iSMA-B-AAC20

User Manual

Platform AAC20

Global Control 5 Sp. z o.o.
Warsaw, Poland

www.gc5.pl

Table of contents

Introduction

This manual contains information about commissioning iSMA-B-AAC20 controller

1 Revision history

Rev	Date	Description
1.1	28.08.2015	• First edition
1.2	20.04.2017	• Added DHCP option

2 Sedona Updater

Sedona Updater is a piece of software designed to administer an operating system of iSMA-B-AAC20 controller. The software is free and can be obtained from your distribution partner or download from GC5 support website: www.support.gc5.pl. The software allows the following actions (using the buttons in the lower part of the window):

- Send File – Sends a file into the device, displays the transfer progress bar at the bottom of the window, correct transfer is confirmed with a message "Transfer complete",
- Reload Firmware – Uploads a firmware file. This operation must be preceded by a transfer of an adequate firmware files to the device. Upon update, the device shall be rebooted,
- Device Reboot - Reboots the device,
- Default Settings - Sets default value of IP address, RS485 parameters, platform password,
- Console - Displays the operating system console,

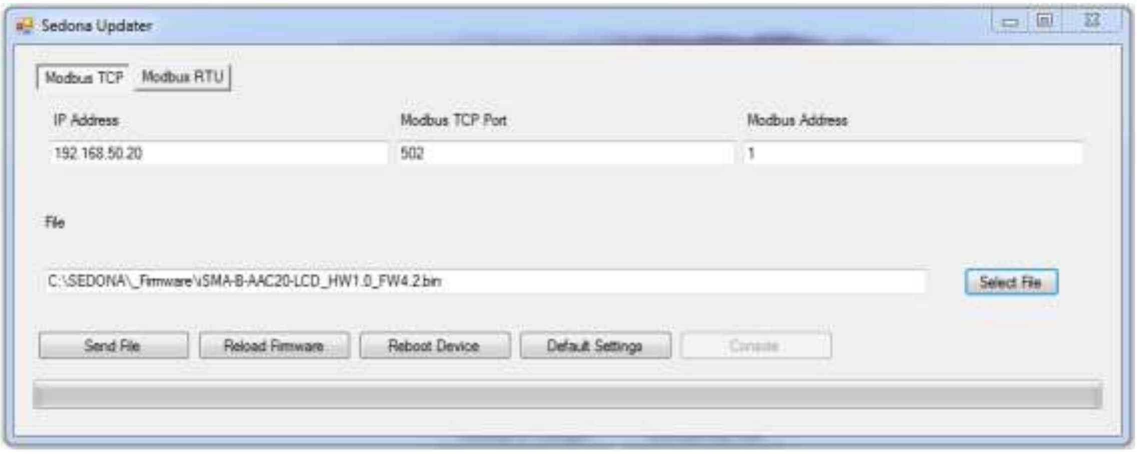
Sedona Updater communicates with the device in two ways:

- Modbus TCP - using the Ethernet port,
- Modbus RTU - using the RS485 port. In this type of communication, the device must be in the Bootloader mode.

The software allows to connect using IP connection, RS485 port or USB A-to-A cable connection (USB only for power supply or/and diagnostic to read console).

2.1 IP connection

To connect to the device using IP connection, open the SedonaUpdater.exe tool and click “Modbus TCP” button. Activity under the IP connection does not require the device to be in the “Bootloader” mode – it is enough to input: the device’s IP address (default 192.168.1.123), Modbus TCP/IP port number (default 502), and the device Modbus Address (default 1).



IP Address: 192.168.50.20

Modbus TCP Port: 502

Modbus Address: 1

File: C:\SEDONA_Firmware\iSMA-B-AAC20-LCD_HW1.0_FW4.2.bin

2.2 RS485 connection

To connect to the device using the RS485 connection, run SedonaUpdater.exe and click the “Modbus RTU” button. In the COM Port text field, enter the RS485 port number of your PC, set the Baud Rate (the default is 115200), and the device Modbus Address (default 1).

In order to operate using the RS485 port connection, you must run “Bootloader” mode on the device.

Bootloader in version with built-in display (iSMA-B-AAC20-LCD):

Step 1 - Disconnect the power and wait for the device to fully shut down,

Step 2 - Hold down the Esc key, and turn the power on again. The screen should now display “Bootloader”.

Step 3 - Restarting exits the Bootloader mode and returns to normal operation.

Bootloader in version without a display (iSMA-B-AAC20):

Step 1 - Disconnect the power and wait for the device to fully shut down, Step 2 - With the device off, take off the cover and move the DIP Switch no 8 to ON position, turn the power on again. Activation of the Bootloader mode is confirmed by alternating power and communication LEDs.

Step 3 - To exit the Bootloader mode, switch 8 to OFF and restart the device.

2.3 Sending file to the device

The files can be sent using Modbus TCP protocol over an IP connection. This is the recommended method, as it does not require switching the device into the "Bootloader" mode, and provides the fastest way to transfer data.

If you send files using the RS485 port, the device must first be switched into the “Bootloader” mode, as described above. Then, using the "Select File" button, select a file to upload. The upload is done through selecting the right protocol (Modbus TCP/Modbus RTU) and approving the process by pressing the "Send File" button. Progress of the transfer is indicated by a progress bar at the bottom of the screen. After a successful upload, a message pops up saying "Transfer Complete".

2.4 Restoring the device’s default settings

Pressing the "Default Setting", restore the factory settings, such as:

- IP address 192.168.1.123,
- Mask 255.255.255.0,
- Default Gateway 192.168.1.1,
- Modbus TCP/IP protocol port 502,
- The password for “platform” user to default 1000.

2.5 System Console

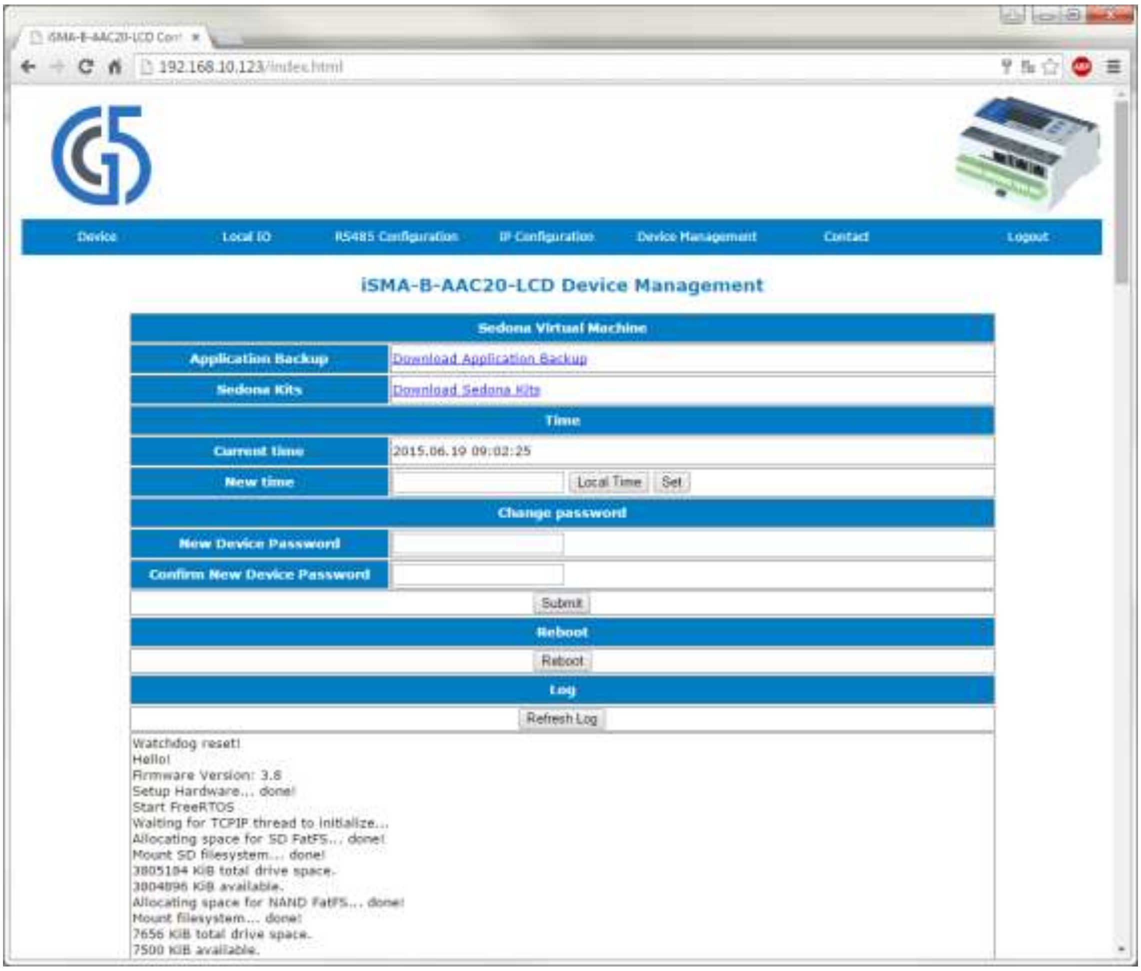
You can view system logs of the controller operating system. There are two options to view system console:

- By web page,
- By USB connection.

Web page system console

To view system console by web page enter controller IP address in web browser and login to system (default user: platform, default password 1000). After login go to “Device

Management" section and press "Refresh Log" button. All system logs will be displayed on the window bellow.



iSMA-B-AAC20-LCD Device Management

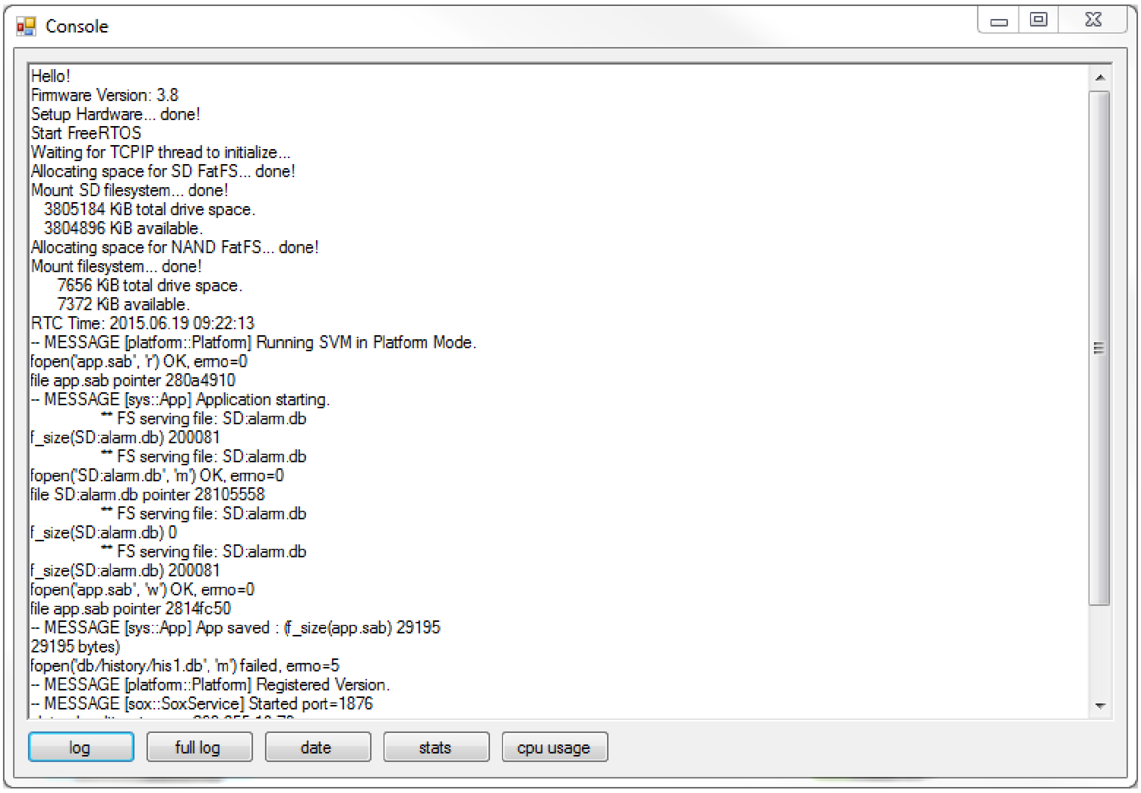
Sedona Virtual Machine	
Application Backup	Download Application Backup
Sedona Kits	Download Sedona Kits
Time	
Current time	2015.06.19 09:02:25
New time	Local Time Set
Change password	
New Device Password	
Confirm New Device Password	
	Submit
Reboot	
	Reboot
Log	
	Refresh Log

```
Watchdog reset!
Hello!
Firmware Version: 3.8
Setup Hardware... done!
Start FreeRTOS
Waiting for TCPIP thread to initialize...
Allocating space for SD FatFS... done!
Mount SD filesystem... done!
3805184 KiB total drive space.
3804896 KiB available.
Allocating space for NAND FatFS... done!
Mount filesystem... done!
7656 KiB total drive space.
7500 KiB available.
```

USB connection system console

To view the logs,use SedonUpdater.exe software and connect device using the A-A USB cable. After connection the system automatically recognize controller without installing additional driver (HMI driver is used default) and Console button will be active. Open System Console window by pressing Console button .w There are available display options:

- Log – print only last written data,
- Full log – print all log memory,
- Date – print controller current date,
- Stats – list of all running task and amount of processing time that has been used by each task,
- CPU usage – current CPU usage.



```
Hello!
Firmware Version: 3.8
Setup Hardware... done!
Start FreeRTOS
Waiting for TCPIP thread to initialize...
Allocating space for SD FatFS... done!
Mount SD filesystem... done!
  3805184 KiB total drive space.
  3804896 KiB available.
Allocating space for NAND FatFS... done!
Mount filesystem... done!
    7656 KiB total drive space.
    7372 KiB available.
RTC Time: 2015.06.19 09:22:13
-- MESSAGE [platform::Platform] Running SVM in Platform Mode.
fopen('app.sab', 'r') OK, errno=0
file app.sab pointer 280a4910
-- MESSAGE [sys::App] Application starting.
            ** FS serving file: SD:alarm.db
f_size(SD:alarm.db) 200081
            ** FS serving file: SD:alarm.db
fopen('SD:alarm.db', 'm') OK, errno=0
file SD:alarm.db pointer 28105558
            ** FS serving file: SD:alarm.db
f_size(SD:alarm.db) 0
            ** FS serving file: SD:alarm.db
f_size(SD:alarm.db) 200081
fopen('app.sab', 'w') OK, errno=0
file app.sab pointer 2814fc50
-- MESSAGE [sys::App] App saved : (f_size(app.sab) 29195
29195 bytes)
fopen('db/history/his1.db', 'm') failed, errno=5
-- MESSAGE [platform::Platform] Registered Version.
-- MESSAGE [sox::SoxService] Started port=1876
```

Figure 1 Sedona updater console view

This connection method is recommended when user cannot connect using IP or RS485 ports or don't know connection parameters for example IP address.

2.6 Firmware upgrade

Firmware can be updated by uploading the device with an appropriate firmware file, and reloaded with new firmware. The following files are necessary to upgrade the firmware (all files are available on GC5 support website: www.support.gc5.pl):

- SedonaUpdater.exe – Software can send files to controller and do system actions,
- Firmware file – Binary file with firmware code,
- The latest kits – All kits must be updated to the current version, they can be downloaded and imported as one bundle zip file,
- Default app.sab file – (optional) Default Sedona application file, adapted to the current firmware version,
- Default kits.scode – (optional) Default Sedona kits image, adapted to the current firmware version.

Step 1 – Download files from GC5 support website: www.support.gc5.pl to local PC drive,

Step 2 – Upgrade kits files in WorkPlace. (see Importing kits to Workplace chapter), Step 3a - Send files by IP connection – Using IP connection controller will automatically run Bootloader mode.

Open SedonaUpdater.exe and set up communication parameters :

- Tab Modbus TCP,
- IP Address – Controller IP address (default for new devices 192.168.1.123),
- Modbus TCP/IP port – Controller Modbus TCP/IP port (default 502),
- Modbus Address – Controller Modbus address (default 1).

Step 3b - Send files by RS485 connection.

WARNING! Using RS485 port controller must be manually switched to Bootloader mode (see RS485 connection chapter).

Open SedonaUpdater.exe and set up communication parameters :

- Tab Modbus RTU,
- COM port – PC RS485 COM port number,
- Baud Rate – Controller RS485 port baud rate (default 115200),
- Modbus Address – Controller Modbus address (default 1).

Step 4 - Select the firmware file (.bin file) using the "Select File" button and send to the controller using the "Send File" button, after success transfer you will get message “Transfer OK”.

Step 5 – Send default app.sab and kits.scode files (optional). WARNING! This operation will erase controller application, not recommended if you want to keep your application or backup application on your PC drive.

Select app.sab file and send to the controller after “Transfer OK” message select second file kits.scode and send to the controller.

Step 6 – Reboot controller using Reload Firmware button. The controller will load new firmware and restart.

Step 7 – Check controller firmware version (see Checking firmware version) if it does notchange repeat again steps from this chapter.

2.7 Checking firmware version

There are three options to check controller current firmware version.

By Workplace

To verify the current firmware version, log in to the device and go to the Plat component. The Plat component is located in the Service folder, directly below the App component (Sedona -> App -> Service -> plat). Plat contains the "Firmware Version" slot identifying the working firmware version.

