



# CHAPTER 10

## Alarms

---

The Cisco IOS XR XML API supports the registration and receipt of notifications; for example, asynchronous responses such as alarms, over any transport. The system supports alarms and event notifications over XML/SSH.

An asynchronous registration request is followed by a synchronous response and any number of asynchronous responses. If a client wants to stop receiving a particular set of asynchronous responses at a later stage, the client sends a deregistration request.

One type of notification that is supported by the Cisco IOS XR XML API is alarms; for example, syslog messages. The alarms that are received are restricted by a filter, which is specified in the registration request. An alarm registration request is followed by a synchronous response. If successful, the synchronous response contains a RegistrationID, which is used by the client to uniquely identify the applicable registration. A client can make many alarm registrations. If a client wants to stop receiving a particular set of alarms at a later stage, the client can send a deregistration request for the relevant RegistrationID or all Registration IDs for the session.

When an asynchronous response is received that contains an alarm, the registration that resulted in the alarm is determined from the RegistrationID.

These sections describe the XML used for every operation:

- [Alarm Registration, page 10-117](#)
- [Alarm Deregistration, page 10-118](#)
- [Alarm Notification, page 10-119](#)

## Alarm Registration

Alarm registration and deregistration requests and responses and alarm notifications use the <Alarm> operation tag to distinguish them from other types of XML operations. A registration request contains the <Register> tag, which is followed by several tags that specify the filter requirement. If registration for all alarms is required, no filter is specified. These filter criteria are listed:

- SourceID
- Category
- Group
- Context
- Code

- Severity
- BiStateOnly

If it succeeds, the response contains a <Register> tag with a RegistrationID attribute. If it fails, the filter tag that caused the error appears with an error message attribute. This example shows a registration request to receive all alarms for configuration change; for example, commit notifications:

#### Sample XML Request from the Client Application

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Alarm>
    <Register>
      <Group>CONFIG</Group>
      <Code>DB_COMMIT</Code>
    </Register>
  </Alarm>
</Request>
```

#### Sample XML Response from the Router

```
<?xml version="1.0" encoding="UTF-8"?>
Response MajorVersion="1" MinorVersion="0">
  <Alarm>
    <Register RegistrationID="123"/>
  </Alarm>
  <ResultSummary ErrorCount="0"/>
</Response>
```



#### Note

If a second registration is made with the same filter, or if the filters with two registrations overlap, these alarms that match both registrations are received twice. In general, each alarm is received once for each registration that it matches.

If a session ends (for example, the connection is dropped), all registrations are automatically canceled.

## Alarm Deregistration

An alarm deregistration request consists of the <Alarm> operation tag followed by the <Deregister> tag, with the optional attribute RegistrationID. If RegistrationID is specified, the value must be that returned from a previous registration request. The registration with that ID must not have already been deregistered or an error is returned. If it is not specified, the request results in all alarm registrations for that session being deregistered.

This example shows a deregistration request for the RegistrationID returned from the registration request example:

#### Sample XML Request from the Client Application

```
<?xml version="1.0" encoding="UTF-8"?>
<Request MajorVersion="1" MinorVersion="0">
  <Alarm>
    <Deregister RegistrationID="123"/>
  </Alarm>
</Request>
```

**Sample XML Response from the Router**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response MajorVersion="1" MinorVersion="0">
  <Alarm>
    <Deregister RegistrationID="123"/>
  </Alarm>
  <ResultSummary ErrorCount="0"/>
</Response>
```

## Alarm Notification

Alarm notifications are contained within a pair of <Notification> tags to distinguish them from normal responses. Each notification contains one or more alarms, each of which is contained within a pair of <Alarm> tags. The tags have an attribute RegistrationID, where the value is the RegistrationID returned in the registration that resulted in the alarm.

The tags contain these fields for the alarm:

- SourceID
- EventID
- Timestamp
- Category
- Group
- Code
- Severity
- State
- CorrelationID
- AdditionalText

This example shows the configuration commit alarm notification:

```
<?xml version= "1.0" encoding= "UTF-8"?>
<Notification MajorVersion="1" MinorVersion="0">
  <Alarm RegistrationID="123">
    <SourceID>RP/0/0/CPU0</SourceID>
    <EventID>84</EventID>
    <Timestamp>1077270612</Timestamp>
    <Category>MGBL</Category>
    <Group>CONFIG</Group>
    <Code>DB_COMMIT</Code>
    <Severity>Informational</Severity>
    <State>NotAvailable</State>
    <CorrelationID>0</CorrelationID>
    <AdditionalText>config[65704]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed
by user &#39;admin&#39;. Use &#39;show commit changes 100000490&#39; to view
the changes.</AdditionalText>
  </Alarm>
</Notification>
```

