**3GPP TSG-SA5 Meeting #162*****S5-*** ***253896***

, , - revision of S5- 253679

**Source: Ericsson**

**Title: Rel-19 pCR TS 28.572 Plan Configuration Stage-3**

**Document for: Approval**

**Agenda Item: 6.19.9.1 - Management of planned configurations**

# 1 Decision/action requested

***The group is requested to discuss and approve the pCR below.***

# 2 References

[1] 3GPP TS 28.572: "Management of planned configurations"

# 3 Rationale

Provide an OpenAPI stage 3 solution.

# 4 Detailed proposal

The following changes are proposed for TS 28.572[1].

|  |
| --- |
| **Begin of modifications** |

# Annex A (normative): Solution sets

## A.1 RESTful HTTP-based solution set

### A.1.1 General Considerations

The Naming of information elements in stage-3 generally is the same as in stage-2. However, there will be elements/resources that have different names between stage 2 and stage 3 dependent on the contentType of the planned configuration. Any such deviations are documented in clause TBD.

The planConfigContentType specifies the type of the content in the 'planConfig'. This is required so that the MnS Producer may correctly interpret the provided plan configuration data. The configurationContentType application/3gpp-yang-patch shall be one such supported content type. This type is based on the yang patch specification as per RFC 8072 but with the following exceptions:

*'target'* : As per section 2.4 of RFC-8072 but allowing for the omission of the module prefix (e.g. it is possible to specify a *target* like '/\_3gpp-common-subnetwork:SubNetwork=Ireland /\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp-common-managed-element:ManagedElement=Dublin-1/\_3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1' or in the relaxed form '/SubNetwork=Ireland/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1').

*'value'* : Encoded according to RFC-7951 but allowing the omission of the module prefix. The MnS Producer shall accept this shorter form of the value if it determines the overall value is unambiguous.

### A.1.2 Resource structure

The resource structure on the MnS producer is as follows:

…/{MnSName}/{/MnSVersion}/plan-descriptors/{id}

…/{MnSName}/{/MnSVersion}/plan-descriptors/{id}/plan-config

…/{MnSName}/{/MnSVersion}/plan-group-descriptors/{id}

…/{MnSName}/{/MnSVersion}/plan-validation-jobs/{id}

…/{MnSName}/{/MnSVersion}/plan-validation-jobs/{id}/status

…/{MnSName}/{/MnSVersion}/plan-validation-jobs/{id}/validation-details

…/{MnSName}/{/MnSVersion}/plan-activation-jobs/{id}

…/{MnSName}/{/MnSVersion}/plan-activation-jobs/{id}/status

…/{MnSName}/{/MnSVersion}/plan-activation-jobs/{id}/activation-details

**Creating, reading, updating and deleting planned configurations**

The data node tree on the MnS producer is as follows upon system start up.

|  |
| --- |
| {  "plan-descriptors": {},  "plan-group-descriptors": {},  "fallback-config-descriptors": {},  "trigger-condition-descriptors": {},  "plan-validation-jobs": {},  "plan-activation-jobs": {}  } |

### A.1.3 OpenAPI definitions

OpenAPI definitions for the management of planned configurations are specified in Forge, refer to clause 4.3 of TS 28.623 [16] for the Forge location. An example of Forge location is: "https://forge.3gpp.org/rep/sa5/MnS/-/tree/Tag\_Rel18\_SA105/".

Directory: OpenAPI

Files:

TS28572\_PlanManagement.yaml

### A.1.4 Plan Descriptor Examples

A new item of the collection resource "plan-descriptors" is created by MnS consumers using HTTP POST.

|  |
| --- |
| POST 3gpp/ProvMnS/v1/plan-descriptors HTTP/1.1  Host: example.org  Content-Type: application/json  {  "name": "NewBts10Plan",  "version": "2.0",  "description": "This is the plan for the new BTS 10.",  "planConfigContentType": "application/vnd.3gpp.json-patch+json",  "currentConfigAddress": "http://example.org/3gpp/ProvMnS/v1",  "applyMode": "ATOMIC",  "planConfig": [  {  "key": "op1",  "op": "add",  "path": "/SubNetwork=SN1/ManagedElement=ME10",  "value": {  "id": "ME10",  "attributes": {  "userLabel": "Berlin NW 1",  "vendorName": "Company XY",  "location": "Castle Charlottenburg"  }  }  }  ]  } |

The MnS producer allocates the identifier "p1" for the new resource and returns the response. The location header contains the URI of the new resource. The response body contains the representation of the new resource which is equal to the representation received in the request with the "lastModifiedAt" and "validationState" properties added.

|  |
| --- |
| HTTP/1.1 201 Created  Date: Wed, 21 Aug 2024 15:39:57 GMT  Location: http://example.org/3gpp/ProvMnS/v1/plan-descriptors/p1  Content-Type: application/json  {  "name": "NewBts10Plan",  "version": "2.0",  "description": "This is the plan for the new BTS 10.",  "planConfigContentType": "application/vnd.3gpp.json-patch+json",  "currentConfigAddress": "http://example.org/3gpp/ProvMnS/v1",  "applyMode": "ATOMIC",  "lastModifiedAt": " 2025-03-06T16:50:26-08:00 ",  "validationState": "NOT\_VALIDATED",  "planConfig": [  {  "key": "op1",  "op": "add",  "path": "/SubNetwork=SN1/ManagedElement=ME10",  "value": {  "id": "ME10",  "attributes": {  "userLabel": "Berlin NW 1",  "vendorName": "Company XY",  "location": "Castle Charlottenburg"  }  }  }  ]  } |

The resource structure on the MnS producer contains the new resource.

|  |
| --- |
| {  "plan-descriptors": {  "p1": {  "name": "NewBts10Plan",  "version": "2.0",  "description": "This is the plan for the new BTS 10.",  "planConfigContentType": "application/vnd.3gpp.json-patch+json",  "currentConfigAddress": "http://example.org/3gpp/ProvMnS/v1",  "applyMode": "ATOMIC",  "lastModifiedAt": " 2025-03-06T16:50:26-08:00 ",  "validationState": "NOT\_VALIDATED",  "planConfig": [  {  "key": "op1",  "op": "add",  "path": "/SubNetwork=SN1/ManagedElement=ME10",  "value": {  "id": "ME10",  "attributes": {  "userLabel": "Berlin NW 1",  "vendorName": "Company XY",  "location": "Castle Charlottenburg"  }  }  }  ]  }  },  "plan-group-descriptors": {},  "fallback-config-descriptors": {},  "trigger-condition-descriptors": {},  "plan-validation-jobs": {},  "plan-activation-jobs": {}  } |
|  |

The next example shows how the value of the "location" attribute in the operation can be changed from "Castle Charlottenburg" to "Summer palace Charlottenburg".

|  |
| --- |
| PATCH 3gpp/ProvMnS/v1/plan-descriptors/p1 HTTP/1.1  Host: example.org  Content-Type: application/json-patch+json  [  {  "op": "replace",  "path": "/planConfig/0/value/attributes/location",  "value": "Summer palace Charlottenburg"  }  ] |

To delete the planned configuration "p1" a MnS consumer might send.

|  |
| --- |
| DELETE 3gpp/ProvMnS/v1/plan-descriptors/p1 HTTP/1.1  Host: example.org |

In case of success, the MnS producer returns the following message.

|  |
| --- |
| HTTP/1.1 204 No Content  Date: Wed, 21 Aug 2024 16:12:45 GMT |

The next example shows how a planned configuration can be created for the case where the current configuration data node tree is specified with YANG.

|  |
| --- |
| POST 3gpp/ProvMnS/v1/plan-descriptors HTTP/1.1  Host: example.org  Content-Type: application/json  Accept : application/json,application/problem+json  {  "name" : "planxyz",  "applyMode" : "BEST\_EFFORT",  # Execute in a best effort manner trying all operations  "configurationContentType" : "application/3gpp-yang-patch+json"  "planConfig": {  "  "edit" : [  {  "operation": "create",  "editId" : "opId-001",  "target": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/:GNBDUFunction=1",  "value": {  "NRCellDU": [  {  "id": "1",  "attributes": {  "administrativeState" : "LOCKED",  ...  }  }  }  },  {  "operation": "merge",  "editId" : "opId-002",  "target": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=2",  "value": {  "attributes": {  "administrativeState": "LOCKED",  }  }  },  {  "operation": "remove",  "editId" : "opId-003",  "target": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=3"  }  ]  }  }  } |

To get the planConfig :

|  |
| --- |
| GET 3gpp/ProvMnS/v1/plan-descriptors/p1/plan-config HTTP/1.1  Host: example.org  Content-Type: application/json |

To get a specific operation/edit in a planConfig :

|  |
| --- |
| GET 3gpp/ProvMnS/v1/plan-descriptors/p1/plan-config/{editId} HTTP/1.1  Host: example.org  Content-Type: application/json |

