

# Adding a new alarm

## Adding a new alarm

### Introduction

- All processes (C++, java, and scripts) can raise alarms, which are logged in a single file (/var/log/sipxpbx/sipXalarms.log)
- Internally, alarms are raised by specifying a unique string id and a list of any desired string parameters. There are no restrictions on the alarm code: examples could be "ALARM\_FILE\_REP\_FAILED" or "ALARM\_12345" or "ANY\_OLD\_ALARM\_CONDITION" (this string is not viewable by users).
- A set of XML alarm definition files contains details on each alarm, identified by its unique id (above). This includes an external unique alarm code, severity, component, description, resolution, and actions.
- High-level alarm configuration data is stored in /etc/sipxpbx/alarm-config.xml. This includes the email address from which alarms should be sent.
- Non-localized per-alarm configuration data is stored in /etc/sipxpbx/alarms/\*.xml. This includes the unique internal id and external code for each alarm, and the actions which should be taken when this alarm occurs. Alarm definitions can be split among several files as desired by the application.
- Description and resolution strings are stored in separate, localizable, files in /share/sipxecs/alarms/\*-strings.xml. These files are not backed up.
- Actions currently include log and email. These are configurable for each alarm, as well as global enable/disable flags for each notification method. Max\_report and thresholds are also provided on a per-alarm basis (e.g. only report this alarm x times; or only report this alarm after it occurs x times).

### Adding new alarms

To add a new alarm:

1. Add a new section into the appropriate config.xml file in sipXsupervisor/etc/, with a new alarm id.
  2. Add the description and resolution strings in the appropriate \*-strings.xml file in sipXsupervisor/meta/alarms/. Normally there is no need to create new files; however if adding an optional package it might be desirable to capture the alarms in separate files.
- The alarm id is a required field, and can be any unique string. This string is only used internally.
  - The alarm code is specified in the config file (e.g. sipXsupervisor/etc/sipXalarms-config.xml). New alarms should be added at the end of the appropriate file, and the code should follow the naming pattern for that file (e.g. SIPXnnnnn, incremented for each new alarm).

### Generating alarms

Interfaces are provided for Java and C++ code to call to generate alarms. There are only two parameters: the unique internal ID and the parameter string (or list of strings).

- For C++, there is a static function raiseAlarm(alarmId, alarmParams). See sipXsupervisor/src/SipXProcess.cpp for an example ([sample C++ update that adds an alarm](#)).
- For sipXconfig (Java), the object should inject an AlarmContext, then call AlarmContext.raiseAlarm(alarmId, alarmParams...). See sipXconfig/neoconf/src/org/sipfoundry/sipxconfig/login/LoginContextImpl.java for an example.
- For other Java processes, use the SipXAlarmClient in sipXcommons to send xmlrpc to the Alarm Server on the localhost (see sipxbridge/Gateway.java for an example) ([sample Java code that adds an alarm](#)).
- Many alarm conditions are detected by the various scripts used by sipx. A script sipx-alarm exists to use the xml-rpc interface to report alarms. Note however that the script can only be used once sipxsupervisor is up and running. See sipXtools/src/sipx-mrtg-threshok.in for an example.