
**Intelligent transport systems —
Roadside modules SNMP data
interface —**

**Part 3:
Triggers**

*Systèmes de transport intelligents — Interface de données SNMP pour
les modules en bord de route —*

Partie 3: Déclencheurs



STANDARDSISO.COM : Click to view the full PDF of ISO/TS 20684-3:2022



COPYRIGHT PROTECTED DOCUMENT

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Conformance	2
5 User needs	4
5.1 Schedule triggers	4
5.1.1 Schedule triggers user need	4
5.1.2 Schedule triggers design overview	5
5.1.3 Schedule triggers graphical relationships	5
5.2 Schedule day plans	6
5.2.1 Schedule day plans user need	6
5.2.2 Schedule day plans design overview	6
5.2.3 Schedule day plans graphical relationships	6
5.3 Condition-based triggers	7
5.3.1 Condition-based triggers user need	7
5.3.2 Condition-based triggers design overview	8
5.3.3 Graphical relationships	8
6 Requirements	9
6.1 Action manager	9
6.1.1 Action manager definition	9
6.1.2 Action manager data exchange requirements	9
6.1.3 Action manager capability requirements	9
6.2 Conditional trigger	9
6.2.1 Conditional trigger definition	9
6.2.2 Conditional trigger data exchange requirements	10
6.2.3 Capability requirements	10
6.3 Day plan	13
6.3.1 Day plan definition	13
6.3.2 Day plan data exchange requirements	13
6.3.3 Day plan capabilities	13
6.4 Day plan scheduler	14
6.4.1 Day plan scheduler definition	14
6.4.2 Day plan scheduler data exchange requirements	14
6.4.3 Day plan scheduler capabilities	14
6.5 Trigger schedule	15
6.5.1 Trigger schedule definition	15
6.5.2 Trigger schedule data exchange requirements	15
6.5.3 Trigger schedule capabilities	15
7 Security vulnerabilities	15
Annex A (normative) Management information base (MIB)	17
Annex B (normative) Requirements traceability matrix (RTM)	51
Bibliography	58

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

A list of all parts in the ISO 20684 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

0.1 Background

The need for standardized communication with ITS field devices is growing around the world. Several countries have adopted Simple Network Management Protocol (SNMP) based field device communication standards.

There is a growing view and empirical evidence that standardizing this activity will result in improved ITS performance, reduced cost, reduced deployment time, and improved maintainability. The ISO 20684 series extends ISO 15784-2 by defining the management information necessary to monitor, configure and control features of field devices. The data elements defined in all parts of ISO 20684 series may be used with any protocol but were designed with an expectation that they would be used with one of the ISO 15784-2 protocols.

By using this approach, agencies can specify open procurements and systems can be expanded geographically in an open and non-proprietary manner, which reduces costs, speeds up deployment, and simplifies integration.

0.2 Overview

SNMP is a collection of well-thought-out and well-proven concepts and principles. SNMP employs the sound principles of abstraction and standardization. This has led to SNMP being widely accepted as the prime choice for communication between management systems and devices on the internet and other communications networks.

The original implementation of SNMP was used to manage network devices such as routers and switches. Since then, the use of SNMP has grown into many areas of application on the internet and has also been used successfully over various serial communications networks.

This document defines management information for ITS field devices following the SNMP conventions.

0.3 Document approach and layout

This document defines:

- a) the conformance requirements for this document ([Clause 4](#));
- b) a set of user needs for user-defined trigger conditions that can “fire” to initiate actions ([Clause 5](#));
- c) a set of detailed requirements for the identified user needs ([Clause 6](#));
- d) security considerations for the information defined in this document ([Clause 7](#));
- e) the management information bases that define the data for the defined requirements ([Annex A](#));
- f) the requirements traceability matrix (RTM) that traces the requirements to the design elements ([Annex B](#)).

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 20684-3:2022

Intelligent transport systems — Roadside modules SNMP data interface —

Part 3: Triggers

1 Scope

Field devices are a key component in intelligent transport systems (ITS). Field devices include traffic signals, message signs, weather stations, traffic sensors, roadside equipment for connected ITS (C-ITS) environments, etc.

Field devices often need to exchange information with other external entities (managers). Field devices can be quite complex, necessitating the standardization of many data concepts for exchange. As such, the ISO 20684 series is divided several individual parts.

This document specifies the needs, requirements and design for multiple mechanisms to fire triggers, which result in the device attempting to perform an action. Specific types of actions are defined in other documents and can include sending notifications (ISO/TS 20684-4), entering data into a log for later retrieval (ISO/TS 20684-5), and/or initiating SNMP-based requests (ISO/TS 20684-6).

NOTE 1 There are similarities between certain portions of NTCIP 1103 and NTCIP 1201 and this document.

NOTE 2 ISO 20684-1 provides additional details about how the ISO 20684 series relates to the overall ITS architecture.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 20684-1:2021, *Intelligent transport systems — Roadside modules SNMP data interface — Part 1: Overview*

ISO/TS 20684-7, *Intelligent transport systems – Roadside modules SNMP data interface – Part 7: Support features*

IETF RFC 2578, *Structure of Management Information Version 2 (SMIv2)*, April 1999.

IETF RFC 2579, *Textual Conventions for SMIv2*, April 1999.

IETF RFC 2580, *Conformance Statements for SMIv2*, April 1999.

IETF RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, December 2002.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 20684-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

hysteresis event

condition defined by alternating upper and lower limits

Note 1 to entry: The condition only evaluates to true when:

- a) the value rises above the upper limit and the previous true state was when the value was below the lower limit; or
- b) the value falls below the lower limit when the previous true state was above the upper limit.

Note 2 to entry: Hysteretic boundaries can be used to reduce the number of events that can potentially occur when data fluctuates over a small range near the boundary. Crossing the boundary in one direction generally reflects the onset of anomalous conditions while crossing in the other direction indicates a return to normal conditions.

EXAMPLE A user wants to be alerted when the speed along a motorway falls below 50 km/h. If the speed along the motorway was varying between 45 km/h and 55 km/h, the management station would receive an alert each time the average fell below 50 km/h during this variation. With a hysteretic boundary, the user can set a lower bound of 50 km/h and an upper bound of 60 km/h; in this case, the manager receives an alert the first time that the speed falls below 50 km/h, but does not receive another alert until the speed increases to 60 km/h.

4 Conformance

This clause follows the rules defined in ISO 20684-1. [Table 1](#) traces each user need to a set of software features. [Table 2](#) traces each feature to a set of requirements. [Table 3](#) defines terms that are used as predicates in the conformance codes listed in [Tables 1](#) and [2](#). For a full understanding of these tables and codes, see ISO 20684-1.

Table 1 — User need and feature conformance

Need	Requirement	Conformance
5.1: Schedule triggers		0,1 (1..*)
	6.1 : Action manager	M
	6.5 : Trigger schedule	M
	20684-7 6.1: Local clock	M
	20684-7 6.2: UTC clock	M
	20684-7 6.3: Daylight saving time	O
5.2: Schedule day plans		0,1 (1..*)
	6.1 : Action manager	M
	6.3 : Day plan	M
	6.4 : Day plan scheduler	M
	20684-7 6.1: Local clock	M
	20684-7 6.2: UTC clock	M
	20684-7 6.3: Daylight saving time	O
5.3: Condition-based triggers		0,1 (1..*)
	6.1 : Action manager	M
	6.2 : Conditional trigger	M

Table 2 — Requirement conformance

Feature	Requirement	Conformance
6.1: Action manager		
	6.1.2.1 : Determine action manager capabilities	M
	6.1.2.2 : Configure an action manager	M
	6.1.2.3 : Verify action manager configuration	M
	6.1.2.4 : Retrieve action manager statistics	M
	6.1.2.5 : Retrieve action manager enabled status	M
	6.1.2.6 : Toggle action manager	M
	6.1.2.7 : Delete action manager	M
6.2: Conditional trigger		
	6.2.2.1 : Discover triggering capabilities	M
	6.2.2.2 : Configure trigger	M
	6.2.2.3 : Confirm trigger configuration	M
	6.2.2.4 : Delete trigger definition	M
	6.2.2.5 : Retrieve statistics for trigger	M
	6.2.2.6 : Retrieve summary statistics for triggers	M
	6.2.2.7 : Toggle trigger enabled status	M
	6.2.2.8 : Monitor trigger status	M
	6.2.3.1.1 : Support for creation event	0,2(1..*)
	6.2.3.1.2 : Support for deletion event	0,2(1..*)
	6.2.3.1.3 : Support for change in value event	0,2(1..*)
	6.2.3.1.4 : Support for equal event	0,2(1..*)
	6.2.3.1.5 : Support for not equal event	0,2(1..*)
	6.2.3.1.6 : Support for greater than event	0,2(1..*)
	6.2.3.1.7 : Support for less than event	0,2(1..*)
	6.2.3.1.8 : Support for hysteresis event	0,2(1..*)
	6.2.3.1.9 : Support for periodic event	0,2(1..*)
	6.2.3.1.10 : Support for bitwise 'and' event on an INTEGER	0,2(1..*)
	6.2.3.1.11 : Support for bitwise 'and' event on an OCTET STRING	0,2(1..*)
	6.2.3.2.1 : Support for triggers based on current values	M
	6.2.3.2.2 : Support for triggers based on delta values	O
	6.2.3.3.1 : Support for creation wildcards	creation:O
	6.2.3.3.2 : Support for deletion wildcards	deletion:O
	6.2.3.3.3 : Support for on change wildcards	onChange:O
	6.2.3.4.1 : Support for local data with the default context	M
	6.2.3.4.2 : Support for local data with a specialized context	O
	6.2.3.4.3 : Support for remote data	O
	6.2.3.5 : Number of triggers	M
6.3: Day plan		
	6.3.2.1 : Configure a day plan	M
	6.3.2.2 : Verify day plan configuration	M
	6.3.2.3 : Toggle the enabled status of a day plan	M
	6.3.2.4 : Determine the enabled status of a day plan	M
	6.3.2.5 : Delete a day plan	M
	6.3.2.6 : Configure a day plan trigger	M

Table 2 (continued)

Feature	Requirement	Conformance
	6.3.2.7 : Verify day plan trigger configuration	M
	6.3.2.8 : Toggle the enabled status of a day plan trigger	M
	6.3.2.9 : Determine the enabled status of a day plan trigger	M
	6.3.2.10 : Delete a day plan trigger	M
6.4: Day plan scheduler		
	6.4.2.1 : Configure the day plan selection rules	M
	6.4.2.2 : Verify the day plan selection rule configuration	M
	6.4.2.3 : Disable a day plan schedule rule	M
	6.4.2.4 : Delete a day plan schedule rule	M
	6.4.2.5 : Determine day plan scheduler status	M
	6.4.2.6 : Determine day plan schedule statistics	M
	6.4.2.7 : Toggle the operation of a day plan scheduler	M
	6.4.2.8 : Monitor day plan scheduler errors	M
6.5: Trigger schedule		
	6.5.2.1 : Configure a scheduled trigger	M
	6.5.2.2 : Verify schedule for a trigger	M
	6.5.2.3 : Toggle the enabled status of a trigger	M
	6.5.2.4 : Determine enabled status of scheduled trigger	M
	6.5.2.5 : Determine performance of scheduled trigger	M
	6.5.2.6 : Delete a scheduled trigger	M
	6.5.3.1 : Support calendar triggers	0,4(1..*)
	6.5.3.2 : Support one-shot triggers	0,4(1..*)

Table 3 — External standard reference

Predicate	Subclause
creation	6.2.3.1.1
deletion	6.2.3.1.2
on-change	6.2.3.1.3

5 User needs

5.1 Schedule triggers

5.1.1 Schedule triggers user need

A manager needs to be able to schedule a field device to perform actions at one or more future known dates and times. The action to be performed can be to issue any command that the manager is authorized to issue via SNMP, log information, or send a manager a notification. Multiple independent managers can potentially wish to schedule these actions for various purposes with a level of confidence that they will not be inadvertently overwritten by other managers.

5.1.2 Schedule triggers design overview

5.1.2.1 Required features

In the simplest case, the “schedule actions” user need shall support the following features.

- Trigger schedule, as specified by this document, which defines rules for when a scheduled trigger should fire.
- Action manager, as specified by this document, which identifies the actions that are to be performed by the device when each trigger fires.
- Clock - local, as specified by ISO/TS 20684-7, which is based on the UTC clock with adjustments to reflect the local time zone. It is used by the trigger schedule to determine when it is time to fire the trigger.
- Clock - UTC, as specified by ISO/TS 20684-7, which is used to track the current UTC time.

5.1.2.2 Optional features

An implementation may support the DST feature, as defined in ISO/TS 20684-7, which defines the specific details about daylight saving time events that adjust the local clock forwards and backwards.

5.1.3 Schedule triggers graphical relationships

The relationships among these features are depicted in [Figure 1](#).

