|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS #60 |
| Source:\* | Andreas Kraft, DT, [Andreas.Kraft@t-systems.com](mailto:Andreas.Kraft@t-systems.com)  Andreas Neubacher, DT, [Andreas.Neubacher@magenta.at](mailto:Andreas.Neubacher@magenta.at)  Miguel Angel Reina Ortega, ETSI, [MiguelAngel.ReinaOrtega@etsi.org](mailto:MiguelAngel.ReinaOrtega@etsi.org) |
| Date:\* | 2023-07-13 |
| Reason for Change/s:\* | TS-0001: Add “enableEventNotificationOriginator” attribute to subscription resource type |
| CR against: Release\* | Release 5 |
| CR against: WI\* | Active WI-xxxx  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 v.5.2.0 |
| Clauses \* | 9.6.8 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Impacted other TS/TR(s) |  |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2017 (Do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

Introduction

This Change Request follows up on a discussion of SDS-2023-0096 “Set “creator” attribute to request originator in notifications (TS-0004)”, where the conclusion was that the presence of the creator attribute in a notification, and thereby the ID of the originator who causes that notification, should be optional and configurable as part of the subscription.

Therefore, this CR introduces a new attribute “enableEventNotificationOriginator” to the <subscription> resource type.

This change request is for R5, if accepted then further Change Requests for R2, R3, and R4 will be submitted.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 9.6.8 Resource Type *subscription*

The *<subscription>* resource contains subscription information for its subscribed-to resource.

A subscription to a resource allows an entity in the oneM2M architecture to be notified about changes of the subscribed-to resource. The *<subscription>* resource shall represent a subscription to a subscribed-to resource. In order to establish a subscription, a *<subscription>* resource shall be created as a child resource of the subscribed-to resource. The <*subscription*> child resource contains information about the exact scope of the subscription and targets to be notified. For example, a *<container>* resource having a *<subscription>* resource as a child resource (see clause 9.6.6) shall result in notification(s) of target(s) configured in the <*subscription*> child resource when changes to the parent <*container*> resource matching with notification event criteria described by the child <*subscription*> resource occur. A *<subscription>* resource shall be deleted when the parent subscribed-to resource is deleted.

In general, an Originator shall be able to create a resource of *<subscription>* resource type when the Originator has RETRIEVE privilege to the subscribed-to resource. The Originator which creates a *<subscription>* resource becomes the resource subscriber.

A <subscription> resource can be configured to implement a blocking "UPDATE" to a resource or attributes of a resource whereby a notification is sent to the notification target to respond with the result of the "UPDATE" request.

Each *<subscription>* may include notification policies that specify which, when, and how notifications are sent. These notification policies may work in conjunction with CMDH policies.

When a *<subscription>* resource is deleted, a Notify request shall be sent to the target indicated by the attribute *subscriberURI* if it is provided by the Subscriber.

The *<subscription>* resource shall contain the child resources specified in table 9.6.8-1.

Table 9.6.8-1: Child resources of *<subscription>* resource

| Child Resources of <*subscription*> | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *notificationSchedule* | *<schedule>* | 0..1 | In the context of the *<subscription>* resource, the *notificationSchedule* specifies when notifications may be sent by the Hosting CSE to the *notificationURI(s).* See clause 9.6.9. |
| *[variable]* | *<notificationTargetMgmtPolicyRef>* | 0..n | See 9.6.31 for this type of resource. |
| *nstr* | *<notificationTargetSelfReference>* | 1 | See 9.6.34 for this type of resource. |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48. |

The *<subscription>* resource shall contain the attributes specified in table 9.6.8-2.

