

如何生成ACI故障以及如何选择性地防止故障生成

目录

[简介](#)

[如何生成故障以及如何选择性地防止故障生成](#)

[高级机制](#)

[示例1 — 租户故障](#)

[示例2 — 物理故障](#)

简介

本文档介绍以应用为中心的基础设施(ACI)故障生成的高级流程，以及如何防止生成特定故障。本文档通过两个示例演示了这一点。

如何生成故障以及如何选择性地防止故障生成

高级机制

1. 每个故障都是类faultInst (或faultDelegate) 的托管对象(MO)。此故障MO由另一个MO (通常是其父MO) 生成，因为某些规则被违反。
2. 树中每个可以生成故障的MO都具有一个属性monPolDn，该属性指向另一个MO，该MO是监控策略对象。此对象允许修改属性，并允许触发器生成故障。监视策略对象有多个类，例如：
monInfraPol — 处理基础设施策略 (VMM管理器、接入端口策略、物理端口等) — 位于交换矩阵>访问策略>监控策略monFabricPol — 处理交换矩阵监控 — 位于交换矩阵>交换矩阵策略>监控策略monEPGPOL — 处理租户监控>位于租户>监控策略菜单中
3. 通常，它将是默认监控对象。但是，通过转到对象模型的特定区域，可以为任何这些监控策略类创建特定用户定义的监控策略。
4. 您可以修改这些监控策略的许多属性。本示例将展示如何防止为应用监控策略的所有对象生成给定故障。但是，您还可以修改故障生命周期计时器 (保留时间、浸泡时间等)。
5. 要修改故障严重性或防止生成故障，您需要选择与生成此对象的MO类 (例如，故障的父级) 对应的监控对象。
6. 然后，在此类下，选择要修改的故障代码，并选择初始严重性值“squeched”。
这可以防止分配给此特定监控策略的MO生成该代码的任何故障。

示例1 — 租户故障

每个故障都与对象关联。

```
admin@apic:~> moquery -d "uni/tn-RD/ipToEpg-Ext_10.200.1.101/rstoEpg-[uni/tn-RD/ap-App_RD1/epg-EPG_RD11]/fault-F0879"
Total Objects shown: 1
# fault.Inst code           : F0879 ack           : no cause           : resolution-failed
changeSet                   : childAction         : created            : 2015-01-22T00:05:00.286+01:00
descr                       : Failed to form relation to MO uni/tn-RD/ap-App_RD1/epg-EPG_RD11 of class
fvAEPg dn                   : uni/tn-RD/ipToEpg-Ext_10.200.1.101/rstoEpg-[uni/tn-RD/ap-App_RD1/epg-
```

```
EPG_RD11]/fault-F0879 domain          : infra highestSeverity : warning lastTransition :
2015-01-22T00:05:00.286+01:00 lc      : raised modTs           : never
occur          : 1 origSeverity      : warning prevSeverity   : warning rn             :
fault-F0879 rule          : dbgac-rs-to-epg-resolve-fail
```

上一个故障是类故障的MO。Inst，代码为F0879。

故障与终端组(EPG)对象关联，如下所示。

此对象是故障父级的可分辨名称(DN)。此父对象为dbg.RsToEpg类。

```
admin@apic:~> moquery -d uni/tn-RD/ipToEpg-Ext_10.200.1.101/rstoEpg-[uni/tn-RD/ap-App_RD1/epg-
EPG_RD11]
Total Objects shown: 1
# dbgac.RsToEpg tDn          : uni/tn-RD/ap-App_RD1/epg-EPG_RD11 childAction : dn          :
uni/tn-RD/ipToEpg-Ext_10.200.1.101/rstoEpg-[uni/tn-RD/ap-App_RD1/epg-EPG_RD11] forceResolve : no
lcOwn          : local modTs          : 2014-12-05T12:56:29.340+01:00 monPolDn          : uni/tn-
RD/monepg-RD_Monitoring
rType          : mo
rn             : rstoEpg-[uni/tn-RD/ap-App_RD1/epg-EPG_RD11]
state          : missing-target
stateQual      : none
status         :
tCl            : fvAEPg
tType         : mo
uid            : 15374
```

