User Guide | PUBLIC
Document Version: 2206a – 2022-06-02

Rule-based IoT Data Processing

Managing activities and events in response to timeseries data

THE BEST RUN SAP

Content

1 Rules: Overview

Overview

Speaking of the Internet of Things (IoT) is closely related to Big Data, that is, the handling of very large amounts of data that is generated by devices that are equipped with all kinds of sensors. However, while the large amounts of data can be used to open new perspectives on things and to gain completely new insights, they can, at the same time, also be seen as a problem. This is because the sheer amount of data can make it difficult to find the needle in the haystack (for example, the one or two outliers in a huge pile of data that indicate a technical problem that has either already occurred or is likely to happen in the near future).

At this point, rules for processing time series data come in handy. With rules, you can leave it up to the system to permanently keep an eye on each and every sensor data point coming in. You start with a virtual thing that you want to monitor, pick the properties to be observed, and associate both with a rule. In the *Rule Projects* app, you define the values for each property and what shall happen if that particular value is observed in the stream of incoming data.

The facts explained so far are all related to a near-real time scenario, where sensor data is monitored and evaluated almost immediately after being recorded. However, there are also use cases where an immediate response is not needed, not possible (for example, due to a missing network connection), or where the amount of incoming data is so extreme that not each individual data point can be processed; the system then generates aggregated data instead. With SAP IoT, you can handle this kind of use cases as well by setting up scheduled rules (or "batch rules", as they are called when accessing them via the API).

> **i** Note
>
> Defining a rule-based system reaction to a particular value that has been observed is a two-stage process:
>
> - In the *Rule Projects* app, you set up the rules to be applied to incoming values. If a value matches the rule condition, the system triggers an event.
> - The event triggered by the rule is then used as input for the *Action Modeler* where you define what the system reaction to the observed value shall look like (for example, send an HTTP request, send a notification mail, or trigger another action).

With these preparations done, you can rest assured that none of the critical values that you might think of goes unnoticed, thus making sure that all necessary action can be taken without delay.

Data Protection and Privacy

Due to the enforcement of data protection and privacy, it is **not** allowed during rule definition to enter personal or sensitive data (for a definition, see Annotation) or information in any input field of rule projects or rules (for example, rule project description, rule name, rule description, or in the rule condition). If you do **not** adhere to

this guidance, the risk of violating applicable law such as the European General Data Protection Regulation (GDPR) may arise. Under no circumstances shall SAP be held liable for any such violations.

Rule modeling in SAP IoT makes internal use of SAP Business Rules. This means that creating and maintaining objects with SAP IoT also leads to the creation and modification of objects that are handled by SAP Business Rules. Depending on the nature of the data that you process in rules, questions of data protection and privacy may arise. For more information on how personal data may occur in rules and on how the underlying rule engine handles such cases, see Data Protection and Privacy.

Limitations

To ensure fluent rule processing as well as keeping the system load at a reasonable level, make sure that you don't exceed the following thresholds:

- Maximum number of draft rules: 1000
- Maximum number of active rules: 1000
- Maximum number of projects: 100
- Maximum number of custom data objects per project: 100

> **i** Note
>
> You can determine the number of rules per project by simply starting the *Rule Projects* app and opening the details screen for a rule project. The number of rules is displayed in the title area on top of the list of available rules. Note that the displayed number depends on the current filter settings. So, to find out the total number of rules, make sure that there is **no** filter defined.
>
> The given maximum numbers are in effect per tenant. Here, the idea is that - within the maximum number of draft rules - you're free to activate any number of draft rules from zero to maximum.
>
> If you have a use case in mind that can't be handled within the given limits, feel free to get in touch with SAP. We'll then investigate the use case and try to find a satisfying solution.

Rate Limits

In addition to the limits described above, which are maximum total numbers, some time-based limits are in effect. These time-based limits are aiming at limiting the number of newly created objects within a certain timespan. The purpose of this kind of limits is to prevent the system from being flooded with new objects, which is typically caused by poorly defined batch processes. The following limits apply:

Rate Limits

Action	Limit (# objects/minute)
Create a rule (all rule types)	30
Activate a rule	30
Deactivate a rule	30
Create a project	20
Update a project	20

> **i Note**
>
> All numbers given in the *Rate Limits* table above refer to the number of objects within each tenant.

Related Information

SAP Business Rules

1.1 Naming Conventions for Rule Processing

Names and strings that may not be used in names of objects used in rules.

Reserved Names for Data Objects and Thing Properties

In SAP IoT, there's a set of names or strings that you may **not** use in names for objects that are referenced by rules or rule contexts. See the following table for details.

> **i Note**
>
> These naming conventions are only valid for use in the context of rule processing. E.g., you are free to define a thing property named `SAP_RoomTemperature` in the Properties Catalog and use it for analyzing time series data with an application built in SAP Web IDE. However, if you try to use that property in a rule context, the system detects the name and raises an error message because property names starting with "SAP_" are **not** allowed in the context of rule processing.
>
> The reserved names are **not** case-sensitive. That is, a property name fails the validation, regardless whether the name is "`system`", "`System`", or "`SYSTEM`".

Reserved Names

Object Type	Reserved Name
Data Object	*_aggregate
Data Object	*_dimension
Data Object	*_dimensions
Data Object	*_previousValue
Data Object	SAP
Data Object	SAP_*
Data Object	System
Data Object	System_*

Object Type	Reserved Name
Data Object	Result_Stream
Data Object	Result_Batch
Property	SAP
Property	SAP_*
Property	System
Property	System_*
Property	IoT_Rule_ID
Property	IoT_Event_Name
Property	IoT_Event_Severity

Naming Conventions for Rules and Rule Projects

- For the names of rules and rule projects, the following naming conventions are in effect:
  Allowed characters: `[A..Z]`, `[a..z]`, `[0..9_]`. Leading or trailing whitespace is **not** allowed. However, you may use whitespace inside of a name.
- For the names of data objects referenced by rule projects, the same applies as for rules and rule projects. However, for data objects, whitespace is generally **not** allowed, regardless of its position within the name string.

> **i** Note
>
> The names of data objects are automatically derived from the name of the underlying property set or property set type. However, you can overwrite the derived name so that it fits better into the context of a particular use case for which you set up a rule project.

1.2 Referential Integrity for Objects Related to Rules

How the system ensures a consistent database status after changes to rule-related objects.

Introduction

Whenever an object is changed (that is, modified or deleted) that is used by a rule, the rule processing framework evaluates the consequences of that change for the affected rules. If necessary, the system automatically takes care of all the rule adjustments required to keep the database in a consistent state.

During rule modeling, quite a number of different types of objects are involved, such as rule projects, rules, property sets, thing types, and things. In addition, there's a hidden layer of even more object types that belong

to an internal rule processing engine that is used by SAP IoT. Modeling rules basically means putting together instances of all these different object types according to the use case that you have in mind. While you put together all the objects you're interested in, the system takes care of placing all the objects involved into a mesh of relationships behind the scenes.

However, when it comes to subsequent changes to objects referred to by a rule that has already been set up, chances are that a human expert wouldn't be able to keep track of all those interrelated (and - from a UI perspective - invisible) objects. Therefore, it's essential that the system adapts the underlying database objects on all layers exactly to the changes that a user performs on the surface (such as modifying or deleting rule-related objects).

Impacts of Changes in Detail

In the following table, we have listed in detail what it means for the system when you add, change, or delete objects related to a rule.

Impact of Changes to Rule-Related Objects

Object Type	Change	Impact
Property Set Type	Deletion	• All usages in all rule projects are deleted. • All usages in all rule inputs are deleted. • All usages of batch aggregates based on the property set type are deleted.
Reference Property Set Type	Deletion	• All usages in all rule projects are deleted. • All usages in all rule inputs are deleted.
Thing Type	Deletion	• All data objects using the thing type are deleted. • All rules using a rule context based on the thing type are deleted. • All explicit rule assignments to the thing type are removed.
Property Set	Removal from Thing Type	• All usages in all data objects are deleted. • All usages in all rule inputs are deleted. • All usages of batch aggregates based on the property set type are deleted. • All thresholds that belong to the property set are removed.

Object Type	Change	Impact
Property	Addition to Property Set Type	• Property is propagated to underlying rule engine. • Aggregates are created. • All changes take immediate effect in underlying rule engine.
Threshold Property	Addition	• Property is propagated to underlying rule engine. • All changes take immediate effect in underlying rule engine.
Property	Deletion from Property Set Type	• Property is removed from underlying rule engine and cannot be used anymore. • Aggregates are removed and cannot be used anymore. • All thresholds referring to the property are deleted.
Sensitivity Flag of Property Set Type	Change to `True`	• Sensitive property sets cannot be used in active rules. • All active rules referring to the property set type are deactivated.
Property Set Type of Property Set within Thing Type	Change	• Change is propagated to underlying rule engine. • Aggregates are updated, created, or deleted as required.
Property Reference	Change (e.g. different data type)	• All usages in all rule projects are deleted. • All usages in all rule inputs are deleted. • All usages of batch aggregates based on the property set type are deleted.

Related Information

Reference Property Set Type
Property Set Type
Thing Type
Data Protection and Privacy

1.3 Error Codes for Rule Execution

Overview of all types of errors that can occur during scheduled rule execution.

Error Codes

In this section, you can inform yourself about the different types of errors that can occur when the system processes scheduled rules. Do **not** confuse these error codes with errors that can occur during the execution of actions that a rule may trigger. For an exhaustive overview of errors related to action execution, see Action-Related Error Codes.

> **i** Note
>
> For those error codes in the following table where the suggested solution is creating a support ticket, choose `IOT-BSV-RLS-BAT` for the ticket component.

Error Codes for Rule Execution

Error Code	Category	Root Cause	Solution
`810000000`	Runtime Exception	Unexpected error occurred during scheduled rule execution. Contact SAP support for further information.	Create a support ticket.
`810000001`	Runtime Exception	Error making remote service call.	Create a support ticket.
`810000002`	Rule Modeling Exception	Error while parsing message for rule activation payload.	Create a support ticket.
`810000003`	Runtime Exception	Rule not found in runtime environment.	Make sure that the rule has been activated.
`810000004`	Rule Modeling Exception	Invalid rule condition. At least one aggregate property required.	Provide aggregate property in rule condition.
`810000005`	Rule Modeling Exception	Incorrect time frame format.	Create a support ticket.
`810000006`	Rule Modeling Exception	Incorrect windowing format.	Create a support ticket.
`810000007`	Runtime Exception	Rule result conversion error.	Create a support ticket.
`810000008`	Runtime Exception	Date conversion error.	
`810000009`	Runtime Exception	Failed to get rule engine instance.	Create a support ticket.
`810000010`	Request Rate Limiting	Execution suppressed by rate limiter. Request fell into 2-minutes time window between subsequent requests.	Adhere to 2-minutes interval for subsequent requests.

