ONAP Application Controller (APPC) API Guide

Revision **Version 2.0.0**

Revision Date **25 August 2017**

Copyright © 2017 AT&T Intellectual Property.
Copyright © 2017 Amdocs
All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License. You may obtain a copy of the License at

https://creativecommons.org/licenses/by/4.0/

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Date	Revision	Author	Changes
2017-08-25	2.0.0	Paul Miller	Updates for software contribution AT&T Release 17.10
2017-01-17	1.0.0	mjf	“OpenECOMP”, cosmetic changes
2017-02-06	1.0.0	mjf	copyright updated

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Table of Contents

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

1 Introduction

This guide describes the APPC API that allows you to manage and control the life cycle of controlled virtual network functions (VNFs).

Target Audience

This document is intended for an advanced technical audience, such as the engineers or architects who need to use this guide to develop an interfacing application. The guide assumes a knowledge of the Open Network Automation Platform (ONAP) components and features, and familiarity with JSON notation.

Conventions

Convention	Description

`Monospace`	This font indicates sample codes, screenshots, or elements. For example:
	`"contact": {`
	`  "contactType": "USER",`
	`  "source": "app1",`
	`}`

Italics	▪ Emphasizes a point or denotes new terms at the place where they are defined in the text. ▪ Indicates an external book title reference.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

2 Life Cycle Management Commands

APPC receives commands from external ONAP components or Portals to manage the life cycle of virtual applications and their components.

A virtual application is composed of the following layers of network technology:

- Virtual Network Function (VNF)
- Virtual Network Function Component (VNFC)
- Virtual Machine (VM)

A Life Cycle Management (LCM) command may affect one or more of these layers.

An LCM command is sent as a request to the APPC using an HTTP POST request or in a message on a message bus (for example, DMaaP). A request may result in either a single synchronous response or multiple asynchronous responses:

- An **asynchronous** command, which is sent as an authorized and valid request, results in at least two discrete response events:
    - an accept response, to indicate that the request is accepted for processing
    - a final response to indicate the status and outcome of the request processing
        - An unauthorized or invalid request results in a single ERROR response.
- A **synchronous** command, such as Lock or Unlock, results in a single response that is either SUCCESS or ERROR.

  NOTE: For both asynchronous or synchronous commands, the first response is always returned using the same transport that the initial action used. For example, if the action request was via the message bus (such as when it originates from Policy), then the response is also via the message bus. However, if the request was via a direct HTTP call, the response is similarly a synchronous HTTP response.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Message Bus and the LCM API Client Library

The recommended approach for sending/receiving requests to APPC is via the message bus. To support this approach, an APPC client library is available and should be used. The client library aims to provide consumers of APPC capabilities with a strongly-typed Java interface and to encapsulate the actual interaction with APPC component via the message bus.

For more details, see the APPC Client Library Guide at:

(INSERT LINK)

The client library supports both synchronous and asynchronous flows as follows.

Asynchronous Flow

- The APPC Client Library is called via an asynchronous API using a full command object, which is mapped to a JSON representation.
- The APPC client calls the message bus client and sends the JSON command to a configured topic.
- The APPC client pulls response messages from the configured topic.
- On receiving the response for the command, APPC client runs the relevant callback method of the consumer ResponseHandler.

Synchronous Flow

- The APPC Client Library is called via a synchronous API using a full command object, which is mapped to a JSON representation.
- The APPC client calls the message bus client and sends the JSON command to a configured topic.
- The APPC client pulls response messages from the configured topic.
- On receiving the *final* response for the command, the APPC client returns the response object with a final status.

The client library adds the following wrapper around request and responses to the LCM API (described below).

**Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.**

```
{
"version" : "2.0",
"cambria.partition" : "<TOPIC>",
"correlation-id" :"<CORRELATION_ID>",
"rpc-name" : "<RPC_NME>",
"type" : <MESSAGE_TYPE>
"body" : <RPC_SPECIFIC_BODY>
}
```

Table 1 Request / Response Message Fields

Field	Description	Required?
version	Indicates the version of the message bus protocol with APPC. Version 2.0 should be used.	Yes
cambria. partition	Indicates the specific topic partition that the message is intended for. For example: ■ For incoming messages, this value should be `APPC`.	No
correlation-id	Correlation ID used for associating responses in APPC Client Library. Built as: <request-id>-<sub-request-id>	Yes
rpc-name	The target Remote Processing Call (RPC) name which should match the LCM command name. For example: `configure`	Yes
type	Message type: request, response or error	Yes
body	Contains the input or output LCM command content, which is either the request or response body. The body field format is identical to the equivalent HTTP Rest API command based on the specific RPC name. ~~For~~	Yes

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

GENERIC REQUEST FORMAT

The LCM API general request format is applicable for both POST HTTP API and for the message body received via the message bus.

LCM Request

The LCM request comprises a common header and a section containing the details of the LCM action.

