EagleSchedule Tutorial

This document explains how to install, run and define project definition scripts for the *eagleschedule* project scheduler.

Installation

Depending on your Linux architecture being 32 or 64 bits, download respectively the **eagleschedule_x86** or the **eagleschedule_x64** binary to your local Linux folder.

Usage

eagleschedule_x86 <project definition script> (x86 architectures)

eagleschedule_x64 <project definition script> (x64 architectures)

The parameter <project definition file> is a script in a specific syntax to define project constraints, which is described in detail in the next section. Some example scripts, explained and commented, are available for download as *.def files.

Syntax of project definition files

The following section describes the syntax and format of a project definition script to pass as parameter to *eagleschedule*.

Comments

Comment lines start by **#**. Blank lines are also admitted.

Project header

Every script starts by an instruction defining the number of project activities and the number of different project resources.

PROJECT n r

The PROJECT instruction defines a project with n activities and with r different resources. This instruction occurs only once in the script and must be the first one.

Activities duration

After defining the number of activities, the duration of each activity must be set.

ACTIVITY x, t

The ACTIVITY instruction sets t days (or any other time unit) of duration for the activity number x. For a project with n activities, the activity number ranges from 1 to n. Each activity is referred by a unique number.

Resources capacities

After defining the number of different resources, the capacity of each resource must be set.

RESOURCE x, k

The RESOURCE instruction sets k units of total available capacity for the resource x. For a project with r resources, the resource number ranges from 1 to n. Each resource is referred by a unique number.

Activity dependencies

Set the activities whose start depends on the others ending.

SEQUENCE x, y

The SEQUENCE instruction sets that the activity number x must finish for the activity number y to start

Resources allocation

Set the resources consumption by activities

ALLOCATE x, y, k

The ALLOCATE instruction sets that the activity number x consumes k units of resource number y.

Example

Suppose a short project to repaint two shops, with the following constraints:

- The project has six activities:
    - Emptying the first shop (1 day)
    - Painting the first shop (2 day)
    - Refilling back the first shop (1 day)
    - Emptying the second shop (1 day)
    - Painting the second shop (2 day)
    - Refilling back to second shop (1 day)
- There is only one painter available
- There are two assistants available to carry the furniture
- A shop can only be painted after the shop is empty
- A shop can only be refilled back after the painting is done

This would be a possible specification script for solving by *eagleschedule*:

```
# Project has 6 activities and 2 different types of resources (assistants and painter)
PROJECT 6, 2
# Emptying the first shop takes one day
ACTIVITY 1, 1
# Painting the first shop takes two days
ACTIVITY 2, 2
# Refilling back the first shop takes one day
ACTIVITY 3, 1
# Emptying the second shop takes one day
ACTIVITY 4, 1
# Painting the second shop takes two days
ACTIVITY 5, 2
# Refilling back the second shop takes one day
ACTIVITY 6, 1
# There are two assistants available
RESOURCE 1, 2
# There is one painter available
RESOURCE 2, 1
# The first shop can only be painted after being emptied
SEQUENCE 1, 2
# The first shop can only be refilled after the painting is done
SEQUENCE 2, 3
# The second shop can only be painted after being emptied
SEQUENCE 4, 5
# The first shop can only be refilled after the painting is done
SEQUENCE 5,6
# The activities related to emptying and refilling the shops require two assistants
ALLOCATE 1,1,2
ALLOCATE 3,1,2
ALLOCATE 4,1,2
ALLOCATE 6,1,2
# The activities related to painting require one painter
ALLOCATE 5,2,1
ALLOCATE 2,2,1
```