Error Code	Category	Root Cause	Solution
810000011	Runtime Exception	Rule executed successfully only for parts of the data, but not for all.	Create a support ticket (for details, see rule execution log).
810000012	Runtime Exception	Error occurred during rule execution.	Create a support ticket (for details, see rule execution log).
810000013	Runtime Exception	Rule execution details not found or not present.	Ensure that a valid execution ID is used in request for retrieving rule execution details. If the ID is correct but the problem still persists, create a support ticket.

Related Information

Action and Rule Execution Log: Overview

1.4 Derivation Rules

Derive values from measured sensor data.

Introduction

In many of the Internet of Things (IoT) scenarios, there's a need to derive new time series data based on certain calculations performed on the time series data sent by things. For example, you could model the electrical voltage and electrical current as *TimeSeriesData* and calculate electrical power as the *DerivedData*. For such scenarios, SAP IoT provides the *Rule Projects* application to configure derivation rules and derive new values based on computations performed on the time series data for a thing. The derived data for a thing is computed based on the rules configured for a specific property set type of data category *TimeSeriesData*.

> **i** Note
>
> For the first time activation of derivation rules, expect a delay ranging to a maximum of 1 business day for the processing of derived sensor data. After that, subsequent rule activations are promptly applied to the incoming data.

Data Objects

Unlike other rule types, it's mandatory for a derivation rule to provide 2 distinct data objects to be assigned as input and output, respectively:

- Input to derivation rules is the property set type of data category *TimeSeriesData* (called *Measured Values* in the **Thing Modeler** application).
- Output of derivation rules is the property set type of data category *DerivedData* (called application. When you activate the rule, the computation is performed only on the time series data that is ingested into the Time Series Store from the time after the rule is activated. You must note that the computation is *Calculated Values* in the **Thing Modeler** application).

General Procedure

You can define the derivation rules using the SAP IoT *Rule Projects***not** application. When you activate the rule, the computation is performed only on time series data that is ingested in the Time Series Store from the time after the rule is activated. That is, the computation is **not** performed on time series data that is already stored in Time Series Store before activating the rule. The derived data is then stored in the Time Series Store as defined in the rules. There could be a maximum time delay of 15 minutes between the time series data and its derived data availability. This delay exists to consider time series data that is arriving late in the Time Series Store.

SAP IoT currently supports the following types of derivation rules:

- Rules that don't have any dependency on previous data.
- Rules that are dependent on previous data.

For more information on this aspect, see Delayed Data Ingestion and Out-of-Order Data [page 12].

Setting up a derivation rule scenario basically consists of the following aspects:

- Thing Configuration [page 11]
- Derivation Rule Configuration [page 12]

Data Protection and Privacy

Due to the enforcement of data protection and privacy, it is **not** allowed during rule definition to enter personal or sensitive data. For more information, see SAP IoT Security Guide.

1.4.1 Thing Configuration

Configure thing data for derivation rules.

You need to configure the following:

- A property set type of data category *TimeSeriesData* to ingest and store time series data for a thing.
- A property set type of data category *DerivedData* to store the derived values in the time series store.
- A thing type to be associated with the previously mentioned property set types.
- A thing for which the time series data is computed to derive values.

1.4.2 Derivation Rule Configuration

Configure rules to compute derived values

You can configure a derivation rule using the SAP IoT *Rule Projects* application.

- In the *Data Objects* section of a rule project, you define a vocabulary with property sets of data category *TimeSeriesData* and *DerivedData*, which act as input and output data objects respectively for the rule definition. Ensure that you define only one property set of data category *TimeSeriesData* and only one property set of data category *DerivedData*.
- In the *Rules* section of a rule project, you select the data objects you defined for derivation. You define a new rule of type *Streaming Cloud - Derivation*, which facilitates the computation of derived values. You select the data object as input based on the property set of data category *TimeSeriesData* and a data object as output based on the property set of data category *DerivedData*.
    - In the *Rule Editor* section, you define the rule condition using the **If...Then** statement to compute the derived values. In the **If** condition field, define the condition using the properties of the input data object based on the property set of data category *TimeSeriesData*. In the **Then** condition field, define the condition for one or more properties of the output data object based on the property set of data category *DerivedData*. You can also use the **Else If** statement to define what happens if the first **If** condition results in a false value. To execute the rule on all incoming time series data without an **If** condition, you should enter **true** in the **If** condition field.

> **i** Note
>
> - You cannot use the same property set of data category *DerivedData* as output object in multiple active rules.
> - You can use the same property set of data category *TimeSeriesData* as input object in multiple active rules.
> - For properties of data type *Decimal* used as input and/or output, ensure that the derivation formula computes a result value that fits into the output data type. Otherwise, the derived value cannot be written to the time series store.

You can opt to compute the derivation values in one of the following ways:

- Computation to be performed on the current value of a property by choosing the property name.
- Computation to be performed on the previous value of a property by choosing the `<property_name>_PREVIOUS` object.

The rule editor is based on the SAP Business Rules service with Rule Expression Language 2.0. Accordingly, only the features supported by this service can be used for configuring derivation rules. For more information, see Expression Language 2.0.

1.4.3 Delayed Data Ingestion and Out-of-Order Data

Handling of data delays with derivation rules.

When you define derivation rules for the current values or for the previous values of a property, there are chances of computation errors if there is a delay in ingesting the time series data into the time series store or if the business timestamps of the time series data values are not in sequence (out-of-order data). A built-in functionality of derivations handles these errors.

Time Window for Delayed Ingestion and Out-of-Order Data

To handle the out-of-order data with a delayed ingestion, a time limit is defined. The delay in the data ingestion is determined by comparing the business timestamp of the time series data and the timestamp at which the data is ingested into the SAP IoT Time Series Store.

By default, the system supports a maximum time window (also called **delay-time-window**) of one hour between the business timestamp of the time series data and the timestamp at which the data is ingested into the SAP IoT Time Series Store. During this time window, the errors occurring due to out-of-order data with a delayed ingestion are automatically corrected. Any data that is delayed beyond **delay-time-window** is ignored by derivation rules that have a dependency on previous values. Derivation rules without a dependency on previous values still derive values seamlessly for the input time series data. This time window can be changed on request. The maximum time window for out-of-order data ingestion is 120 hours (five days). The time window setting is effective for an entire tenant. To change this time window, raise a support ticket assigned to component `IOT-BSV-BDI`.

SAP IoT also supports **Streaming Cloud - Event** rules to be defined on top of derived values. Due to out-of-order data, the derivation rule can derive multiple values for the same timestamp. With an active cloud streaming rule, this could result in more events triggered for the same timestamp. For more information about **Streaming Cloud - Event**, see Create a Rule (Thing Model) [page 23].

1.4.3.1 Delayed Data Ingestion

Handling of late coming time series data.

Delay in data ingestion is defined as the lag between the time at which a measurement is sensed (business timestamp) and the time at which it reaches the SAP IoT Time Series Store (processing timestamp). Possible reasons for such delay are poor network connectivity, sensor malfunction, issues with the thing or device transmitting sensed measurements, and so on. The computation of derived values on such delayed data is as follows:

- Derivation rules are executed without any restrictions for the time series data that are ingested into SAP IoT with a delay but only if the ingested data is in correct order.
- The delayed and the out-of-order messages are handled differently as described in the Out-of-Order Data [page 13].

> **i Note**
>
> If you have ingested future data, the computation of derived values will be blocked until the current time progresses to the future data ingested timestamp.

1.4.3.2 Out-of-Order Data

Handling of time series data received outside their natural sequence.

In the stream of incoming time series data for a thing, there are chances that at times the data may not be ingested in the order of business timestamp; such a stream of data is referred to as out-of-order data. Out-of-order data doesn't affect derivation rules that are not dependent on previous values. Every incoming data is

processed not just to derive values for the current data but also to determine the earlier computed values that need correction. Subsequently, any affected values are recomputed and persisted in Time Series Store. This ensures that the derived values retrieved are always accurately updated until that time. As a consequence, there could be multiple derived values for the same timestamp arrived at different points in time, based on the data available until those points. With an active cloud streaming rule, this could result in more events triggered for the same timestamp. The computation of derived values on such data is as follows:

- Derivation rule uses the previous value of the time series data for computation:
  - For out-of-order messages ingested with a delay less than 10 seconds:
    - Derived values are computed correctly.
    - Derived values are available within 15 minutes.
    - Derived values are ordered by business timestamp and then stored in the Time Series Store.
    - If **Streaming Cloud - Event** rules that create events are configured on the derived values, no multiple events are created.
  - For out-of-order messages with ingestion delay more than 10 seconds but less than the *delay-time-window* (by default, one hour):
    - Derived values are computed correctly.
    - Derived values are available within 15 minutes.
    - Derived values are ordered by business timestamp and then stored in the Time Series Store.
    - If **Streaming Cloud - Event** rules that create events are configured on the derived values, multiple events could be created.
  - For out-of-order messages with ingestion delay more than the *delay-time-window* (by default, one hour):
    - Derived values are **not** computed.
- Derivation rule does **not** use the previous value of the time series data for consumption:
  - For out-of-order messages ingested with a delay less than 10 seconds):
    - Out-of-order data has no consequence and the derived values are computed correctly.
    - Derived values are stored in the Time Series Store and are ordered by business timestamp.
    - If **Streaming Cloud - Event** rules that create events are configured on the derived values, no multiple events are created.
  - For out-of-order messages with ingestion delay more than 10 seconds but less than the *delay-time-window* (by default, one hour):
    - Out-of-order data is **not** considered for computing the derived values.
    - Derived values are stored in the Time Series Store and may not be ordered by business timestamp.
    - If rules that create events are configured on the derived values, multiple events could be created.

1.4.4 Read Derived Data

On activation of the derivation rule, it is applied on the time series data ingested from the time of activation and the derived data is stored in the Time Series Store. You can read the derived data using the time series data API. Aggregates calculation is also supported on derived data. For more information about retrieving time series data and derived data, see Thing:Time Series Data. The time series data APIs return derived values only for the period for which the corresponding rules were active.

2 Rule Projects: Overview

Introducing rule projects in SAP IoT.