The LCM request conforms to the following structure:

```
{
"input": {
"common-header": {"timestamp": "<TIMESTAMP>",
"api-ver": "<API_VERSION>",
"originator-id": "<ECOMP_SYSTEM_ID>",
"request-id": "<ECOMP_REQUEST_ID>",
"sub-request-id": "<ECOMP_SUBREQUEST_ID>",
"flags": {
"mode": "<EXCLUSIVE|NORMAL>",
"force": "<TRUE|FALSE>",
"ttl": "<TTL_VALUE>"
}
},
"action": "<COMMAND_ACTION>",
"action-identifiers": {
"vnf-id": "<ECOMP_VNF_ID>",
"vnfc-name": "<ECOMP_VNFC_NAME>",
"vserver-id": "VSERVER_ID"
},
["payload": "<PAYLOAD>"]
}
```

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Table 2 LCM Request Fields

Field	Description	Required?
input	The block that defines the details of the input to the command processing. Contains the common-header details.	Yes
common- header	The block that contains the generic details about a request.	Yes
timestamp	The time of the request, in ISO 8601 format, ZULU offset. For example: `2016-08-03T08:50:18.97Z`. APPC will reject the request if timestamp is in the future (due to clock error), or timestamp is too old (compared to TTL flag)	Yes
api-ver	Identifies the API version, in X.YY format, where X denotes the major version increased with each APPC release, and YY is the minor release version. For example: ■ `5.00` for this version	Yes
originator-id	An identifier of the calling system limited to a length of 40 characters. It can be used for addressing purposes, such as to return an asynchronous response to the correct destination, in particular where there are multiple consumers of APPC APIs.	Yes
request-id	The UUID for the request ID, limited to a length of 40 characters. The unique OSS/BSS identifier for the request ID that triggers the current LCM action. Multiple API calls can be made with the same request-id. The request-id is stored throughout the operations performed during a single request.	Yes
sub-request-id	Uniquely identifies a specific LCM or control action, limited to a length of 40 characters. Persists throughout the life cycle of a single request.	No
flags	Generic flags that apply to all LCM actions: ■ "MODE" : • "EXCLUSIVE" - accept no queued requests on this VNF while processing, or • "NORMAL" - queue other requests until complete ■ "FORCE" : "TRUE"|"FALSE" - run action even if target is in an unstable state (for example, if VNF is busy processing another LCM command or if a previous command failed and VNF was indicated as not STABLE), or not. The specific behavior of forced actions varies, but implies cancellation of the previous action and an override by the new action. The default value is FALSE. Force flag are used to bypass APPC’s working state management for the VNF(VNF	No

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Field	Description	Required?
	working State Management) : APPC maintains working state (in the VNF_STATE_MANAGEMENT table present in the APPC-DB) for the VNF depending on the last action performed on it: There are below 3 states appc have for VNF while performing non-read only operation (Read-Only operations are : Lock, Unlock, CheckLock, Sync, Audit etc.) : 1) Stable – If the last action performed on a VNF is Successful (returning Success). 2) Unstable – This is the intermediate state for any VNF on which operation is being performed. 3) Unknown – This is the status when the last action performed on a VNF is not successful. APPC have validation that it will not allow any operations on VNF which is in Unstable or Unknown state. To skip this check end-user can pass Force-flag=true in the request. ▪ "TTL": <0....N> - The timeout value for the action to run, between action received by APPC and action initiated. If no TTL value provided, the default/configurable TTL value is to be used.	
action	The action to be taken by APPC, for example: Test, Start, Terminate. ***NOTE:*** *The specific value for the action parameter is provided for each* command.	Yes
action- identifiers	A block containing the action arguments. These are used to specify the object upon which APPC LCM command is to operate. At least one action-identifier must be specified (note that vnf-id is mandatory). For actions that are at the VM level, the action-identifiers provided would be vnf-id and vserver-id.	Yes
vnf-id	Identifies the VNF instance to which this action is to be applied. Required for actions.	Yes
vnfc-name	Identifies the VNFC instance to which this action is to be applied. Required if the action applied to a specific VNFC.	No
vserver-id	Identifies a specific VM instance to which this action is to be applied. Required if the action applied to a specific VM. (Populate the vserver-id field with the UUID of the VM)	No
vf-module-id	Identifies a specific VF module to which this action is to be applied. Required if the action applied to a specific VF module.	No

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Field	Description	Required?
payload	An action-specific open-format field. The payload can be any valid JSON string value. JSON escape characters need to be added when an inner JSON string is included within the payload, for example: `"{\" vnf -host-ip -address\": \"<VNF-HOST-IP-ADDRESS>\"}"`. The payload is typically used to provide parametric data associated with the command, such as a list of configuration parameters. Note that not all LCM commands need have a payload. ***NOTE:*** *See discussion below on the use of payloads for self-service actions.*	No

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Generic Response Format

This section describes the generic response format.

The response format is applicable for both POST HTTP API and for the message body received via the message bus.

LCM Response

The LCM response comprises a common header and a section containing the payload and action details.

