-normal, and alarm acknowledgment. Items to be logged in addition to the alarm event must include date and time of alarm event, Alarm Group, Alarm Name, Alarm Data Type (real/integer/boolean), Alarm Type (LoLo, Lo, Hi, HiHi, Rate of change, deviation, disc, etc.), Operator Name, Operator Node of alarm acknowledgement, and Alarm Priority.

An Alarm Purge service must be provided to automatically purge and optionally archive alarms that are older than an defined period of days online.

Alarms may be printed to a locally connected or network printer. The alarms printed from a particular node may be all alarms, only unacknowledged alarms, only acknowledged alarms, alarms from a particular alarm group or groups, alarms from a particular priority to a particular priority or alarms from multiple alarm providers.

When alarms are generated, the alarm management system must allow for sending emails to key operators with a summary of the alarms for action.

4.2 Distributed Architecture

The runtime environment must be based on distributed, peer-to-peer system architecture. It must be possible to scale the architecture from a single, self-contained node, to over 50 nodes. The architecture must contain a multi-computer model that is seen as a single distributed namespace in the runtime environment and does not require replication of data from one node to another.

4.3 Runtime Data Viewer

The system must provide a utility to view the real-time status, quality and value of any application/module attribute.

4.4 CCDA System Failover

The CCDA system software must provide high availability for all functions within a normal CCDA controls environment. High availability requirements also apply to logging of historical process data.

In redundant failover configuration, there must be a Primary and a Standby system that manages the primary active system and secondary system in case of failure. The system must execute active data and synchronize active data with the standby system. In the event of detection of any failure in data execution or communication with the active system, the standby system must take control and execute and communicate within the system.

Due to large numbers of devices to be managed by the CCDA, the system must support a redundant load sharing configuration allowing application loads to be shared amongst multiple servers until a failure occurs.

4.4.1 Defined Failure Events

The CCDA system must detect the following events within the CCDA system and network objects:

Communications failure to a single device or mediation system

Communications failure to multiple devices or mediations systems

Communications failure to of application logic

Alarm printer failure (Off-line, out of paper) and Alarm Manager Failure

Communications failure to the Data Logger

Data Logger rate of collection deviation

Low Disk Space on any Data Logger on the network

The CCDA System must detect any or all of the possible failures and allow client data recovery without operator intervention.

4.4.2 Application Redundancy (Workstation HMI)

The system must provide for the execution of standby applications that become active upon the failure of execution of active application or failure to communicate with the active applications. Separate configuration of standby systems must not be required. In normal operation, the Primary server along with its contained applications must be active. The backup server and contained applications must be kept in standby and must be synchronized with their active pair node.

Workstations with running visualization applications must be capable of automatically failing over to a redundant workstation. No Operator intervention must be required. The system must support execution of the visualization software and the engineering development tools in terminal services sessions while enforcing the configured operating system security model.

4.4.3 Alarm Redundancy

The system must provide for the handling of alarms from standby device applications that become active upon the failure of execution of active device applications or failure to communicate with the active device applications. Separate configuration of alarms in standby device applications must not be required. As with workstation application redundancy (HMI), device applications must be allocated to a Primary system that in turn ensures that contained standby device applications are created and deployed in standby mode for handling of alarms.

4.4.4 Communications Redundancy

The Active and Standby servers must be interconnected with redundant message channels if either one of the channel becomes inoperable.

4.4.5 Data Logger Storage Redundancy

The system must provide for storage of historical data values from active applications/modules. Upon the failure of execution of active primary applications/modules, standby applications/modules must be activated and assumed the task of providing data for historical data storage. Separate configuration of historical data storage for standby objects must not be required.

If the Data Logger is off-line or unreachable, the engines servicing active applications/modules must store the historical data locally, and forward the buffered data to the Data Logger when the logger server is available. Primary and standby engines must synchronize any buffered historical data storage data.

4.5 Runtime Security

Security of access is paramount in the Runtime environment.

The runtime environment must be able to be configured through and administration interface such that Operators of different abilities, roles and responsibilities are permitted or denied access to system functionality as appropriate.