您可以看到此EPG对象与monPolDn对象关联。树中的大多数对象都由监控对象监控。

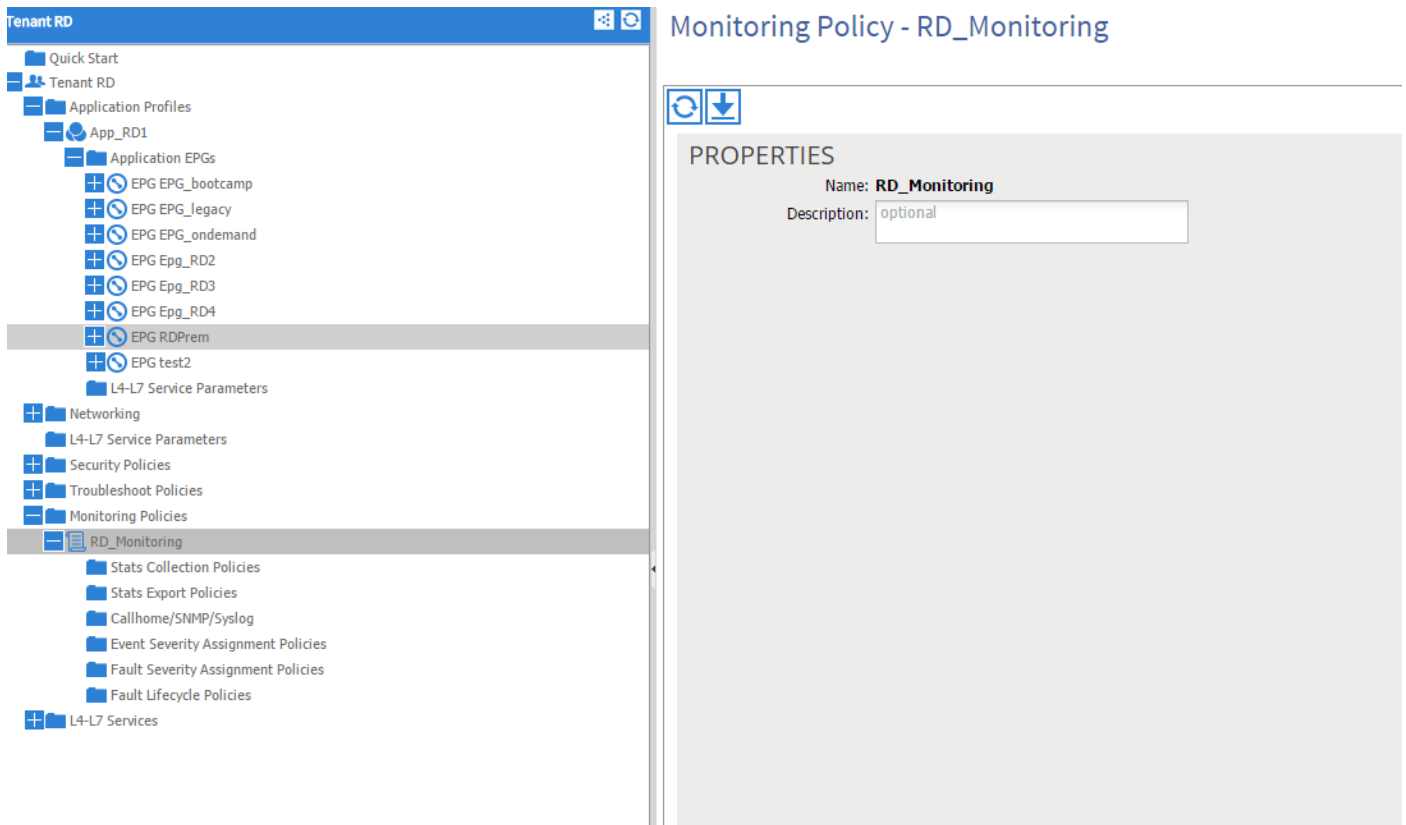
这是一个用户定义的带有dn的monEPGPol类监控对象。

```
uni/tn-RD/monepg-RD_Monitoring
```