Overview

In SAP Leonardo IoT, the predecessor product of SAP IoT, setting up rules for time series data processing required using two different apps:

- Rules: Definition of rule type, name, description, and other general data.
- Rule Contexts: Definition of the data streams to be monitored by a rule.

After having set up the required entities of both types, rules and rule contexts had to be associated with each other.

As opposed to this earlier approach, the Project Rules app integrates all the rule-related activities within one single app that covers all aspects of setting up a rule.

As rule projects can comprise numerous rules as well as many different data objects (based on property set types or property sets), they can serve as a natural container, which holds all the objects that are relevant and useful for solving a particular task related with rule processing. There's no predefined way with regards to organizing the objects that you set up. That is, you can think about different modeling approaches and choose the one that suits you best:

> ✤ Example
>
> - You associate a rule project with a group of thing types and collect all the rules in the project that shall monitor the sensor data sent by the things derived from the thing types.
> - You set up rule projects where each project is focused on a certain quality measured by the sensors, such as temperature, pressure, rotation speed, fill level, and so on.

Rule projects serving as containers for a set of rule-related entities provides another advantage: As the project has all the information about the object it contains, it provides you with some statistics about these objects, for example:

- Number of active rules, broken down by rule type
- Number of sensitive data objects, broken down by sensitivity level
- A list of rules that have been last changed.

> ℹ Note
>
> For a general overview of the rule concept in SAP IoT, see Rules: Overview [page 3].

Related Information

Rules: Overview [page 3]

2.1 Vocabulary

Data definition for rule processing.

Overview

In a rule project, rules are bundled together with their data source. This data source is referred to as the so-called vocabulary of a rule project.

While a rule project can contain any number of rules, there's only one vocabulary per rule project. However, the vocabulary itself can contain any number of data objects. These data objects, in turn, refer to property sets or property set types that have been set up during thing modeling or event modeling. You assign property sets and property set types to a data object by adding references to these entities. In addition to the user-defined property sets and property set types, each rule project offers a number of predefined data objects, which are dedicated for special purposes.

Data Models

For rule processing, SAP IoT supports the following data models:

- Thing Model: Rules monitoring data streams provided via properties defined for virtual things. You set up these properties either in the *Properties Catalog* app or via the corresponding APIs. In the thing model domain, all types of rules are supported.
- Event Model: Rules listening to custom events that you have set up via the corresponding APIs. Like things, events can have associated properties that a rule evaluates when the specified event is detected. In the event model domain, only rules of type *Streaming Rule - Event Creation* are supported.

> **i** Note
>
> As opposed to properties derived from the thing model, there is no app for end users to model events. Event modeling is only possible by programming against the corresponding APIs of the Event service.

It's worth mentioning that thing model and event model aren't mutually exclusive. Instead, it's perfectly fine to set up a rule project using properties from both data models simultaneously.

Data Objects

The following table gives an overview of the different kinds of data objects that can be part of a rule vocabulary:

Data Objects in Rule Vocabulary

Type	Origin	Description
Property Set	Thing modeling - thing type	A set of properties that has been defined in the context of a thing type. Can be used with things derived from that thing type.
	Event modeling - event type	A set of properties that has been defined in the context of an event type. Can be used with events derived from that event type.
Property Set Type	Thing modeling - stand-alone data structure	A set of properties that has been defined as a stand-alone data structure. Can be used with all thing types.
Predefined	Rule project	System-provided data objects for specific rule types or purposes. For a detailed list, see the *Predefined Data Objects* section below.

You can define attributes such as unit of measure and thresholds for a property using the configuration service. In this way, all things and all data records of the property have the same unit of measure and threshold value.

In addition, SAP IoT allows you to use the data category `ReferencePropertyData` to define reference properties with different predefined attribute types. With that, all data records of a property for a thing instance can have different units of measure and thresholds at different points in time, as defined.

Predefined Data Objects

In addition to data objects based on user-defined property sets and property set types, each rule project provides a set of predefined data objects:

Predefined Data Objects

Name	Purpose
Result_Batch	Default data object used for the output of scheduled rules.
Result Stream	Default data object used for the output of streaming rules creating an event.
SAP_GeofenceData	Data object for holding geofence information used in geofence rules.
SAP_Stream_Derivation	Data object holding additional information that can be used within streaming derivation rules.
SAP_Stream_Event	Data object holding additional information that can be used within streaming rules creating events.

In the *Data Objects* section of the *Rule Projects* app, you can click on any of the assigned data objects (including the predefined data objects) to have the system display the *Data Object Details* popup, which presents all the detail information available for a data object, such as:

- All property fields
- Rule types supporting the use of the data object
- Navigation links pointing to the associated source object like property set type or thing type

Related Information

Thing Properties Catalog: Overview
Event Services

2.2 Rule Modeling: Overview

General information on defining rules

Introduction

Collecting time series data in an IoT scenario isn't a purpose in itself. Rather, in most cases, you want to monitor and analyze the incoming data. Even more important is to watch out for all kinds of irregularities (for example, an unwanted drift of sensor data, or a sudden increase of outliers) indicating that something is going wrong in the area where the sensors are installed. In such cases, it's important to keep the responsible experts informed as fast as possible so that they can react in an appropriate way.

With rules for IoT time series data, you can define a widely automated process to handle such critical situations. You decide which sensor data you want to monitor, which value ranges reflect a normal situation, and which not, and you define what the system is supposed to do in case the values indicate an error situation.

Each rule contains a condition that is tested by the rule and an event that the rule triggers if the condition is fulfilled. Once you've specified a rule, you can define an action in the Action Modeler that the system performs when the rule condition is fulfilled.

Rule Status

Rules can be active or inactive (actually, the status concept has a finer granularity, as shown in the following table for the sake of accuracy). Only active rules are executed by the system. Leaving rules in inactive status is a way for you to prepare either comprehensive changes to one rule or changes to a bigger set of rules that you want to make effective at a later point in time. The system supports you in handling bigger sets of rules by offering mass activation, or deactivation, of rules. You can accomplish this by selecting all relevant rules in the list offered on the entry screen of the app and choose *Activate*, or *Deactivate*.

In the following table, you find a list of all the different status values that can apply for a rule:

Rule Status Values

Status	Description
Inactive	Inactive. This is the initial status for every rule, indicating that there has never been any attempt to activate it.
Activation Triggered	Activation message has been sent to the respective runtime component waiting for confirmation.
Active	Activation has been confirmed by the respective runtime component.
Deactivation Triggered	Deactivation message has been sent to the respective runtime component waiting for confirmation.
Deactivated	Deactivation has been confirmed by the respective runtime component.

Rule Types

In SAP IoT, you can use the following types of rules:

- Streaming - Event Creation
- Streaming - Derivation
- Scheduled

You specify the type of rules upon rule creation. Once a new rule has been saved, you can't change its type anymore.

Streaming Rules

You use streaming rules for monitoring the measured values that are sent from the sensors of a device in real time, or near real time, respectively. For the rules themselves, you can define expressions for describing conditions that the rule shall check. For example, a rule might check if a certain peak temperature is exceeded. For these expressions, a small set of comparison operators is available that helps you describe the condition check.

When working with streaming rules, a problem can arise when rule execution leads to a high number of events triggered by the rule (for example, by sending dozens of more or less identical emails to an administrator) where one single event would already be sufficient to raise the attention of a person in charge. To tackle this challenge, you can decide to run a rule in a special "don't panic" execution mode, such as sleep mode or toggle mode. For more information, see Reducing Event Creation [page 28].

Streaming Rules - Event Creation versus Derivation

When you start setting up a streaming rule, the very first thing that you must decide is about the subtype of the new rule, either event creation or derivation. The difference is the following:

- Event Creation: As the name suggests, rules of this subtype are monitoring the sensor data registered by the input data object of the rule. After checking the values against the rule conditions, the rule either raises an event for further processing or not.
- Derivation: This rule subtype requires an additional data object for output data. After checking the values against the rule conditions, the rule can assign certain values to the properties that are available in the output data object. For example, this can be used for data cleansing by ignoring outlier data, or for any kind of transformation or recalculation of the incoming data.

Scheduled Rules

Very soon after new measured values have been ingested (and maybe already processed by a streaming rule), the system stores the values in a dedicated storage section of the underlying database infrastructure. Once the values have been stored, you can access and analyze them at any later point in time. Also, while streaming rules take each single value into account, scheduled rules operate on aggregated chunks of values. For this kind of a deferred analysis of aggregated values, you use scheduled rules.

A scheduled rule is very similar to a streaming rule. However, for scheduled rules, you have to make additional settings:

- Windowing: You define the time frame for the values to be analyzed. Only values with a timestamp within the specified time frame are taken into account for the analysis.
- Scheduling: You specify whether a rule shall be executed only once at a particular point in time or if it shall be executed on a regular basis.

Another difference between streaming rules and scheduled rules is that for scheduled rules, the system supports a set of aggregate functions that you can use to analyze the values in the specified sample.

> ✤ Example
>
> If the average temperature of a cooler aggregate in the past two hours is higher than ten centigrade, send an alert to the person responsible for that aggregate.

Related Information

Reducing Event Creation [page 28]

2.3 Create a Rule Project

Process steps for creating a rule project.

Context

You define a rule project based on one or more data objects, which refer to property sets or property set types. Property set types are self-contained, referable data entities, while property sets are directly associated with a thing type. The data objects used by the rule project are sometimes referred to as the rule project vocabulary. After defining the data objects, you define the rules that use the different properties contained therein. Depending on the settings, these rules are then executed at runtime depending on the values recorded for a particular property.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. Choose *Create*.

   The system presents the *New Project* dialog box.
3. In the dialog box, enter the general data for the new project, such as the *Name* and optionally an additional *Short Text* as well as a *Description*.
4. Choose *Create*.

   The system navigates you to the *Data Objects* tab.
5. Set up the data objects by specifying the data to be processed by the rules in your rule project. For a detailed procedure, see Create a Vocabulary [page 22].
6. Once you're done with setting up the data objects, choose the *Rules* tab.
7. Set up the rules that are required in your rule project. For a detailed procedure, see Create a Rule (Thing Model) [page 23].
8. Once you're done with setting up both the data objects and the rules, choose *Save*.

Results

The rule project is now ready for use. Note that you have to activate each rule before it can start monitoring incoming time series data.

Related Information

Create a Vocabulary [page 22]
Create a Rule (Thing Model) [page 23]
Create a Rule (Event Model) [page 38]

2.3.1 Create a Vocabulary

Process steps for specifying the data objects that are relevant for a rule project.

