# WEBs-AX™
## Configuration & Installation Guide

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>2</td>
</tr>
<tr>
<td>1.1 Purpose</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Scope</td>
<td>2</td>
</tr>
<tr>
<td>2. System Overview</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Niagara AX® Framework</td>
<td>2</td>
</tr>
<tr>
<td>2.2 WEBs-AX™ (JACE) Controllers</td>
<td>3</td>
</tr>
<tr>
<td>2.3 WEBStation-AX™ (Workbench)</td>
<td>3</td>
</tr>
<tr>
<td>3. System Architecture</td>
<td>4</td>
</tr>
<tr>
<td>3.1 Niagara Architecture</td>
<td>4</td>
</tr>
<tr>
<td>3.2 Types of Architectural Configurations</td>
<td>4</td>
</tr>
<tr>
<td>3.3 Ports and Protocols</td>
<td>5</td>
</tr>
<tr>
<td>3.4 Data Transmission</td>
<td>5</td>
</tr>
<tr>
<td>4. Security and Auditing</td>
<td>6</td>
</tr>
<tr>
<td>4.1 Security</td>
<td>6</td>
</tr>
<tr>
<td>4.2 Audit Records</td>
<td>6</td>
</tr>
<tr>
<td>5. Installation and Configuration</td>
<td>7</td>
</tr>
<tr>
<td>5.1 WEBStation-AX™ Installation/ Requirements</td>
<td>7</td>
</tr>
<tr>
<td>5.2 WEBs-AX™ Controller Installation/ Requirements</td>
<td>7</td>
</tr>
<tr>
<td>5.3 Dependencies &amp; Interoperability Requirements</td>
<td>8</td>
</tr>
<tr>
<td>6. Hardening Guidelines</td>
<td>8</td>
</tr>
<tr>
<td>6.1 User Access</td>
<td>8</td>
</tr>
<tr>
<td>6.2 Category Restrictions</td>
<td>10</td>
</tr>
<tr>
<td>7. Administration</td>
<td>10</td>
</tr>
<tr>
<td>7.1 User Account Management</td>
<td>10</td>
</tr>
<tr>
<td>7.2 Configuration Management/ Provisioning</td>
<td>10</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 Purpose

This document captures high-level information on WEBs-AX™ system, based on the Niagara AX® Framework. The content is intended for end-users, especially those in IT department, to understand the fundamental workings of the software and its installation/configuration at site. It can also serve as an index document for people not familiar with the system.

1.2 Scope

The document does not capture all the details for the various features, options and services that can be configured with WEBs-AX™. In most cases, it includes references to other documentation available as part of the software installation (docs folder). Also, this document does not include site-specific settings and configurations which may vary depending on the system requirements at any given site.

2 SYSTEM OVERVIEW

2.1 Niagara AX® Framework

Niagara AX® Framework is a collection of software components that enable monitoring and control of Direct Digital Control (DDC) systems based on a wide range of legacy field protocols, such as LonWorks®, BACnet®, Modbus, etc, and work together as a seamless web-enabled system. Optionally, it also includes integrated network management tools to support the design, configuration, installation and maintenance of interoperable networks.

One of the most basic Niagara AX® concepts is a host: a host may be the supervisory workstation OR a specialized piece of hardware, such as a networked controller, which is designed specifically for integrating building systems into an existing TCP/IP network. Therefore, the same Niagara AX® software components can run on both the workstation and on the controllers providing common functions across the two hardware platforms.