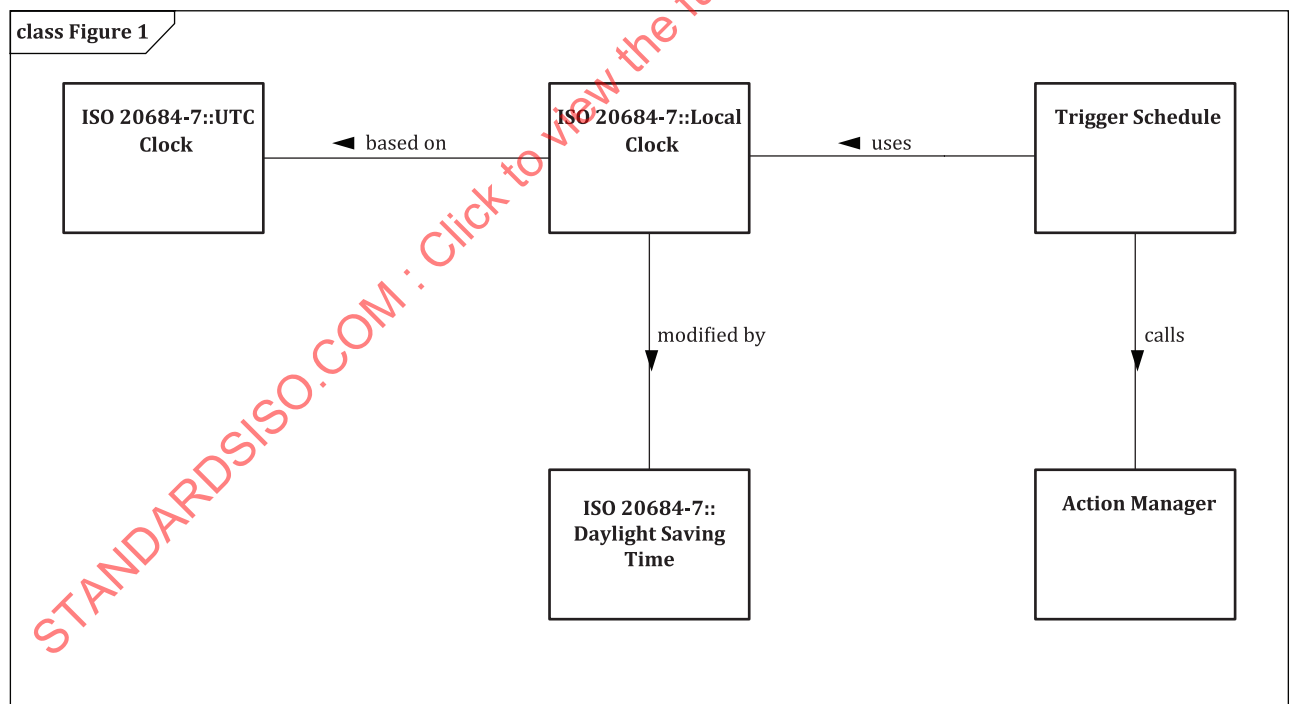


Figure 1 — Schedule triggers

Each trigger schedule defines time(s) at which a trigger will fire. The trigger schedule uses the ClockLocal to determine when to fire the trigger and thereby implement the action(s). The ClockLocal is based on the ClockUTC modified by the time zone and optionally by the daylight saving (i.e., summer) time (DST). When a trigger fires, it causes the ActionManager to perform one or more defined actions; the mechanisms used to define these actions are covered by other documents, such as ISO/TS 20684-4

(for issuing notifications), ISO/TS 20684-5 (for logging data), and ISO/TS 20684-6 (for issuing SNMP requests).

5.2 Schedule day plans

5.2.1 Schedule day plans user need

A manager needs to be able to activate a daily schedule of actions where the schedule is selected based on the current day-of-week and date and the actions within the daily schedule are based on local time-of-day. Being able to activate a daily schedule as a single unit can simplify the definition of the schedule of actions that tend to follow daily patterns. For example, with this mechanism a manager could define two day plans (i.e. two plans, each of which covers one 24-hour day): a 'normal' day plan and a 'holiday' day plan. The scheduling table would only need one entry to select the normal plan for every day of the year and a separate entry for each specific day when the holiday schedule is desired. According to the scheduler logic, the holiday day plan would be selected on the specified days because those entries would be more specific.

NOTE 1 Achieving the same level of configuration with the more generic "schedule triggers" user need would produce a much more convoluted configuration. However, if the scheduling logic does not require a day plan, the "schedule triggers" user need provides a very simple design.

NOTE 2 As only one day plan schedule can be active at any one time, it is recommended that only a single manager be granted access to the day plan schedule.

5.2.2 Schedule day plans design overview

5.2.2.1 Required features

In the simplest case, the "schedule day plans" user need shall support the following features.

- a) Day plan schedule, as specified by this document, which defines the rules for selecting a day plan to run; only one day plan can be active at any time.
- b) Day plan, as specified by this document, which provides a description of the day plan and a list of triggers to be fired at defined times during the day.
- c) Action manager, as specified by this document, which defines the actions that are to be performed by the device when each trigger fires.
- d) Clock - local, as specified within ISO/TS 20684-7, which is based on the UTC clock with adjustments to reflect the local time zone. It is used by the day plan schedule to select a day plan and is used by the day plan to identify when it is time to fire each trigger.
- e) Clock - UTC, as specified by ISO/TS 20684-7, which is used track the current UTC time.

5.2.2.2 Optional features

An implementation may support the DST feature, as defined in ISO/TS 20684-7, which defines the specific details about daylight saving time events that adjust the local clock forwards and backwards.

5.2.3 Schedule day plans graphical relationships

The relationships among these features are depicted in [Figure 2](#).

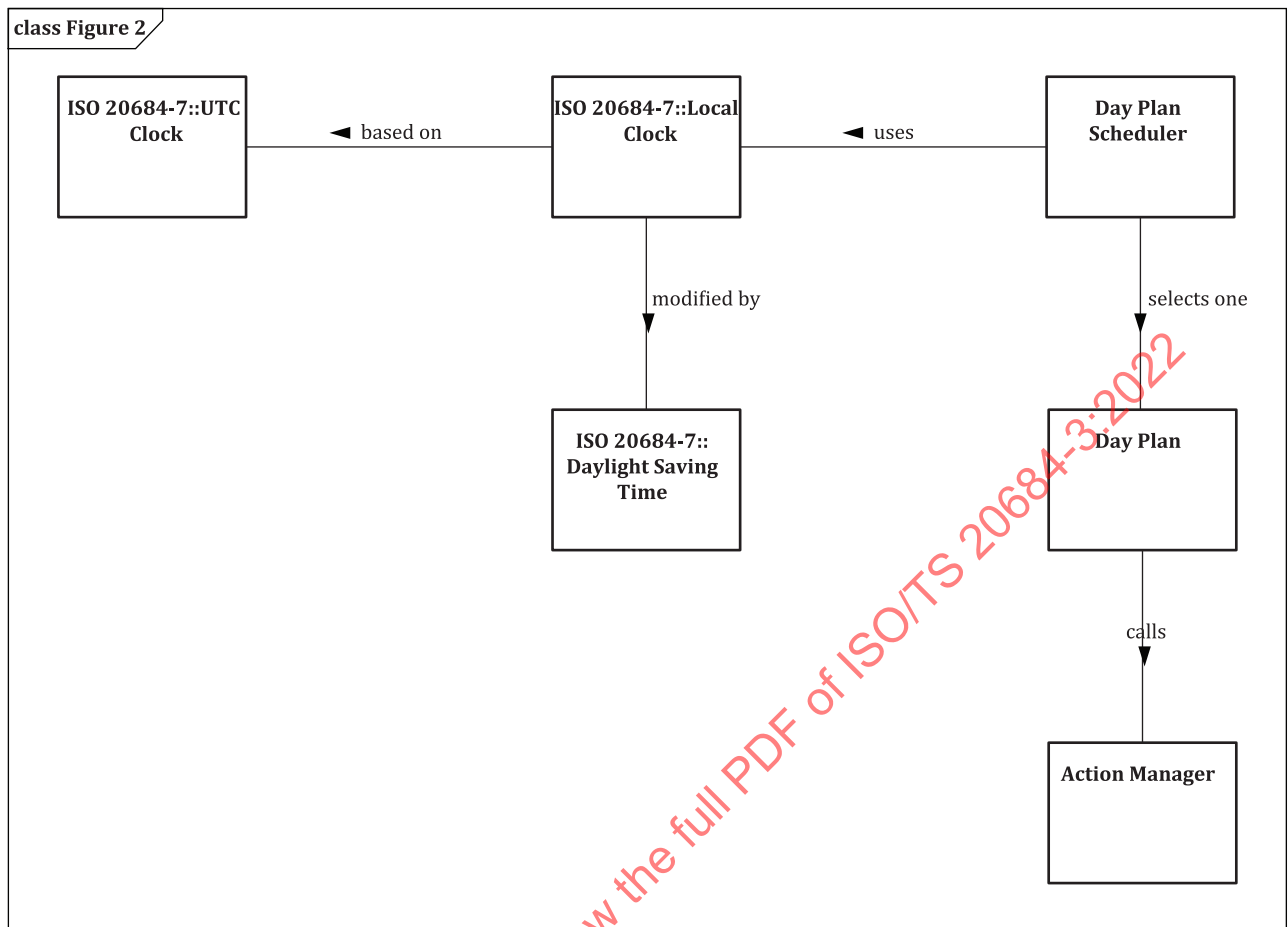


Figure 2 — Schedule day plans

The DayPlanSchedule selects a specific DayPlan based on the local month, day of month, and day of week. Each DayPlan consists of a description and series times at which a trigger will fire during the day. The DayPlanSchedule and DayPlanEvent both use the ClockLocal to determine the current local time. The ClockLocal is based on the ClockUTC modified by the time zone and optionally by the daylight saving time (DST). When a trigger fires, it causes the ActionManager to perform one or more defined actions; the mechanisms used to define these actions are covered by other standards, such as ISO/TS 20684-4 (for issuing notifications), ISO/TS 20684-5 (for logging data), and ISO/TS 20684-6 (for issuing SNMP requests).

5.3 Condition-based triggers

5.3.1 Condition-based triggers user need

One or more managers need to be able to configure a field device to fire a trigger when defined conditions occur. The trigger can cause the device to issue any command that the manager is authorized to issue via SNMP, log information, or send the manager a notification. Multiple independent managers can wish to schedule these triggers for various purposes with a level of confidence that they will not be inadvertently overwritten by other managers.

EXAMPLE 1 A manager wants the device to record the number of vehicles counted during every signal cycle.

EXAMPLE 2 A manager wants the device to issue a notification to a maintenance agency immediately when a cabinet door opens.

EXAMPLE 3 A manager wants to activate external equipment under certain conditions, such as when the temperature drops below freezing.

5.3.2 Condition-based triggers design overview

5.3.2.1 Required features

In the simplest case, the “condition-based triggers” user need shall support the following features:

- Conditional trigger, as specified by this document, which defines the conditions under which the trigger fires.
- Action manager, as specified by this document, which identifies the actions that are to be performed by the device when each trigger fires.

5.3.2.2 Optional feature group #1

An implementation may support the following features as a single group for this user need.

- SNMP target, as specified in ISO/TS 20684-7, which can be used to base conditions on data from remote devices rather than local data.
- SNMP target parameters, as specified in ISO/TS 20684-7, which defines parameters used to communicate with a remote SNMP entity.

5.3.3 Graphical relationships

The relationships among these features are depicted in [Figure 3](#).

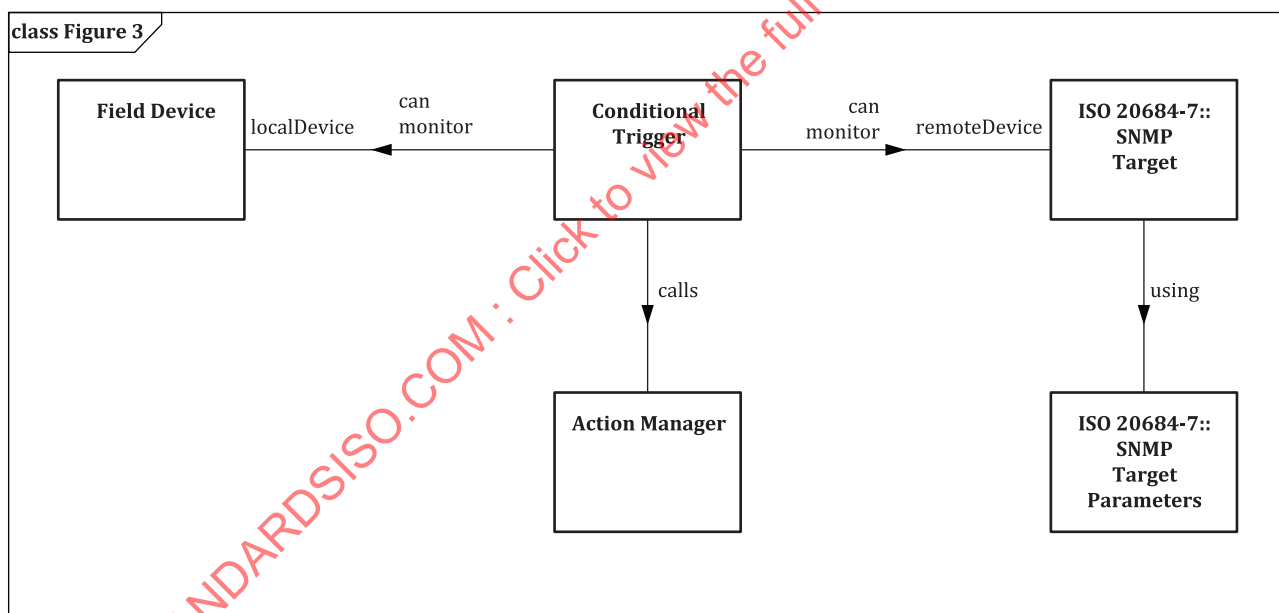


Figure 3 — Condition-based triggers

Each trigger is configured to monitor:

- local SNMP data (i.e. any data from the local field device); or
- SNMP data from a remote device (called an SNMP target).