By Web page

To verify the current firmware version, open browser, enter controller IP address and enter user platform and 4 digits password (default 1000, it can by changed in Sedona -> App -> Service -> plat component in slot “Device Password”). Firmware version is displayed on Device tab in Basic Information.

By LCD system menu

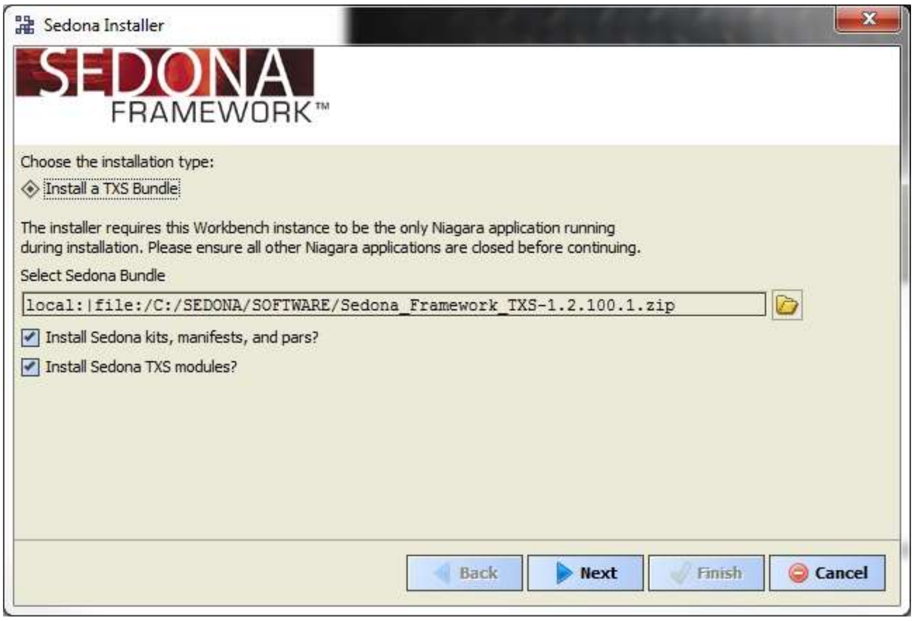
To verify the current firmware version, login to system menu by holding F1 button, enter LCD platform user 4 digits password (default 1000 or from plat component), and open Info page.

3 Sedona in WorkPlace

Sedona device configuration and programming is based on the WorkPlace software. Out of the box, WorkPlace has no plugin to handle Sedona devices and it should be installed. The installation files can be downloaded from the "Niagara Central" website or obtained from your distribution partner. WorkPlace 3.7 version and older require bundle Sedona_Framework_TXS-1.2.28. WorkPlace 3.8 version and later use bundle Sedona_Framework_TXS-1.2.100. To get Sedona Framework installation file, go to Niagara Central or contact your distributor.

3.1 Environment installation step by step

Step 1 – In the menu bar, select the Tools tab and application Sedona Installer. The first installation window pops up, where you choose the Sedona bundle file to select.



Choose the installation type:

◈ Install a TXS Bundle

The installer requires this Workbench instance to be the only Niagara application running during installation. Please ensure all other Niagara applications are closed before continuing.

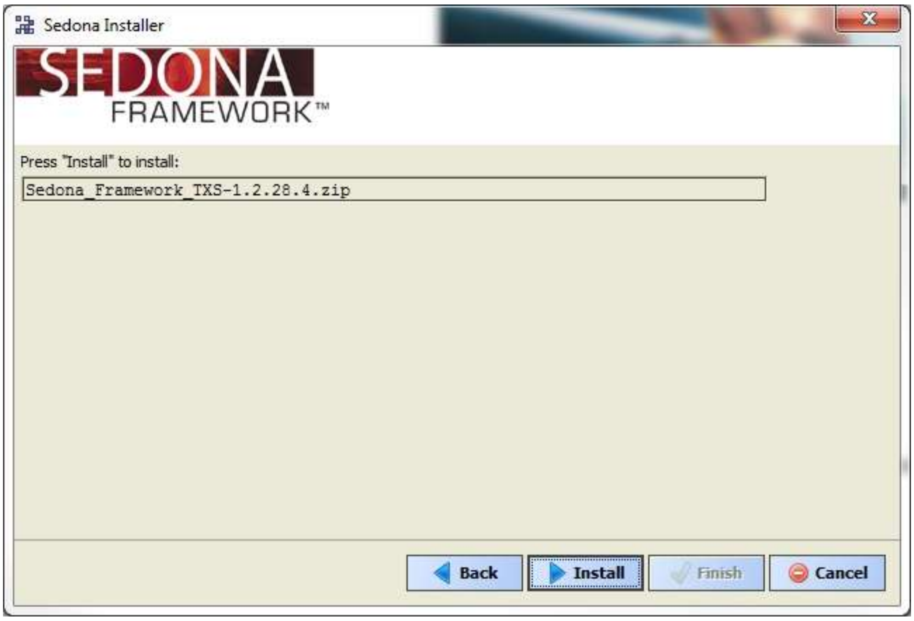
Select Sedona Bundle

`local:|file:/C:/SEDONA/SOFTWARE/Sedona_Framework_TXS-1.2.100.1.zip`

☑ Install Sedona kits, manifests, and pars?

☑ Install Sedona TXS modules?

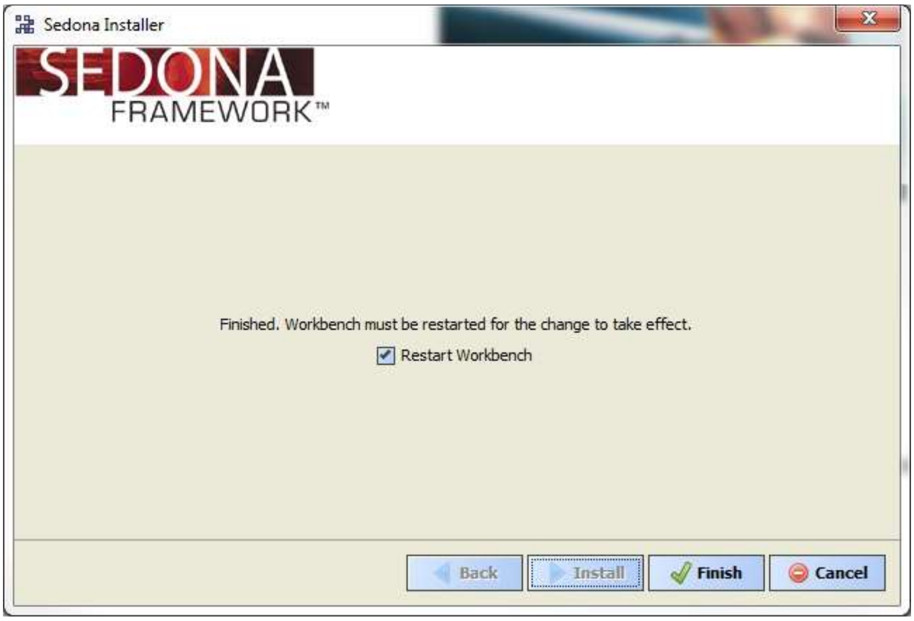
Step 2 – Click “Next” and move on to terms and conditions of the license. To continue, accept and confirm your installation by clicking Install.



Press "Install" to install:

Sedona_Framework_TXS-1.2.28.4.zip

Step 3 – Upon successful installation of the Sedona environment, restart WorkPlace by clicking the “Restart Workbench” option followed by the Finish button.



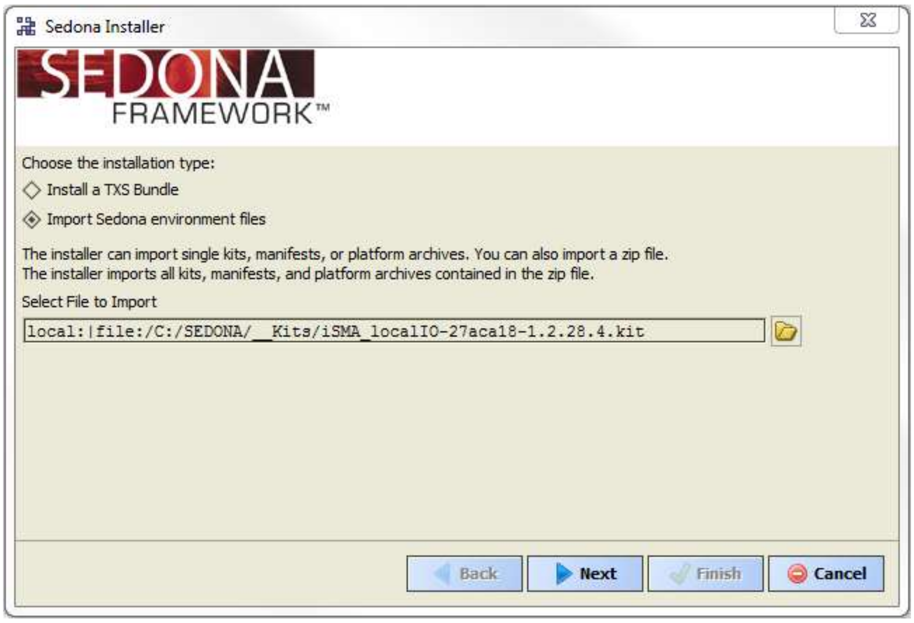
Finished. Workbench must be restarted for the change to take effect.

☑ Restart Workbench

3.2 Importing kits to Workplace

To import files to WorkPlace it is recommended to use “Sedona Installer” from Tools menu and choose “Import Sedona environment files”. The latest kits and kits bundle can be downloaded from support.gc5.pl web site. There are two options to import kits to Workplace:

- Singly – by choosing single kit file with .kit extension



Choose the installation type:

○ Install a TXS Bundle

● Import Sedona environment files

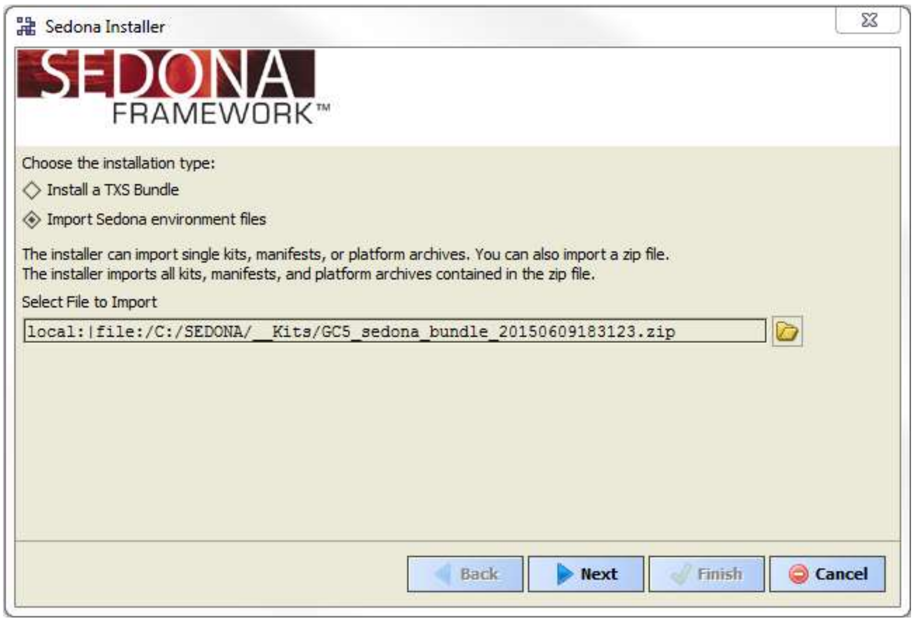
The installer can import single kits, manifests, or platform archives. You can also import a zip file.
The installer imports all kits, manifests, and platform archives contained in the zip file.

Select File to Import

local:|file:/C:/SEDONA/__Kits/iSMA_localIO-27aca18-1.2.28.4.kit

Open Sedona installer from Tools menu in Workplace, mark option "Install Sedona environment files" and select kit file. Press next button to finish importing process. This process copies this file to Sedona kits folder.

- Groups – by choosing prepared bundle file with zip extension



Choose the installation type:

○ Install a TXS Bundle

● Import Sedona environment files

The installer can import single kits, manifests, or platform archives. You can also import a zip file.
The installer imports all kits, manifests, and platform archives contained in the zip file.

Select File to Import

local:|file:/C:/SEDONA/__Kits/GC5_sedona_bundle_20150609183123.zip

Open Sedona installer from Tools menu in Workplace, mark option "Install Sedona environment files" and select zip file. Press next button to finish importing process. This process copies all kit files from zip archives to Sedona kits folder.

3.3 Installing ismaUI module

ismaUI module is a special module which include WorkPlace views for Sedona components.

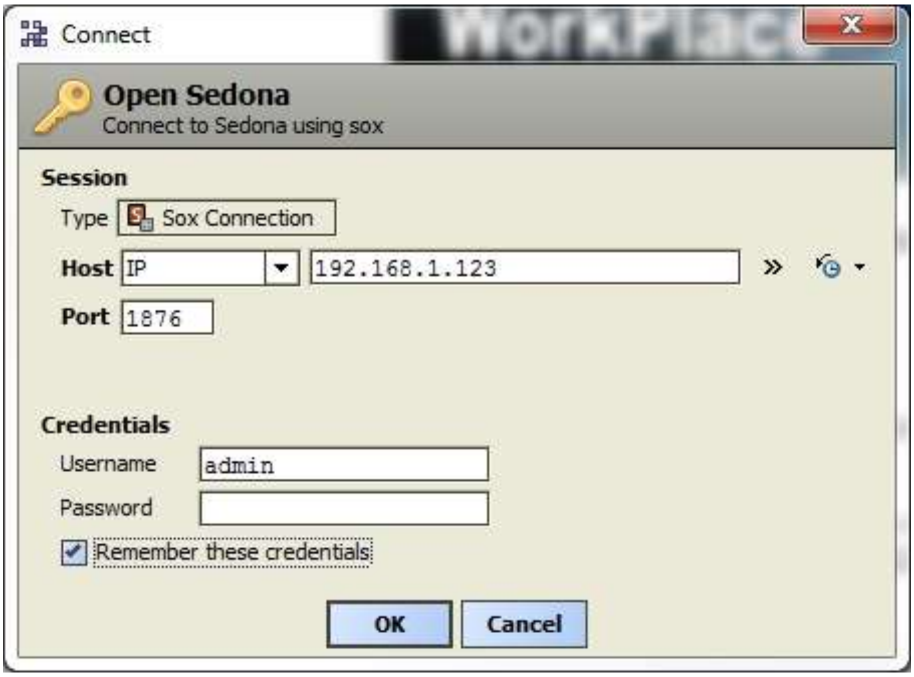
To install ismaUI module download jar file from support.gc5.pl website and copy ismUi.jar file to Niagara/module folder on local PC. To make changes, restart WorkPlace.

3.4 Connecting to Sedona device

In order to connect to the device, select "Open" Device" in the menu bar (File -> Open -> Open Device)

Note: No option "Open Device" should be available, no Sedona plugin has been installed in WorkPlace. In this case, proceed to install the Sedona environment as described in the previous section.

After selecting the above option, a device login window pops up.



Session

Type: Sox Connection

Host: IP 192.168.1.123

Port: 1876

Credentials

Username: admin

Password:

☑ Remember these credentials

By default, a new device should have the following values:

- IP address – 192.168.1.123,

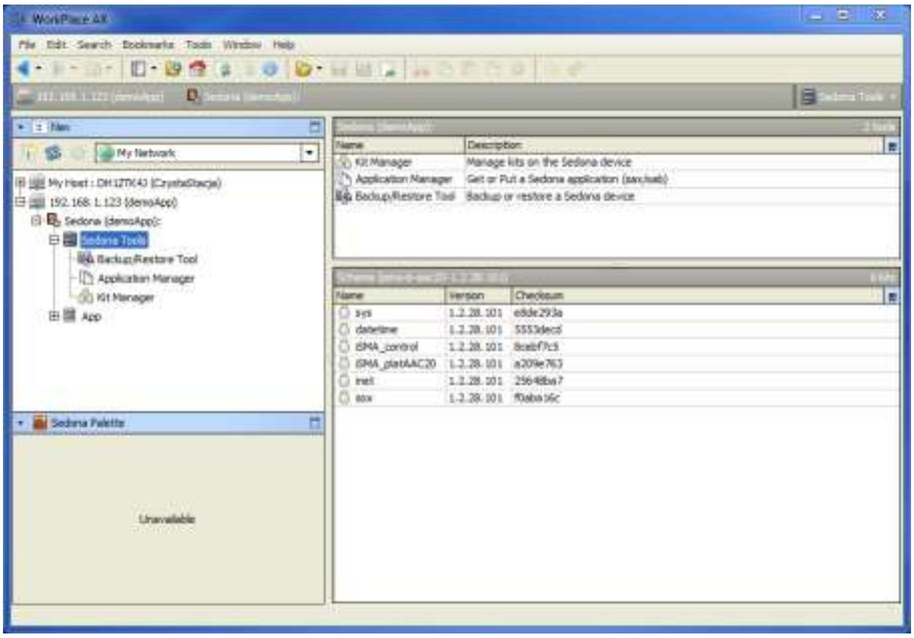
- Port – 1876,
- User – admin,
- Password – empty box (no characters).

The connection parameters can be changed after connect to controller:

- IP address – in plat component (see Change IP Address chapter),
- Sox Port - in sSox service (see Sox chapter),
- Admin password/new user – in user service (see user chapter).

