

Technical Document

Mobile Guide

May 27, 2018

niagara⁴

Mobile Guide

Tridium, Inc.

3951 Westerre Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2018 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

Contents

About this guide	5
Document change log	5
Related documentation	5
Chapter 1 Getting Started with NiagaraAX Mobile	7
Latest changes	8
NiagaraAX supported mobile operating systems	9
Mobile apps prerequisites	9
Licensing mobile apps	9
Installing NiagaraAX mobile apps	10
Install a NiagaraAX Mobile app	10
Enable or disable a Mobile app	11
Common Mobile interface controls and indicators	11
Mobile desktop view (App Desktop view)	12
Navigation bar	12
AppsFolder component	13
Chapter 2 Configuring a Mobile app	15
About Bajascript	15
About Mobile user access	15
Types of Mobile components	16
About BOX caching	17
Common Mobile App properties	18
About the Apps container	19
About the Mobile Property Sheet app	19
Mobile Property Sheet app interface	20
About status coloring in Mobile property sheet view	20
About the Alarm Console App	21
Mobile Alarm Console app interface	21
Alarm source data	23
Local storage	23
About the Mobile History app	23
Mobile History app interface	24
About the Mobile Schedule app	26
Mobile Schedule app interface	27
About the Mobile Px app	30
About Mobile Px views	31
About dynamic Mobile Px views	31
About absolute positioning in Mobile Px	32
About Bound tables in Mobile Px	35
Field editors, widgets, and bindings supported in Mobile Px	36
Chapter 3 Component guides	43
Types of component modules	43
Components in the App module	43
app-AppContainer	43

app-AppFolder	43
Components in the Box module	43
box-BoxService	44
box-ORDChannel	44
box-RegistryChannel	44
box-ServerSessionChannel	44
box-SystemChannel	44
Components in the Mobile module	44
mobile-NiagaraMobileWebApps	45
mobile-MobilePropertySheetApp	45
mobile-MobileAlarmConsoleApp	45
mobile-MobileHistoryApp	45
mobile-MobileScheduleApp	45
mobile-MobilePxApp	45
mobile-BasicMobilePane	45
mobile-CanvasPane	46
mobile-MobileGridPane	46
mobile-MobileJavaScriptButton	46
Chapter 4 Plugins (views)	47
Plugins in the app module	47
app-AppManager	47
Index	49

About this guide

This guide explains to the Systems Integrator how to install and configure the NiagaraAX Mobile applications (apps) that come with the NiagaraAX-3.7-3.8 Framework.

NOTE: With the release of Niagara 4.6, Mobile features are deprecated, being replaced by HTML5 based features. For details, see the : “Latest changes in mobile” in this guide.

Document change log

Updates (changes and additions) to this NiagaraAX Mobile Guide document are listed below.

- Revised: May 27, 2018
Added statement to *About this guide* topic to indicate that Mobile features are deprecated starting in Niagara 4.6.
- Revised: November 12, 2013
Updated for NiagaraAX-3.8 enhancements.
- Initial publication: August, 16,2012

Related documentation

The following documents are related to the content in this guide and provide additional information:

- *NiagaraAX User Guide*
- *NiagaraAX Video Guide*

Chapter 1 Getting Started with NiagaraAX Mobile

Topics covered in this chapter

- ◆ Latest changes
- ◆ NiagaraAX supported mobile operating systems
- ◆ Mobile apps prerequisites
- ◆ Licensing mobile apps
- ◆ Installing NiagaraAX mobile apps
- ◆ Common Mobile interface controls and indicators

In NiagaraAX-3.7 and 3.8, mobile applications (apps) are supported and included in the NiagaraAX Framework. These apps and the supporting NiagaraAX environment are designed to work with modern browsers on a personal computer, a mobile phone, and other hand-held devices.

A NiagaraAX app is similar to a NiagaraAX “service” except that apps bring more capabilities to the Framework. With the current popularity of apps and handheld devices in the market, most users understand the app concept and are comfortable using apps. In NiagaraAX, the app provides a focus or center point for a specific set of tasks and functions that are grouped together to address a single objective. The app may use other services and drivers, but the app itself is clearly identified by the common set of functions that it performs or views that it provides for doing a particular job.

Figure 1 NiagaraAX Mobile apps



The following apps are included in the NiagaraAX-3.7-3.8 mobile module and are available from the mobile palette:

- **Mobile Property Sheet**

This app provides a mobile interface for NiagaraAX station property sheet views.

- **Mobile Alarm Console**

This app provides a mobile interface for the NiagaraAx Alarm Console.

- **Mobile History**

This app provides a mobile interface for NiagaraAX History displays.

- **Mobile Schedule**

This app provides a mobile interface for NiagaraAX Scheduling displays.

- **Mobile Px**

This app provides an interface that allows System Integrators to build their own Mobile Px views.

Latest changes

If you are already familiar with the mobile features in prior releases of Workbench, the following summary of changes may be helpful in using subsequent versions.

New in Niagara 4.6

IMPORTANT: In Niagara 4.6 and later, legacy Niagara Mobile technology is deprecated. Although it continues to be supported and works as expected, you should not use it for any new projects. Instead, use the mobile-friendly features of the HTML5 Hx Profile which provides a significantly improved mobile user experience.

The following changes are implemented in Niagara 4.6 and later.

- Mobile friendly Px improvements adapt the display of Px graphics on various devices (PC, cellphone, tablet, etc.).
- HxPx enhancements provide consistency of views and graphics across Workbench and Mobile. These Px improvements enable you to create graphics once for multiple environments..
 - Flexible graphics and legacy Mobile Px views look and function the same in an **HxPx View** as they do in Workbench.
- The HTML5 Hx Profile provides numerous usability improvements. These mobile-friendly features enhance the user experience. So much so, that this is now the designated Default Web Profile as well as the default Mobile Web Profile. For more details, see "HTML5 Hx Profile" in the *Niagara 4 Graphics Guide*.

Changes in NiagaraAX-3.8

- Starting in NiagaraAX-3.8, Workbench includes SVG rendering capability. With the svgBatik module installed, SVG images can be used in your Px pages just about anywhere a GIF, PNG, or JPG can. An SVG image can be scaled up to larger sizes without degrading (becoming pixelated) as PNG images do. Also, large images scale down in size while retaining image quality and legibility. In order to provide this scaling functionality, a new widget called a Picture has been added to the **bajau** palette.
- PxEditor includes zooming capability. You can zoom in and out on Px pages while working in edit mode. Zoom changes the magnification of the page bringing it in closer for detailed work and zooming out to see the whole page.
- A number of graphical widgets have been added to the kitPxGraphics module. New widgets in the kitPxGraphics palette include various animated, static, and high-detail graphics in multiple sizes.

For more details, see About zoom capability in Px Editor, and About Widgets.

NiagaraAX supported mobile operating systems

These mobile apps have the potential to run on a variety of operating systems, including: iOS (iPhone), Blackberry, and Android operating systems.

Targeted operating systems include:

- Android default browser
 - Android phone 2.3 or later
 - Processor: 1 gigahertz or greater
- Blackberry
 - Blackberry Playbook OS 2.0.0.7971
- iOS Mobile Safari
 - iPhone 4 or later
 - iPad 1 or later

Mobile apps prerequisites

Each app requires the same set of prerequisites.

To install and configure any Mobile app for a hand-held device the following prerequisites are required. If any of these is not fulfilled, instead of displaying as `{disabled}`, the app simply does not appear on the desktop.

- **NiagaraAX-3.7 or 3.8 Framework**

The Niagara Framework includes Workbench, which is required to configure mobile apps.

- **Apps container (if not already present)**



The **Apps** container should already be located in the mobile module. If your station does not have one, install the **Apps** container from the mobile palette by dragging it onto the **Config** node in the Nav tree.

- **A license for each app**

Each of the apps documented in this guide is already licensed by default. To verify the license, check that the `propsheet="true"` attribute is in the mobile feature line of your license. For details, see [Licensing mobile apps, page 9](#).

- **End user security**

The ultimate hand-held user (as defined in the User Service, Permissions) must have at least a "read" level access for the app or the app does not appear in the browser when the user logs in.

Licensing mobile apps

Mobile apps are provided as a standard feature with NiagaraAX-3.7 and later, requiring no additional licensing.

By default each app is enabled individually through the license file. Individual apps can be disabled through the license file using the "mobile" feature line and attribute settings. An attribute value of `"true"` indicates that the app is enabled.

Figure 2 License file with mobile licensing included

```
<feature name="mcquay" expiration="2013-03-31" schedule.limit="none"
point.limit="none" history.limit="none" device.limit="none" parts="T-ENG-AX-DEMO"/>
<feature name="milestoneVideo" expiration="2013-03-31" schedule.limit="none"
foxStream.limit="4" camera.limit="4" point.limit="none" history.limit="none"
device.limit="4" parts="T-ENG-AX-DEMO"/>
<feature name="mobile" expiration="2013-03-31" history="true" schedule="true"
alarm="true" session.limit="none" px="true" propsheet="true"
parts="T-ENG-AX-DEMO"/>
<feature name="modbusAsync" expiration="2013-03-31" schedule.limit="none"
point.limit="none" history.limit="none" device.limit="none" parts="T-ENG-AX-DEMO"/>
<feature name="modbusSlave" expiration="2013-03-31" schedule limit="none"
```

If an app is not enabled ("false"), it does not appear on the associated device.

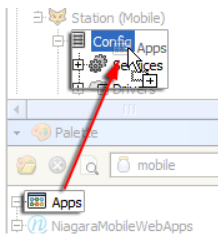
Installing NiagaraAX mobile apps

The following procedures describe how to install and configure typical mobile apps in a NiagaraAX station.

Install a NiagaraAX Mobile app

In NiagaraAX-3.7 and AX-3.8, new stations come with the **Apps** node already added. There is no need to add a second Apps component if one is already present in the station.

