
Developing with DREAM Platform

Nexedi 2015-03-26

Agenda

1. Install DREAM Runner
2. Edit DREAM platform source code
3. Run DREAM platform unit tests
4. Accessing DREAM GUI
5. Introduction to JSON format
6. Configuring Graph Editor
7. Configuring Application configuration, adding a spreadsheet tab and a new gadget
8. Configuring Knowledge extraction plugins
9. Cloud Execution of ACO

Install DREAM Runner



Easy way: use vifib hosting

1. Create an account on <https://www.slapos.org/>

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service



The screenshot shows the SLAPOS interface. At the top, there's a teal header bar with the SLAPOS logo (a stylized blue 'S' inside a cloud shape) and the text "SLAPOS". To the right of the logo is a "My Space / Logout" link. Below the header is a navigation menu with four items: "Documentation", "Download", "Forum", and "Blog", separated by vertical lines. The main content area has a white background. On the left, there's some partially visible text starting with "ices". In the center, there's a wooden crate icon containing various small icons (a gear, a lightbulb, etc.). To the right of the icon, the text "My Services" is displayed in a large, bold, teal font. Below this, a smaller text says "This page is dedicated to administration of your services". In the bottom right corner of the main area, there's a teal button with a white checkmark icon and the text "New service".

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”

	Title	
<input type="radio"/>	[EXPERIMENTAL] ERP5	
<input type="radio"/>	[EXPERIMENTAL] ERP5 Test Node	
<input type="radio"/>	[EXPERIMENTAL] Zabbix Agent	
<input type="radio"/>	KVM	
<input type="radio"/>	SlapOS Monitor	
<input checked="" type="radio"/>	SlapOS Web Runner	

Order

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version

<input type="radio"/>	SlapOS Web Runner (Outdated)	slapos-0.244	Production and Commercial version of SlapOS Web Runner
<input type="radio"/>	SlapOS Web Runner	slapos-0.254	Production and Commercial version of SlapOS Web Runner
<input type="radio"/>	SlapOS Web Runner (Outdated)	slapos-0.252.1	Production and Commercial version of SlapOS Web Runner
<input checked="" type="radio"/>	SlapOS Web Runner	slapos-0.258	Production and Commercial version of SlapOS Web Runner

 Order

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter

SlapOS Web Runner
http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/tags/slapos-0.258:/software/slaphandler/software.cfg



Service Title

Parameter XML

```
<?xml version="1.0" encoding="utf-8"?>
<instance>
  <parameter id="slapos-software">software/dream</parameter>
  <parameter id="slapos-reference">dream</parameter>
  <parameter id="auto-deploy">true</parameter>
  <parameter id="user-authorized-key">(your ssh public key)</parameter>
  <parameter id="custom-frontend-backend-url">http://[$(slaphandler:ipv6]":8080</parameter>
</instance>
```

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
 - slapos-software: the software to use, available in slapos git repository, see <http://git.erp5.org/gitweb/slapos.git/tree/HEAD:/software?js=1>
 - slapos-reference: The git branch from slapos repository
 - auto-deploy: Auto deploy the software
 - user-authorized-keys: Your ssh public key, if you want to use ssh
 - custom-frontend-backend-url: This will generate an IPv4 frontend for you if you cannot access it through IPv6.

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear

Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear
7. Click on **URL** parameter

Key	Value
backend_url	https://[2001:67c:1254:e:1c::b4c0]:50005
2_info	In order to set up your account, get the recovery-code from notification on monitor_info.
ssh_command	ssh 2001:67c:1254:e:1c::b4c0 -p 22222
url	https://softinst59074.host.vifib.net
public_url	https://softinst59072.host.vifib.net/public/
webdav_url	https://softinst59072.host.vifib.net/share/
git_public_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git-public/
git_private_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git/
custom-frontend-url	https://softinst59073.host.vifib.net
access_url	https://softinst59074.host.vifib.net/login
monitor_backend_url	https://[2001:67c:1254:e:1c::b4c0]:9684
1_info	On your first run, Use "access_url" to setup you account. Then "backend_url" if you want to use ipv6. Set up your account being able to clone your git repositories from the runner.
monitor_url	https://softinst59072.host.vifib.net



Install DREAM Runner

Easy way: use vifib.com hosting

1. Create an account on <https://www.slapos.org/>
2. Request New Service
3. Choose “SlapOS web runner”
4. Choose latest version
5. Set Service Title and XML Parameter
6. Wait for a while and connection parameter will appear
7. Click on URL parameter
8. ***Pay invoices every month***



Install DREAM Runner

... or Install SlapOS in your IT infrastructure.

Follow the tutorials from:

<http://community.slapos.org/wiki/developer-Allocate.SlapOS.Master.Instance>
<http://community.slapos.org/wiki/developer-Installing.SlapOS.Package>

Install DREAM Runner

Before start, configure monitoring from “Monitoring” link on the slapos page.

1. Set monitoring password

This is the monitoring interface

Please set your password for later access

Password*:

Verify Password*:

Install DREAM Runner

Before start, configure monitoring from “Monitoring” link on the slapos page.

1. Set monitoring password
2. Set recovery code

Monitoring

MONITOR-PUBLIC
rssfeed.html
MONITORING
status-history.cgi
status.cgi
ZERO-KNOWLEDGE
settings.cgi
monitor-password.cgi
cors-domain.cgi

Values that can be defined :

recovery-code	<input type="text" value="uamdntxf"/>
status-history-length	<input type="text" value="5"/>
shell-password	<input type="text" value="flcymqwh"/>

Save



Other values :

Install DREAM Runner



Upon first login, configure your account

Set your name, email, login and password

Enter Password Recovery Code from monitoring interface

Update your account - No account found

Your personal information

Your name:

Your email address:

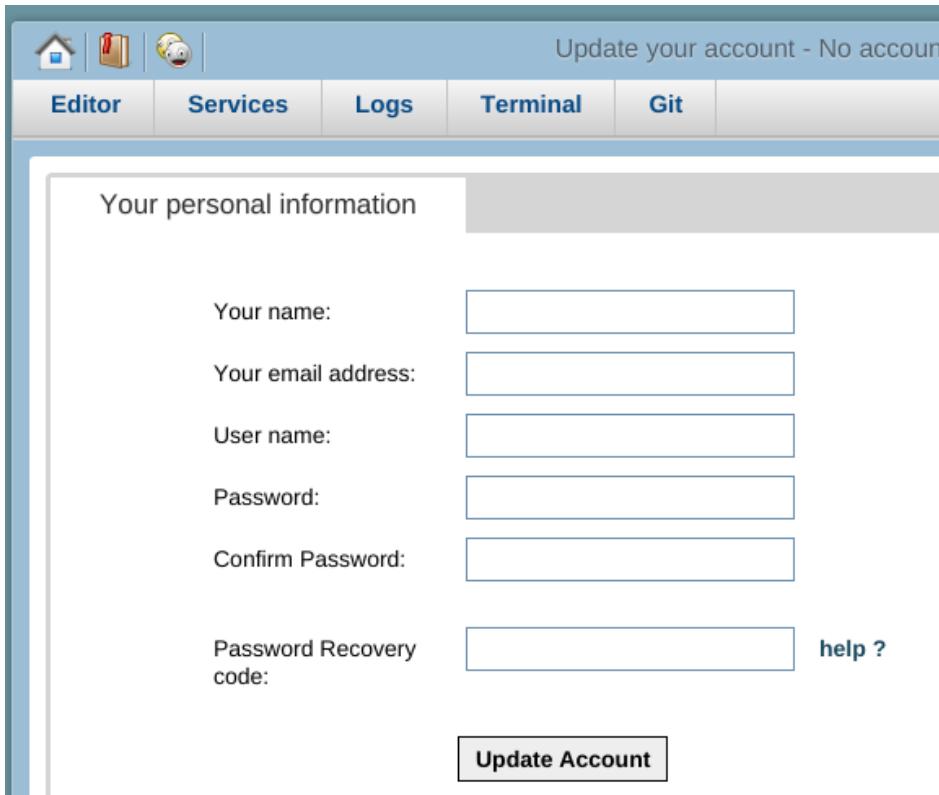
User name:

Password:

Confirm Password:

Password Recovery code: help ?

Update Account





Install DREAM Runner

Click play button to compile software and create instance

View log - dream (workspace/slapos/software)

Editor Services Logs Terminal Git

Log from file instance.log

```
directly instead of attempting to iterate over the result.  
[2015-03-25 16:42:16,763] INFO stacklevel=1,  
[2015-03-25 16:42:16,879] INFO /srv/slapgrid/slappart7/srv/runner/software/580e2a7f4944408226774940b7014ae4/eggs/zc.buildout-1.7.1.post6-py2.7.egg/zc/buildout/easy_install.py:1135: UserWarning: Module argparse was already imported from /opt/slapgrid/f2a5f59d0d2b521681b9333ee76a2859/part/python2.7/lib/python2.7/argparse.pyc, but /srv/slapgrid/slappart7/srv/runner/software/580e2a7f4944408226774940b7014ae4/eggs/argparse-1.3.0-py2.7.egg is being added to sys.path  
[2015-03-25 16:42:16,879] INFO ws.add(dist)  
[2015-03-25 16:42:18,837] INFO Installing directory.  
[2015-03-25 16:42:18,838] INFO Installing dream_simulation.  
[2015-03-25 16:42:18,838] INFO Unused options for dream_simulation: 'parameters-extra' 'command-line' 'wrapper-path'.  
[2015-03-25 16:42:18,854] INFO Installing instance-parameter.  
[2015-03-25 16:42:18,855] INFO Installing dream_platform.  
[2015-03-25 16:42:18,856] INFO Unused options for dream_platform: 'parameters-extra' 'command-line' 'wrapper-path'.  
[2015-03-25 16:42:18,858] INFO Installing dream_test_suite.  
[2015-03-25 16:42:18,858] INFO Unused options for dream_test_suite: 'parameters-extra' 'command-line' 'wrapper-path'.  
[2015-03-25 16:42:18,860] INFO Installing grunt_watch.  
[2015-03-25 16:42:18,861] INFO Unused options for grunt_watch: 'command-line' 'wrapper-path'.  
[2015-03-25 16:42:18,861] INFO Installing nupublish-connection-parameter
```

Select log file to view

Software.log

Instance.log

Install DREAM Runner



Wait for building state and running state to be Complete

The screenshot shows a software interface for managing software builds. At the top right are navigation icons: a blue arrow pointing right, a grey arrow pointing down, and a grid icon. Below these are two main sections: "Building State" and "Running State".

Building State: A green square icon indicates "Complete". Below it, the text "last build: 2015-03-25 16:46:13" is displayed. A descriptive paragraph explains that SlapOS rebuilds software from source, mentioning the ability to patch or add free software, and includes a link to "Learn how!".

Running State: A green square icon indicates "Complete". Below it, the text "last run: 2015-03-25 16:46:18" is displayed. A descriptive paragraph explains that SlapOS configures your system.

Install DREAM Runner

Check your processes are running in “Process” tab.
 From this screen, you can also restart processes.

Instance inspection - dream (workspace/slapos/software)

Editor Services Logs Terminal Git

Process Connection Information Parameters Partitions Content Monitoring

This tab show all process generated by slapgrid for your application. You can click on the process name to display log.

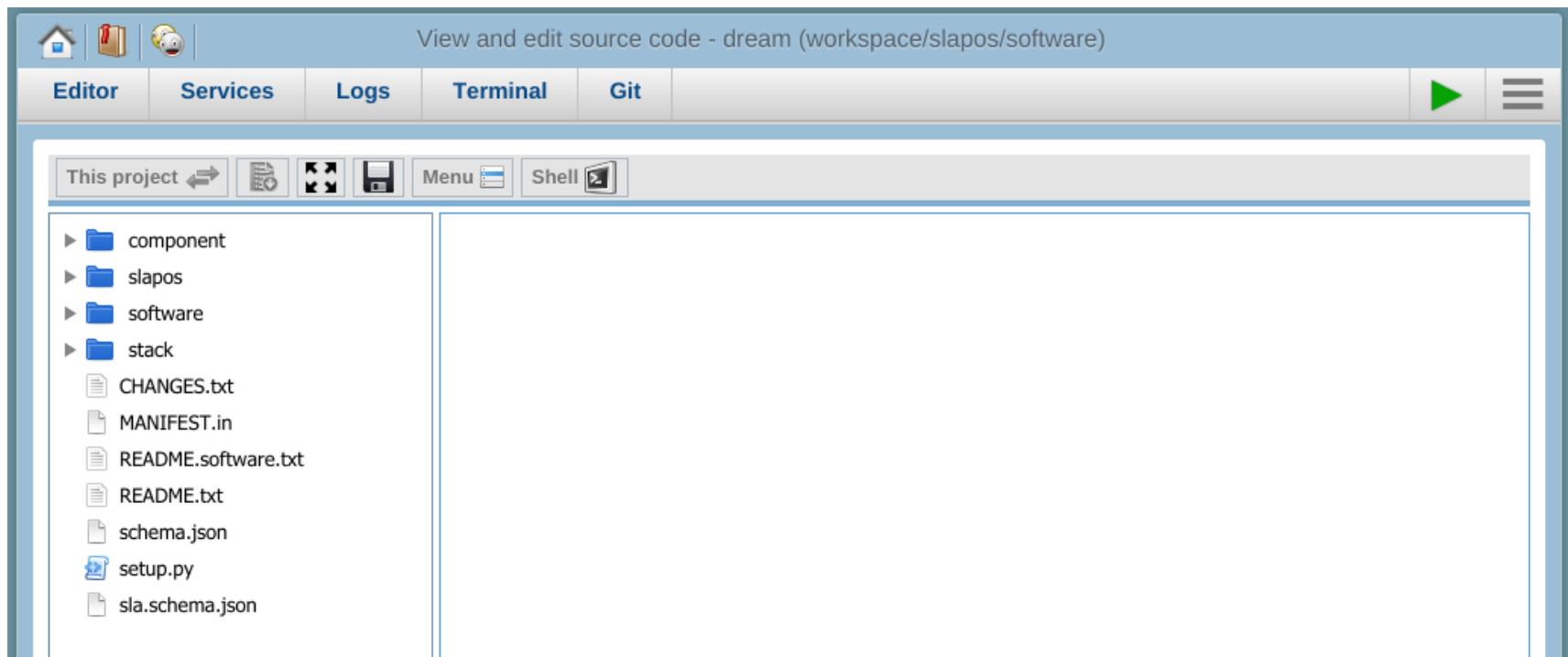
Partition and Process name	Status	Process PID	UpTime	
slappart0:dream_grunt_watch-on-watch	RUNNING	12567	0:04:52	Restart
slappart0:dream_platform-on-watch	RUNNING	12538	0:04:52	Restart
slappart0:dream_simulation	EXITED	25	PM	Restart
slappart0:dream_test_suite	RUNNING	18925	0:00:53	Restart