4 Sedona Tools

Sedona Tools is a part of the WorkPlace tools dedicated to manage device applications and kits. In this view we can also see device installed kits with kits checksum (to recognize kits version). More information about Sedona Tools can be found in docSedonaSoxTools.pdf under Niagara docs files, installed in your machine in the Niagara installation process.



Name	Description
Kit Manager	Manage kits on the Sedona device
Application Manager	Get or Put a Sedona application (sax/sab)
Backup/Restore Tool	Backup or restore a Sedona device

Name	Version	Checksum
sys	1.2.28.101	e8de293a
datetime	1.2.28.101	5553decd
iSMA_control	1.2.28.101	8cebf7c5
iSMA_platAAC20	1.2.28.101	a209e763
inet	1.2.28.101	25648ba7
sox	1.2.28.101	f0aba16c

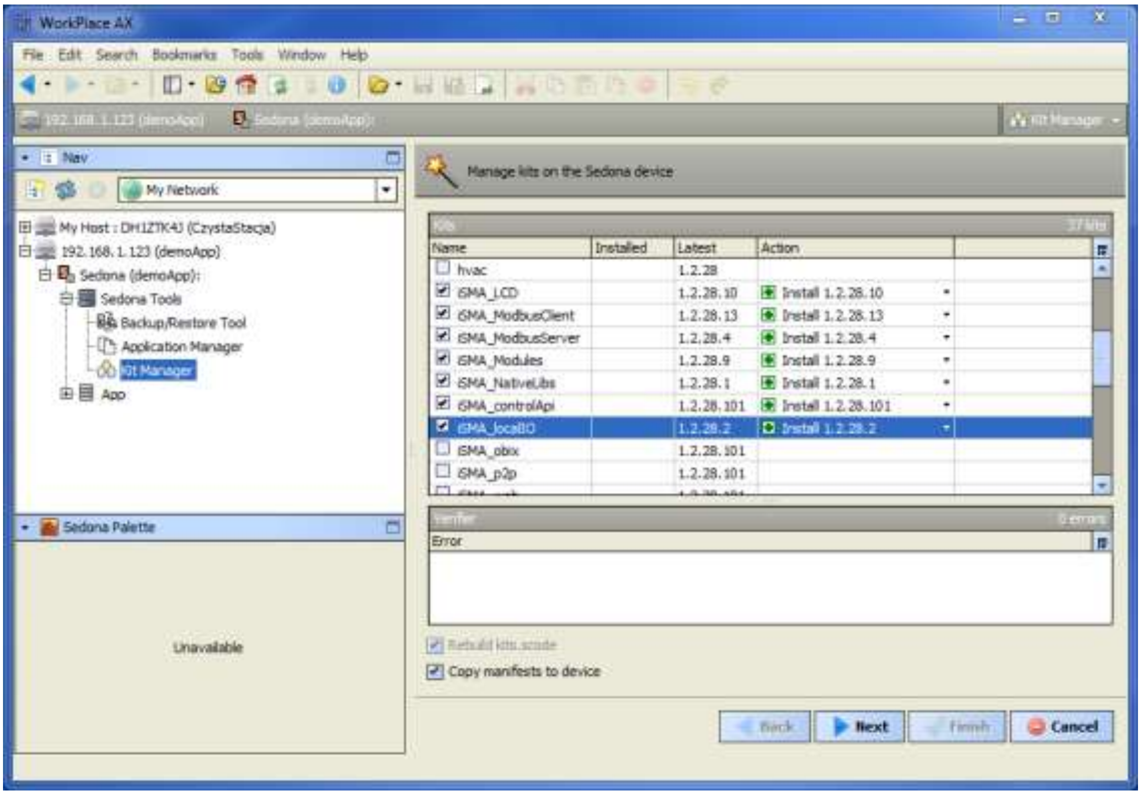
4.1 Sedona kits

Sedona environment kits are sets of components for programming applications. The device is provided with a set of basic kits and in order to develop applications, the set should be uploaded to your device. To add a new or update kit, it must first be imported into WorkPlace (see chapter Importing kits to Workplace). In device, the set of kits is stored in file kits.scode. To manage controller kits and compile kits.scode the Workplace use tool Kit Manager (see Kit Manager chapter).

4.2 Kit Manager

The Kit Manager Tool is used to manage kits on Sedona devices. The kits.scode file is a file which contains a set of kits in a Sedona Framework code image file with an .scode extension. The scode image is executed by the SVM (Sedona Virtual Machine) in the device. Using the Kit Manager, you can add and remove kits, as well as change versions of kits deployed in the scode running on the device. Also, you have the option to force the kits.scode to be rebuilt.

The kits.scode must be rebuilt whenever a change affects the schema. This ensures such changes are included in the recompiled binary .scode image. When you make any kit selection changes in the Kit Manager, it automatically selects and check this option. The only time you would select this option is when you want to force the scode to be rebuilt (even if no change was made to kit selections). This option is most useful in a development environment where the actual code in a kit might change while the kit version and checksum do not. In this case, no change is required in the Kit Manager but the scode must be regenerated in order to deploy the new code on a device.



Name	Installed	Latest	Action
□ hvac		1.2.28	
☑ iSMA_LCD		1.2.28.10	Install 1.2.28.10
☑ iSMA_ModbusClient		1.2.28.13	Install 1.2.28.13
☑ iSMA_ModbusServer		1.2.28.4	Install 1.2.28.4
☑ iSMA_Modules		1.2.28.9	Install 1.2.28.9
☑ iSMA_NativeLibs		1.2.28.1	Install 1.2.28.1
☑ iSMA_controlApi		1.2.28.101	Install 1.2.28.101
☑ iSMA_localIO		1.2.28.2	Install 1.2.28.2
□ iSMA_obix		1.2.28.101	
□ iSMA_p2p		1.2.28.101	

Error

☑ Rebuild kits.scode

☑ Copy manifests to device

The "Action" drop-down list options can be one or more of the following:

- Keep at [the current version],
- Upgrade to [a newer version],
- Downgrade to [an older version].

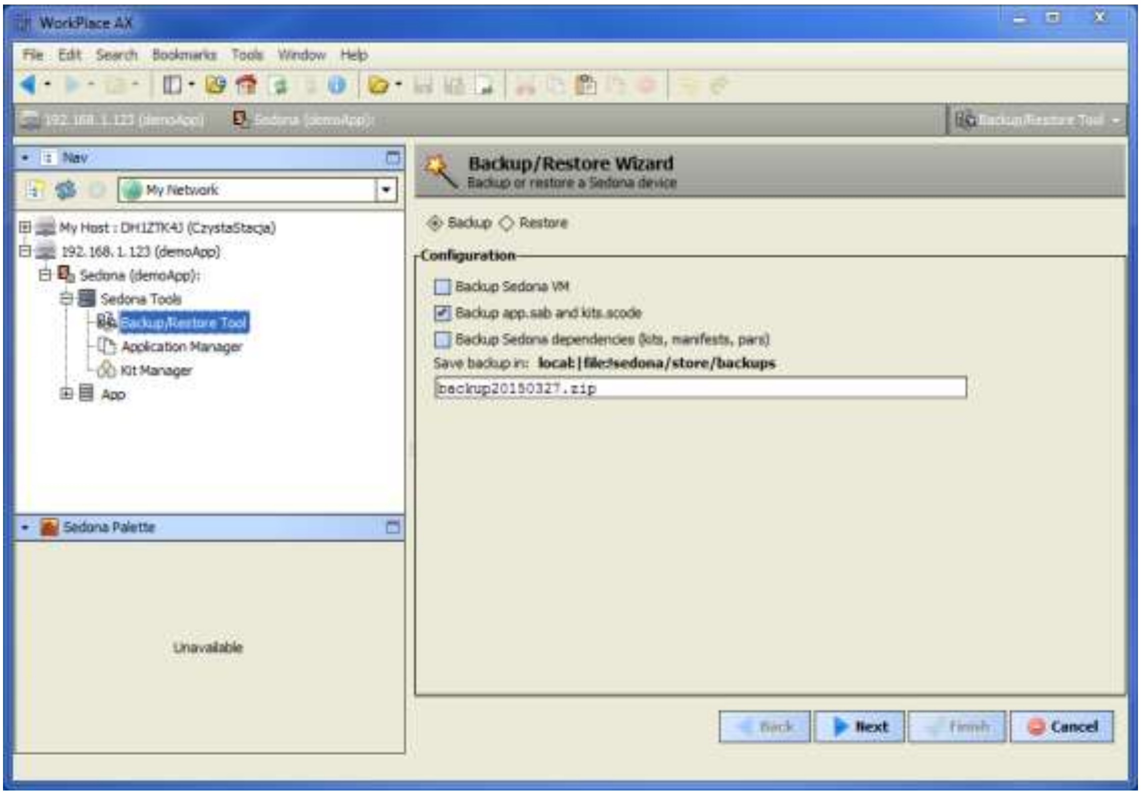
The newer/older kit version levels shown in the Action drop-down menu are available in Workbench. They are stored in the NIAGARA\sedona\kits directory.

In order to remove a kit, first remove all its components from the application. If the application does not contain any kit components, its icon in Kit Manager shall be replaced with a check box. To delete it, you must deselect it in the "name" column (the "action" column will now show "remove"). The following steps are the same as in the case of installation.

4.3 Backup/Restore Tool

The Application Backup/Restore Tool is used to backup or restore the device application from the file. The Backup option downloads the necessary files from the device, and saves them in the directory "/Niagara_Home / Sedona / store / backup" in the form of a zip archive. This tool backs up all the necessary files needed to restore the application on a "clean" device. There are three backup options available:

- Sedona VM Backup - backup of the Sedona Virtual Machine (not supported by iSMA-B-AAC20)
- app.sab and kits.scode Backup - backup of application files app.sab and kits.scode (kits image). (Selected by default)
- Sedona dependencies Backup (kits, manifests, pars) - This option backs up all files needed for the app to run on another or new device.



Backup/Restore Wizard
Backup or restore a Sedona device

◈ Backup ◇ Restore

Configuration

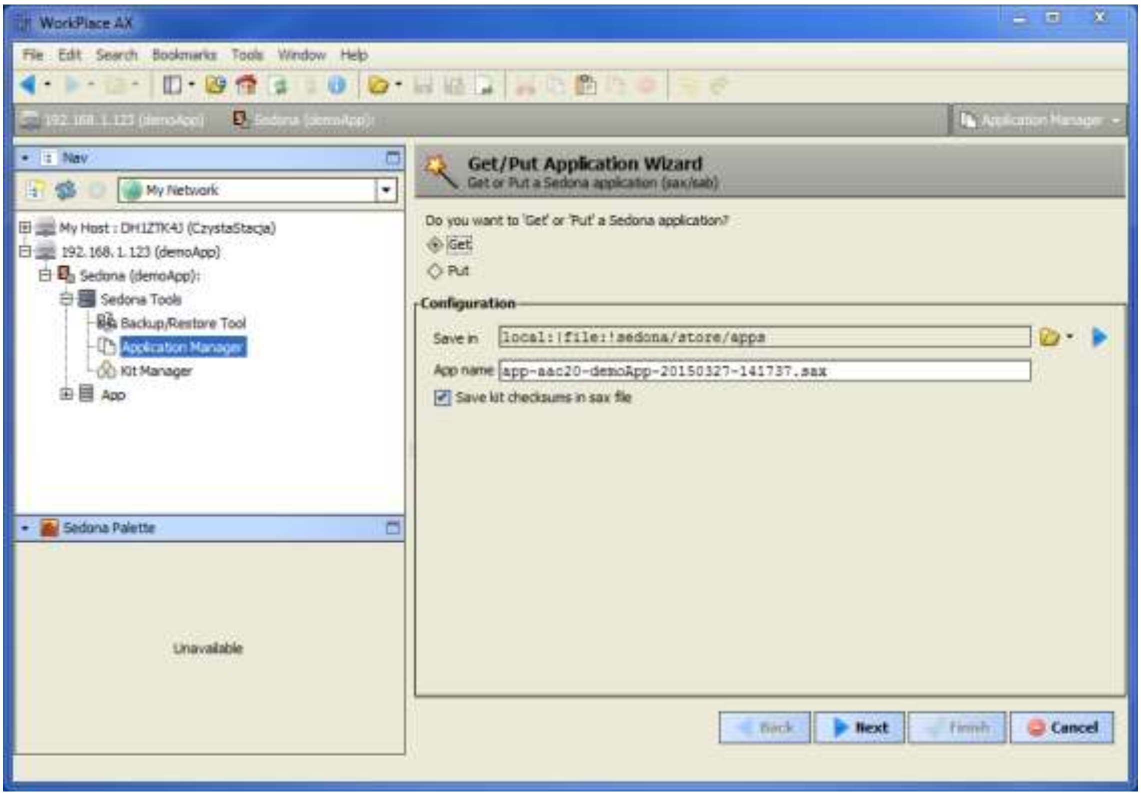
□ Backup Sedona VM
☑ Backup app.sab and kits.scode
□ Backup Sedona dependencies (kits, manifests, pars)

Save backup in: **local:|file:sedona/store/backups**

backup20150327.zip

4.4 Application Manager

The App Manager tool is used to manage app.sax file. The app.sax is a Sedona Application XML file which is a simple XML representation of a Sedona Framework app that is easily generated and consumed by Sedona Framework software tools. This file can be opened and changed in offline mode. The Manager has two options Get and Put. Choosing the Get option allows you to save app.sax file on a local PC drive. Choosing the Put option allows you to send file to the device.



Get/Put Application Wizard

Get or Put a Sedona application (sax/sab)

Do you want to 'Get' or 'Put' a Sedona application?

● Get

○ Put

Configuration

Save in: local:|file:!sedona/store/apps

App name: app-aac20-demoApp-20150327-141737.sax

☑ Save kit checksums in sax file

5 License

Sedona controllers have built-in license mechanism. All out of box controllers have installed standard license version and it is ready to work. But in case of license changing or after memory clear the license must be re-uploaded. There are two modes of doing it::

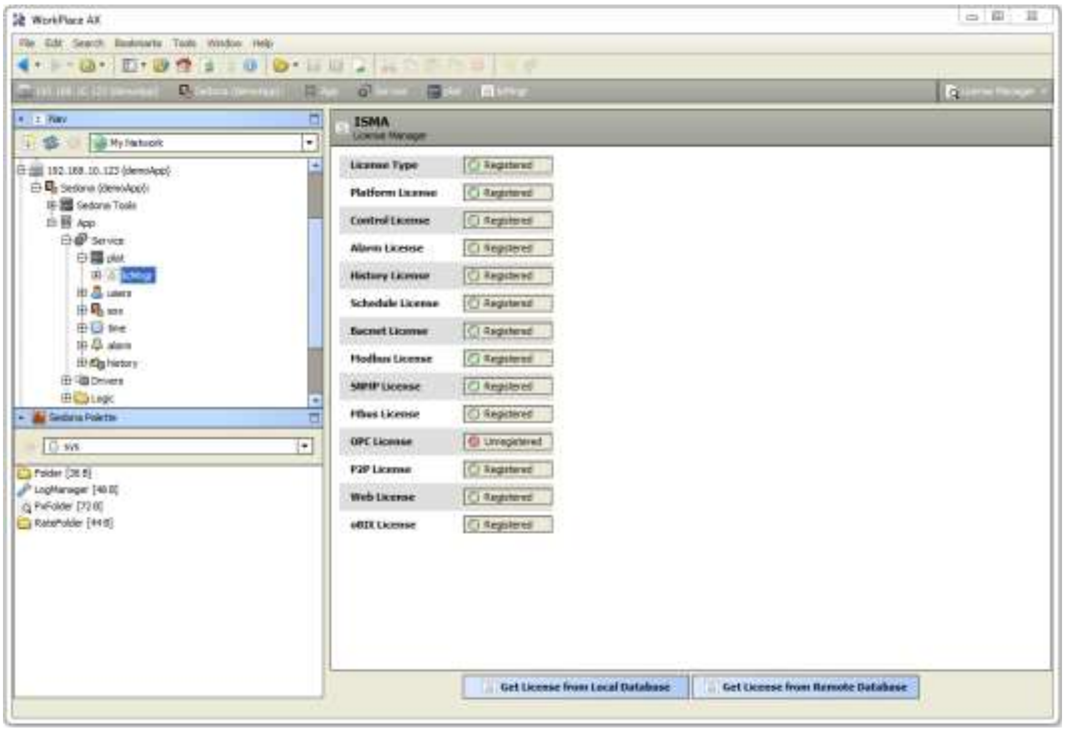
- Online mode,
- Offline mode.

The licenses is generated by serial number or MAC address. The proper file name format is “MAC address without dots”.dat.

WARNING! To manage license under WorkPlace in Online mode, you must download ismaUI.jar module file with contain special view for Sedona components. The ismaUI file must be placed in Niagara/modules folder.