Basic features of the Niagara AX® Framework include:

- Java™-enabled or standard HTML-based graphical user interface within a browser
- Support for an unlimited number of users over the Internet / intranet with a standard Web browser
- Audit trail of database changes indicating user, time stamp and audit action
- Synchronization of controller databases, database storage scheduling, control and energy management routines
- Sophisticated alarm processing and routing which includes email and paging
- Access to alarms, logs, graphics, schedules and configuration data with a standard Web browser
- Password protection and security using authentication and encryption methods
- HTML based help system that includes comprehensive online system documentation
- Supports multiple host/stations connected to a local Ethernet or the Internet
- Provides online/off line use of the graphical application configuration tool and an Object Library

For more information on capabilities and the entire software stack, please refer to docs/docUser/docUser.pdf (pages 1-1 to 1-6).
2.2 WEBs-AX™ (JACE) Controllers

WEBs-AX™ controllers, powered by the Niagara Framework, are dedicated Niagara host platforms that provide integrated control, supervision, and network management services for networks of building monitoring and control devices. This specialized hardware is also referred to as Java Application Control Engine, or JACE in short. When the controllers are connected over an Ethernet network using TCP/IP, they can communicate with each other on a peer-to-peer basis as well as communicate with other IP-enabled devices using common open protocols.

WEBs-AX™ controller integrates a variety of devices and protocols into a common distributed building automation system. It integrates diverse DDCs and field protocols into a common object model, embedded at the controller level and supported by a standard Web browser interface.

These controllers can also be configured to serve dynamic displays of the information contained in the connected devices to any standard web browser. WEBs-AX™ controllers can run different operating systems, such as QNX and Microsoft Windows (32-bit). The following WEBs-AX™ controllers are offered:

- WEB-201 (J-201) or WEB-600 (J-600): This device is a compact embedded controller and server platform with battery backup. It runs QNX RTOS, IBM J9 JVM, and Niagara AX Framework® software
- WEB-4XX/5XX (JACE-4/JACE-5): These devices are compact embedded processor platforms with flash memory, running QNX RTOS
- WEB-NXT (JACE-NXT): This device is a compact PC with a conventional hard drive that runs an embedded version of Microsoft Windows XP® and the Sun Hotspot VM
- WEB-7XX (JACE-7): This device is a compact embedded controller and server platform with battery backup. It runs QNX RTOS, IBM J9 JVM, and Niagara AX Framework® software

For the latest information on WEBs-AX™ controllers, please visit http://customer.honeywell.com.

2.3 WEBStation-AX™ (Workbench)

WEBStation-AX™, powered by the Niagara AX Framework®, is a Java-based desktop application that runs on the networked workstation and is designed to provide a front-end to the end-user for managing diverse building systems. It is typically configured to provide support services to other hosts within an automation system such as graphics, history, or alarm concentration. WEBStation-AX™ has the following features:

- Provisioning of the multi-controller systems (tools for updating and installation of software modules)
- Central database storage for attached controllers
- Archive destination/repository for log and alarm data
- Central server of graphics and aggregated data (single point of access to the system – one IP address)
- Optional support for integration with standard RDBMS (MS SQL, Oracle, DB2, etc.)
- Platform for optional enterprise applications

WEBStation-AX™ acts a network server or a ‘Supervisor’ for all connected WEBs-AX™ Controllers. It creates a network environment for the management of these controllers, alarms and messaging services. In addition, WEBStation-AX™ also provides an engineering tool, also called Workbench, and a graphical user interface.
3 SYSTEM ARCHITECTURE

3.1 Niagara Architecture

Niagara hosts are typically configured to remotely manage and share data across an installed IP network (LAN) on site. Figure 1 shows a typical configuration where both types of hosts are connected using more than one protocol.

For more information on various architecture components, please refer to docs/docIT/NetworkingITGuide.pdf (Chapter 1).

3.2 Types of Architectural Configurations

WEBs-AX™, through the flexibility of the Niagara Framework®, provides a scalable architecture from a standalone WEBs-AX™ Controller up to multiple controllers connected to a WEBStation-AX™.