The LCM response conforms to the following structure:

```
{
"output": {
"common-header": {
"api-ver": "<API_VERSION>",
"flags": {
"ttl": <TTL_VALUE>,
"force": "<TRUE|FALSE>",
"mode": "<EXCLUSIVE|NORMAL>"
},
"originator-id": "<ECOMP_SYSTEM_ID>",
"request-id": "<ECOMP_REQUEST_ID>",
"sub-request-id": "<ECOMP_SUBREQUEST_ID>",
"timestamp": "2016-08-08T23:09:00.11Z",
},
"payload": "<PAYLOAD>",
[Additional fields],
"status": {
"code": <RESULT_CODE>,
"message": "<RESULT_MESSAGE>"
}
}
}
```

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Table 3 LCM Response Fields

Field	Description	Required?
output	The block that defines the details of the output of the command processing. Contains the `common-header` details.	Yes
common-header	The block that contains the generic details about a request.	Yes
api-ver	Identifies the API version, in X.YY format, where X denotes the major version increased with each APPC release, and YY is the minor release version. For example: • `5.00` for this version	Yes
originator-id	An identifier of the calling system limited to a length of 40 characters. It can be used for addressing purposes, such as to return an asynchronous response to the correct destination, in particular where there are multiple consumers of APPC APIs.	Yes
request-id	The UUID for the request ID, limited to a length of 40 characters. The unique OSS/BSS identifier for the request ID that triggers the current LCM action. Multiple API calls can be made with the same request-id. The request-id is stored throughout the operations performed during a single request.	Yes
sub-request-id	Uniquely identifies a specific LCM or control action, limited to a length of 40 characters. Persists throughout the life cycle of a single request.	No
timestamp	The time of the request, in ISO 8601 format, ZULU offset. For example: `2016-08-03T08:50:18.97Z`.	Yes
status	The status describes the outcome of the command processing. Contains a `code` and a `message` providing success or failure details. ***NOTE:*** *See* status *for code values.*	Yes
payload	An open-format field. The payload can be any valid JSON string value. JSON escape characters need to be added when an inner JSON string is included within the payload, for example: `"{\"upload_config_id\": \"<value\"}"`. The payload is typically used to provide parametric data associated with the response to the command. Note that not all LCM commands need have a payload. ***NOTE:*** *The specific value(s) for the response payload, where relevant, is provided for in each* command *description.*	No
[Field name]	Additional fields can be provided in the response, if needed, by specific commands.	No
code	A unique pre-defined value that identifies the exact nature of the success or failure status.	No
message	The description of the success or failure status.	No

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Status Codes

The status code is returned in the response message as the `code` parameter, and the description as the `message` parameter

The different responses are categorized as follows:

ACCEPTED

Request is valid and accepted for processing.

ERROR

Request invalid or incomplete.

REJECT

Request rejected during processing due to invalid data, such as an unsupported command or a non-existent service-instance-id.

SUCCESS

Request is valid and completes successfully.

FAILURE

The request processing resulted in failure.

A FAILURE response is always returned asynchronously via the message bus.

PARTIAL SUCCESS

The request processing resulted in partial success where at least one step in a longer process completed successfully.

A PARTIAL SUCCESS response is always returned asynchronously via the message bus.

PARTIAL FAILURE

The request processing resulted in partial failure.

A PARTIAL FAILURE response is always returned asynchronously via the message bus.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Category	Code	Message / Description
ACCEPTED	100	ACCEPTED - Request accepted
ERROR	200	UNEXPECTED ERROR - ${detailedErrorMsg}
REJECT	300	REJECTED - ${detailedErrorMsg}
	301	INVALID INPUT PARAMETER -${detailedErrorMsg}
	302	MISSING MANDATORY PARAMETER - Parameter ${paramName} is missing
	303	REQUEST PARSING FAILED - ${detailedErrorMsg}
	304	NO TRANSITION DEFINED - No Transition Defined for ${actionName} action and ${currentState} state
	305	ACTION NOT SUPPORTED - ${actionName} action is not supported
	306	VNF NOT FOUND - VNF with ID ${vnfId} was not found
	307	DG WORKFLOW NOT FOUND - No DG workflow found for the combination of ${dgModule} module ${dgName} name and ${dgVersion} version
	308	WORKFLOW NOT FOUND - No workflow found for VNF type ${vnfTypeVersion} and ${actionName} action
	309	UNSTABLE VNF - VNF ${vnfId} is not stable to accept the command
	310	LOCKING FAILURE -${detailedErrorMsg}
	311	EXPIREDREQUEST. The request processing time exceeded the maximum available time
	312	DUPLICATEREQUEST. The request already exists
	313	MISSING VNF DATA IN A&AI - ${attributeName} not found for VNF ID = ${vnfId}
SUCCESS	400	The request was processed successfully
FAILURE	401	DG FAILURE - ${ detailedErrorMsg }
	402	NO TRANSITION DEFINED - No Transition Defined for ${ actionName} action and ${currentState} state
	403	UPDATE_AAI_FAILURE - failed to update AAI. ${errorMsg}
	404	EXPIRED REQUEST FAILURE - failed during processing because TTL expired
	405	UNEXPECTED FAILURE - ${detailedErrorMsg}
	406	UNSTABLE VNF FAILURE - VNF ${vnfId} is not stable to accept the command
	450	Requested action is not supported on the VNF
PARTIAL SUCCESS	500	PARTIAL SUCCESS
PARTIAL FAILURE	501 - 599	PARTIAL FAILURE

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Malformed Message Response

A malformed message is an invalid request based on the LCM API YANG scheme specification. APPC rejects malformed requests as implemented by ODL infrastructure level.