Context

In a rule project, you want to specify the data objects representing the time series data to be monitored by the rules in the project. Depending on your usage scenario, you may have to include different types of data objects in your project:

- Property Sets vs. Property Set Types: You can use both kinds of data objects. A property set is always bound to a particular thing type, while property set types exist independent of a thing type and can be used by all things within the scope of a package.
- For all rule types, you need property sets or property set types containing properties for *Measured Values*. These properties can be used with the input data object of a rule.
- For derivation rules, you need property sets or property set types containing properties for *Calculated Values*.
- Sensitive data: Property sets or property set types containing properties that have been flagged as sensitive **cannot** be used for rule processing. If a rule project contains sensitive data, the Rule Project app makes you aware of this in the header area of the app.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. Select the rule project for which you want to create a vocabulary.

   The system navigates you to the *General Information* screen of the selected rule project.
3. Choose *Data Objects*.
4. To add a property set or a property set type to the vocabulary, choose Add or click the "+" (plus) symbol on one of the empty cards.

   The system navigates you to the *Select Property Sets*/*Select Property Set Types* screen.
5. Depending on which type of object you want to add, proceed as follows:
   - **Property Set**: Choose a *Package* and a *Thing Type* to populate the list of available *Property Sets*.
   - **Property Set Type**: Choose a *Package* to populate the list of available *Property Set Types*.
6. In the lists of available property sets or property set types, tick the checkbox of the elements you want to use in your rule project and choose *OK*.

   > **i** Note
   >
   > In case of a property set (**not** for a property set **type**), the system presents an additional dialog box where you can choose from the available threshold values, which of these values shall be included in the vocabulary. By default, all available threshold values are selected.
   >
   > Note that threshold values are only supported for rules of type *Streaming - Event*.

> The dot sign (".") is **not** allowed in threshold value names.

The system navigates you back to the *Data Objects* screen, where the previously selected property sets or property set types have been added by the system.

7. Choose *Save*.

Results

All the properties that are available in the selected property sets and property set types are now part of the rule project and can be used by the rules defined in the project.

If you have decided to include threshold values in the vocabulary, you can find the corresponding objects listed in the *Related* section of the *Data Object Details* screen.

2.3.2 Create a Rule (Thing Model)

Process steps for specifying the rules (based on properties of things or thing types) that are bundled within a rule project.

Context

In a rule project, you want to set up the rules for calculating values based on time series data or for monitoring time series data, based on the data objects that have been specified for the project.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. Select the rule project for which you want to create a rule.

   The system navigates you to the *General Information* screen of the selected rule project.
3. Choose *Rules*.
4. From the *Create Rule* dropdown list, choose the desired rule type (*Scheduled*, *Streaming → Event Creation*, or *Streaming → Derivation* [early adopters only], or *Streaming Hybrid*).

   > **i Note**
   >
   > For an explanation of the different rule types, see Rule Modeling: Overview [page 18].

   The system presents an empty detail screen where you can enter the settings for the new rule on the *General Information* section.

5. Enter a *Name* for the rule. Optionally, you can also enter a *Short Text* and a *Description*. These three fields serve the following different purposes:

Field	Purpose
Name	Mandatory, **not** translatable. A name indicating the purpose of the rule. You may use this field for entering some abbreviated or coded information that is common to your company (for example, the ID of a building or a department). This field is also used as sorting criterion in the value helps of the Rules Modeler, so it's advisable to follow a systematic naming convention.
Short Text	Optional, translatable. A description in natural language indicating the purpose of the rule.
Description	Optional, translatable. Use this field for a few sentences that help other users understand what the rule contains, where it is used, and so on.

6. Optionally, you may define a set of freely defined *Tags* that you can attach to a rule. You can use tags as filter criteria for the list of rules displayed on the *Rules* screen of the app.
7. In the *Input Data Object* field, click the value help button.
8. From the list of available property sets or property set types, choose the one you want to use as input data object for the rule.

> **i Note**
>
> Depending on the rule type, the objects offered for selection may vary. For example, for streaming rules (event creation), you can select the event to be triggered.

   Depending on the settings of the properties, the system automatically derives the *Sensitivity Level* of the data object. This is relevant for ensuring compliance with data protection and privacy laws.

   You may choose *Details* to see a list of all the properties that belong to the selected property set or property set type.
9. Choose *Confirm* to transfer the data-related information to the rule editor. This is required for you to continue setting up the rule.
10. In the *Rule Editor* section, define the *If* condition that the rule shall test.

    The system expects the If condition in a simple, predefined syntax, which slightly differs depending on the rule type:

    - Streaming rule:
      `<property name> of <property set name> <operator> <value>`

      > **Example**
      >
      > `WaterTemperature of Engine5 is greater than 95`

    - Scheduled rule:
      `<property name>_<aggregate_function> of <property set name>_aggregate <operator> <value>`

      > **Example**
      >
      > `WaterTemperature_AVG of Engine5_aggregate is greater than 95`

The following aggregate functions can be used for analyzing time series data with scheduled rules:

Aggregate Functions for Scheduled Rules

Function	Description
COUNT	Returns the number of records in the sample. Supported for all data types.
FIRST LAST	Returns the first or last record in the sample. Supported for all data types.
TFIRST TLAST	Returns the timestamp of the first or last record in the sample. Supported for all data types.
MIN MAX	Returns the lowest or highest measured value. Supported for numeric data types.
TMIN TMAX	Returns the timestamp of the lowest or highest measured value in the sample. Supported for numeric data types.
SUM AVG STDDEV	Returns the sum, average, or standard deviation of the measured values in the sample. Supported for numeric data types.

> **i Note**
>
> We don't recommend typing into the entry fields of the rule editor. Instead, the system offers you a list of all field names and operators that can be used for setting up the rule definition. To expand the list of possible entries, type a space character. Then, pick the desired object with the mouse or the cursor keys.

> **i Note**
>
> Rule modeling in SAP IoT supports many, but **not all** of the functions and operators that are available in the underlying rule engine. For a complete list of the supported elements, see Supported Functions and Operators [page 36].

The following steps depend on the rule type:

Rule Type-Specific Settings

Rule Type	Steps
Scheduled Rule	1. In the *Settings* section, Specify the desired *Execution Mode*. 2. In the *Windowing* section, define a time window for aggregated time series data to be monitored by the rule. To accomplish this, specify a *Time Base* (for example, days, hours, or minutes) and a numeric *Value* that specifies the timespan backwards from the moment of rule execution. 3. In the *Scheduling* section, specify either the *Recurrence* pattern for a rule that shall be executed on a regular basis, or the *Start Time* and *End Time* for a rule that shall be executed only once. **i Note** ○ The specified time window must lie within a range from 2 minutes through 30 days. ○ You can't define a combination of different *Time Base* values, for example, "2 days 8 hours". Instead, adjust the numeric value in combination with one single time base, for example, "56 hours". ○ If you choose **`Minute`** as the *Time Base*, the *Value* must be a multiple of 2.

Rule Type	Steps
Streaming - Event	1. In the *Settings* section, choose the desired *Execution Mode* for the rule. You have the following options: ○ *Default*: For each monitored value that fulfills the rule condition, the rule triggers an event. ○ *Sleeping*: The rule stops triggering events for a specified duration even if the rule condition is fulfilled. ○ *Toggle*: The rule only triggers an event if the rule condition is fulfilled **and** the condition was **not** fulfilled for previously tested values. ○ *Duration*: Specify a period of time, between 10 seconds and 1 hour, during which no event is triggered even though the rule condition is met. Only if the rule condition is constantly met for a time period longer than defined an event is triggered. ○ *Consecutive condition fulfillment*: Specify a number of met rule conditions higher than 1 for which no event is triggered even though the rule condition is met. Only if the rule condition is met in a gapless sequence more often than the specified number, an event is triggered. All these modes are useful to avoid event flooding in certain scenarios. For more information, see Reducing Event Creation [page 28]. **i Note** The execution modes *Sleeping* and *Duration* are very similar at first sight. Here's the difference: ○ *Sleeping*: **After** a successful rule execution, the rule triggers no events for the specified time period. ○ *Duration*: The rule ignores a continuously fulfilled rule condition and triggers no events **until** the specified time period is over. Any sensor data value that doesn't fulfill the condition resets the time period to the specified duration. 2. Depending on the chosen execution mode, proceed as follows: ○ *Default*: No further settings required. ○ *Sleeping*: Specify the desired *Sleeping Period*. This value must lie within a range from 10 seconds to 1 hour. ○ *Toggle*: No further settings required. ○ *Duration*: For the *Event Creation Mode*, choose *First Match Only* to trigger no more than exactly one event after the specified duration, regardless how often the rule condition is met. ○ *Consecutive condition fulfillment*: For the *Event Creation Mode*, choose *First Match Only* to trigger no more than exactly one event after the specified number is exceeded, regardless how often the rule condition is met. **i Note** If the rule project vocabulary contains threshold values, these values are automatically added to the list of valid input data objects offered by the rule editor.

Rule Type	Steps
Streaming Hybrid	*Streaming Hybrid* rules can be activated for use both in the cloud as well as at the edge (that is, for immediate processing in the direct local surrounding of the physical device, without the need for transferring data to the cloud first). However, in edge mode, fewer features are supported. This limited feature set is also in effect for hybrid rules used in the cloud as well (in the sense of a least common denominator). Here is the list of features supported by hybrid rules: ○ Rule context must refer to a property set that contains time series data. ○ The rule condition can handle only one property, which must be numeric. ○ The rule condition supports the following operators: `greater than`, `less than`, `equals`. ○ Execution modes: Default and sleep mode (that is, toggle mode is **not** supported).

11. Choose *Save*.

Results

The new rule is ready for use. However, for the rule to take effect, you must *Activate* it first.

2.3.2.1 Reducing Event Creation

Various different approaches for reducing the number of alerts during rule execution

Overview

Processing data in the Internet of Things means almost always processing mass data. Depending on the use case, device sensors may collect and send huge amounts of measured values, which can impose a significant load on the software layers that are in charge of processing the incoming data. Moreover, there are many use cases where it is important and required to collect large amounts of measured values, but, at the same time, it is perfectly fine **not** to react on each single value.