When the conditions defined by the trigger occur, the trigger fires causing the Action Manager to perform one or more defined actions. The mechanisms used to define these actions are covered by other standards, such as ISO/TS 20684-4 (for issuing notifications), ISO/TS 20684-5 (for logging data), and ISO/TS 20684-6 (for issuing SNMP requests).

6 Requirements

6.1 Action manager

6.1.1 Action manager definition

The action manager associates the firing of a trigger with the actions that are to be performed, such as notifying a manager of information, as defined in ISO/TS 20684-4; logging information, as defined in ISO/TS 20684-5; and/or issuing commands, as defined in ISO/TS 20684-6.

NOTE In theory, the action manager could be activated by a mechanism other than the schedule triggers, schedule day plans, or condition-based triggers mechanisms defined in this document.

6.1.2 Action manager data exchange requirements

6.1.2.1 Determine action manager capabilities

The field device shall allow a manager to determine the types of actions that are supported.

6.1.2.2 Configure an action manager

The field device shall allow a manager to configure an action manager by defining its name, description and actions to be performed.

6.1.2.3 Verify action manager configuration

The field device shall allow a manager to verify the configuration of an action manager.

6.1.2.4 Retrieve action manager statistics

The field device shall allow a manager to retrieve the number of action attempts and failures that have occurred in activating each action.

6.1.2.5 Retrieve action manager enabled status

The field device shall allow a manager to retrieve the current enabled status of each action manager.

6.1.2.6 Toggle action manager

The field device shall allow a manager to toggle the enabled status of each action manager. When "off" the action manager will not implement any actions. When "on" the action manager will implement all actions associated with the entry.

6.1.2.7 Delete action manager

The field device shall allow a manager to delete an action manager.

6.1.3 Action manager capability requirements

No capability requirements are defined for the event manager.

6.2 Conditional trigger

6.2.1 Conditional trigger definition

A conditional trigger is a feature that monitors a pre-defined condition and "fires" when the condition transitions to a true state. The parameters that define the trigger also define when the trigger resets

and whether the monitored condition changes upon reset. For example, a trigger may be defined to fire when a parameter exceeds a defined value. Depending on the configuration, the trigger may not reset until the parameter drops below the value (and then it monitors the same condition) or the trigger may immediately reset and begin monitoring when the value drops below a defined value.

A trigger can be based on:

- a) any visible data from the local field device;
- b) optionally, any visible data from a remote field device target; and
- c) optionally, an expression that may use data from the local field device and/or one or more remote field devices.

6.2.2 Conditional trigger data exchange requirements

6.2.2.1 Discover triggering capabilities

The field device shall allow a manager to discover the capabilities of the conditional trigger feature.

6.2.2.2 Configure trigger

The field device shall allow a manager to configure a trigger.

6.2.2.3 Confirm trigger configuration

The field device shall allow a manager to confirm the current configuration of a trigger.

6.2.2.4 Delete trigger definition

The field device shall allow a manager to delete a trigger.

6.2.2.5 Retrieve statistics for trigger

The field device shall allow a manager to retrieve statistics regarding the firing of a trigger.

6.2.2.6 Retrieve summary statistics for triggers

The field device shall allow a manager to retrieve statistics regarding all defined triggers.

6.2.2.7 Toggle trigger enabled status

The field device shall allow a manager to toggle whether or not the trigger is currently enabled.

6.2.2.8 Monitor trigger status

The field device shall allow a manager to monitor the current status of the trigger.

6.2.3 Capability requirements

6.2.3.1 Conditional trigger types

6.2.3.1.1 Support for creation event

The field device shall support triggers that fire when the specified object [or instance of the specified object type(s), if a wildcard is used] is created.

6.2.3.1.2 Support for deletion event

The field device shall support triggers that fire when the specified object [or instance of the specified object type(s), if a wildcard is used] is deleted.

6.2.3.1.3 Support for change in value event

The field device shall support triggers that fire when the monitored value (or values, if a wildcard is used) changes.

6.2.3.1.4 Support for equal event

The field device shall support triggers that fire when the monitored value equals a specified value.

6.2.3.1.5 Support for not equal event

The field device shall support triggers that fire when the monitored value does not equal a specified value.

6.2.3.1.6 Support for greater than event

The field device shall support triggers that fire when the monitored value first exceeds a specified value.

6.2.3.1.7 Support for less than event

The field device shall support triggers that fire when the monitored value first falls below a specified value.

6.2.3.1.8 Support for hysteresis event

The field device shall support triggers that fire based on a hysteresis condition. In other words, rising trigger will fire when the monitored value first exceeds an upper threshold and this event will also reset the falling trigger; likewise, a falling trigger will fire when the monitored value falls below a lower threshold and this event will also reset the rising trigger.

6.2.3.1.9 Support for periodic event

The field device shall support triggers that fire periodically.

6.2.3.1.10 Support for bitwise 'and' event on an INTEGER

The field device shall support triggers that fire when the result of performing a bitwise 'AND' operation with the monitored value and a specified value produces an integer that contains at least one set bit.

6.2.3.1.11 Support for bitwise 'and' event on an OCTET STRING

The field device shall support triggers that fire when the result of performing a bitwise 'AND' operation with the monitored value and a specified value produces a bit string that contains at least one set bit.

6.2.3.2 Trigger sample types**6.2.3.2.1 Support for triggers based on current values**

The field device shall support triggers based on the current value of a specified object.

6.2.3.2.2 Support for triggers based on delta values

The field device shall support triggers based on comparisons performed on the relative change in the value of a specified object since its last reading. In other words, the field device will determine the difference between two readings and compare this delta value in the operation (i.e. greater than, less than; equal, not equal; etc.) specified in the trigger.

6.2.3.3 Wildcards

6.2.3.3.1 Support for creation wildcards

The field device shall support the use of wildcards when specifying creation triggers. In other words, a single conditional trigger can be defined to monitor the creation of any object that starts with a particular OID string.

6.2.3.3.2 Support for deletion wildcards

The field device shall support the use of wildcards when specifying creation triggers. In other words, a single conditional trigger can be defined to monitor the deletion of any object that starts with a particular OID string.

6.2.3.3.3 Support for on change wildcards

The field device shall support the use of wildcards when specifying on change triggers. In other words, a single conditional trigger can be defined to monitor changes to any object that starts with a particular OID string.

6.2.3.4 Data source

6.2.3.4.1 Support for local data with the default context

The field device shall support configuring triggers based on data from the local field device using the default context. In other words, this is the data that represents the current status and configuration of the local field device.

6.2.3.4.2 Support for local data with a specialized context

The field device shall support configuring triggers based on data from the local field device using a special context. For example, the local field device can potentially serve as a proxy engine for multiple other devices. While the data can potentially be stored within the local device, the data represents the configuration and status of another device or subcomponent of this device.

6.2.3.4.3 Support for remote data

The field device shall support configuring triggers based on data from a remote device. In other words, the field device shall be able to be configured to monitor values that it obtains by sending SNMP requests to other field devices (i.e. presumably field devices that do not offer their own support for triggers).

6.2.3.5 Number of triggers

In the absence of any other specification, the field device shall support at least one trigger.

6.3 Day plan

6.3.1 Day plan definition

A day plan consists of a sequence of triggers that are to be fired at specific times during the local calendar day. Only one day plan may be active at any time. The active day plan is determined by the day plan schedule.

6.3.2 Day plan data exchange requirements

6.3.2.1 Configure a day plan

The field device shall allow a manager to configure information that applies to all actions performed within a day plan (e.g. by defining a description and specifying the type of memory to use).

6.3.2.2 Verify day plan configuration

The field device shall allow a manager to determine the configuration of a day plan.

6.3.2.3 Toggle the enabled status of a day plan

The field device shall allow a manager to enable/disable an entire day plan.

6.3.2.4 Determine the enabled status of a day plan

The field device shall allow a manager to determine if a selected day plan is currently allowed to be active.

6.3.2.5 Delete a day plan

The field device shall allow a manager to delete an entire day plan along with all of its scheduled triggers.

6.3.2.6 Configure a day plan trigger

The field device shall allow a manager to configure a trigger to call an action to be performed as a part of a day plan.

6.3.2.7 Verify day plan trigger configuration

The field device shall allow a manager to determine the configuration of a day plan trigger.

6.3.2.8 Toggle the enabled status of a day plan trigger

The field device shall allow a manager to enable/disable a specific trigger within a day plan.

6.3.2.9 Determine the enabled status of a day plan trigger

The field device shall allow a manager to determine if a selected trigger within a day plan is enabled.

6.3.2.10 Delete a day plan trigger

The field device shall allow a manager to delete a scheduled trigger within a day plan.

6.3.3 Day plan capabilities

No capability requirements are defined for the day plan.

6.4 Day plan scheduler

6.4.1 Day plan scheduler definition

The day plan scheduler allows a field device to run a specified day plan, which consists of a series of triggers (e.g. commands) performed at pre-defined times of day. These actions can be enabled even if communications to the manager are not available when the actions are supposed to occur. Only one day plan can be active at any one time; if multiple managers are authorized to write to the day plan schedule, extra care should be taken to coordinate the actions taken.

6.4.2 Day plan scheduler data exchange requirements

6.4.2.1 Configure the day plan selection rules

The field device shall allow a manager to configure the rules for selecting a specific day plan and recording a description for each rule written.

6.4.2.2 Verify the day plan selection rule configuration

The field device shall allow a manager to verify the configuration of the day plan schedule.

6.4.2.3 Disable a day plan schedule rule

The field device shall allow a manager to disable the operation of a configured day plan schedule rule.

6.4.2.4 Delete a day plan schedule rule

The field device shall allow a manager to delete a day plan schedule rule.

6.4.2.5 Determine day plan scheduler status

The field device shall allow a manager to determine the current status of the day plan schedule, including which rule is currently selected based on the local clock, which day plan is selected and whether the selected day plan schedule is currently enabled (i.e. able to issue commands).

6.4.2.6 Determine day plan schedule statistics

The field device shall allow a manager to determine statistics about the operation of the scheduler, such as the number of times the scheduler has fired a trigger.

6.4.2.7 Toggle the operation of a day plan scheduler

The field device shall allow a manager to enable/disable the operation of the day plan schedule. When disabled, the day plan schedule shall continue to select the applicable rule within the schedule, but it shall not issue any of the resultant day plan schedule commands.

6.4.2.8 Monitor day plan scheduler errors

The field device shall allow a manager to determine details about the last error received related to a day plan command.

6.4.3 Day plan scheduler capabilities

No capability requirements are defined for the day plan scheduler.

6.5 Trigger schedule

6.5.1 Trigger schedule definition

The trigger schedule causes the field device to fire specific triggers at defined times. Each scheduled trigger consists of a specification of when to fire the trigger and a reference to the action(s) to be performed. The trigger is fired even when communications to the manager are not available. Each defined trigger schedule acts independently.

6.5.2 Trigger schedule data exchange requirements

6.5.2.1 Configure a scheduled trigger

The field device shall allow a manager to configure the schedule for firing triggers and recording a description for each entry.

6.5.2.2 Verify schedule for a trigger

The field device shall allow a manager to determine the schedule defined for scheduled triggers.

6.5.2.3 Toggle the enabled status of a trigger

The field device shall allow a manager to toggle the enabled status of a scheduled trigger.

6.5.2.4 Determine enabled status of scheduled trigger

The field device shall allow a manager to determine the current enabled status of a scheduled trigger.

6.5.2.5 Determine performance of scheduled trigger

The field device shall allow a manager to determine how many times the trigger has fired and information about any errors that have occurred.

6.5.2.6 Delete a scheduled trigger

The field device shall allow a manager to delete a scheduled trigger.

6.5.3 Trigger schedule capabilities

6.5.3.1 Support calendar triggers

The command scheduler shall support calendar-based schedule commands.

6.5.3.2 Support one-shot triggers

The command scheduler shall support one-shot schedule commands.