The next example shows how the planConfig edit / operation with editId=opId-001 can be modified to change the value of the "administrativeState" attribute from "LOCKED" to "UNLOCKED". The 'editId' may be used to uniquely identify the specific configuration operation/edit to be modified in the planConfig.

|  |
| --- |
| PATCH 3gpp/ProvMnS/v1/plan-descriptors/p1/planConfig/opId-002 HTTP/1.1  Host: example.org  Content-Type: application/json  {  "value": {  "attributes": {  "administrativeState": "LOCKED",  }  }  } |

To create a new edit entry to the end of the edits:

|  |
| --- |
| PATCH 3gpp/ProvMnS/v1/plan-descriptors/p1/planConfig HTTP/1.1  Host: example.org  Content-Type: application/json  {  "operation": "merge",   "editId" : "opId-004",  "target": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=4",  "value": {  "attributes": {  "administrativeState": "UNLOCKED",  }  }  } |

To create a new edit entry before/after an existing edit use "before" or "after" parameter options:

|  |
| --- |
| PATCH 3gpp/ProvMnS/v1/plan-descriptors/p1/planConfig?after=opId-001 HTTP/1.1  Host: example.org  Content-Type: application/json  {  "operation": "merge",   "editId" : "opId-004",  "target": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=4",  "value": {  "attributes": {  "administrativeState": "UNLOCKED",  }  }  } |

### A.1.5 Plan Activation Job Examples

This section shows examples for plan activation job related resources. This example uses the 'inline' form of the plan descriptor:

|  |
| --- |
| **# Create plan-activation-job**  POST {apiRoot}/ProvMnS/1900/plan-activation-jobs HTTP/1.1  Host: example.org  Content-Type: application/json  Accept : application/json,application/problem+json  {  "name" : "job-xyz",  "description" : "optimize the Dublin area network",  "isFallbackEnabled" : true,  "planConfigDescrId" : "planxyz"  }  **Response**  {  "id": "myjob-111",  "name" : "job-xyz",  "description" : "optimize the Dublin area network",  "isFallbackEnabled" : true,  "planConfigDescrId" : "planxyz",  "jobState" : "RUNNING",  "startedAt" : "<some-time>",  "activationState" : "NOT\_STARTED",  "activationDetails": {  "href" : "{apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/activation-details"  },  "cancelRequest" : false,  "links" : {  "self" : { "href" : "{apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111"},   "planDescriptor" : "{apiRoot}/ProvMnS/1900/…/planxyz"},  "status": { "href" : "{apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/status"},  "fallback": { "href" : "…"}  }  }  **# GET plan-activation-job status (job COMPLETE). Report all the relevant status information**  **# related to the plan activation job**  GET {apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/status  **Response**  {  "jobState": "COMPLETE",  "activationState": "ACTIVATED",  "startedAt": "2024-12-02T13:16:54.088Z",  "stoppedAt": "2024-12-02T13:16:58.088Z"  }  **# GET plan-activation-job activation-details**  **# - jobState=COMPLETED / activationState=ACTIVATED**  **# - All configuration edits/operations are successfully activated**  **# - no detailed info on failed edits/operations are reported (none failed)**  GET {apiRoot}/ProvMnS /1900/plan-activation-jobs/myjob-111/activation-details  **Response**  {  "summary" : {  "unprocessed": 0,  "succeeded": 3,  "failed": 0  }  }  **# GET plan-activation-job activation-details with verbose option (expand=all) # - jobState=COMPLETED / activationState=ACTIVATED # - All configuration edits/operations are successfully activated**  **# - no detailed info on failed edits/operations are reported (none failed)**  GET {apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/activation-details?expand=all  **Response**  {  "editStatus" : [  {  "planDescriptorId" : "plan-desc-001",  "edit" : [  {  "editId" : "opId-001",  "state" : "SUCCEEDED"  },  {  "editId" : "opId-002",  "state" : "SUCCEEDED"  },  {  "editId" : "opId-003",  "state" : "SUCCEEDED"  }  ]  }  ],  "summary" : {  "unprocessed": 0,  "succeeded": 3,  "failed": 0  }  }  **# GET plan-activation-job result-details with complete configuration edit/operation failure**  **# - jobState=COMPLETED / activationState=ACTIVATION\_FAILED**  **# - All configuration edits/operations failed to activate**  **# - Activation details on failed edits/operations are reported**  GET {apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/activation-details  **Response**  {  "editStatus" : [  {  "planDescriptorId" : "planxyz",  "edit" : [  {  "editId" : "opId-001",  "state" : "FAILED",  "errors" : {  "error" : [  {  "type": "application",  "reason": "data-exists",  "path": "/\_3gpp-common-subnetwork:SubNetwork=Irl/\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp-common-managed-element:ManagedElement=Dublin-1/\_3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1/\_3gpp-nr-nrm-nrcelldu:NRCellDU=1",  "message": "Data already exists; cannot be created"  }  ]  }  },  {  "editId" : "opId-002",  "state" : "FAILED",  "errors" : {  "error" : [  {  "type": "application",  "reason": "data-exists",  "path": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=2",  "message": "Data already exists; cannot be created"  }  ]  }  },  {  "editId" : "opId-003",  "state" : "FAILED",  "errors" : {  "error" : [  {  "type": "application",  "reason": "data-exists",  "path": "/SubNetwork=Irl/MeContext=Dublin-1/ManagedElement=Dublin-1/GNBDUFunction=1/NRCellDU=3",  "message": "Data already exists; cannot be created"  }  ]  }  }  ]  }  ],  "summary" : {  "unprocessed": 0,  "succeeded": 0,  "failed": 3  }  }  **# GET plan-activation-job result-details with operation/edit failure (with expand=all option)**  **# - jobState=COMPLETED / activationState=PARTIALLY\_ACTIVATED**  **# - Some configuration edits/operations failed to activate**  **# - Activation details on failed edits/operations are reported**  **# - Activation details on successful edits/operations are reported (expand=all)**  GET {apiRoot}/ProvMnS/1900/plan-activation-jobs/myjob-111/activation-details?expand=all  **Response**  {  "editStatus" : [  {  "planDescriptorId" : "planxyz",  "edit" : [  {  "editId" : "opId-001",  "state" : "FAILED",  "errors" : {  "error" : [  {  "type": "application",  "reason": "data-exists",  "path": "/\_3gpp-common-subnetwork:SubNetwork=Irl/\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp-common-managed-element:ManagedElement=Dublin-1/3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1/\_3gpp-nr-nrm-nrcelldu:NRCellDU=1",  "message": "Data already exists; cannot be created"  }  ]  }  },  {  "editId" : "opId-002",  "state" : "SUCCEEDED"  },  {  "editId" : "opId-003",  "state" : "SUCCEEDED"  }  ]  }  ],  "summary" : {  "unprocessed": 0,  "succeeded": 2,  "failed": 1  }  } |

|  |
| --- |
| **End of modifications** |

# 5 TS28572\_PlanManagement.yaml

Editor's note: This clause shall NOT be part of the specification. It is only included here as a copy of the normative Forge specification.

Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1875> at commit 34770b79e8f46cd395dae549a927fccfbca9f57a

\*\*\* START OF CHANGE 1 \*\*\*

\*\*\* OpenAPI/TS28572\_PlanManagement.yaml \*\*\*

<CODE BEGINS>

openapi: 3.0.0

info:

title: 3GPP Plan Provisioning Management API

version: 19.0.0

description: API for managing network configuration plans and related jobs

externalDocs:

description: 3GPP TS 28.572; Generic management services

url: http://www.3gpp.org/ftp/Specs/archive/28\_series/28.572/

servers:

- url: '{MnSRoot}/ProvPlanMnS/{MnSVersion}'

variables:

MnSRoot:

description: See clause 4.4.2 of TS 32.158

default: http://example.com/PlanManagement

MnSVersion:

description: Version number of the OpenAPI definition

default: v1

paths:

/plan-descriptors:

post:

tags:

- Plan Descriptor Management

summary: Create a new plan descriptor

description: Creates a new configuration plan descriptor that can be later activated.

operationId: createPlanDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationDescriptor'

example:

{

"name": "Rollout-5G-Dublin-East",

"version" : "1.0.0",

"description": "This is the plan for the new 5G rollout in Dublin east.",

"customProperties": {

"technology-type": "NR",

"location": "Dublin"

},

"currentConfigAddress": "http://example.org/3gpp/ProvMnS/v1",

"configurationContentType" : "application/vnd.3gpp.yang-patch+json",

"planConfig": {

...

