Cylon Unitron UC32 Driver for Tridium Niagara

User Guide

Contents

1 Introduction

Cylon [1] Unitron UC32 is a well-known BMS system, which includes a wide range of programmable controllers for HVAC control and building automation. There are two types of Unitron devices: * field controllers for direct digital control * communication controllers for information exchange, global schedules and LAN connection

Cylon driver for Niagara is designed to enable communication between Cylon controllers and Tridium Niagara powered devices. It provides an efficient solution for retrofit of building management systems, facilitates seamless integration of multiple protocols and allows convenient makeover of front-end software.

2 Requirements

- Niagara-powered device with software v3.8 (AX), v4.1 (N4) or later, including Jace2, Jace3, Jace6, Jace8000, Supervisor or their OEM versions [2]
- Cylon driver module and license

Niagara Supervisor or Jace is connected a communication controller UC32.net via TCP/IP network. The communication controller is connected to field controllers via RS-485 port.

3 Quick Start

1. Copy cylon.jar (AX) or cylon-rt.jar (N4) to both Jace and WorkPlace
2. Add **Cylon Network** to Jace station
3. Enter license code in **License** property under **Cylon Network**
4. Enter IP address and port in **Tcp Config** property; default port number is 4950
5. Open **Cylon Network** and **Discover** devices – it should find one communication controller and field controllers if available
6. Add controllers to the station
7. *For each field controller* open **Points** extension, **Discover** points or **Import File**, add points to the station
8. *For each global schedule* create a **Boolean Schedule** in the station, then open communication controller **Schedules** extension, use **Import File** to add **Schedule Export** to the station and select **Boolean Schedule** in **Supervisor Ord**; run **Read from controller** action and check if **Boolean Schedule** is filled.

4 Cylon Network

Cylon Network contains many standard Niagara properties, as well as few Cylon-specific:

- **License** – the code which allows driver to run on your Host ID
- **Tcp Config / Ip Address** – IP and port (4950 by default) of UC32.net communication controller
- **Tcp Config / Site** – Unitron site number.
- **Tcp Config / UC32net Address** – Communication controller address. Both **Site** number and **UC32net Address** could be found in Unitron project or on UC32.net display.

[1] All trademarks or registered trademarks are property of their respective owners
[2] If support for older Niagara versions is required, please contact the vendor