7 Security vulnerabilities

There are data elements defined in this document with a MAX-ACCESS clause of read-write and/or read-create. These and other data elements are sensitive and need to be protected from malicious and inadvertent manipulation and/or disclosure. The support for requests in a non-secure environment without proper protection can have a negative effect on network operations. A sampling of the vulnerabilities includes:

- a) the ability to change when triggers are fired;

- b) the ability to delete triggers and thereby disable alerts;
- c) the ability to create additional triggers that may flood the network or processor; and
- d) the ability to monitor current configurations.

To overcome these vulnerabilities, it is highly recommended that SNMPv3 with transport layer security (TLS) support, as defined in RFC 6353 or later, is used to exchange the data.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 20684-3:2022

Annex A (normative)

Management information base (MIB)

This annex provides definitions which it is useful to import into other management information base (MIB) modules of this document.

A.1 Action MIB

```
-- *****
-- A.1.1 Action MIB Header
-- *****

ACTION-MIB DEFINITIONS ::= BEGIN
IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY, Integer32, Counter32, Unsigned32
    FROM SNMPv2-SMI
    -- RFC 2578

StorageType, RowStatus
    FROM SNMPv2-TC
    -- RFC 2579

MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF
    -- RFC 2580

SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
    -- RFC 3411

fieldDevice, iso20684p3
    FROM FIELD-DEVICE-TC-MIB
    -- ISO 20684-1 Annex A

;
fdActionMIB MODULE-IDENTITY
    LAST-UPDATED "201912292026Z"
    ORGANIZATION "ISO TC 204 WG 9"
    CONTACT-INFO
        "name: Kenneth Vaughn
        phone: +1-571-331-5670
        email: kvaughn@trevilon.com
        postal: 6606 FM 1488 RD STE 148-503
        Magnolia, TX 77354
        USA"
    DESCRIPTION
        "This MIB defines actions for a device to perform when a trigger fires. A
        trigger can fire due to a schedule, a day plan schedule, or another
        defined condition, per the rules of ISO/TS 20684-3 or may activate due to
        triggers defined in other standards."

    REVISION "201912292026Z"
    DESCRIPTION
        "Initial revision of the document as proposed for CD ballot"

    ::= {iso20684p3 1}

-- *****
-- A.1.2 Node Definitions
```

```

-- *****

fdActionConformance OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node containing conformance statements related to the fdActionMIB."
    ::= {fdActionMIB 2}

fdActionCompliances OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node for compliance statements for the fdActionMIB."
    ::= {fdActionConformance 1}

fdActionGroups OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node for group definitions related to fdActionMIB."
    ::= {fdActionConformance 2}

fdAction OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node defining management information related to the field device's
        Action table."
    ::= {fieldDevice 4}

-- *****
-- A.1.3 Action objects
-- *****

fdActionsSupportedTypes OBJECT-TYPE
    SYNTAX      BITS { command (0), log (1), notification (2) }

    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "An indication of the action types that are allowed. The bits are:
        Bit | Description
          0 | The COMMAND-MIB is supported
          1 | The LOG-MIB is supported
          2 | The NOTIFICATION-MIB is supported"
    ::= {fdAction 1}

fdActionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF FdActionEntry

    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that manages the actions that can be internally called by the
        device."
    ::= {fdAction 2}

fdActionEntry OBJECT-TYPE
    SYNTAX      FdActionEntry

    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An action that can be performed by the device. The action table provides a
        way for a single trigger to call multiple specific actions."

    INDEX       {fdActionOwner, fdActionName, fdActionIndex}
    ::= {fdActionTable 1}

FdActionEntry ::= SEQUENCE {
    fdActionOwner      SnmpAdminString,
    fdActionName       SnmpAdminString,
    fdActionIndex      Unsigned32,
    fdActionDescription SnmpAdminString,

```

```

fdActionType          INTEGER,
fdActionTypeOwner     SnmpAdminString,
fdActionTypeName      SnmpAdminString,
fdActionTypeNumber    Integer32,
fdActionTriggerCount  Counter32,
fdActionFailureCount  Counter32,
fdActionDisabledCount Counter32,
fdActionStorageType   StorageType,
fdActionRowStatus     RowStatus }

fdActionOwner OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..32))

    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "The owner of this entry. The exact semantics of this string are subject to
        the security policy defined by the security administrator."
    ::= {fdActionEntry 1}

fdActionName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..32))

    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "The unique name of the action within the scope of the fdActionOwner."
    ::= {fdActionEntry 2}

fdActionIndex OBJECT-TYPE
    SYNTAX      Unsigned32

    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "The index of this component within the scope of the fdActionOwner and
        fdActionName. When a trigger fires, it will call a specific fdActionOwner
        and fdActionName; this third index allows multiple actions to be
        associated with a single trigger."
    ::= {fdActionEntry 3}

fdActionDescription OBJECT-TYPE
    SYNTAX      SnmpAdminString

    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "The human readable description of the purpose of this entry."
    ::= {fdActionEntry 4}

fdActionType OBJECT-TYPE
    SYNTAX      INTEGER { other (1),
                          command (2),
                          log (3),
                          notification (4),
                          ascAction (5),
                          dmsAction (6) }

    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "The type of action to be performed by this entry. Valid values are:
        Value | Description
        other | Action not defined in this standard
        command | This entry shall activate the entry in the fdCommandTable with
                  | an fdCommandOwner equal to fdActionTypeOwner and an
                  | fdCommandName equal to fdActionTypeName
        log | This entry shall activate the entry in the fdLogTable with
              | an fdLogOwner equal to fdActionTypeOwner and an
              | fdLogName equal to fdActionTypeName
        notification | This entry shall activate the entry in the

```

```

        | fdNotificationTable with an fdNotificationOwner equal to
        | fdActionTypeOwner and an fdNotificationName equal to
        | fdActionTypeName.
ascAction | This entry shall activate the entry in the timebaseAscActionTable
        | with a timebaseAscActionNumber equal to fdActionTypeNumber
dmsAction | This entry shall activate the entry in the dmsActionTable with
        | a dmsActionIndex equal to fdActionTypeNumber
"
::= {fdActionEntry 5}

fdActionTypeOwner OBJECT-TYPE
SYNTAX      SnmpAdminString (0..32)

MAX-ACCESS read-create
STATUS      current
DESCRIPTION
    "The owner of the action to be performed. This shall reference the
    corresponding fdCommandOwner, fdLogOwner, or fdNotificationOwner as
    indicated by fdActionType."
::= {fdActionEntry 6}

fdActionTypeName OBJECT-TYPE
SYNTAX      SnmpAdminString (0..32)

MAX-ACCESS read-create
STATUS      current
DESCRIPTION
    "The name of the action to be performed. This shall reference the
    corresponding fdCommandName, fdLogName, or fdNotificationName as indicated
    by fdActionType."
::= {fdActionEntry 7}

fdActionTypeNumber OBJECT-TYPE
SYNTAX      Integer32

MAX-ACCESS read-create
STATUS      current
DESCRIPTION
    "The number of the action to be performed. This shall reference the
    corresponding timebaseAscActionTable or dmsActionTable entry."
::= {fdActionEntry 8}

fdActionTriggerCount OBJECT-TYPE
SYNTAX      Counter32

MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The number of times this action has been called by a triggering event when
    the entry was in the active and enabled state. Triggering events include
    those from the fdTriggerSchedTable, fdDayPlanSchedTable, and
    fdCondTriggerTable."
::= {fdActionEntry 9}

fdActionFailureCount OBJECT-TYPE
SYNTAX      Counter32

MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "The number of times this action has been called by a triggering event when
    the entry was in the active and enabled state and the action referenced by
    this entry was found not to exist in an active and enabled state."
::= {fdActionEntry 10}

fdActionDisabledCount OBJECT-TYPE
SYNTAX      Counter32

MAX-ACCESS read-only
STATUS      current
DESCRIPTION

```

```

    "The number of times this action has been called by a triggering event when
    the entry was either not in the active state (fdActionRowStatus) or was
    not enabled (fdActionEnabled)."
```

::= {fdActionEntry 11}

fdActionStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

 "This object defines whether this action is kept in volatile storage and
 lost upon reboot or if this row is backed up by non-volatile or permanent
 storage. Conceptual rows having the value 'permanent' must allow write
 access to the fdActionEnabled columnar object."

::= {fdActionEntry 12}

fdActionRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

 "The status of this action. A control that allows entries to be added and
 removed from this table.

 fdActionDescription and fdActionEnabled can be modified independent of the
 current row status. All other read-create objects in this entry can only
 be modified when the value of this object is not 'active'."

::= {fdActionEntry 13}

-- *****

-- A.1.4 Conformance Information

-- *****

fdActionMIBCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

 "The conformance statement for actions for a device to perform when a
 trigger fires."

MODULE -- this module

MANDATORY-GROUPS {

 fdActionCapabilitiesGroup,

 fdActionManagementGroup,

 fdActionConfigurationGroup,

 fdActionStatusGroup

};

::= {fdActionCompliances 1}

fdActionCapabilitiesGroup OBJECT-GROUP

OBJECTS {

 fdActionsSupportedTypes

};

STATUS current

DESCRIPTION

 "Management information that identifies the capabilities of the action
 group."

REFERENCE "Clause 6.1.2.1"

::= {fdActionGroups 1}

fdActionManagementGroup OBJECT-GROUP

OBJECTS {

 fdActionRowStatus

};

STATUS current

DESCRIPTION

 "Management information that provides for creation, deletion, and
 management of an fdActionEntry."

```

REFERENCE "Clause 6.1.2.2, Clause 6.1.2.3, Clause 6.1.2.5, Clause 6.1.2.6,
Clause 6.1.2.7"
::= {fdActionGroups 2}

fdActionConfigurationGroup OBJECT-GROUP
OBJECTS {
    fdActionDescription,
    fdActionType,
    fdActionTypeOwner,
    fdActionTypeName,
    fdActionTypeNumber,
    fdActionStorageType
}
STATUS current
DESCRIPTION
    "Management information used for configuration of actions."
REFERENCE "Clause 6.1.2.2, Clause 6.1.2.3"
::= {fdActionGroups 3}

fdActionStatusGroup OBJECT-GROUP
OBJECTS {
    fdActionTriggerCount,
    fdActionFailureCount,
    fdActionDisabledCount
}
STATUS current
DESCRIPTION
    "Management information that identifies the status of the action."
REFERENCE "Clause 6.1.2.4"
::= {fdActionGroups 4}

END
-- ASN1END

```

A.2 Conditional Trigger MIB

```

-- *****
-- A.2.1 Conditional Trigger Header
-- *****
COND-TRIGGER-MIB DEFINITIONS ::= BEGIN
IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY, Counter32, Integer32, Unsigned32
    FROM SNMPv2-SMI
    -- RFC 2578

TruthValue, StorageType, RowStatus
    FROM SNMPv2-TC
    -- RFC 2579

MODULE-COMPLIANCE OBJECT-GROUP
    FROM SNMPv2-CONF
    -- RFC 2580

SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
    -- RFC 3411

fieldDevice, iso20684p3
    FROM FIELD-DEVICE-TC-MIB
    -- ISO 20684-1 Annex A

fdActionOwner
    FROM ACTION-MIB
    -- ISO/TS 20684-3
    -- Annex A.1

;
fdCondTriggerMIB MODULE-IDENTITY
    LAST-UPDATED "201912302026Z"

```

ORGANIZATION "ISO TC 204 WG 9"

CONTACT-INFO

"name: Kenneth Vaughn
 phone: +1-571-331-5670
 email: kvaughn@trevilon.com
 postal: 6606 FM 1488 RD STE 148-503
 Magnolia, TX 77354
 USA"

DESCRIPTION

"This MIB defines a mechanism by which a manager can configure actions to be performed based on user-defined comparisons using data within the field device."

REVISION "201912302026Z"

DESCRIPTION

"Initial revision of the document as proposed for CD ballot"

::= {iso20684p3 2}

-- *****
 -- A.2.2 Node Definitions
 -- *****

fdCondTriggerConformance OBJECT-IDENTITY

STATUS current

DESCRIPTION

"A node containing conformance statements related to the fdCondTriggerMIB, as defined in ISO/TS 20684-3."

::= {fdCondTriggerMIB 2}

fdCondTriggerCompliances OBJECT-IDENTITY

STATUS current

DESCRIPTION

"A node for compliance statements for the fdCondTriggerMIB."

::= {fdCondTriggerConformance 1}

fdCondTriggerGroups OBJECT-IDENTITY

STATUS current

DESCRIPTION

"A node for group definitions related to fdCondTriggerMIB."

::= {fdCondTriggerConformance 2}

fdCondTrigger OBJECT-IDENTITY

STATUS current

DESCRIPTION

"A node defining management information related to the field device's Trigger Schedule."