[Refresh Status](#) [Stop all process](#)



Edit DREAM platform source code

Editor tab let you edit files:





Edit DREAM platform source code

To access DREAM source code, first switch from “This project” to “Working dir”

This project ↗

- component
- slapos
- software
- stack
- CHANGES.txt
- MANIFEST.in
- README.software.txt
- README.txt
- schema.json
- setup.py
- sla.schema.json



Edit DREAM platform source code

DREAM source code is in:

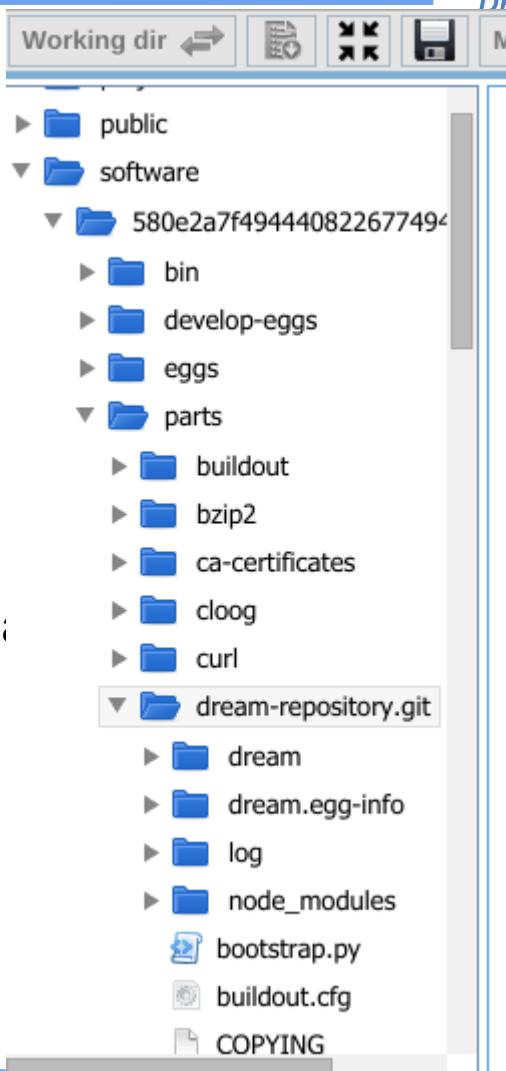
software

(the hash of the software)

parts

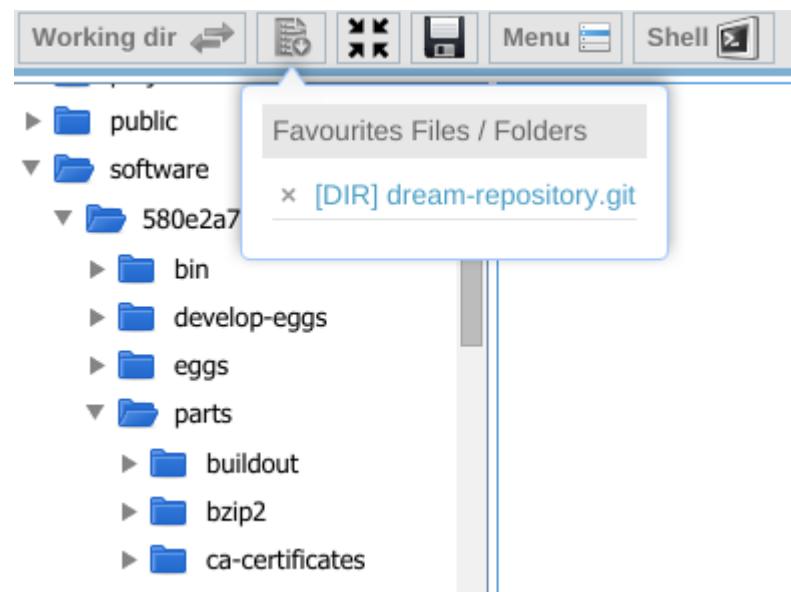
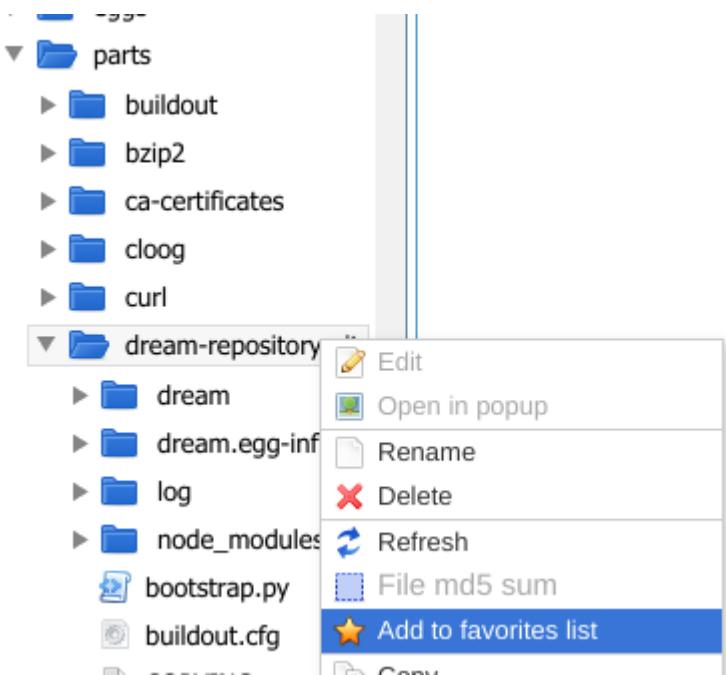
dream-repository.git

This is a git working copy of <http://git.erp5.org/gitweb/dream/>
git/



Edit DREAM platform source code

Tips: add this folder for favourites, it will be available from the favourites menu



Edit DREAM platform source code



ManPy code can be edited :

