



Recovering a Disconnected Leaf

If all fabric interfaces on a leaf are disabled (interfaces connecting a leaf to the spine) due to a configuration pushed to the leaf, connectivity to the leaf is lost forever and the leaf becomes inactive in the fabric. Trying to push a configuration to the leaf does not work because connectivity has been lost. This chapter describes how to recover a disconnected leaf.

- [Recovering a Disconnected Leaf Using the REST API, on page 1](#)

Recovering a Disconnected Leaf Using the REST API

To recover a disconnected leaf, at least one of the fabric interfaces must be enabled using the following process. The remaining interfaces can be enabled using the GUI, REST API, or CLI.

To enable the first interface, post a policy using the REST API to delete the policy posted and bring the fabric ports Out-of-Service. You can post a policy to the leaf to bring the port that is Out-of-Service to In-Service as follows:



Note In the following examples, the assumption is that 1/49 is one of the leaf ports connecting to the spine.

Procedure

Step 1 Clear the blacklist policy from the APIC (using the REST API).

Example:

```
$APIC_Address/api/policymgr/mo/.xml
<polUni>
  <fabricInst>
    <fabricOOServicePol>
      <fabricRsOosPath tDn="topology/pod-1/paths-$LEAF_Id/pathep-[eth1/49]"
        lc="blacklist" status="deleted" />
    </fabricOOServicePol>
  </fabricInst>
</polUni>
```

Step 2 Post a local task to the node itself to bring up the interfaces you want using `l1EthIfSetInServiceLTask`.

Example:

```
$LEAF_Address/api/node/mo/topology/pod-1/node-$LEAF_Id/sys/action.xml  
<actionLSubj oDn="sys/phys-[eth1/49]">  
<llEthIfSetInServiceLTask adminSt='start' />  
</actionLSubj>
```