::= {fieldDevice 5}

-- *****
 -- A.2.3 Trigger Schedule
 -- *****

fdCondTriggersSupport OBJECT-TYPE

SYNTAX BITS {
 current (0),
 delta (1),
 onChange (2),
 greaterThan (3),
 lessThan (4),
 hysteresis (5),
 periodic (6),
 andedWithValue (7),
 equal (8),
 notEqual (9),
 creation (10),
 deletion (11),
 integerBitwiseAnd (12),
 octetBitwiseAnd (13) }

MAX-ACCESS read-only

```

STATUS      current
DESCRIPTION
  "The options supported by the conditional triggering system within the
  field device: Valid bits are:
    Value      | Description
    current    | The field device supports 'current' sample types
    delta      | The field device supports 'delta' sample types
    onChange   | The field device supports 'onChange' triggers (mode)
    greaterThan | The field device supports 'greaterThan' triggers
    lessThan    | The field device supports 'lessThan' triggers
    hysteresis  | The field device supports 'hysteresis' triggers
    periodic    | The field device supports 'periodic' triggers
    equal       | The field device supports 'equal' triggers
    notEqual    | The field device supports 'notEqual' triggers
    creation    | The field device supports 'creation' triggers
    deletion    | The field device supports 'deletion' triggers
    integerBitwiseAnd | The field device supports 'integerBitwiseAnd' triggers
    octetBitwiseAnd | The field device supports 'octetBitwiseAnd' triggers"

 ::= {fdCondTrigger 1}

fdCondTriggersFrequencyLimit OBJECT-TYPE
SYNTAX      Unsigned32

MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The minimum value of fdCondTriggerObjectFrequency that can be set under
  any condition."
 ::= {fdCondTrigger 2}

fdCondTriggersFrequencyNotes OBJECT-TYPE
SYNTAX      SnmpAdminString

MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "An explanation (or a reference to an explanation, such as with a URL that
  points to a user guide) for any special rules for setting frequency
  limits. For example, if a handful of objects support a zero frequency but
  others have higher limits, the fdCondTriggersFrequencyLimit would be set
  to zero (0), but the notes would provide complete details about what
  limits apply to all objects."
 ::= {fdCondTrigger 3}

fdCondTriggersFires OBJECT-TYPE
SYNTAX      Counter32

MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of times any enabled conditional trigger has evaluated to
  'true' (fired) since the last reboot."
 ::= {fdCondTrigger 4}

fdCondTriggersEvalFailures OBJECT-TYPE
SYNTAX      Counter32

MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of times the evaluation of any enabled conditional trigger has
  failed since the last reboot. For example, a trigger evaluation might fail
  if a variable in an expression cannot be obtained."
REFERENCE   "RFC 2981 mteTriggerFailures"

 ::= {fdCondTrigger 5}

fdCondTriggersActionFailures OBJECT-TYPE
SYNTAX      Counter32

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of times the action called by a conditional trigger failed
    since the last reboot."
 ::= {fdCondTrigger 6}

fdCondTriggerTable OBJECT-TYPE
SYNTAX SEQUENCE OF FdCondTriggerEntry

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "A table containing the configuration and status information for each
    conditional trigger."
REFERENCE "RFC 2981 mteTriggerTable"

 ::= {fdCondTrigger 7}

fdCondTriggerEntry OBJECT-TYPE
SYNTAX FdCondTriggerEntry

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "A row in the fdCondTriggerTable."
REFERENCE "RFC 2981 mteTriggerEntry"
INDEX {fdActionOwner, fdCondTriggerName}
 ::= {fdCondTriggerTable 1}

FdCondTriggerEntry ::= SEQUENCE {
    fdCondTriggerName SnmpAdminString,
    fdCondTriggerDescription SnmpAdminString,
    fdCondTriggerMode INTEGER,
    fdCondTriggerSampleType INTEGER,
    fdCondTriggerValue Integer32,
    fdCondTriggerValue2 Integer32,
    fdCondTriggerValueOctet OCTET STRING,
    fdCondTriggerObject OBJECT IDENTIFIER,
    fdCondTriggerWildcard TruthValue,
    fdCondTriggerObjectTarget SnmpAdminString,
    fdCondTriggerObjectContext SnmpAdminString,
    fdCondTriggerObjectFrequency Unsigned32,
    fdCondTriggerTruthDuration Unsigned32,
    fdCondTriggerStartup TruthValue,
    fdCondTriggerStartup2 TruthValue,
    fdCondTriggerActionOwner SnmpAdminString,
    fdCondTriggerAction SnmpAdminString,
    fdCondTriggerActionOwner2 SnmpAdminString,
    fdCondTriggerAction2 SnmpAdminString,
    fdCondTriggerCfgMessage SnmpAdminString,
    fdCondTriggerFires Counter32,
    fdCondTriggerEvalErrors Counter32,
    fdCondTriggerActionErrors Counter32,
    fdCondTriggerStorageType StorageType,
    fdCondTriggerRowStatus RowStatus }

fdCondTriggerName OBJECT-TYPE
SYNTAX SnmpAdminString

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "A locally-unique, administratively assigned name for the trigger within
    the scope of fdActionOwner."
REFERENCE "NTCIP 2981 mteTriggerName"

 ::= {fdCondTriggerEntry 1}

fdCondTriggerDescription OBJECT-TYPE
SYNTAX SnmpAdminString

```

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "A description of the trigger's function and use."
 REFERENCE "NTCIP 2981 mteTriggerComment"

::= {fdCondTriggerEntry 2}

fdCondTriggerMode OBJECT-TYPE

SYNTAX INTEGER {
 other (1),
 onChange (2),
 greaterThan (3),
 lessThan (4),
 hysteresis (5),
 periodic (6),
 equal (7),
 notEqual (8),
 creation (9),
 deletion (10),
 integerBitwiseAnd (12),
 octetBitwiseAnd (13) }

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The type of trigger test to perform. The following text provides the special rules for each value:

The term 'monitored value' referenced in the text below is

1. The value referenced by the fdCondTriggerObject, if the fdCondTriggerSampleType is 'current' or
2. The current value referenced by the fdCondTriggerObject minus the previously read value of this same object, if the fdCondTriggerSampleType is 'delta'

other:

Fires: Not defined by this standard; use of all objects is undefined.

Startup: Not defined by this standard.

Resets: Not defined by this standard.

onChange:

Fires: fdCondTriggerAction when the monitored value changes (includes creation and deletion)

fdCondTriggerSampleType: Must be 'current'

Startup: Does not fire on initialization

Resets: Upon each firing

greaterThan:

Fires: fdCondTriggerAction when the monitored value is greater than fdCondTriggerValue for a period longer than fdCondTriggerTruthDuration

fdCondTriggerObject: Points to an integer-based object instance

Startup: Based on fdCondTriggerStartup

Resets: When value falls below fdCondTriggerValue for a period longer than fdCondTriggerTruthDuration

lessThan:

Fires: fdCondTriggerAction when the monitored value is less than fdCondTriggerValue for a period longer than fdCondTriggerTruthDuration

fdCondTriggerObject: Points to an integer-based object instance

Startup: Based on fdCondTriggerStartup

Resets: When value rises above fdCondTriggerValue for a period longer than fdCondTriggerTruthDuration

hysteresis:

Fires:

- fdCondTriggerAction when the monitored value is greater than the value of fdCondTriggerValue
- fdCondTriggerAction2 when the monitored value is less than the value of the fdCondTriggerValue2

fdCondTriggerObject: Points to an integer-based object instance

Startup: Based on fdCondTriggerStartup and fdCondTriggerStartup2

Resets:

The fdCondTriggerAction when the fdCondTriggerAction2 fires
The fdCondTriggerAction2 when the fdCondTriggerAction fires

periodic:
Fires: fdCondTriggerAction every fdCondTriggerValue seconds
Startup: Based on fdCondTriggerStartup
Resets: Resets upon firing

equal:
Fires: fdCondTriggerAction when the monitored value is equal to
fdCondTriggerValue for a period longer than
fdCondTriggerTruthDuration
fdCondTriggerObject: Points to an integer-based object instance
Startup: Based on fdCondTriggerStartup
Resets: When value is no longer equal to fdCondTriggerValue for a
period longer than fdCondTriggerTruthDuration

notEqual:
Fires: fdCondTriggerAction when the monitored value is not equal
to fdCondTriggerValue for a period longer than
fdCondTriggerTruthDuration
fdCondTriggerObject: Points to an integer-based object instance
Startup: Based on fdCondTriggerStartup
Resets: When value is equal to fdCondTriggerValue for a period
longer than fdCondTriggerTruthDuration

creation:
Fires: fdCondTriggerAction when the object pointed to by
fdCondTriggerObject is created
Startup: Based on fdCondTriggerStartup; must be false if wildcarded
Resets: Resets upon firing

deletion:
Fires: fdCondTriggerAction when the object pointed to by
fdCondTriggerObject is deleted
Startup: Based on fdCondTriggerStartup; must be false if wildcarded
Resets: Resets upon firing

integerBitwiseAnd:
Fires: fdCondTriggerAction when any bits set in the object pointed
to by fdCondTriggerObject are also set in fdCondTriggerValue for
a period longer than fdCondTriggerTruthDuration.
In other words, a bitwise AND is performed on the values
referenced by fdCondTriggerObject and fdCondTriggerValue and
if any bits are set in the resulting value, the trigger is
fired. The value referenced by fdCondTriggerObject must be
derived from an INTEGER or the evaluation shall fail.
Startup: Based on fdCondTriggerStartup
Resets: When no bits match for a period longer than
fdCondTriggerTruthDuration

octetBitwiseAnd:
Fires: fdCondTriggerAction when any bits set in the object pointed
to by fdCondTriggerObject are also set in
fdCondTriggerValueOctet for a period longer than
fdCondTriggerTruthDuration. In other words, a bitwise AND is
performed on the values referenced by fdCondTriggerObject and
fdCondTriggerValue and if any bits are set in the resulting
value for the defined duration, the trigger is fired. The value
referenced by fdCondTriggerObject must be derived from an OCTET
STRING or BITS; otherwise the evaluation shall fail.
Startup: Based on fdCondTriggerStartup
Resets: When no bits match for a period longer than
fdCondTriggerTruthDuration"

REFERENCE "NTCIP 2981 mteTriggerTest, mteTriggerExistenceTest,
mteTriggerBooleanComparison"

::= {fdCondTriggerEntry 3}

fdCondTriggerSampleType OBJECT-TYPE
SYNTAX INTEGER { other (1), current (1), delta (2) }

MAX-ACCESS read-create
STATUS current
DESCRIPTION
"An indication of the value to monitor. Allowed values are:
Value | Description
other | Not defined in this standard

current | The monitored value shall be the current value of the object
 | instance referenced by fdCondTriggerObject
 delta | The monitored value shall be the value calculated by taking
 | the current value of the object instance referenced by
 | fdCondTriggerObject and subtracting the previously reported
 | value of the same object.

NOTE: Delta operations require at least two readings, separated by
 fdCondTriggerObjectFrequency, for a comparison to be made."

REFERENCE "NTCIP 2981 mteTriggerSampleType"

::= {fdCondTriggerEntry 4}

fdCondTriggerValue OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value to use for the test specified by fdCondTriggerMode. This object
 is ignored when the mode is 'onChange', 'periodic', 'creation', 'deletion',
 or 'octetBitwiseAnd'. For the 'hysteresis' mode, this value shall represent
 the rising threshold. NOTE: SNMP does not allow the definition of 64-bit
 values. When using this object, the scope of the comparison check is
 limited to signed 32-bit values, the conditional trigger shall allow
 comparisons between this value and other INTEGER types (e.g., Unsigned32,
 Counter64, etc). Such comparisons shall be processed in a manner that
 follows the conceptual mathematical logic. For example, a conforming
 implementation shall determine that an Integer32 value of 0xFF FF FF FF
 (-1) is less than Unsigned32 value of 0xFF FF FF FE (4,294,967,295)."

REFERENCE "NTCIP 2981 mteTriggerBooleanValue, mteTriggerThresholdRising"

::= {fdCondTriggerEntry 5}

fdCondTriggerValue2 OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"When the fdCondTriggerMode value is 'hysteresis' mode, this value shall
 represent the falling threshold. This value shall be ignored for all
 other values of fdCondTriggerMode."

REFERENCE "NTCIP 2981 mteTriggerThresholdFalling"

::= {fdCondTriggerEntry 6}

fdCondTriggerValueOctet OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value to use for the test when the fdCondTriggerMode is
 'octetBasedBitwiseAnd'."

::= {fdCondTriggerEntry 7}

fdCondTriggerObject OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The object identifier of the MIB object instance to sample to see if the
 trigger should fire.

If the fdCondTriggerMode is 'onChange', 'creation', or 'deletion', this
 value may be wildcarded by truncating all or part of the instance
 portion, in which case the value is obtained as if with a GetNext
 function, checking multiple instances of the same object type. To
 implement such wildcarding fdCondTriggerWildcard must be 'true' and if
 not it must be 'false'.

Bad object identifiers or a mismatch between truncating the identifier and the value of fdCondTriggerWildcard result in an operation as one would expect when providing the wrong identifier to a Get or GetNext operation. The Get will fail or get the wrong object. The GetNext will indeed get whatever is next, proceeding until it runs past the initial part of the identifier and perhaps many unintended objects for confusing results.

Each instance that fills the wildcard is independent of any additional instances, that is, wildcarded objects operate as if there were a separate table entry for each instance that fills the wildcard without having to actually predict all possible instances ahead of time."