The above security constraints must be provided by inherent capabilities of the CCDA software and not by the engineered application.

4.5.1 Changes to Runtime Data

Runtime changes to device application values must be subject to security authorization. Permissions that are configured using the IDE must be automatically checked at runtime for authorization including verification of identity and access permission related to the originator of the runtime change request.

Failed authentication requests should be logged.

The manufacturer must have published security guidelines to assist in securing the entire CCDA system.

4.5.2 Runtime Audit Trail

It should be possible to configure the system such that any runtime changes to a variable must provide an audit trail of application developer ID, full application developer name, previous value, and new value.

Attributes configured for verification must provide an audit trail of application developer ID, full application developer name, verifier username and full user name, previous value, and new value.

4.5.3 Workstation Security

The workstation must use the security model defined by the CCDA user profiles and groups.

The software must use data level security where the ability to modify a setpoint or other value is determined in the configuration database. Any changes to the data level security model must be seen by all operator stations without any modifications to the operator stations.

The security system must be capable of disabling access to all Microsoft Windows controls (file menu, close, minimize, etc.) and keyboard commands (Ctrl-ESC, Alt-Tab, and limiting external system access via Ctrl-Alt-Del).

4.5.4 Logging Operator Actions

All operator actions must be logged to an event logger. The event logger must keep track of each new operator log-on, log-off, setpoint change, or device control.

Each event log must record the date, time, operator logged in and the type of action taken (setpoint change, state change, etc.).

4.5.5 Value Change Event Logging

Any configured attribute may also be configured as the source for triggering an event. The event must be logged any time when the attribute value changes.

Events must be logged and include a date and time of the event and an event priority.

5 Workstations and Domain-Specific Applications (HMI)

5.1 Workstations – General Information

The CCDA software must be capable of running on commercially available computer systems running a supported operating system.

The CCDA software manufacturer must offer as a standard product on their standard price list a selection of suitable HMI panel hardware available with the CCDA software preinstalled.

The CCDA must be capable of supporting multi monitor operation.

If not built-in into the CCDA's suite of software, the CCDA must provide the ability to develop various applications running on a workstation that provide the following functions: Operator User Interface, Enrolment User Interface, Administrative User Interface, Diagnostics and Statistics Report User Interface, Configuration and Deployment User Interface and Maintenance User Interface.

This CCDA workstation must accept an RFID card input to login-logoff the user out of the system or from the user interface and automatically logout the current user of the system after a preconfigured elapsed time of inactivity

5.1.1 Thin-Client Workstation

No CCDA HMI software must be required to be installed on a Thin Client Operator Workstation. This workstation must require only the firmware or software required to initiate a Remote Desktop session. The CCDA HMI must support the latest Windows operating software with the capability of running Remote Desktop Protocol (RDP) to access an application's HMI.

No modifications to the CCDA HMI configuration must be required to allow running in a thin client configuration. The exact same application running on a Full Function Operator Workstation (thick client) must run in a terminal services (Remote Desktop) session.

A management interface to the thin client terminal population must be provided. It must be capable of remote monitoring, reboot, failover and screen shadowing.

The thin client hardware must be capable of supporting multi monitor operation.

5.2 Operator Workstation

The CCDA system operator must be able to execute all monitoring and supervisory control functions from this workstation. Typical operator commands include modifying setpoints for control loops, alarm acknowledgment and setpoint adjustment, auto/manual switching and on/off control of field devices and taking points or devices on/off scan.

The CCDA system operator must be able to access all CCDA device attributes or graphic displays from any workstation on the network without having to know which Data Logger or server the point or display resides on.

The CCDA system operator must be able to execute all monitoring and supervisory control functions from a thin client workstation or environment.

5.3 Enrolment Workstation

The CCDA must provide the ability to integrate an RFID card enrollment system to provide the appropriate institutional representative with the ability to add, modify and remove RFID cards with a user.

The Enrolment workstation must allow the assignment of the preconfigured CCDA security groups to an RFID card users with different operational roles to access the various workstations in the system.