Response Format for Malformed Requests

```
{
"errors": {
"error": [
{
"error-type": "protocol",
"error-tag": "malformed-message",
"error-message": "<ERROR-MESSAGE>",
"error-info": "<ERROR-INFO>"
}
]
}
}
```

Example Response

```
{
"errors": {
"error": [
{
"error-type": "protocol",
"error-tag": "malformed-message",
       "error-message": "Error parsing input: Invalid value 'Stopp' for enum type. Allowed values are: [Sync, Audit, Stop, Terminate]",
       "error-info": "java.lang.IllegalArgumentException: Invalid value 'Stopp' for enum type. Allowed values are: [Sync, Audit, Stop, Terminate]..."
}
]
}
}
```

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

3 API Scope

Defines the level at which the LCM command operates for the current release of APPC and the VNF types which are supported for each command.

Commands, or actions, can be performed at one or more of the following scope levels:

VNF	Commands can be applied at the level of a specific VNF instance using the vnf-id.
VF-Module	Commands can be applied at the level of a specific VF-Module using the vf-module-id.
VNFC	Commands can be applied at the level of a specific VNFC instance using a vnfc-name.
VM	Commands can be applied at the level of a specific VM instance using a vserver-id.

VNF's Types Supported

Commands, or actions, may be currently supported on all VNF types or a limited set of VNF types. Note that the intent in the 1710 release is to support all actions on all VNF types which have been successfully onboarded in a self-service mode.

Any - Currently supported on any vnf-type.

Any (requires self-service onboarding) – Currently supported on any vnf-type which has been onboarded using the APPC self-service onboarding process. See further discussion on self-service onboarding below.

**Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.**

Command	VNF	VF-Module	VNFC	VM	VNF/VM Types Supported
Audit	Yes				Any (requires self-service onboarding)
CheckLock	Yes				Any (APPC internal command)
Configure	Yes		Yes		Any (requires self-service onboarding)
ConfigModify	Yes		Yes		Any (requires self-service onboarding)
ConfigBackup	Yes				Chef and Ansible only (requires self-service onboarding)
ConfigRestore	Yes				Chef and Ansible only (requires self-service onboarding)
Evacuate				Yes	Any (uses OpenStack Evacuate command)
HealthCheck	Yes				Any (requires self-service onboarding)
Lock	Yes				Any (APPC internal command)
Migrate				Yes	Any (uses OpenStack Migrate command)
Rebuild				Yes	Any (uses OpenStack Rebuild command)
Restart	Yes			Yes	Any (uses OpenStack Start and Stop commands)
Snapshot				Yes	Any (uses OpenStack Snapshot command)
Start	Yes	Yes		Yes	Any (uses OpenStack Start command)
StartApplication	Yes				Chef and Ansible only (requires self-service onboarding)
Stop	Yes	Yes		Yes	Any (uses OpenStack Stop command)
StopApplication	Yes				Chef and Ansible only (requires self-service onboarding)
Sync	Yes				Any (requires self-service onboarding)
Unlock	Yes				Any (APPC internal command)

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Self-Service VNF Onboarding

The APPC architecture is designed for VNF self-service onboarding (i.e., a VNF owner or vendor through the use of tools can enable a new VNF to support the LCM API actions that are designate as self-service). The VNF must support one or more of the following interface protocols:

- Netconf with uploadable Yang model (requires a Netconf server running on the VNF)
- Chef (requires a Chef client running on the VNF)
- Ansible (does not require any changes to the VNF software)

The self-service onboarding process is done using an APPC Design GUI which interacts with an APPC instance which is dedicated to self-service onboarding. The steps in the onboarding process using the APPC Design GUI are:

- Define the VNF capabilities (set of actions that the VNF can support).
- Create a template and parameter definitions for actions which use the Netconf, Chef, or Ansible protocols. The template is an xml or JSON block which defines the "payload" which is included in the request that is downloaded the VNF (if Netconf) or Chef/Ansible server.
- Test actions which have templates/parameter definitions.
- Upload the VNF definition, template, and parameter definition artifacts to SDC which distributes them to all APPC instances in the same environment (e.g., production).

For more details, see the APPC Self-Service VNF Onboarding Guide (add link).

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

4 LCM Commands

The LCM commands that are valid for the current release.

Audit

The Audit command compares the configuration of the VNF associated with the current request against the most recent configuration that is stored in APPC's configuration database.

A successful Audit means that the current VNF configuration matches the latest APPC stored configuration.

A failed Audit indicates that the configurations do not match.