> ✤ Example
>
> In a building automation scenario, a number of sensors have been installed that measure the current wind speed. In addition, there are sensors used to detect whether a window is closed or not. A rule has been set up to alert a facility manager in case a certain wind speed is exceeded and there are still windows open that need to be shut manually.
>
> Assuming the wind speed threshold has been set to 50 km/h and the wind exceeds that threshold at 10:10 a.m.. The wind speed may even rise for the next 5 hours up to a maximum of 90 km/h, before the weather calms down and the wind speed decreases to 40 km/h at 4:00 p.m.. With the rule described above, this would mean that the system is flooding the facility manager with alert messages during a period of more than six hours (that is, almost the entire workday), or at least up to the moment when all windows have been shut. However, for drawing the facility manager's attention to the potential threat caused by the

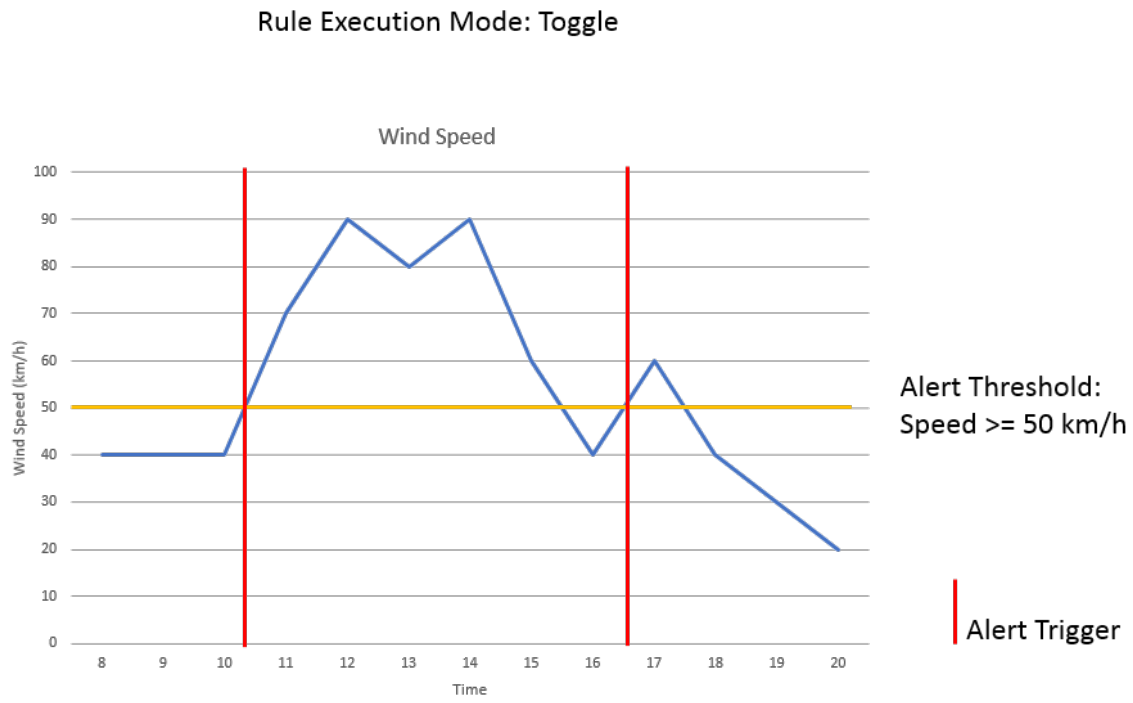
combination of increasing wind speed and open windows, one single alert (instead of hundreds or thousands) would already be sufficient.

With the rule modeling feature of SAP IoT, you can take advantage of the following approaches that help you reduce the number of alerts caused by a scenario as described in this example. You choose the desired approach by selecting one of the following rule execution modes:

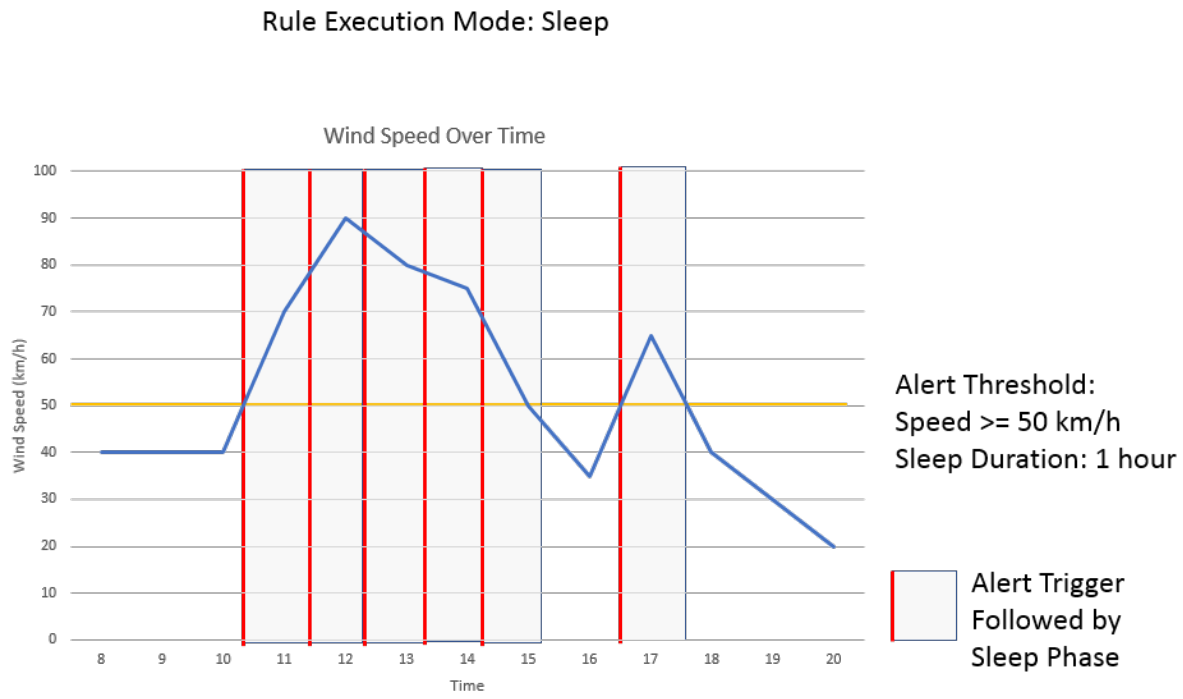
- Sleep Mode: A mode of rule execution in IoT scenarios where a rule stops triggering events for a specified duration even if the rule condition is fulfilled. This execution mode helps to prevent event flooding in case a specified threshold value is constantly exceeded but does not require an action for each individual occurrence.
- Toggle Mode: A mode of rule execution in IoT scenarios where a rule only triggers an event if the rule condition is fulfilled and the condition was **not** fulfilled for previously tested values. This execution mode helps to prevent event flooding in case a specified threshold value is constantly exceeded.
- Duration Mode: A mode of rule execution in IoT scenarios where a rule triggers an event only if the condition is continuously fulfilled over a certain period of time.
- Consecutive Condition Fulfillment Mode: A mode of rule execution in IoT scenarios where a rule triggers an event only if the condition is fulfilled multiple times in a row (where it's up to the user to specify the number of repeated condition fulfillments).

Toggle Mode and Sleep Mode

For the "high wind speed/open windows" scenario described in the example, toggle mode would be the desired setting: With that, the rule that is monitoring the sensor data would trigger exactly one alert after the wind speed exceeds the threshold value of 50 km/h and then remain silent for the entire rest of the day. Only if the wind speed would rise again after having calmed down to less than 50 km/h, the rule might send a second alert. The following diagram illustrates the effect:



In contrast to toggle mode, see the effect of the same rule in sleep mode with the longest possible sleep phase of one hour after the rule condition was fulfilled:



As you can see, running the rule in toggle mode results in only two alerts, whereas in sleep mode, the rule triggers at least six alerts for the same set of time series data. However, even six alerts shouldn't be too annoying for the person in charge compared with approximately 400 alerts that the system would send based on a really moderate scan rate of just one measurement per minute.

Duration Mode and Consecutive Condition Fulfillment Mode

The common element of these two execution modes is that the first occurrence of a fulfilled rule condition is ignored and doesn't trigger an event. After that first occurrence, the rule would either wait for the specified time or count further occurrences. The following applies:

- **Duration Mode**: As soon as the rule condition is fulfilled, a timer is started. The initial fulfillment of a condition in a rule with duration mode does **not** trigger an event. Further condition fulfillments are ignored for the specified duration. With the next fulfillment of the condition, the timer is evaluated. If the duration exceeds the event suspension, an event is raised. If the condition is **not** fulfilled by the next incoming data, the timer is reset.

  > ### Example
  >
  > The door of a cold storage room is equipped with a sensor that determines whether the door is closed properly. Of course, it's allowed to open the door and enter the room, but the door must be closed as soon as possible to avoid critical warming of the goods stored inside. The door sensor data is monitored with a rule in execution mode *Duration*, with a duration specified as 1 minute. With this setting, authorized persons can enter the room without causing alerts as long as they manage to close the door within one minute.

> **Note**
>
> For rules in duration mode, the following applies:
>
> - The duration of event suspension must lie within a range from 20 seconds through 1 hour.
> - Persistence period: 36 hours.

- **Consecutive Condition Fulfillment Mode**: As soon as the rule condition is fulfilled a counter starts. The initial fulfillment of a condition in a rule with consecutive condition fulfillment mode does **not** trigger an event. With the next fulfillment of the condition the counter is evaluated. If the counter exceeds the event suspension count an event is raised. If the condition is **not** fulfilled by the next incoming data, the counter is reset.

  > **Example**
  >
  > In a certain section of a plant area, only 10 vehicles may be present simultaneously. Access to that section is monitored with a sensor that sends a data package for each entering vehicle. The sensor data is tracked by a rule in consecutive condition fulfillment mode with an event suspension count value of 10. As soon as vehicle #10 has entered the restricted section, the rule triggers an event, which could be used for closing the entrance gate of the restricted section.

  > **Note**
  >
  > The minimum event suspension count is 1.

Time stamps and Processing Sequence

Each single measured value comes along with an exact timestamp that indicates when the respective sensor has recorded that value.

> **Note**
>
> For the event reduction modes, the following applies:
>
> - Sleep mode as well as toggle mode are supported for streaming rules, that is, for rules dealing with discrete, individual measured values. For scheduled rules dealing with aggregated data, only toggle mode is supported.
> - Sleep mode and toggle mode are mutually exclusive.
> - Due to delays and data point sequence reordering caused by the involved software layers as well as hardware behavior, SAP cannot guarantee that the ingested data is always processed in exactly the same order as the values have been originally measured by the sensors. This can lead to unexpected effects during rule processing. For further information, see the processing examples given for the different rule modes in the *Details* section.