At the enterprise level, WEBStation-AX™ is scalable to handle even the largest of multi-site (building) configurations. Each site or building can have its own WEBs-AX™ controller, providing local system (HVAC, Energy System, etc) monitoring and control. Only one central site hosts a WEBStation-AX™ (Supervisor) on a PC or server class machine for global management of data and administrative functions. By using the automation infrastructure based on Niagara AX®, information about buildings' HVAC system from many WEBs-AX™ controllers is pulled together into one central node (WEBStation-AX™) providing a site-wide solution. Various sub-systems can utilize this infrastructure within the site for data gathering and reporting, and ultimately close the loop for real-time control.
For more information on typical architectures, please refer to docs/docIT/NetworkingITGuide.pdf (Pages 2-12 to 2-15)

### 3.3 Ports and Protocols

Niagara AX® Hosts are typically configured to communicate on a peer-to-peer basis and optionally to support various communication protocols that could be IP or Non-IP based. Many WEBs-AX™ Controllers (JACEs) have two LAN ports that can be configured to support separation of IT/Data network at a site from a building system network.

At a minimum, two ports are used to enable peer-to-peer communication and support proprietary protocols: Fox and Niagarad. By default, these ports are configured on 1911 and 3011 respectively but can be changed, if required. These would allow only a Workbench (thick JAVA client) connection to the automation system on site.

To enable access to the system via a browser, an HTTP or HTTPS port must be enabled. These ports default to 80 and 443 respectively, but can be configured as well.

Niagara Framework® is extensible and 3rd-party solution vendors may implement their own protocols using different ports. For a complete list of optional ports, please refer to docs/docIT/NetworkingITGuide.pdf (Page 3-12).

### 3.4 Data Transmission

Data flow on the network can be classified into two groups:

**Browser User (Client) – System (Server) traffic**: This is the data traffic that is initiated by a user that logs into the system via a browser and Niagara host(s) serve-up web pages.

**Niagara Data traffic**: This refers to the data shared across different Niagara hosts for correct operation of the system. It includes Platform, Station-to-Station and Workbench (thick client)-to-Station communication. There are four categories of Niagara data traffic that could affect network bandwidth:

- **Configuration Traffic**: This is traffic that is associated with the initial setup and commissioning of a Niagara implementation. During system commissioning bandwidth varies depending on the number and type of objects being configured.

- **Logging (History) Traffic**: This is the scheduled bulk transfer of historical data being passed from the JACE to the Supervisor. This can be tuned to fulfill operational requirements and bandwidth considerations. The formula for calculating analog logs is:

  \[ \text{kbps} = 0.0012 \times \frac{\text{Number of points}}{\text{logging interval in minutes}} \times \frac{9}{\text{bytes per analog log value record}} \]

  For 100 analog points from a JACE being logged every 5 minutes this would mean a bandwidth utilization of approximately 0.024 kbps. For planning purposes you may want to add in 5-10% to account for IP packet overhead.

- **Real Time Data/Interstation Link Traffic**: This is data that is transferred from station to station for operational purposes. Inter-station links might be used for peer-to-peer control or other similar activities. This can be tuned to fulfill operational requirements and bandwidth considerations. Niagara supports a maximum of 50,000 links per Web Supervisor, and 1000 JACEs per Web Supervisor. Each record transmitted is approximately 9 bytes. The kbps for inter-station links can be calculated with the same formula used in calculating Logging Traffic. This will be the worst-case scenario if all the points changed within the interval in minutes.

  \[ \text{kbps} = 0.0012 \times \frac{\text{Number of links}}{\text{fixed interval in minutes}} \times \frac{16}{\text{bytes per analog log value record}} \]

  For 100 links points from a JACE that all happened to update during the same 5 minute period this would mean a bandwidth utilization of approximately 0.024 kbps. For planning purposes you may want to add in 5-10% to account for IP packet overhead.
Alarming Traffic: This is data that is sent during alarm conditions, though it cannot be predicted, it can be managed based on how aggressive alarm set points are configured. The size of a typical alarm message is approximately 256 bytes.

The services offered by Niagara should be properly configured to ensure minimal impact to site’s networking environment. For more information on tuning various services, please refer to docs/docIT/NetworkingITGuide.pdf (Page 4-9 to 4-21).

