



Error Handling Procedures

- [Error Handling Using the ETSI API, on page 1](#)

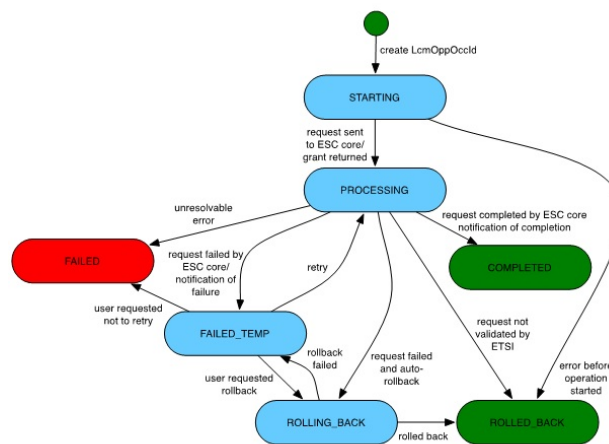
Error Handling Using the ETSI API

ETSI invokes the following error handling procedures for all its ETSI VNF lifecycle management (LCM) operations:

- Retry
- Rollback
- Fail
- Cancel

The image below represents the transitional states of the VNF lifecycle management operational occurrence.

Figure 1: VNF Lifecycle Management Transitional States





Note The *vnfLcmOpOccId* is encoded into the URI, which is the primary key to retrieve the request details.

The retry, rollback and fail requests are rejected if the LCM operation is in any other state other than the FAILED_TEMP state. This error returns HTTP code 409.

The retry, rollback, fail and cancel requests are not supported for the particular VNF LCM operation for the particular VNF. This error returns HTTP code 404.

An error occurs if the *vnfLcmOpOccId* does not exist in the ETSI database. This error returns HTTP code 404.

Retry

A retry request is applicable if there is a possibility of the LCM operation to succeed. The operation should be (pre-condition) in the FAILED_TEMP state for a retry request. You can send several retry requests, as long as the operation is in the FAILED_TEMP state.

Precondition	FAILED_TEMP state
Request	POST {api_root}/vnf_lcm_op_occs/{vnfLcmOpOccId}/retry()
Postcondition	PROCESSING state

Upon successful retry, ESC sends a START or PROCESSING notification. If the retry request fails, then ESC sends a notification to the NFVO with the details.

Rollback

A rollback request is made if it is not possible for the operation to succeed even after a retry request.

Set the *rollback_required* flag to true. If this is not set to true, then rollback is not performed.

Precondition	FAILED_TEMP state
Request	POST {api_root}/vnf_lcm_op_occs/{vnfLcmOpOccId}/rollback()
Postcondition	ROLLED_BACK

Upon successful rollback, the LCM operation is rolled back. If the rollback request fails, then the LCM operation is back to the failed_temp state.

Fail

When an LCM operation does not require a retry request, or a clean up, a fail request allows you to free up resources for a subsequent request.

If the *rollback_required* flag is set to true, a fail request cannot be made.

Precondition	FAILED_TEMP state
Request	POST {api_root}/vnf_lcm_op_occs/{vnfLcmOpOccId}/fail()
Postcondition	FAILED state

Upon successful execution of this request, the LCM operation is in FAILED state.

Cancel

A cancel request is possible if the operation is in STARTING state.

Precondition	STARTING state
Request	POST {api_root}/vnf_lcm_op_occs/{vnfLcmOpOccId}/cancel (CancelMode)
Postcondition	ROLLED_BACK

The cancel request is Forceful.



Note ETSI supports canceling an LCM operation in starting state only. The cancel request for LCM operations in processing or rolling back states are currently not supported.

Example JSON payload (CancelMode):

```
{
  "cancelMode": "FORCEFUL",
  "action": "cancel"
}
```

Set the *IsCancelPending* attribute of the *VnfLcmOpOcc* to true. This will stop the processing request, and move the LCM operation to ROLLED_BACK state.

Error Handling Procedures for ETSI VNF Lifecycle Operations

If the LCM operation for a VNF instance fails, the operation moves to the FAILED_TEMP state according to the state machine. To complete the intended operation, you must either run the retry or rollback request.

- If creating a VNF identifier fails, then no further action is required. The rollback request is not supported.
- If instantiating the VNF fails, then ESC terminates the request, and sends a new instantiation request.
- If operating the VNF fails, then no further action is required.
- If terminating the VNF fails, you must retry the operation, as rollback is not supported.
- If deleting the VNF operation fails, then no further action is required. The rollback request is not supported.



Note The error handling requests do not impact the operating VNF lifecycle operation.

For information on VNF lifecycle operations, see [VNF Lifecycle Operations Using ETSI API](#).

