



PMSH API Enhancements Release 1

2021/05/31

PM Control Use Cases

UC#	Use Case Name	Description	
UC1	Create PM Subscription	A user creates a new PM subscription and configures it	
UC2	Activate PM Subscription	A user activates an inactive subscription, which triggers the pushing of the corresponding PM configuration to the matching xNF instances	
UC3	Deactivate PM Subscription	A user deactivates an active subscription, which triggers the deletion of the corresponding PM configuration on the matching xNF instances	
UC4	xNF Instance Removed From Network(*)	An xNF instance is removed from ONAP, which triggers the automatic removal of this xNF instance from all subscriptions it is currently included in	ONAP Rel F Scope
UC5	Delete PM Subscription	A user deletes a subscription	
UC6	New xNF Instance Added To Network(*)	A new xNF instance is added in ONAP, which triggers the automatic adding of this xNF instance to any subscription with a matching xNF filter	
UC7	Support for multiple NF config blueprints	Support for mixed NF type and/or versions.	ONAP Rel G
UC8	Configure xNF filter in PM Subscription (<i>Extended filtering attributes</i>)	A user configures the xNF filter in a subscription (added new filter attribs)	ONAP Rel H
UC9	Update Filter	Update filter attributes, e.g. Add or remove xNF instances.	ONAP Rel I
UC10	Modify Measurement Parameters	A user edits the subscription measurement parameters.	REQ-715
NN	JobId/Filter Enhancements		

PMSH Add Public API – Rel I

- Currently, there is no complete API to support all PMSH operations
 - only a limited API with ability to retrieve subscriptions and perform health check
- This limits the use of PMSH in other projects and components which wish to manage NF performance subscription
 - policy updates and subscription events are available, but awkward to implement
- PMSH API will support the following “use cases”:
 - Subscription read (i.e. get list of subscriptions; get specific subscription)
 - Subscription creation
 - Subscription update (i.e. update parms, e.g. measurement group data)
 - Subscription deletion
 - Filter read (i.e. get filter criteria; get list of NFs to which subscription applies)
 - Filter update
 - Filter read scope (i.e. get list of NFs which match current filter)
 - Admin state update (i.e. lock/unlock subscription)
- API enhancements require architectural updates to PMSH. Some behaviour will be modified from previous versions, such as:
 - adding support for multiple subscriptions in PMSH
 - adding support for multiple measurement groups per subscription
 - update/overwrite behaviour via API

PMSH API – Model

- Different PM Use Cases require different subscription updates
 - e.g. update filter, update measurements
- The current PMSH subscription could be viewed as three different “objects”, by parameter grouping ([see slide](#)):
 - Subscription
 - NF Filter
 - Measurement Group
- **but** we want to avoid defining specific objects and operations for each, so the “subscription” object will be used in all cases; the difference is which parameters are used for which use cases
 - i.e. generic CRUD interface, with same object type for all use cases

PMSH Schema (Rel I)

Subscription

Field	Description	Type	Required	Values
subscriptionName	Name of the subscription.	string	True	
administrativeState	Setting a subscription to UNLOCKED will apply the subscription to the NF instances immediately. If it is set to LOCKED, it will not be applied until it is later unlocked.	enum	True	LOCKED (default), UNLOCKED
fileBasedGP	The frequency at which measurements are produced.	Int		
fileLocation	Location of Report Output Period file.	String		
measurementGroups	List containing measurementGroup.	Measurement Group		
nfFilter	The network function filter will be used to filter the list of nf's stored in A&AI to produce a subset.	NFFilter		

New attribute to identify measurement groups (e.g. jobId) within the subscription.

NFFilter

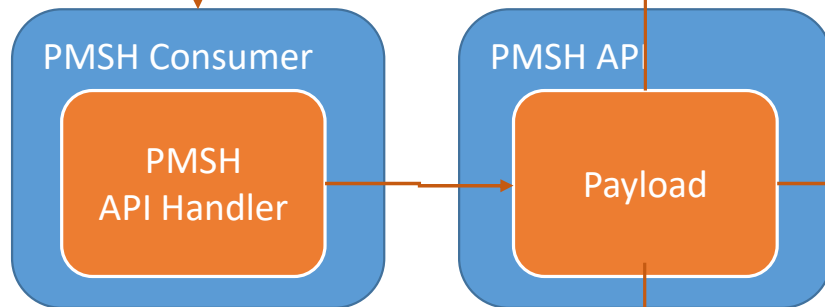
Field	Description	Type	Required
nfNames	List of NF names. These names are regexes, which will be parsed by the PMSH.	list	False
modelInvariantIDs	List of modelInvariantIDs. These UUIDs will be checked for exact matches with AAI entities.	list	False
modelVersionIDs	List of modelVersionIDs. These IDs will be checked for exact matches with AAI entities.	list	False