5.1 Sedona license – Online mode

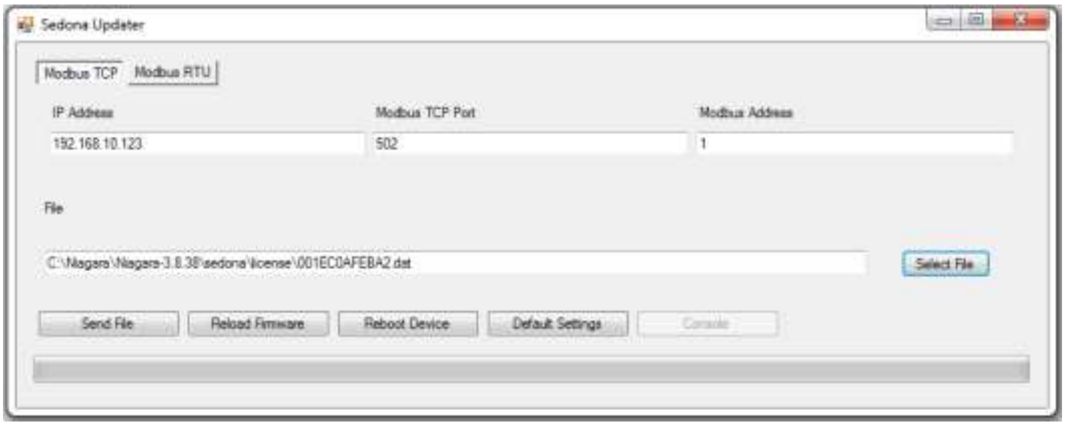
Using Sedona Online mode Workplace can connect to license server, download and save it on local PC in Niagara/Sedona/license folder. To download license from server pleas use “Get License from Remote Database”. During download process will pop-up windows with status progress. After completed download operation use “Get License from Local Database” to send license file to the controller. After device reboot Sedona will boot with new license.



License Type	Registered
Platform License	Registered
Control License	Registered
Alarm License	Registered
History License	Registered
Schedule License	Registered
Bacnet License	Registered
Modbus License	Registered
SNMP License	Registered
Mbus License	Registered
OPC License	Unregistered
P2P License	Registered
Web License	Registered
oBIX License	Registered

5.2 Sedona license – Offline mode

License in offline mode can be sent to controller as a standard file using SedonaUpdater software. License file can be download using ismaUI to local database, copy from other PC or sent by email from local iSMA distributor. Open SedonaUpdater.exe software and choose license file (proper file format is "MAC address (without dots)".dat). After success upload reboot controller using "Reboot Device" button. (For more information about sending files see "Sending file to device" chapter). After device reboot Sedona will boot with new license.



IP Address: 192.168.10.123

Modbus TCP Port: 502

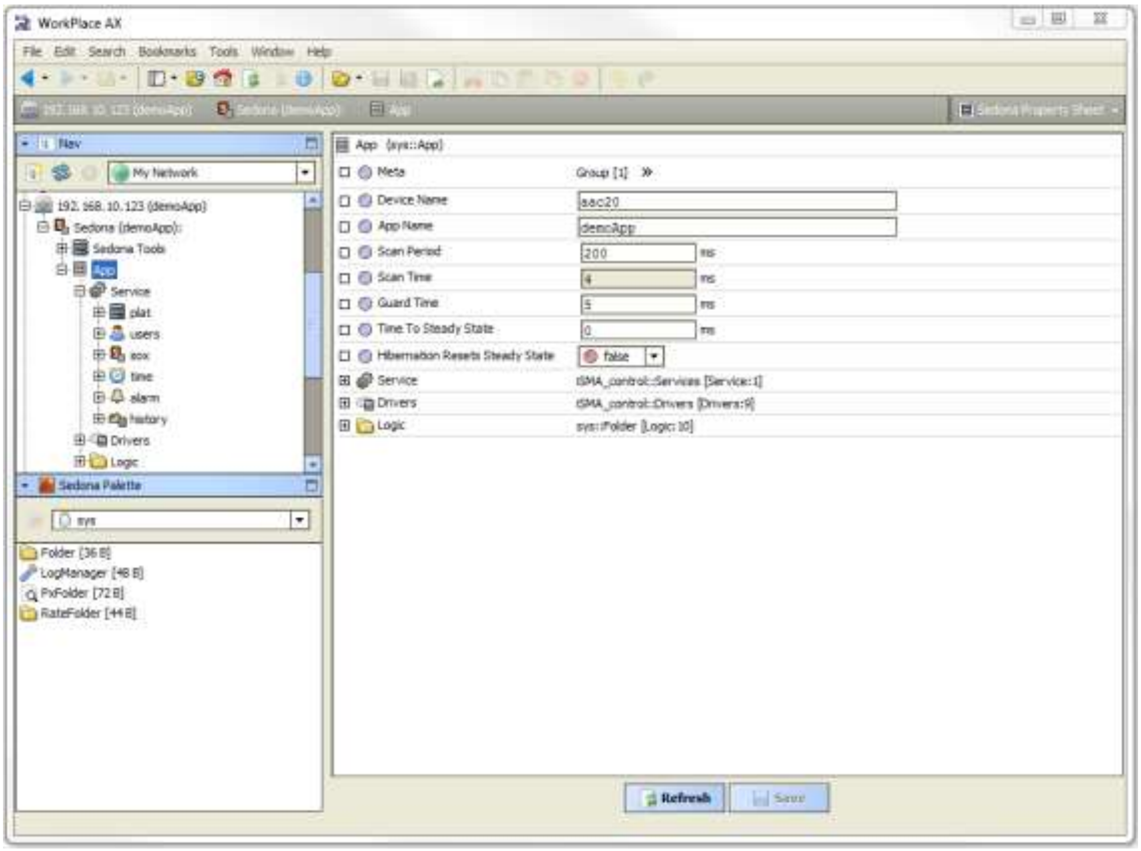
Modbus Address: 1

File: C:\Niagara\Niagara-3.8.38\sedona\license\001EC0AFEBA2.dat

6 Sedona App component

Sedona Application is made of services and components available from the Palettes. Components are processed in every working cycle of the device. Services are certain components enabling system functions such as user management. All items should be placed under the App main component. When the application is modified, the App icon is displaying a reminder that the application should be saved. The component has the following actions available under the right mouse button click:

- Save – save Sedona application in the device's flash memory,
- Restart – restart application (Sedona Virtual Machine),
- Reboot – reboot device.



App (sys::App)

Meta	Group [1] »	
Device Name	aac20	
App Name	demoApp	
Scan Period	200	ms
Scan Time	4	ms
Guard Time	5	ms
Time To Steady State	0	ms
Hibernation Resets Steady State	false	
Service	iSMA_control::Services [Service:1]	
Drivers	iSMA_control::Drivers [Drivers:9]	
Logic	sys::Folder [Logic: 10]	

- Device Name – Device Name text,
- App Name – Application Name text,
- Scan Period – One cycle execution time,
- Scan Time – Real time of one cycle execution,
- Guard Time – Reserve time to finish system tasks,
- Time To Steady State - Time from app start to steady state,
- Hibernation Resets Steady State – not active in AAC20.

Folder

Folder is a component of no properties and it is used exclusively to arrange applications visually. The folder component is available from the Sys Kit Palette.

Rate Folder

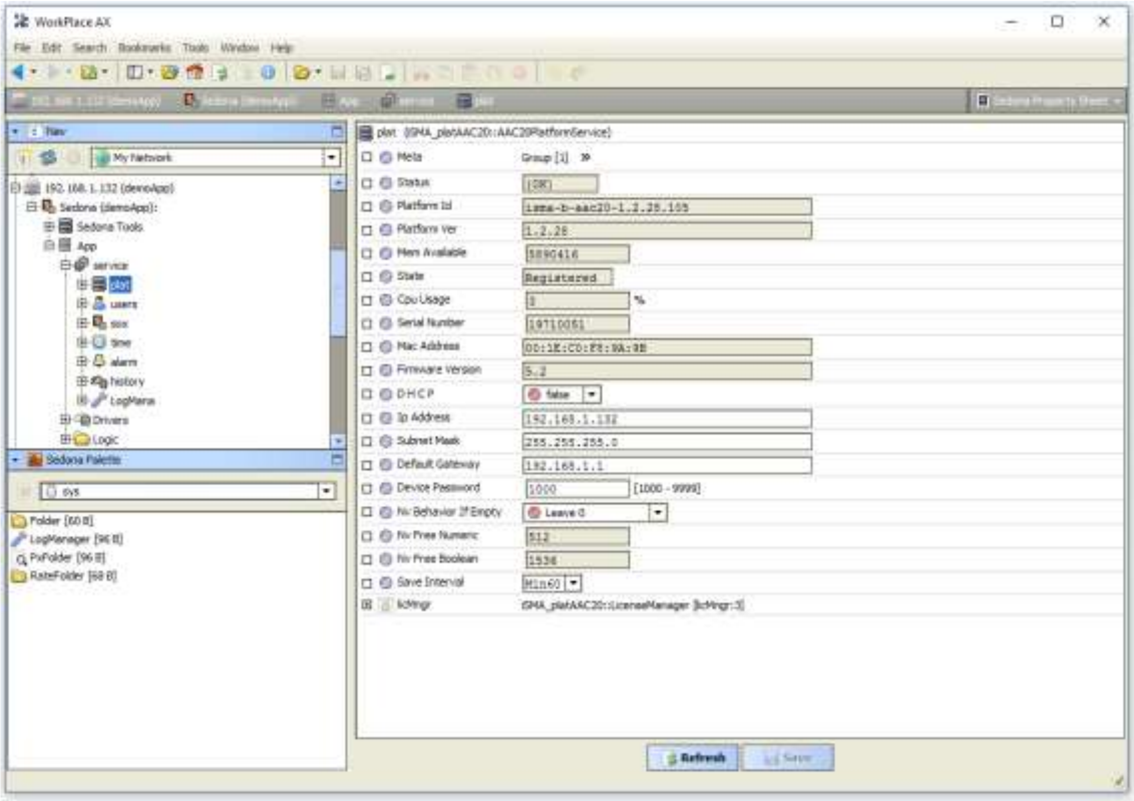
This is a folder used to reduce the rate of application performance. While developing major applications, it allows for reduction in demand for processing power, consuming it for major or quick functions of the device algorithm. The folder's parameter "App Cyclesd To Skip" defines how frequently, in no. of scanning cycles, components under the folder are processed.

6.1 Plat service

The Plat service is a component, which shows device mains parameters, license status and allows to manage IP address. This component is placed under the Service folder and is associated with device hardware.

The component has the following actions:

- Restart – restarts the application (Sedona Virtual Machine),
- Reboot – reboots the device,
- Copy From Nv To Default – copies values from Out to Default slot in all NV components (see NV component chapter),
- Copy From Nv To User – copies values from Out to User slot in all NV components (see NV component chapter),
- Copy From Default To NV – copies values from Default to Out slot in all NV components (see NV component chapter),
- Copy From User To NV – copies values from User to Out slot in all NV components (see NV component chapter),
- Set All Nv In Auto – sets all NV components in auto mode.



plat (iSMA_platAAC20::AAC20PlatformService)

Meta	Group [1] »
Status	{ok}
Platform Id	isma-b-aac20-1.2.28.105
Platform Ver	1.2.28
Mem Available	5890416
State	Registered
Cpu Usage	3 %
Serial Number	19710051
Mac Address	00:1E:C0:F8:9A:9B
Firmware Version	5.2
DHCP	false
Ip Address	192.168.1.132
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
Device Password	1000 [1000 - 9999]
Nv Behavior If Empty	Leave 0
Nv Free Numeric	512
Nv Free Boolean	1536
Save Interval	Min60
lcMngr	iSMA_platAAC20::LicenseManager [lcMngr:3]

- Status – Platform status,
- Platform ID – Platform ID,
- Platform Ver – Platform Version,

- Mem Available – Controller RAM memory available,
- State – License status,
- Cpu Usage – CPU usage from last 5 seconds,
- Serial Number – Device unique serial number,
- Mac Address – Ethernet mac address,
- Firmware Version – Controller firmware version,
- DHCP – Address setting by DHCP server
- IP Address – Device IP address (when DHCP is enable this slot is showing current IP address),
- Subnet Mask – Device subnet mask, (when DHCP is enable this slot is showing current IP mask),
- Default Gateway – Device default gateway Ip address, (when DHCP is enable this slot is showing current gateway IP address),
- Device Password – password for "platform" user,
- Nv Behavior If Empty – Non-Volatile components after copy output value (Options Leave 0, copy value from Default slot),
- Nv Free Numeric – number of available Numeric Non-Volatile components,
- NV Free Boolean – number of available Boolean Non-Volatile components,
- Save Interval – application auto save interval

6.2 Change IP address

The device has a built-in 2 Ethernet ports working in switch mode. By default, the new device's address is set to 192.168.1.123, subnet mask to 255.255.255.0 and default gateway to 192.168.1.1. The device offers the option to change the IP address at three levels: the WorkPlace application level, the website, and the device display (only devices with built-in display). All those levels allow to set static IP address or enable option to addressing by DHCP server.

Changing the IP address in the application

To change the IP address in the application, log into the device (admin or user with authorisation to change the address in the plat component). Then go to "plat" (Sedona -> App -> Service -> plat) and make changes to the slots: IP Address, Subnet Mask, Default Gateway, and save the application (App Component -> Save action). To make changes, restart the device using the Reboot action in the App component. The device shall restart with the new IP address.

Changing the IP address on the website

To change the IP address of the device, go to website by entering the current IP address in the URL field of any browser. Log in, using platform username and a numerical password from the "Device password" slot in the plat component (default - 1000). Then, go to IP Configuration tab and make appropriate changes in the following slots: IP Address, Subnet Mask, Gateway (Slot Mac Address is read-only), click Submit to save the application and restart the device using the Reboot button. The device will start up with the new IP address. Attention! After IP controller reboot, change URL address in browser.

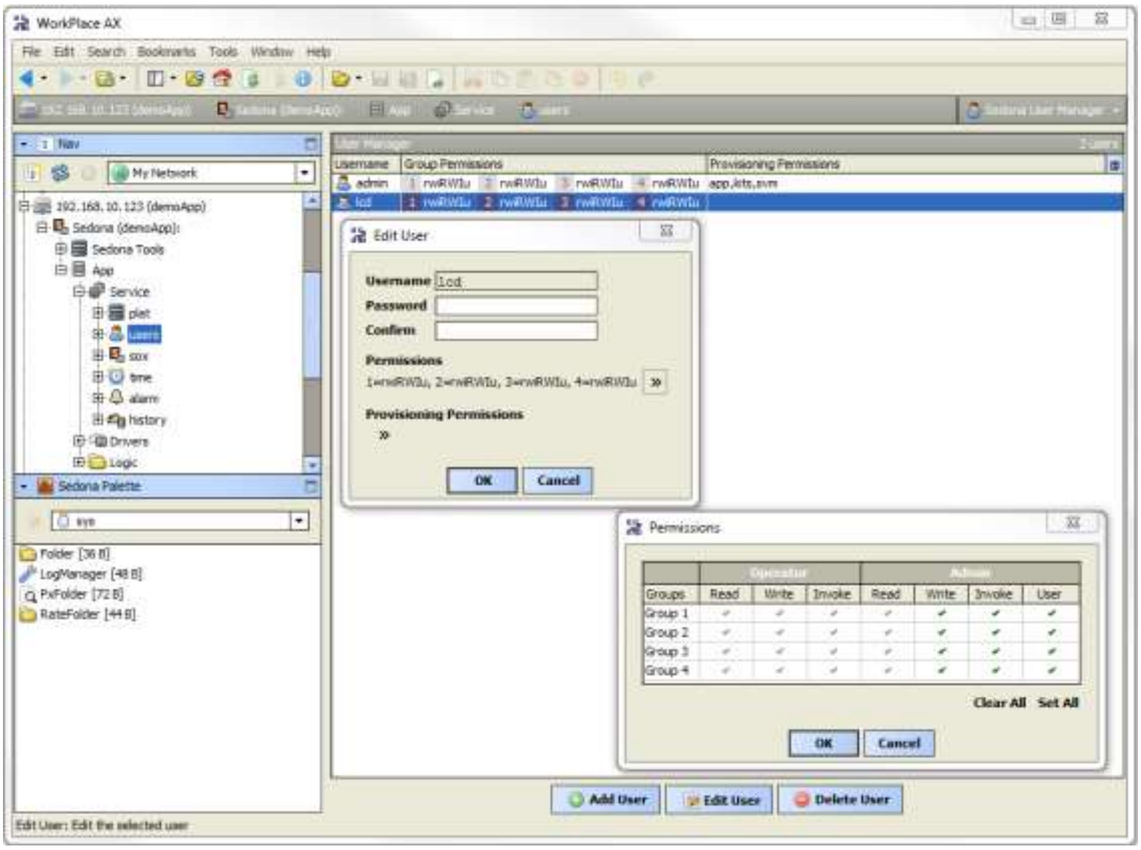
Changing the IP address using the display (only devices with built-in display)

To change the IP address from the display, hold the F1 function key down until the screen shows "Enter password". Then input a numeric password (slot "Device password", plat component, the default is 1000). In the system menu, use arrows on the right side to go to the Network Config line and press "Enter". Then select the appropriate line and press "Enter" again. The changes make the arrow on the right side (+ / -) move to the next position by pressing the "Enter" key. To validate the change, hold “Enter” down until transferred to the system menu and then select “Reboot”. The device shall start up with the new IP address.

6.3 Users service

Users Service is a service responsible for supporting users. Here is where you add and remove users as well as define their access rights to individual components. Each Sedona application component has a Meta slot, used to assign it to one or more groups. Sedona has 4 predefined groups.

Warning! In User service we can modify only Seodna users. User platform is controller operating system user and can by modify (password change) only from plat component, build in display or from controller web page.