}

}

responses:

'201':

description: Plan descriptor created successfully

headers:

Location:

description: URI of the created plan descriptor.

schema:

type: string

format: uri-reference

example: "/plan-descriptors/my-plan11"

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationDescriptorResponse'

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'409':

description: A plan descriptor with the given ID already exists.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

get:

tags:

- Plan Descriptor Management

summary: Get plan configuration descriptors

description: Retrieve a list of existing plan descriptors.

operationId: getPlanDescriptors

responses:

'200':

description: List of the plan configuration descriptors retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/DescriptorListEntry'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/plan-descriptors/{id}:

parameters:

- in: path

name: id

schema:

type: string

description: Unique identifier of the plan descriptor.

example: "NewNetworkElement10-group-plan-001"

required: true

get:

tags:

- Plan Descriptor Management

summary: Get a specific plan descriptor by ID

description: Retrieve the details of a single plan descriptor using its unique identifier.

operationId: getPlanDescriptorById

responses:

'200':

description: Plan descriptor retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationDescriptorResponse'

'404':

description: Plan descriptor does not exist

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

put:

tags:

- Plan Descriptor Management

summary: Replace a plan descriptor

description: Replace a configuration plan descriptor

operationId: putPlanDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationDescriptor'

example:

{

"name": "Rollout-5G-Dublin-East",

"description": "This is the plan for the new 5G rollout in Dublin east.",

"customProperties": {

"technology-type": "NR",

"location": "Dublin"

},

"currentConfigAddress": "http://example.org/3gpp/ProvMnS/v1",

"planConfig": {

...

}

}

responses:

'204':

description: Plan descriptor replaced successfully. No content is returned

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'404':

description: Plan descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Plan Descriptor Management

summary: Delete a plan descriptor by ID

description: Deletes a specific configuration plan descriptor using its unique identifier

operationId: deletePlanDescriptorById

responses:

'204':

description: Plan descriptor deleted successfully.

'404':

description: Plan descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail' ##EDITOR The example in swagger is very strange. The example cannot be the same for put and delete. TODO LATER

'409':

description: Conflict - The plan descriptor cannot be deleted due to its current state or dependencies (For example, it is referenced in a plan group descriptor).

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/plan-group-descriptors:

post:

tags:

- Plan Group Descriptor Management

summary: Create a new plan group descriptor

description: Creates a new configuration plan group descriptor that can be later activated.

operationId: createPlanGroupDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationGroupDescriptor'

responses:

'201':

description: Plan group descriptor created successfully

headers:

Location:

description: URI of the created plan descriptor.

schema:

type: string

format: uri-reference

example: "/plan-group-descriptors/mygroup-11"

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationGroupDescriptorResponse'

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

get:

tags:

- Plan Group Descriptor Management

summary: Get plan configuration descriptors

description: Retrieve a list of existing plan descriptors.

operationId: getPlanGroupDescriptors

responses:

'200':

description: List of the plan configuration descriptors retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/DescriptorListEntry'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/plan-group-descriptors/{id}:

parameters:

- in: path

name: id

schema:

type: string

description: Unique identifier of the plan group descriptor.

example: "NewNetworkElement10-group-plan-001"

required: true

get:

tags:

- Plan Group Descriptor Management

summary: Get a specific plan group descriptor by ID

description: Retrieve the details of a plan group descriptor using its unique identifier.

operationId: getPlanGroupDescriptorById

responses:

'200':

description: Plan group descriptor retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationGroupDescriptorResponse'

'404':

description: Plan group descriptor does not exist

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

put:

tags:

- Plan Group Descriptor Management

summary: Replace a plan group descriptor

description: Replace a configuration plan group descriptor

operationId: putPlanGroupDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationGroupDescriptor'

example:

{

"name": "Rollout-5G-Dublin-East",

"version" : "1.0.0",

"description": "This is the plan for the new 5G rollout in Dublin east.",

"activationMode" : "BEST\_EFFORT",

"isOrdered" : true,

"isFailOnMemberConflicts" : true,

"members": [{"type" : "PLAN", "identifier" : "plan-descriptor-001"}, {"type" : "PLAN\_GROUP", "identifier" : "plan-group-descriptor-001"}]

}

responses:

'200':

description: Plan group descriptor replaced successfully

content:

application/json:

schema:

$ref: '#/components/schemas/PlanConfigurationGroupDescriptorResponse'

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'404':

description: Plan group descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Plan Group Descriptor Management

summary: Delete a plan group descriptor by ID

description: Deletes a plan group descriptor using its unique identifier

operationId: deletePlanGroupDescriptorById

responses:

'204':

description: Plan descriptor deleted successfully.

'404':

description: Plan descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'409':

description: The plan group descriptor cannot be deleted due to its current state or dependencies (For example, it is referenced in a plan group descriptor).

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/fallback-descriptors:

get:

tags:

- Fallback Descriptor Management

summary: Get fallback configuration descriptors

description: Retrieve a list of existing fallback descriptors.

operationId: getFallbackDescriptors

responses:

'200':

description: List of the fallback configuration descriptors retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/DescriptorListEntry'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/fallback-descriptor/{id}:

parameters:

- in: path

name: id

schema:

type: string

description: Unique identifier of the fallback descriptor.

example: "Fallback-Dublin-South-plan-001"

required: true

get:

tags:

- Fallback Descriptor Management

summary: Get a specific fallback descriptor by ID

description: Retrieve the details of a single fallback descriptor using its unique identifier.

operationId: getFallbackDescriptorById

responses:

'200':

description: Fallback Plan descriptor retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/FallbackConfigurationDescriptorResponse'

'404':

description: Fallback descriptor does not exist

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Fallback Descriptor Management

summary: Delete a Fallback descriptor by ID

description: Deletes a specific fallback plan descriptor using its unique identifier

operationId: deleteFallbackDescriptorById

responses:

'204':

description: Fallback descriptor deleted successfully.

'404':

description: Fallback descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail' ##EDITOR The example in swagger is very strange. The example cannot be the same for put and delete. TODO LATER

'409':

description: Conflict - The fallback descriptor cannot be deleted due to its current state.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/trigger-descriptors:

post:

tags:

- Trigger Descriptor Management

summary: Create a new trigger descriptor

description: Creates a new trigger descriptor

operationId: createTriggerDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/TriggerDescriptor'

responses:

'201':

description: Trigger descriptor created successfully

headers:

Location:

description: URI of the created trigger descriptor.

schema:

type: string

format: uri-reference

example: "/trigger-descriptors/mytrigger-11"

content:

application/json:

schema:

$ref: '#/components/schemas/TriggerDescriptorResponse'

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

get:

tags:

- Trigger Descriptor Management

summary: Get trigger descriptors

description: Retrieve a list of existing trigger descriptors.

operationId: getTriggerDescriptors

responses:

'200':

description: List of the trigger descriptors retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/DescriptorListEntry'

/trigger-descriptors/{id}:

parameters:

- in: path

name: id

schema:

type: string

description: Unique identifier of the descriptor.

example: "NewNetworkElement10-trigger-001"

required: true

get:

tags:

- Trigger Descriptor Management

summary: Get a specific trigger descriptor by ID

description: Retrieve the details of a single descriptor using its unique identifier.

operationId: getTriggerDescriptorById

responses:

'200':

description: Plan descriptor retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/TriggerDescriptorResponse'

'404':

description: Trigger descriptor not found

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

put:

tags:

- Trigger Descriptor Management

summary: Replace a trigger descriptor

description: Replace a trigger descriptor

operationId: putTriggerDescriptor

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/TriggerDescriptor'

responses:

'204':

description: Trigger descriptor created successfully

'400':

description: Invalid request parameters or malformed input.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'404':

description: Trigger descriptor not found

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Trigger Descriptor Management

summary: Delete a trigger descriptor by ID

description: Deletes a specific trigger descriptor using its unique identifier

operationId: deleteTriggerDescriptorById

responses:

'204':

description: Trigger descriptor deleted successfully.

'404':

description: Trigger descriptor not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/activation-jobs:

post:

tags:

- Activation Management

summary: Create a new plan activation job

description: Creates and starts a new plan activation job based on an existing plan descriptor (or plan group descriptor or fallback descriptor). The new job's ID will be generated by the server and returned in the Location header.

operationId: createActivationJob

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/ActivationJobRequest'

responses:

'201':

description: Plan activation job created successfully.

The response body provides job details, and the Location header points to the new job.

headers:

Location:

description: URI of the created job resource.

schema:

type: string

format: uri-reference

example: "/activation-jobs/myjob-111"

content:

application/json:

schema:

$ref: '#/components/schemas/ActivationJob'

'400':

description: Invalid request payload or parameters (e.g., malformed JSON, missing required fields).