- Step 1** From NiagaraAX Workbench, open the **mobile** palette and drag the **Apps** component directly on to the **Config** node in the Nav tree, as shown here:

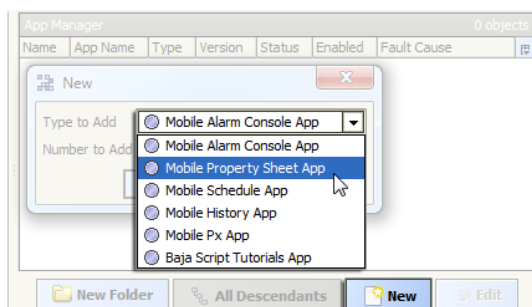


- Step 2** To display the **App Manager** view in Workbench, double-click the **Apps** node in the Nav tree.

- Step 3** In the **App Manager** view, click the **New** button to display the **New** dialog box.

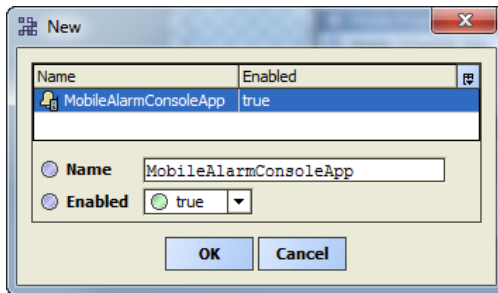
NOTE: You can add a group of apps if they are contained in an AppFolder, such as the NiagaraMobileWebApps folder simply by dragging and dropping the AppFolder onto the Apps node or onto the **App Manager** view.

- Step 4** In the **New** dialog box, choose the desired App from the **Type to Add** option field and click **OK**.

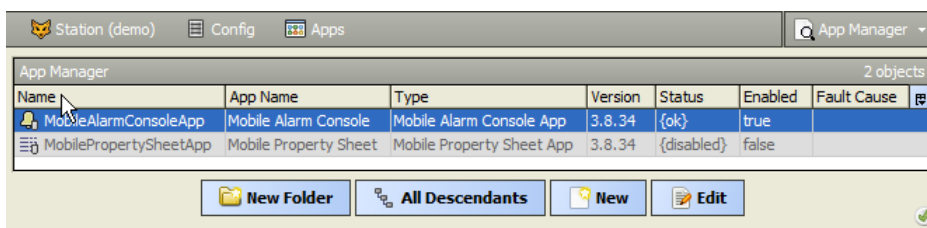


A second New dialog box appears.

- Step 5 In this **New** dialog box, edit the name and enable the app, as desired, and click **OK** to complete the installation.



All installed apps display in the **App Manager** view with information identifying the app and its status (enabled or disabled), as shown below.



There is also a **BajaScriptTutorialsApp** that can only be added using the **New** button in the **App Manager** view. This is intended to be used in a demo station for demonstrating, developing, and testing applications using bajascript.

Enable or disable a Mobile app

The NiagaraAX mobile apps come standard with NiagaraAX-3.8 and later versions. To configure the apps for a mobile device each needs to be enabled.

- Step 1 To change the Enabled status of an app, in NiagaraAX Workbench double-click on the app in the Nav tree.

The app property sheet appears.

- Step 2 In the property sheet, set the **Enabled** status to the desired state and click **Save**:

- `true` enables the app.
- `false` disables the app.

Common Mobile interface controls and indicators

The NiagaraAX Mobile interface is comprised of mobile-friendly views that use common controls and indicators to make the NiagaraAX station features and assets available from mobile devices.

Mobile desktop view (App Desktop view)

Figure 3 App Desktop view — common controls and indicators



- **Back button**

This button is located on the left side of the header. When you click it, it displays the previous view. If you choose to hide the header, this button does not display.

- **Header**

This indicator contains the name, or display name, of the app as well as the **Back** button and the **Command** button. You can hide or show the header by using the **Show Header** property in the user profile settings. Hiding the header removes the **Back** and **Command** buttons.

- **Apps icons**

The icons below the header identify each of the apps that are available to the user.

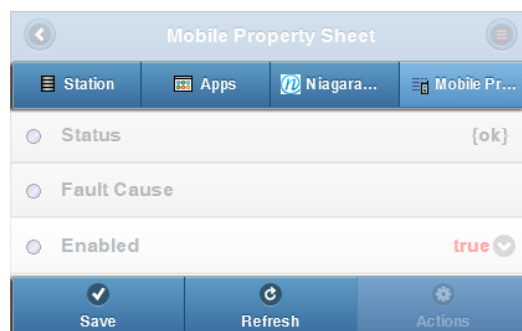
- **Command button**

When tapped, this button opens the **Commands** menu. If you choose to hide the header, this button does not display.

Navigation bar

This set of tabs displays below the Header on property sheet views to allow you to navigate through accessible areas of the station. The left to right arrangement reflects the top to bottom hierarchy of the station Nav tree.

Figure 4 Mobile nav bar and control buttons



- **Control buttons**

These buttons display horizontally across the bottom of the view. The buttons are available, as appropriate, for saving changes, refreshing the view or initiating actions in the view. In history table views, **Next** (>) and **Previous** (<) buttons appear to let you show the next or previous set of records.

- **Color indicators**

For any control, the following colors indicate:

- **Green**

Green text indicates a field that can be edited.

- **Red**

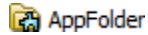
Red text and red graphics indicate an unsaved change.

For more details about the Mobile Property Sheet, see [Mobile Property Sheet app interface, page 20](#).

AppsFolder component

This optional component is available from the App palette.

Figure 5 AppsFolder icon



This component is a container for apps that you can use to help organize apps in the station. You can add more folders or more apps to this view as they are available. The default view for this component is the Apps Desktop view in a browser or the App Manager view in Workbench.

Chapter 2 Configuring a Mobile app

Topics covered in this chapter

- ◆ About Bajascript
- ◆ About Mobile user access
- ◆ Types of Mobile components
- ◆ Common Mobile App properties
- ◆ About the Mobile Property Sheet app
- ◆ About the Alarm Console App
- ◆ About the Mobile History app
- ◆ About the Mobile Schedule app
- ◆ About the Mobile Px app

The Default Mobile Web profile includes properties that let you choose among various Mobile themes as well as choose to display or hide certain elements of the profile. Themes affect the colors and graphic shapes, but do not change the functionality of the apps themselves. The default elements of the interface are included in all themes.

Developers can modify themes or create new themes using the jQuery mobile framework website (<http://jquerymobile.com>).

About Bajascript

Bajascript is a data API for accessing (and manipulating) NiagaraAX data.

The bajascript API enables access to NiagaraAX data, so that you can build Web apps around it. The browser already defines the whole user interface piece. Bajascript adds the NiagaraAX data part. Bajascript works with the BOX service, which is added to your station automatically when you install one or more apps.

There is a **BajaScriptTutorialsApp** that can only be added using the New button in the App Manager view. This app is intended to be used in a demo station for purposes of demonstrating, developing, and testing applications using bajascript.

NOTE: A bajascript demo app can also be accessed (with appropriate credentials) at bajascript.com. For more details, see www.bajascript.com.

About Mobile user access

The combined use of Profile, Nav File, Px views, and standard NiagaraAX security determine a user's access to a station and navigation experience using Mobile apps.

• Profiles

When a user logs in to a station from a browser on a mobile device or a PC, the station detects the type of device and uses the appropriate profile, which is assigned to the User Name:

- The Default Web Profile is used when logging in from a PC.
- The Mobile Web Profile is used when logging in from a handheld device.

For example, if connecting from a browser running on a laptop PC, the **Nav File** and **Default Web Profile** affect: the initial page display, navigation, and display options. If logging in from a mobile phone, the **Mobile Nav File** and **Mobile Web Profile** affect the initial view and navigation options.

NOTE:

The Mobile Web Profile has a sub property named Mobile Nav File. While the Nav file and Default Web Profile are both actual properties of the user account.

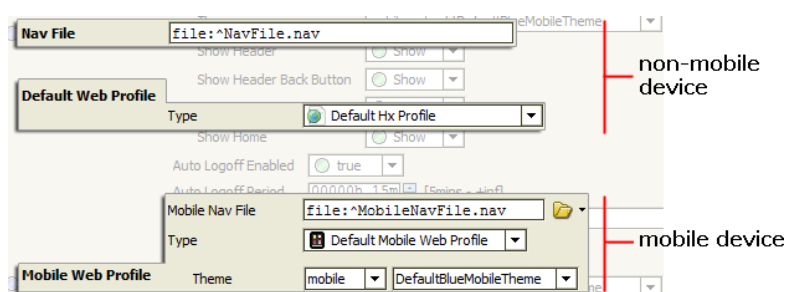
As a convenience, the Default Mobile Web profile is also available from the Default Web Profile property options for working with mobile views on your PC browser.

The Default Mobile Web profile includes three mobile themes to choose from. Each theme has a set of configurable display options:

- Show Header
- Show Header Back Button
- Show Select Views
- Show Home
- Show Logout (new in AX-3.8)

These properties are described in [Common Mobile interface controls and indicators, page 11](#)

Figure 6 Choose Nav File and profile for mobile and non-mobile devices



• Nav files

The Mobile Nav File is used only to designate a user's landing page for login (and Home page designation). The user Nav File is used to configure navigation links. Typically, you create the Nav file and then assign the file to a user. The Nav file assigned to the user's profile determines the initial targeted component for that user. The actual view may depend on user permissions and category settings for the views assigned to that component.

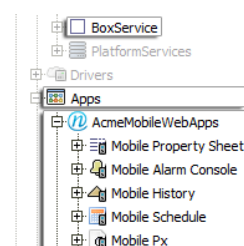
• Security

User access to individual mobile views is controlled by the standard NiagaraAX security model of: Users, Permissions, and Categories. So, by default, you must have at least "operator read" permission on an app, or a specific view in that app to see the view. However, if you are designing an app and you want to specify different views for different users, you can use a combination of alternate Px views and permissions to designate the view that a particular user sees. For more details about NiagaraAX security features, see the "Security" chapter in the *NiagaraAX User Guide*.

Types of Mobile components

With Mobile apps installed the following components are added to a station.

Figure 7 Mobile components in a station Nav tree



• Apps

This container is provided by default in a new NiagaraAX-3.7 or AX-3.8 station. It is a necessary container for using Mobile apps.

