



CHAPTER 26

Drools Rules Examples

This chapter provides examples to illustrate how Drools Rules can be used in Cisco Prime Network 3.8 to enable alarm forwarding using e-mails.

Topics include:

- [Example 1: Generate E-mail Upon Link Down/Up Ticket, page 26-1](#)
- [Example 2: Generate E-mail for Critical Ticket, page 26-2](#)

Example 1: Generate E-mail Upon Link Down/Up Ticket

This example is a of a Drools Rule that generates an e-mail if a LinkDown / LinkUp Ticket is raised.

```
<rule name="Severity critical Alert">
  <parameter identifier="networkEvent">
    <class>com.sheer.imo.newalarm.INetworkEvent</class>
  </parameter>
  <java:condition>networkEvent.getName() == 1</java:condition>
  <java:consequence>
    import javax.mail.*;
    import javax.mail.internet.*;
    import java.util.*;
    import com.sheer.system.os.interfaces.Logger;
    Logger.getInstance().fatal("\nCritical Ticket Rule Fired\n");
    String recipients[ ] = new String[] { "<emailID1>", "<emailID2>" };
    String subject = networkEvent.getDescription();
    String message = "A Link Down/up Ticket has been raised. Please attend to the
    problem immediately";
    String from = "CiscoANA@company.com";
    boolean debug = false;
    //Set the host smtp address
    Properties props = new Properties();
    props.put("mail.smtp.host", "<smtp server name>");
    // create some properties and get the default Session
    Session session = Session.getDefaultInstance(props, null);
    session.setDebug(debug);
    // create a message
    Message msg = new MimeMessage(session);
    // set the from and to address
    InternetAddress addressFrom = new InternetAddress(from);
    msg.setFrom(addressFrom);
    InternetAddress[] addressTo = new InternetAddress[recipients.length];
    for (int i = 0; i < recipients.length; i++) {
      addressTo[i] = new InternetAddress(recipients[i]);
    }
  </java:consequence>
</rule>
```

```

        msg.setRecipients(Message.RecipientType.TO, addressTo);
        // Setting the Subject and Content Type
        msg.setSubject(subject);
        msg.setContent(message, "text/plain");
        Transport.send(msg);
    }
</java:consequence>
</rule>

```

Example 2: Generate E-mail for Critical Ticket

The following Drools Rule example generates an email if a Ticket with severity Critical has been raised.



Note

Only new ticket notifications will be captured here—updates on this ticket will not be received. For example, if a ticket is initiated with Aggregated Severity MINOR and then due to correlation with a critical alarm the severity was changed to CRITICAL, it won't be captured in this rule.

```

<rule name="Severity critical Alert">
  <parameter identifier="ticket">
    <class>com.sheer.imo.newalarm.ITicket</class>
  </parameter>
  <java:condition>ticket.getAggregatedSeverityEnum() == 6</java:condition>
  <java:consequence>
    import javax.mail.*;
    import javax.mail.internet.*;
    import java.util.*;
    import com.sheer.system.os.interfaces.Logger;
    import com.sheer.imo.newalarm.ITicket;
    import com.sheer.imo.keys.INewAlarmOid;
    import com.sheer.client.common.util.DisplayNameFactory;
    Logger.getInstance().fatal("\nCritical Ticket Rule Fired\n");
    String recipients[ ] = new String[ ] { "<emailID1>", "<emailID2>" };
    String subject = ticket.getDescription();
    String message = "Ticket ID = " + ((INewAlarmOid)ticket.getObjectId()).getId() + "\n"
    +
    "Severity = " + ITicket.S_AGGREGATED_SEVERITY[ticket.getAggregatedSeverityEnum()] +
    "\n" + "Creation Time = " + ticket.getCreationTime()
    + "\n" + "Location = " + DisplayNameFactory.getTextForIOid(ticket.getSource());
    String from = "CiscoANA@company.com";
    boolean debug = false;
    //Set the host smtp address
    Properties props = new Properties();
    props.put("mail.smtp.host", "<smtp server name>");
    // create some properties and get the default Session
    Session session = Session.getDefaultInstance(props, null);
    session.setDebug(debug);
    // create a message
    Message msg = new MimeMessage(session);
    // set the from and to address
    InetAddress addressFrom = new InetAddress(from);
    msg.setFrom(addressFrom);
    InetAddress[] addressTo = new InetAddress[recipients.length];
    for (int i = 0; i < recipients.length; i++) {
      addressTo[i] = new InetAddress(recipients[i]);
    }
    msg.setRecipients(Message.RecipientType.TO, addressTo);
    // Setting the Subject and Content Type
    msg.setSubject(subject);
    msg.setContent(message, "text/plain");
  </java:consequence>
</rule>

```

```

        Transport.send(msg);
    </java:consequence>
</rule>

```

**Note**

To create this rule for a MAJOR ticket, change the <java:condition> section as follows:

```

<java:condition>((ticket.getAggregatedSeverityEnum() ==
6) || (ticket.getAggregatedSeverityEnum() == 5))</java:condition>

```

**Note**

The Properties object in the above examples is used only the first time Session.getDefaultInstance (props, null) method is called, when a new Session object is created. Subsequent calls return the Session object that was created by the first call, and ignore the passed Properties object. Use the Session.getInstance (props, null) method to get a new Session object every time the method is called (for example, if you need to change the Mail SMTP Host).

The following table describes the properties used in the above examples.

Property	Description
rule name	The name of the rule.
parameter identifier	A definition of a parameter to use in this rule context. The parameter value will be any instance of the class defined within the <class></class> brackets.
java:condition	The condition that initiates the e-mail generation, based on the rule definitions. For Example 1, the condition is that a LinkDown/Up ticket is raised. For Example 2, the condition is that a ticket with Critical severity is raised.
java:consequence	The set of Java operations to run in case the previous conditions have been met. This section should include your drool rule operations.
Logger.getInstance	Helps to identify whether a rule was triggered. After you reload the rule, go to ~/Main/logs directory and enter tail -f 11.out . Check the log file to find out whether the rule was triggered.
String recipients	The e-mail IDs to which you want the system to send an e-mail.
String subject	The string to fetch the description for the alarm / ticket. In example 1, the string is networkEvent.getDescription().
String from	The e-mail ID to be displayed in the From field.

■ Example 2: Generate E-mail for Critical Ticket