5.4 Administrative Workstation

The CCDA administration workstation must provide the ability to enable or disable system users from the pool of previously enrolled users and to assign them system privileges. A user assigned to a particular role and group must inherit all privileges that were assigned to the security group.

The CCDA administration workstation must provide the ability to create various security groups that can be assigned to system users. The security groups provide users the ability to view, operate, modify or access the various workstations based on their operational role in the correctional institution.

5.5 Data Analysis and Statistical Report Workstation

The CCDA system software must include a set of easy-to-use client software tools for real-time, historical and trend analysis reports. This client analysis software may be used by engineering, maintenance or supervisory personnel who need information from the CCDA system but do not require access to domain-specific applications (operator, enrolment, administrative, configuration workstations). The client tools must be able to access data from multiple Data Logger repositories on the CCDA system network.

Engineering, maintenance or supervisory personnel must be required to log in with a password to access the database server. The individual must not have to know the location of the server on the network, only the name of the server. The data analysis software must include tools for advanced trending analysis and viewing of reports in spreadsheet or free form format.

The client tools must be available as a stand-alone program or as an applet embedding into the CCDA HMI displays so that any full function or view only operator workstation may have the same capability.

The system must provide the ability to create statistical and trend analysis reports of the system based on queries to the Data Logger database

The system must integrate tools to easily create customized reports by selecting any managed device attributes, a date/time, a specific event or an alarm, a user activity, etc.

The CCDA must also allow for creating and saving customized reports as templates for reuse.

The system must support the use of standard SQL to perform queries against the Data Logger database.

Integrated tools in the CCDA must be available to export the statistical and analytical information obtained from reports to Microsoft Excel and Word, .pdf or .csv file formats.

5.6 Configuration and Deployment Workstation

A Configuration and Deployment User Interface provides integrators or designated representative with the ability to configure all of the variable parameters of the various domain-specific applications, including the creation of screen layouts, maps, security devices, mediation systems, etc.

This workstation must also provide the ability to deploy applications and devices to remote servers and workstations (which put applications and devices active or “in-service” mode).

5.7 Maintenance Workstation

A Maintenance User Interface provides the designated maintenance operator with the ability to access all maintenance and diagnostic services, tools and menus available in the CCDA.

Maintenance User Interface must allow access to all of the functionality associated with the other User Interfaces, except for the Administrative User Interface.

The Maintenance User Interface must have utilities to view the real-time status and operational values of any application object attributes managed by the platform.

6 Data Logger (Historical Data Repository)

The CCDA system software must provide a real-time relational database (historical data repository) for long-term storage of operational data. The Data Logger must provide for the storage of real-time and historical data for each analog, discrete or string tag name. The Data Logger must also store summary, event, alarm and configuration data.

The Data Logger database must acquire and store process data at full resolution. The Data Logger database must include normalized extension tables for real time data and include a set of client tools for data analysis and reporting such as those described in earlier sections.

The Data Logger must be capable of running in a stand-alone mode without connection to, or configuration from the CCDA system. While there are always physical limiting factors such as disk space, there must be no programmatic limit to the amount of data that may be stored on-line. Additionally, there must be no performance penalty for long-term data storage. There must be no discernible difference in retrieval speed of data based on the age of the data. For example, the retrieval of two hours data stored two years prior must be the same as for two hours of data stored one day ago.

The Data Logger must automatically begin to acquire device attribute data immediately after a device configuration has been committed to the database.

6.1 Remote Station Acquisition and Analysis

A Data Logger must be able to be configured at a remote site to acquire data locally, and to be able to transmit that data to a second Data Logger in a tiered network architecture. Data must be able to be stored locally at the remote site and analyzed with the manufacturer's client tools.

Data must be able to be aggregated and transmitted to a second level Data Logger for further analysis.

Multiple remote Data Logger must be able to send consolidated or raw information to a single second tier Data Logger, or must be able to send the same information to multiple second tier systems.

A Data Logger must be able to accept data delivered from a previously disconnected remote location once connection is restored (high data traffic bursts). Data must be able to be stored on a remote collection system and forwarded to the Data Logger without operator intervention on restoration of network connectivity.