Table 9.6.8-2: Attributes of *<subscription>* resource

| Attributes of *<subscription>* | Multiplicity | RW/  RO/  WO | Description |
| --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. |
| *resourceID* | 1 | RO | See clause 9.6.1.3. |
| *resourceName* | 1 | WO | See clause 9.6.1.3. |
| *parentID* | 1 | RO | See clause 9.6.1.3. |
| *expirationTime* | 1 | RW | See clause 9.6.1.3. |
| *creationTime* | 1 | RO | See clause 9.6.1.3. |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. |
| *creator* | 0..1 | RO | See clause 9.6.1.3. |
| *custodian* | 0..1 | RW | See clause 9.6.1.3. |
| *eventNotificationCriteria* | 0..1 | RW | This attribute (notification policy) indicates the event criteria for which a notification is to be generated. When no *eventNotificationCriteria* attribute is present in a <*subscription*> resource, the Hosting CSE shall trigger notifications for this subscription when any of the attributes of the subscribed-to resource is modified. |
| *expirationCounter* | 0..1 | RW | This attribute (notification policy) indicates that the subscriber wants to set the life of this subscription to a limit of a maximum number of notifications. When the number of notifications sent reaches the count of this counter, the *<subscription>* resource shall be deleted, regardless of any other policy. |
| *notificationURI* | 1 (L) | RW | This attribute shall be configured as a list consisting of one or more targets that the Hosting CSE shall send notifications to. A target shall be formatted as a oneM2M compliant Resource-ID as defined in clause 7.2 or as an identifier compliant with a oneM2M supported protocol binding (e.g. http, coap, mqtt).  If a target is formatted as a oneM2M compliant Resource-ID, then the target shall be formatted as a structured or unstructured CSE-Relative-Resource-ID, SP-Relative-Resource-ID, and/or Absolute-Resource-ID of an <*AE*> or <CSEBase> resource. A Hosting CSE shall use this information to determine proper pointOfAccess, requestReachability and/or pollingChannel information needed to send a notification to the target. The following is an example.   * /CSE0001/AE0001   For a target that is formatted as an identifier compliant with a oneM2M supported protocol binding, the details of this format are defined by the respective oneM2M protocol specification. The following is an example of an HTTP URI compliant with oneM2M HTTP protocol binding.   * https://172.25.30.25:7000/notification/handler   For a subscription to a <fanoutpoint> resource, if <subscription> resource in request contains a notificationForwardingURI, then the group hosting CSE shall configure the *notificationURI* of the fanout subscription request with an address specified by the Group Hosting CSE that can be used by the Group Hosting CSE to receive aggregated notifications.  A notification serialization type may be appended to each notification target configured in this list. The Hosting CSE shall serialize notifications and send it to a notification target based on this serialization type indicator. Possible serialization types are defined in oneM2M TS-0004 [3] (e.g. XML, JSON or CBOR). If a notification serialization type is not appended to a notification target, a default shall apply based on the Hosting CSE local policy. The syntax for appending a serialization type to a notification target shall use the "?" delimiter character as shown in the below examples.   * http://mydomain/notificationHandler?ct=json * CSE02/base/ae2?ct=xml |
| *groupID* | 0..1 | RW | The ID of a *<group>* resource in case the subscription is made through a group. This attribute may be used in the ***Filter Criteria*** to discover all subscription resources created via a <fanOutPoint> resource to a specific groupID. |
| *notificationForwardingURI* | 0..1(L) | RW | The attribute shall be present only for group related subscriptions. If the subscriber intends the Group Hosting CSE to aggregate the notifications, the attribute shall be set identical to the *notificationURI* attribute. It shall be used by Group Hosting CSE for forwarding aggregated notifications. See clauses 10.2.7.10 and 10.2.7.11. |
| *batchNotify* | 0..1 | RW | This attribute (notification policy) indicates that the subscription originator wants to receive batches of notifications rather than receiving them one at a time. This attribute includes: the number of notifications to be batched for delivery and the duration. When only the number is specified by the subscription originator, the Hosting CSE shall set the default duration given by M2M Service Provider.  If *batchNotify* is used simultaneously with *latestNotify*, only the latest notification shall be sent and have the ***Event Category*** set to "latest". |
| *rateLimit* | 0..1 | RW | This attribute (notification policy) indicates that the subscriber wants to limit the rate at which it receives notifications. This attribute expresses the subscriber's notification policy and includes two values: a maximum number of events that may be sent within some duration, and the *rateLimit* window duration. When the number of generated notifications within the *rateLimit* window duration exceeds the maximum number, notification events are temporarily stored, until the end of the window duration, when the sending of notification events restarts in the next window duration. The sending of notification events continues as long as the maximum number of notification events is not exceeded during the window duration. The *rateLimit* policy may be used simultaneously with other notification policies. |