The screenshot shows a software interface with a title bar "Working dir" and several icons (File, New, Save, etc.). The main area has two panes: a file browser on the left listing Python files like `minTopologies.py`, `Frame.py`, etc., and a code editor on the right displaying the contents of `Machine.py`. The code editor pane has tabs for "Menu", "Shell", and "Machine.py". The code itself is a Python script with comments explaining the license under the GNU Lesser General Public License, version 3. It includes sections for creation date ("Created on 8 Nov 2012") and author ("@author: George"). A specific class definition for "Machine" is shown at the bottom, which "Models a machine that can also have failures".

```
1 # =====
2 # Copyright 2013 University of Limerick
3 #
4 # This file is part of DREAM.
5 #
6 # DREAM is free software: you can redistribute it and/or modify
7 # it under the terms of the GNU Lesser General Public License
8 # the Free Software Foundation, either version 3 of the License,
9 # (at your option) any later version.
10 #
11 # DREAM is distributed in the hope that it will be useful,
12 # but WITHOUT ANY WARRANTY; without even the implied warranty
13 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
14 # GNU Lesser General Public License for more details.
15 #
16 # You should have received a copy of the GNU Lesser General
17 # along with DREAM. If not, see <http://www.gnu.org/licenses/>.
18 # =====
19 ...
20 Created on 8 Nov 2012
21
22 @author: George
23 ...
24 ...
25 Models a machine that can also have failures
26 ...
27
28 # from SimPy.Simulation import Process, Resource, SimEvent
29 # from SimPy.Simulation import activate, deactivate, waitin
```

Running DREAM test suite

After making changes, run tests to validate your changes

Process	Connection Information	Parameters	Partitions Content	Monitoring
This tab show all process generated by slapgrid for your application. You can click on the process name to display log.				
Partition and Process name	Status	Process PID	UpTime	
slappart0:dream_grunt_watch-on-watch	RUNNING	12567	0:23:24	Restart
slappart0:dream_platform-on-watch	RUNNING	12538	0:23:24	Restart
slappart0:dream_simulation	EXITED	25	PM	Restart
slappart0:dream_test_suite	EXITED	25		Restart

Refresh Status | **Stop all process**

Running DREAM test suite

After making changes, run tests to validate your changes

Process
Connection Information
Parameters
Partitions Content
Monitoring

This tab shows all processes generated by slapgrid for your application. You can click on the process name to display log.

Partition and Process name	Status	Process PID	UpTime	
slappart0:dream_grunt_watch-on-watch	RUNNING	12567	0:23:24	Restart
slappart0:dream_platform-on-watch	RUNNING	12538	0:23:24	Restart
slappart0:dream_simulation	EXITED	25	PM	Restart
slappart0:dream_test_suite	EXITED	25	PM	Restart

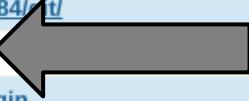
Refresh Status
Stop all process

2. Click on the process name to view the output

Accessing DREAM GUI

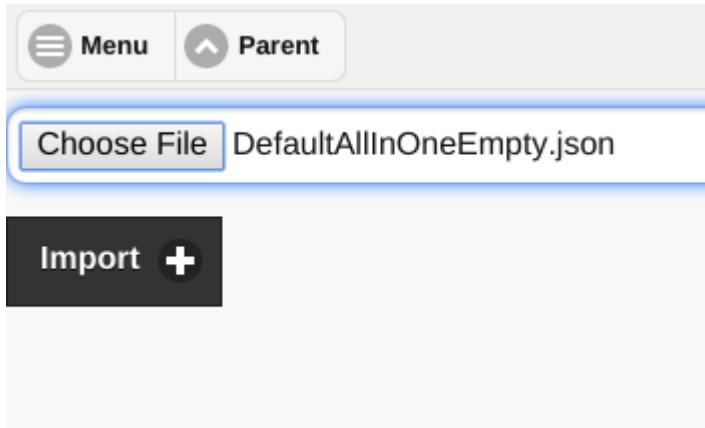
The GUI instance can be accessed from ***custom frontend URL*** in SlapOS parameter page.

Key	Value
backend_url	https://[2001:67c:1254:e:1c::b4c0]:50005
2_info	In order to set up your account, get the recovery-code from the monitoring interface. Before read the notification on monitor_info.
ssh_command	ssh 2001:67c:1254:e:1c::b4c0 -p 22222
url	https://softinst59074.host.vifib.net
public_url	https://softinst59072.host.vifib.net/public/
webdav_url	https://softinst59072.host.vifib.net/share/
git_public_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git-public/
git_private_url	https://[2001:67c:1254:e:1c::b4c0]:9684/git/
custom-frontend-url	https://softinst59073.host.vifib.net
access_url	https://softinst59074.host.vifib.net/login
monitor_backend_url	https://[2001:67c:1254:e:1c::b4c0]:9684
1_info	On your first run, Use "access_url" to setup you account. Then you can use both "url" or "access_url". Or "backend_url" if you want to use ipv6. Set up your account in the webrunner in order to use webdav, and being able to clone your git repositories from the runner.
monitor_url	https://softinst59072.host.vifib.net