There is also an “App Folder” which is a BAppFolder component that is available in the **app** palette. When using the App Manager view and clicking the New Folder button, a folder of the type, “App Folder”, is added to the App Manager view. Note, the App Folder is used in the “Apps” container or other “App Folders” to organize apps in a folder hierarchy.

- **Box service**

Box is the protocol that Bajascript uses. This service is automatically added when you add one or more Mobile apps to the Apps folder.

- **Mobile app set**

This is the default set of Mobile apps provided in NiagaraAX-3.8.

- Mobile Property Sheet
- Mobile Alarm Console
- Mobile History
- Mobile Schedule
- Mobile Px

For details, see [Components in the Mobile module, page 44](#).

About BOX caching

Starting in AX-3.8, the speed at which BOX generates Type and Contract information for BajaScript has been improved.

A new Property called **Server Cache Configuration** has been added to the BOX Service. You can configure the server cache from **Low** to **High** to decrease or increase performance speed:

- **Low** (no caching)
- **Medium** (cache the bare minimum)
- **High** (cache everything)

The default setting is Medium. Medium does some caching, it does not cache everything. This is the safest option for a JACE. Setting the cache to Low removes all caching. Setting it to High causes everything to be cached, which consumes more memory.

NOTE: Increasing the BOX server cache can significantly improve server performance when computing Type and Contract information on a heavily loaded JACE 6s or JACE 2s.

BajaScript web applications work faster due to optimizations made in the BOX Server. BajaScript also caches Type and Contract information at the client. The result is that web pages that use BajaScript load faster on subsequent visits.

Increasing BOX cache to improve performance

Prerequisites:

- Target platform is running AX-3.8
- Niagara Mobile is installed

Increasing the BOX server cache can significantly improve server performance when computing Type and Contract information on heavily loaded JACE 6s or JACE 2s.

NOTE: Increasing the server cache consumes more memory.

Increase BOX server cache

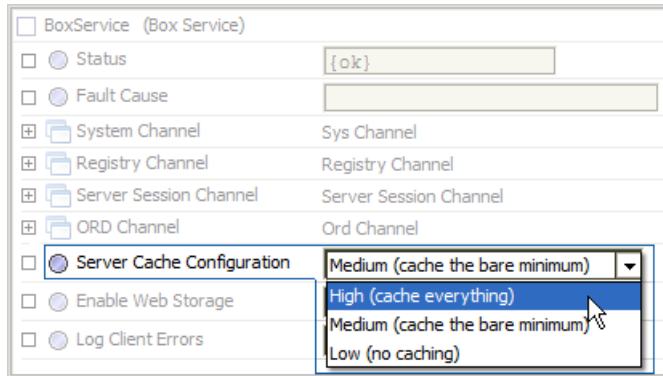
Step 1 Connect to the station.

Step 2 Expand the **Services** node.

Step 3 Double-click **BoxService** to open the Property Sheet view.

Step 4 In the **Server Cache Configuration** slot, click the select icon (▾) to the right of the default setting (Medium) to see options.

Step 5 Click **High**, as shown.



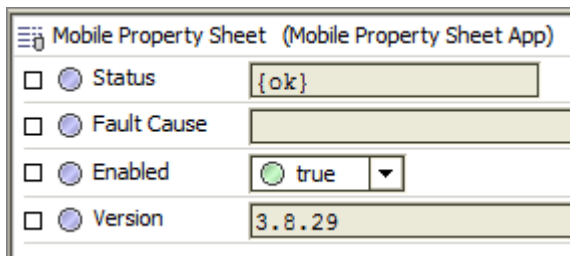
Step 6 Click **Save**.

Increasing the BOX cache improves performance. Screen refresh/redraw is noticeably faster.

Common Mobile App properties

Different mobile apps share many of the same properties.

Figure 8 NiagaraAX Mobile Apps properties



The default set of mobile apps have the following properties, as seen from their Property Sheet view:

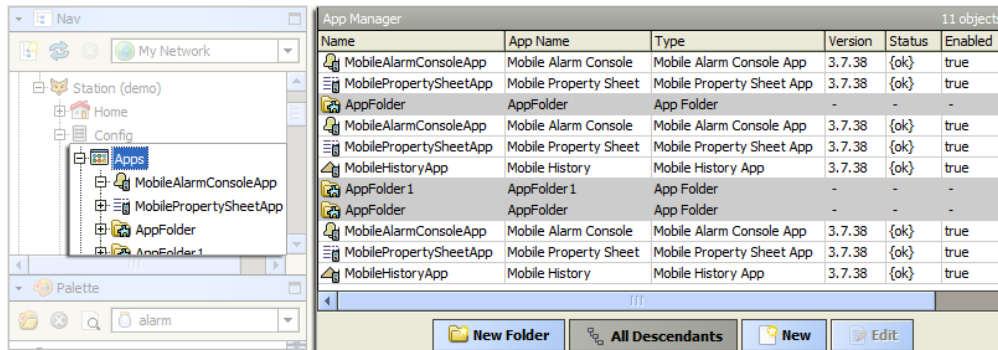
- **Status**
This is a standard NiagaraAX status indicator. The read-only field indicates the status of the associated app: {ok}, {down}, {disabled}, or other.
- **Fault Cause**
This is a standard NiagaraAX fault display. This read-only field indicates the possible cause of a fault, if app status is {down}, {disabled}.
- **Enabled**
This is a standard NiagaraAX property for enabling and disabling components. It allows you to enable (set to true) or disable (set to false) the app.
- **Version**
This read-only field displays the app version.

About the Apps container

The Apps container provides a central location for all designated mobile apps including the provided NiagaraMobileApps.

In NiagaraAX-3.7 and AX-3.8 new stations come with this **Apps** node already installed directly under the station **Config** node. You can install individual apps from the Apps container default view, the App Manager view (using the **New** button), or you can drag the **NiagaraMobileApps** container from the mobile palette directly onto the **Apps** folder. You can also drag individual apps, or multiple selected apps, from the mobile palette to the App Manager view.

Figure 9 Apps container with the default App Manager view



Apps may use other services and drivers, but the idea is that an App serves as a central focus for something that provides a specific function or set of views for doing something in the Niagara Framework.

About the Mobile Property Sheet app

The Mobile Property Sheet app allows you to navigate through the entire component hierarchy of your NiagaraAX station. You can use this app to view and edit points as well as initiate actions.

Figure 10 Mobile Property Sheet app in Workbench and in the Mobile Desktop view

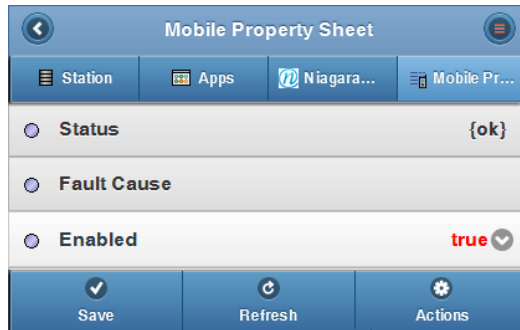



NOTE: Not all field editors are supported in the NiagaraAX-3.8 release of Niagara Mobile, but the primary ones are: fields for points and point extensions. Niagara Mobile uses a public API that is available so that developers can add field editors in the supplied apps. Future releases may provide more field editors for the Mobile app. For more details, refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#)

Mobile Property Sheet app interface

NOTE: Changes to the Workbench Property Sheet update almost immediately to the mobile view. If you delete an element in your station from Workbench, the mobile view may display it with a “removed” annotation until the next view refresh.

Figure 11 Example of the Mobile Property Sheet view



The app default (Property Sheet) view appears when you select the Mobile Property Sheet icon () from the Mobile Desktop view.

NOTE: If you disable your Property Sheet app from this view and navigate away from it in your mobile device you may not be able to get back to it without using a Workbench connection to enable it.

In addition to the standard NiagaraAX Mobile controls and indicators (See [Common Mobile interface controls and indicators, page 11](#)), the Mobile Property Sheet app uses the following:

- **Save**
The **Save** button is available whenever a value is changed. Changed values that are not saved are displayed in red.
- **Refresh**
The **Refresh** button is typically displayed for all pages. When you tap Refresh, the software prompts you to save any unsaved changes on the page.
- **Actions**
The **Actions** button is displayed on views that have actions available.

About status coloring in Mobile property sheet view

Starting in AX-3.8, status color enhancements have been added to the Niagara Mobile Property Sheet view to increase usability.

The Property Sheet view on mobile devices uses the standard Niagara status colors to easily convey status. The use of status colors gives you a visual indication about the current state of your Station.

If a component in the property sheet has a status (typically, device networks, devices and points), the view shows the respective status color for the slot on the property sheet. This allows you to instantly see if components are in alarm or disabled, etc.

Niagara uses the following set of colors to convey status:

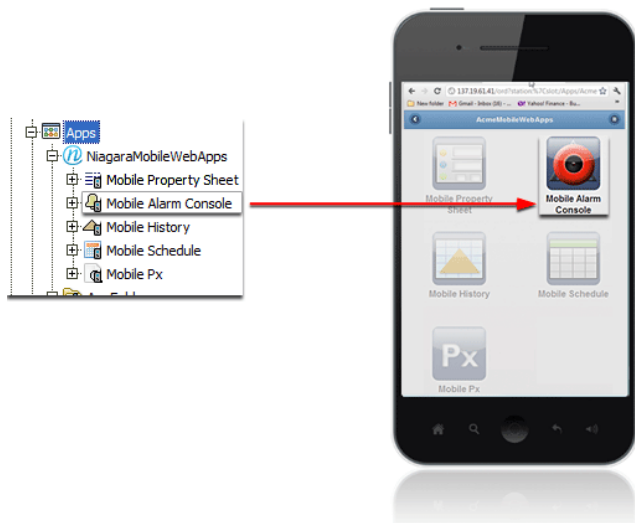
- gray = disabled
- orange = fault
- yellow = down
- red = alarm
- tan = stale

- purple = overridden
- no color = null

About the Alarm Console App

The Mobile Alarm Console app provides a mobile view of a standard NiagaraAX station Alarm Console Recipient.