| *preSubscriptionNotify* | 0..1 | WO | This attribute (notification policy) indicates that the subscriber wants to be sent notifications for events that were generated prior to the creation of this subscription. This attribute has a value of the number of prior notification events requested. If up-to-date caching of retained events is supported on the Hosting CSE and contains the subscribed events, then prior notification events will be sent up to the number requested. The *preSubscriptionNotify* policy may be used simultaneously with any other notification policy. |
| *pendingNotification* | 0..1 | RW | This attribute (notification policy), if set, indicates how missed notifications due to a period of no connectivity are handled (according to the reachability and notification schedules). The possible values for *pendingNotification are*:   * "sendLatest"; * "sendAllPending".   This policy depends upon caching of retained notifications on the hosted CSE. When this attribute is set to "sendLatest", only the last notification shall be sent and it shall have the ***Event Category*** set to "latest". If this attribute is not present, the Hosting CSE sends no missed notifications. This policy applies to all notifications regardless of the selected delivery policy (*batchNotify*, *latestNotify*, etc.). Note that unreachability due to reasons other than scheduling is not covered by this policy. |
| *notificationStoragePriority* | 0..1 | RW | Indicates that the subscriber wants to set a priority for this subscription relative to other subscriptions belonging to this same subscriber. This attribute sets a number within the priority range. When storage of notifications exceeds the allocated size, this policy is used as an input with the storage congestion policy (*notificationCongestionPolicy*) specified in clause 9.6.3 to determine which stored and generated notifications to drop and which ones to retain. |
| *latestNotify* | 0..1 | RW | This attribute (notification policy) indicates if the subscriber wants only the latest notification. If multiple notifications of this subscription are buffered, and if the value of this attribute is set to true, then only the last notification shall be sent and it shall have the ***Event Category*** value set to "latest". |
| *notificationContentType* | 1 | RW | Indicates a notification content type that shall be contained in notifications. For example, "modified attributes" or "all attributes".   * For a list of the default and allowed values of *notificationContentType* for each of the supported values of *notificationEventType* refer to table 9.6.8-4. |
| *notificationEventCat* | 0..1 | RW | This attribute (notification policy) indicates the subscriber's requested ***Event Category*** to be used for notification messages generated by this subscription. |
| *subscriberURI* | 0..1 | WO | This attribute shall be configured with the target of the subscriber. The target is used by the Hosting CSE to determine where to send a notification when the subscription is deleted. A target shall be formatted as a oneM2M compliant Resource-ID as defined in clause 7.2 or as an identifier compliant with one of the oneM2M supported protocol bindings (the detailed format of which are defined by each respective oneM2M protocol binding specification). |
| *associatedCrossResourceSub* | 0..1 | RW | This attribute lists *the identifier of <crossResourceSubscription>* resources where this *<subscription>* is involved in. |
| *primitiveProfileID* | 0..1 | RW | This attribute lists the identifier of a *<primitiveProfile>* resource that specifies attributes and parameters to be added, removed, or modified in the notifications for this subscription. |
| *notificationStatsEnable* | 1 | RW | When set to "TRUE", the Hosting CSE shall clear any statistics that were previously stored in the *notificationStatsInfo* attribute and start recording notification statistics for each notification generated for this resource.  When set to "FALSE", the Hosting CSE shall stop recording notification statistics for this resource and maintain the current value of the *notificationStatsInfo* attribute.  Default is "FALSE". |
| *notificationStatsInfo* | 0..1(L) | RO | A list containing notification statistics recorded by the Hosting CSE for each notification target specified by the *notificationURI* attribute of this resource. The Hosting CSE shall maintain a separate set of notification statistics that include:   * Total number of notification requests sent to a notification target * Total number of notification responses received from a notification target   Refer to oneM2M TS‑0004 [3] for further details regarding the format of this attribute. |
| *enableEventNotificationOriginator* | 0..1 | RW | This attribute determines whether the originator of a request that causes a notification is part of a subscription notification.  Refer to oneM2M TS-0004 [3], clause 7.5.1.2.2, for further details regarding the subscription notification procedure. |

Table 9.6.8-3 describes the *eventNotificationCriteria* conditions.