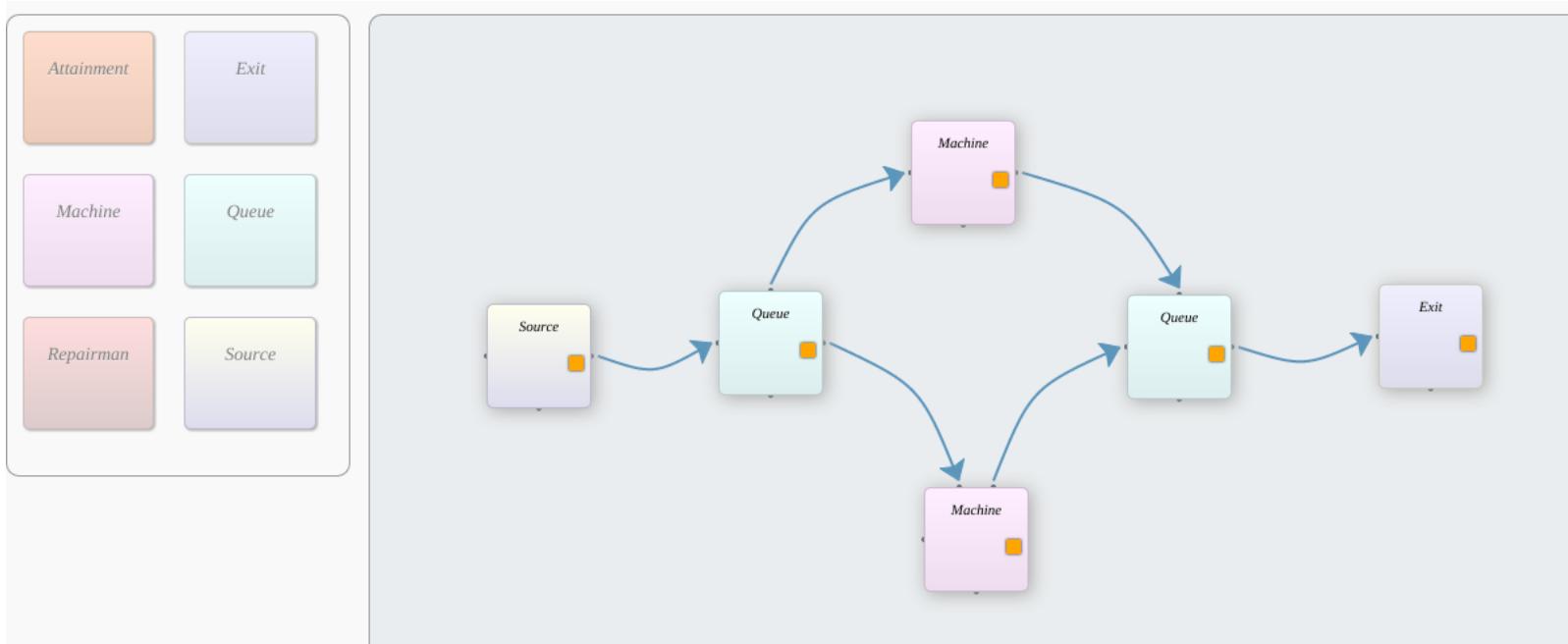
Accessing DREAM GUI

Load a model, for example https://raw.githubusercontent.com/nexedi/dream/master/dream/simulation/Examples/GUI_instances/DefaultAllInOneEmpty.json



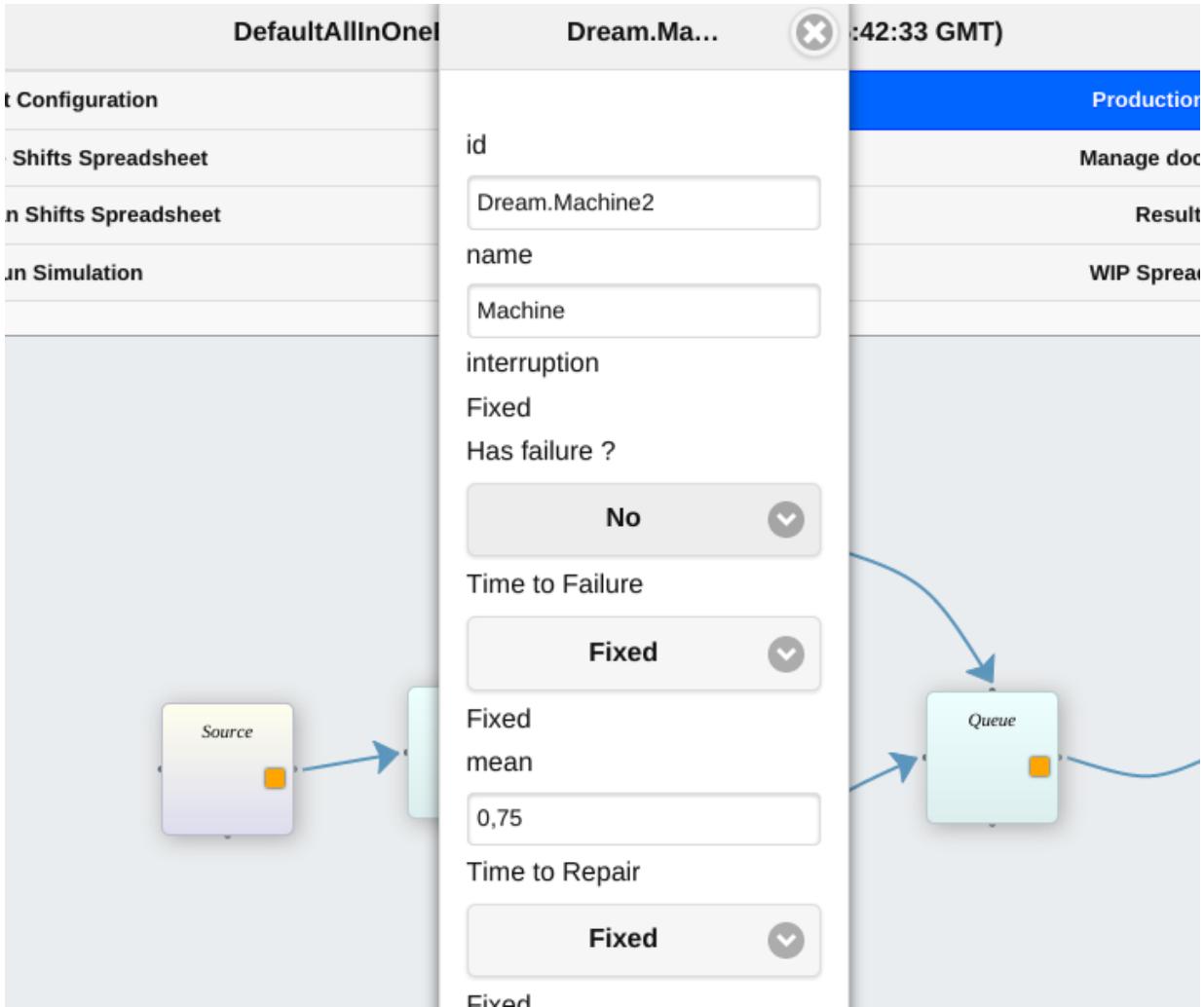
Accessing DREAM GUI

Create a simple model



Accessing DREAM GUI

Edit model properties



The screenshot shows the DREAM GUI interface. On the left, there is a navigation sidebar with links like 'DefaultAllInOne', 'Shifts Configuration', 'Shifts Spreadsheet', 'Run Shifts Spreadsheet', and 'Run Simulation'. The main area displays a process flow diagram with a 'Source' node connected to a 'Queue' node. The 'Queue' node has two outgoing arrows, one pointing down and one pointing right. To the right of the diagram is a configuration panel for 'Dream.Machine2'.

Property	Value
id	Dream.Machine2
name	Machine
interruption	Fixed
Has failure ?	No
Time to Failure	Fixed
mean	0,75
Time to Repair	Fixed
mean	5,00