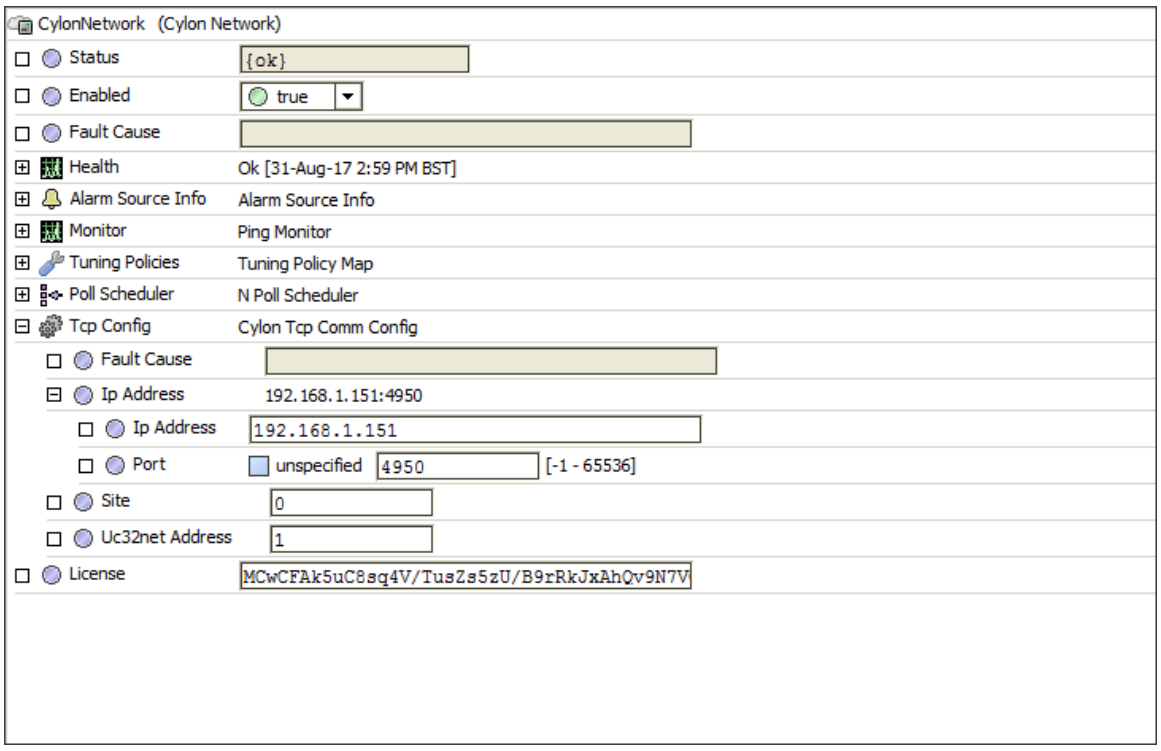

Figure 1: Cylon Network properties

5 Cylon Devices

There are two types of Unitron devices:

1. **Cylon Comm Device** – a communication controller with LAN connection. There is only one per network and always with address 0. This device has **Schedules** extension for global schedules.
2. **Cylon Device** – a field controller with **Points** extension.

6 Cylon Points

Unitron field controllers points are identified by a type and an address from 1 to 1024 (some models less). Types could be:

1. Analog hardware
2. Digital hardware
3. Analog software
4. Digital software

Hardware points are physical inputs and outputs, depending on controller model and configuration. Software points are variables, they could be writable – also called setpoints – or read-only.

Writing into software point *overwrites* its value, i.e. “old” value will be lost. Writing into hardware point *overrides* its value, so then one could return it back to the previous automatic value.

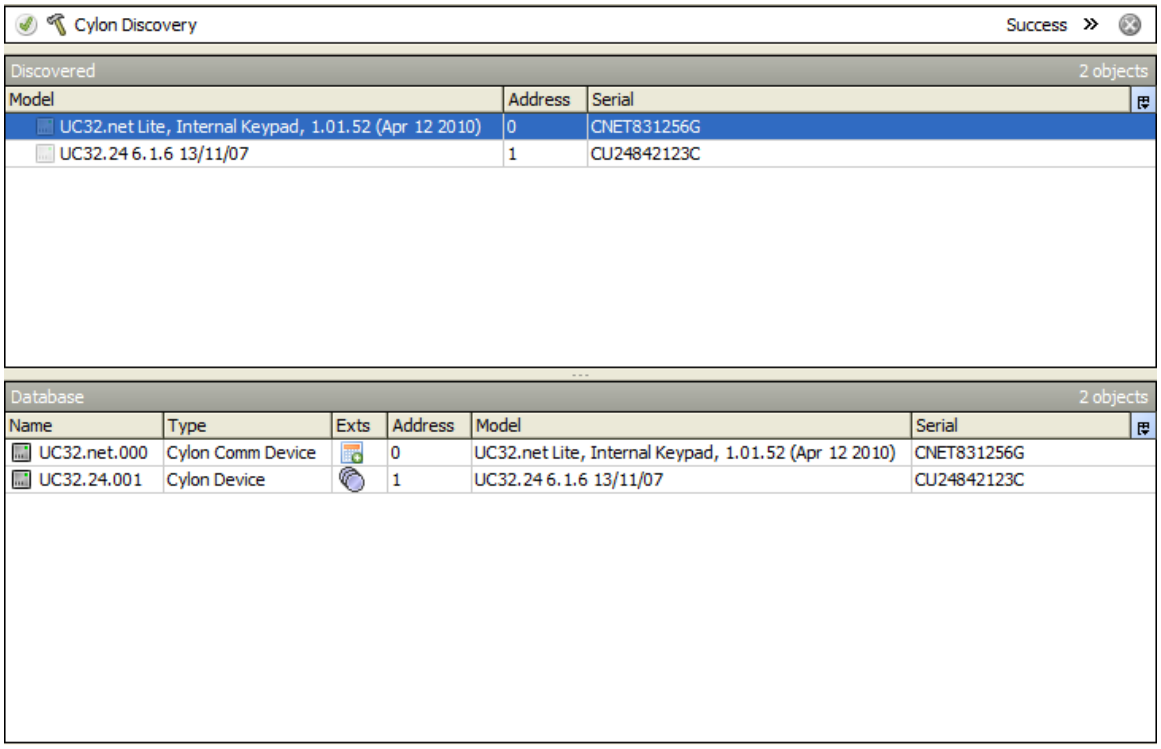

Figure 2: Cylon device discovery

Each point is polled by Jace when it is required. Polling frequencies are specified for each point individually and reference polling rated defined in Poll Scheduler as in the most Niagara drivers. This allows to find an optimal solution when reading a lot of points as often as suitable.