MeasurementGroup

Field	Description	Type	Required
measurementTypes	List of measurement types. These are regexes, and it is expected that either the CDS blueprint, or NF can parse them. As the PMSH will not do so.	list	True
managedObjectDNs Basic	List of managed object distinguished names.	List	False
name	Name of the measurement group.	String	True

PMSH API: Use Cases

PMSH Consumer (e.g. CLAMP) provides required parameters (“payload”) per Use Case via PMSH API.



Use Case: create/update subscription:

- **subscriptionName**
- **administrativeState**
- fileBasedGP
- fileLocation
- Parameters-nfFilter
 - nfNames
 - modelInvariantIDs
 - modelVersionIDs
- Parameters-measurementGroups
 - measurementTypes
 - managedObjectDNsBasic

Use Case: update reporting

- fileLocation
- fileBasedGP

Use Case: update admin state

- administrativeState

Use Case: update filter

- nfNames
- modelInvariantIDs
- modelVersionIDs

Use Case: update measurement groups

- name
- measurementTypes
- managedObjectDNsBasic

PMSH API – resource definitions

- “Generic” endpoint for subscription objects with get and put operations for “subscription data”
 - PMSH JSON object
- Priority for Rel I is ability to manage “complete” subscription objects
 - /pms/subscriptions
 - /pms/subscriptions/{subscriptionName} (by subscription name/id)
 - /pms/subscriptions/{subscriptionName}/adminState (change admin state)
 - /pms/healthcheck (service status)
- Finer grain attribute management is stretch goal
 - /pms/subscriptions/{subscriptionName}/fileBasedGP
 - /pms/subscriptions/{subscriptionName}/fileLocation
 - /pms/subscriptions/{subscriptionName}/nfFilter
 - /pms/subscriptions/{subscriptionName}/nfFilter/nfNames
 - /pms/subscriptions/{subscriptionName}/nfFilter/nfinvariantModelIds
 - /pms/subscriptions/{subscriptionName}/nfFilter/modelVersionIds
 - /pms/subscriptions/{subscriptionName}/measurementGroups
 - /pms/subscriptions/{subscriptionName}/measurementGroups/{measurementGroupName}/name (by meas group name/id)
 - /pms/subscriptions/{subscriptionName}/measurementGroups/{measurementGroupName}/measurementTypes
 - /pms/subscriptions/{subscriptionName}/measurementGroups/{measurementGroupName}/managedObjectDNsBasic

Subscription Object Payload

- The expected format for subscriptions in PMSH Rel I
 - [Subscription Schema](#)
 - [Example Subscription Object](#)
- There are several required fields in the schema, used as follows for:
 - Create: all need to be present (or default is used)
 - Read: requires subscriptionName
 - Update: requires subscriptionName; if other required fields are omitted they are unchanged on the NF
 - Delete: requires subscriptionName

PMSH object examples

PMSH Subscription (Initial):

```
"subscription": {
  "subscriptionName": "test123",
  "administrativeState": "UNLOCKED",
  "fileBasedGP": 15,
  "fileLocation": "/pm/pm.xml",
  "measurementGroups": [{
    "measurementGroupA": {
      "managedObjectDNsBasic": [{
        "DN": "dna"
      }],
      "measurementTypes": [{
        "measurementType": "countera"
      }]
    }
  ]
},
"nfFilter": {
  "modelInvariantIDs": [],
  "modelName": [],
  "modelVersionIDs": [],
  "nfNames": [
    "^pnf.*",
    "^vnf.*"
  ]
}}
```