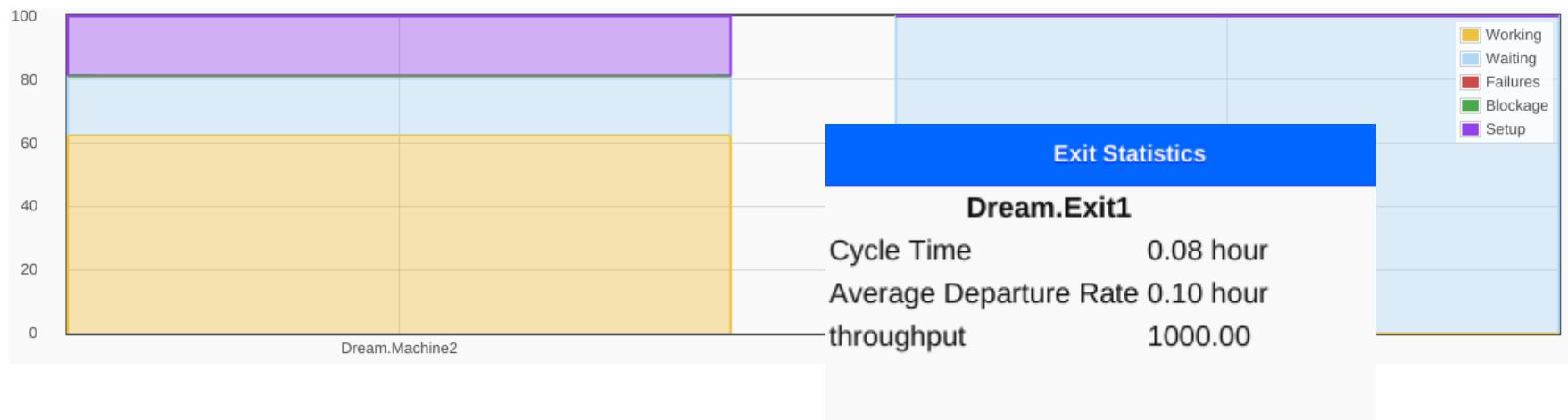
Accessing DREAM GUI

Running simulation

Machine Shifts Spreadsheet	
Repairman Shifts Spreadsheet	
Run Simulation	
Run Simulation C	
Confidence level for statistical analysis of stochastic experiments	
0,95	
The day the experiment starts, in YYYY/MM/DD HH:MM:SS format	
2015/01/01 09:00:00	
The URL for knowledge extraction to access its data for example http://git.erp5.org/gitweb/dream.git/blob_plain/HEAD:/drear	
Length of the simulation run	
100	

Accessing DREAM GUI

Visualise results



Exit Statistics		Queue Statistics		Station Utilization		Tabular_Results
Exit Id	Throughput	Takt Time		Lifespan		
Dream.Exit1	1000	0.099899999999986		0.08125000000000017		

Introduction to JSON format

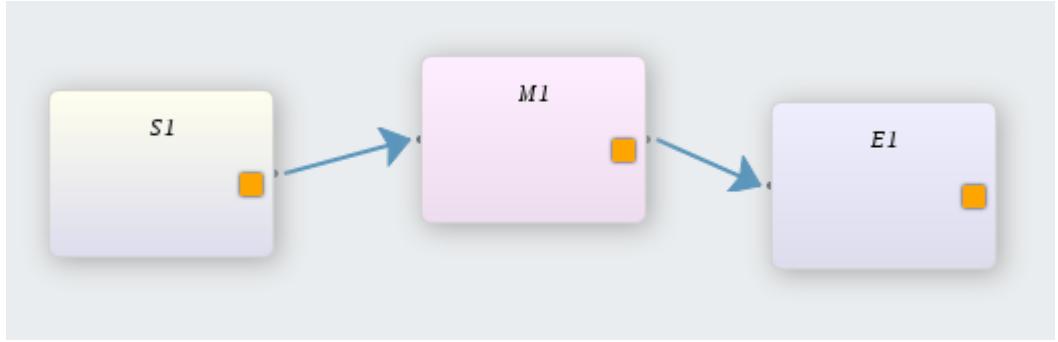


Simple Representation of a Directed Graph

Only nodes

```
{
  "node": {
    "S1": {
      "name": "S1"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": {}
}
```

Introduction to JSON format



Nodes connected by edges

```

{
  "node": {
    "S1": {
      "name": "S1"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": {
    "edge1": {
      "source": "S1",
      "destination": "M1",
    },
    "edge2": {
      "source": "M1",
      "destination": "E1",
    }
  }
}
  
```

Introduction to JSON format



Properties of nodes

```
{
  "node": {
    "S1": {
      "name": "Parts",
      "_class": "Dream.Source",
      "interarrivalTime": {
        "distributionType": "Fixed",
        "mean": 0.5
      },
      "entity": "Dream.Part"
    },
    "M1": {
      "name": "M1"
    },
    "E1": {
      "name": "E1"
    }
  },
  "edge": {
    "edge1": {
      "source": "S1",
      "destination": "M1",
    },
    "edge2": {
      "source": "M1",
      "destination": "E1",
    }
  }
}
```

Introduction to JSON format

Dream.So...

ID	S1
Name	Parts
Interarrival Time	
Distribution Type	Fixed
Mean	0.5
Standard Deviation	
Minimum Value	
Maximum Value	
Entity Class	Dream.Part

- What are the editable properties ?
- Which value type (text, number) ?
- What possible values ?
- What default value ?



Introduction to JSON format



Properties are defined as JSON Schema

Dream.So...

ID	S1
Name	Parts
Interarrival Time	
Distribution Type	Fixed
Mean	0.5
Standard Deviation	
Minimum Value	
Maximum Value	
Entity Class	Dream.Part

```
"class_definition": {  
    "Dream.Source" : {  
        "properties" : {  
            "id" : {  
                "type" : "string",  
                "default" : "S",  
                "required" : true,  
                "name": "ID",  
                "description": "The ID of this source"  
            },  
            "name" : {  
                "type" : "string",  
                "default" : "Source"  
            },  
            "interarrivalTime" : {  
                "type": "object",  
                "properties": {  
                    "distributionType": {  
                        "type": "string",  
                        "enum": [ "Fixed",  
                                "Normal",  
                                "Exponential"],  
                        "default": "Fixed"  
                    },  
                    "mean": {  
                        "type": "number",  
                        "default": 0.5  
                    }  
                }  
            }  
        }  
    }  
}
```



Introduction to JSON format

Dream.So...

ID
S1

Name
Parts

Interarrival Time

Distribution Type
Fixed

Mean
0.5

Standard Deviation

Minimum Value

Maximum Value

Entity Class
Dream.Part

```

"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
                    }
                }
            }
        }
    }
}

```



Introduction to JSON format

Dream.So...

ID
S1

Name
Parts

Interarrival Time

Distribution Type
Fixed

Mean
0.5

Standard Deviation

Minimum Value

Maximum Value

Entity Class
Dream.Part

```

"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
                    }
                }
            }
        }
    }
}

```



Introduction to JSON format

Dream.So...

ID
S1

Name
Parts

Interarrival Time

Distribution Type
Fixed

Mean
0.5

Standard Deviation

Minimum Value

Maximum Value

Entity Class
Dream.Part

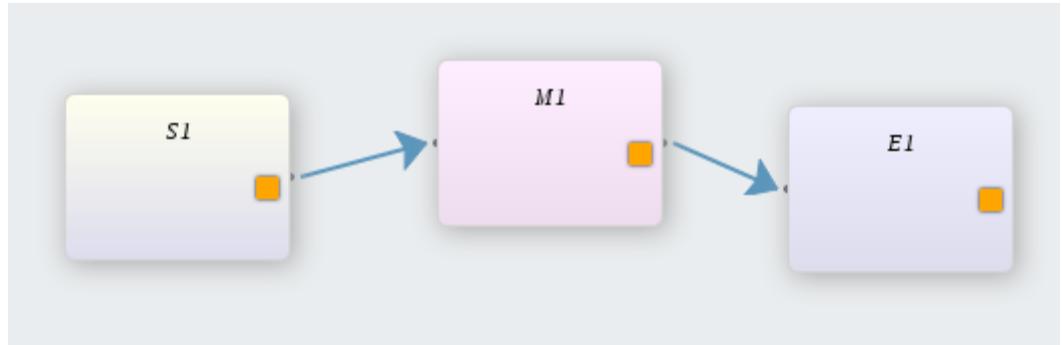
```

"class_definition": {
    "Dream.Source" : {
        "properties" : {
            "id" : {
                "type" : "string",
                "default" : "S",
                "required" : true,
                "name": "ID",
                "description": "The ID of this source"
            },
            "name" : {
                "type" : "string",
                "default" : "Source"
            },
            "interarrivalTime" : {
                "type": "object",
                "properties": {
                    "distributionType": {
                        "type": "string",
                        "enum": [ "Fixed",
                                  "Normal",
                                  "Exponential"],
                        "default": "Fixed"
                    },
                    "mean": {
                        "type": "number",
                        "default": 0.5
                    }
                }
            }
        }
    }
}

