# Rules Engine

The main function of the rules engine is the evaluation of rules. The rules engine object is what determines whether a rule is true or false.

## Overview

A rules engine object is created by the broadcast receiver once an event has been triggered. The broadcast receiver passes in the cause type, parameter and the context to the rules engine. The rules engine uses the type to send a request to the database for all rules of that type. Once the list of rules has been received, the rules engine evaluates each rule in separate threads using Android’s AsyncTask.

In each thread, an expression tree of rule causes is built using a stored character string in each rule. This expression tree is then evaluated recursively in a depth-first fashion. Each individual cause object within the tree has its own isTrue method that returns either true or false depending on what value is evaluated for the cause. Using these returned values, the expression tree will return a final value, true or false, to the AsyncTask. If the tree of the rule returns true, then the task will request the effect list for the rule from the database. This effect list is then sent to an ActionExecutor object for execution, along with the context initially passed from the broadcast receiver.

## Structural View

If rule is true

Broadcast Receiver

Rules Engine

Action Executor

Database

Get rules

Create

Create

Get effects

The rules engine interacts with the broadcast receiver, the database, and the action executor depending on if the rule cause tree returns true or not. If the rule evaluates to be false, then the right half of the view does not occur.

## Thread View

Broadcast Receiver

Rules Engine

Rules Engine

Rule

Rule

Rule

Rule

Action Executor

Action Executor

Action Executor

Action Executor

Each rule stemming from a rules engine object is evaluated in its own thread. Should the rule return true, then an effect list is fetched from the database and an action executor object is created, which takes in the effect list and executes each effect.

## Expression Tree

RETURN

OR

AND

AND

1554

1685

1234

1121

This is a view of the expression tree used to evaluate rule causes. This particular tree is the tree for a Boolean algebra equivalent of (1121 & 1234) + (1554 & 1685). The character string stored in the rule for this tree would be “+:&:1121@:1234@@:&:1554@:1685$”. ‘+’ indicates an OR node. ‘&’ indicates an AND node. ‘:’ indicates moving down a level while ‘@’ indicates moving up one level. The numbers are the cause ids that will be used to receive actual cause objects from the database for evaluation. The ‘$’ character indicates the end of the string.

## Individual Cause Evaluation

Causes are evaluated using a function containing a switch statement that chooses evaluation methods based on the rule type.

Pseudocode for cause evaluation:

Bool isTrue(eventType, param)

{

// event type is given to use by the broadcast receiver

Switch(eventType)

{

// each case returns true or false depending on specific //conditions of each case

Case wifi: // do something

Break;

Case phoneCall: // do something

Break;

Case time: // do something

Break;

Default: return false;

}

}