



Private Cloud Simulation Service

Overview

- ◆ EurekaCloud is a self-hosted (private) distributed service that allows automating and managing the simulation workflow.
- ◆ Distributed architecture and multi-user support.
- ◆ Simulation projects can be queued by simply copying them in a shared folder, which is constantly monitored by the service.
- ◆ Simulations can be started at a predefined time (batch mode) or can be launched “on-demand”.
- ◆ The simulation queue can be managed by means of a web interface.
- ◆ EurekaCloud can be extended and customized by means of C# scripts that allow the implementation of any further automation step.

Advantages

- ◆ Simulation process automation.
- ◆ Optimization of Eureka licenses usage.
- ◆ From design to production automation.
- ◆ Possible integration with existing management tools (e.g. Product Lifecycle Management tools).
- ◆ Real time notification.
- ◆ Comprehensive reporting.
- ◆ History of simulations.

The workflow



- ◆ EurekaCloud allows a complete automation of the design-to-production workflow.
- ◆ The simulation service is private:
 - ◆ It runs on the local network or intra-net
 - ◆ It's self-hosted: configuration and access control are completely customizable.
- ◆ Programs correctly simulated can be automatically deployed to CNC machines.

The architecture



What's new in version 2

- ◆ Extensibility and customization by means of C# scripts.
- ◆ Browser compatibility has been improved (IE 7+, Chrome, Firefox, Safari and Mobile Safari).
- ◆ Better logging and internationalization.
- ◆ Improved stability.

Scripting

- ◆ EurekaCloud fires three events that can be caught by means of a C# script:
 - ◆ *OnSimulationAdd*: fired when a new project is detected in the shared folder;
 - ◆ *OnSimulationEnd*: fired when a simulation has ended correctly;
 - ◆ *OnSimulationAborted*: fired when a simulation ended unexpectedly.
- ◆ Each event handler can access the simulation details and the Eureka instance that executed the simulation:
 - ◆ Save Eureka log messages and reports
 - ◆ Retrieve part programs
 - ◆ Send reports by email
 - ◆ You can leverage the power of C# and .NET framework to accomplish any automation task

A script example

```
public class EurekaCloudScript
{
    /* Called when a simulation is added to the remote folder
    * and it is detected by EurekaCloud. The Simulation s object
    * contains information about the added simulation as remote
    * project folder path, EPF path. Custom parameters can also
    * be added to s (e.g. operator's name, the code of the
    * simulated object, etc.)
    */
    public static Simulation OnAddSimulation(Simulation s)
    {
        Logger.Log("OnAddSimulation");
        s.AddParameter(new SimulationParam("operator", "string",
                                           "John"));

        return s;
    }

    /* Called when a simulation ends. The Simulation s object
    * contains information about the added simulation as its
    * status. The eurekaApp object is a reference to the Eureka
    * instance which executed the simulation. By means of this
    * reference it's possible to retrieve logs, reports, part
    * programs.
    */
    public static bool OnSimulationEnd(Simulation s,
                                       Eureka.Application eurekaApp, string reportLog)
    {
        Logger.Log("OnSimulationEnd");
        return true;
    }

    /* Called when a simulation is aborted
    */
    public static bool OnSimulationAborted(Simulation s)
    {
        Logger.Log("OnSimulationAborted");
        return true;
    }
}
```