Username	Group Permissions	Provisioning Permissions
admin	1 rwRWIu 2 rwRWIu 3 rwRWIu 4 rwRWIu	app,kits,svm
lcd	1 rwRWIu 2 rwRWIu 3 rwRWIu 4 rwRWIu	

Edit User

Username: lcd
Password:
Confirm:

Permissions
1=rwRWIu, 2=rwRWIu, 3=rwRWIu, 4=rwRWIu »

Provisioning Permissions
»

OK Cancel

Permissions

	Operator			Admin			
Groups	Read	Write	Invoke	Read	Write	Invoke	User
Group 1	✓	✓	✓	✓	✓	✓	✓
Group 2	✓	✓	✓	✓	✓	✓	✓
Group 3	✓	✓	✓	✓	✓	✓	✓
Group 4	✓	✓	✓	✓	✓	✓	✓

Clear All Set All

OK Cancel

Users can have the following types of rights:

- Operator Read – Read components, read values of operator properties,
- Operator Write – Change values of operator properties,
- Operator Invoke – Invoke operator actions,
- Admin Read – Read values of properties, read links, generate components links,
- Admin Write – Change values properties, add components, sort dub components, rename components, generate links to the components, delete links to the components,
- Admin Invoke – Invoke admin actions of the components,
- Admin User – User management (read, write, edit, delete).

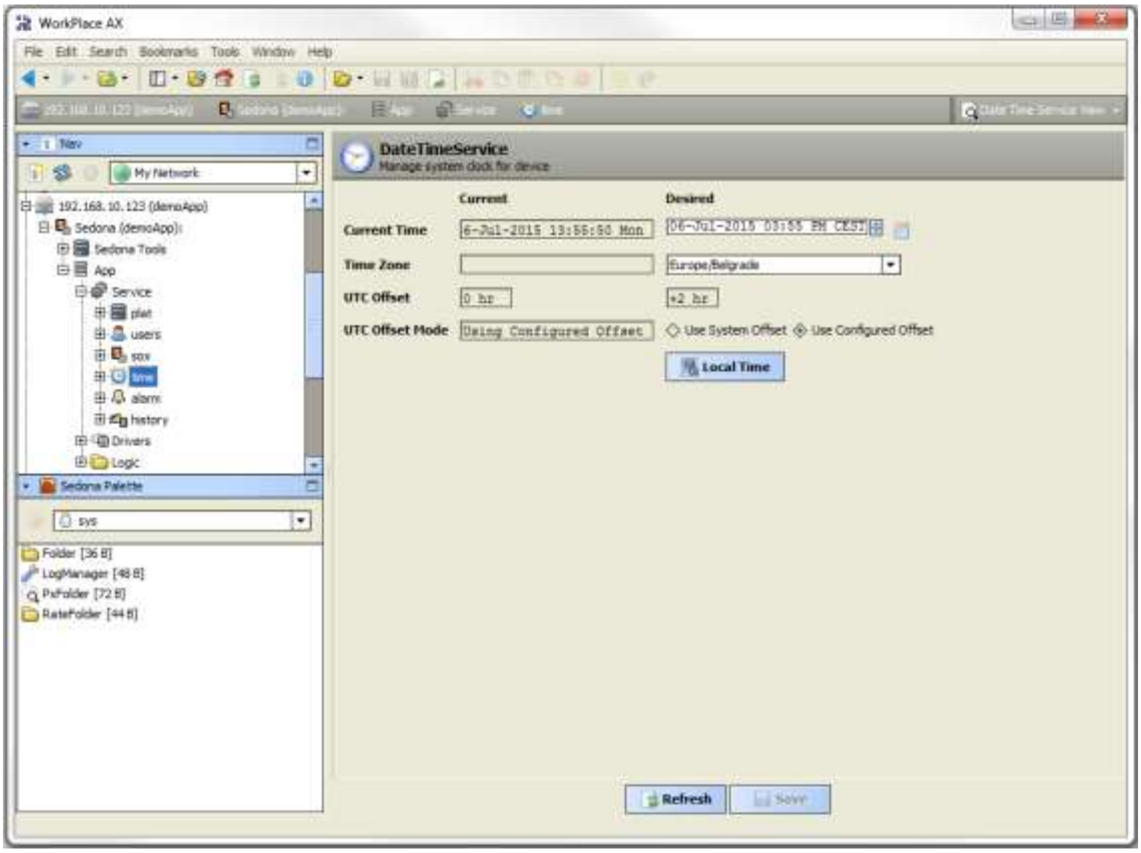
Provisioning Permissions:

- Can provision app – can read/write app.sab file,
- Can provision app – can read/write kits.scode file,
- Can provision app – can read/write SVM files.

For devices with built-in LCD, displaying components such as action editing/invoking is defined by assigning the given component to a group and defining access rights for user from this group.

6.4 Date service

Date service is a component, which services the built-in real time clock (RTC). The clock can by synchronized with a local PC's clock, or using iSMASox driver, with Niagara device.



DateTimeService
Manage system clock for device

	Current	Desired
Current Time	6-Jul-2015 13:56:50 Mon	06-Jul-2015 03:55 PM CEST
Time Zone		Europe/Belgrade
UTC Offset	0 hr	+2 hr
UTC Offset Mode	Using Configured Offset	○ Use System Offset ● Use Configured Offset

Local Time

7 NV components

NV Components (Non-Volatile) are those components whose value can be recorded in an EEPROM device's non-volatile memory. Whenever such a device is restarted or the power is down, the values of NV components remain saved. The device has three types of NV components, broken down by the type of variables they support. They include:

- Boolean variables - component NVBooleanWritable
- Integer variables - component NVIntegerWritable
- Numeric (float) variables - component NVNumericWritable

NV components can operate in the Auto mode (the "In" slot values are transferred to the "Out" slot) or in the Hand mode (the "Out" value is entered manually by the user and can't be changed by the application).

In iSMA-B-AAC-20 there is available 512 memory cells for numeric (float/numeric and integer) values and 1536 memory cells for boolean values. NVNumeric and NVInteger components use one numeric cell memory for Out value and one boolean memory cell for Auto / Hand switch mode. NVBoolean components use two boolean memory cells one for Out value and one for Auto / Hand switch mode.

Since the values of the components are not stored in the Sedona application but in the non-volatile memory of the device, when an application is copied between two devices, output values are not saved and will assume the values stored in the local EEPROM memory. To copy NV components to another device along with their values (e.g. setpoint), use global actions of the plat component:

Step 1 - Use global action "Copy From NV The Default / Copy From NV To User"

Step 2 - Save the application and copy it to another device

Step 3 - Use global action on the target device "Copy From Default To NV / Copy From User To Nv"

7.1 NVBooleanWritable

NVBooleanWritable is a component that stores the output value in non-volatile memory of an EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. The device iSMA-B-AAC20 has 1536 Boolean memory cells. The occupied space meter for EEPROM is embedded in the plat component. The

NVBooleanWritable component occupies two Boolean memory cells (value of the component, and position of the Auto / Hand switch).

The NVBooleanWritable has the following actions:

- Set - this option sets User slot and In slot if there is no link on In slot
- Set In Hand - this option sets the value on Out slot and blocks changing from any other slots,
- Set In Auto - this option switches off the Hand mode and sets Out slot according to the In slot's value.

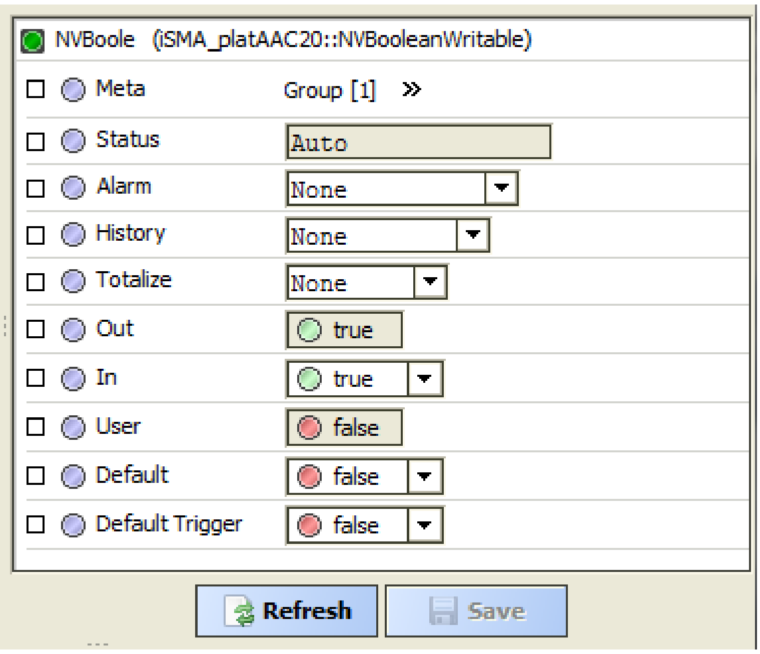
The NVBooleanWritable component allows adding extensions, such as:

Alarm Change Of State - to generate alarms when the component is in a particular state,

History - Record the value at a point in time, over a defined period of time, a change by a certain value, or in both of these cases,

Totalize - counting the time when the value of a point in a particular state.

The NVBooleanWritable component is also used to integrate Boolean variables from various sources. This is done using the "reverse following the link" function. The Out slot is connected to the In slots of various protocols, for example LCD or Modbus variable. When changing a value in one of the components, the device will perform the Set action on the NV component to synchronize the values in all the connected components.



NVBoole (iSMA_platAAC20::NVBooleanWritable)

□	Meta	Group [1] »
□	Status	Auto
□	Alarm	None
□	History	None
□	Totalize	None
□	Out	true
□	In	true
□	User	false
□	Default	false
□	Default Trigger	false

The NVBooleanWritable component has following slots:

- Status – Point actual status Auto / Hand,
- Alarm – Alarm extension,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

7.2 NVIntegerWritable

NVIntegerWritable is a component that stores the output value in non-volatile memory of an EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. The iSMA-B-AAC20 device has 1536 Boolean memory cells and 512 numeric cells. Space meter of the occupied EEPROM's memory is located in the plat component. NVIntegerWritable component occupies a single memory cell of the Numeric type (component value) and one memory cell of the Boolean type (switch position Auto/Hand).

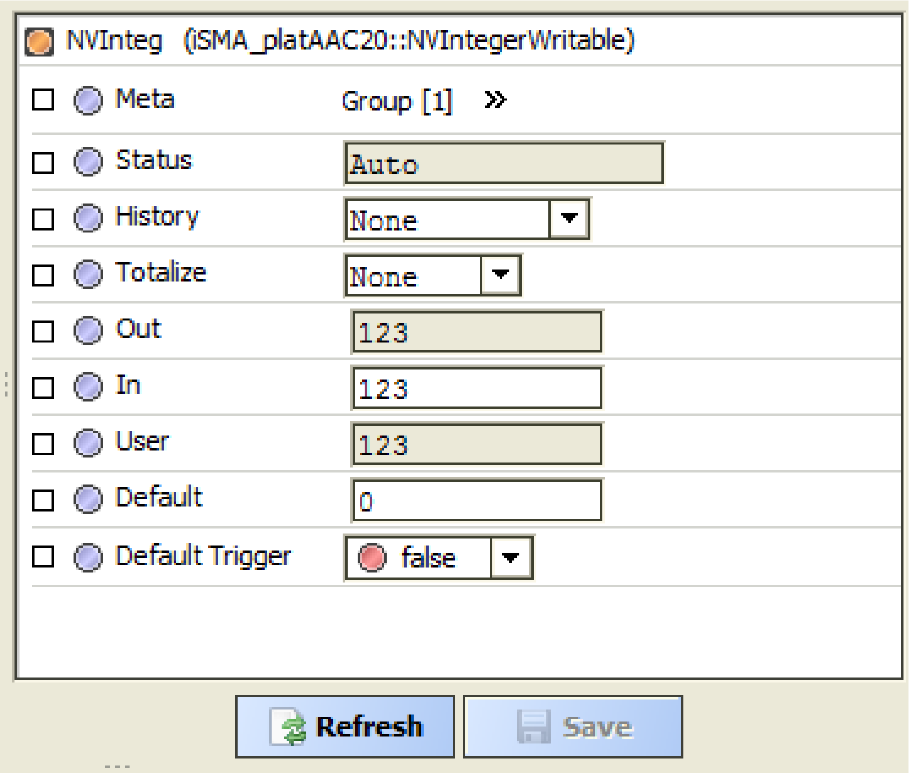
NVIntegerWritable Component allows you to add extensions such as:

History - Record the point in time in a defined period of time, to change a certain value, or in both of these cases,

Totalize - counting the time when the point value is not zero.

The NVIntegerWritable has following actions:

- Set – this option sets User slot and In slot if there is no link on In slot ,
- Set In Hand – this option sets value on Out slot and block changing from any other slots,
- Set In Auto – this option switches off Hand mode and sets Out slot according to In slot value.



NVInteg (iSMA_platAAC20::NVIntegerWritable)

□ Meta Group [1] »

□ Status Auto

□ History None

□ Totalize None

□ Out 123

□ In 123

□ User 123

□ Default 0

□ Default Trigger false

- Status – Point actual status Auto / Hand,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

7.3 NVNumericWritable

NVNumericWritable is a component that stores the output value in non-volatile memory of an EEPROM device. After rebooting the device or power failure, the component value is restored from this particular memory. The device iSMA-B-AAC20 has 1536 Boolean memory cells and 512 numeric cells. Space meter of the occupied EEPROM's memory is located in the plat component. The NVNumericWritable component occupies a single memory cell of the Numeric type (component value) and one memory cell of the Boolean type (switch position Auto/Hand).

The NVBooleanWritable has the following actions:

- Set – this option sets User slot and In slot if there is no link on In slot ,
- Set In Hand – this option sets value on Out slot and block changing from any other slots,
- Set In Auto – this option switches off Hand mode and sets Out slot according to In slot value.

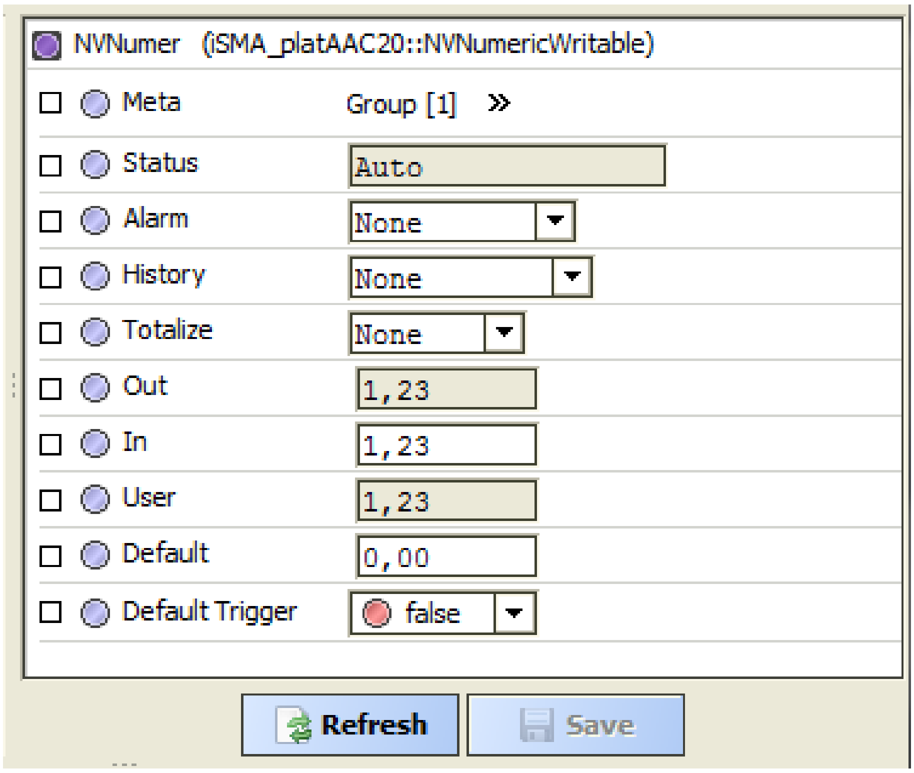
NVNumericWritable Component allows you to add extensions such as:

Alarm Out Of Range – generating alarms whenever the value of the component exceeds a given value,

History - Record the point in time in a defined period of time, to change a certain value, or in both of these cases,

Totalize - counting the time when the point value is not zero.

The NVNumericWritable component is also used to integrate Numeric (float) variables from various sources. This is done using the "reverse following the link" function. The Out slot is connected to the In slots of various protocols, for example LCD or Modbus variable. When changing a value in one of the components, the device will perform the Set action on the NV component to synchronize the values in all the connected components.



NVNumer (iSMA_platAAC20::NVNumericWritable)

□ Meta	Group [1] »
□ Status	Auto
□ Alarm	None
□ History	None
□ Totalize	None
□ Out	1,23
□ In	1,23
□ User	1,23
□ Default	0,00
□ Default Trigger	false

- Status – Point actual status Auto / Hand,
- Alarm – Alarm extension,
- History – History extension type,
- Totalize – Totalize extension,
- Out – Output slot,
- In – Input Slot,
- User – User value slot (setting by Set action),
- Default – Default value slot (setting by global command from plat action),
- Default Trigger – Copying trigger from Default to Out.