This command can be applied to any VNF type. The only restriction is that the VNF has been onboarded in self-service mode (which requires that the VNF supports a request to return the running configuration).

The Audit action does not require any payload parameters.

NOTE: Audit does not return a payload containing details of the comparison, only the Success/Failure status.

Target URL	/restconf /operations/ appc-provider-lcm:audit
Action	Audit
Action-Identifiers	vnf-id
Payload Parameters	See below
Revision History	Unchanged in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Parameter	Description	Required?	Example
publish-config	* If the publish_config field is set to Y in the payload, then always send the running configuration from the VNF using the Data Router * If the publish_config field is set to N in the payload, then: - If the result of the audit is ‘match’ (latest APPC config and the running config match), do not send the running configuration in the Data Router - If the result of the audit is ‘no match’, then send the running configuration on the Data Router	Yes	"publish-config": "<Y|N>"

Audit Response

The audit response returns an indication of success or failure of the audit. If a new configuration is uploaded to the APPC database, the payload contains the ‘upload_config_id’ and values for any records created. In addition, the configuration is sent to the ONAP Data Router bus which may be received by an external configuration storage system.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

CheckLock

The CheckLock command returns true if the specified VNF is locked; otherwise, false is returned.

A CheckLock command is deemed successful if the processing completes without error, whether the VNF is locked or not. The command returns only a single response with a final status.

Note that APPC locks the target VNF during any VNF command processing, so a VNF can have a locked status even if no Lock command has been explicitly called.

The CheckLock command returns a specific response structure that extends the default LCM response.

The CheckLock action does not require any payload parameters.

Target URL	/restconf/operations/appc-provider-lcm:checklock
Action	CheckLock
Action-Identifiers	vnf-id
Payload Parameters	None
Revision History	Unchanged in this version.

CheckLock Response

The CheckLock command returns a customized version of the LCM response.

Parameter	Description	Required?	Example
locked	"TRUE"|"FALSE" - returns TRUE if the specified VNF is locked, otherwise FALSE.	No	`"locked": "<TRUE|FALSE>"`

Example

```
{
  "output": {
    "status": {
```

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

```
    "code": <RESULT_CODE>, "message":
    "<RESULT_MESSAGE>"
  },
  "common-header": {
    "api-ver": "<API_VERSION>",
    "request-id": "<ECOMP_REQUEST_ID>", "originator-
    id": "<ECOMP_SYSTEM_ID>",
    "sub-request-id": "<ECOMP_SUBREQUEST_ID>",
    "timestamp": "2016-08-08T23:09:00.11Z",
    "flags": {
      "ttl": <TTL_VALUE>, "force":
      "<TRUE|FALSE>",
      "mode": "<EXCLUSIVE|NORMAL>"
    }
  },
  "locked": "<TRUE|FALSE>"
}
```

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Configure

Configure a VNF or a VNFC on the VNF after instantiation.

A set of configuration parameter values specified in the configuration template is included in the request. Other configuration parameter values may be obtained from an external system ~~(such as INSTAR).~~

A successful Configure request returns a success response.

A failed Configure action returns a failure response and the specific failure messages in the response block.

Target URL	/restconf/operations/appc-provider-lcm:configure
Action	Configure
Action-Identifiers	vnf-id
Payload Parameters	See below
Revision History	Unchanged in this version.

Payload Parameter	**Description**	**Required?**	**Example**

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF, if Netconf. A template-name may also be included in the event that a specific configuration template needs to be identified. If the request is vnfc-specific, the vnfc-type must be included.	Yes	"payload":
			"{\"request-parameters
			\": {
			\"host-ip-address\":
			\”value\”,
			\”vnfc-type\”: \”value\”’,
			\”template-name\”: \”name\”
			}

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

configuration-parameters	A set of instance specific configuration parameters should be specified. If provided, APPC replaces variables in the configuration template with the values supplied.	No	\"configuration-parameters\": {\"<CONFIG- PARAMS>\"}

Configure Response

The Configure response returns an indication of success or failure of the request. If successful, the return payload contains the ‘upload_config_id’ and values for any records created.   In addition, the configuration is sent to the ONAP Data Router bus which may be received by an external configuration storage system.

In 1710, ~~M~~SO is creating the VNFC records in A&AI. APPC is updating the VNFC status.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

ConfigModify

Modifies the configuration on a VNF or VNFC in service.

A successful ConfigModify request returns a success response.

A failed ConfigModify action returns a failure response code and the specific failure message in the response block.

NOTE: See also Configure

Target URL	/restconf/operations/appc-provider-lcm:configmodify
Action	ConfigModify
Action-Identifiers	Vnf-id
Payload Parameters	See below
Revision History	Unchanged in this version.

Payload Parameter	**Description**	**Required?**	**Example**

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF, if Netconf. A template-name may also be included in the event that a specific configuration template needs to be identified. If the request is vnfc-specific, the vnfc-type must be included.	Yes	`"payload":`
			`  "{\"request-parameters`
			`\": {`
			`\"host-ip-address\":`
			`\”value\”,`
			`\”vnfc-type\”: \”value\”’`
			`\”template-name\”: \”name\”,`
			`}`

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

configuration-parameters	A set of instance specific configuration parameters should be specified. If provided, APPC replaces variables in the configuration template with the values supplied.	No	`\"configuration-parameters\": {\"<CONFIG- PARAMS>\"}`

If successful, this request returns a success response.