以下是用于监控的完整对象。

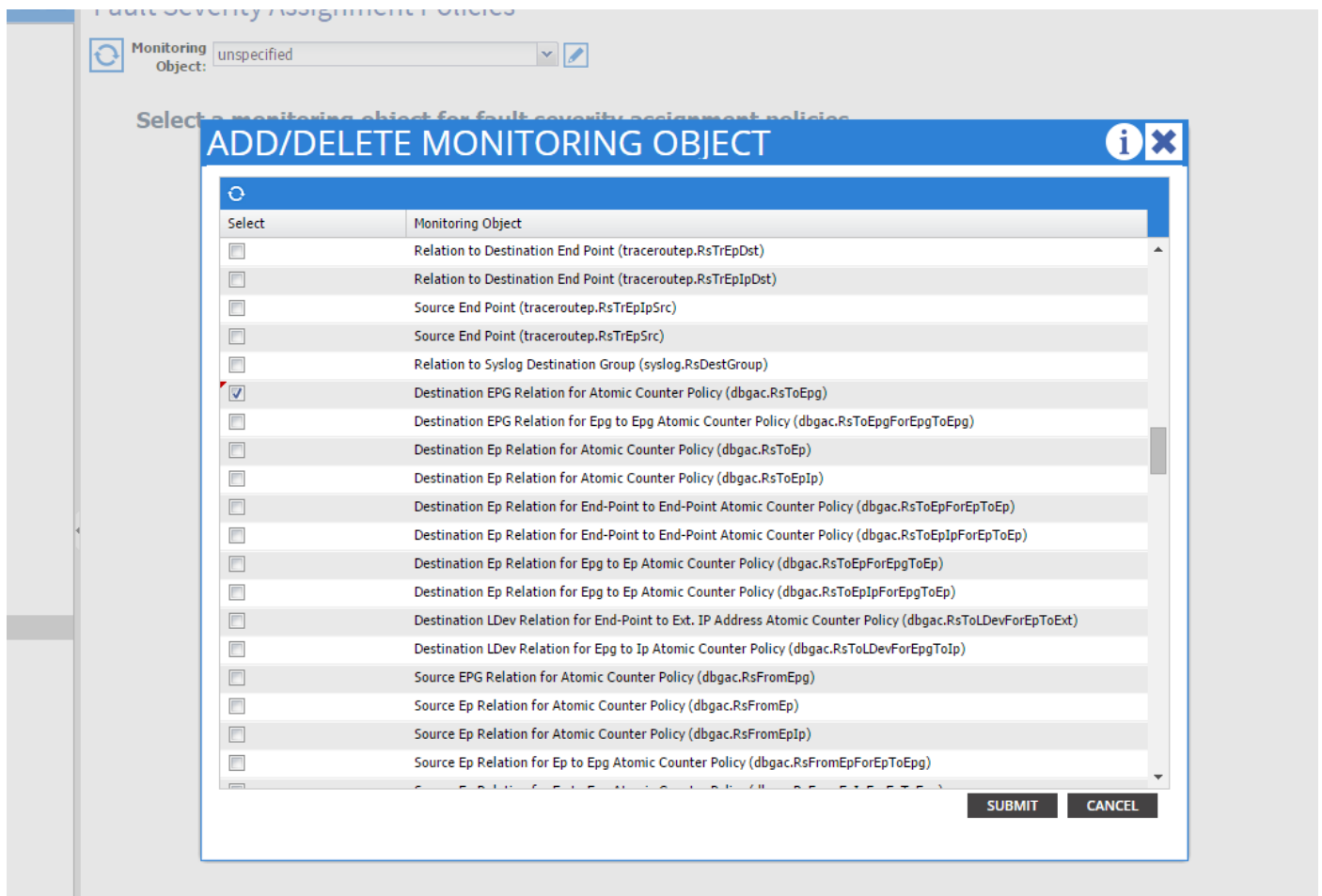
```
admin@apic:~> moquery -d uni/tn-RD/monepg-RD_Monitoring
Total Objects shown: 1
# mon.EPGPol name          : RD_Monitoring childAction : descr          : dn          : uni/tn-
RD/monepg-RD_Monitoring lcOwn          : local modTs          : 2014-11-13T15:41:45.326+01:00
monPolDn          : uni/tn-RD/monepg-RD_Monitoring ownerKey          : ownerTag          : rn          :
monepg-RD_Monitoring status          : uid          : 10673
```