There are two ways how to get Unitron points into Niagara: **Import File** and **Discover**.

6.1 Point Import from CSV File

Import File is a preferred method as it creates all points with names, units, limits and boolean facets. In order to utilise it, one should have access to Unitron project and export its points to CSV files in Cylon Database Interface software. Then in device **Points** extension press Import File and choose exported CSV.

6.2 Point Discovery

Discover is another method, which allows to get some point information directly from device. It could be used if Unitron project files are unaccessible. There are few limitations in this method: read-only software points could not be discovered and should be created manually; no point names or facets are available. In general this method still allows to retrieve enough information to build all graphics, although it will require more investigation and engineering.

7 Schedules

Communication controllers store global time schedules and send them to field controllers. Each schedule occupy a number of *blocks* in its memory. Each *block* represent one weekly period for each

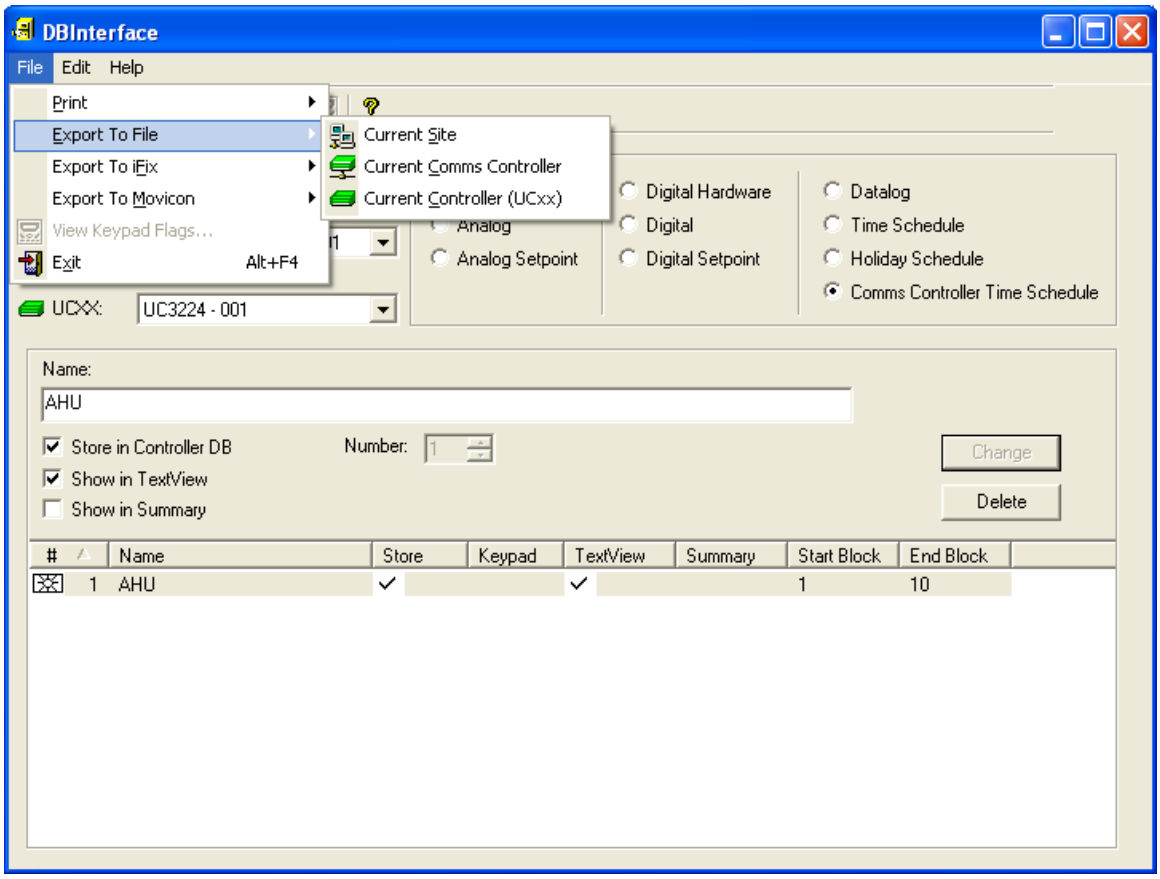

Figure 3: Export points to CSV file

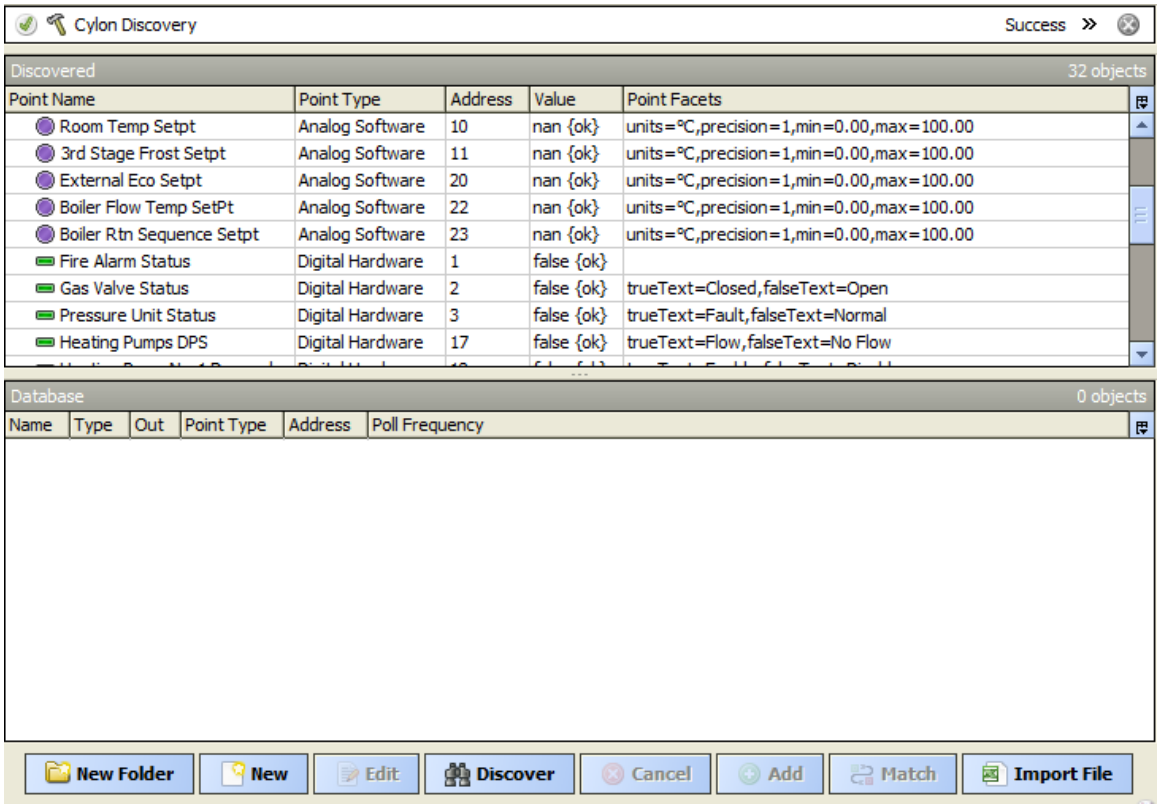

Figure 4: Import points from CSV file

day or one special date range.

Niagara could read schedule from Cylon device, map it into a standard **Boolean Schedule** and then write it to controller when necessary. These are the steps:

1. Create **Boolean Schedule** somewhere in Niagara station.
2. Open communication controller **Schedules** extension, **Import File** or create schedule manually (you will have to know start and end blocks, which this schedule occupies).
3. Select **Boolean Schedule** in **Supervisor Ord** property.
4. When the schedule is created, right-click on it run **Read from controller** action. Open **Boolean Schedule** and check if it is filled properly.
5. Change some periods in **Boolean Schedule**, save it and then **Export** it back to controller. It could be done with **Execute** action, **Export** button, periodically using **Execution Time** trigger or by demand.