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

get:

tags:

- Activation Management

summary: Get plan activation jobs

description: Retrieve a list of plan activation jobs.

operationId: getActivationJobs

parameters:

- in: query

name: job-state

schema:

$ref: '#/components/schemas/JobState'

description: Filter jobs by their current state.

example: "RUNNING"

responses:

'200':

description: List of plan activation jobs retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/JobListEntry'

'400':

description: Invalid query parameters

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/activation-jobs/{id}:

parameters:

- $ref: '#/components/parameters/jobId'

get:

tags:

- Activation Management

summary: Get plan activation job details by ID

description: Retrieve detailed information about a specific plan activation job using its unique identifier.

operationId: getActivationJobById

responses:

'200':

description: Job details retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/ActivationJob'

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Activation Management

summary: Delete a plan activation job by ID

description: Deletes a specific plan activation job, typically if it's not in a terminal state (e.g., running, completed, failed).

operationId: deleteActivationJobById

responses:

'204':

description: Job deleted successfully. No content is returned.

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'409':

description: Cannot delete job in its current state (e.g., already running).

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/activation-jobs/{id}/status:

parameters:

- $ref: '#/components/parameters/jobId'

get:

tags:

- Activation Management

summary: Get job status

description: Retrieve the current status of a specific plan activation job using its unique identifier.

operationId: getJobStatus

responses:

'200':

description: Job status retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/ActivationJobStatus'

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

patch:

tags:

- Activation Management

summary: Cancel the activation job

description: Cancel the activation job

operationId: cancelActivationJobById

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/CancelRequest'

responses:

'200':

description: Job cancel request was successfully completed

'202':

description: Job cancel request was accepted and cancellation is ongoing

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'422':

description: Semantic error - e.g. job was not in RUNNING state

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/activation-jobs/{id}/activation-details:

get:

tags:

- Activation Management

summary: Get activation details

description: Retrieve detailed information about the activation results of a job

operationId: getActivationDetails

parameters:

- $ref: '#/components/parameters/jobId'

responses:

'200':

description: Activation details retrieved successfully

content:

application/json:

schema:

$ref: '#/components/schemas/ExecutionDetails'

'404':

description: Job not found

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/validation-jobs:

post:

tags:

- Validation Management

summary: Create a new plan validation job

description: Creates and starts a new plan validation job based on an existing plan descriptor. The new job's ID will be generated by the server and returned in the Location header.

operationId: createValidationJob

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/ValidationJobRequest'

responses:

'201':

description: Plan validation job created successfully.

The response body provides job details, and the Location header points to the new job.

headers:

Location:

description: URI of the created job resource.

schema:

type: string

format: uri-reference

example: "/validation-jobs/myjob-111"

content:

application/json:

schema:

$ref: '#/components/schemas/ValidationJob'

'400':

description: Invalid request payload or parameters (e.g., malformed JSON, missing required fields).

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

get:

tags:

- Validation Management

summary: Get plan validation jobs

description: Retrieve a list of plan validation jobs.

operationId: getValidationJobs

parameters:

- in: query

name: job-state

schema:

$ref: '#/components/schemas/JobState'

description: Filter jobs by their current status.

example: "RUNNING"

responses:

'200':

description: List of plan validation jobs retrieved successfully.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/JobListEntry'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/validation-jobs/{id}:

parameters:

- $ref: '#/components/parameters/jobId'

get:

tags:

- Validation Management

summary: Get plan validation job details by ID

description: Retrieve detailed information about a specific plan validation job using its unique identifier.

operationId: getValidationJobById

responses:

'200':

description: Job details retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/ValidationJob'

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

delete:

tags:

- Validation Management

summary: Delete a plan validation job by ID

description: Deletes a specific plan validation job, typically if it's not in a terminal state (e.g., running, completed, failed).

operationId: deleteValidationJobById

responses:

'204':

description: Job deleted successfully. No content is returned.

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/validation-jobs/{id}/status:

get:

tags:

- Validation Management

summary: Get job status

description: Retrieve the current status of a specific plan validation job using its unique identifier.

operationId: getValidationJobStatus

parameters:

- $ref: '#/components/parameters/jobId'

responses:

'200':

description: Job status retrieved successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/ValidationJobStatus'

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

patch:

tags:

- Activation Management

summary: Cancel the validation job

description: Cancel the validation job

operationId: cancelActivationJobById

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/CancelRequest'

responses:

'200':

description: Job cancel request was successfully completed

'202':

description: Job cancel request was accepted and cancellation is ongoing

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'422':

description: Semantic error - e.g. job was not in RUNNING state

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/validation-jobs/{id}/validation-details:

get:

tags:

- Validation Management

summary: Get validation details

description: Retrieve detailed information about the validation results of a job

operationId: getValidationDetails

parameters:

- $ref: '#/components/parameters/jobId'

- name: details

in: query

description: Selects the level of details to return.

schema:

type: string

enum: [summary, all]

default: all

responses:

'200':

description: Validation details retrieved successfully

content:

application/json:

schema:

$ref: '#/components/schemas/ExecutionDetails'

'404':

description: Job not found

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

/validation-jobs/{id}/cancel-request:

parameters:

- $ref: '#/components/parameters/jobId'

put:

tags:

- Validation Management

summary: Cancel the validation job

description: Cancel the validation job

operationId: cancelValidationJobById

responses:

'200':

description: Job cancel request was successfully completed

'202':

description: Job cancel request was accepted and cancellation is ongoing

'404':

description: Job not found.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'422':

description: Semantic error - e.g. job was not in RUNNING state

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

'500':

description: Internal server error.

content:

application/problem+json:

schema:

$ref: '#/components/schemas/ErrorDetail'

components:

parameters:

jobId:

name: id

in: path

description: Unique identifier of the job

required: true

schema:

type: string

example: "myjob-111"

descriptorId:

name: descriptorId

in: path

description: Unique identifier of the plan descriptor

required: true

schema:

type: string

example: "plan-descriptor-001"

schemas:

PlanConfigurationDescriptor:

type: object

required:

- configurationContentType

- planConfig

properties:

id:

type: string

description: Unique id of the plan configuration descriptor

example: "plan-001"

name:

type: string

description: Descriptive name of the plan group configuration descriptor

example: "Rollout-5G-Dublin-East"

version:

type: string

description: The version of the planned configuration. Its format is implementation specific.

example: 1.0.0

description:

type: string

description: Used to describe the purpose of the plan configuration

example: "This is the plan for the new 5G rollout in Dublin east."

customProperties:

type: object

description: A dynamic set of custom properties provided by client

additionalProperties: true

example:

technology-type: NR

location: Dublin

currentConfigAddress:

type: string

format: uri-reference

description: A reference address to the current configuration associated with the configuration in this descriptor

example: "http://example.org/3gpp/ProvMnS/v1"

activationMode:

type: string

enum: [ATOMIC, BEST\_EFFORT, STOP\_ON\_ERROR]

default : "BEST\_EFFORT"

description: Specifies the execution behavior when the plan is activated

example: "BEST\_EFFORT"

lastModifiedAt:

type: string

format: date-time

description: the last time the plan was modified

example: "2025-03-06T16:50:26-08:00"

validationState:

type: string

enum: [NOT\_VALIDATED, VALIDATING, VALID, INVALID]

default : "NOT\_VALIDATED"

description: The validation state of the plan.

example: "NOT\_VALIDATED"

lastValidatedAt:

type: string

format: date-time

description: last time the plan was validated

example: "2025-03-06T16:50:29-08:00"

configurationContentType:

type: string

enum:

- application/vnd.3gpp.json-patch+json

- application/vnd.3gpp.yang-patch+json

description: The format/type of the configuration in planConfig

example: "application/vnd.3gpp.yang-patch+json"

planConfig:

description: "The plan configuration changes"

oneOf:

- type: array

items:

$ref: '#/components/schemas/Change'

- type: object

additionalProperties:

$ref: '#/components/schemas/Change'

description: A map of changes. The key of the map shall have the same value as the changeId.