monEPGPol对象在租户监控策略下配置，您可以在其中创建新策略或修改默认策略。以下是monEPGPol名称RD_Monitoring的示例。

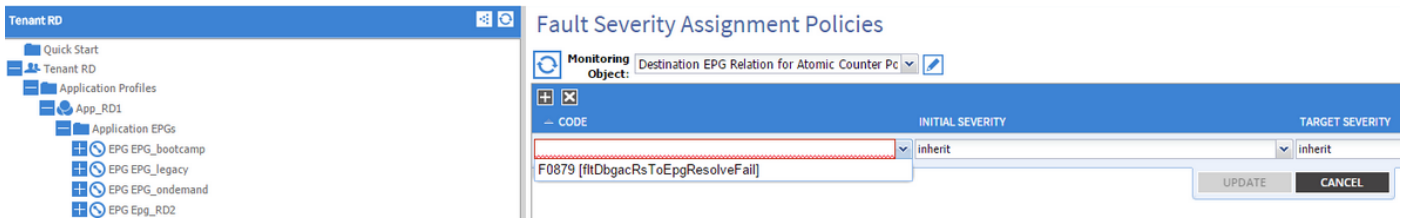


您可以选择故障严重性分配策略，然后点击铅笔（在监控对象旁边）。

然后，如果在该监控策略的监控对象列表中选择，则会为其创建故障的类(此处为 **dbgac.RsToEpg**)。



您可以看到与该特定类关联的所有故障（此处显示的唯一故障是F0789）。



故障F0789是示例开头故障显示的代码。

您可以选择此故障，如果将初始严重性设置为静止（可以将目标严重性保留为继承），则它会防止将来生成此故障，假定这些故障是由与刚刚修改的监控策略有链接的对象生成的。

但是，它不会清除现有故障，而只会清除新故障。

示例2 — 物理故障

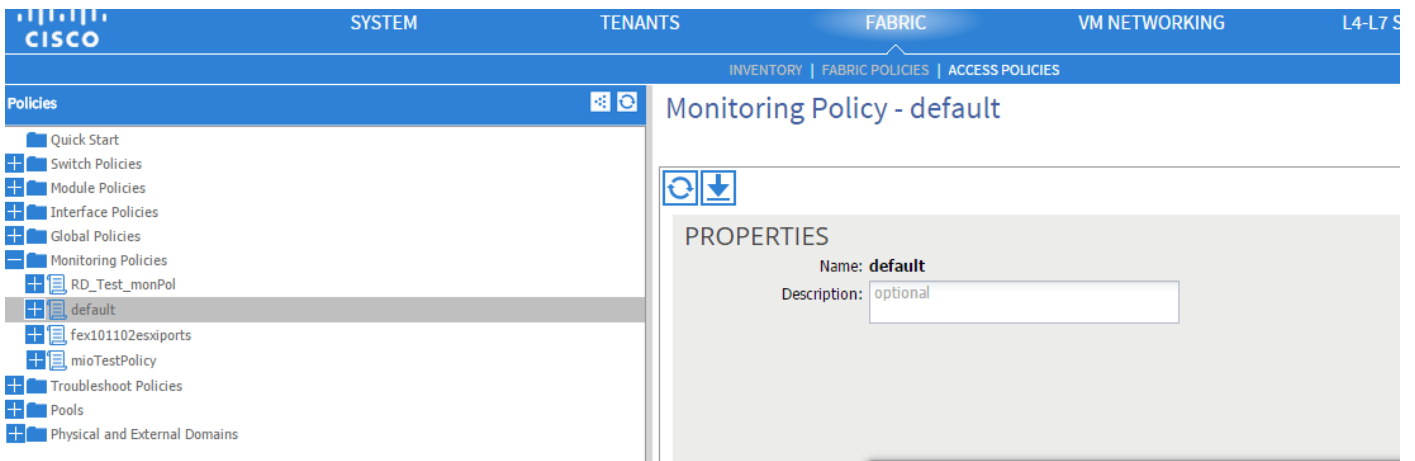
在本例中，由于枝叶上的端口1/25处于管理状态，但没有SFP，因此会生成故障。

```
admin@apic:~> moquery -c faultInst -f 'fault.Inst.code == "F1678"'
Total Objects shown: 2
# fault.Inst code          : F1678 ack          : no cause          : port-failure
changeSet                 : usage (New: epg) childAction      : created          : 2015-01-
19T14:26:13.862+01:00 descr          : TEST FAULT -- Port is down,
reason:sfpAbsent(connected), used by:EPG,
lastLinkStChg:1970-01-01T01:00:00.000+01:00, operSt:down dn          : topology/pod-1/node-
101/sys/phys-[eth1/25]/phys/fault-F1678 domain          : access highestSeverity : critical
lastTransition           : 2015-01-19T14:28:41.668+01:00 lc          : raised modTs          :
never occur              : 1 origSeverity          : critical prevSeverity          : critical
rn                        : fault-F1678 rule          : ethpm-if-port-down-infra-epg-test
severity                 : critical status          : subject              : port-down type          :
communications uid          :
```