Cylon global schedules are not identical to Niagara **Boolean Schedules**, so they can't be mapped fully. Because of these limitations in **Boolean Schedule** only **Day** and **Date Range** special events could be used, both with exactly one time period; dates could not have wildcards (*).

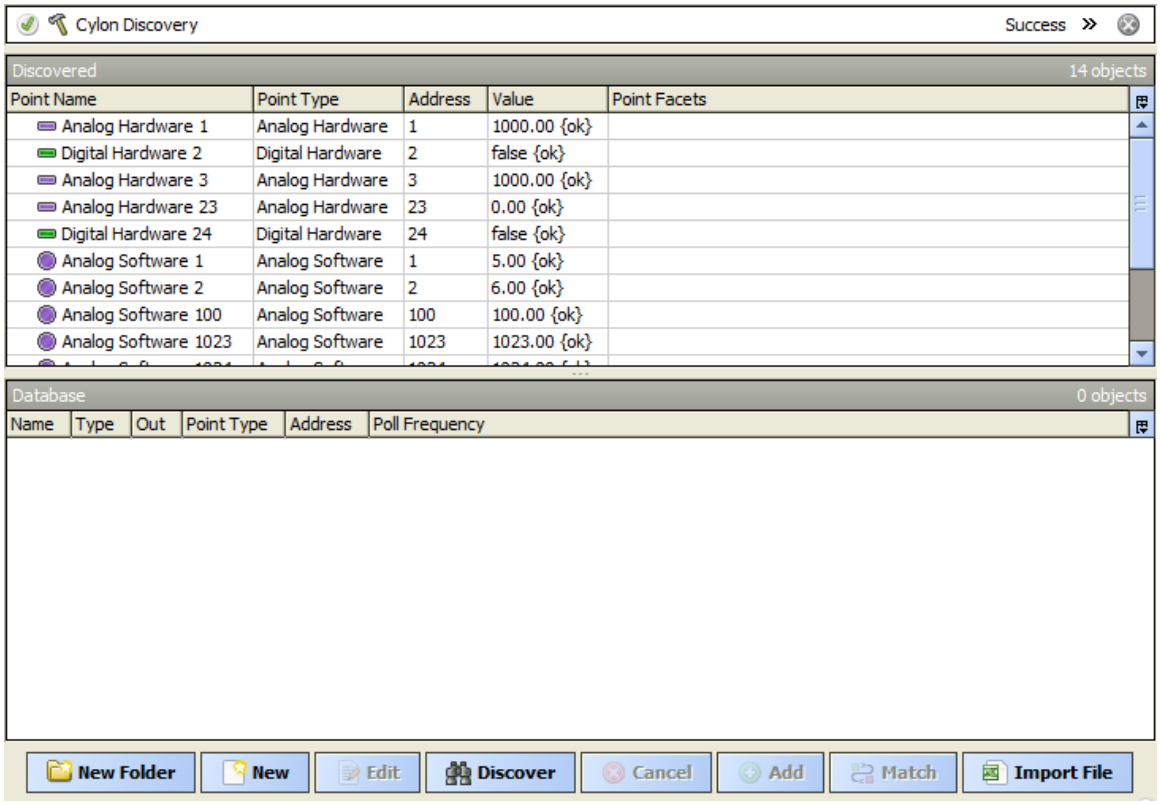

Figure 5: Point discovery

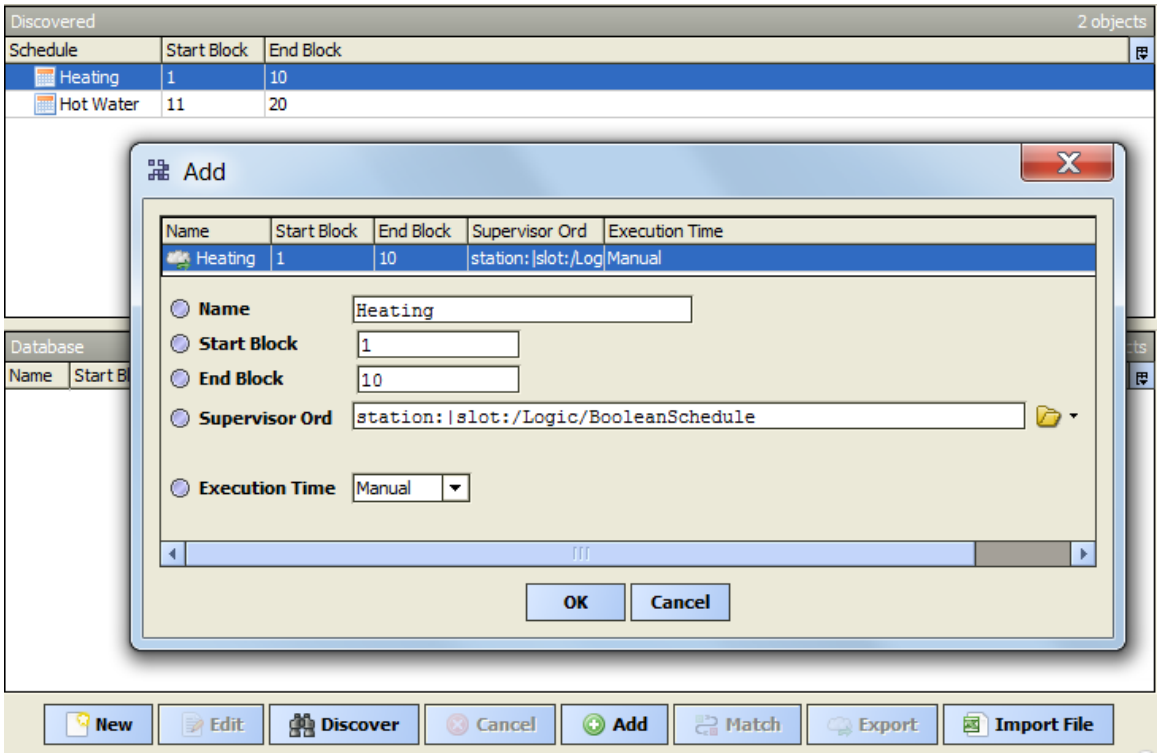