Details

Processing Time versus Business Time

For rule processing, the system makes a differentiation between different time horizons, namely processing time and business time. These terms have the following meaning:

- Business time: The time at which a particular value has been measured by a device sensor. Each sensor takes care of providing a time stamp for each recorded measurement, which is a prerequisite for any subsequent time series analysis.
- Processing time: The time at which a rule is processing the sensor data. In SAP IoT, rule processing of sensor data is never a real-time process due to various reasons (for example, delays in the underlying ingestion pipeline, communication channel latencies, longer sensor synchronization intervals for power saving, etc.). The minimum delay that you can expect on average is approximately ten seconds. For example, a temperature that has been measured at 09:10:25 can be processed by a rule at 09:10:35 or later.

> **i** Note

Persistence Period and Handling of Latecomer Events

For streaming rules, a special mechanism is in place called "persistence period": This feature serves the following purpose: For rule processing in sleep mode as well as in toggle mode, the processing time at which the rule is executed takes precedence over the business time at which a particular value has been measured. In case of potential delays during the data ingestion process, this can lead to unexpected results. For details, see the examples for the two different execution modes.

For many use cases, it may be sufficient to pass on the recorded time series data to the rule processing unit in bigger chunks (for example, once per day), rather than continuously listening to the current stream of data. For each measured value that fulfills the rule condition, the system keeps track of the end of the sleeping time according to the rule settings (for example, measurement timestamp plus 45 minutes). This information on the expiry of the sleep time is stored in the system for the length of the persistence period. The persistence period is a hard-coded value of 36 hours. The value of 36 hours has been chosen to ensure that it's safe for customers to follow a 24 hours cycle for rule processing without running into situations where measurements with a timestamp near the edge of the 24 hours range could be lost.

Also, as already mentioned, it isn't unusual for an IoT scenario that the original data sequence is disturbed or delayed for various reasons that aren't always predictable. As a consequence, it could happen that data reaches the ingestion pipeline with massive delays so that it wouldn't make much sense for a rule to trigger any alerts based on the delayed data. To avoid such useless rule processing on outdated data, the system dismisses any data if the measurement time stamps are older than the predefined persistence period for data.

As an intentional and desired side effect, dismissing data older than the persistence period also serves the purpose of an implicit cleanup of the data ingestion queue. Otherwise, it could happen that over time, chunks of delayed data mess up the ingestion queue and impose unnecessary load on the storage system.

> **i** Note
>
> The fixed duration of the persistence period (currently 36 hours) may be subject to change in future releases.

Sleep Mode

A rule in sleep mode behaves in the following way:

As soon as the rule condition is fulfilled, the rule triggers an event as any other rule would do. However, after triggering the event, the rule stops responding to any further occurrences of fulfilled rule conditions for a specified time frame. That is, although the rule is still active, it ceases triggering events based on its settings. Once the specified time is over, the rule waits for the next fulfillment of its condition, triggers the event, and falls asleep again, and so on.

For rules in sleep mode, the following limitations apply:

- Sleep time: minimum 10 seconds, maximum one hour
- Persistence period: 36 hours

In the following table, you can find an example of how the system responds to measured values that the rule processes in a sequence that deviates from the measurement time stamps:

Rule Processing in Sleep Mode (sleep time: 1 minute)

Processing Time	Business Time	Condition fulfilled	Event Triggered
Mo 14:51:00	Mo 14:50:30	True	True
Mo 14:51:20	Mo 14:50:50	True	False (sleeping)
Mo 14:51:50	Mo 14:51:40	True	True
Mo 14:51:55	**Mo 14:49:40**	True	False (data with newer business time 14:51:40 already processed)
Th 15:00:00	Mo 14:52:30	True	True (would be false if processing time would be before Wednesday, 02:51:40)

> **i** Note
>
> Time stamps deviating from the normal sequence are given in **emphasized** typeface.

Toggle Mode

A rule in toggle mode behaves in the following way:

As soon as the rule condition is fulfilled, the rule triggers an event as any other rule would do. However, after triggering the event, the rule stops responding to any further occurrences of values that fulfill the condition unless a value has been determined that does **not** fulfill the condition. After that, the next value that fulfills the condition triggers the event, and so on.

In the following table, you can find an example of how the system responds to measured values that the rule processes in a sequence that deviates from the measurement time stamps:

Rule Processing in Toggle Mode

Processing Time	Business Time	Condition fulfilled	Event Triggered	Comment
Mo 14:51:00	Mo 14:50:30	True	True	

Processing Time	Business Time	Condition fulfilled	Event Triggered	Comment
Mo 14:51:20	Mo 14:50:50	True	False	No event creation for subsequent values matching condition
Mo 14:51:50	Mo 14:51:40	True	False	No event creation for subsequent values matching condition
Mo 14:51:55	**Mo 14:49:40**	False	False	Value does not match condition. However, toggle is **not** set back to initial state because business time (Mo 14:49:40) is earlier than previous ingestion (Mo 14:51:40)
Mo 14:52:50	Mo 14:52:40	False	False	Toggle is set back to initial state because business time (Mo 14:52:40) is later than previous ingestion (Mo 14:51:40)
Mo 14:52:55	Mo 14:52.42	False	False	
Mo 14:53:56	**Mo 14:52.41**	True	True	Event created because business timestamp is not considered
Th 15:00:00	Mo 14:58.30	True	True	would be false if processing time would be before We 02:53:56

> **i** Note
>
> Time stamps deviating from the normal sequence are given in **emphasized** typeface.

Duration Mode

A rule in duration mode behaves in the following way:

As soon as the rule condition is fulfilled, a timer is started. The initial fulfillment of a condition in a rule with duration mode does **not** trigger an event. Further condition fulfillments are ignored for the specified duration. With the next fulfillment of the condition, the timer is evaluated. If the duration exceeds the event suspension, an event is raised. If the condition is **not** fulfilled by the next incoming data, the timer is reset.

> Example
>
> The door of a cold storage room is equipped with a sensor that determines whether the door is closed properly. Of course, it's allowed to open the door and enter the room, but the door must be closed as soon as possible to avoid critical warming of the goods stored inside. The door sensor data is monitored with a rule in execution mode *Duration*, with a duration specified as 1 minute. With this setting, authorized persons can enter the room without causing alerts as long as they manage to close the door within one minute.

> **i Note**
>
> For rules in duration mode, the following applies:
>
> - The duration of event suspension must lie within a range from 20 seconds through 1 hour.
> - Persistence period: 36 hours.

In the following table, you can find an example of how the system responds to measured values that the rule processes with an event suspension time of 60 seconds:

Rule Processing in Duration Mode

Business Time	Condition fulfilled	Duration since first fulfilled condition	Event Triggered	Comment
Mo 14:50:30	True	0	False	No event creation; timer starts.
Mo 14:50:50	True	20	False	No event creation; duration not exceeded.
Mo 14:51:40	True	70	True	Event created for match mode ONCE and EACH_TIME; duration exceeded.
Mo 14:51:50	True	80	Match Mode `ONCE`: False Match Mode `EACH_TIME`: True	Event created for match mode `EACH_TIME`; duration exceeded.
Mo 14:52.42	False	0	False	Duration timer is reset.

Consecutive Condition Fulfillment Mode

A rule in consecutive condition fulfillment mode behaves in the following way:

Consecutive Condition Fulfillment Mode: As soon as the rule condition is fulfilled a counter starts. The initial fulfillment of a condition in a rule with consecutive condition fulfillment mode does **not** trigger an event. With the next fulfillment of the condition the counter is evaluated. If the counter exceeds the event suspension count an event is raised. If the condition is **not** fulfilled by the next incoming data, the counter is reset.

> **Example**
>
> In a certain section of a plant area, only 10 vehicles may be present simultaneously. Access to that section is monitored with a sensor that sends a data package for each entering vehicle. The sensor data is tracked by a rule in consecutive condition fulfillment mode with an event suspension count value of 10. As soon as vehicle #10 has entered the restricted section, the rule triggers an event, which could be used for closing the entrance gate of the restricted section.

> **i Note**
>
> The minimum event suspension count is 1.

In the following table, you can find an example of how the system responds to measured values that the rule processes with an event suspension count of 1:

Rule Processing in Consecutive Condition Fulfillment Mode

Business Time	Condition fulfilled	Number of consecutively fulfilled conditions	Event Triggered	Comment
Mo 14:50:30	True	1	False	No event creation; event suspended once.
Mo 14:50:50	True	2	True	Event created for match modes `ONCE` and `EACH_TIME`; event suspension count is exceeded.
Mo 14:51:40	True	3	Match Mode `ONCE`: False Match Mode `EACH_TIME`: True	Event created for match mode `EACH_TIME`.
Mo 14:51:50	False	0	False	Sequence counter is reset.

Related Information

Rule Modeling: Overview [page 18]

2.3.2.2 Supported Functions and Operators

Elements for defining rule conditions.

Overview

Rule processing with SAP IoT is internally relying on the rules engine provided by SAP Business Rules. However, only a subset of the features and functions of that engine are available in SAP IoT. The following list gives a complete overview of the operators and functions that you can use for defining a rule condition in SAP IoT.

Supported Functions and Operators

Numeric Operations

- Add
- Subtract
- Multiply
- Divide
- Round
- Power
- Sin
- Cos

Logical Operations

- And
- Or
- Not

Comparison Operations

- Is equal to
- Is not equal to
- Is greater than
- Is less than
- Is equal or greater than
- Is equal or less than

String Operations

- Matches
- Does not match
- Contains string
- Does not contain string
- Starts with
- Does not start with
- Ends with
- Does not end with
- Concatenate

Time and Duration

SAP Business Rules provides a comprehensive set of time-related functions for almost any use case you may possibly think of. Instead of replicating the list of functions, we would rather point you to the *Time and Duration* section in the respective help topic: Functions

Related Information

SAP Business Rules: Expression Language 2.0

2.3.3 Create a Rule (Event Model)

Process steps for specifying the rules (based on event properties) that are bundled within a rule project.

Context

In a rule project, you want to set up the rules for monitoring events, based on the data objects that have been specified for the project.

Procedure

1. From the launchpad, start the *Rule Projects* app.
   The system presents the list of rule projects available in the current tenant.
2. Select the rule project for which you want to create a rule.
   The system navigates you to the *General Information* screen of the selected rule project.
3. Choose *Rules*.
4. From the *Create Rule* dropdown list, choose *Streaming → Event Creation*.