REFERENCE "NTCIP 2981 mteTriggerValueID"

::= {fdCondTriggerEntry 8}

fdCondTriggerWildcard OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Control for whether fdCondTriggerObject is to be treated as fully-specified or wildcarded, with 'true' indicating wildcard. This value is ignored (i.e., always treated as 'false') if the value of fdCondTriggerMode is not 'onChange', 'creation', or 'deletion'."

REFERENCE "NTCIP 2981 mteTriggerValueIDWildcard"

::= {fdCondTriggerEntry 9}

-- This MIB limits the scope of this object to creation, deletion, and change
-- events.

fdCondTriggerObjectTarget OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE(1..32))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The unique identifier associated with the snmpTargetAddrEntry from which to obtain the value to be checked.

A length of 0 indicates the local system. In this case, access to the object(s) indicated by fdCondTriggerObject is under the security credentials of the requester that set fdCondTriggerRowStatus to 'active'. Those credentials are the input parameters for isAccessAllowed from the Architecture for Describing SNMP Management Frameworks (RFC 3411). Otherwise, for values other than a zero-length string, access rights are checked according to the security parameters for the indicated target as defined in the snmpTargetParamsTable."

REFERENCE "NTCIP 2981 mteTriggerTargetTag"

::= {fdCondTriggerEntry 10}

fdCondTriggerObjectContext OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The management context from which to obtain fdCondTriggerObject. For example, if a device is a proxy agent, it might support local data in addition to data representing one or more remote devices. Each set of data is contained within a named context per the rules defined in the Architecture for Describing SNMP Management Frameworks (RFC 3411)."

REFERENCE "NTCIP 2981 mteTriggerContextName"

::= {fdCondTriggerEntry 11}

fdCondTriggerObjectFrequency OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The number of seconds to wait between trigger samples. To encourage consistency in sampling, the interval is measured from the beginning of one check to the beginning of the next and the timer is restarted immediately when it expires, not when the check completes. If the next sample begins before the previous one completed the system shall record an evalError. A frequency of 0 indicates instantaneous recognition of the condition. This is not possible in many cases, but may be supported in cases where it makes sense and the system is able to do so. This feature allows the MIB to be used in implementations where such interrupt-driven behavior is possible and is not likely to be supported for all MIB objects even then since such sampling generally has to be tightly integrated into low-level code. Systems that can support a value of zero SHOULD document those cases where it can be used. In cases where it can not, setting this object to 0 shall result in an 'inconsistentValue' error."

REFERENCE "NTCIP 2981 mteTriggerFrequency"

::= {fdCondTriggerEntry 12}

fdCondTriggerTruthDuration OBJECT-TYPE
 SYNTAX Unsigned32

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"If the sample type is 'current' and the mode is 'greaterThan', 'lessThan', 'equal', 'notEqual', 'integerBitwiseAnd', or 'octetBitwiseAnd', the logic within this paragraph applies. If the frequency is zero (0), this value indicates the duration, in tenths of seconds, that the condition must be true prior to the trigger firing. If the frequency is greater than zero (0), this value represents the number of sequential times the the condition must evaluate to true before the trigger fires. In the latter case, the value zero (0) shall operate as if it were a one (1). For example, if the mode is 'equal' and the frequency is 60, the field device will check the object instance every minute. If the fdCondTriggerTruthDuration is set to zero or one, the trigger will fire the first time the evaluation returns 'true'. If the value of fdCondTriggerTruthDuration is two, the trigger will only fire when the evaluation returns 'true' two times in a row. In either case, the trigger will not reset until the evaluation returns false.

If the sample type is 'delta' or if the mode is 'onChange', 'hysteresis', 'periodic', 'creation', or 'deletion', this value shall be ignored and the trigger will fire upon the monitored condition initially evaluating to true."

::= {fdCondTriggerEntry 13}

fdCondTriggerStartup OBJECT-TYPE
 SYNTAX TruthValue

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The initial state of the trigger upon fdCondTriggerEnabled being enabled. In the case of the 'greaterThan', 'lessThan', 'equal', and 'notEqual' modes, if this value is true, the trigger shall initiate in a ready-to-fire state upon fdCondTriggerEnabled changing to 'enabled'. As a result, if the condition specified by this trigger is true when placed in the 'enabled' state, the trigger will immediately fire. It will then have to reset before firing again. If this value is false, the trigger shall initiate in a fired state upon fdCondTriggerEnabled changing to 'enabled'. As a result, if the condition specified by this trigger is true when placed in the 'enabled' state, the trigger will not fire immediately; instead, the trigger will first have to reset before firing.

In the case of the 'hysteresis' mode, this object indicates the initial state of the rising trigger; the initial state of the falling trigger shall be defined by fdCondTriggerStartupReady2.

In the case of the 'periodic' mode, this value shall indicate whether the trigger will fire instantaneously (true) upon enablement and every fdCondTriggerFrequency seconds thereafter, or if it will only start firing after the first fdCondTriggerFrequency seconds after being enabled.

In the case of the 'creation' and 'deletion' modes, this value shall be 'false' if fdCondTriggerWildcard is 'true'; otherwise, the value of fdCondTriggerRowStatus shall automatically transition to 'notReady'. The first reading of existing object instances shall be used to identify the baseline from which the trigger will detect creation or deletion events; the trigger will then reset after this initialization. In the case of 'creation' or 'deletion' modes with fdCondTriggerWildcard set to 'false', this value shall operate as described above.

In the case of the 'onChange' mode, this value shall be ignored. The initial check in the onChange mode is always used to set the baseline against which changes are detected."

REFERENCE "NTCIP 2981 mteTriggerBooleanStartup"

::= {fdCondTriggerEntry 14}

fdCondTriggerStartup2 OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the value of fdCondTriggerMode is 'hysteresis', this object indicates the initial state of the falling trigger. The initial state of the rising trigger shall be defined by fdCondTriggerStartup. This object is ignored for all other values of fdCondTriggerMode."

REFERENCE "NTCIP 2981 mteTriggerThresholdStartup"

::= {fdCondTriggerEntry 15}

fdCondTriggerActionOwner OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The fdActionOwner of the action to be implemented when this trigger fires. In the case of the hysteresis mode, this only represents the action to be implemented for the rising trigger."

REFERENCE "NTCIP 2981 mteTriggerExistenceEventOwner, mteTriggerBooleanEventOwner, mteTriggerThresholdRisingEventOwner, mteTriggerThresholdDeltaRisingEventOwner"

::= {fdCondTriggerEntry 16}

fdCondTriggerAction OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The fdActionName of the action to be implemented when this trigger fires. In the case of the hysteresis mode, this only represents the action to be implemented for the rising trigger."

REFERENCE "NTCIP 2981 mteTriggerExistenceEvent, mteTriggerBooleanEvent, mteTriggerThresholdRisingEvent, mteTriggerThresholdDeltaRisingEvent"

::= {fdCondTriggerEntry 17}

fdCondTriggerActionOwner2 OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If the mode is hysteresis, this object specifies the fdActionOwner of the

action to be implemented when the falling trigger fires. This value is ignored for all other modes."

REFERENCE "NTCIP 2981 mteTriggerThresholdFallingEventOwner, mteTriggerThresholdDeltaFallingEventOwner"

::= {fdCondTriggerEntry 18}

fdCondTriggerAction2 OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If mode is hysteresis, this object specifies the fdActionName of the action to be implemented when the falling trigger fires. This value is ignored for all other modes."

REFERENCE "NTCIP 2981 mteTriggerThresholdFallingEvent, mteTriggerThresholdDeltaFallingEvent"

::= {fdCondTriggerEntry 19}

fdCondTriggerCfgMessage OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Implementation-specific message that may be used by the field device to report errors and explain why the status is notInService or notReady."

::= {fdCondTriggerEntry 20}

fdCondTriggerFires OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times this conditional trigger has evaluated to 'true' (fired) since the last reboot."

::= {fdCondTriggerEntry 21}

fdCondTriggerEvalErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times the evaluation of this conditional trigger has failed since the last reboot. For example, a trigger evaluation might fail if a variable in an expression cannot be obtained."

::= {fdCondTriggerEntry 22}

fdCondTriggerActionErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times the action called by this conditional trigger failed since the last reboot."

::= {fdCondTriggerEntry 23}

fdCondTriggerStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

```

::= {fdCondTriggerEntry 24}

fdCondTriggerRowStatus OBJECT-TYPE
    SYNTAX      RowStatus

    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this conceptual row.

        Any attempt to modify any read-create object within this conceptual row
        while the value of this object is active(1) shall result in an
        inconsistentValue error."
    REFERENCE   "NTCIP 2981 mteTriggerEntryStatus"

::= {fdCondTriggerEntry 25}

-- *****
-- A.2.4 Conformance Information
-- *****

fdCondTriggerMIBCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The conformance statement for conditional triggers."
    MODULE      -- this module
        MANDATORY-GROUPS {
            fdCondTriggerCapabilitiesGroup,
            fdCondTriggerManagementGroup,
            fdCondTriggerConfigurationGroup,
            fdCondTriggerStatusGroup,
            fdCondTriggersStatusGroup,
            fdCondTriggerMessageGroup
        }
    ::= {fdCondTriggerCompliances 1}

fdCondTriggerCapabilitiesGroup OBJECT-GROUP
    OBJECTS {
        fdCondTriggersSupport,
        fdCondTriggersFrequencyLimit,
        fdCondTriggersFrequencyNotes
    }
    STATUS      current
    DESCRIPTION
        "Management information that identifies the capabilities of conditional
        triggers."
    REFERENCE   "Clause 6.2.2.1"
    ::= {fdCondTriggerGroups 1}

fdCondTriggerManagementGroup OBJECT-GROUP
    OBJECTS {
        fdCondTriggerRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Management information that provides for creation, deletion, and
        management of a conditional trigger."
    REFERENCE   "Clause 6.2.2.2, Clause 6.2.2.3, Clause 6.2.2.4, Clause 6.2.2.7,
        Clause 6.2.2.8"
    ::= {fdCondTriggerGroups 2}

fdCondTriggerConfigurationGroup OBJECT-GROUP
    OBJECTS {
        fdCondTriggerDescription,
        fdCondTriggerMode,
        fdCondTriggerSampleType,
        fdCondTriggerValue,
        fdCondTriggerValue2,
        fdCondTriggerValueOctet,
        fdCondTriggerObject,
        fdCondTriggerWildcard,

```

```

        fdCondTriggerObjectTarget,
        fdCondTriggerObjectContext,
        fdCondTriggerObjectFrequency,
        fdCondTriggerTruthDuration,
        fdCondTriggerStartup,
        fdCondTriggerStartup2,
        fdCondTriggerActionOwner,
        fdCondTriggerAction,
        fdCondTriggerActionOwner2,
        fdCondTriggerAction2,
        fdCondTriggerStorageType
    }
    STATUS      current
    DESCRIPTION
        "Management information used for configuration of an conditional trigger."
    REFERENCE "Clause 6.2.2.2, Clause 6.2.2.3"
    ::= {fdCondTriggerGroups 3}

fdCondTriggerStatusGroup OBJECT-GROUP
    OBJECTS      {
        fdCondTriggerFires,
        fdCondTriggerEvalErrors,
        fdCondTriggerActionErrors
    }
    STATUS      current
    DESCRIPTION
        "Management information that identifies the status of the conditional
        trigger."
    REFERENCE "Clause 6.2.2.5"
    ::= {fdCondTriggerGroups 4}

fdCondTriggersStatusGroup OBJECT-GROUP
    OBJECTS      {
        fdCondTriggersFires,
        fdCondTriggersEvalFailures,
        fdCondTriggersActionFailures
    }
    STATUS      current
    DESCRIPTION
        "Management information that identifies the status of all conditional
        trigger."
    REFERENCE "Clause 6.2.2.6"
    ::= {fdCondTriggerGroups 5}

fdCondTriggerMessageGroup OBJECT-GROUP
    OBJECTS      {
        fdCondTriggerCfgMessage
    }
    STATUS      current
    DESCRIPTION
        "Management information that provides information about configuration
        errors."
    REFERENCE "Clause 6.2.2.3"
    ::= {fdCondTriggerGroups 6}

END
-- ASN1END

```

A.3 Day Plan MIB

```

-- *****
-- A.3.1 Day Plan Header
-- *****
DAY-PLAN-MIB DEFINITIONS ::= BEGIN
IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY, Counter32, Unsigned32
    FROM SNMPv2-SMI
    -- RFC 2578