A failed Configure action returns a failure response and the specific failure message in the response block.

ConfigModify Response

The ConfigModify response returns an indication of success or failure of the request. If successful, the return payload contains the ‘upload_config_id’ and values for any records created. In addition, the configuration is sent to the ONAP Data Router bus which may be received by an external configuration storage system.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

ConfigBackup

Stores the current VNF configuration on a local file system (not in APPC). This is limited to Ansible and Chef. There can only be one stored configuration (if there is a previously saved configuration, it is replaced with the current VNF configuration).

A successful ConfigBackup request returns a success response.

A failed ConfigBackup action returns a failure response code and the specific failure message in the response block.

Target URL	/restconf/operations/appc-provider-lcm:configbackup
Action	ConfigBackup
Action-Identifiers	Vnf-id
Payload Parameters	See below
Revision History	New in this version.

Payload Parameter	Description	Required?	Example

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF (for Chef and Ansible, this will be the url to connect to the server).	Yes	"payload":
			"{\"request-
			parameters
			\": {
			\"host-ip-address\":
			\”value\”
			}
			\"configuration-
			parameters\":
			{\"<CONFIG-
			PARAMS>\"}

configuration-parameters	A set of instance specific configuration parameters should be specified, as required by the Chef cookbook or Ansible playbook.	No	

ConfigBackup Response

The ConfigBackup response returns an indication of success or failure of the request.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

ConfigRestore

Applies a previously saved configuration to the active VNF configuration. This is limited to Ansible and Chef. There can only be one stored configuration.

A successful ConfigRestore request returns a success response.

A failed ConfigRestore action returns a failure response code and the specific failure message in the response block.

Target URL	/restconf/operations/appc-provider-lcm:configrestore
Action	ConfigRestore
Action-Identifiers	Vnf-id
Payload Parameters	request-parameters, configuration-parameters
Revision History	New in this version.

Parameter	**Description**	**Required?**	**Example**

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF (for Chef and Ansible, this will be the url to connect to the server).	Yes	`"payload":`
			`  "{\"request-parameters`
			`\": {`
			`\"host-ip-address\":`
			`\”value\”`
			`}`
			
			`\"configuration-`
			`parameters\":`
			`{\"<CONFIG-`
			`PARAMS>\"}`

configuration-parameters	A set of instance specific configuration parameters should be specified, as required by the Chef cookbook or Ansible playbook.	No	

ConfigRestore Response

The ConfigRestore response returns an indication of success or failure of the request.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Evacuate

Evacuates a specified VM from its current host to another. After a successful evacuate, a rebuild VM is performed if a snapshot is available (and the VM boots from a snapshot.

The host on which the VM resides needs to be down.
If the node is not specified in the request, it will be selected by relying on internal rules to evacuate.
The Evacuate action will fail if the specified target host is not UP/ENABLED.

After Evacuate, the rebuild VM can be disabled by setting the optional rebuild-vm parameter to false.

A successful Evacuate action returns a success response. A failed Evacuate action returns a failure.

NOTE: The command implementation is based on Openstack functionality.
For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:evacuate
Action	Evacuate
Action-identifiers	Vnf-id, vserver-id
Payload Parameters	vm-id, identity-url, tenant-id, rebuild-vm, targethost-id
Revision History	Unchanged in this version.

Parameter	Description	Required?	Example
vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	"payload": "{\"vm-id\": \"<VM-ID> \", \"identity-url\": \"<IDENTITY-URL>\", \"tenant-id\": \"<TENANT-ID> \", \"rebuild-vm\": \"false\", \"targethost-id\": \"nodeblade7\"}"
identity-url	The identity URL used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

rebuild-vm	A boolean flag indicating if a Rebuild is to be performed after an Evacuate. The default action is to do a Rebuild. It can be switched off by setting the flag to "false".	No	
targethost-id	A target hostname indicating the host the VM is evacuated to. By default, the cloud determines the target host.	No	

**Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.**

HealthCheck

This command runs a VNF health check and returns the result.

A health check is VNF-specific. For a complex VNF, APPC initiates further subordinate health checks.

HealthCheck is a VNF level command which interrogates the VNF in order to determine the health of the VNF and the VNFCs. The HealthCheck will be implemented differently for each VNF.

Target URL	/restconf/operations/appc-provider-lcm:health-check
Action	HealthCheck
Action-Identifiers	Vnf-id
Payload Parameters	vnf-host-ip-address
Revision History	Changed in this version.

Paramete	**Description**	**Requir**	**Example**

vnf-host-ip-address	The IP address used to connect to the VNF, using a protocol such as SSH. For example, for a vSCP VNF, the floating IP address of the SMP should be used.	Yes	`"payload":`
			`"{\"vnf-host-ip-address\":`
			`\"10.222.22.2\"}"`

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Lock

Use the Lock command to ensure exclusive access during a series of critical LCM commands.