这与物理端口关联。这是生成该故障的父MO。

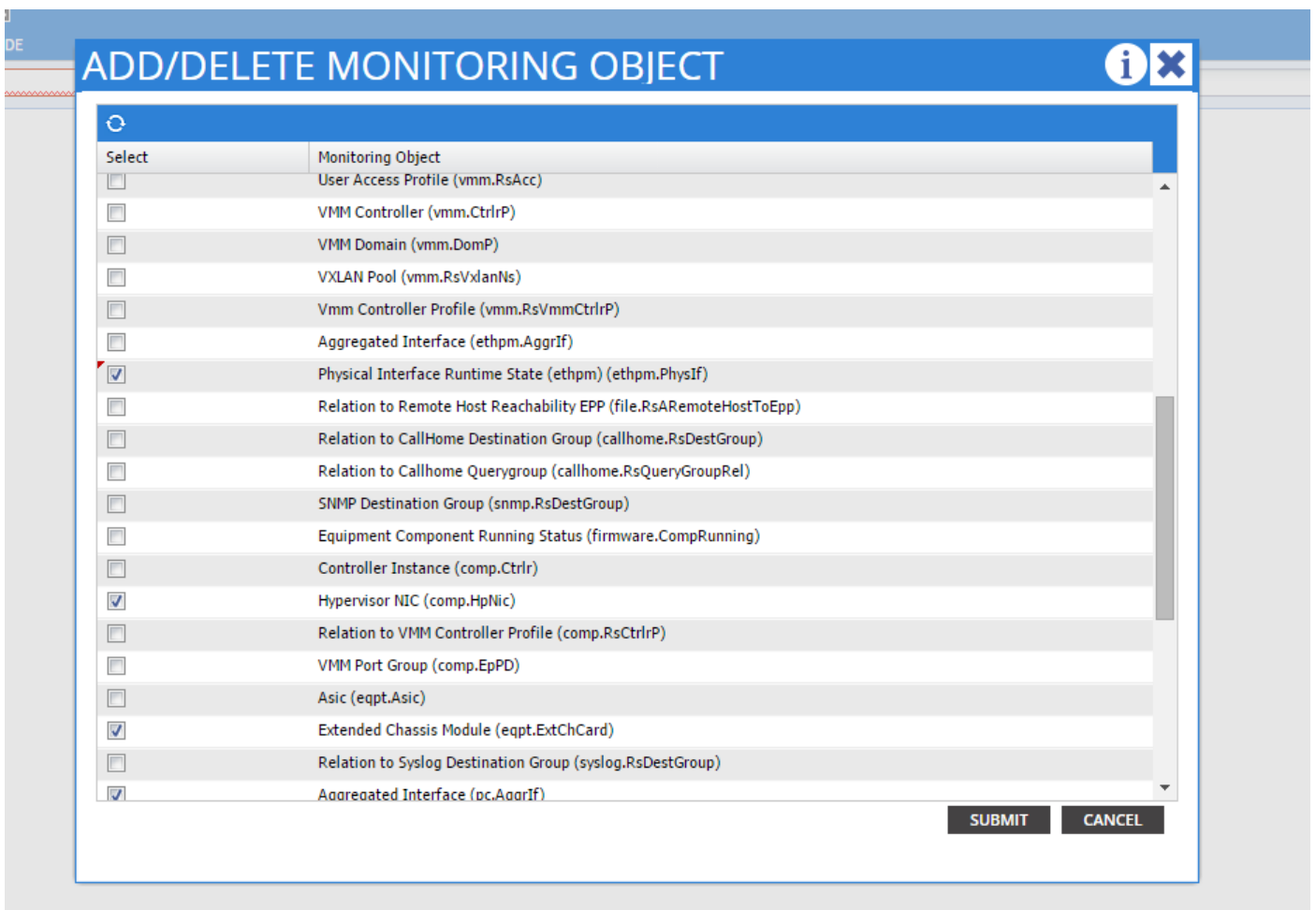
```
admin@apic:~> moquery -d topology/pod-1/node-101/sys/phys-[eth1/25]/phys
Total Objects shown: 1
# ethpm.PhysIf accessVlan          : vlan-1 allowedVlans          : backplaneMac          :
50:87:89:A2:2A:C1 bundleBupId          : 1 bundleIndex          : unspecified cfgAccessVlan          :
vlan-1 cfgNativeVlan          : vlan-1 childAction          : currErrIndex          : 4294967295
diags                          : none dn                    : topology/pod-1/node-101/sys/phys-[eth1/25]/phys
encap                          : 3 errDisTimerRunning : no errVlanStatusHt          : 0 errVlans          :
hwBdId                         : 0 intfT                    : phy iod                : 29 lastErrors          : 0
lastLinkStChg                 : 1970-01-01T01:00:00.000+01:00 media          : 2 modTs                :
never monPolDn                 : uni/infra/moninfra-default nativeVlan          : vlan-1
```

这与如下所示配置的monInfraPol对象关联。



```
admin@apic:~> moquery -c monInfraPol
Total Objects shown: 4
# mon.InfraPol name          : default childAction : descr          : dn          :
uni/infra/moninfra-default lcOwn          : local modTs      : 2014-08-06T07:58:19.494+01:00
monPolDn       : uni/infra/moninfra-default ownerKey      : ownerTag      : rn          : moninfra-
default status  : uid          : 0
```

在故障严重性分配策略下，点击工作窗格中监控对象下拉列表旁边的铅笔。添加一个类，在其中修改监控属性。然后选择生成故障的对象的类，即ethmPhysIf。



选择此类并点击+图标，以查看为该对象生成的每个故障。

在本例中，您可以看到故障F1678，并且可以修改其属性。选择初始严重性压缩和目标严重性继承可防止从应用了此监控策略的对象生成该代码的新故障。



进行更改后，如果启用端口1/25且端口中没有SFP，则不会生成任何故障！

注意：在软件版本2.2之前的版本中：不会清除现有故障（即使在清除保留模式中）。

注意：在软件版本2.2及更高版本中：即使现有故障也会受到新策略的影响。