PlanConfigurationDescriptorResponse:

type: object

required:

- id

- activationMode

- lastModifiedAt

- validationState

- configurationContentType

- planConfig

allOf:

- $ref: '#/components/schemas/PlanConfigurationDescriptor'

properties:

\_links:

type: object

description: Hypermedia links for plan descriptor

allOf:

- $ref: '#/components/schemas/SelfLink'

example:

self:

href: "{root-url}/ProvMnS/v1/plan-descriptors/pd-001"

templated: true

type: "application/json"

title: "The newly created PlanConfigurationDescriptor"

PlanConfigurationGroupDescriptor:

type: object

required :

- members

properties:

id:

type: string

description: Unique id of the plan group configuration descriptor

example: "plan-001"

name:

type: string

description: Descriptive name of the plan group configuration descriptor

example: "Rollout-5G-Dublin-East"

description:

type: string

description: Used to describe the purpose of the plan group configuration

example: "This is the plan for the new 5G rollout in Dublin east."

customProperties:

type: object

description: A dynamic set of custom properties provided by client

additionalProperties: true

example:

technology-type: NR

location: Dublin

isOrdered:

type: boolean

description: Specifies if the members of the planned configuration group are ordered. When ordered, the planned configuration group members shall be validated/activated in the specified order. When not ordered the planned configuration group members can be validated/activated in any order

default: "false"

isFailOnMemberConflicts:

type: boolean

description: Specifies if the activation shall fail on detection of conflicts between planned configuration group members, or if the operations shall be processed as if there were no conflicts

default: "false"

activationMode:

type: string

enum: [ATOMIC, BEST\_EFFORT, STOP\_ON\_ERROR]

default : "BEST\_EFFORT"

description: Specifies the execution behavior when the plan configuration group is activated

example: "BEST\_EFFORT"

lastModifiedAt:

type: string

format: date-time

description: the last time the plan was modified

example: "2025-03-06T16:50:26-08:00"

validationState:

type: string

enum: [NOT\_VALIDATED, VALIDATING, VALID, INVALID]

default : "NOT\_VALIDATED"

description: The validation state for the last time plan configuration group was validated

example: "NOT\_VALIDATED"

lastValidatedAt:

type: string

format: date-time

description: last time the plan was validated

example: "2025-03-06T16:50:29-08:00"

members:

type: array

description: list of plan or plan group descriptor identifiers

items:

type: object

required:

- identifier

properties:

type:

type: string

enum: [PLAN, PLAN\_GROUP]

default: PLAN

identifier:

type: string

example: [{"type" : "PLAN", "identifier" : "plan-descriptor-001"}, {"type" : "PLAN\_GROUP", "identifier" : "plan-group-descriptor-001"}]

FallbackConfigurationDescriptor:

type: object

required:

- configurationContentType

- planConfig

properties:

id:

type: string

description: Unique id of the fallback configuration descriptor

example: "fallback-001"

name:

type: string

description: Descriptive name of the fallback configuration descriptor

example: "Fallback-Rollout-5G-Dublin-East"

version:

type: string

description: The version of the fallback configuration. Its format is implementation specific.

example: 1.0.0

description:

type: string

description: Used to describe the purpose of the fallback configuration

example: "Fallback for configuration plan Rollout-5G-Dublin-East."

customProperties:

type: object

description: A dynamic set of custom properties provided by client

additionalProperties: true

example:

technology-type: NR

location: Dublin

activationJob:

type: string

description: The identifier of the related activation job.

configurationContentType:

type: string

enum:

- application/vnd.3gpp.json-patch+json

- application/vnd.3gpp.yang-patch+json

description: The format/type of the configuration in planConfig

example: "application/vnd.3gpp.yang-patch+json"

planConfig:

description: "The fallback configuration."

oneOf:

- type: string

- type: object

additionalProperties:

$ref: '#/components/schemas/Change'

description: A map of changes. The key of the map shall have the same value as the changeId.

FallbackConfigurationDescriptorResponse:

type: object

required:

- id

- configurationContentType

- planConfig

allOf:

- $ref: '#/components/schemas/FallbackConfigurationDescriptor'

properties:

\_links:

type: object

description: Hypermedia links for fallback descriptor

allOf:

- $ref: '#/components/schemas/SelfLink'

example:

self:

href: "{root-url}/ProvMnS/v1/fallback-descriptors/pd-001"

templated: true

type: "application/json"

title: "The newly created FallbackConfigurationDescriptor"

PlanConfigurationGroupDescriptorResponse:

type: object

required:

- id

- activationMode

- validationState

- isOrdered

- isFailOnMemberConflicts

- members

allOf:

- $ref: '#/components/schemas/PlanConfigurationGroupDescriptor'

properties:

\_links:

type: object

description: Hypermedia links for plan descriptor

allOf:

- $ref: '#/components/schemas/SelfLink'

example:

self:

href: "{root-url}/ProvMnS/v1/plan-group-descriptors/pgd-001"

templated: true

type: "application/json"

title: "The newly created PlanConfigurationGroupDescriptor"

TriggerDescriptor:

type: object

required:

- conditionExpression

- evaluationPeriod

properties:

id:

type: string

description: Unique id of the plan configuration descriptor

example: "trigger-001"

name:

type: string

description: Descriptive name of the trigger descriptor

example: "Rollout-5G-Dublin-East"

description:

type: string

description: Used to describe the purpose of the trigger

example: "This is the plan for the new 5G rollout in Dublin east."

version:

type: string

description: an optional version for the Trigger Descriptor

example: "v1.0"

customProperties:

type: object

description: A dynamic set of custom properties provided by client

additionalProperties: true

example:

technology-type: NR

location: Dublin

conditionExpression:

type: string

description: The condition expression ##BALAZS JEX on NRM - OPEN

activationJobs:

type: array

items:

type : string

description: The identifiers of one or more activation jobs that shall be triggered by this condition.

startAt:

type: string

format: date-time

description: The date and time at which the evaluation of the condition expression shall start. The evaluation result is set to "False" before that date and time. If the information element is not specified, evaluation of the trigger condition shall start immediately.

example: "2025-03-06T16:50:26-08:00"

stopAt:

type: string

format: date-time

description: The date and time at which the evaluation of the condition expression shall stop. The evaluation result is set to "False" after that date and time. If the information element is not specified, evaluation of the trigger condition shall continue until the deletion of the trigger condition descriptor.

example: "2025-03-06T16:50:26-08:00"

currentEvaluationResult:

type: boolean

default : false

description: The current result of evaluating the "condition-expression".

example: false

lastModifiedAt:

type: string

format: date-time

description: The date and time at which the trigger condition was modified the last time by a MnS consumer. Upon creation of the trigger condition descriptor the value of the information element is set to the date and time at which the descriptor is created.

example: "2025-03-06T16:50:29-08:00"

lastTriggeredAt:

type: string

format: date-time

description: Th date and time at which the evaluation result of the trigger condition changed the last time from "False" to "True".

example: "2025-03-06T16:50:29-08:00"

isTriggerActive:

type: boolean

default : false

description: The indication if the trigger can start activation jobs (trigger is active), or if the trigger cannot start activation jobs (trigger is inactive).

example: false

isTriggerOnce:

type: boolean

default : true

description: The boolean indication, if the trigger is disarmed after the first firing.

example: false

evaluationPeriod:

type: integer

description: The evaluation period specifies the interval of time in seconds between two consecutive condition expression evaluations.

hysteresis:

type: integer

description: The hysteresis, when present, specifies that the trigger shall not be activated immediately when the evaluation result changes from false to true or a specified number of times. Values greater or equal to 1 are allowed

TriggerDescriptorResponse:

type: object

required:

- id

- conditionExpression

- evaluationPeriod

- isTriggerOnce

- currentEvaluationResult

- isTriggerActive

allOf:

- $ref: '#/components/schemas/TriggerDescriptor'

properties:

\_links:

type: object

description: Hypermedia links for trigger descriptor

allOf:

- $ref: '#/components/schemas/SelfLink'

example:

self:

href: "{root-url}/ProvMnS/v1/trigger-descriptors/trigger-001"

templated: true

type: "application/json"

title: "The newly created TriggerDescriptor"

DescriptorListEntry:

type: array

items:

type: object

properties:

id:

type: string

description: id of the descriptor.

example: plan-descriptor-1

name:

type: string

description: name of the descriptor.

example: Dublin East Rollout

description:

type: string

description: description/purpose of the descriptor.

example: plan-descriptor-1

required :

- id

- name

JobListEntry:

type: object

allOf:

- $ref: '#/components/schemas/JobState'

properties:

id:

type: string

description: id of the job.

example: plan-job-1

name:

type: string

description: name of the job.

example: Dublin East Rollout Job

description:

type: string

description: description/purpose of the job.

example: Job to do dublin east rollout

required :

- id

- name

- jobState

ActivationJobRequest:

type: object

properties:

name:

type: string

description: Name of the activation job

example: "Dublin East Cell Deployment"

description:

type: string

description: Human-readable description of the job

example: "Optimize the Dublin area network"

mnsConsumerId:

type: array

description: The user that created and/or started the job. It may indicated a human user and/or one or more applications initiating the job. E.g. ["userid:janedoe", "appid:12314"]

items:

type: string

isFallbackEnabled:

type: boolean

description: Whether fallback should be enabled for this job

default: true

serviceImpact:

type: string

enum: [LEAST\_SERVICE\_IMPACT, SHORTEST\_TIME]

description: Additional job-specific parameters

default: "SHORTEST\_TIME"

isImmediateActivation:

type: boolean

description: Specifies if the activation job shall start immediately or, alternatively, by conditional activation.

default: true

oneOf:

- type: object # Alt.1 planConfigDescrId

required:

- planConfigDescrId

properties:

planConfigDescrId:

type: string

description: Unique id reference to the plan descriptor to activate

example: "planxyz"

- type: object # Alt.2 planConfigDescr embedded

required:

- planConfigDescr

properties:

planConfigDescr:

type: object

description: Inline plan configuration descriptor to activate

allOf:

- $ref: '#/components/schemas/PlanConfigurationDescriptor'

example:

name: "myjob-111"

activationMode: "BEST\_EFFORT"

customProperties: {

techology-type: "NR",

indoor: false

}

planConfig:

...