   > **i** Note
   >
   > For event-based rules, *Streaming → Event Creation* is the only supported rule type.

   The system presents an empty detail screen where you can enter the settings for the new rule on the *General Information* section.
5. Enter a *Name* for the rule. Optionally, you can also enter a *Short Text* and a *Description*. These three fields serve the following different purposes:

Field	Purpose
Name	Mandatory, **not** translatable. A name indicating the purpose of the rule. You may use this field for entering some abbreviated or coded information that is common to your company (for example, the ID of a building or a department). This field is also used as sorting criterion in the value helps of the Rules Modeler, so it's advisable to follow a systematic naming convention.

Field	Purpose
Short Text	Optional, translatable. A description in natural language indicating the purpose of the rule.
Description	Optional, translatable. Use this field for a few sentences that help other users understand what the rule contains, where it is used, and so on.

6. Optionally, you may define a set of freely defined *Tags* that you can attach to a rule. You can use tags as filter criteria for the list of rules displayed on the *Rules* screen of the app.
7. In the *Input* field, click the value help button.

   The system displays the *Select Rule Input* dialog box.
8. To restrict the available property sets to only those based on events, choose *Event Property Set-based*.

   > **i Note**
   >
   > You can further restrict the number of entries displayed in the list by using the filter fields (*Package* and *Event Type*). Note that filtering by event type requires to first filter by package as well.
9. From the list of available event property sets, choose the ones you want to use as input data objects (one or many) for the rule.

   > **i Note**
   >
   > Depending on the settings of the properties, the system automatically derives the *Sensitivity Level* of the data object. This is relevant for ensuring compliance with data protection and privacy laws.
   >
   > You may choose *Details* to see a list of all the properties that belong to the selected property set.
10. Choose *OK* to confirm your selection.

    > **i Note**
    >
    > The system may suggest adding additional data objects to the rule, which you can confirm or not, depending on your requirements. After that, the system takes you back to the rule definition screen, with the chosen event property sets already entered as *Input*.
    >
    > The system checks and makes sure that all event properties assigned to the rule as input belong to the same event type. Otherwise, you can't save the rule.
11. In the *Output Data Object* field, click the value help button.
12. Choose the event property set to be used as output, just as you did before for the input. However, unlike for input, you can only assign exactly one property set as output.
13. Choose *OK* to confirm your selection.

    > **i Note**
    >
    > The system takes you back to the rule definition screen, with the chosen event property sets already entered as *Output Data Object*.
    >
    > The system checks and makes sure that the event property set assigned to the rule as output isn't already assigned as input (which would result in an infinite loop). Otherwise, you can't save the rule.
14. Choose *Confirm* to transfer the data-related information to the rule editor. This is required for you to continue setting up the rule.

15. In the *Rule Editor* section, define the *If* condition that the rule shall test.

    The system expects the If condition in a simple, predefined syntax, which slightly differs depending on the rule type. For streaming rules, the system expects the following syntax:

    ```
    <property name> of <property set name> <operator> <value>
    ```

    > ### Example
    >
    > ```
    > WaterTemperature of Engine5 is greater than 95
    > ```

    > ### Note
    >
    > We don't recommend typing into the entry fields of the rule editor. Instead, the system offers you a list of all field names and operators that can be used for setting up the rule definition. To expand the list of possible entries, type a space character. Then, pick the desired object with the mouse or the cursor keys.

    > ### Note
    >
    > Rule modeling in SAP IoT supports many, but **not all** of the functions and operators that are available in the underlying rule engine. For a complete list of the supported elements, see Supported Functions and Operators [page 36].

The following steps depend on the rule type:

Rule Type-Specific Settings

Rule Type	Steps
Streaming - Event	1. In the *Settings* section, choose the desired *Execution Mode* for the rule. You have the following options: ○ *Default*: For each monitored value that fulfills the rule condition, the rule triggers an event. ○ *Sleeping*: The rule stops triggering events for a specified duration even if the rule condition is fulfilled. ○ *Toggle*: The rule only triggers an event if the rule condition is fulfilled **and** the condition was **not** fulfilled for previously tested values. ○ *Duration*: Specify a period of time, between 10 seconds and 1 hour, during which no event is triggered even though the rule condition is met. Only if the rule condition is constantly met for a time period longer than defined an event is triggered. ○ *Consecutive condition fulfillment*: Specify a number of met rule conditions higher than 1 for which no event is triggered even though the rule condition is met. Only if the rule condition is met in a gapless sequence more often than the specified number, an event is triggered. All these modes are useful to avoid event flooding in certain scenarios. For more information, see Reducing Event Creation [page 28]. **i Note** The execution modes *Sleeping* and *Duration* are very similar at first sight. Here's the difference: ○ *Sleeping*: **After** a successful rule execution, the rule triggers no events for the specified time period. ○ *Duration*: The rule ignores a continuously fulfilled rule condition and triggers no events **until** the specified time period is over. Any sensor data value that doesn't fulfill the condition resets the time period to the specified duration. 2. Depending on the chosen execution mode, proceed as follows: ○ *Default*: No further settings required. ○ *Sleeping*: Specify the desired *Sleeping Period*. This value must lie within a range from 10 seconds to 1 hour. ○ *Toggle*: No further settings required. ○ *Duration*: For the *Event Creation Mode*, choose *First Match Only* to trigger no more than exactly one event after the specified duration, regardless how often the rule condition is met. ○ *Consecutive condition fulfillment*: For the *Event Creation Mode*, choose *First Match Only* to trigger no more than exactly one event after the specified number is exceeded, regardless how often the rule condition is met. **i Note** If the rule project vocabulary contains threshold values, these values are automatically added to the list of valid input data objects offered by the rule editor.

16. After having defined the *If* condition, you'll find the *Then* block already prepared with the list of all properties belonging to the event property set you have assigned as output data object. Enter the desired values into the relevant fields.
17. Choose *Save*.

Results

The new rule is ready for use. However, for the rule to take effect, you must *Activate* it first.

2.4 Maintain a Rule Project

Process steps for maintaining a rule project.

Context

You want to maintain an existing rule project. For example, you may want to extend the vocabulary so that additional data sources can be used for monitoring. Or you might want to set up additional rules to evaluate the same time series data according to different evaluation criteria.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. In the list of rule projects, click the name of the project you want to change. If you don't see the project that you want to change, use the *Search* or the filter fields to narrow down the number of projects presented in the list.

   After clicking the project name, the system presents the detail screen where you can maintain the settings for the project.
3. Apply your changes as required.
4. Choose *Save*.

2.5 Delete a Rule Project

Process steps for deleting a rule project.

Context

You want to delete a rule project. Possible reasons could be, for example:

- The business scenario covered by a rule project is no longer relevant for your company.
- You want to clean up the system to get rid of rule projects that have been defined for test purposes only.
- Certain changes were made to the devices and sensors monitored by rules, and adapting the existing elements of the rule project would be more complicated and error-prone than setting it up from scratch.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. In the list of rule projects, select the row containing the project you want to delete. If you don't see the project that you want to delete, use the *Search* or the filter fields to narrow down the number of projects presented in the list.
3. Choose *Delete*.

2.6 Data Model Version 2

Differences between version 2 and version 1

Overview

When SAP IoT (or its predecessor products using different names) was introduced into the market, it was centered around the architecture of its own thing model. Along with the thing model, several applications have been added over time to take the best advantage of the data handled by the thing model.

Over time, however, new requirements have arisen. For example, accessing thing models with a different architecture from third-party vendors is now seen as a desirable extension. Also, edge computing for IoT scenarios running in system environments with poor network connections has become more important.

These are examples of new requirements that couldn't be handled easily within the original internal architecture of SAP IoT data model version 1. Therefore, a new version 2 has been introduced with SAP IoT release 2203a in spring 2022, which is better equipped for new requirements like the ones mentioned above.

Availability of the different data model versions

In principle, both versions can be used in parallel within one system. However, a given rule project or action can only be accessed by API services of the particular version in which it was created. The distinction between the models is made via different sets of API services, recognizable via the `/v1` or `/v2` shortcut as part of the service URLs. However, SAP reserves the right of offering only one particular version within the scope of a given data center, or as a part of the customer license agreement. Also, SAP IoT applications that already support version 2 (for example, rule modeling applications) will concentrate all future feature extensions on that new version2, while version 1 will be kept as is without functional extensions.

2.7 Data Object (v.2)

Explains the data object concept in version 2

Overview

Data objects used for input and output of rules that are part of version 2 rule projects are based on the property sets maintained for the following entities in SAP IoT:

- Event Types
  For defining SAP IoT event types, there's currently no end-user application available. Instead, you have to define them programmatically with the help of the respective API services dedicated to this purpose. For more information, see Event Services.

> **i** Note
>
> Currently, only event property sets are supported in version 2. That is, in version 2, you cannot use property sets or property set types defined for the SAP IoT thing model. If the latter is required, you can still accomplish this with rule projects based on the version 1 data model.

Input and Output Data Objects

For processing streaming rules, you need to assign data objects for both input and output. Upon saving (and in some cases even before that), the system checks the inner structure of the assigned data objects and makes sure that all settings are correct and consistent. The following requirements apply:

- You must assign one or more input data objects. All input data objects must refer to the same event type.
- You must assign exactly one output data object. However, it's allowed to refer to an event type using nested property sets, which may originate from different event types.
- The output data object must not refer to any of the event types used by the input data objects.
- The sensitivity level of the output data object must be equal or higher than the highest sensitivity level of all the input data objects.

Example

Data Object Sensitivity Levels

Input	Output	Allowed
personal	none	no
none	none	yes
personal	sensitive	yes

- As soon as a newly created rule has been saved in the last step of the rule creation wizard, the specified data object assignments of the rule cannot be changed anymore. Should you determine any issues or mistakes regarding the assigned data objects, the only way to solve the situation is to create a new rule with the proper data objects assigned.

2.8 Create a Rule Project (v.2)

Process steps for creating a rule project.

Context

You define a rule project based on one or more data objects, which refer to property sets. Property sets are associated with either an event or an event type. The data objects used by the rule project are sometimes referred to as the rule project vocabulary. After defining the data objects, you define the rules that use the different properties contained therein. Depending on the settings, these rules are then executed at runtime depending on the values recorded for a particular property.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. Choose *Create*.

   The system presents the *New Project* dialog box.
3. In the dialog box, enter the general data for the new project, such as the *Name*, and optionally an additional *Short Text* as well as a *Description*. If there are any *External IDs* to be associated with the new rule project, you can enter them in the respective field.