```



Introduction to JSON format



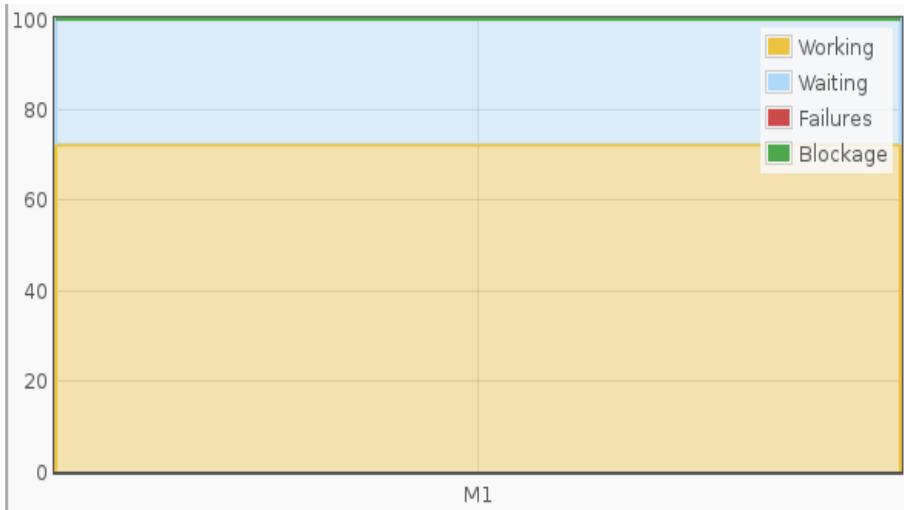
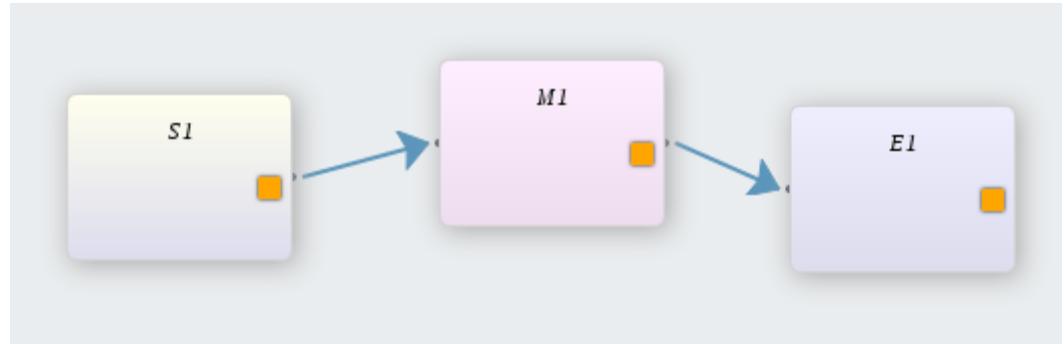
Order ID	Due Date	Priority	Project Manager	Part	Part Type	Sequence		
Order 1	2014/03/15	1	PM1	Design	Design	CAD		
				Part1	Basic	CAM-MILL-EDM-MILL-MASS		
				Part2	Basic	CAM-MILL-EDM-MILL-EDM-MAS		
				Assemble	Mould	MASS-IM	2-12	
Order 2	2014/03/14	1	PM1	Design	Design	CAD	6	
				Part1	Basic	CAM-MILL-EDM-MILL-MASS	8-4-2-8-0	
				Part2	Basic	CAM-MILL-EDM-MASS	20-15-8-8-0	
				Assemble	Mould	MASS-IM	1-12	
Order 3	2014/03/15	1	PM1	Design	Design	CAD	6	
				Part1	Basic	CAM-MILL-EDM-MASS	8-4-2-0	
				Part2	Basic	CAM-MILL-EDM-MASS	7-15-8-8-0	
				Assemble	Mould	MASS-IM	1-3	

```

{
  "graph": {
    "node": { ... },
    "edge": { ... }
  },
  "class_definition": { ... },
  "input": [
    "WipSpreadsheet" : [
      ["Order 1", "2014/03/15",
      "1", "PM1", "Design" ... ],
      ...
    ]
  ],
  ...
}
  
```

General purpose tabular inputs in the same JSON

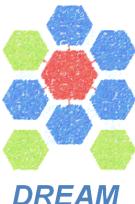
Introduction to JSON format



```
{
  "graph": {
    "node": { ... },
    "edge": { ... }
  },
  "class_definition": { ... },
  "input": { ... },
  "output": {
    "M1": {
      "working_ratio": 0.7,
      "waiting_ratio": 0.3,
      "failure_ratio": 0,
      "blockage_ratio": 0
    }
  }
}
```

Outputs - results of simulation - are also in the same JSON

Introducing Edit Configuration tab



Live edition of JSON

Save C

```
        "description": "Confidence level for statistical analysis of stochastic experiments",
        "title": "Confidence level",
        "type": "number"
    },
    "currentDate": {
        "default": "2014/10/01 09:00:00",
        "description": "The day the experiment starts, in YYYY/MM/DD HH:MM:SS format",
        "title": "SimulationStartTime",
        "type": "string"
    },
    "ke_url": {
        "default": "http://git.erp5.org/gitweb/dream.git/blob_plain/HEAD:/dream/KnowledgeExtraction/Mockup",
        "description": "The URL for knowledge extraction to access its data for example http://git.erp5.org",
        "title": "URL for Knowledge Extraction Spreadsheet",
        "type": "string"
    },
    "maxSimTime": {
        "default": 100,
        "description": "Length of the simulation run",
        "title": "Length of Experiment",
        "type": "number"
    }
}
```

Introducing Edit Configuration tab

Live edition of JSON

Features non obtrusive assistance v

Save C

```
1▼ 2▼ 3▼ 4▼ 5▼ { "application_configuration": { "general": { "npronerties": { 6▼ ...}, ERROR 7▼ -----^ 8▼ Expecting 'STRING', got 'undefined' 9▼ }}, ERROR 10▼ 11▼ 12▼ 13▼ 14▼ 15▼ "currentDate": { "default": "2014/10/01 09:00:00", "description": "The day the experiment st 16▼ "title": "SimulationStartTime". 17▼ 18▼ }
```