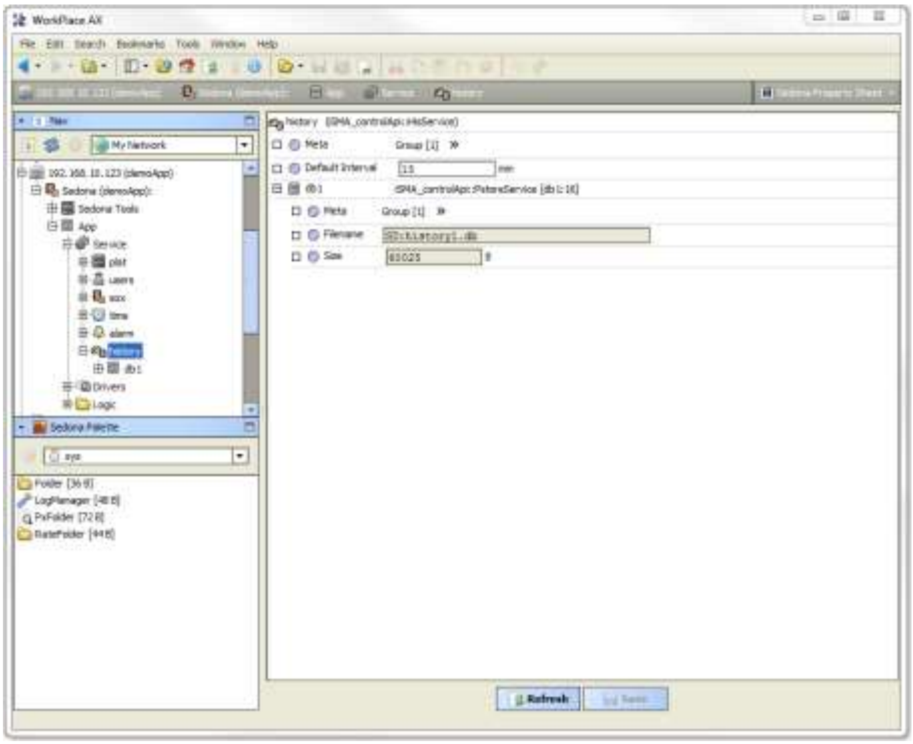
8 Historical

The iSMA-B-AAC20 device has an embedded historical extension for writing trends in microSD card memory. By default, the device is not equipped with an SD card, to use Historical Service SD card should be installed. The historical extension can be added only in NVBooleanWirtable, NVIntegerWritable and NVNumericWritable components. To add the extension you should choose one of the three available types in NV History slot components. Every historical has a predefined 2500 recordable samples. After reaching the limit, the oldest recordings are rewritten. While creating a historical, the device opens a database file on the SD card. Any such file can serve up to 100 extensions. Once the limit is reached, a new database file is opened.

8.1 History service

History service is a component, which is responsible for writing historical values in microSD memory. The History Service component is available from iSMA_controlApi kit palette and the component must be placed under the Service folder.

After you add any historical extension on NV component, a historical database will be created under the history service component in order to store NV points of historical components.



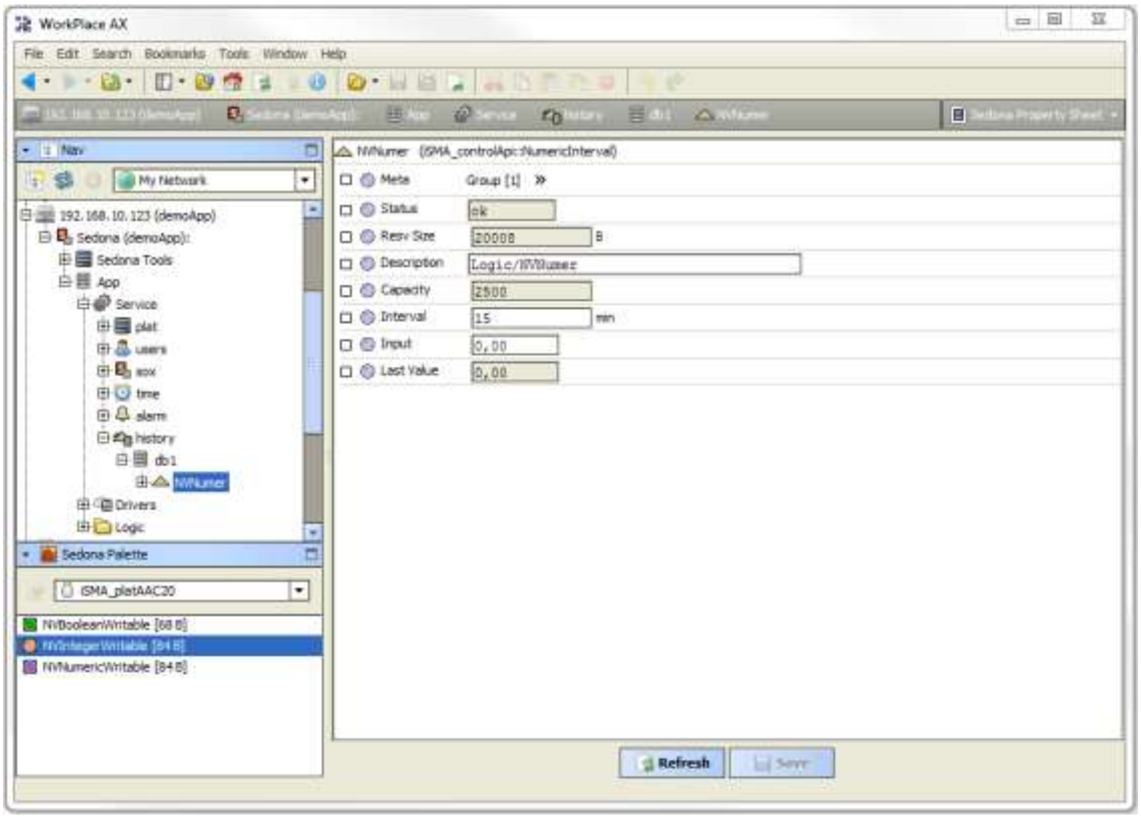
history (iSMA_controlApi::HisService)

Meta	Group [1]	
Default Interval	15	min
db1	iSMA_controlApi::FstoreService [db1:16]	
Meta	Group [1]	
Filename	SD:history1.db	
Size	65025	B

8.2 History extensions – Interval

Interval history extension is used to collect data at specified time intervals. When you choose interval option in NV component, a history interval will be created under the history and link with NV point Out slot will be made. For configuration and settings, please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.



NVNumer (iSMA_controlApi::NumericInterval)

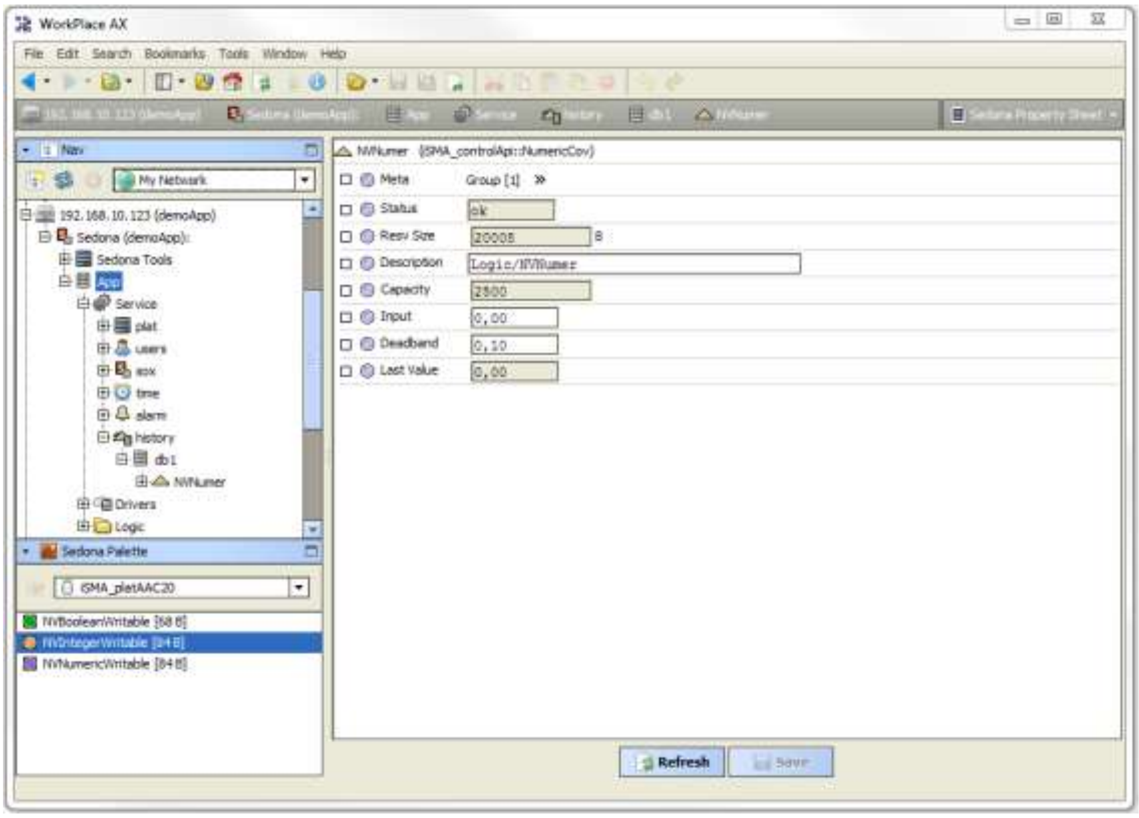
Meta	Group [1] »
Status	ok
Resv Size	20008 B
Description	Logic/NVNumer
Capacity	2500
Interval	15 min
Input	0,00
Last Value	0,00

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Interval – Saving time interval,
- Input – Current value,
- Last Value – Last saved value.

8.3 History extensions – COV

COV history extension is used to collect data whenever required changes has been made. When you chose COV option in NV component, a history COV component will be created under the history service and a link with NV point Out slot will be made. For configuration settings please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.



NVNumer (iSMA_controlApi::NumericCov)

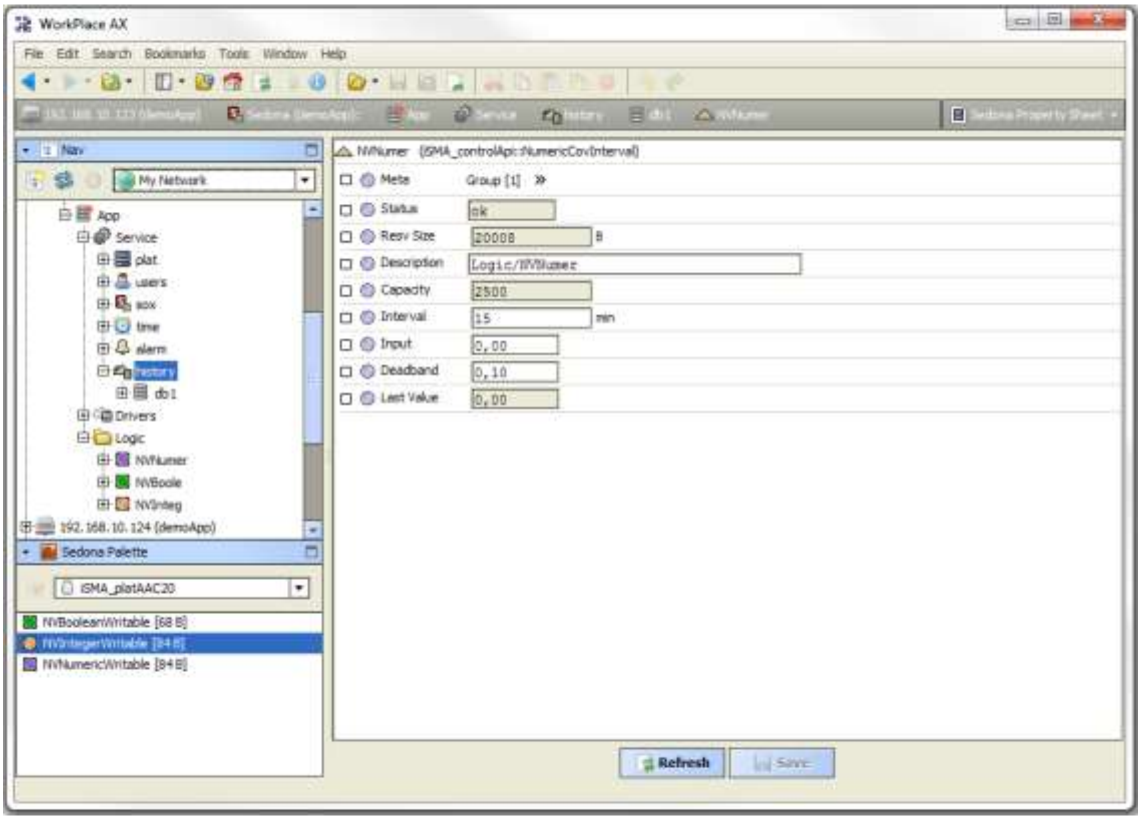
- □ Meta: Group [1] »
- □ Status: ok
- □ Resv Size: 20008 B
- □ Description: Logic/NVNumer
- □ Capacity: 2500
- □ Input: 0,00
- □ Deadband: 0,10
- □ Last Value: 0,00

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Deadband – Saving hysteresis,
- Input – Current value,
- Last Value – Last saved value.

8.4 History extensions – COV_Interval

COV_Interval history extension is used to collect data whenever required changes has been made or at specified time intervals (whatever happens first). When you chose COV_Interval option in NV component, a history COVInterval component is created under the history service and a link with NV point Out slot will be made. For configuration settings please refer to historical components under the history service.

WARNING! To delete or change historical extension type, don't remove it manually from the history service. Go to NV component, chose option "None" in history slot and save it (extension will be deleted by the system, along with all the database settings). If you want to change the type, chose "None" option first, do save and after that chose your required option.



NVNumer (iSMA_controlApi::NumericCovInterval)

□ Meta Group [1] »
□ Status ok
□ Resv Size 20008 B
□ Description Logic/NVNumer
□ Capacity 2500
□ Interval 15 min
□ Input 0,00
□ Deadband 0,10
□ Last Value 0,00

- Status – History component status,
- Resv Size – History memory reserve size,
- Description – Component description,
- Capacity – Number of available history records,
- Interval – Saving time interval,
- Deadband – Saving hysteresis,
- Input – Current value,
- Last Value – Last saved value.

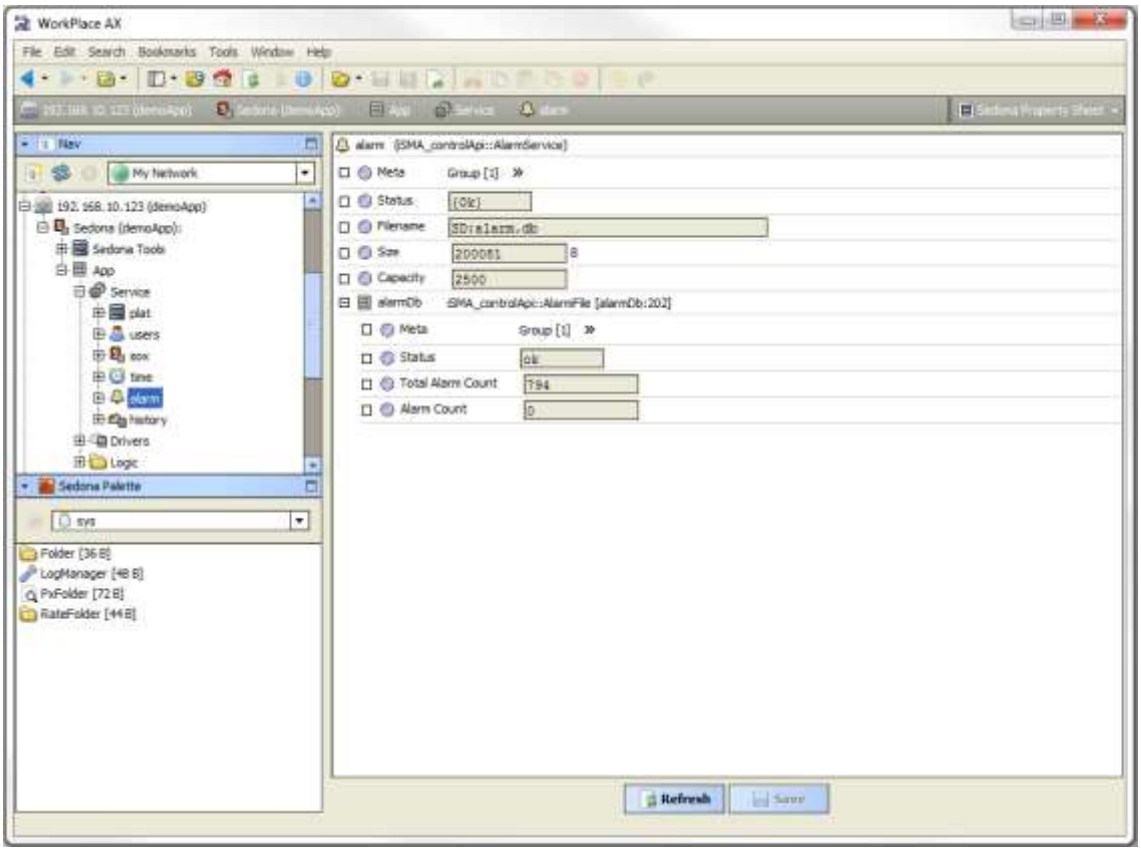
9 Alarms

The device has a service that allows recording events / alarms. Alarms can be generated off the NV-type components only. Alarms can be recorded only on the SD card. . By default, the device is not equipped with an SD card, to use Alarm Service SD card should be installed. When you create an extension of the alarm, a database file is created on the SD card. The device stores up to 2500 alarm events. If you exceed this number, a new event will overwrite the oldest records.

To add an extension to the component alarm you must select an appropriate option in the Alarm slot (Change Of State for Boolean components, and Out Of Range for Numeric components). After changing and saving the settings, an alarm extension is created under the component, where you are able to make all the settings associated with the process of generating an alarm event.

9.1 Alarm service

Alarm service is a component, which is responsible for servicing alarms events. The Alarm Service component is available from iSMA_controlApi kit palette and the component must be placed under Service folder.