- type: object # Alt.3 planConfigGroupDescrId

required:

- planConfigGroupDescrId # Alt.3 planConfigGroupDescrId

properties:

planConfigGroupDescrId:

type: string

description: Unique id reference to the plan group descriptor to activate

example: "plan-group-xyz"

- type: object # Alt.4 planConfigGroupDescr embedded

required:

- planConfigGroupDescr

properties:

planConfigGroupDescr:

type: string

description: Inline plan group descriptor to activate

allOf:

- $ref: '#/components/schemas/PlanConfigurationGroupDescriptor'

- type: object # Alt.5 fallbackConfigDescrId

required:

- fallbackConfigDescrId

properties:

fallbackConfigDescrId:

type: string

description: Unique id reference to the fallback descriptor to activate

example: "planxyz"

ActivationJob:

type: object

required:

- id

- status

- isImmediateActivation

- isFallbackEnabled

- serviceImpact

- jobDetails

- \_links

description: An object representing a plan activation job.

allOf:

- type: object

properties:

id:

type: string

description: ID of the activation job.

example: "job-id-3985199134"

status:

$ref: '#/components/schemas/ActivationJobStatus'

jobDetails:

$ref: '#/components/schemas/JobDetails'

activationDetails:

allOf:

- $ref: '#/components/schemas/LinkObject'

- type: object

properties:

href:

type: string

description: A URI reference to the activation details.

example: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/activation-details"

title:

type: string

enum:

- "Link to the activation details"

method:

type: string

enum:

- "GET"

type:

type: string

description: The content type expected when following this link (MIME type).

default: "application/json"

mnsConsumerId:

type: array

description: The user that created and/or started the job. It may indicate a human user and/or one or more applications initiating the job. E.g. ["userid:janedoe", "appid:12314"].

items:

type: string

\_links:

description: Hypermedia links for this resource, including fixed and dynamic relations.

allOf:

- $ref: '#/components/schemas/JobLinks'

- type: object

properties:

self:

allOf:

- $ref: '#/components/schemas/LinkObject'

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/activation-jobs/{ActivationJobId}"

title:

type: string

enum:

- "Link to the plan activation job"

method:

type: string

enum:

- "GET"

example:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/activation-job-001"

title: "Link to the plan activation job"

type: "application/json"

templated: true

method: "GET"

fallback:

allOf:

- $ref: '#/components/schemas/LinkObject'

- type: object

description: A URI reference to the fallback plan descriptor.

example:

href: "{apiRoot}/ProvMnS/v1/plan-descriptors/myjob-111-fallback"

templated: true

type: "application/json"

title: "A URI reference to the fallback plan descriptor"

method: "GET"

additionalProperties:

$ref: '#/components/schemas/LinkObject'

- $ref: '#/components/schemas/ActivationJobRequest'

example:

id: "myjob-111"

name: "5G-Dublin-East-Rollout"

description: "Optimize the 5G network in Dublin East"

isFallbackEnabled: true

serviceImpact: "SHORTEST\_TIME"

isImmediateActivation: true

status:

jobState: "RUNNING"

startedAt: "<date-time>"

activationState: "NOT\_STARTED"

jobDetails: {} # Assuming this is a simple object

activationDetails:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/activation-details"

templated: true

type: "application/json"

title: "The activation details of the plan configuration(s)"

method: "GET"

planConfigDescrId: "plan-descriptor-001"

\_links:

self:

href: "{apiRoot}/ProvMnS/1900/activation-jobs/myjob-111"

templated: true

type: "application/json"

title: "The newly created activation job"

method: "GET"

planDescriptor:

href: "{apiRoot}/ProvMnS/v1/plan-descriptors/planxyz"

templated: true

type: "application/json"

title: "plan descriptor link"

method: "GET"

status:

href: "{apiRoot}/ProvMnS/1900/activation-jobs/myjob-111/status"

templated: true

type: "application/json"

title: "activation status link"

method: "GET"

activation-details:

href: "{apiRoot}/ProvMnS/1900/activation-jobs/myjob-111/activation-details"

templated: true

type: "application/json"

title: "activation details link"

method: "GET"

cancel:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/cancel-request"

templated: true

type: "application/json"

title: "cancel the job"

method: "PUT"

fallbackPlan:

href: "{apiRoot}/ProvMnS/v1/plan-descriptors/myfallback-plan-111"

templated: true

type: "application/json"

title: "fallback plan descriptor link"

method: "GET"

additionalProperties: true

ValidationJobRequest:

type: object

required:

- \_links

description: An object representing a plan activation job.

properties:

name:

type: string

description: Name of the validation job

example: "Dublin East Cell Deployment"

description:

type: string

description: Human-readable description of the job

example: "Optimize the Dublin area network"

validationMode:

type: string

enum: [CONTINUE\_ON\_ERROR, STOP\_ON\_ERROR]

default : "CONTINUE\_ON\_ERROR"

description: Specifies the execution behavior when the plan is activated

mnsConsumerId:

type: array

description: The user that created and/or started the job. It may indicated a human user and/or one or more applications initiating the job. E.g. ["userid:janedoe", "appid:12314"]

items:

type: string

oneOf:

- type: object # Alt.1 planConfigDescrId

required:

- planConfigDescrId

properties:

planConfigDescrId:

type: string

description: Unique id reference to the plan descriptor to validate

example: "planxyz"

- type: object # Alt.2 planConfigDescr embedded

required:

- planConfigDescr

properties:

planConfigDescr:

type: object

description: Inline plan configuration descriptor to validate

allOf:

- $ref: '#/components/schemas/PlanConfigurationDescriptor'

example:

name: "myjob-111"

activationMode: "BEST\_EFFORT"

customProperties: {

techology-type: "NR",

indoor: false

}

planConfig:

...

- type: object # Alt.3 planConfigGroupDescrId

required:

- planConfigGroupDescrId

properties:

planConfigGroupDescrId:

type: string

description: Unique id reference to the plan group descriptor to validate

example: "plan-group-xyz"

- type: object # Alt.4 planConfigGroupDescr embedded

required:

- planConfigGroupDescr

properties:

planConfigGroupDescr:

type: string

description: Inline plan group descriptor to validate

allOf:

- $ref: '#/components/schemas/PlanConfigurationGroupDescriptor'

- type: object # Alt.5 fallbackConfigDescrId

required:

- fallbackConfigDescrId

properties:

fallbackConfigDescrId:

type: string

description: Unique id reference to the fallback descriptor to validate

example: "planxyz"

example:

{

name : "5G-Dublin-East-Rollout",

description : "Optimize the 5G network in Dublin East",

planConfigDescrId : "plan-descriptor-001",

}

additionalProperties : true

ValidationJob:

type: object

required:

- id

- status

- jobDetails

- validationDetails

- \_links

allOf:

- $ref: '#/components/schemas/ValidationJobRequest'

- type: object

properties:

id:

type: string

description: id of the validation job

example: "job-id-3985199134"

status:

allOf:

- $ref: '#/components/schemas/ValidationJobStatus'

jobDetails:

allOf:

- $ref: '#/components/schemas/JobDetails'

currentConfigTime:

type: string

format: date-time

validationDetails:

allOf:

- $ref: '#/components/schemas/LinkObject'

- type: object

properties:

href:

type: string

title: A URI reference to the validation details

example: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/validation-details"

title:

type: string

enum:

- "Link to the validation details"

method:

type: string

enum:

- "GET"

type:

type: string

description: The content type expected when following this link (MIME type).

default: "application/json"

\_links:

description: Hypermedia links for this resource, including fixed and dynamic relations

allOf:

- $ref: '#/components/schemas/JobLinks'

- type: object

properties:

self:

allOf:

- $ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/validation-jobs/{validationJobId}"

title:

type: string

enum:

- "Link to the plan validation job"

method:

type: string

enum:

- "GET" # This means the method MUST be "GET"

example:

href: "{apiRoot}/ProvMnS/v1/validation-jobs/validation-job-001"

title: "Link to the plan validation job"

type: "application/json"

templated: true

method: GET

cencel:

allOf:

- $ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/validation-jobs/{validationJobId}/status"

title:

type: string

enum:

- "Link to the plan validation job"

method:

type: string

enum:

- "PATCH" # This means the method MUST be "GET"

example:

href: "{apiRoot}/ProvMnS/v1/validation-jobs/validation-job-001/status"

title: "Link to the cancel validation job - cancelRequest : true must be in body"

type: "application/json"

templated: true

method: PUT

required :

- self

- cancel

# additionalProperties to allow any other dynamic links

additionalProperties:

$ref: '#/components/schemas/LinkObject'

example:

self:

href: "{apiRoot}/ProvMnS/1900/activation-jobs/myjob-111"

templated: true

type: "application/json"

title: "The newly created activation job"

method : "GET"

planDescriptor:

href: "{apiRoot}/ProvMnS/v1/plan-descriptors/planxyz"

templated: true

type: "application/json"

title: "plan descriptor link"

method : "GET"

status:

href: "{apiRoot}/ProvMnS/1900/activation-jobs/myjob-111/status"

templated: true

type: "application/json"

title: "activation status link"

method : "GET"

cancel:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/status"

templated: true

type: "application/json"

title: "cancel the job"

method : "PATCH"

example:

{

id : "myjob-111",

name : "5G-Dublin-East-Rollout",

description : "Optimize the 5G network in Dublin East",

isFallbackEnabled : true,

serviceImpact : "SHORTEST\_TIME",

isImmediateActivation : true,

status : {

jobState : "RUNNING",

jobDetails : {...},

startedAt : "<date-time>",

activationState : "NOT\_STARTED"

},

validationDetails : {

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/validation-details",

templated: true,

type: "application/json",

title: "The validation details of the plan configuration(s)",

method : "GET"

},

planConfigDescrId : "plan-descriptor-001",

}

additionalProperties : true

CancelRequest:

type: object

properties:

cancelRequest:

type: boolean

description: property to cancel the activation/validation job

example: true

Change:

type: object

properties:

changeId:

type: string

description: Unique identifier for this change

example: "opId-001"

operation:

type: string

enum: [create, merge, update, delete]

description: The operation to perform

example: "create"

description:

type: string

description: text describing the change

example: "modify NR cell for optimisation"

target:

type: string

description: Target data node path

example: "/\_3gpp-common-subnetwork:SubNetwork=Irl/3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp\_nrm\_managedelement:ManagedElement=1/\_3gpp\_nrm\_nrcelldu:NRCellDU=4"

value:

type: object

additionalProperties: true # Allows any nested properties within 'value'

description: Value to apply (for create/update/merge/delete operations)

additionalProperties: true # Allows for any nested properties

required:

- operation

- target

JobLinks:

type: object

allOf:

- $ref: '#/components/schemas/SelfLink'

properties:

planDescriptor:

description: A URI reference to the plan (or plan group) configuration descriptor

allOf:

- $ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/plan-descriptors/{planDescriptorId}"

title:

type: string

enum:

- "Link to the plan (or plan group) configuration descriptor"

method:

type: string

enum:

- "GET" # This means the method MUST be "GET"

example:

href: "{apiRoot}/ProvMnS/v1/plan-descriptors/plan-descriptor-001"

title: "Link reference to the plan (or plan group) configuration descriptor"

type: "application/json"

templated: true

method: GET

status:

allOf:

- $ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/activation-jobs/{jobId}/status"

title:

type: string

enum:

- "Link to GET the job status"

method:

type: string

enum:

- "GET" # This means the method MUST be "GET"

description: A URI reference to the status information

example:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/status"

title: "Link to GET the job status"

type: "application/json"

templated: true

method: GET

cancel:

allOf:

- $ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

- type: object

properties:

href:

type: string

default: "{apiRoot}/ProvMnS/v1/plan-{job-type}-jobs/{jobId}/status"

title:

type: string

enum:

- "Link to cancel the job"

method:

type: string

enum:

- "PATCH" # This means the method MUST be "PATCH"

description: A URI reference to cancel the job

example:

href: "{apiRoot}/ProvMnS/v1/activation-jobs/myjob-111/status"

title: "Link to cancel the job"

type: "application/json"

templated: true

method: PATCH

JobState:

type: string

enum: [NOT\_STARTED, QUEUED, RUNNING, CANCELLING, CANCELLED, COMPLETED, FAILED]

example: "COMPLETED"

JobDetails:

properties:

jobInformation:

type: string

additionalProperties: true

ActivationJobStatus:

properties:

jobState:

allOf:

- $ref: '#/components/schemas/JobState'

example: "COMPLETED"

activationState:

allOf:

- $ref: '#/components/schemas/ActivationState'

example: "ACTIVATED"

startedAt:

type: string

format: date-time

example: "2024-12-02T13:16:54.088Z"

stoppedAt:

type: string

format: date-time

example: "2024-12-02T13:16:58.088Z"

ValidationJobStatus:

properties:

jobState:

allOf:

- $ref: '#/components/schemas/JobState'

example: "COMPLETED"

validationState:

allOf:

- $ref: '#/components/schemas/ValidationState'

example: "VALID"

startedAt:

type: string

format: date-time

example: "2024-12-02T13:16:54.088Z"

stoppedAt:

type: string

format: date-time

example: "2024-12-02T13:16:58.088Z"

ExecutionDetails:

type: object

description: Details of the execution of the operations that are contained in the planned configuration or planned configuration group referenced in the job.

properties:

results:

type: array

items:

type: object

properties:

planDescriptorId:

type: string

example: "planconfig-descriptor-001"

changes:

type: array

items:

$ref: '#/components/schemas/ChangeStatus'

summary:

$ref: '#/components/schemas/SummaryStatus'

example:

{

"results": { ...},

"summary" : {

"unprocessed": 0,

"succeeded": 2,

"failed": 3

}

}

ValidationState:

type: string

enum: [UNKNOWN, VALIDATION\_SUCCEEDED, VALIDATION\_FAILED]

example: "VALIDATED"

ActivationState:

type: string

enum: [NOT\_STARTED, VALIDATING, ACTIVATED, ACTIVATION\_FAILED, PARTIALLY\_ACTIVATED, ACTIVATION\_FAILED\_ROLLED\_BACK, ACTIVATION\_FAILED\_ROLLBACK\_FAILED]

example: "ACTIVATED"

ChangeStatus:

type: object

properties:

changeId:

type: string

example: "changeId-001"

state:

type: string

enum: [NOT\_STARTED, PROCESSING, SUCCEEDED, FAILED]

default: NOT\_STARTED

example: "SUCCEEDED"

target:

type: string

example: "/SubNetwork=1/MeContext=2/ManagedElement=3"

errors:

type: array

items:

$ref: '#/components/schemas/ErrorDetail'

conflicts:

type: array

items:

$ref: '#/components/schemas/Conflict'

required:

- changeId # either the changeId from the plan or the array index in the array of Change;

- target

- state

example:

changeId: changeId-002-update-config

target: /SubNetwork=1/MeContext=2/ManagedElement=3

state: FAILED

errors:

- type: VALIDATION\_ERROR

title: Validation Error

status: 400

reason: NEW\_ATTRIBUTE\_VALUE\_INVALID

detail: The provided value 'some-value' for 'someAttr' is not allowed. Valid values are 'some-value-x', 'some-value-y'.

path: /SubNetwork=1/MeContext=2/ManagedElement=3/attributes/someAttr

errorInfo:

attributeName: someAttr

invalidValue: some-value

SummaryStatus:

type: object

properties:

notFinished:

type: integer

example: 1

succeeded:

type: integer

example: 3

failed:

type: integer

example: 3

rollbackSucceeded:

type: integer

example: 0

rollbackFailed:

type: integer

example: 0

Conflict:

type: object

allOf:

- $ref: '#/components/schemas/Change'

properties:

planDescriptorId:

type: string

change:

$ref: '#/components/schemas/Change'

conflictingChanges:

type: array

items:

$ref: '#/components/schemas/ConflictingChange'

example:

{

"planDescriptorId" : "plan-descriptor-001",

"changeId": "optimization-change-A",

"operation": "merge",

"target": "/\_3gpp-common-subnetwork:SubNetwork=HQ/NRCellDU=cell-22",

"value" : {

"attributes" : {

"some-attribute-name-a" : "some-attribute-value-b"

}

},

"conflictingChanges": [

{

"planDescriptorId": "plan-emergency-fix",

"change": {

"changeId": "fix-power-issue-X",

"operation": "merge",

"target": "/\_3gpp-common-subnetwork:SubNetwork=HQ/NRCellDU=cell-22",

"value" : {

"attributes" : {

"some-attribute-name-x" : "some-attribute-value-y"

}

},

},

"conflictType": "VALUE\_OVERWRITE",

"conflictMessage": "This change attempts to set 'transmitPower' to 45, but 'fix-power-issue-X' from 'plan-emergency-fix' sets it to 40. Only one value can persist.",

"severity": "HIGH"

}

]

}

ConflictingChange:

type: object

properties:

planDescriptorId:

type: string

change:

$ref: '#/components/schemas/Change'

conflictType:

type: string

conflictMessage:

type: string

severity:

type: string

enum:

- LOW

- MEDIUM

- HIGH

- UNKNOWN

default: UNKNOWN

required:

- change

- conflictType

- severity

ErrorDetail:

type: object

properties:

title:

type: string

description: A short, human-readable summary of the problem type

example: "Data already exists; cannot be created"

status:

type: integer

format: int32

enum:

- 400

- 403

- 404

- 405

- 406

- 408

- 410

- 411

- 413

- 414

- 415

- 422

- 426

- 429

- 451

type:

type: string

description: The type of the error

enum:

- VALIDATION\_ERROR

- REQUEST\_OBJECT\_TREE\_MISMATCH

- IE\_NOT\_FOUND

- MODIFICATION\_NOT\_ALLOWED

- RETRIEVAL\_NOT\_ALLOWED

- SERVER\_LIMITATION

- SERVICE\_DISABLED

- APPLICATION\_LAYER\_ERROR

reason:

type: string

enum:

- RESPONSE\_TOO\_LARGE

- NO\_DATA\_ACCESS

- QUERY\_MALFORMED

- QUERY\_PARAM\_NAMES\_INVALID

- QUERY\_PARAM\_VALUES\_INVALID

- QUERY\_PARAMS\_MISSING

- QUERY\_PARAMS\_INCONSISTENT

- ATTRIBUTES\_NOT\_READABLE

- QUERY\_PARAMS\_TOO\_COMPLEX

- NEW\_ATTRIBUTE\_VALUE\_INVALID

- NEW\_ATTRIBUTE\_NAME\_INVALID

- ATTRIBUTE\_NOT\_WRITABLE

- ATTRIBUTE\_INVARIANT

- ATTRIBUTE\_NOT\_FOUND

- OBJECT\_CREATION\_NOT\_ALLOWED

- OBJECT\_DELETION\_NOT\_ALLOWED

- NEW\_OBJECT\_CLASS\_NAME\_INVALID

- NEW\_OBJECT\_REPRESENTATION\_INVALID

- NEW\_OBJECT\_CONTAINMENT\_INVALID

- NEW\_OBJECTS\_ID\_EXISTS

- NEW\_OBJECTS\_PARENT\_NOT\_FOUND

- NEW\_OBJECT\_ATTRIBUTE\_VALUE\_MISSING

- OBJECTS\_CARDINALITY\_INVALID

- OBJECT\_NOT\_A\_LEAF

- OBJECT\_NOT\_FOUND

- OP\_UNKNOWN

- RESOURCE\_LOCKED

- SERVICE\_LOCKED

- UNKNOWN

example: "UNKNOWN"

detail:

type: string

description: A human-readable explanation specific to this occurrence of the problem.

example: "NRCellDU=1234 already exists in the network"

path:

type: string

example: "/\_3gpp-common-subnetwork:SubNetwork=Irl/\_3gpp-common-mecontext:MeContext=Dublin-1"

errorInfo :

type: object

description: additional error info (e.g. stackdump)

additionalProperties: true

required:

- type

- title

- status

# Use oneOf and discriminator to define the mapping

oneOf:

# Define a specific schema for each 'type' enum value

- $ref: '#/components/schemas/ValidationErrorDetails'

- $ref: '#/components/schemas/RequestObjectTreeMismatchDetails'

- $ref: '#/components/schemas/IENotFoundDetails'

- $ref: '#/components/schemas/ModificationNotAllowedDetails'

- $ref: '#/components/schemas/RetrievalNotAllowedDetails'

- $ref: '#/components/schemas/ServerLimitationDetails'

- $ref: '#/components/schemas/ServiceDisabledDetails'

- $ref: '#/components/schemas/ApplicationLayerErrorDetails'

# Add other specific error types here

discriminator:

propertyName: type # This is the property that determines which oneOf schema applies

mapping:

VALIDATION\_ERROR: '#/components/schemas/ValidationErrorDetails'

REQUEST\_OBJECT\_TREE\_MISMATCH: '#/components/schemas/RequestObjectTreeMismatchDetails'

IE\_NOT\_FOUND: '#/components/schemas/IENotFoundDetails'

MODIFICATION\_NOT\_ALLOWED: '#/components/schemas/ModificationNotAllowedDetails'

RETRIEVAL\_NOT\_ALLOWED: '#/components/schemas/RetrievalNotAllowedDetails'

SERVER\_LIMITATION: '#/components/schemas/ServerLimitationDetails'

SERVICE\_DISABLED: '#/components/schemas/ServiceDisabledDetails'

APPLICATION\_LAYER\_ERROR: '#/components/schemas/ApplicationLayerErrorDetails'

# Individual schemas for each specific error type

ValidationErrorDetails:

type: object

properties:

type:

type: string

enum: [VALIDATION\_ERROR]

title:

type: string

enum: ["Validation Error"]

status:

type: integer

enum: [400]

RequestObjectTreeMismatchDetails:

type: object

properties:

type:

type: string

enum: [REQUEST\_OBJECT\_TREE\_MISMATCH]

title:

type: string

enum: ["Request Object Tree Mismatch"]

status:

type: integer

enum: [422]

IENotFoundDetails:

type: object

properties:

type:

type: string

enum: [IE\_NOT\_FOUND]

title:

type: string

enum: ["Information Element Not Found"]

status:

type: integer

enum: [400]

ModificationNotAllowedDetails:

type: object

properties:

type:

type: string

enum: [MODIFICATION\_NOT\_ALLOWED]

title:

type: string

enum: ["Modification Not Allowed"]

status:

type: integer

enum: [403]

RetrievalNotAllowedDetails:

type: object

properties:

type:

type: string

enum: [RETRIEVAL\_NOT\_ALLOWED]

title:

type: string

enum: ["Retrieval Not Allowed"]

status:

type: integer

enum: [403]

ServerLimitationDetails:

type: object

properties:

type:

type: string

enum: [SERVER\_LIMITATION]

title:

type: string

enum: ["Server Limitation"]

status:

type: integer

enum: [500]

ServiceDisabledDetails:

type: object

properties:

type:

type: string

enum: [SERVICE\_DISABLED]

title:

type: string

enum: ["Service Disabled"]

status:

type: integer

enum: [503]

ApplicationLayerErrorDetails:

type: object

properties:

type:

type: string

enum: [APPLICATION\_LAYER\_ERROR]

title:

type: string

enum: ["Application Layer Error"]

status:

type: integer

enum: [500]

LinkObject:

type: object

description: Defines the structure of a single hypermedia link.

properties:

href:

type: string

format: uri-reference # Use uri-reference for relative paths (allows for absolute or relative uri)

description: The target URI of the link.

templated:

type: boolean

description: Indicates if the href is a URI Template (RFC 6570).

default: true

type:

type: string

description: The content type expected when following this link (MIME type).

title:

type: string

description: A human-readable title that describes the link's purpose.

method:

type: string

enum: [GET, POST, PUT, DELETE, PATCH]

description: The HTTP method to use for this action link.

required:

- href

SelfLink:

type: object

description: Hypermedia links for this resource, including fixed and dynamic relations.

properties:

self:

allOf:

- $ref: '#/components/schemas/LinkObject'

description: A link to the resource itself.

# additionalProperties to allow any other dynamic links

additionalProperties:

$ref: '#/components/schemas/LinkObject' # Any other link will conform to LinkObject schema

required:

- self

example: # demonstrates a typical \_links object in an actual response

self:

href: "{root-url}/ProvMnS/v1/plan-descriptors/pd-001"

templated: true

type: "application/json"

title: "The newly created PlanConfigurationDescriptor"

help:

href: "{root-url}/help-service/v1/topics/plan-descriptors"

templated: true

type: "application/json"

title: "online help for the plan descriptor"

examples:

# Reusable Configuration Change Examples

AddNrCellChange:

summary: Add a new NR Cell (create operation)

value:

operation: create

changeId: add-nr-cell-001

description: Add new NR cell for initial deployment in Dublin-1 area.

target: /\_3gpp-common-subnetwork:SubNetwork=Irl/\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp\_nrm\_managedelement:ManagedElement=1/\_3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1/\_3gpp\_nrm\_nrcelldu:NRCellDU=4

value:

...

UpdateNrCellChange:

summary: Update an existing NR Cell (merge operation)

value:

operation: update

changeId: update-nr-cell-002

description: Update administrativeState for existing NRCellDU=1.

target: /\_3gpp-common-subnetwork:SubNetwork=North/\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp\_nrm\_managedelement:ManagedElement=ENB-A/\_3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1/\_3gpp\_nrm\_nrcelldu:NRCellDU=1

value:

...

RemoveNrCellChange:

summary: Remove an NR Cell (remove operation)

value:

operation: delete

changeId: remove-nr-cell-003

description: Decommission NRCellDU=3.

target: /\_3gpp-common-subnetwork:SubNetwork=Irl/\_3gpp-common-mecontext:MeContext=Dublin-1/\_3gpp\_nrm\_managedelement:ManagedElement=1/\_3gpp-nr-nrm-gnbdufunction:GNBDUFunction=1/\_3gpp\_nrm\_nrcelldu:NRCellDU=3

<CODE ENDS>

\*\*\* END OF CHANGE 1 \*\*\*