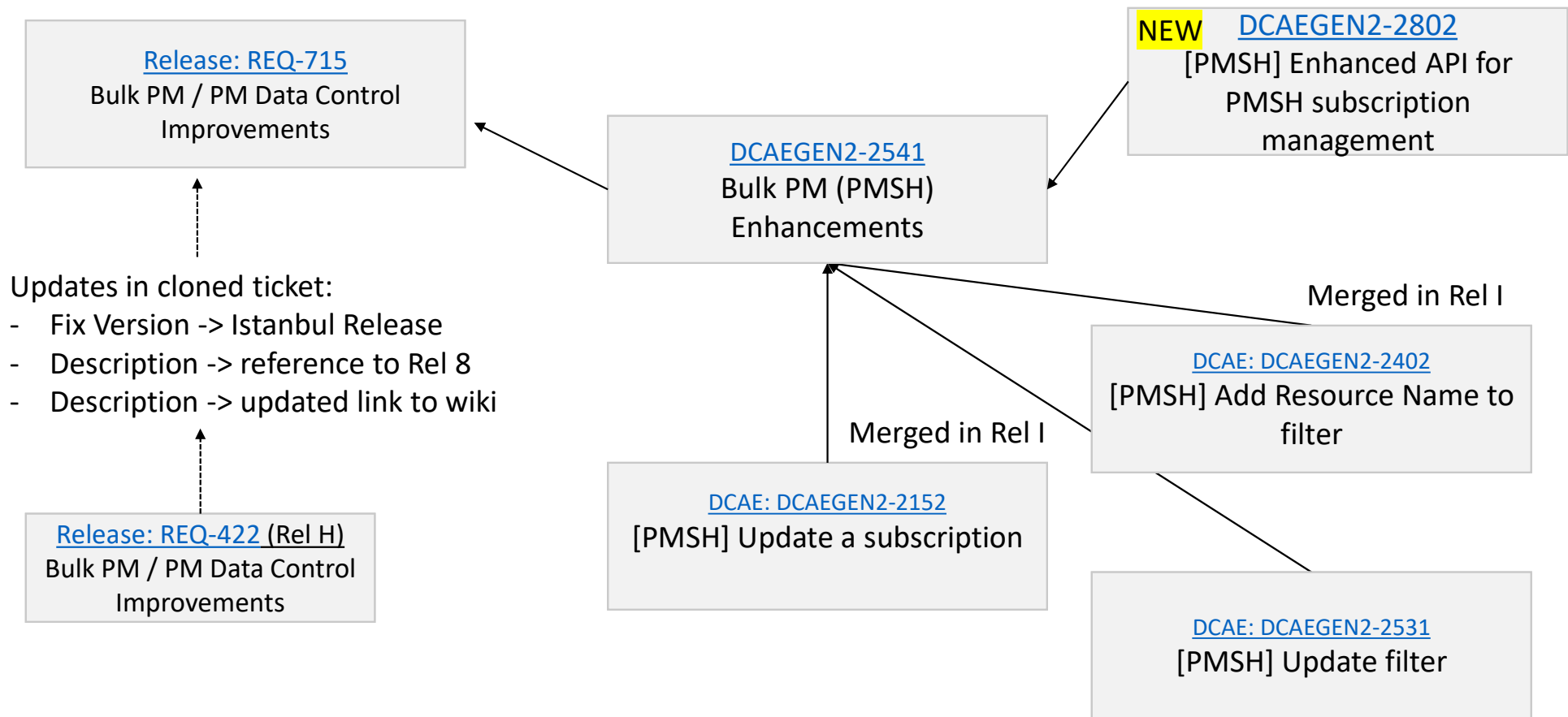
Subscription update (measurement group change):

```
"subscription": {
  "subscriptionName": "test123",
  "administrativeState": "UNLOCKED",
  "measurementGroups": [{
    "measurementGroupA": {
      "managedObjectDNsBasic": [{
        "DN": "dna"
      }],
      "measurementTypes": [{
        "measurementType": "COUNTERB"
      }]
    }
  ]
}
```

Subscription after update:

```
"subscription": {
  "subscriptionName": "test123",
  "administrativeState": "UNLOCKED",
  "fileBasedGP": 15,
  "fileLocation": "/pm/pm.xml",
  "measurementGroups": [{
    "measurementGroupA": {
      "managedObjectDNsBasic": [{
        "DN": "dna"
      }],
      "measurementTypes": [{
        "measurementType": "COUNTERB"
      }]
    }
  ]
},
"nfFilter": {
  "modelInvariantIDs": [],
  "modelName": [],
  "modelVersionIDs": [],
  "nfNames": [
    "^pnf.*",
    "^vnf.*"
  ]
}}
```

PMSH JIRA (Rel I Content)



PMSH – API Enhancements, Rel I



ONAP

OPEN NETWORK AUTOMATION PLATFORM