Figure 6: Schedule creation

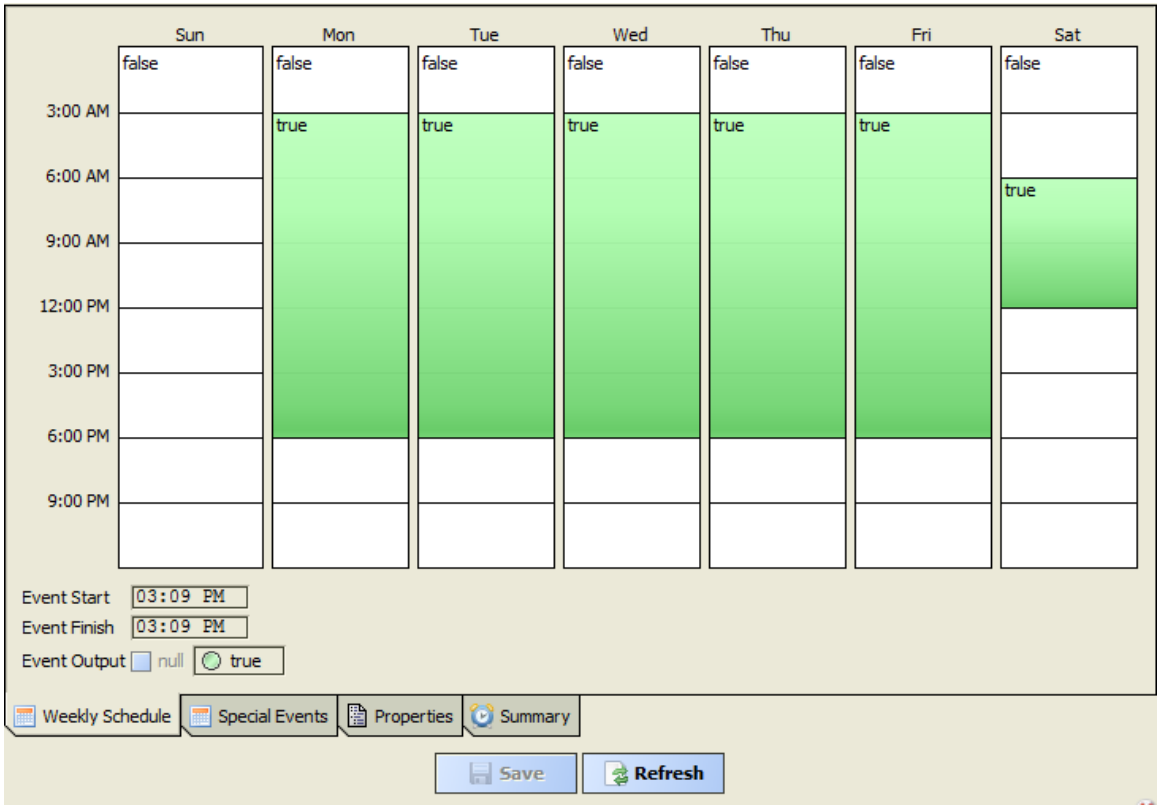

Figure 7: Weekly schedule

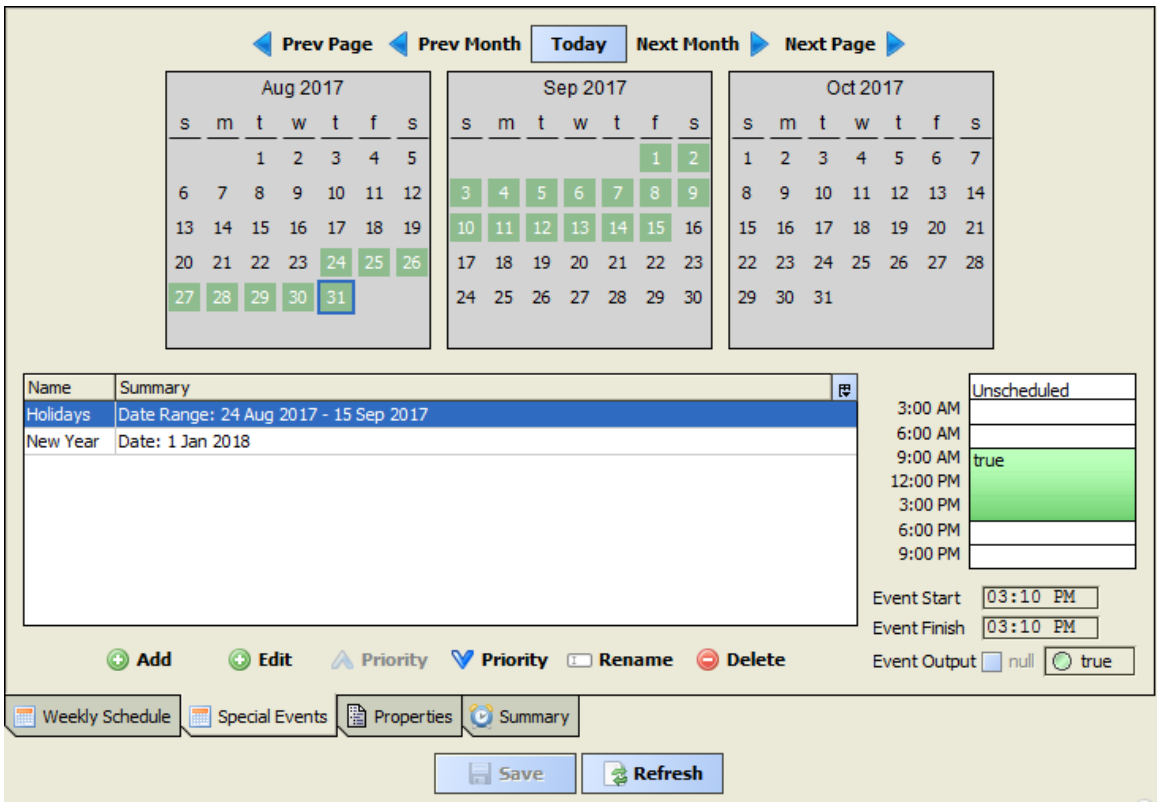

Figure 8: Special days schedule