   You can enter multiple IDs in one go by hitting the `Enter` key after each ID.
4. Choose *Create*.

The system navigates you to the *General Information* tab. Here, you can check or modify your entries. Also, in the *Advanced Options* section, you can specify additional settings, such as adding *Annotations* to the rule project, or by specifying its *Managing Type*.

5. Once you're done with the general data of the rule project, choose *Data Objects* to open the section for specifying the data to be used by the rule project.
6. Set up the data objects by specifying the data to be processed by the rules in your rule project. For a detailed procedure, see Create a Data Object (v.2) [page 46].
7. Once you're done with setting up the data objects, choose the *Rules* tab.
8. Set up the rules that are required in your rule project. For a detailed procedure, see Create a Rule (v.2) [page 48].
9. As soon as you're done with setting up both the data objects and the rules, choose *Save*.

Results

The rule project is now ready for use. Note that you have to activate each rule before it can start monitoring incoming time series data.

Related Information

Create a Data Object (v.2) [page 46]
Create a Rule (v.2) [page 48]

2.8.1 Create a Data Object (v.2)

Process steps for specifying the data objects that are relevant for a rule project.

Context

In a rule project, you want to specify the data objects representing the data (either captured by system events or by sensors) to be monitored by the rules in the project. Depending on your usage scenario, you may have to include different types of data objects in your project:

- For all rule types, you need property sets containing properties for *Measured Values*. These properties can be used with the input data object of a rule.
- For derivation rules, you need property sets or property set types containing properties for *Calculated Values*.
- Sensitive data: Property sets containing properties that have been flagged as sensitive **cannot** be used for rule processing. If a rule project contains sensitive data, the Rule Project app makes you aware of this in the header area of the app.

Procedure

1. From the launchpad, start the *Rule Projects* app.
   The system presents the list of rule projects available in the current tenant.
2. Select the rule project for which you want to create a vocabulary.
   The system navigates you to the *General Information* screen of the selected rule project.
3. Choose *Data Objects*.
4. To add a property set to the vocabulary, choose *Add* or click the "+" (plus) symbol on the empty card.
   The system starts the wizard, which guides you through the creation process.
5. Depending on which type of object you want to add, choose from the following options:
   - **Event Model**: Choose a *Package* and an *Event Type* to populate the list of available *Property Sets*.
   - **Custom**: Make the required settings for a custom data object.
6. Once you've made your choice, choose *Step 2* to enter the next step of the wizard.
7. Depending on your previous choice, proceed as follows:

Data Object Type	Settings
Event Model	1. Choose a *Package* and an *Event Type*. 2. In the list of available property sets , tick the check-box of the properties you want to use in your rule project and choose *Review*. 3. Check the settings you've made so far for the new data object: ○ If you see the need for adjustments, choose the corresponding *Edit* link and make the required changes. ○ If you're satisfied with everything, choose *Create*.
Custom	On the details screen for a custom data object, make the required settings and choose *Create*.

   After having chosen *Create*, the system takes you back to the *Data Objects* section of the rule project.
8. Choose *Save*.

Results

All the properties that are available in the selected data objects are now part of the rule project and can be used by the rules defined in the project.

If you have decided to include threshold values in the vocabulary, you can find the corresponding objects listed in the *Related* section of the *Data Object Details* screen.

2.8.2 Create a Rule (v.2)

Process steps for specifying the rules (based on event properties) that are bundled within a rule project.

Context

In a rule project, you want to set up the rules for monitoring events, based on the data objects that have been specified for the project.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. Select the rule project for which you want to create a rule.

   The system navigates you to the *General Information* screen of the selected rule project.
3. Choose *Rules*.
4. Choose *Create Rule*.

   The system presents an empty detail screen where you can enter the settings for the new rule in the *General Information* step of the rule creation wizard.
5. Enter a *Name* for the rule. Optionally, you can also enter a *Short Text* and a *Description*. These three fields serve the following different purposes:

Field	Purpose
Name	Mandatory, **not** translatable. A name indicating the purpose of the rule. You may use this field for entering some abbreviated or coded information that is common to your company (for example, the ID of a building or a department). This field is also used as sorting criterion in the value helps of the Rules Modeler, so it's advisable to follow a systematic naming convention.
Short Text	Optional, translatable. A description in natural language indicating the purpose of the rule.
Description	Optional, translatable. Use this field for a few sentences that help other users understand what the rule contains, where it is used, and so on.

6. Optionally, you may define a set of freely defined *Annotations* that you can attach to a rule.
7. Choose *Step 2* to proceed to the *Input and Output* step of the rule creation wizard.
8. In the *Input Data Objects* section, choose *Add*.

   The system presents the *Select Rule Input* dialog box, which lists all data objects that you have added to the rule project.
9. From the list of *Data Objects*, choose one or more that you want to use as input data provider for the rule. If you choose more than one data object, make sure that all data objects refer to the same event type. When you're done, choose *OK*.

The system takes you back to the *Input and Output* step of the rule creation wizard.

10. In the *Output Data Object* section, choose *Add*.

    The system presents the *Select Output Data Object* dialog box, which lists all data objects that you have added to the rule project.

11. From the list of *Data Objects*, choose exactly one that you want to use as output for the rule. Make sure that the event type of the output data object is **not** used in any of the input data objects. When you're done, choose *OK*.

    The system takes you back to the *Input and Output* step of the rule creation wizard.

12. Choose *Step 3* to proceed to the *Rule Condition* step of the rule creation wizard.
13. Enter the condition to be checked by the rule into the respective fields of the *If-Then-Else* schema offered by the UI.

    In the input fields, the system offers a value help with all the data object properties and the available operators.

    > **i** Note
    >
    > We don't recommend typing into the entry fields of the rule editor. Instead, the system offers you a list of all field names and operators that can be used for setting up the rule definition. To expand the list of possible entries, type a space character. Then, pick the desired object with the mouse or the cursor keys.

    > **i** Note
    >
    > Rule modeling in SAP IoT supports many, but **not all** of the functions and operators that are available in the underlying rule engine. For a complete list of the supported elements, see Supported Functions and Operators [page 36].

14. Once you're done with specifying the rule condition, choose *Step 4* to proceed to the *Execution Mode* step of the rule creation wizard.
15. From the list of possible Execution Modes, choose the one that shall take effect for the rule.

    For a quick overview of the different execution modes, click the ⓘ *Information* symbol on the screen. For more detailed information on execution modes, see Reducing Event Creation [page 28].

16. Choose *Review* to gain a complete overview of all the settings you've made during rule definition.
17. If you see the need for adjustments, choose the corresponding *Edit* link and make the required changes. Otherwise, choose either *Create* or *Create and Activate*.

    > **i** Note
    >
    > After having created (that is, saved) the new rule, it is **not** possible to change any of the data object assignments of the rule. If you should realize at a later point in time that the rule should operate on data provided by more, less, or different data objects, you have to create a new rule with the proper data objects assigned.

Results

The new rule is ready for use. However, for the rule to take effect, you must *Activate* it first. For more information on rule activation, see Activate a Rule (v.2) [page 50].

2.8.3 Activate a Rule (v.2)

Explains the different options regarding rule activation.

Context

When creating a new rule, it's often important to consciously decide where and when that rule should be activated for productive use. The reason for that is that, depending on the scenario, an active rule may immediately start generating large amounts of data, which is often not what you desire. Also, depending on the business scenario and IT setup, numerous settings may be needed for a proper rule activation.

Therefore, as a last step during rule creation, you can choose between just saving the new rule definition and saving plus immediate activation. A nonactivated rule can be activated at any later point in time, and you can always switch between active and inactive. Also, the system keeps track of the rule activation status in a detailed activation log.

Procedure

1. In the *Rule Projects* app, select the rule project containing the rule you're interested in.
2. Choose *Rules* to see the rules contained in the project.
3. Click the rule you want to activate.

   The system displays the rule details screen.
4. Choose *Activate*.

   The system displays the *Rule Activation Details* dialog box. Depending on the setup of your system environment, this dialog box can contain different elements, as shown in the following table:

Activation Target	Description
Cloud	Default target.
Edge	Activation target for rule processing at the edge. Only available in edge-enabled tenants. If edge activation is supported, the dialog box offers an additional section for the *Selection of Edge Targets*. Here, you can choose from the available *Edge Node Descriptors* to be used for rule processing. For more information, see Edge Node Descriptors.

Activation Target	Description
Cloud and Edge	Activation target for rule processing both in the cloud and at the edge. Only available in edge-enabled tenants.

Especially in edge-enabled scenarios with a high number of edge nodes, the activation settings can be highly complex. Therefore, you may find it helpful to fall back to previous settings via *Load Last Activation Settings*.

5. Choose *Activate*.

   The system starts the rule activation process on the selected targets.

> **i** Note
>
> In edge-enabled scenarios, activation as well as deactivation of rules can take a significant amount of time, depending on the availability of the various selected activation targets. Therefore, we recommend refreshing the application several times to ensure that the activation has been done successfully on all selected activation targets.

Results

The rule has been activated on the selected targets and starts listening to the assigned events.

2.9 Maintain a Rule Project (v.2)

Process steps for maintaining a rule project.

Context

You want to maintain an existing rule project. For example, you may want to extend the vocabulary so that additional data sources can be used for monitoring. Or you might want to set up additional rules to evaluate the same time series data according to different evaluation criteria.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. In the list of rule projects, click the name of the project you want to change. If you don't see the project that you want to change, use the *Search* or the filter fields to narrow down the number of projects presented in the list.

After clicking the project name, the system presents the detail screen where you can maintain the settings for the project.

3. Apply your changes as required.
4. Choose *Save*.

2.10 Delete a Rule Project (v.2)

Process steps for deleting a rule project.

Context

You want to delete a rule project. Possible reasons could be, for example:

- The business scenario covered by a rule project is no longer relevant for your company.
- You want to clean up the system to get rid of rule projects that have been defined for test purposes only.
- Certain changes were made to the devices and sensors monitored by rules, and adapting the existing elements of the rule project would be more complicated and error-prone than setting it up from scratch.

Procedure

1. From the launchpad, start the *Rule Projects* app.

   The system presents the list of rule projects available in the current tenant.
2. In the list of rule projects, select the row containing the project you want to delete. If you don't see the project that you want to delete, use the *Search* or the filter fields to narrow down the number of projects presented in the list.
3. Choose *Delete*.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP