



---

# **LONMARK®**

# **Functional Profile:**

# **Calendar**

## **SFPTcalendar**

---

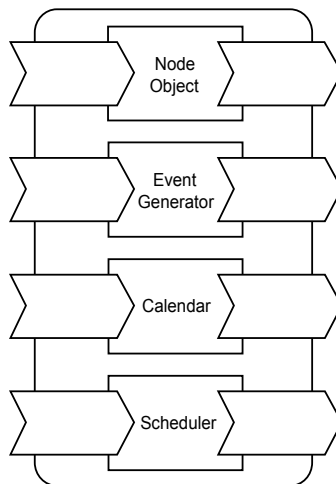
## Overview

The Calendar functional profile defines a standardized interface for identifying active and inactive schedules based on dates. Schedules may become active on a particular date, on a range of dates, or on a repeating interval. A schedule is identified by a schedule index on a device, and by an optional schedule name that is used to communicate schedule status between devices.

A schedule may be *active* or *inactive*. A schedule changes to active or inactive at midnight. Multiple schedules may be active on any given day. The output of the Calendar updates the status of all schedules controlled by the Calendar upon reset, time change, daily, and upon request. The optional output of the Calendar consists of *date events*, where each date event specifies the name of a schedule as well as the number of days that the schedule has been active or inactive, and the number of days until the next time the schedule will be active or inactive.

The date event output of a Calendar functional block is typically used as an input to one or more Scheduler functional blocks on other devices. A Calendar functional block may also provide schedule status updates directly to Schedulers on the same device.

The Calendar functional profile makes use of optional interfaces through Node Object functional block as described in this document.



**Figure 1** Device Concept

---

## Requirements

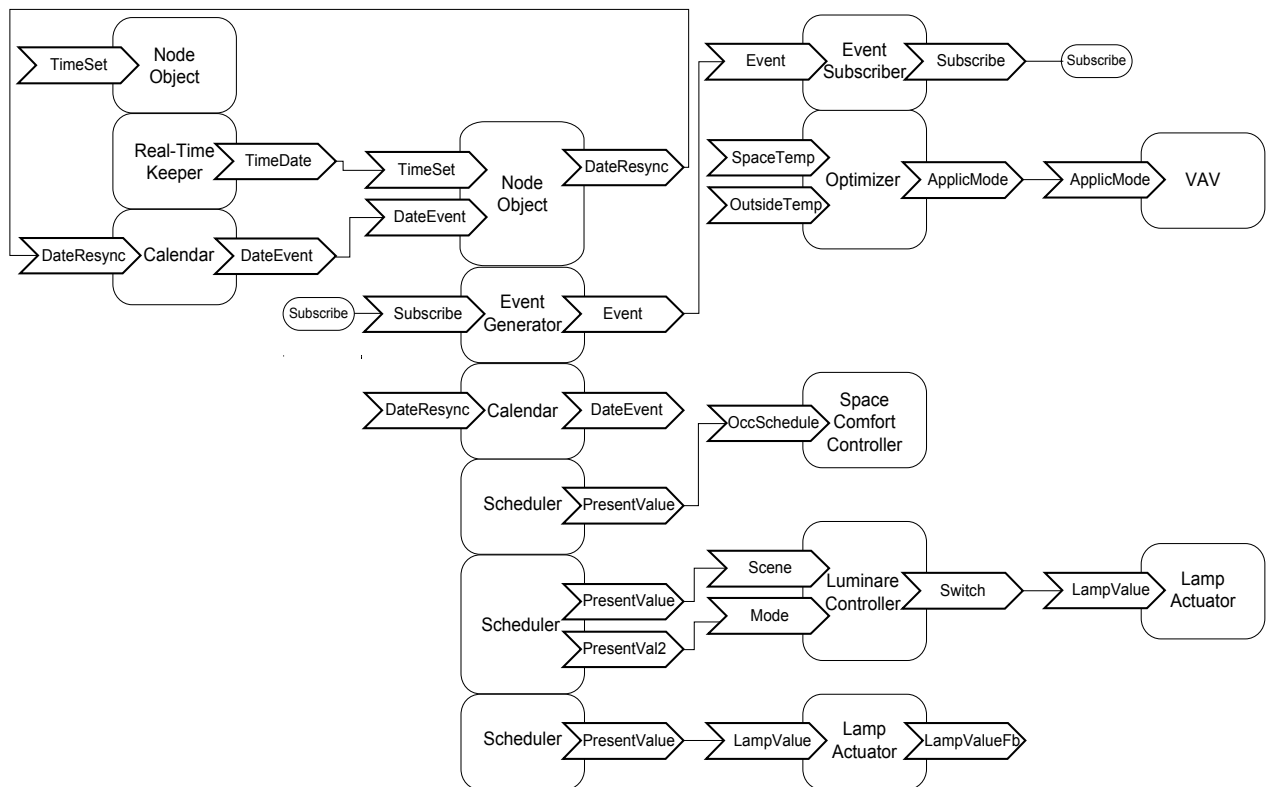
The Calendar functional block shall meet the following requirements:

1. An optional `nvoDateEvent` network variable output may be provided that reports the status of all schedules controlled by the Calendar on reset, time change, at midnight, and upon request. Status for each schedule shall include the schedule name, the schedule's time to active, and the schedule's time to inactive. A Calendar functional block may also provide schedule status updates directly to any Scheduler functional blocks on the same device.
2. No more than 256 schedules may be reported by an `nvoDateEvent` output. None, any, or all of the schedules may be active at the same time. The maximum for a particular implementation may be less than 256.
3. The `nvoDateEvent` output shall be set to the invalid value if no schedule dates are defined.
4. If an `nvoDateEvent` output is provided, an `nviDateResynch` network variable input shall be provided so that any subscriber to date events may request an update to all schedules controlled by the Calendar. This can be used by a Scheduler functional block to get all date events upon reset. This input is not required for Scheduler functional blocks on the same device with the Calendar functional block.
5. `SCPTovrBehave` and `SCPTovrValue` configuration properties may be included to specify override behavior and the override value. If not included, the override behavior shall be to retain the last setting if `SCPTovrValue` is not included or is invalid, or to output the override value if `SCPTovrValue` is included and is valid.
6. Active schedules are determined by *schedule dates*, where each schedule date specifies a date, a range of dates, or a repeating interval.
7. A schedule date that specifies a single day may be identified as *temporary*, in which case it is deleted at the end of the day that it is active.

---

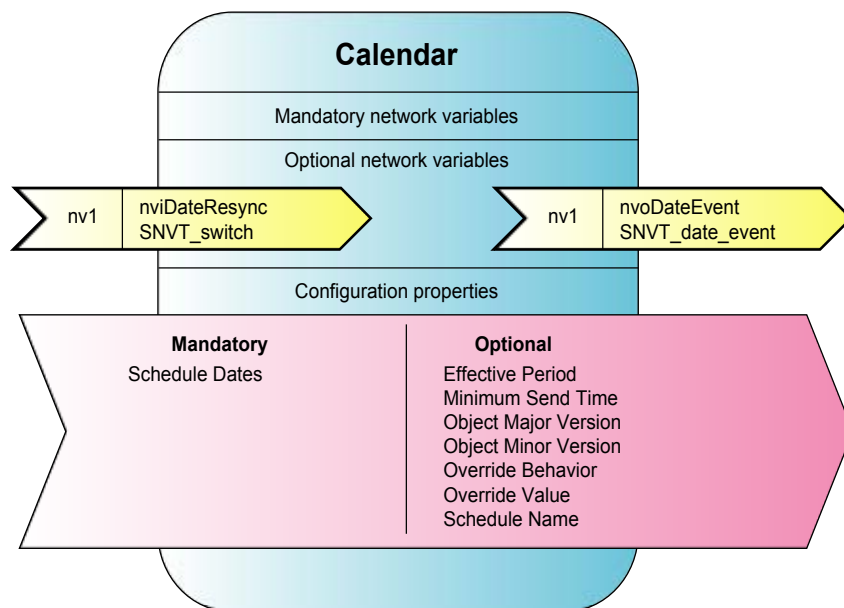
## Example Usage

A Calendar functional block may be used as a single object on a device, or in conjunction with other functional blocks including a node object, real-time keeper, and schedulers. A Calendar functional block may maintain its own real-time clock or it may receive time from a Node Object or Real-Time Keeper functional block.

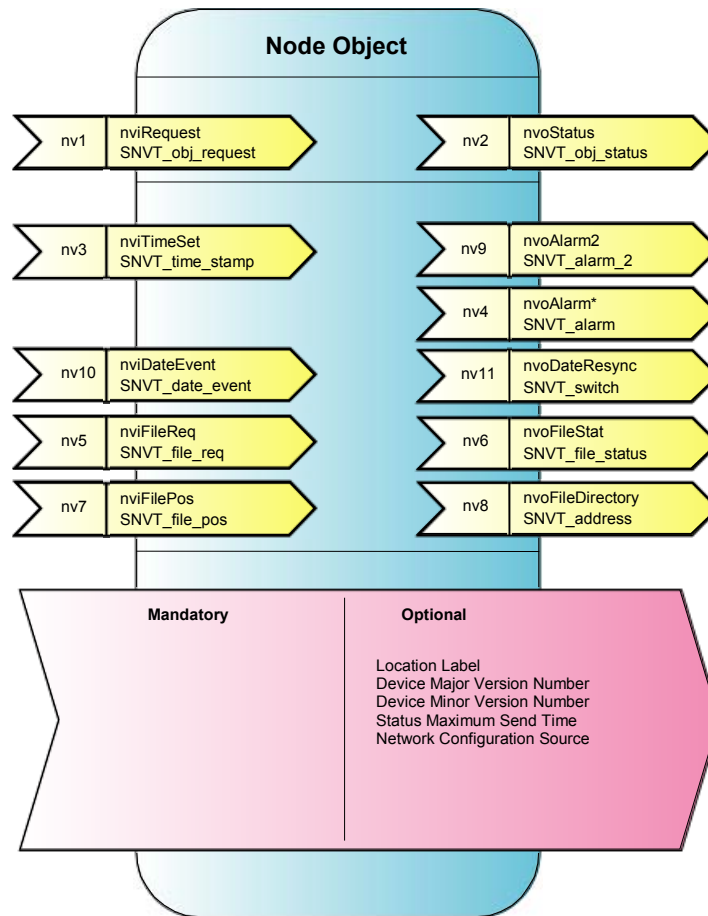


**Figure 2** Example Usage

## Object Details



**Figure 3** Profile Details



**Figure 4** Node Object Details

**Table 1** Node Object Network Variable Members

NV # (M/O)*	Variable Name	SNVT Name	SNVT Index	Description
1 (M)	nviRequest	SNVT_obj_request	1	A requested mode for a specified functional block in the device
2 (M)	nvoStatus	SNVT_obj_status	2	Reports the status of the requested functional block in the device
3 (M**)	nviTimeSet	SNVT_time_stamp	3	Synchronize the device's internal real time clock with an external time source

\* M = mandatory, O = optional, M\*\* = mandatory for this profile

**Table 2** Calendar Network Variable Members

<b>NV # (M/O)*</b>	<b>Variable Name</b>	<b>SNVT Name</b>	<b>SNVT Index</b>	<b>Description</b>
1 (O)	nvoDateEvent	SNVT_date_event	176	Reports the status of each schedule controlled by the Calendar
2 (O)	nviDateResync	SNVT_switch	95	Requests an update for all schedules controlled by the Calendar via the nvoDateEvent output

\* M = mandatory, O = optional

**Table 3** Calendar Configuration Property Members

<b>Man. Opt. *</b>	<b>SCPT Name NV Name Type or SNVT</b>	<b>SCPT Index</b>	<b>Associated NVs **</b>	<b>Description</b>
Opt	SCPTeffectivePeriod	272	Entire FB	Effective period for the calendar
Man	SCPTscheduleDates	273	Entire FB	Array of schedule dates, each associated with a schedule index and a temporary flag. Multiple schedule date entries may be associated with the same schedule.
Opt	SCPTminSendTime	52	nv1	Minimum period of time between nvoDateEvent output updates
Opt	SCPTobjMajVer	167	Entire FB	Functional block implementation major version number
Opt	SCPTobjMinVer	168	Entire FB	Functional block implementation minor version number
Opt	SCPTovrBehave	32	nv1	Output behavior when an override request is received
Opt	SCPTovrValue	33	nv1	Output value when an override request is received
Opt	SCPTscheduleName	279	Entire FB	Array of schedule names

\* Man = mandatory, Opt = optional

\*\* List of NVs to which this configuration property applies.

---

## Mandatory Network Variables

None.

---

## Optional Network Variables

---

### nvoDateEvent Output (Optional)

```
network output bind_info(ackd) SNVT_date_event nvoDateEvent;
```

This output network variable reports the status of each schedule referenced in the SCPTscheduleDates array.

The nvoDateEvent output shall be set to the invalid value if no schedules are defined.

#### *Valid Range*

The valid range specified by the SNVT definition.

#### *Default Value*

Invalid values for the days\_to\_active and days\_to\_inactive fields, null string for the name field.

### *Configuration Considerations*

The maximum update rate of this network variable is regulated by the time specified in the SCPTminSendTime configuration property unless the configuration property has a value of 0.0, or other invalid value; in which case, the network variable is not regulated by the SCPTminSendTime value.

### *When Transmitted*

Data is only transmitted on reset, time change, at midnight, and upon request. If multiple schedules are referenced by the schedule dates array, they are all transmitted in series, with a delay between each update as defined by the SCPTminSendTime configuration property.

## *Default Service Type*

The default service type is acknowledged.

---

## **nviDateResync Input (Optional)**

```
network input SNVT_switch nviDateResync;
```

This input network variable requests an update on the status of all schedules controlled by the Calendar functional block. When the state of this input changes from Off to On, the calendar sends a series of updates on the nvoDateEvent output, one for each schedule referenced in the SCPTscheduleDates array. The input is On when the state value is one (1) and the level value is greater than zero (0). This network variable is required if an nvoDateEvent output is provided.

## *Valid Range*

The state value may be zero (0) or one (1). The level value may be zero (0) to 200.

## *Default Value*

Off {0, 0}.

## *Configuration Considerations*

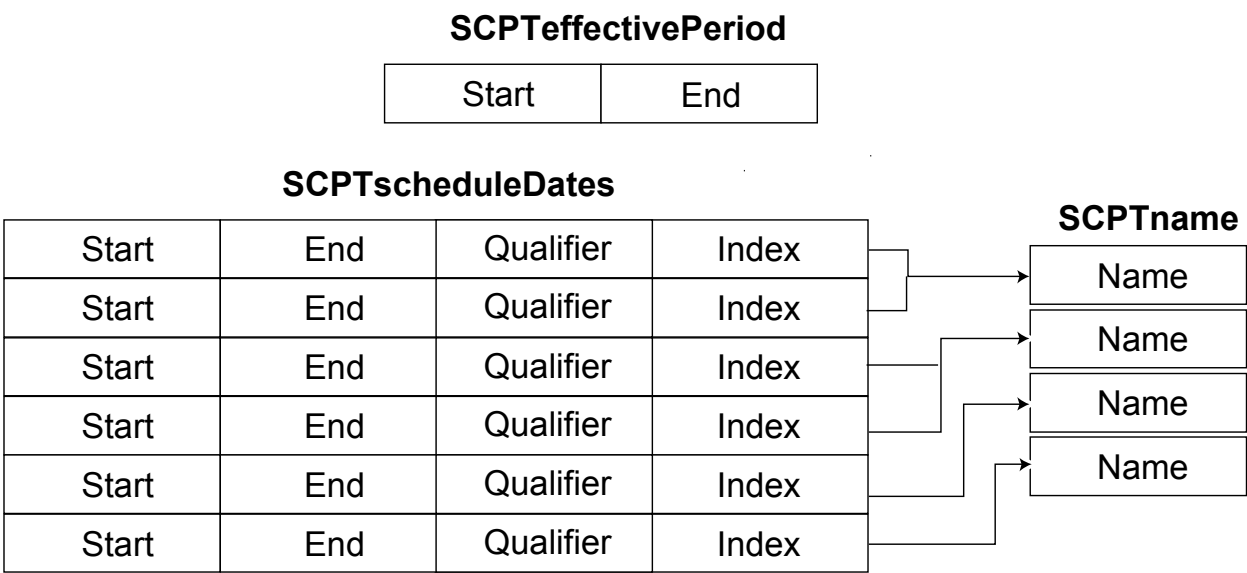
None.



---

# Configuration Properties

The Calendar SCPTs and SCPT arrays will be used to define calendars as shown in the following figure.



**Figure 4** Calendar SCPTs

---

## Effective Period (Optional)

This configuration property defines the time period during which a Calendar functional block is effective. During the effective period, the output of the scheduler will be defined by the schedules referenced in the SCPTscheduleDates array. Outside the effective period, the value of output will be the undefined value for SNVT\_date\_event. The effective period is defined by a start date and an end date. If the start date is undefined, it means any date up to and including the end date. If the end date is undefined, it means any date from the start date. If both are undefined, it means the Calendar functional block is always active.

### *Valid Range*

The valid range specified by the SCPT definition.

### *Default Value*

The default value specified by the SCPT definition.

### *Configuration Requirements/Restrictions*

This configuration property may only be modified when the functional block is disabled.

### *SCPT Reference*

SCPTeffectivePeriod (272)

---

## **Minimum Send Time (Optional)**

This configuration property sets the minimum period of time between updates to the nvoDateEvent output. This throttles the series of outputs that occur whenever schedule status is updated via nvoDateEvent.

### *Valid Range*

0.0 .. 6553.4 sec (0.1 sec). The value 0xFFFF represents invalid data.

### *Default Value*

The default value is 0.2, or 5 updates per second.

### *Configuration Requirements/Restrictions*

This configuration property may be modified at any time.

### *SCPT Reference*

SCPTminSendTime (52)

---

## **Object Major Version (Optional)**

This configuration property sets the major version number for a functional block implementation. It is used with the SCPTobjMinVer configuration property. See the *LONMARK Application-Layer Interoperability Guidelines* for details on usage.

### ***Valid Range***

The valid range specified by the SCPT definition.

### ***Default Value***

The default value is manufacturer-defined.

### ***Configuration Requirements/Restrictions***

This configuration property is Constant.

### ***SCPT Reference***

SCPTobjMajVer (167)

---

## **Object Minor Version (Optional)**

This configuration property sets the minor version number for a functional block implementation. It is used with the SCPTobjMajVer configuration property. See the *LONMARK Application-Layer Interoperability Guidelines* for details on usage.

### ***Valid Range***

The valid range specified by the SCPT definition.

### ***Default Value***

The default value is manufacturer-defined.

### ***Configuration Requirements/Restrictions***

This configuration property is Device Specific.

### ***SCPT Reference***

SCPTobjMinVer (168)

---

## Override Behavior (Optional)

This configuration property is used to define the behavior of the nvoDateEvent output when an override request is received for the calendar.

### *Valid Range*

The valid range for this configuration property is any value within the defined limits of the SNVT\_override network variable type. The calendar can retain its last setting, go to a specified value, or go to the default output value.

### *Default Value*

The default value is for the functional block output to retain its last setting (OV\_RETAIN).

### *Configuration Requirements/Restrictions*

This configuration property may be modified at any time.

### *SCPT Reference*

SCPTovrBehave (32)

---

## Override Value (Optional)

This configuration property is used to set the value the nvoDateEvent output should adopt when it is overridden and the value of Override Behavior is OV\_SPECIFIED.

### *Valid Range*

The valid range for this configuration property is any value within the defined limits of SNVT\_date\_event.

### *Default Value*

The default value is Invalid values for the days\_to\_active and days\_to\_inactive fields, nul string for the name field.

## *Configuration Requirements/Restrictions*

This configuration property may be modified at any time.

### *SCPT Reference*

SCPTovrValue (33)

---

## **Schedule Dates (Mandatory)**

This configuration property is used to create an array of schedule dates. Each entry consists of a starting date, ending date, date qualifier for recurring events, and a schedule index. The schedule index is used as an index into the optional SCPTscheduleName array, and may also be used as an index into SCPTschedule arrays for any Scheduler functional blocks on the same device.

### *Valid Range*

The valid range specified by the SCPT definition.

### *Default Value*

The default value specified by the SCPT definition.

## *Configuration Requirements/Restrictions*

This configuration property may only be modified when the functional block is disabled.

### *SCPT Reference*

SCPTscheduleDates (273)

---

## **Schedule Name (Optional)**

This configuration property is used to create an array of schedule names. An index into this array is provided by each entry in the SCPTscheduleDates array. The array will typically be shared by all Calendar and Scheduler functional blocks on a device. The Schedule Name array is required if the Calendar supports an nvoDateEvent output.

If a Calendar functional block supports temporary schedule dates, and a schedule name is only referenced by temporary schedules and temporary

schedule dates, then the schedule name shall be deleted when all the referencing schedules and schedule dates are deleted.

No more than 256 schedules may be reported by an nvoDateEvent output. None, any, or all of the schedules may be active at the same time. The maximum for a particular implementation may be less than 256.

### *Valid Range*

The valid range specified by the SCPT definition.

### *Default Value*

The default value specified by the SCPT definition.

### *Configuration Requirements/Restrictions*

This configuration property may only be modified when the functional block is disabled.

### *SCPT Reference*

SCPTscheduleName (TBD)

---

## **Data Transfer**

None specified.

---

## **Power-up State**

All schedules referenced in the SCPTscheduleDates array are updated via the nvoDateEvent output on reset.

---

## **Boundary and Error Conditions**

None specified.

---

## **Additional Considerations**

None specified.

---

Echelon, LON, Neuron, LONWORKS, LonTalk, LONMARK, and the LONMARK logo are trademarks of Echelon Corporation registered in the United States and other countries.