Table 9.6.8-3: *eventNotificationCriteria* conditions

| Condition tag | Multiplicity | Matching condition |
| --- | --- | --- |
| *createdBefore* | 0..1 | The *creationTime* attribute of the resource is chronologically before the specified value. |
| *createdAfter* | 0..1 | The *creationTime* attribute of the resource is chronologically after the specified value. |
| *modifiedSince* | 0..1 | The *lastModifiedTime* attribute of the resource is chronologically after the specified value. |
| *unmodifiedSince* | 0..1 | The *lastModifiedTime* attribute of the resource is chronologically before the specified value. |
| *stateTagSmaller* | 0..1 | The *stateTag* attribute of the resource is smaller than the specified value. |
| *stateTagBigger* | 0..1 | The *stateTag* attribute of the resource is bigger than the specified value. |
| *expireBefore* | 0..1 | The *expirationTime* attribute of the resource is chronologically before the specified value. |
| *expireAfter* | 0..1 | The *expirationTime* attribute of the resource is chronologically after the specified value. |
| *sizeAbove* | 0..1 | The *contentSize* attribute of the *<contentInstance>* resource is equal to or greater than the specified value. |
| *sizeBelow* | 0..1 | The *contentSize* attribute of the *<contentInstance>* resource is smaller than the specified value. |
| *notificationEventType* | 0..5 | The type of event that shall trigger a notification. If multiple *notificationEventType* tags are present, a notification shall be triggered if any of the configured events occur. Note that not all combinations of event type are meaningful. Possible notification event type values are:   1. Update to attributes of the subscribed-to resource 2. Deletion of the subscribed-to resource, 3. Creation of a direct child of the subscribed-to resource, 4. Deletion of a direct child of the subscribed-to resource 5. An attempt to retrieve a <*contentInstance*> direct-child-resource of a subscribed-to <*container*> resource is performed while this <*contentInstance*> child resource is an obsolete resource or the reference used for retrieving this resource is not assigned. This retrieval is performed by a RETRIEVE request targeting the subscribed-to resource with the Result Content parameter set to either "child-resources" or "attributes+child-resources". This value for the *notificationEventType* tag implies that the subscribed-to resource shall be an <*container*> resource. Otherwise this setting is not valid. 6. Trigger Received targeting the MN/ASN-AE associated with the <AE> parent resource. This implies that the subscribed-to resource shall be an <*AE*> resource instance. Otherwise this setting is not valid. 7. Update to attributes of thesubscribed-to resource with blocking of the triggering UPDATE operation. For this *notificationEventType* value setting, only one single Notification Target shall be present in the *notificationURI* attribute - see *notificationURI* attribute definition. This value for the *notificationEventType* tag shall not be combined with any other *notificationEventType* tag value. This value for *notificationEventType* establishes a subscription that is triggered for the same events as for the value "Update to attributes of the subscribed-to resource". However, upon occurrence of a triggering UPDATE operation that has been validated and results in an authorized UPDATE operation, the triggering UPDATE operation shall be blocked by the Hosting CSE until a notification request was sent out and a corresponding response message was received or a timeout happens. When the response status code of the notification response message indicates a successful notification reception in combination with a successful notification action taken by the Notification Target entity, the triggering UPDATE operation shall be completed with a successful update of the targeted attribute(s). If the notification response message indicates an unsuccessful notification reception or a successful notification reception with unsuccessful notification action by the targeted entity or times out, the blocked UPDATE operation shall be completed with no success and no change of the targeted attribute(s).   For any subscribed-to resource there shall exist a maximum of one subscription with this setting of *notificationEventType*. All other notification policies shall not be allowed when this setting of *notificationEventType* is used. The *notificationContentType* shall be "modified attributes". When an UPDATE operation has been blocked due to triggering this type of notification, any other occurring UPDATE or DELETE requests to the same resource shall be handled only after the blocked UPDATE operation has been completed.   1. Report on missing data points. This *notificationEventType* value shall not be combined with any other *notificationEventType* value.   