Figure 12 Alarm Console app in Workbench (left) and the Mobile Desktop view (right)

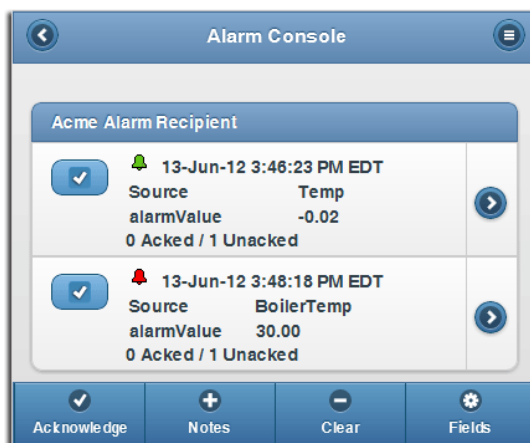


The app allows you to view the alarms for each of your alarm recipients with all detail fields of individual alarms. In addition, you can acknowledge alarms and add notes, as you would with the standard NiagaraAX alarm console.

Mobile Alarm Console app interface

The Mobile Alarm Console view appears when you select the Mobile Alarm Console icon (📱) from the Mobile Desktop view or when you select it from a Mobile Property Sheet view (using the Mobile Property Sheet app).

Figure 13 Example of the Alarm Console view



Like the standard NiagaraAX alarm console, the Mobile Alarm Console app manages alarms on a per-point basis. Each row in the alarm console is the most recent alarm from a point. To view all the current alarms or

to get more details about a particular alarm from that point, tap the Details button (🔍) at the right side of the alarm row.

Figure 14 Example of the Open Alarm Sources view



In addition to the standard NiagaraAX Mobile controls and indicators (see [Common Mobile interface controls and indicators, page 11](#)), the Mobile Alarm Console app uses the following:

- **Select All**

Tap this check box to select all alarms that are currently loaded on the page.

NOTE: This option selects any alarms that are not visible but loaded on the page (for example, alarms 21–30 in the range above. You can scroll down to see them). Alarms that are on **Previous** or **Next** pages are not selected.

- **Select**

Tap this check box to select the associated alarm.

- **Details**

This button displays either the Alarm Details view or, if you are looking at the most recent of multiple alarms, it displays a listing of all the alarms from that point. If you have multiple pages of alarms listed, use the page navigation buttons (**Previous** and **Next**) to move among the pages.

- **Page Nav**

The page navigation buttons are active as needed to display consecutive alarm records or alarm data. Use the **Previous** and **Next** buttons to navigate through consecutive pages or records.

- **Commands**

These buttons display on views that have one or more alarm Select boxes checked.

- **Acknowledge**

Tap this button to acknowledge all selected alarms.

- **Notes**

Tap this button to open the **Notes** dialog box to add notes to selected alarms.

- **Clear**

Tap this button to clear the selected alarms.

– **Fields**

Tap this button to open the **Choose Fields** dialog box. Use this dialog box to add or hide alarm data in alarm rows using the alarm data facets in the same way that you use them in the standard NiagaraAX alarm console.

NOTE: The NiagaraAX-3.7 and AX-3.8 versions of the Mobile Alarm Console do not play sounds.

Alarm source data

Alarm source data is translated for mobile presentation:

- BFormat text is translated from the alarm source.
- Custom hyperlinks
- Custom icons

Local storage

This is an HTML 5 feature that provides local device storage for user-designated preferences. For example, you can configure your alarm console columns to display only the data you want and those options are cached on the device for use the next time you log in on that device.

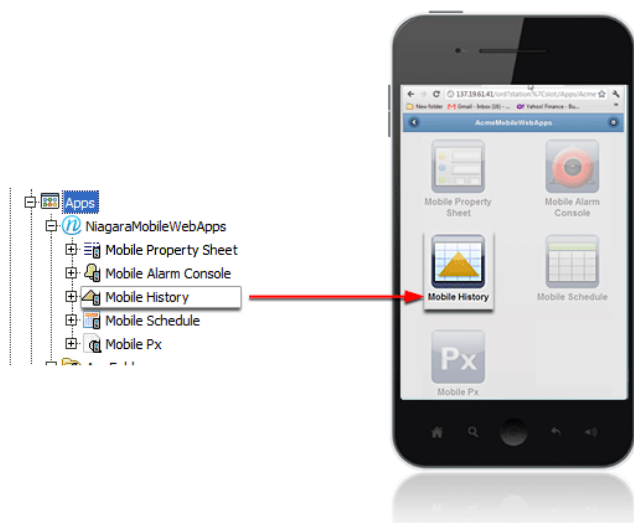
About the Mobile History app

The Mobile History app provides mobile views of history data on both chart and table displays.

You navigate to the Mobile History app through the Desktop view or via Px page links. This app supports the following standard NiagaraAX history features:

- Delta mode reporting for charts and table views
- Relative and specific time ranges
- Zooming (for charts)
- Column selection (for tables)

Figure 15 Mobile History app in Workbench (left) and the Mobile Desktop view (right)



Mobile History app interface

The Mobile History listings view appears when you select the Mobile History Console icon (📄) from the Mobile Desktop view or from a hyperlink in Mobile Px.

The first view is a listing of histories directly under the station. The station name appears in the header area.

Figure 16 Example of a Mobile History view with no grouping



If you are using the **History Grouping** folder (under the **History Services** node), the histories are grouped according to the organization of your history grouping folders.

Figure 17 Example comparing grouped history views in the Workbench Nav tree (left) with the Mobile app view (right)

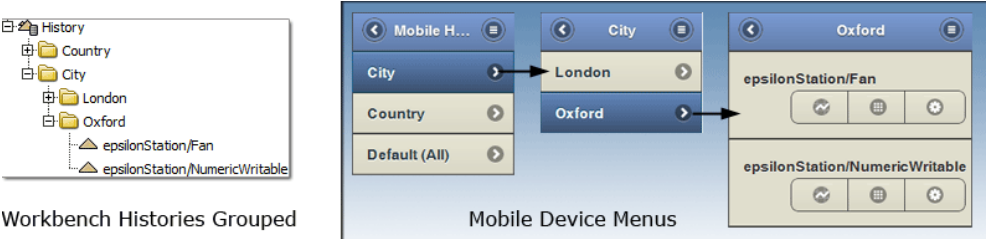
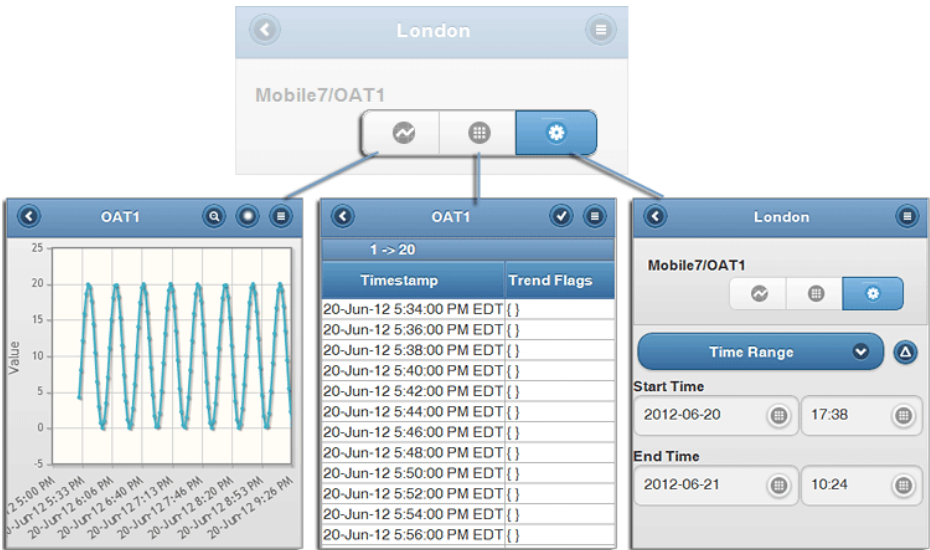


Figure 18 Mobile History Control display a Chart, Table or Time Range dialog box



In addition to the standard Niagara Mobile controls and indicators (see [Common Mobile interface controls and indicators, page 11](#)), the Mobile History Console app uses the following:

- **Chart (Mobile 7/OAT1 in the example above)**

Tap this control to create a chart of the active history record.

In addition to the standard mobile controls (see [Common Mobile interface controls and indicators, page 11](#)), chart controls include the following:

- **Highlight**

Use this control to toggle between highlight enabled (🔍) and highlight disabled (🚫). When highlight is enabled, you can touch and drag a defining area of the chart to enlarge (or “zoom in”) it. When the control is toggled to the disabled setting, zoom-highlighting is disabled. You can disable highlighting when you want to touch the screen without zooming.

- **Reset Zoom**

Tap the reset zoom control (🔄) to return a zoomed-in display to normal size (no zoom).

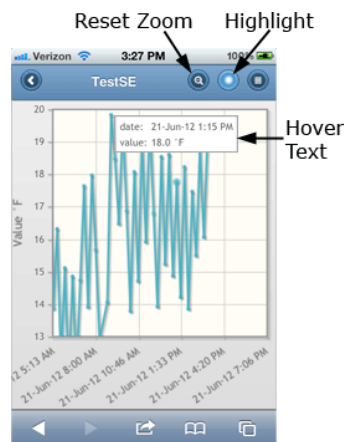
- **Delta**

Tap the delta control (⬆️) to create charts with delta reporting. This option is useful to display value changes (delta) in your report. This control is located in the **Time Range** dialog box.

- **Hover Text**

You can display X and Y axis data from a point on the chart display by touching the point. The point data displays as shown in the following illustration.

Figure 19 Mobile History Chart Zoom and Hover Text feature



- **Table**

Tap this control to display a table of the data for the selected history. In addition to the standard table controls (see [Common Mobile interface controls and indicators, page 11](#)) you can use the Time Range dialog box to select a unique range or a preset range of data. Additional table controls include the following:

- **Column sorting**

Tap the heading of the column to sort on.

