Drools Rules Engine

About this appendix:

This appendix introduces and describes the Drools rules engine.

Drools Rules Engine Overview, introduces the Drools rules engine components and terminology.

Drools and Cisco ANA Integration, explains how Drools is integrated within Cisco ANA.

Drools Definitions in Cisco ANA, describes Drools within Cisco ANA.

Upgrading Rule Files, describes the rule upgrade process.

Drools Rules Engine Overview

Drools rules engine is a general-purpose expert-system generator and combines rule-based techniques and object-oriented programming. It also provides a customizable mechanism to add decision support and data flow control functions to business applications.

Drools rules engine is based on an object-oriented paradigm and uses user-defined rules to perform pattern matching on different conditions. The rules are written in a java like syntax, and are organized into source files (known as a ***rule files***), which are plain ASCII files.

Drools Components and Terminology

Drools uses *objects* as marked out by *patterns* and *rules* that invoke certain *Actions*:

- Drools ***objects*** are Java objects and can be represented by instances of Java classes or XML schemas.
- A ***pattern*** is a coded expression (“program”), which manipulates one or more objects to form a pattern to make, adapt or fashion behavior according to designed logic.
- ***Working Memory*** is where Drools stores all the objects, which it is currently handling.
- ***Actions*** are operations that may change the working memory.
- A ***rule*** can perform many types of actions, such as:
  - Add or remove an object from the working memory
  - Modify an object
  - Execute a method on one of the objects
- The ***Agenda*** is where Drools stores the list of rules to be fired.

Drools and Cisco ANA Integration

The Drools rule engine enables the user to extend the Cisco ANA alarm correlation mechanism with user-defined rules and business logic.

The Drools rule engine is fully integrated within the Cisco ANA Gateway and does not require any synchronization or maintenance. It can access all Cisco ANA information and functions:

- Use up-to-date network information (topology, hierarchy, inventory) as part of its rule processing
- Create and manipulate alarms, and send commands to NEs
- Define operator-logic rules, such as:
  - If the alarm was not attended to within 2 hours then…
  - If there are more than 5 open alarms on this device then…
  - If today is Tuesday then…

The rules are written for Drools in xml format files, and can be modified and reloaded in runtime. There is neither a need to compile the rules, nor to restart any Cisco ANA component.

Cisco ANA maintains two Drools processing instances (Contexts), with two respective rule files:

- ***Pre-correlation*** processing context
- ***Post-correlation*** processing context

The Rule files are located under **~sheer/Main/data** in the Cisco ANA Gateway server.

Drools Definitions in Cisco ANA

The Drools mechanism runs in two processing contexts—

- Pre-correlation context – defined in the **pre.drl** rule file
- Post-correlation context – defined in the **post.drl** rule file.

The Drools parameters are stored in the Cisco ANA registry file **mmvm.xml** under the event-management key:

- Context ID - the Drools context name
- Rule file name – the name of the respective rule file.
  - the specified rule (the file name under **mmvm.xml**).

Upgrading Rule Files

To upgrade a rule file:

Step 1 Make a copy of the required rule file (*pre.drl* or *post.drl*), and edit it.

Step 2 Copy the updated file (under a temporary name) to the Cisco ANA Gateway (directory **~sheer/Main/data**).

Note In case the rule file is edited on a PC, make sure that the text format is compliant to the UNIX version that runs on the Cisco ANA Gateway. If necessary, use a utility such as ***DOS2UNIX*** for conversion.

Step 3 Check the validity of the new file, by running the ***checkrules.cmd*** utility (developed by Cisco to validate the rules syntax), as follows—**checkrules.cmd <rule-file-name>**

Step 4 Once the rule file has been validated:

- Copy the new rule file on top of the respective existing rule file.
- Reload rules files by running the reloadrules.cmd utility, as follows:
- reloadrules.cmd <contex-id>

For information about using Shell commands to manipulate Drools, refer to the *Cisco Active Network Abstraction Shell User's Guide*.

For more information about Drools, go to **www.drools.org**