The other conditions in *eventNotificationCriteria* conditions apply within the scope of the selected *notificationEventType.*  For example, if notificationEventType is "Creation of a direct child of the subscribed-to resource" then other *eventNotificationCriteria* conditions is applied to the direct child resources of the subscribed-to resource.  If this condition is not specified, the default value is "Update to attributes of the subscribed-to resource". This default value shall apply only if *operationMonitor* is not present in the resource.  The notion of "obsolete resource" is defined in clause 9.6.1.3.2 (Common attributes).  For a list of the default and allowed values of *notificationContentType* for each of the supported values of *notificationEventType* refer to table 9.6.8-4. |
| *operationMonitor* | 0..n | The operations and/or the Originators accessing the subscribed-to resource matches with the specified value. It allows monitoring which operation and/or which Originator is attempting to the access subscribed-to resource regardless of whether the operation is performed. This feature is useful to detect AEs that send requests to a subscribed-to resource and that result in a successful or failure response. Possible arguments are operation(s) (e.g.: CREATE, RETRIEVE, UPDATE, DELETE, NOTIFY) and/or Originator identifier(s).  If a set of Originator identifier(s) is included in this tag and no operations are listed, any operations initiated from any of the indicated Originator(s) shall trigger a notification.  If a set of operation(s) is included in this tag and no Originator identifier, any of the listed operations shall trigger a notification.  If both, a set of Originator identifiers and a set of operations are listed, then any of the listed operations initiated from any of the listed Originators shall trigger the notification. |
| *attribute* | 0..n | A list of attribute names of a subscribed-to-resource. This list is only applicable when *notificationEventType* has a value of "Update to attributes of the subscribed-to resource". or "Update to attributes of the subscribed-to resource with blocking of the triggering UPDATE operation".  If this list is present, then it is used to specify a subset of a subscribed-to resource's attributes for which updates shall result in a notification. If ANY attribute specified on this list is updated, then a notification shall be generated. If an attribute that is not specified in this list is updated, then a notification shall not be generated.  If this list is not presented, then the default attribute list is the full set of a subscribed-to resource's attributes. If ANY attribute of a subscribed-to resource is updated, then a notification shall be generated. |
| *childResourceType* | 0.. 1 (L) | A list of resource types. This list is only applicable when *notificationEventType* has a value of "Creation of a direct child of the subscribed-to resource" or "Deletion of a direct child of the subscribed-to resource".  If this list is present, then it is used to specify a subset of resource type for direct child resource of which creation or deletion shall result in a notification. If ANY resource type specified on this list is created or deleted, then a notification shall be generated. If a resource type that is not specified in this list is created or deleted, then a notification shall not be generated.  If this list is not present, then the default resource type list is the full set of a direct child resource. |
| *missingData* | 0..1 | The *missingData* includes two values: a minimum specified missing number of the Time Series Data within the specified window duration, and the window duration. The condition only applies to subscribed-to resources of type *<timeSeries>*. This condition is ignored unless *notificationEventType* has a value of "Report on missing data points". If this attribute is modified by an UPDATE the associated timer/counter are stopped and restarted with the new values.  The first detected missing data point starts the timer associated with the window duration.  The window duration is restarted upon its expiry until such time as the entire subscription is terminated or not refreshed. More details about NOTIFICATIONS related to data reporting is found in clause 10.2.4.29. |
| *filterOperation* | 0..1 | Indicates the logical operation (AND/OR/XOR) to be used for the condition tags *createdBefore, createdAfter, modifiedSince, unmodifiedSince, stateTagSmaller, stateTagBigger, expireBefore, expireAfter, sizeAbove, sizeBelow*. The default value is logical AND. |

The rules when multiple conditions are used together shall be as follows:

* Different condition tags shall use the "AND/OR/XOR" logical operation based on the *filterOperation* specified.
* Same condition tags shall use the "OR" logical operation.