- **Column selection**

Show or hide table columns using the columns chooser. Click the (☑️) to show the column chooser.

- **Next > Prev < buttons**

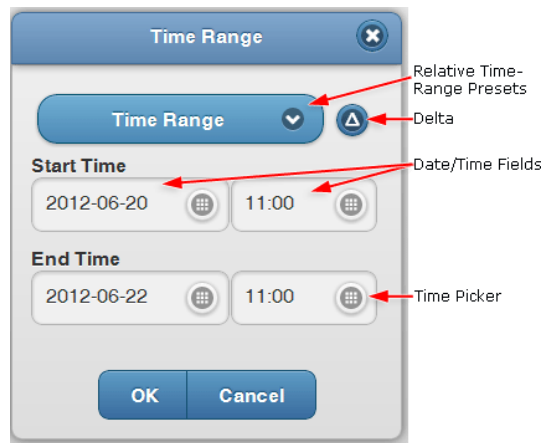
These buttons appear in the control bar when there are more records in the table than displayed on one view. Use the buttons to advance to the next or previous set of records.

- **Time Range dialog box**



This dialog box is used for setting a time range of records to display in your table or chart. You can type directly in the date and time fields or use the date and time chooser tool by clicking the **Time Picker** button.

Figure 20 Time Range dialog box



- **Time Range**
Use this button to choose from a list of relative time range options.
- **Delta**
Tap the delta control (Δ) to create charts with delta reporting. This option is useful to display value changes (delta) in your report..
- **Date/Time Fields**
Type directly in these fields to designate a custom time range.
- **Time Picker**
Tap this button to open the **Time Picker** tool, a convenient way to enter time data.

About the Mobile Schedule app

The Mobile Schedule app supports mobile viewing and touch-based editing of standard NiagaraAX schedule components and views. On a mobile touchscreen, tap the Mobile Schedule icon on the Mobile Desktop to open the Mobile Schedule app.

Figure 21 Mobile Schedule app in Workbench (left) and the Mobile Desktop view (right)



NOTE: With the Mobile Schedule app enabled you can also use a Mobile Property Sheet view or a link from a Mobile Px page view to navigate to individual schedules in your station from a mobile device.

All the standard NiagaraAX schedule views are available using the Mobile Schedule app, including:

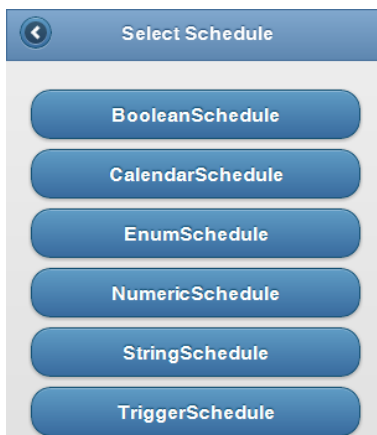
- Boolean Schedule
- Calendar Schedule
- Enum Schedule
- Numeric Schedule
- String Schedule
- Trigger Schedule

Mobile Schedule app interface

The Mobile Select Schedule view appears when you select the Mobile Schedule Console icon (📱) from the Mobile Desktop view or from a hyperlink in Mobile Px.

The default view for the Mobile Schedule is a listing of all the station schedules with the station name in the header area.

Figure 22 Schedule Selector view from the Mobile Schedule app



In this view, a separate link is displayed for each schedule in the station.

NOTE: The Mobile Schedule app shows schedules from all locations in the station. Schedules do not have to be located in the same directory.

Within each type of schedule the standard NiagaraAX schedule view types are available. Within these views, in addition to the standard Niagara Mobile controls and indicators (see [Common Mobile interface controls and indicators, page 11](#)), the Mobile Schedule app has some unique controls for each of the views as necessary:

- **Weekly Schedule tab**

In addition to the mobile versions of all the standard NiagaraAX Schedule controls, this view provides color display for boolean schedules as shown below.

Figure 23 Mobile Schedule app Weekly Schedule tab



Touch the calendar in this view to open the **Edit Day** view where you can **Create**, **Edit**, **Delete**, and assign **Options** to the calendar using appropriate touch controls.

- **Special Events**

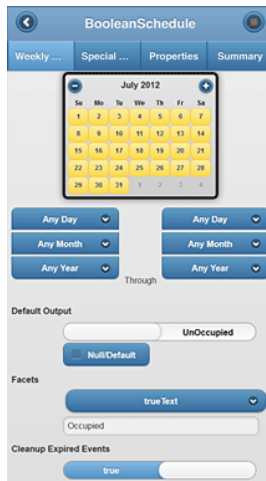
Figure 24 Mobile Schedule app Special Events tab



This view allows you to use touch controls to add Special Events, such as holidays, to your schedule. Use the controls at the bottom of the view to **Add**, **Edit**, **Rename**, **Delete**, and manage the priority of the calendar's Special Events.

- **Properties view**

Figure 25 Mobile Schedule app Properties tab



This view displays all the standard properties of the selected schedule. You can use the touch controls to change the date, **Default Output**, **Facets**, and other options as appropriate.

- **Summary view**

This view displays a single-month, touch-sensitive calendar that is color-coded to indicate days that have scheduled events. Events for the selected (single) day display in a list below the calendar. You can change the active day display by touching the desired day on the calendar.

Figure 26 Example of a Mobile Schedule app Summary tab



The calendar has the following touch controls:

- **Event day**

Touch a day icon to display events for that day. In the example above (that uses the active Mobile theme):

- ◆ Yellow-colored days (3) have an event scheduled.
- ◆ Blue-colored days (2) do not have an event scheduled.

- **Month controls**

To change months, use the plus (+) and minus (–) buttons in the top corners of the active calendar.

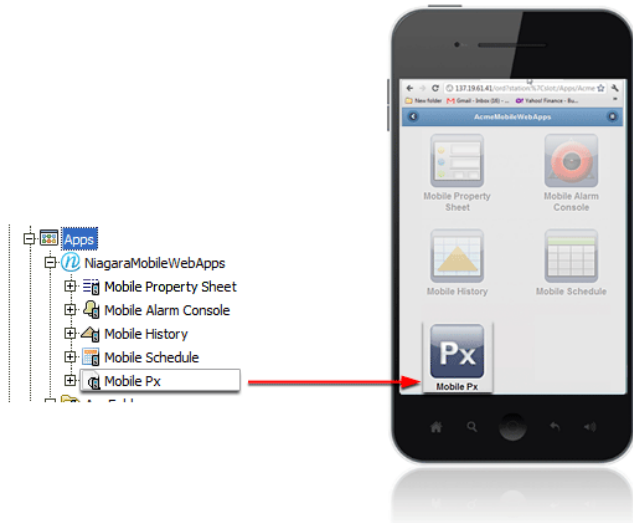
NOTE: Display controls appear differently depending on the mobile theme that is active for the user.

About the Mobile Px app

The Mobile Px app allows you to create your own app views in the Px Editor without having to use any software programming language.

The ability to make app views includes using the standard NiagaraAX Px Editor to design navigation and custom controls that display and work well in a handheld device. Essentially, think of this app as an “app builder.”

Figure 27 Mobile Px app in Workbench (left) and the Mobile Desktop view (right)



Mobile Px provides a way to present live data and field editors in *expandable lists* that are designed to work well with mobile devices.

If using NiagaraAX-3.7 to create views with the Mobile Px app, the display is not an exact representation of what you have in the Px Editor or what you would expect to see when you are displaying views created in Hx. Instead, for Mobile Px views in AX-3.7, the Px Editor works as a tool to create views that the Mobile Px app interprets and presents using such mobile-friendly features as: rounded buttons, “liquid” layout, dynamic views, and others.

Starting with AX-3.8, Niagara provides support for absolute-positioning for graphics in Mobile Px. Absolute positioning is achieved by adding a CanvasPane to your Mobile Px page. The CanvasPane scales to fit different sized displays. Absolutely positioned widgets scale proportionately with the parent CanvasPane. For more details, see [About absolute positioning in Mobile Px, page 32](#).

Many of the standard Px graphic widget and property options are available for configuring your Px page, including:

- **Buttons**

Add standard buttons (Action Button, Hyperlink Button, Save Button) from the kitPx palette. these buttons work the same way, but they appear rounded in mobile views.

- **Labels and bound labels**

Add standard labels from the kitPx palette.

- **Field editors**

- **Graphic alignment**

- **Font family**

- **Font size and style**

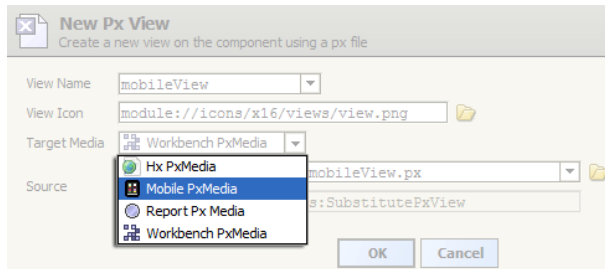
Refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#) for more details.

About Mobile Px views

Mobile Px views are different from standard Px views. To create a Mobile Px view you must select the **Mobile Px View** option from the option list in the **New Px View** dialog box.

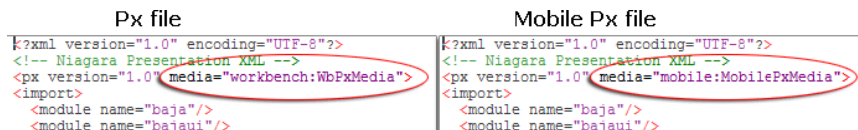
NOTE: If you do not assign Mobile PxMedia to your view, the view does not display or displays an error when opened from a mobile device.