```

```

TruthValue, StorageType, RowStatus

FROM SNMPv2-TC
-- RFC 2579

MODULE-COMPLIANCE, OBJECT-GROUP

FROM SNMPv2-CONF
-- RFC 2580

SnmpAdminString

FROM SNMP-FRAMEWORK-MIB
-- RFC 3411

ITSDailyTimeStamp, ITSDateStamp, ITSPduErrorStatus, fieldDevice, iso20684p3
FROM FIELD-DEVICE-TC-MIB
-- ISO 20684-1 Annex A

;
fdDayPlanMIB MODULE-IDENTITY
    LAST-UPDATED "201912312026Z"
    ORGANIZATION "ISO TC 204 WG 9"
    CONTACT-INFO
        "name: Kenneth Vaughn

        phone: +1-571-331-5670

        email: kvaughn@trevilon.com

        postal: 6606 FM 1488 RD STE 148-503

                Magnolia, TX 77354

                USA"
    DESCRIPTION
        "This MIB defines a mechanism by which a manager can schedule a series of
        actions to occur during one day (from local midnight to local midnight)."

    REVISION "201912312026Z"
    DESCRIPTION
        "Initial revision of the document as proposed for CD ballot"

    ::= {iso20684p3 3}

-- *****
-- A.3.2 Node Definitions
-- *****

fdDayPlanConformance OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "A node containing conformance statements related to the fdDayPlanMIB, as
        defined in ISO/TS 20684-3."
    ::= {fdDayPlanMIB 2}

fdDayPlanCompliances OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "A node for compliance statements for the fdDayPlanMIB."
    ::= {fdDayPlanConformance 1}

fdDayPlanMIBGroups OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "A node for group definitions related to fdObjectSetMIB."
    ::= {fdDayPlanConformance 2}

fdDayPlan OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "A node defining management information related to the field device's Day
        Plan Schedule."
    ::= {fieldDevice 6}

```

```

-- *****
-- Day Plan Schedule Objects
-- *****

fdDayPlanScheduleTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF FdDayPlanScheduleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A dynamic table that uses fdClockLocalDate to determine which day plan
        should be active. To determine the current day plan, each row in this
        table, with a RowStatus of 'active', shall be evaluated to identify the
        most specific rule that is currently applicable. The field device shall
        then apply the day plan associated with that rule.

        A row is only applicable for the current day if the current
        fdClockLocalDate meets the conditions specified by fdDayPlanScheduleMonth,
        fdDayPlanScheduleDayOfMonth, and fdDayPlanScheduleDayOfWeek. If multiple
        entries are applicable, the field device shall identify those with the
        minimum number of bits set within fdDayPlanScheduleMonth. If multiple
        entries are identified, the field device will identify those with the
        minimum number of bits set in fdDayPlanScheduleDayOfMonth. If multiple
        entries are still identified, the field device will identify those with the
        minimum number of bits set in fdDayPlanScheduleDayOfWeek. If multiple
        entries are still identified, the field device will select the remaining
        entry with the lowest value for fdDayPlanScheduleNumber.

        If no entry is applicable, no day plan will be active."
    REFERENCE   "NTCIP 2101 2.4.3.2"

    ::= { fdDayPlan 1 }

fdDayPlanScheduleEntry OBJECT-TYPE
    SYNTAX      FdDayPlanScheduleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the fdDayPlanScheduleTable."
    REFERENCE   "NTCIP 1201 2.4.3.2"
    INDEX       { fdDayPlanScheduleNumber }
    ::= { fdDayPlanScheduleTable 1 }

FdDayPlanScheduleEntry ::= SEQUENCE {
    fdDayPlanScheduleNumber      Unsigned32,
    fdDayPlanScheduleDescription SnmpAdminString,
    fdDayPlanScheduleMonth       BITS,
    fdDayPlanScheduleDayOfWeek   BITS,
    fdDayPlanScheduleDayOfMonth  BITS,
    fdDayPlanScheduleDayPlan      Unsigned32,
    fdDayPlanScheduleStorageType  StorageType,
    fdDayPlanScheduleRowStatus    RowStatus }

fdDayPlanScheduleNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index used to identify the entry within the fdDayPlanScheduleTable."
    REFERENCE   "NTCIP 1201 2.4.3.2.1"

    ::= { fdDayPlanScheduleEntry 1 }

fdDayPlanScheduleDescription OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A textual description of the day plan selection rule."
    ::= { fdDayPlanScheduleEntry 2 }
-- Proposed object

```

fdDayPlanScheduleMonth OBJECT-TYPE

```

SYNTAX      BITS {
                reserved (0),  january (1),  february (2),
                march (3),     april (4),    may (5),
                june (6),      july (7),     august (8),
                september (9), october (10),  november (11),
                december (12) }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The month(s) of the year during which this entry is applicable."
REFERENCE   "NTCIP 1201 2.4.3.2.2"
::= {fdDayPlanScheduleEntry 3}

```

```

-- NTCIP uses INTEGERS to convey the information, which results in a violation
-- of SNMPv1 rules against a 32-bit unsigned integer. The proposal uses the
-- SNMP convention of a BITS structure to present bit-level data (which
-- reverses the order of the bitmap).

```

fdDayPlanScheduleDayOfWeek OBJECT-TYPE

```

SYNTAX      BITS {
                reserved (0),  monday (1),  tuesday (2),
                wednesday (3), thursday (4), friday (5),
                saturday (6),  sunday (7) }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The day(s) of week during which this entry is applicable."
REFERENCE   "NTCIP 1201 2.4.3.2.3"

```

```

::= {fdDayPlanScheduleEntry 4}
-- NTCIP uses INTEGERS to convey the information, which results in a violation
-- of SNMPv1 rules against a 32-bit unsigned integer. The proposal uses the
-- SNMP convention of a BITS structure to present bit-level data (which
-- reverses the order of the bitmap).

```

fdDayPlanScheduleDayOfMonth OBJECT-TYPE

```

SYNTAX      BITS {
                reserved (0),  first (1),      second (2),
                third (3),     fourth (4),     fifth (5),
                sixth (6),     seventh (7),    eighth (8),
                ninth (9),     tenth (10),     eleventh (11),
                twelfth (12),  thirteenth (13),  fourteenth (14),
                fifteenth (15), sixteenth (16),  seventeenth (17),
                eighteenth (18), nineteenth (19), twentieth (20),
                twenty-first (21), twenty-second (22), twenty-third (23),
                twenty-fourth (24), twenty-fifth (25), twenty-sixth (26),
                twenty-seventh (27), twenty-eighth (28), twenty-ninth (29),
                thirtieth (30), thirty-first (31) }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The day(s) of the month during which this entry is applicable."
REFERENCE   "NTCIP 1201 2.4.3.2.4"

```

```

::= {fdDayPlanScheduleEntry 5}
-- NTCIP uses INTEGERS to convey the information, which results in a violation
-- of SNMPv1 rules against a 32-bit unsigned integer. The proposal uses the
-- SNMP convention of a BITS structure to present bit-level data (which
-- reverses the order of the bitmap).

```

fdDayPlanScheduleDayPlan OBJECT-TYPE

```

SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The day plan that shall be active when this entry is selected."
REFERENCE   "NTCIP 1201 2.4.3.2.5"

```

```

::= {fdDayPlanScheduleEntry 6}

```

```

fdDayPlanScheduleStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The storage type for this conceptual row. Conceptual rows having the
        value 'permanent' need not allow write-access to any columnar objects in
        the row."
    ::= {fdDayPlanScheduleEntry 7}
-- Proposed object to conform to SNMP conventions

fdDayPlanScheduleRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this conceptual row. This object may be used to create or
        remove rows from this table.
        A row that contains a value for fdScheduleMonth, fdScheduleDayOfMonth, or
        fdScheduleDayOfWeek where no bits are assigned shall have a status of
        notReady.

        Any attempt to modify fdDayPlanScheduleStorageType within this conceptual
        row while the value of this object is active(1) shall result in an
        inconsistentValue error."
    ::= {fdDayPlanScheduleEntry 8}

fdDayPlanSchedulerEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "A control object that indicates whether the scheduler will result in the
        selected day plan issuing commands. The scheduler shall always process the
        rules to select a day plan and reflect that selection in the value of
        fdDayPlanSchedulerCurrentDayPlan. When this object has a value of 'true',
        the day plan scheduler shall also issue commands according to the entries
        in the selected day plan table."
    ::= {fdDayPlan 2}
-- Proposed object

fdDayPlanSchedulerSelectedRule OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The fdDayPlanScheduleNumber for the entry in the fdDayPlanScheduleTable
        that has been selected according to the defined logic. A value of zero
        shall indicate that no entry in the fdDayPlanScheduleTable is applicable
        for the current date."
    REFERENCE  "NTCIP 1201 2.4.4.5"
    ::= {fdDayPlan 3}

fdDayPlanSchedulerCurrentDayPlan OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The day plan that has been selected by the day plan schedule logic. This
        shall be identical to value of the fdDayPlanScheduleDayPlan object for the
        selected entry of the fdDayPlanScheduleTable with a
        fdDayPlanScheduleNumber equal to fdDayPlanSchedulerSelectedRule if, and
        only if, the associated fdDayPlanRowStatus is 'active'. If no rule is
        applicable for the current date, or if the indicated day plan is not
        active, this object shall have a value of zero."
    REFERENCE  "NTCIP 1201 2.4.4.4"
    ::= {fdDayPlan 4}

fdDayPlanSchedulerCallCounter OBJECT-TYPE

```

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The total number of commands that have been called by the day plan
    scheduler since the last reboot. This counter will only increment by one,
    even if the called command is sent to multiple targets per the
    fdCommandTable."
 ::= {fdDayPlan 5}
-- Proposed object

fdDayPlanSchedulerCallFailures OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of calls to a command that resulted in an 'errorNotActive' or
    'errorPending' value for fdCommandLastAttemptStatus."
 ::= {fdDayPlan 6}
-- Proposed object

fdDayPlanSchedulerLastError OBJECT-TYPE
SYNTAX      ITSPduErrorStatus
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The SNMP error status received in response to the last command issued by
    the scheduler that resulted in a non-zero error status."
 ::= {fdDayPlan 7}

fdDayPlanSchedulerLastErrorTime OBJECT-TYPE
SYNTAX      ITSDailyTimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The local time of day that the last non-zero error status was received in
    response to a command issued by the day plan scheduler."
 ::= {fdDayPlan 8}

fdDayPlanSchedulerLastErrorDate OBJECT-TYPE
SYNTAX      ITSDateStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The date on which the last non-zero error status was received in response
    to command issued by the day plan scheduler."
 ::= {fdDayPlan 9}

fdDayPlanTable OBJECT-TYPE
SYNTAX      SEQUENCE OF FdDayPlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The day plan table that indicates the commands that are issued each day."
 ::= {fdDayPlan 10}

fdDayPlanEntry OBJECT-TYPE
SYNTAX      FdDayPlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the day plan table."

INDEX      {fdDayPlanNumber}
 ::= {fdDayPlanTable 1}

FdDayPlanEntry ::= SEQUENCE {
    fdDayPlanNumber      Unsigned32,
    fdDayPlanDescription SnmpAdminString,
    fdDayPlanStorageType StorageType,
    fdDayPlanRowStatus   RowStatus }

```

```

fdDayPlanNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index used to identify the instance of the day plan."
    ::= {fdDayPlanEntry 1}

fdDayPlanDescription OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A description for the day plan."
    ::= {fdDayPlanEntry 2}

fdDayPlanStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The storage type for this conceptual row and all related conceptual rows
        in the fdDayPlanTriggerTable.

        Conceptual rows having the value 'permanent' must allow write-access to
        fdDayPlanDescription and full write, create, and delete access to all
        objects for all rows in the fdDayPlanTriggerTable. (i.e., otherwise, this
        object should have the value 'read-only' rather than 'permanent')."
    ::= {fdDayPlanEntry 3}

fdDayPlanRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of the conceptual row. Data contained within this row can not
        be modified when in the 'active' state. A request to 'destroy' a conceptual
        row of this table shall also result in destroying all associated conceptual
        rows in the fdDayPlanTriggerTable."
    ::= {fdDayPlanEntry 4}

fdDayPlanTriggerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF FdDayPlanTriggerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A dynamic table where each entry defines an action to be performed for a
        specified day plan at a specified time."
    ::= {fdDayPlan 11}

fdDayPlanTriggerEntry OBJECT-TYPE
    SYNTAX      FdDayPlanTriggerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the fdDayPlanActionTable."

    INDEX      {fdDayPlanNumber, fdDayPlanTriggerTime}
    ::= {fdDayPlanTriggerTable 1}

FdDayPlanTriggerEntry ::= SEQUENCE {
    fdDayPlanTriggerTime      ITSDailyTimeStamp,
    fdDayPlanTriggerOwner     SnmpAdminString,
    fdDayPlanTriggerName      Unsigned32,
    fdDayPlanTriggerRowStatus RowStatus}

fdDayPlanTriggerTime OBJECT-TYPE
    SYNTAX      ITSDailyTimeStamp
    MAX-ACCESS  not-accessible
    STATUS      current

```

```

DESCRIPTION
    "The local time of day at which the action shall be taken."
 ::= {fdDayPlanTriggerEntry 1}

fdDayPlanTriggerOwner OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The fdActionOwner from the fdActionTable for the action to be taken."
 ::= {fdDayPlanTriggerEntry 2}

fdDayPlanTriggerName OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The fdActionName from the fdActionTable for the action to be taken."
 ::= {fdDayPlanTriggerEntry 3}

fdDayPlanTriggerRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of the conceptual row. Data contained within this row can not
        be modified when in the 'active' state. An 'active' row of this table shall
        only call an action if the corresponding row in the fdDayPlanTable is also
        'active'."
 ::= {fdDayPlanTriggerEntry 4}

-- *****
-- A.3.4 Conformance Information
-- *****

fdDayPlanMIBCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The conformance statement for day plans."
    MODULE      -- this module
        MANDATORY-GROUPS {
            fdDayPlanManagementGroup,
            fdDayPlanConfigurationGroup,
            fdDayPlanTriggerManagementGroup,
            fdDayPlanTriggerConfigurationGroup,
            fdDayPlanSchedulerMgmtGroup,
            fdDayPlanSchedulerCfgGroup,
            fdDayPlanSchedulerStatusGroup,
            fdDayPlanSchedulerStatisticsGroup,
            fdDayPlanSchedulerEnableGroup,
            fdDayPlanSchedulerErrorGroup
        }
    ::= { fdDayPlanCompliances 1 }

fdDayPlanManagementGroup OBJECT-GROUP
    OBJECTS      {
        fdDayPlanRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Management information that provides for creation, deletion, and
        management of an fdDayPlan."
    REFERENCE    "Clause 6.3.2.1, Clause 6.3.2.2, Clause 6.3.2.3, Clause 6.3.2.4,
    Clause 6.3.2.5"
    ::= {fdDayPlanMIBGroups 1}

fdDayPlanConfigurationGroup OBJECT-GROUP
    OBJECTS      {
        fdDayPlanDescription,

```

```

        fdDayPlanStorageType
    }
    STATUS current
    DESCRIPTION
        "Management information used for configuration of an fdDayPlan."
    REFERENCE "Clause 6.3.2.1, Clause 6.3.2.2"
    ::= {fdDayPlanMIBGroups 2}

fdDayPlanTriggerManagementGroup OBJECT-GROUP
    OBJECTS {
        fdDayPlanTriggerRowStatus
    }
    STATUS current
    DESCRIPTION
        "Management information that provides for creation, deletion, and
        management of an fdDayPlanTriggerEntry."
    REFERENCE "Clause 6.3.2.6, Clause 6.3.2.7, Clause 6.3.2.8, Clause 6.3.2.9,
    Clause 6.3.2.10"
    ::= {fdDayPlanMIBGroups 3}

fdDayPlanTriggerConfigurationGroup OBJECT-GROUP
    OBJECTS {
        fdDayPlanTriggerOwner,
        fdDayPlanTriggerName
    }
    STATUS current
    DESCRIPTION
        "Management information used for configuration of an
        fdDayPlanTriggerEntry."
    REFERENCE "Clause 6.3.2.6, Clause 6.3.2.7"
    ::= {fdDayPlanMIBGroups 4}

fdDayPlanSchedulerMgmtGroup OBJECT-GROUP
    OBJECTS {
        fdDayPlanScheduleRowStatus
    }
    STATUS current
    DESCRIPTION
        "Management information that provides for creation, deletion, and
        management of the a fdDayPlanScheduleEntry."
    REFERENCE "Clause 6.4.2.1, Clause 6.4.2.2, Clause 6.4.2.3, Clause 6.4.2.4"
    ::= {fdDayPlanMIBGroups 5}

fdDayPlanSchedulerCfgGroup OBJECT-GROUP
    OBJECTS {
        fdDayPlanScheduleDescription,
        fdDayPlanScheduleMonth,
        fdDayPlanScheduleDayOfWeek,
        fdDayPlanScheduleDayOfMonth,
        fdDayPlanScheduleDayPlan,
        fdDayPlanScheduleStorageType
    }
    STATUS current
    DESCRIPTION
        "Management information used for the configuration of an
        fdDayPlanScheduleEntry."
    REFERENCE "Clause 6.4.2.1, Clause 6.4.2.2"
    ::= {fdDayPlanMIBGroups 6}

fdDayPlanSchedulerStatusGroup OBJECT-GROUP
    OBJECTS {
        fdDayPlanSchedulerSelectedRule,
        fdDayPlanSchedulerCurrentDayPlan
    }
    STATUS current
    DESCRIPTION
        "Management information used to determine the status of the day plan
        scheduler."
    REFERENCE "Clause 6.4.2.5"
    ::= {fdDayPlanMIBGroups 7}

```

```

fdDayPlanSchedulerStatisticsGroup OBJECT-GROUP
    OBJECTS      {
        fdDayPlanSchedulerCallCounter,
        fdDayPlanSchedulerCallFailures
    }
    STATUS        current
    DESCRIPTION   "Management information used to monitor the statistics related to the day
        plan scheduler."
    REFERENCE     "Clause 6.4.2.6"
    ::= {fdDayPlanMIBGroups 8}

fdDayPlanSchedulerEnableGroup OBJECT-GROUP
    OBJECTS      {
        fdDayPlanSchedulerEnable
    }
    STATUS        current
    DESCRIPTION   "Management information that provides for enabling and disabling the day
        plan scheduler."
    REFERENCE     "Clause 6.4.2.5, Clause 6.4.2.7"
    ::= {fdDayPlanMIBGroups 9}

fdDayPlanSchedulerErrorGroup OBJECT-GROUP
    OBJECTS      {
        fdDayPlanSchedulerLastError,
        fdDayPlanSchedulerLastErrorTime,
        fdDayPlanSchedulerLastErrorDate
    }
    STATUS        current
    DESCRIPTION   "Management information that provides for receiving error information
        regarding the day plan scheduler."
    REFERENCE     "Clause 6.4.2.8"
    ::= {fdDayPlanMIBGroups 10}

END
-- ASN1END

```

A.4 Trigger Schedule MIB

```

-- *****
-- A.4.1 Trigger Schedule Header
-- *****
TRIGGER-SCHED-MIB DEFINITIONS ::= BEGIN
IMPORTS

    MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY, Counter32
        FROM SNMPv2-SMI
        -- RFC 2578

    StorageType, RowStatus
        FROM SNMPv2-TC
        -- RFC 2579

    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
        -- RFC 2580

    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
        -- RFC 3411

    ITSDailyTimeStamp, ITSDateStamp, fieldDevice, iso20684p3
        FROM FIELD-DEVICE-TC-MIB
        -- ISO 20684-1 Annex A

    fdActionOwner
        FROM ACTION-MIB
        -- ISO/TS 20684-3
        -- Annex A.1

;

```

```

fdTriggerSchedMIB MODULE-IDENTITY
    LAST-UPDATED "202001012026Z"
    ORGANIZATION "ISO TC 204 WG 9"
    CONTACT-INFO
        "name: Kenneth Vaughn
         phone: +1-571-331-5670
         email: kvaughn@trevilon.com
         postal: 6606 FM 1488 RD STE 148-503
                Magnolia, TX 77354
                USA"
    DESCRIPTION
        "This MIB defines a mechanism by which a manager can schedule actions to be
         performed at specified times and dates, regardless of other scheduled
         actions."

    REVISION "202001012026Z"
    DESCRIPTION
        "Initial revision of the document as proposed for CD ballot."

::= {iso20684p3 4}

-- *****
-- A.4.2 Node Definitions
-- *****

fdTriggerSchedConformance OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node containing conformance statements related to the fdTriggerSchedMIB,
         as defined in ISO/TS 20684-3."
::= {fdTriggerSchedMIB 2}

fdTriggerSchedCompliances OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node for compliance statements for the fdTriggerSchedMIB."
::= {fdTriggerSchedConformance 1}

fdTriggerSchedGroups OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node for group definitions related to fdTriggerSchedMIB."
::= {fdTriggerSchedConformance 2}

fdTriggerSched OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "A node defining management information related to the field device's
         Trigger Schedule."
::= {fieldDevice 7}

-- *****
-- A.4.3 Trigger Schedule
-- *****

fdTriggerScheduleTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF FdTriggerScheduleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that manages when actions are to be performed by the field
         device."
    REFERENCE   "RFC 3231 schedTable"
::= {fdTriggerSched 1}

fdTriggerScheduleEntry OBJECT-TYPE
    SYNTAX      FdTriggerScheduleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry defining when to perform a scheduled action."

```

Each trigger schedule defines time(s) at which specific action(s) are to be performed. The defined times are relative to the local timezone, which may be affected by summer time schedules."

REFERENCE "RFC 3231 schedEntry"

INDEX {fdActionOwner, fdTriggerScheduleName}

::= {fdTriggerScheduleTable 1}

```
FdTriggerScheduleEntry ::= SEQUENCE {
    fdTriggerScheduleName          SnmpAdminString,
    fdTriggerScheduleDescription   SnmpAdminString,
    fdTriggerScheduleWeekDay       BITS,
    fdTriggerScheduleMonth         BITS,
    fdTriggerScheduleDay           BITS,
    fdTriggerScheduleHour          BITS,
    fdTriggerScheduleMinute        BITS,
    fdTriggerScheduleType          INTEGER,
    fdTriggerScheduleActionOwner   SnmpAdminString,
    fdTriggerScheduleAction        SnmpAdminString,
    fdTriggerScheduleCount         Counter32,
    fdTriggerScheduleFailures      Counter32,
    fdTriggerScheduleLastFailedDate ITSDateStamp,
    fdTriggerScheduleLastFailedTime ITSDailyTimeStamp,
    fdTriggerScheduleStorageType   StorageType,
    fdTriggerScheduleRowStatus     RowStatus }
```

fdTriggerScheduleName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (1..32))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The locally-unique, administratively assigned name for this entry. This object allows a fdActionManagerOwner to have multiple entries in the table."

REFERENCE "RFC 3231 schedName"

::= {fdTriggerScheduleEntry 1}

fdTriggerScheduleDescription OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The human readable description of the purpose of this entry."

REFERENCE "RFC 3231 schedDescr"

::= {fdTriggerScheduleEntry 2}

fdTriggerScheduleWeekDay OBJECT-TYPE

```
SYNTAX BITS {
    reserved (0),
    monday (1),
    tuesday (2),
    wednesday (3),
    thursday (4),
    friday (5),
    saturday (6),
    sunday (7) }
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The set of weekdays on which the scheduled action shall be called if all other constraints are satisfied. Setting multiple bits will include several weekdays in the set of possible weekdays for this schedule. Setting all bits will allow the scheduler to perform the action on any weekday."

Note that implementations which maintain a list of pending activations must re-calculate them when this object is changed."

REFERENCE "RFC 3231 schedWeekDay"

::= {fdTriggerScheduleEntry 3}

fdTriggerScheduleMonth OBJECT-TYPE

```
SYNTAX      BITS {
              reserved (0),
              january (1),
              february (2),
              march (3),
              april (4),
              may (5),
              june (6),
              july (7),
              august (8),
              september (9),
              october (10),
              november (11),
              december (12) }
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The set of months during which the scheduled action shall be called if all other constraints are satisfied. Setting multiple bits will include several months in the set of possible months for this schedule. Setting all bits will allow the scheduler to perform the action in any month."

Note that implementations which maintain a list of pending activations must re-calculate them when this object is changed."

REFERENCE "RFC 3231 schedMonth"

::= {fdTriggerScheduleEntry 4}

fdTriggerScheduleDay OBJECT-TYPE

```
SYNTAX      BITS { reserved0 (0),
                    d1(1), d2(2), d3(3), d4(4), d5(5),
                    d6(6), d7(7), d8(8), d9(9), d10(10),
                    d11(11), d12(12), d13(13), d14(14), d15(15),
                    d16(16), d17(17), d18(18), d19(19), d20(20),
                    d21(21), d22(22), d23(23), d24(24), d25(25),
                    d26(26), d27(27), d28(28), d29(29), d30(30),
                    d31(31), reserved32(32),
                    r1(33), r2(34), r3(35), r4(36), r5(37),
                    r6(38), r7(39), r8(40), r9(41), r10(42),
                    r11(43), r12(44), r13(45), r14(46), r15(47),
                    r16(48), r17(49), r18(50), r19(51), r20(52),
                    r21(53), r22(54), r23(55), r24(56), r25(57),
                    r26(58), r27(59), r28(60), r29(61), r30(62),
                    r31(63) }
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The set of days in a month on which a scheduled action shall be called if all other constraints are satisfied. There are two sets of bits one can use to define the day within a month:

Enumerations starting with the letter 'd' indicate a day in a month relative to the first day of a month. The first day of the month is specified by setting the bit d1(1) and the thirty-first is specified by the bit d31(31).

Enumerations starting with the letter 'r' indicate a day in a month in reverse order, relative to the last day of a month. The last day in the month is specified by setting the bit r1(33) and the first day of a month with 31 days is specified by setting the bit r31(63).

Setting multiple bits will include several days in the set of possible days for this schedule. Setting all bits will allow the scheduler to perform the action on any day within a month. Setting all bits starting with the letter 'd' or the letter 'r' will also allow the scheduler to perform the action on any day within a month.

Note that implementations which maintain a list of pending activations must re-calculate them when this object is changed."

REFERENCE "RFC 3231 schedDay"

::= {fdTriggerScheduleEntry 5}