The XOR operation evaluates to true if and only if an odd number of its inputs are true.

No mixed AND/OR/XOR filter operation will be supported.

Table 9.6.8-4 defines the default and allowed values of *notificationContentType* for each of the supported values of *notificationEventType*.

Table 9.6.8-4: Default and allowed values of *notificationContentType*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| notificationEventType  notificationContentType | A | B | C | D | E | F | G | H |
| "modified attributes" | valid | n/a | n/a | n/a | n/a | n/a | valid (default) | n/a |
| "all attributes" | valid (default) | valid (default) | valid (default) | valid (default) | valid (default) | n/a | n/a | n/a |
| "ID" of the resource indicated in the *notificationEventType* condition | valid | valid | valid | valid | valid | n/a | n/a | n/a |
| "Trigger Payload" | n/a | n/a | n/a | n/a | n/a | valid (default) | n/a | n/a |
| "TimeSeries notification" | n/a | n/a | n/a | n/a | n/a | n/a | n/a | valid (default) |

### 9.6.9 Resource Type *schedule*

The *<schedule>* resource contains scheduling information. The usage of the *<schedule>* resource is slightly different depending on the associated resource type, as follows:

* A child *<schedule>* resource of the *<node>* resource shall indicate the time periods when the node can communicate via the Underlying Network. If multiple Underlying Networks are supported, for each there can be a maximum of one <schedule> resources. One <schedule> resource may be used for multiple Underlying Networks.
* The *mgmtLink* attribute of the *<cmdhNwAccessRule>* child of a <node> resource shall link to a <schedule> resource, child of the same <node> resource.

NOTE: The node will obey the communication schedule indicated for the Underlying Network. If the schedule information is modified, the node will ensure that the change of schedule is detected e.g. via external DM, subscription/notification mechanisms, polling, etc.

* A child <*schedule*> resource of the <*CSEBase*> resource shall indicate the anticipated time periods when the CSE is available for processing.
* A child *<schedule>* resource of the *<subscription>* resource shall indicate the time periods when the notifications can be sent to the notification targets.

The <schedule> resource shall contain the child resource specified in table 9.6.9-1.

Table 9.6.9-1: Child resources of *<schedule>* resource

| Child Resources of *<schedule>* | Child Resource Type | Multiplicity | Description | *<scheduleAnnc>* Child Resource Types |
| --- | --- | --- | --- | --- |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 | None |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 | *<transaction>* |

The *<schedule>* resource shall contain the attributes specified in table 9.6.9-2.

Table 9.6.9-2: Attributes of *<schedule>* resource

| Attributes of  *<schedule>* | Multiplicity | RW/  RO/  WO | Description | *<scheduleAnnc>* Attributes |
| --- | --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 1 | RW | See clause 9.6.1.3. | MA |
| *creationTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *labels* | 0..1 (L) | RW | See clause 9.6.1.3. | MA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announceSyncType* | 0..1 | RW | See clause 9.6.1.3. | MA |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3 | NA |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *custodian* | 0..1 | RW | See clause 9.6.1.3. | NA |
| *scheduleElement* | 1 (L) | RW | Each item of the *scheduleElement* list shall be composed from seven fields of second, minute, hour, day of month, month, day of week and year. | OA |
| *networkCoordinated* | 0..1 | RW | Indicates if CSE shall perform schedule coordination with an Underlying Network. This attribute is only applicable when <schedule> is a child resource of <node>. The supported values are:   * True: The CSE shall perform schedule coordination. * False: The CSE may not perform schedule coordination.   See note. | OA |
| NOTE: The schedule coordination is also subject to CSE local policy. | | | | |

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*