Figure 28 Creating a Mobile Px view



Mobile Px view files have the media attribute set to `mobile:MobilePxMedia` as opposed to other media types, such as `workbench:WbPxMedia`, as shown in the following figure

Figure 29 MobilePx files use different media value



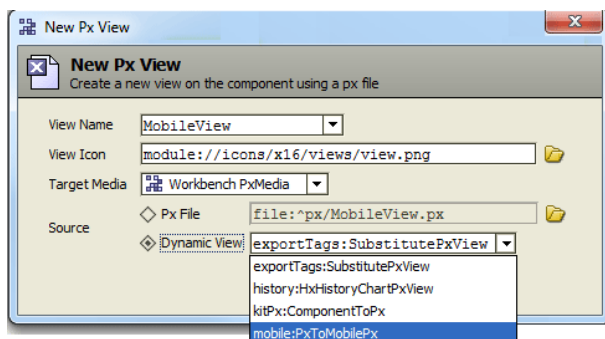
NOTE: You can animate any widget property. This is different from Hx because Mobile Px uses Bajascript and a completely new Px engine. In Hx, you have a virtual Hx page in the station and a copy of that page is rendered in the client (browser). In Mobile Px, everything is running in the browser, which uses Bajascript to run things and to take advantage of the processing power of the mobile device and its browser.

About dynamic Mobile Px views

Dynamic Mobile Px views provide a way to quickly *convert* an existing Workbench Px view into a Mobile Px view.

If you already have a Px view, you can create a dynamic Mobile Px view by selecting **Dynamic View** from the **New Px View** dialog box and choosing the `mobile:PxToMobilePx` option.

Figure 30 Creating a dynamic Mobile Px view



When you select OK, the conversion process creates a new Mobile Px view that includes all of the supported widgets from the source Px view and adds them to a BasicMobilePane in the new Mobile Px view. It requires a fair amount of editing to make this new Mobile Px view usable.

NOTE: When creating a dynamic Mobile Px view, the source Px page is left unchanged and a new Px page is created.

Another method for utilizing existing Px pages is to copy and paste the CanvasPane (with the scale property set to Fit Ratio) from your Px page into the BasicMobilePane in your Mobile Px page. For details, see [Converting existing Workbench graphics for mobile devices, page 35](#).

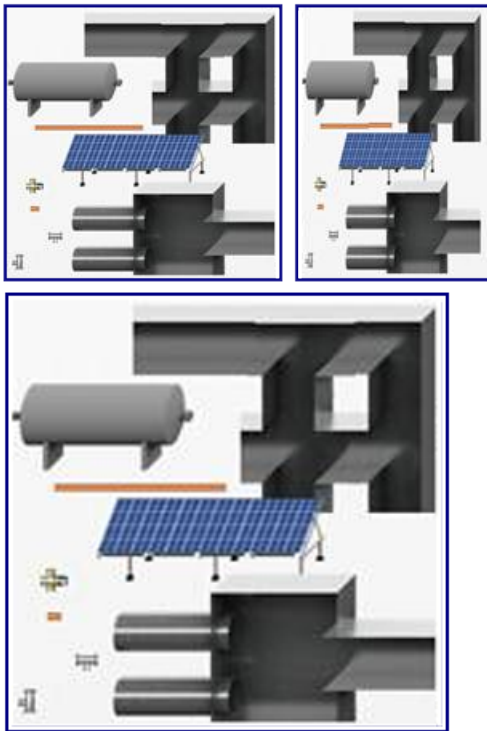
NOTE: Not all Px components support dynamic view conversion. Only supported components are represented on the new Dynamic Px page. Refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#) for more details.

About absolute positioning in Mobile Px

AX-3.8 provides support for absolute-positioning for widgets in Mobile Px. Absolute positioning is achieved by adding a CanvasPane to your Mobile Px page.

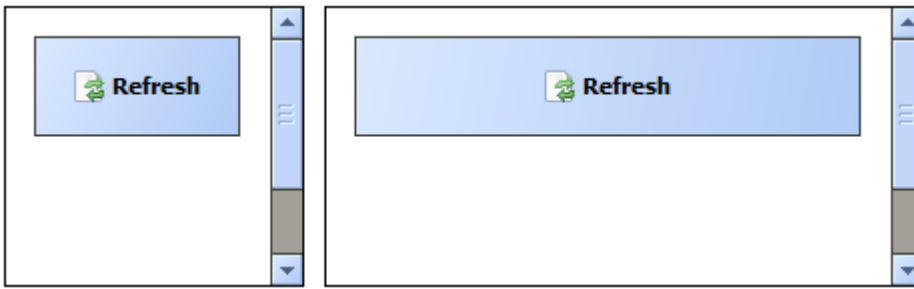
An “absolutely positioned” widget is one whose immediate parent is a CanvasPane. The CanvasPane has the ability to scale to fit different sized displays. Therefore, in a properly configured CanvasPane, every absolutely positioned widget scales proportionately *with* its CanvasPane parent. For example, shown below is a CanvasPane scaled to three different sizes. As the CanvasPane scales larger or smaller the individual widgets in the pane scale proportionately, maintaining their existing ratios.

Figure 31 Absolutely positioned widgets in CanvasPane scaled to three different sizes



In contrast, a widget whose immediate parent is a BasicMobilePane or a MobileGridPane is “fluidly positioned”. Fluidly positioned widgets do not scale (maintain proportions), instead the widgets resize to fit the screen. For example, a button widget in a BasicMobilePane stretches wider or narrower to fit its container while the size of the text label on the button does not change.

Figure 32 Fluidly positioned widget in BasicMobilePane shown in two sizes



CanvasPanes can be included in Mobile Px pages by adding them as child widgets of a MobileGridPane. Setting the CanvasPane's scale mode to `Fit Ratio`, causes the CanvasPane to adjust automatically, expanding or contracting, to properly fit the width of your mobile device. Also, setting the CanvasPane as the root widget of your Mobile Px page results in the entire page having a fixed width and height.

About Canvas Panes in Mobile Px

The addition of CanvasPane support in AX-3.8 Mobile Px provides a simple solution to transitioning your existing Workbench Px graphics to display on mobile devices. Using a CanvasPane with the scale property set to `Fit Ratio`, a complex graphic can be scaled down to fit on smaller screens, while maintaining the existing layout, legibility and functionality.

To incorporate a CanvasPane graphic in a mobile Px page, simply copy the CanvasPane from the existing Px page and paste it into the BasicMobilePane or a MobileGridPane on your mobile page.

Keep in mind that not all widgets will display correctly in Mobile Px. Basic widgets like Labels, Pictures and Buttons are all supported, but some Workbench-only custom widgets, such as `bajoui:PxIncludes` and `kitPx:AnalogMeter`, are not supported and will not display on a mobile page. For a complete list of supported widgets, see [Field editors, widgets, and bindings supported in Mobile Px](#), page 36.

Some editing will likely be necessary to get your existing Px pages looking perfect on mobile devices, but using a copy+paste+tweak approach is easier and faster than recreating mobile-specific versions of your graphics.

Adding a properly configured CanvasPane to a BasicMobilePane automatically scales child widgets based on the available screen resolution. However, this automatic scaling behavior may cause Px views to shrink to an undesirably small size when displayed on mobile devices that have fairly small screens.

Another option is to use a more typical Px view which has a ScrollPane as the root widget with a child CanvasPane configured for no scaling. Provided the media type for the Px view is configured for mobile, the Px view will be accessible when logged in using a mobile profile in a browser. In this case, the Px view does not automatically scale to fit the available screen resolution. Instead, the Px view renders based on the size of the canvas pane and you can use "one finger scrolling" to see all of the content.

NOTE: In many cases when mobile-type devices are logged in using an Hx profile in a browser, you must use "two finger scrolling".

About scaling in Mobile Px

Adding a CanvasPane to a BasicMobilePane and setting the scale mode to `Fit Ratio` ensures that your absolutely-positioned graphic automatically scales and resizes to fit phone and tablet screens.

Mobile Px CanvasPanes support the different scale modes offered in Workbench. If a CanvasPane is absolutely positioned (a child of another CanvasPane) then it supports all five scale modes: `None`, `Fit`, `Fit Ratio`, `Fit Width`, and `Fit Height`. While a CanvasPane that is fluidly positioned (a child of a BasicMobilePane) supports only two scale modes: `None` and `Fit Ratio`.

You can add CanvasPanes to cells in a MobileGridPane to scale them down even smaller. Widgets can be in each cell of the grid pane being scaled. A MobileGridPane supports up to five columns, but most graphics become cramped if your column count exceeds two.

Adding a CanvasPane to a ScrollPane (that is the root widget) with the CanvasPane's scale property is set to `None` can have a potentially undesirable effect of stretching the header in the browser. In the browser, the header will stretch to match the size of the canvas pane, which makes the commands button in the header only accessible after scrolling all the way to the right. This can be remedied by not using a ScrollPane as the root widget. Instead use a CanvasPane as the root widget with scaling set to `Fit Ratio`.

NOTE:

CanvasPane scaling is supported in later browser versions such as Chrome, Firefox, and Opera. Internet Explorer browser support is limited to version 10 and up.

Creating new, unscaled, absolutely positioned mobile Px pages

Starting with a typical Workbench view and later converting it to Mobile, you will get an unscaled, absolutely positioned graphic. If it exceeds the boundaries of your mobile device, it must be scrolled side to side to view the complete graphic.

Prerequisites:

- Target platform is running NiagaraAX 3.8
- Niagara Mobile is installed
- A recent HTML5 compliant browser (issued 2011 or later), for example: FireFox, Chrome, Opera, IE (v.10 or later) and Safari.
- Individual mobile apps licensed, as needed

Create a new unscaled, absolutely positioned Mobile Px page

- Step 1 Right-click target component in nav tree, select **View→New View**.
- Step 2 Leave target media as `Workbench PxMedia`. This will start your Px page with a ScrollPane and CanvasPane as the root widgets.
- Step 3 Switch to Property Sheet view and change the media on the Px View to `MobilePxMedia`. This will make your Px page accessible in Niagara Mobile
- Step 4 Design your Px page as you normally would.

NOTE: Not all widgets are supported in Mobile Px. For example, the `bajoui:PxIncludes` widget and `kitPx:AnalogMeter` widget are not supported. For a complete list of supported widgets, refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#)

The Workbench graphic pasted into a Mobile Px page is unscaled and so it must be viewed by scrolling horizontally.