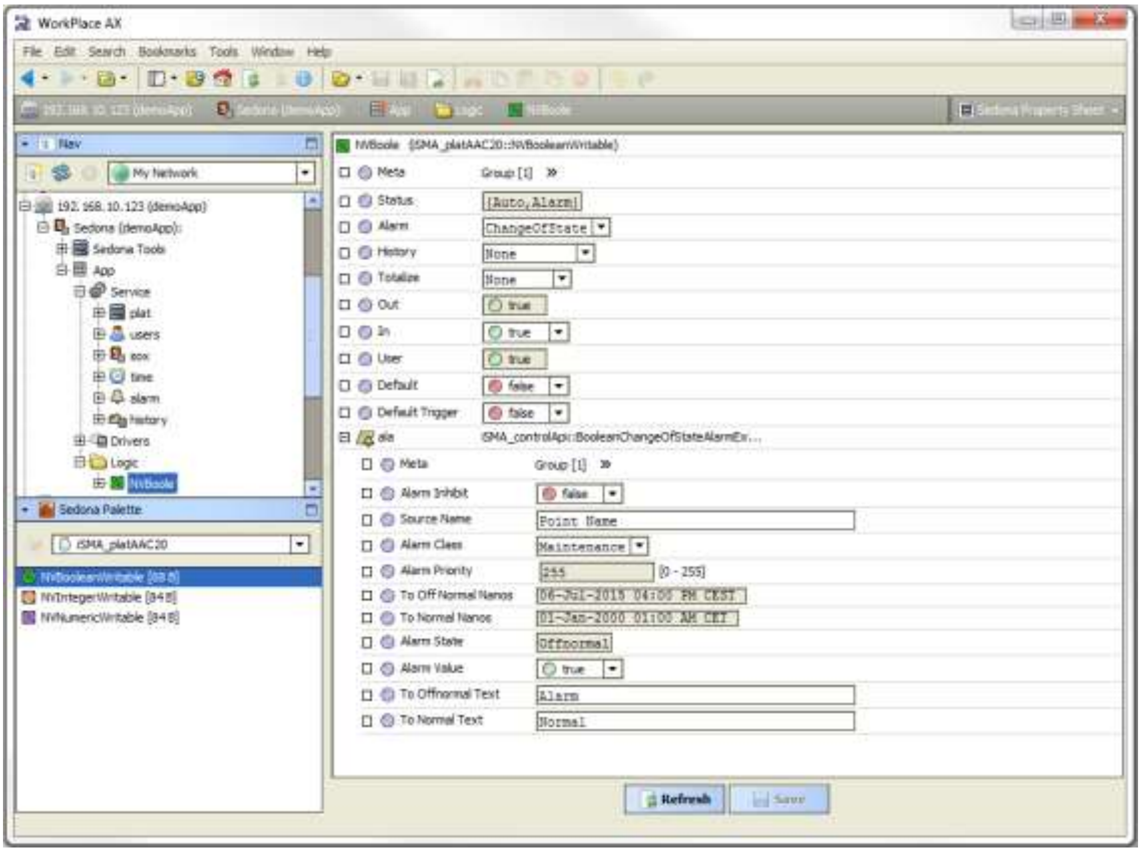


alarm (iSMA_controlApi::AlarmService)

□ Meta	Group [1] »
□ Status	{Ok}
□ Filename	SD:alarm.db
□ Size	200081 B
□ Capacity	2500
⊟ alarmDb	iSMA_controlApi::AlarmFile [alarmDb:202]
□ Meta	Group [1] »
□ Status	ok
□ Total Alarm Count	794
□ Alarm Count	0

9.2 Alarm Boolean points

The Boolean alarm extension can be added only in NVBooleanWritable components. To add the extension, choose “Change Of State” in Alarm slot and an alarm extension will be created automatically under NV component.



NVBoole (iSMA_platAAC20::NVBooleanWritable)

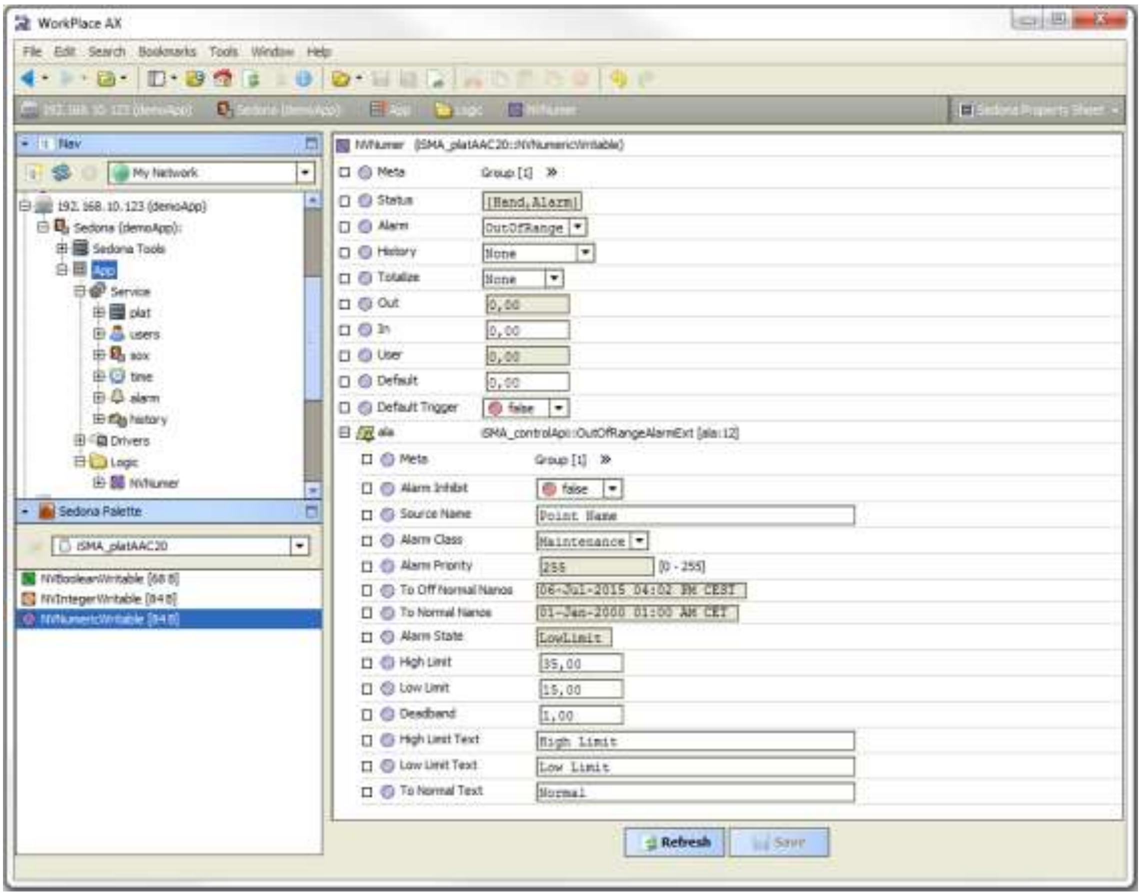
Meta	Group [1] »
Status	{Auto,Alarm}
Alarm	ChangeOfState
History	None
Totalize	None
Out	true
In	true
User	true
Default	false
Default Trigger	false
ala	iSMA_controlApi::BooleanChangeOfStateAlarmEx...
Meta	Group [1] »
Alarm Inhibit	false
Source Name	Point Name
Alarm Class	Maintenance
Alarm Priority	255 [0 - 255]
To Off Normal Nanos	06-Jul-2015 04:00 PM CEST
To Normal Nanos	01-Jan-2000 01:00 AM CET
Alarm State	Offnormal
Alarm Value	true
To Offnormal Text	Alarm
To Normal Text	Normal

Alarm extension slots:

- Alarm Inhibit – Enable / Disable alarm function,
- Source Name – Alarm name in alarms database,
- Alarm Class – Alarm class and priority definition, available options:
- Live Safety – priority 15,
- Critical – priority 75,
- Maintenance – 155,
- Alarm Priority – read only alarm priority number defined by alarm class slot (default value 255),
- To Off Normal Nanos – Time point of an alarm condition,
- To Normal Nanos - Time point of a normal condition,
- Alarm state – Actual alarm state,
- Alarm Value – Alarm generation value,
- To Offnormal Text – Description in alarm state,
- To Normal Text - Description in normal state.

9.3 Alarm Numeric points

The Numeric alarm extension can be added only in NVNumericWritable components. To add the extension, choose “Out Of Range” in Alarm slot, and an alarm extension will be created automatically.



NVNumer (iSMA_platAAC20::NVNumericWritable)

Meta	Group [1] »
Status	[Hand,Alarm]
Alarm	OutOfRange
History	None
Totalize	None
Out	0,00
In	0,00
User	0,00
Default	0,00
Default Trigger	false
ala	iSMA_controlApi::OutOfRangeAlarmExt [ala:12]
Meta	Group [1] »
Alarm Inhibit	false
Source Name	Point Name
Alarm Class	Maintenance
Alarm Priority	255 [0 - 255]
To Off Normal Nanos	06-Jul-2015 04:02 PM CEST
To Normal Nanos	01-Jan-2000 01:00 AM CET
Alarm State	LowLimit
High Limit	35,00
Low Limit	15,00
Deadband	1,00
High Limit Text	High Limit
Low Limit Text	Low Limit
To Normal Text	Normal

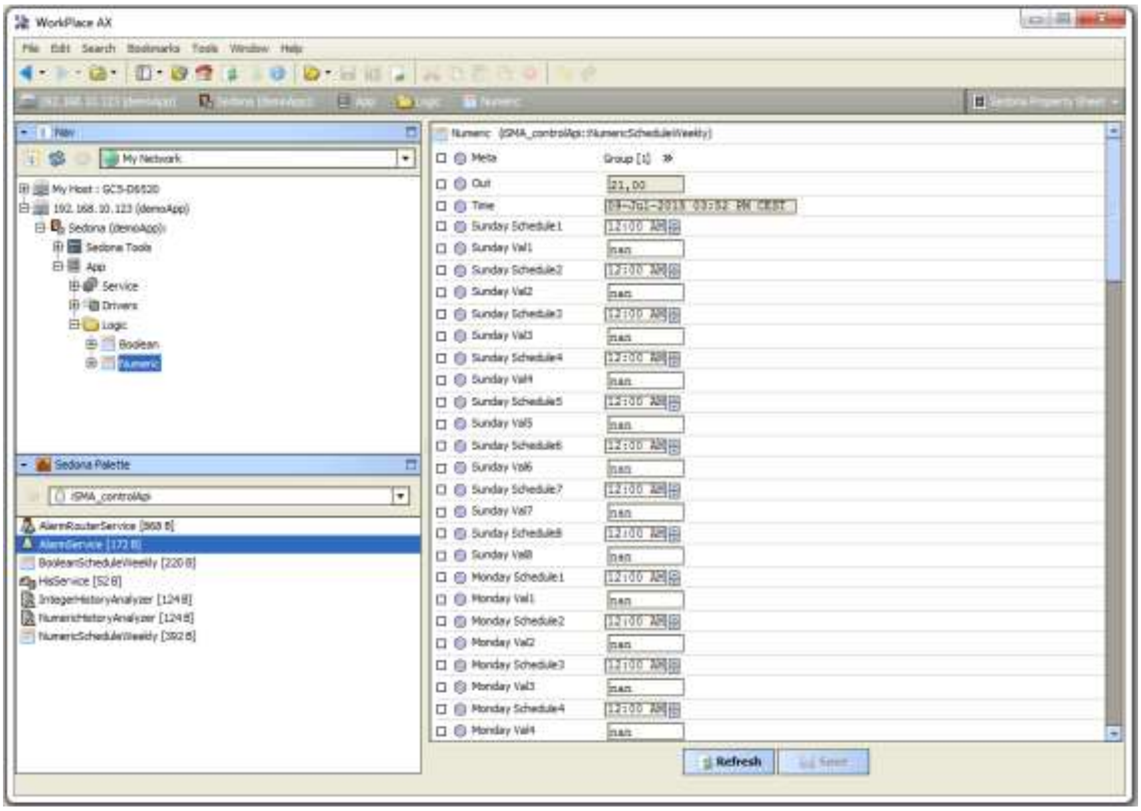
Alarm extension slots:

- Alarm Inhibit – Enable / Disable alarm function,
- Source Name – Alarm name in alarms database,
- Alarm Class – Alarm class and priority definition, available options:
- Live Safety – priority 15,
- Critical – priority 75,
- Maintenance – 155,
- Alarm Priority – read only alarm priority number defined by alarm class slot (default value 255),
- To Off Normal Nanos – Time point of an alarm condition,
- To Normal Nanos - Time point of a normal condition,
- Alarm state – Actual alarm state (Normal / Alarm),
- High Limit – High limit alarm generation value,
- Low Limit – low limit alarm generation value,
- Deadband – High and low limit dead band value

- High Limit Text – Description in alarm high state,
- Low Limit Text – Description in alarm low state,
- To Normal Text - Description in normal state.

10 Scheduler

Schedules in AAC20 controller are done using schedule components, as found in iSMA_controlApi kit (kit is install in default application). There are available two types of Schedulers Boolean Weekly for Boolean values and Numeric Weekly for numeric values. In both types there are available max 8 events for 7 week days. In component we define start time and value. If value is null (for Boolean) or nan (for Numeric) the event is discarding. For non-defined event scheduler will take value from Def Val slot.

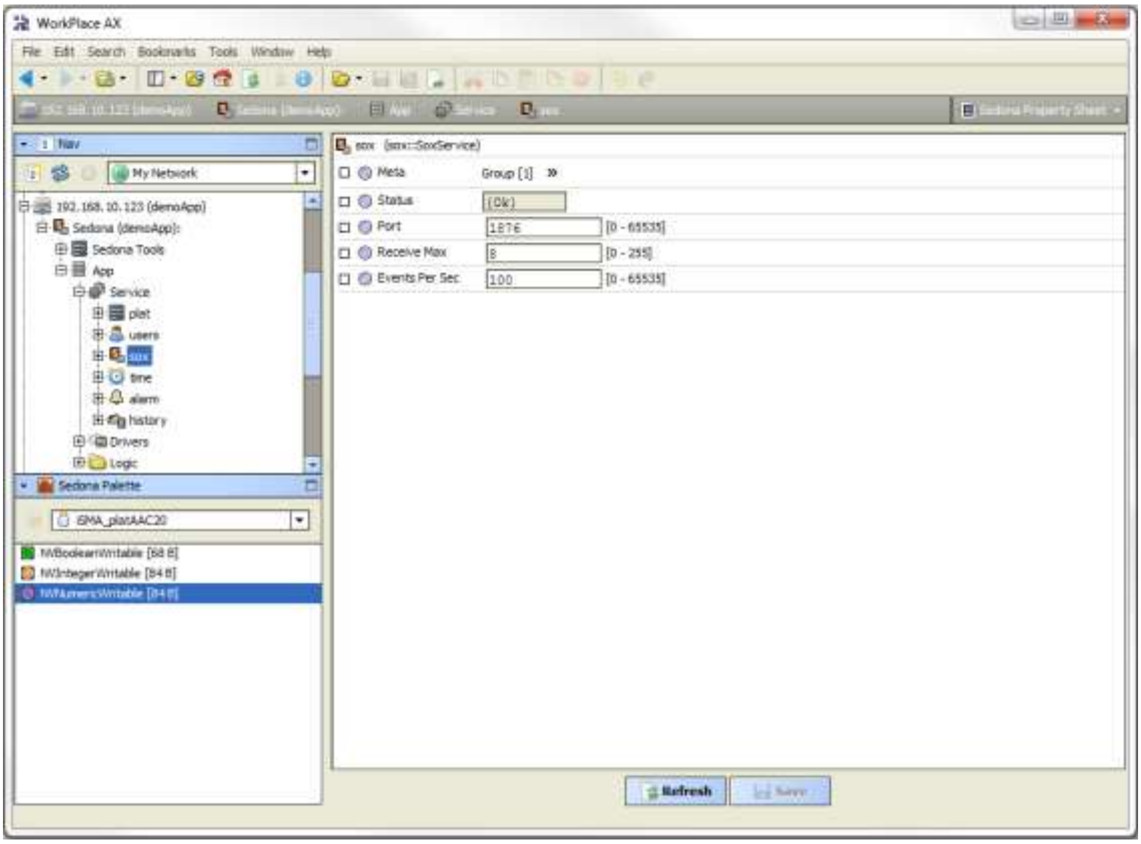


Numeric (iSMA_controlApi::NumericScheduleWeekly)

Meta	Group [1] »
Out	21,00
Time	09-Jul-2015 03:52 PM CEST
Sunday Schedule1	12:00 AM
Sunday Val1	nan
Sunday Schedule2	12:00 AM
Sunday Val2	nan
Sunday Schedule3	12:00 AM
Sunday Val3	nan
Sunday Schedule4	12:00 AM
Sunday Val4	nan
Sunday Schedule5	12:00 AM
Sunday Val5	nan
Sunday Schedule6	12:00 AM
Sunday Val6	nan
Sunday Schedule7	12:00 AM
Sunday Val7	nan
Sunday Schedule8	12:00 AM
Sunday Val8	nan
Monday Schedule1	12:00 AM
Monday Val1	nan
Monday Schedule2	12:00 AM
Monday Val2	nan
Monday Schedule3	12:00 AM
Monday Val3	nan
Monday Schedule4	12:00 AM
Monday Val4	nan

11 SOX protocol

Sox is the standard protocol used to communicate with Sedona devices. To serve Sox protocol Sox is also used by WorkPlace to program controller. Note that Sox is service type and it is execute after application components. So if there is little deference between scan period and Scan Time (see App component) the services don't have time execute and the programming interface can slow down. Sox is designed to be run over UDP/IP protocol (default port 1876).



sox (sox::SoxService)

□ Meta	Group [1] »
□ Status	{Ok}
□ Port	1876 [0 - 65535]
□ Receive Max	8 [0 - 255]
□ Events Per Sec	100 [0 - 65535]

- Status – Sox driver actual status,
- Port – Sox UDP port (default value 1876),
- Receive Max – Max of number of messages in receiving window,
- Events Per Sec – Max of number of async events (telegrams) sent per second

12 Local IO

The iSMA-B-AAC20 device has 22 built-in physical inputs and outputs:

8x Universal Inputs

All Universal Inputs have 16-bit resolution which support the following types of inputs:

- Temperature Input support the following types of sensors:10K3A1, 10K4A1, Carel 10K, 20K6A1, 2.2K3A1, 3K3A1, 30K6A1, SIE1, TAC1, SAT1, Pt1000, Ni1000, accuracy ± 0,1°C
- Voltage Input 0-10VDC o input resistance 100kΩ o accuracy ±1mV
- Resistive Input 0-1000kΩ • Dry contact input.