The Lock action will return a successful result if the VNF is not already locked or if it was locked with the same `request-id`, otherwise the action returns a response with a reject status code.

Lock is a command intended for APPC and does not execute an actual VNF command. Instead, lock will ensure that ONAP is granted exclusive access to the VNF.

When a VNF is locked, any subsequent sequential commands with same `request-id` will be accepted. Commands associated with other `request-id`s will be rejected.

The Lock command returns only one final response with the status of the request processing.
APPC locks the target VNF during any VNF command processing. If a lock action is then requested on that VNF, it will be rejected because the VNF was already locked, even though no actual lock command was explicitly invoked.

Target URL	/restconf/operations/appc-provider-lcm:lock
Action	Lock
Action-Identifier	Vnf-id
Payload Parameters	None
Revision History	Unchanged in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Migrate

Migrates a running target VM from its current host to another.

A destination node will be selected by relying on internal rules to migrate. Migrate calls a command in order to perform the operation.

Migrate suspends the guest virtual machine, and moves an image of the guest virtual machine's disk to the destination host physical machine. The guest virtual machine is then resumed on the destination host physical machine and the disk storage that it used on the source host physical machine is freed.

The migrate action will leave the VM in the same Openstack state the VM had been in prior to the migrate action. If a VM was stopped before migration, a separate VM-level restart command would be needed to restart the VM after migration.

A successful Migrate action returns a success response and the new node identity in the response payload block.

A failed Migrate action returns a failure and the failure messages in the response payload block.

NOTE: The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:migrate
Action	Migrate
Action-Identifiers	Vnf-id, vserver-id
Payload Parameters	vm-id, identity-url, tenant-id
Revision History	Unchanged in this version.

Table 17 Payload Parameters

Parameter	Description	Required?	Example
vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-	Yes	"payload": "{\"vm-id\": \"<VM-ID> \",

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

	link URL of the VM.		\"identity-url\": \"<IDENTITY-URL>\", \"tenant-id\": \"<TENANT-ID>\"}"
identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Rebuild

Recreates a target VM instance to a known, stable state.

Rebuild calls an OpenStack command immediately and therefore does not expect any prerequisite operations to be performed, such as shutting off a VM.

APPC only supports the rebuild operation for a VM that boots from image (snapshot), i.e., APPC rejects a rebuild request if it determines the VM boots from volume (disk).

A successful rebuild returns a success response and the rebuild details in the response payload block. A failed rebuild returns a failure and the failure messages in the response payload block.

NOTE: The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:rebuild
Action	Rebuild
Action-identifiers	Vnf-id, vserver-id
Payload Parameters	vm-id, identity-url, tenant-id
Revision History	Unchanged in this version.

**Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.**

Restart

Use the Restart command to restart a VNF or a single VM. The generic VNF Restart uses a simple restart logic where all VM's are stopped and re-started.

The generic Restart operation is invoked either for the VM or the VNF level.

Input Block	api-ver must be set to 2.00 for VNF Restart
Target URL	/restconf/operations/appc-provider-lcm:restart
Action	Restart
Action-identifiers	Vnf-id is required; if restart is for a single VM, then vserver-id is also required.
Payload Parameters	For VNF Restart: host Identity, vnf-host-ip-address For VM Restart: vm-id, identity-url, tenant-id
Revision History	Revised in this version.

Payload Parameters for **VNF Restart**

Parameter	Description	Required?	Example
Cloud Identity	The identity URL of the OpenStack host on which the VNF resource was created. If not provided, this information will be retrieved from the properties file.	No	"payload": "{\" vnf-host-ip-address \": \"<VNF_FLOATING_IP_ADDRESS> \", \" hostIdentity \": \"<OpenStack IP Address>\" }"
vnf-host-ip-address	The IP address used to connect to the VNF, using a protocol such as SSH. For example, for a vSCP VNF, the floating IP address of the SMP should be used.	Yes	

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Payload Parameters for **VM Restart**

Parameter	Description	Required?	Example

vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	`"payload":`
			`"{\"vm-id\": \"<VM-ID>`
			`\",`
			`\"identity-url\":`
			`\"<IDENTITY-URL>\",`
			`\"tenant-id\": \"<TENANT-`
			`ID>\"}"`

identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Snapshot

Creates a snapshot of a VM.

The Snapshot command returns a customized response containing a reference to the newly created snapshot instance if the action is successful.

This command can be applied to any VNF type. The only restriction is that the particular VNF should be built based on the generic heat stack.

NOTE: The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:snapshot
Action	Snapshot
Action-identifiers	Vnf-id is required. If the snapshot is for a single VM, then the vserver-id is also required.
Payload Parameters	vm-id, identity-url, tenant-id
Revision History	Unchanged in this version.

Payload Parameters

Parameter	Description	Required?	Example
vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	"payload": "{\"vm-id\": \"<VM-ID> \", \"identity-url\": \"<IDENTITY-URL>\", \"tenant-id\": \"<TENANT- ID>\"}"
identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Snapshot Response

The Snapshot command returns an extended version of the LCM response.