Creating new, scaled, absolutely positioned Mobile Px pages

Starting with the Mobile media, you will get a fluidly positioned page layout. Adding a CanvasPane to this layout will cause it to automatically stretch and scale to fit the size of your mobile device.

Prerequisites:

- Target platform is running NiagaraAX 3.8
- Niagara Mobile is installed
- A recent HTML5 compliant browser (issued 2011 or later), for example: FireFox, Chrome, Opera, IE (v.10 or later) and Safari.
- Individual mobile apps licensed, as needed

NOTE:

CanvasPane scaling is not supported in Internet Explorer 9 and earlier.

Create a new scaled, absolutely positioned Mobile Px page

- Step 1 Right-click target component in nav tree, select **View→New View**
- Step 2 Change target media to `Mobile PxMedia`. This will start your Px page with a **BasicMobilePane** as the root widget.
- Step 3 Drag a **CanvasPane** from the `bajoui` or `mobile` palette into the **BasicMobilePane**.
You can also add a **MobileGridPane** (from the `mobile` palette) to the **BasicMobilePane** and add **CanvasPanes** to that. Be careful not to get the page too cramped, horizontally.
- Step 4 Design your Px graphic inside the pasted **CanvasPane**.
NOTE: Not all widgets are supported in Mobile Px. For example, the `bajoui:PxIncludes` widget and `kitPx:AnalogMeter` widget are not supported. For a complete list of supported widgets, refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#)
- Step 5 Set the **CanvasPane** `scaleMode` property to `Fit Ratio`.

NOTE:

In a fluid layout, any scale modes other than `None` or `Fit Ratio` will be ignored.

The Mobile Px page automatically stretches and scales to fit the size of the mobile device.

Converting existing Workbench graphics for mobile devices

Reuse existing Workbench Px graphics by copying the **CanvasPane** from an existing Workbench Px page and pasting it into the **BasicMobilePane** in a Mobile Px page.

Prerequisites:

- Target platform is running NiagaraAX 3.8
- Niagara Mobile is installed
- A recent HTML5 compliant browser (issued 2011 or later), for example: FireFox, Chrome, Opera, IE (v.10 or later) and Safari.
- Individual mobile apps licensed, as needed

Convert an existing Workbench Px graphic for display on mobile devices

- Step 1 In the nav tree under your station, right-click **Config→Drivers** and select **New→Folder** from the menu.
- Step 2 Right-click the new folder, select **Views→New View**
- Step 3 Change the Target Media option to `Mobile PxMedia` and enter a name for the Px File.
The new Mobile Px page opens with a **BasicMobilePane** as the root widget.
- Step 4 Open your existing Workbench Px graphic.
- Step 5 Copy the **CanvasPane** container.
- Step 6 In your Mobile Px page, select the **BasicMobilePane** and paste the copied canvas pane inside it.
NOTE: Not all widgets are supported in Mobile Px. For example, the `bajoui:PxIncludes` widget and `kitPx:AnalogMeter` widget are not supported. For a complete list of supported widgets, refer to [Field editors, widgets, and bindings supported in Mobile Px, page 36](#)
- Step 7 Select the **CanvasPane** and set the `scaleMode` property to `Fit Ratio`.

The pasted **CanvasPane** automatically scales up or down to fit the size of the **BasicMobilePane**.

About Bound tables in Mobile Px

Starting in AX-3.8, enhanced functionality in bound tables provides horizontal scroll bars, sorting by column, and buttons for advancing forward or backward 20 records at a time.

BoundTables now are rendered in the same manner as History Tables. For more details, refer to “About the history table view” in the [NiagaraAX User Guide](#)

Exploring Bound Table enhancements

Prerequisites:

- Target platform is running AX-3.8
- Niagara Mobile is installed

Use the bound table enhancements

- Step 1 From the **bajoui** palette, drag and drop a BoundTable widget to your mobile Px page.
- Step 2 Configure the bound table's table binding ORD with a suitable bql query.
- Step 3 Connect to the station using a browser with a mobile profile and view the mobile Px page.
- Step 4 Click the right arrow button to advance thru the records.
- NOTE:** Clicking the right arrow advances 20 records per page
- Step 5 Click a column head to sort records by the column.
- Step 6 Click and drag in the horizontal scroll bar to view data in the right side of table.

Explore the new AX-3.8 functionality for bound tables.

Field editors, widgets, and bindings supported in Mobile Px

Supported bindings

The following Bindings are supported in Mobile Px.

Bindings	Notes / Supported Properties
bajoui:Binding	<ul style="list-style-type: none"> • ord • degradeBehavior
bajoui:ValueBinding	(extends bajoui:Binding) <ul style="list-style-type: none"> • hyperlink • popupEnabled (must be true to invoke actions)
kitPx:ActionBinding	(extends bajoui:Binding) <ul style="list-style-type: none"> • actionArg • widgetEvent (only "actionPerformed")
kitPx:BoundLabelBinding	(extends bajoui:ValueBinding) statusEffect
kitPx:PopupBinding	(extends bajoui:Binding)Just performs hyperlink
kitPx:IncrementSetPointBinding	(extends kitPx:SetPointBinding) increment
kitPx:SetPointBinding	(extends bajoui:ValueBinding) <ul style="list-style-type: none"> • widgetEvent (only "actionPerformed")

Bindings	Notes / Supported Properties
	<ul style="list-style-type: none"> widgetProperty
kitPx:SpectrumBinding	(extends bajaui:Binding) <ul style="list-style-type: none"> extent highColor lowColor midColor setpoint widgetProperty
kitPx:SpectrumSetpointBinding	(extends bajaui:ValueBinding)
workbench:WbFieldEditorBinding	(extends bajaui:Binding)

Supported field editors

The following field editors are supported in Niagara Mobile/Mobile Px. Unless specified, all are supported in both AX-3.7 and AX-3.8.

Niagara Type
baja:AbsTime
baja:AbsTimeRange
baja:Boolean
baja:Date
baja:Double
baja:DynamicEnum
baja:Facets
baja:Float
baja:FrozenEnum
baja:Integer
baja:Long
baja:RelTime
baja:String
baja:Time
baja:TimeRange
baja:Unit
baja:StatusBoolean
baja:StatusEnum

Niagara Type
baja:StatusNumeric
baja:StatusString
control:BooleanOverride
control:EnumOverride
control:NumericOverride
control:StringOverride

Supported widgets

Following is a list of Niagara Widgets that are supported in the Mobile Px app. Widgets that are new in AX-3.8, and Widgets that are not supported.

NOTE: Widgets that extend another Widget class also support all properties that are supported in their superclass.

Niagara type	Notes/supported properties
bajau:BoundTable	
bajau:Button	<ul style="list-style-type: none"> • background • blink • enabled • font • foreground • halign • image • text • textFormat • textIconGap • textToIconAlign • valign • visible
bajau:CheckBox	(extends, and converts to, bajau:ToggleButton)
bajau:HyperlinkLabel	<ul style="list-style-type: none"> • background • blink • font • foreground • halign • image • text • textFormat

Niagara type	Notes/supported properties
	<ul style="list-style-type: none"> textIconGap textToIconAlign valign visible
bajoui:Slider	<ul style="list-style-type: none"> enabled increment max min value
bajoui:ToggleButton	(extends bajoui:Button) <ul style="list-style-type: none"> selected
kitPx:BarGraph	
kitPx:BackButton	(extends bajoui:Button)
kitPx:BoundLabel	(extends bajoui:Label) <ul style="list-style-type: none"> padding
kitPx:ForwardButton	(extends bajoui:Button)
kitPx:GenericFieldEditor	<ul style="list-style-type: none"> enabled
kitPx:ImageButton	(extends bajoui:Button) <ul style="list-style-type: none"> - disabled - normal
kitPx:LocalizableButton	(extends bajoui:Button) <ul style="list-style-type: none"> confirmRequired hyperlink
kitPx:LocalizableLabel	(extends bajoui:Label) <ul style="list-style-type: none"> backgroundImage
kitPx:LogoffButton	(extends bajoui:Button)
kitPx:RefreshButton	(extends bajoui:Button)
kitPx:SaveButton	(extends bajoui:Button) <ul style="list-style-type: none"> enabled (becomes enabled when widgets are modified)
kitPx:SetPointFieldEditor	
mobile:BasicMobilePane	<ul style="list-style-type: none"> padding rowGap

Niagara type	Notes/supported properties
mobile:MobileGridPane	<ul style="list-style-type: none"> background columnGap foreground padding roundedCorners rowGap shadow
mobile:MobileJavaScriptButton	(extends bajaui:Button) javascript

Widgets new for AX-3.8

The following Widgets are new in Mobile Px for AX-3.8.

Niagara type	Notes/supported properties
bajaui:BorderPane	<ul style="list-style-type: none"> fill margin padding Border not yet supported.
bajaui:Picture	<ul style="list-style-type: none"> halign (absolute positioning only) image valign scale (absolute positioning only)
bajaui:ScrollPane	<ul style="list-style-type: none"> hpolicy vpolicy
mobile:CanvasPane	<ul style="list-style-type: none"> background enabled halign layer layout scale valign viewSize visible

Widgets that are not supported

The following widgets are explicitly not supported:

- `bajau:NullWidget`
- `bajau:PxInclude`
- `bajau:ScrollBar`
- `bajau:Shape`
- `kitPx:AnalogMeter`

Chapter 3 Component guides

Topics covered in this chapter

- ◆ Types of component modules
- ◆ Components in the App module
- ◆ Components in the Box module
- ◆ Components in the Mobile module

Component guides provide summary help about the components associated with a NiagaraAX Framework module.

Types of component modules

Following is a list of the Mobile component modules:

- [Components in the App module, page 43](#)
- [Components in the Box module, page 43](#)
- [Components in the Mobile module, page 44](#)

Components in the App module

Following is a list of the components in the **app** module:

- [app-AppContainer, page 43](#)
- [app-AppFolder, page 43](#)

app-AppContainer



In NiagaraAX-3.7-3.8, new stations come with an **App Container** already installed as a node directly under the station's **Config** node in the Workbench nav tree. This component is also available from the app or mobile palettes. Any mobile apps that you want to add to your station must be placed directly under this container or in an App Folder that is in the Apps container. For more details, see [About the Apps container, page 19](#).

app-AppFolder



The AppFolder component is a container for apps and is available in the app module. You can use these folders to organize apps and nest them under the AppContainer (Apps) component. Only the Apps component and other AppFolder components are legitimate locations for placing these folders. For more details about using these folders, see [About the Apps container, page 19](#).