4 SECURITY AND AUDITING

4.1 Security

Niagara offers various levels of security using two mechanisms that are configurable:

Platform connectivity/authentication: This gives a highest level of access to setup and manage the Niagara host. This is enabled only through the Workbench (thick JAVA client) and requires authentication using valid login credentials. For WEBStation-AX, these credentials can also be linked to LDAP available locally the machine. Common operations include copying station (database) to the host, starting/stopping the station, managing software (configuration management), and managing TCP/IP configuration, among others. For a complete list of services offered through the platform, please refer to docs/docPlatform/docPlatform.pdf (Chapter 1).

Station connectivity/authentication: This is the typical user access enabled via both Workbench and Browser (if web service is enabled). It also requires authentication that is different from the platform connectivity. Access to the system is managed via categories and permissions. A category is a logical group that can be defined to control the type of data accessible in the system. These groupings can be by functions (alarming, scheduling, etc), by sub-systems (HVAC, Lighting, Electrical, etc) and by location (Floor 3, Zone 2.4, etc). Permissions are levels of access (read, read-write, etc) defined for every category for a particular user. Both user and category management is done via services provided at a Niagara station. For more information on the user security model, please refer to docs/docUser/docUser.pdf (Chapter 7).

4.2 Audit Records

Niagara Framework® provides a way to capture audit histories in a host (both WEBStation-AX™ and WEBs-AX™ Controllers). This is configured as part of a station instance, similar to user and category services. Enabling this service allows the system to capture events such as login/logout, logon failures and any change in the system. For more information on the user security model, please refer to docs/docUser/docUser.pdf (Pages 4-4, 7-9 to 7-11). Station security includes categories, users and permissions as depicted in Figure 2 below (Chapter 7 of above document).
5  INSTALLATION AND CONFIGURATION

5.1  WEBStation-AX Installation/ Requirements

WEBStation-AX™ software is distributed via the web or a DVD, and has the following minimum hardware requirements:

Processor: Intel Pentium® IV, 2 GHz or higher
Operating System: Microsoft Windows XP® Professional (32-bit), Windows 2003 Server (64-bit), Vista Ultimate (32-bit), Linux
Web Browser: Mozilla Firefox™ or Internet Explorer™ 5.0 or later
Memory: 1 GB minimum, 2GB or more recommended for large systems
Hard Drive: 1 GB minimum, 5 GB for applications that need more archiving capacity
Display: Video card and monitor capable of displaying 1024 x 768 pixel resolution or greater
Network Support: Ethernet adapter (10/100 Mb with RJ-45 connector)
Modem (if used): 56 KB minimum, full time high speed ISP connection recommended for remote site access (i.e. T1, ADSL, cable modem)

These requirements may be slightly different for different versions of WEBStation-AX™ as support for newer operating systems is added. For the latest product data, please visit http://customer.honeywell.com.

5.2  WEBs-AX Controller Installation/ Requirements

WEBs-AX™ Controllers, such as the WEB-600, require proper mounting/wiring on site. As such, the minimum requirement for these controllers is access to AC electrical power. For more information on installation instructions, please refer to pertinent documents in the docs/MtgWiring folder.

After the controllers are physically mounted, they need to be prepared and configured for the particular installation at site. This is typically done by the system integrator and requires platform connectivity through the workbench. For more information on setup and startup of the controllers, refer to docs/docJaceStartup/docJaceStartup.pdf.
5.3 Dependencies & Interoperability Requirements

WEBStation-AX™ is supported on Microsoft Windows operating systems as well as Linux. The application installation includes Oracle's Java Runtime Environment (JRE) so it is self-contained and does not depend on any other software to run. Also, in a standalone mode without enterprise (database) integration options, WEBStation-AX™ does not need to operate with any other software or system on site.

WEBs-AX™ controllers are self-contained IP-enabled devices, just like printers on a data network, which can be connected through a switch/router or directly to a PC/workstation.

6 HARDENING GUIDELINES

6.1 User Access

User services enabled at the Niagara AX® station provide several mechanisms to protect information in the automation system. However, the security of the system is governed by both physical and logical security.