The Snapshot response conforms to the standard response format, but has the following additional field.

Table 21 Additional Parameters

Parameter	Description	Required?	Example
snapshot-id	The snapshot identifier created by cloud host. This identifier will be returned only in the final success response returned via the message	No	`"snapshot-id": "<SNAPSHOT_ID>"`

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Start

Use the Start command to start a VNF, VF-Module, or VM that is stopped or not running.

> **NOTE:** The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:start
Action	Start
Action-identifiers	Vnf-id is required; vf-module-id or vserver-id is also required if the action is at vf-module or vm level, respectively.
Payload Parameters	None
Revision History	Revised in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

StartApplication

Starts the VNF application, if needed, after a VM is instantiated/configured or after VM start or restart. Supported using Chef cookbook or Ansible playbook only.

A successful StartApplication request returns a success response.

A failed StartApplication action returns a failure response code and the specific failure message in the response block.

Target URL	/restconf/operations/appc-provider-lcm:startapplication
Action	StartApplication
Action-Identifiers	Vnf-id
Payload Parameters	See below
Revision History	New in this version.

Payload Parameter	**Description**	**Required?**	**Example**

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF (for Chef and Ansible, this will be the url to connect to the server).	Yes	"payload":
			"{\"request-parameters \": {
			\"host-ip-address\": \”value\”
			}
			\"configuration-parameters\": {\"<CONFIG- PARAMS>\"}

configuration-parameters	A set of instance specific configuration parameters should be specified, as required by the Chef cookbook or Ansible playbook.	No	

StartApplication Response

The StartApplication response returns an indication of success or failure of the request.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Stop

Use the Stop command to start a VNF, VF-Module, or VM that is stopped or not running.

> **NOTE:** The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:stop
Action	Stop
Action-identifiers	Vnf-id is required; vf-module-id or vserver-id is also required if the action is at vf-module or vm level, respectively.
Payload Parameters	None
Revision History	Revised in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

StopApplication

Stops the VNF application gracefully (not lost traffic), if needed, prior to a Stop command. Supported using Chef cookbook or Ansible playbook only.

A successful StopApplication request returns a success response.

A failed StopApplication action returns a failure response code and the specific failure message in the response block.

Target URL	/restconf/operations/appc-provider-lcm:stopapplication
Action	StopApplication
Action-Identifiers	Vnf-id
Payload Parameters	See below
Revision History	New in this version.

Payload Parameter	**Description**	**Required?**	**Example**

request-parameters	The parameters required to process the request must include the host-ip-address to connect to the VNF (for Chef and Ansible, this will be the url to connect to the server).	Yes	"payload":
			"{\"request-parameters
			\": {
			\"host-ip-address\":
			\”value\”
			}
			\"configuration-
			parameters\":
			{\"<CONFIG-
			PARAMS>\"}

configuration-parameters	A set of instance specific configuration parameters should be specified, as required by the Chef cookbook or Ansible playbook.	No	

StopApplication Response

The StopApplication response returns an indication of success or failure of the request.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Sync

The Sync action updates the current configuration in the APPC store with the running configuration from the device.

A successful Sync returns a success status.

A failed Sync returns a failure response status and failure messages in the response payload block.

This command can be applied to any VNF type. The only restriction is that the VNF has been onboarded in self-service mode (which requires that the VNF supports a request to return the running configuration).

Target URL	/restconf/operations/appc-provider-lcm:sync
Action	Sync
Action-identifiers	Vnf-id
Payload Parameters	None
Revision History	Unchanged in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Unlock

Run the Unlock command to release the lock on a VNF and allow other clients to perform LCM commands on that VNF.

Unlock is a command intended for APPC and does not execute an actual VNF command. Instead, unlock will release the VNF from the exclusive access held by the specific request-id allowing other requests for the VNF to be accepted.

The Unlock command will result in success if the VNF successfully unlocked or if it was already unlocked, otherwise commands will be rejected.

The Unlock command will only return success if the VNF was locked with same request-id.

The Unlock command returns only one final response with the status of the request processing.

Note: APPC locks the target VNF during any command processing. If an Unlock action is then requested on that VNF with a different request-id, it will be rejected because the VNF is already locked for another process, even though no actual lock command was explicitly invoked.

Target URL	/restconf/operations/appc-provider-lcm:unlock
Action	Unlock
Action-identifiers	Vnf-id
Payload Parameters	None
Revision History	Unchanged in this version.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

5 Update History

Lists the changes in LCM scope and command structure across releases of this document.

Version	Commands Removed
2.0.0	LiveUpgrade ModifyConfig (replaced by ConfigModify) Rollback SoftwareUpload Terminate Test

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.

Copyright © 2017 AT&T Intellectual Property. Copyright © 2017 Amdocs. All rights reserved.
Licensed under the Creative Commons License, Attribution 4.0 Intl. (the "License"); you may not use this documentation except in compliance with the License.