It must be possible to configure a Data Logger architecture to guarantee no information loss due to network connection failure.

6.1.1 Event System Configuration

The Data Logger must contain an event sub-system to monitor, record, and or respond to process or system events and to trigger some type of action when the event is detected. The event system must detect an event occurrence using pre-defined and configurable criteria; historically log when an event occurs and trigger designated configurable event actions based on the event detection. Event attributes must be logged to the database and will include the date, time that the event occurred, and the event criteria that were satisfied.

6.2 Disk Storage Management

The Data Logger must not require specialized tools for disk storage management. It must be possible to archive and retrieve historical data files using standard Windows® copy techniques. It must be possible to retrieve select portions of archived data without retrieving all archived data

The Data Logger must provide for a mechanism whereby current files on a disk drive that are nearly full will automatically be moved to a secondary device. The files and available space on the secondary drive must be monitored as well such that when an application developer-defined threshold is reached, the oldest files may be automatically deleted to preserve the integrity of the system. Historical files must never be deleted from the primary storage device if an appropriate secondary device is configured.

7 Software Warranty, Maintenance and Support

The software manufacturer must provide software maintenance and support program to ensure that the user receives full benefit of the software for the duration of its life cycle. The program must provide for basic warranty coverage and include an extended warranty for priority support and software upgrades as they are released. Telephone support must be available through a toll-free number during normal business hours. Support must also be available email or through a technical support website.

7.1 Warranty Support

The software manufacturer must warrant the products for a period of 90 days after delivery. During the warranty period, the manufacturer must offer free technical telephone support during normal business hours through a toll free number. All software defects must be resolved in a timely manner.

7.2 Extended Support and Software Maintenance

After the 90-day warranty period, the user must continue to receive technical support via fax, email or access the technical support website. In order to ensure that the user always has access to the latest software releases, long-term warranty and technical support, the manufacturer must offer an extended support program for a fixed annual fee.

7.2.1 Software Upgrades

The extended support program must entitle the user to receive the latest CCDA system software releases and version upgrades, as they become available. In order to ensure quality support for all users, all software licenses at the site must be maintained at the same version level. If software updates will cause incompatibilities with existing application objects and HMI applications, the incompatibilities must first be listed and an appropriate software development plan must be established to resolve the incompatibilities prior to the installation of the upgrade. All upgrades must first be fully tested and approved prior to adoption and deployment in the “live” or active environment.

The support program must include access to a secure web site for electronic software downloads. New software releases, service packs, patch fixes and other support files must be available from the secure web site for download to another storage medium (e.g. optical disc).

7.2.2 Operating System Patch Support

The manufacturer must test and support operating system patches which are periodically released by Microsoft. The manufacturer must have a defined policy of support for such security patches.

7.2.3 Telephone Support

The extended support program must include telephone support during normal local business hours. A technical support engineer who has been certified by the software manufacturer based on a certified support testing program must provide telephone support. A live person when calling during 24/7 schedule must provide unlimited telephone support. A voice-mail tech support system will not be acceptable.

7.2.4 Email Support

The extended support program must include e-mail support within one business day at a higher priority than non-warranty support users, and will forward them to the nearest certified technical support center. Electronic support must also include expanded access to advanced services on a technical services web page. The extended support program must include real-time access to current and past issues in a call tracking database, as well as the ability to create new issues, which must be immediately assigned to a technical support engineer for resolution.

7.2.5 Web-Based Support

The software manufacturer must have a development and support-oriented web site which provides technical information on the SCADA software, best practices for implementation of the SCADA software and user forums.

7.2.6 Newsletters and Technical Support

The software manufacturer must provide a newsletter and a technical support media with tech notes to all users on the extended support program a minimum of two times per year. The technical support media must include a comprehensive summary of technical notes, technical alerts, applications, application utilities, diagnostic utilities, drivers, scripts, script functions, and helpful hints that can streamline application development.

7.3 Software Backward Compatibility

The software manufacturer must have a track record of no less than 10 years of software backward compatibility and continuous migration path in order to protect engineering investment. Old applications must all be able to be easily migrated to the newest versions of the software without engineering modifications.