4x Digital Inputs

- Dry contact inputs
- High-speed pulse counter up to 100Hz

4/6x Analogue Outputs

All Analog Outputs are equipped with 12-bit ADC provides 10mV resolution and accuracy less than ±0,5%. They support the following output types:

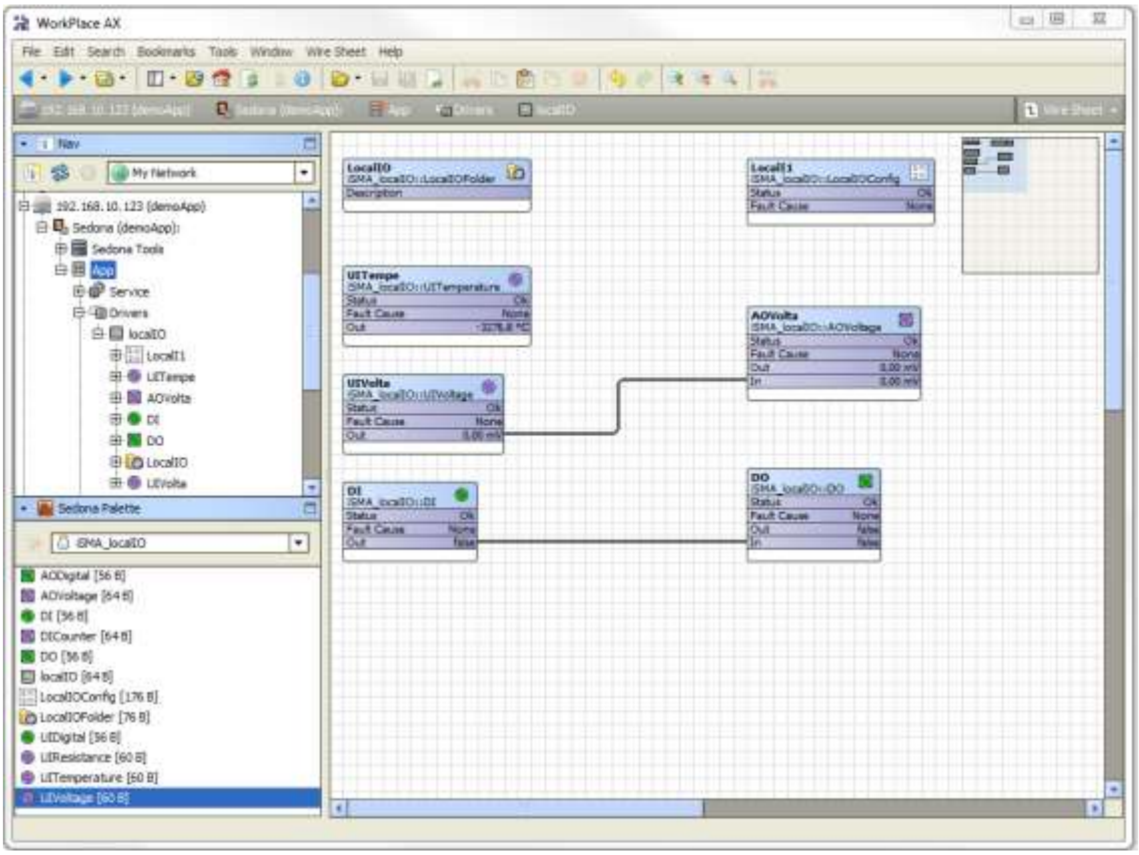
- Output 0-10V DC
- Digital Output with maximum load up to 20mA

4x Digital Outputs

- Relay Output max. 3A, 250V AC / 30 VDC

12.1 Adding Local IO components

Local IO's are supported by the components of the iSMA_LocalIO Kit. The components of this kit should be placed in a special folder localIO. Note! It is recommended to pass the signal inputs and outputs through NV components.

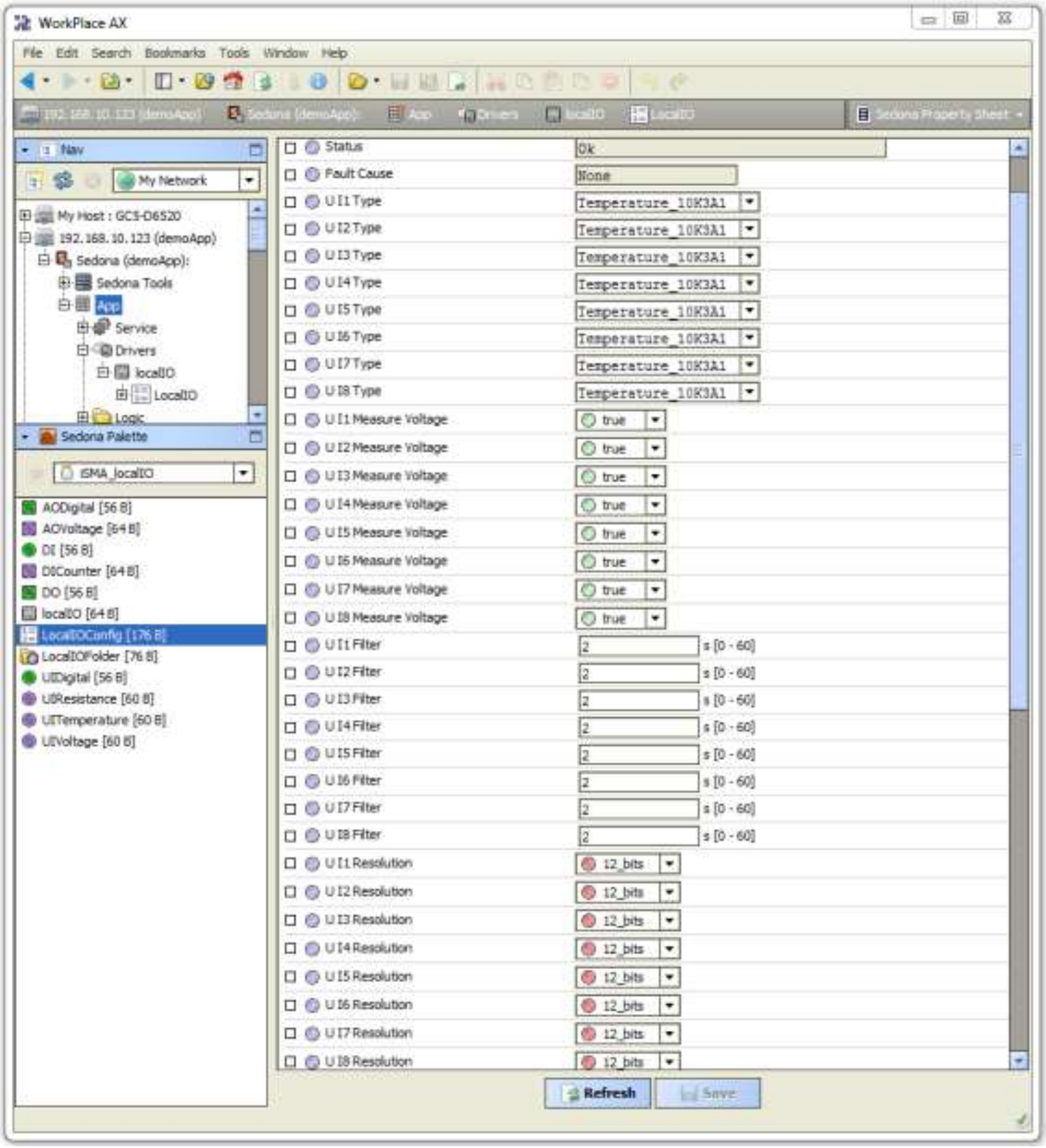


LocalIO kit consists of the following components:

- LocalIO – Main component servicing the physical I/O
- LocalIOFolder – Folder for grouping IO components,
- DI - Digital Input handling component
- UIResistance - Component for servicing Universal Input in the resistance read mode
- UITemperature - Component for servicing Universal Input in the temperature read mode (in order to get reliable readings, you must select the appropriate sensor type in the component LocalIOConfig)
- UIVoltage - Component for servicing Universal Input in the voltage read mode (in order to improve the reading, you must activate the component LocalIOConfig for voltage reading)
- UIDigital - Component for servicing Universal Input in the Dry Contact read mode
- DICounter - Component reading high-speed counter of Digital Inputs.
- DO - Component handling Digital Output
- AOVotlage - Component for servicing Universal Output in the voltage read mode

- AODigital - Component for servicing Universal Output in the digital mode (false - 0V, true - 10V)
- LocalIOConfig - Component for configuration of physical input/output parameters of the device.

12.2Local IO Config



Status	Ok
Fault Cause	None
UI1 Type	Temperature_10K3A1
UI2 Type	Temperature_10K3A1
UI3 Type	Temperature_10K3A1
UI4 Type	Temperature_10K3A1
UI5 Type	Temperature_10K3A1
UI6 Type	Temperature_10K3A1
UI7 Type	Temperature_10K3A1
UI8 Type	Temperature_10K3A1
UI1 Measure Voltage	true
UI2 Measure Voltage	true
UI3 Measure Voltage	true
UI4 Measure Voltage	true
UI5 Measure Voltage	true
UI6 Measure Voltage	true
UI7 Measure Voltage	true
UI8 Measure Voltage	true
UI1 Filter	2 s [0 - 60]
UI2 Filter	2 s [0 - 60]
UI3 Filter	2 s [0 - 60]
UI4 Filter	2 s [0 - 60]
UI5 Filter	2 s [0 - 60]
UI6 Filter	2 s [0 - 60]
UI7 Filter	2 s [0 - 60]
UI8 Filter	2 s [0 - 60]
UI1 Resolution	12_bits
UI2 Resolution	12_bits
UI3 Resolution	12_bits
UI4 Resolution	12_bits
UI5 Resolution	12_bits
UI6 Resolution	12_bits
UI7 Resolution	12_bits
UI8 Resolution	12_bits

LocalCOnfig is a component designed to configure the physical input/output of the device. It consists of the following slots:

- UIx Type - The type of temperature sensor connected to Universal Input; table stored in the device allows you to convert the value of sensor resistance into temperature. Available options: Disable_Resistance, Temp_10K3A1, Temp_10K4A1, Temp_10K, Temp_20K6A1, Temp_2_2K3A1, Temp_3K3A1, Temp_30K6A1, Temp_SEI1, Temp_TAC1, Temp_SAT1, Temp_Pt1000, Temp_Ni1000.
- UIx Measure Voltage - enables / disables the measurement of the voltage at the device's universal input,
- UIx Filter - The time constant of the low pass filter (to eliminate signal noise)
- UIx Resolution – Measurement resolution of a Universal Input. WARNING!16-bit value increases the time to read a single entry. It should be used only when you connect sensors PT1000 and NI1000.
- AOx Type - Analog Output Mode: Voltage 0-10V or PWM
- Default State Of Analog Output – The Analogue Output default value after your reboot the controller expressed in mV.
- Default Digital State Of Analog Output - The default value of the Analogue Output, working in digital mode, when you reboot the device.
- Default State Of Digital Output - Digital Output default when you reboot the device.
- Note: Adding the LocalIOConfig component to the application shall disable the possibility of I / O configuration from the Web Site and Modbus array.
- Any settings of the LocalIOConfig component are carefully stored in the component and can be transferred to other devices (quick setup of multiple devices).

13 One Wire

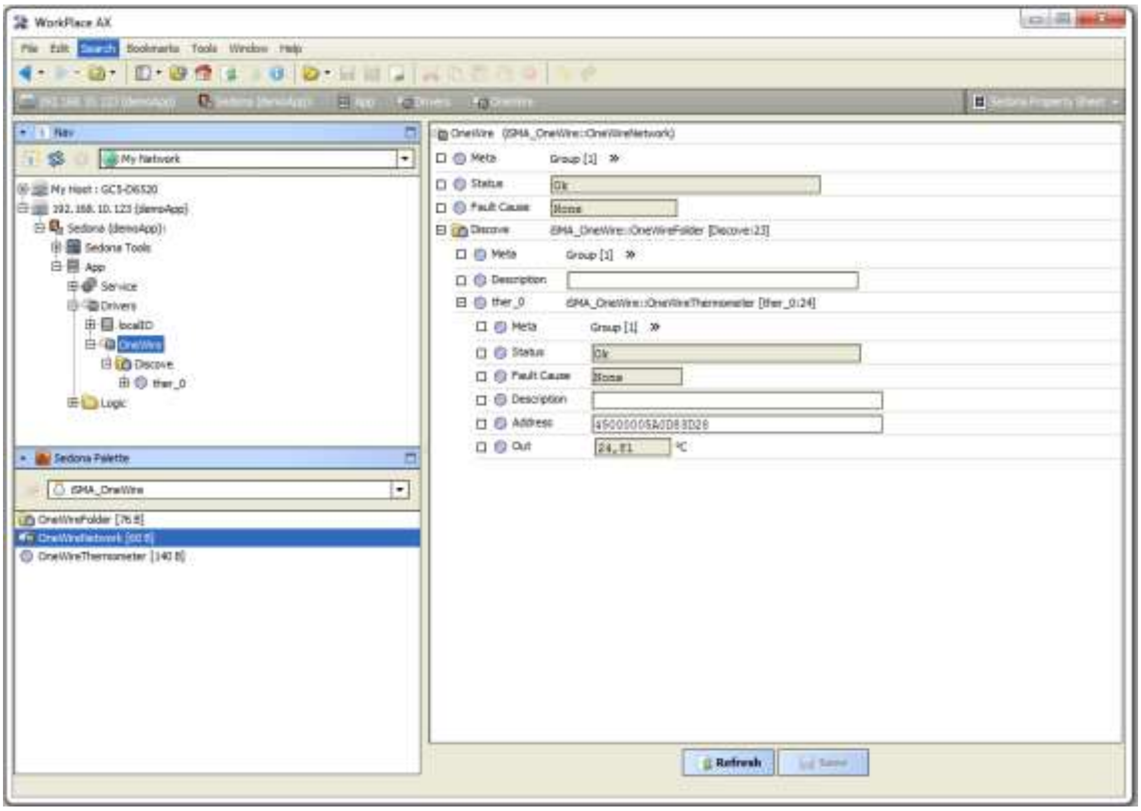
One Wire is a special protocol for device make by Dallas company. Temperature sensors use DS18D20 chip. The OneWire bus use 3 wire cable (+5 VCC, Data, GND) max 100m long (it is recommended to use shorter distance) and up to 32 sensors. All iSMA-B-AAC20 devices are equipment with OneWire port which use two Analog Outputs AO5 and AO6.

WARNING! Before connecting OneWire device to controller please first add OneWire network. It will block the possibility of increasing voltage above 5 V and protect OneWire sensor against damage.

13.1 Adding OneWire network

To use OneWire protocol you need to install iSMA_OneWire kit on controller using Kit Manager (see Kit Manager chapter). After that you need to place OneWireNetwork component under Drivers folder.

Note that if you are using LocalIO in LocalOIConfig component in slots AO5 Type and AO6 Type value will change to OneWire and it can't be changed manually. If you are using OneWire you can't use AO5 and AO6 Analog Outputs.



OneWire (iSMA_OneWire::OneWireNetwork)
□ Meta Group [1] »
□ Status Ok
□ Fault Cause None
⊟ Discove iSMA_OneWire::OneWireFolder [Discove:23]
□ Meta Group [1] »
□ Description
⊟ ther_0 iSMA_OneWire::OneWireThermometer [ther_0:24]
□ Meta Group [1] »
□ Status Ok
□ Fault Cause None
□ Description
□ Address 45000005A0D83D28
□ Out 24,81 °C

13.2Discovering OneWire sensors

OneWire network has Discover action (under right mouse button). This action will with automatically searched and detect all sensors connected to the bus. It will also create Discover folder and place there all found sensors. All sensors have unique address assigned by manufacturer (slot Address). There is also Description slot to describe sensor function / place using max 20 characters. All sensors components can by move and groped in OneWireFolders (folders must be also placed under OneWireNetwork component).

13.3Adding sensors manually

To add sensor manually you have to place OneWireThermometer component form iSMA_OneWire kit palette and place it under OneWireNetwork. Then you have to enter sensor address and save application. If the connection and address are correct, status slot should have ok value and Out slot should have sensor temperature in Celsius degrees.