Following are settings that are recommended for a typical secure site:

Physical:
- Physical access to the controllers should be restricted. Use of panels with keys, locked rooms, etc is recommended, wherever appropriate

Platform:
- Default platform username/passwords on controllers should be changed
- IP/LAN ports should be disabled if not used on the controllers

Station:
- Guest account is disabled
- Admin account is disabled. Instead, a different user is created with ‘super user’ privileges
- Each authorized user has its own credentials that are enabled to have strong passwords (min 8 characters, including upper/lowercase, numbers and special characters)

For embedded controllers, strong passwords cannot be enforced. However, this should not be a major concern if web services are disabled on the controllers.

Disable the Guest User: A station's UserService is designed so that whenever the built-in user "guest" is enabled, someone attempting to access the station will automatically be signed into the station as the Guest user--whether there is a password set for the Guest user or not. What that person is then capable of doing once inside the station is dependent on what permissions are assigned to user "guest". If the user attempts to do something that the Guest user does not have permissions to do, the user is then prompted to sign in as a different user, who does have the correct permissions.

By default, the "guest" user is disabled. We recommend keeping the "guest" user disabled. Then no one can access the station that wasn't given their own login credentials.

Use Strong Passwords: The stronger the password, the harder it will be for someone to hack into your station. From the property sheet of the UserService, you can configure to require strong passwords for all station users. See the "UserService properties" subsection in the "About Security" section of the NiagaraAX User Guide for more details. Please refer to docs/docUser/docUser.pdf.
Limit User Permissions: Only allow users access to the specific things that they need to do their jobs. This is especially important when it comes to the file system. If you give a user access to the entire file system, you have potentially given that user access to the entire station configuration.

Specifically, the config.bog file, located in the station's root directory, can be a security risk. Security should be configured to limit access permissions of the user to only folders the user is required to access such as px, images, and nav folders. To setup these limits, create a new category using the Category Manager. Using the Category Browser assign only file folders that you need users to access such as file:^px, file:^nav, etc. to the newly created category. Setup the user permissions to only include file permissions of the newly created file category. Keep in mind that by default all objects are assigned to category index 1 (Category1, or whatever the first Category component is named).

Example Category Browser view

Use the Lock Out Feature: If someone is trying to "hack" into the station by constantly retrying different passwords for a user, and the specified number of incorrect login attempts occur in a set time period, that user is "locked out" for the specified time period. See the "Lockout notes" subsection in the "About Security" section of the NiagaraAX User Guide for more details. Use the "Lock Out" feature in the UserService, enabled and configured from the service's property sheet.
You should also limit the Services you allow your users access. The UserService and CategoryService provide security risks if allowed to be accessed by all users. For more information about the UserService, Users, Categories, and properties/permissions, see the "About Security" section in NiagaraAX Users Guide. Note a subsection "UserService security notes" explains a special permissions scheme for a station’s UserService. Please refer to docs/docUser/docUser.pdf.

### 6.2 Category Restrictions

Categories should be defined based on local site’s IT Policies on management and operational control. As mentioned, these are logical groups that can be defined based on access to sub-systems, locations, functions or a combination thereof.

For more information on category service and options, please refer to docs/docUser/docUser.pdf (Pages 7-22 to 7-25).

### 7 Administration

#### 7.1 User Account Management

Though the system enables two levels of credentials, platform level access is almost never required for any end user untrained on Niagara AX®. In order to effectively manage user rights on station communication (Workbench or Browser), adequate policies should be implemented to ensure that user accounts are managed/ cleaned-up from time to time.

#### 7.2 Configuration Management/ Provisioning

 Provisioning is an optional service available for WEBStation-AX™ (Supervisor) that can be configured to provide system wide backups and routine software maintenance. For more details on the mechanism and specific features of provisioning, please refer to docs/docProvisioningNiagara/docProvisioningNiagara.pdf.