Components in the Box module

Following is a list of the components in the **box** module:


- [box-BoxService, page 44](#)
- [box-ORDChannel, page 44](#)
- [box-RegistryChannel, page 44](#)
- [box-ServerSessionChannel, page 44](#)

- [box-SystemChannel](#), page 44


box-BoxService

 In NiagaraAX-3.7-3.8, new stations come with the Box Service already installed as a node directly under the station's **Services** node in the Workbench Nav tree. On a station upgraded from 3.6 or earlier, the BOX service will be automatically installed whenever the Niagara Mobile Apps are installed. BOX (Building Object eXchange) is the protocol used between BajaScript and the Station. This component is also available from the **box** palette. For more details, see [About Bajascript](#), page 15 and [About BOX caching](#), page 17.


box-ORDChannel

 In NiagaraAX-3.7-3.8, this component is a slot in the BoxService Property Sheet view. BOX (Building Object eXchange) is the protocol used between BajaScript and the station. This component is also available from the **box** palette.


box-RegistryChannel

 In NiagaraAX-3.7-3.8, this component is a slot in the BoxService Property Sheet view. BOX (Building Object eXchange) is the protocol used between BajaScript and the station. This component is also available from the **box** palette.

box-ServerSessionChannel

 In NiagaraAX-3.7-3.8, this component is a slot in the BoxService Property Sheet view. BOX (Building Object eXchange) is the protocol used between BajaScript and the station. This component is also available from the **box** palette.

box-SystemChannel


 In NiagaraAX-3.7-3.8, this component is a slot in the BoxService Property Sheet view. BOX (Building Object eXchange) is the protocol used between BajaScript and the station. This component is also available from the **box** palette.

Components in the Mobile module


Following is a list of the components in the **mobile** module:

- [mobile-NiagaraMobileWebApps](#), page 45
- [mobile-MobilePropertySheetApp](#), page 45
- [mobile-MobileAlarmConsoleApp](#), page 45
- [mobile-MobileHistoryApp](#), page 45
- [mobile-MobileScheduleApp](#), page 45
- [mobile-MobilePxApp](#), page 45
- [mobile-BasicMobilePane](#), page 45
- [mobile-CanvasPane](#), page 46
- [mobile-MobileGridPane](#), page 46
- [mobile-MobileJavaScriptButton](#), page 46


mobile-NiagaraMobileWebApps

 The Niagara Mobile Web Apps component is a special App Folder that is configured to contain the Niagara Mobile Web Apps. This container component behaves like a regular App Folder. It can contain additional apps that you install from its (default) App Manager view or by dragging and dropping apps onto the container in the Nav tree.


mobile-MobilePropertySheetApp

 The Niagara Mobile Property Sheet App component provides access to a station's property sheet views. This component is available from the Mobile module under the NiagaraMobileWebApps container (drag and drop to add) or from the App Manager view (use the **New** button to add). With the Mobile Property Sheet you can navigate up and down through the entire component hierarchy of your station, view and edit point files, and fire actions. For more details, see [About the Mobile Property Sheet app, page 19](#).


mobile-MobileAlarmConsoleApp

 The Niagara Mobile Alarm Console App provides a mobile alarm console. This component is available from the Mobile module, under the NiagaraMobileWebApps container (drag and drop to add) or from the App Manager view (use the **New** button to add). The Alarm Console App allows you to view the current alarm counts for each of your alarm recipients, view any detail fields of individual alarms, acknowledge them, and add notes. For more details, see [About the Alarm Console App, page 21](#).


mobile-MobileHistoryApp

 The Niagara Mobile Web Apps component provides access to history files and history views. This component is available from the Mobile module under the NiagaraMobileWebApps container (drag and drop to add) or from the App Manager view (use the **New** button to add). With the Mobile History app you can view history data in chart and table form. It supports delta mode as well as relative and specific time ranges, zooming (for charts) and column selection (for tables). For more details, see [About the Mobile History app, page 23](#).


mobile-MobileScheduleApp

 The Niagara Mobile Schedule App component provides access to scheduling views from a mobile device. This component is available from the Mobile module under the NiagaraMobileWebApps container (drag and drop to add) or from the App Manager view (use the **New** button to add). The Mobile Schedule app supports touch-based editing of individual day schedules as well as special events, calendars, and trigger schedules. For more details, see [About the Mobile Schedule app, page 26](#).


mobile-MobilePxApp

 The Niagara Mobile Px App component allows standard Px pages to display on mobile devices. This component is available from the Mobile module under the NiagaraMobileWebApps container (drag and drop to add) or from the App Manager view (use the **New** button to add). The Mobile Px app allows you to create your own Mobile Px pages using the existing Workbench Px editor, and have them then convert on the fly into a mobile-friendly format. Many existing widgets are supported right out of the box, such as buttons, labels, sliders, and field editors, as well as some new ones. For more details, see [About Mobile Px views, page 31](#).

mobile-BasicMobilePane


 The Basic Mobile Pane component provides a convenient set of simple features for use with the Mobile Px page layout. The properties available with this pane (padding, rowGap, visible) help in the layout of menus and graphics for pages that are going to be displayed on a mobile device. The Basic Mobile Pane is available on the mobile palette.

mobile-CanvasPane


 The Mobile CanvasPane component provides a simple solution to transitioning your existing Workbench Px graphics to display on mobile devices. Using a CanvasPane with the scale property set to `Fit Ratio`, a complex graphic can be scaled down to fit on smaller screens, while maintaining the existing layout, legibility and functionality. The Canvas Pane is available on the mobile palette.

Mobile CanvasPanes support the different scale modes offered in Workbench. If a CanvasPane is absolutely positioned (a child of another CanvasPane) then it supports all five scale modes: `None`, `Fit`, `Fit Ratio`, `Fit Width`, and `Fit Height`. While a CanvasPane that is fluidly positioned (a child of a `BasicMobilePane`) supports only two scale modes: `None` and `Fit Ratio`.

mobile-MobileGridPane

 The Niagara Mobile Grid Pane component is a special pane that helps you design pages for mobile devices. Properties in the Mobile Grid Pane property sheet allow you to define row and column spacing. The Mobile Grid Panel is located in the mobile palette under the Widgets folder.

mobile-MobileJavaScriptButton

 The Mobile JavaScript Button component is a button widget that can recognize JavaScript commands when touched. You can use the Px Editor to edit the JavaScript command for the button. The Mobile JavaScript Button is located in the Mobile module under the Widgets folder.

Chapter 4 Plugins (views)

Topics covered in this chapter

◆ Plugins in the app module

Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

For summary documentation on any view, select **Help→On View (F1)** from the menu or press **F1** while the view is open.


Plugins in the app module

The following plugin is in the **app** module:

- [app-AppManager, page 47](#)

app-AppManager

The App Manager is the default view of the Apps component (App Container) as well as any AppFolders.

You access this view by touching the App Manager icon (). You can use this view to show a tabular listing of all apps in your station. The table of apps contains columns that indicate: app Name, Type, Version, Status, Enabled state and Fault Cause (if any).

You can add new apps, add new AppFolders, and edit existing apps from this view. For more details about the Apps Manager, see [About the Apps container, page 19](#).

Index

A

About apps	7
Absolute positioning	
about.....	32
CanvasPanels.....	33
convert existing Workbench graphics.....	35
create scaled page	34
create unscaled page.....	34
scaling	33
Alarm Console app	21
Alarm source data.....	23
App Manager plugin	47
App prerequisites.....	9
app-AppContainer.....	43
app-AppFolder.....	43
Apps container	19
AppsFolder	13

B

Bajascript	15
Bindings	
supported in Mobile Px	36
Bound tables	36
enhancements in AX-3.8.....	35
BOX server cache	17
increasing cache.....	17
box-BoxService	44
box-ORDChannel.....	44
box-RegistryChannel	44
box-ServerSessionChannel.....	44
box-SystemChannel	44

C

Component guides	43
Component modules	43
Components	16, 44
Components in the App module.....	43
Components in the Box module	43
Configuring a Mobile app	15
Controls	11

D

Desktop view	12
Disable an app.....	11
Document change log.....	5
Dynamic Mobile Px views.....	31

E

Enable an app.....	11
--------------------	----

F

Field editors	
supported in Mobile Px	37

H

History app interface	24
-----------------------------	----

I

Indicators	11
Installation	10
Installation in a station	10

L

Licensing	9
Local storage.....	23

M

Mobile Alarm Console app.....	21
Mobile app module plugins.....	47
Mobile apps guide	5
Mobile History app	23
Mobile module components.....	44
Mobile Property Sheet app	19
Mobile Property Sheet interface.....	20
Mobile Px	
CanvasPane	33
Mobile Px app	30
Mobile Px views.....	31
Mobile Schedule app	26
Mobile Schedule app interface	27
mobile-BasicMobilePane	45
mobile-CanvasPane	46
mobile-MobileAlarmConsoleApp	45
mobile-MobileGridPane	46
mobile-MobileHistoryApp.....	45
mobile-MobileJavaScriptButton	46
mobile-MobilePropertySheetApp.....	45
mobile-MobilePxApp	45
mobile-MobileScheduleApp.....	45
mobile-NiagaraMobileWebApps	45

N

Navigation bar.....	12
new features	8

O

Operating systems supported.....	9
----------------------------------	---

P

plugins	47
Prerequisites	9
Properties	18
Property sheet view enhancements.....	20

R

Related documentation.....	5
----------------------------	---

S

Scaling about.....	33
status colors property sheet view.....	20
Supported field editors, widgets, bindings	36

U

User access	15
-------------------	----

V

views.....	47
------------	----

W

Widget CanvasPane	32–33
Widgets supported in Mobile Px	38