Parse error on line 10:
-----^
Expecting 'STRING', got 'undefined'

for stat

Graph editor: adding a property on node

On Edit configuration tab:

Locate the json schema definition of Dream.Queue in *class_definition*

```

1▼   {
2►     "application_configuration": { },
3►     "class_definition": {
4►       "Dream.Edge": { },
5►       "Dream.EventGenerator": { },
6►       "Dream.Exit": { },
7►       "Dream.Machine": { },
8►       "Dream.Queue": {
9►         "_class": "node",
10►         "allof": [
11►           {
12►             "$ref": "#/node"
13►           },
14►           {
15►             "properties": {
16►               "capacity": {
17►                 "$ref": "#/definitions/_capacity",
18►                 "required": true
19►               },
20►               "id": {
21►                 "default": "Q",
22►                 "type": "string"
23►               }
24►             }
25►           }
26►         ]
27►       }
28►     }
29►   }
30► }
```

Graph editor: adding a property on node

On Edit configuration tab:

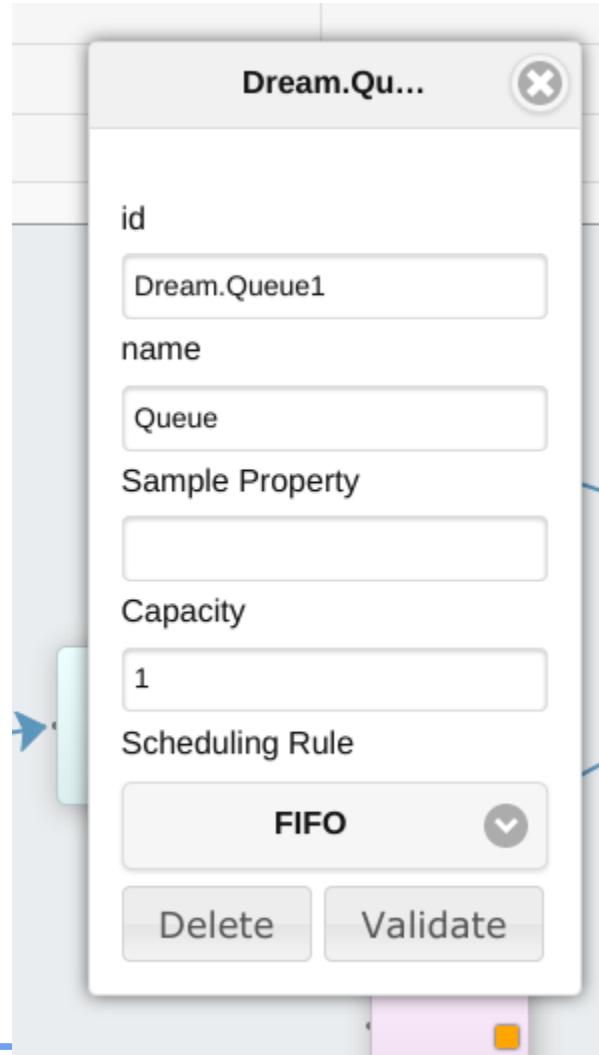
Add a new property, in this example it is called “Sample property”

```
264▶ "Dream.EventGenerator": {↔},
329▶ "Dream.Exit": {↔},
358▶ "Dream.Machine": {↔},
433▼ "Dream.Queue": {
434    "_class": "node",
435    "allOf": [
436      {
437        "$ref": "#/node"
438      },
439      {
440        "properties": {
441          "sample_property": {
442            "name": "Sample Property",
443            "description": "The tooltip to display.",
444            "type": "string"
445          },
446          "capacity": {
447            "$ref": "#/definitions/_capacity",
448            "required": true
449          },
450          "id": {
451            "default": "Q",
452            "type": "string"
453        }
454      }
455    ]
456  }
457  "capacity": {
458    "$ref": "#/definitions/_capacity",
459    "required": true
460  },
461  "id": {
462    "default": "Q",
463    "type": "string"
464  }
465}
```

Graph editor: adding a property on node

On “Production Line” tab, we have a new property when editing a Dream.Queue.

Set value “user value” and save



Graph editor: adding a property on node

Go back to “Edit configuration” tab, we can see that the sample_property is saved as a property on the node.

```
Search: sample_property (Use /re/ syntax for regexp search)
254 ►   "application_configuration": { },
916   "class_definition": { },
917   "constraints": {},
918   "general": { },
928 ▼   "graph": {
929 ►     "edge": { },
961 ▼     "node": {
962 ►       "Dream.Source1": { },
978 ▼       "Dream.Queue1": {
979 ►         "coordinate": {
980           "left": 0.26706827309236947,
981           "top": 0.3948110549926837
982         },
983         "_class": "Dream.Queue",
984         "name": "Queue",
985         "id": "Dream.Queue1",
986         "sample_property": "user value",
987         "capacity": 1,
988         "schedulingRule": "FIFO"
989       },
990 ▼       "Dream.Machine1": {
991         "coordinate": { }
```

Application configuration: adding a tab

On Edit configuration tab:

Locate the input action definition in *application_definition*

input is for input action

output is for results visualisation

```

1 ▼
2 ▼
3 ▶
73 ▼
74 ▼
75
76
77
78
79 ▼
80
81
82
83
84 ▼
85 ▼
86 ▼
87 ▼
88
89
90
91
92 ▼
{
  "application_configuration": {
    "general": { },
    "input": {
      "debug": {
        "gadget": "Input_viewDebugJson",
        "title": "Edit Configuration",
        "type": "object_view"
      },
      "view": {
        "gadget": "Input_viewProductionLine",
        "title": "Production Line",
        "type": "object_view"
      }
    },
    "view_machine_shift_spreadsheet": {
      "configuration": {
        "columns": [
          {
            "format": "date-time",
            "name": "Date",
            "type": "string"
          }
        ]
      }
    }
  }
}

```

Application configuration: adding a tab

Add an action:

gadget: The gadget to use

title: The name to display on the tab

type: Always `object_view` for now

configuration: Gadget configuration, passed to gadget constructor.

columns: columns definition

input_id: key for this input

```

    },
    },
    "input": {
      "sample_spreadsheet": {
        "gadget": "Input_viewSpreadsheet",
        "title": "Sample Spreadsheet",
        "type": "object_view",
        "configuration": {
          "input_id": "sample_spreadsheet_input",
          "columns": [
            {
              "name": "Sample Column",
              "type": "string"
            }
          ]
        }
      },
      "debug": {
        ...
      }
    }
  }
}

```

Application configuration: adding a tab

The new tab appears. Set some values

Sample Spreadsheet
Production Line
Manage document
Results
WIP Spreadsheet
Sample Column
value 1
value 2
value 3

Application configuration: adding a tab

The values are saved in the JSON, as input / (the input_id configured)

```
267 ►   "class_definition": { ↗ },
929
930 ►   "constraints": {},
941 ►   "general": { ↗ },
941 ►   "graph": { ↗ },
1079 ▼   "input": {
1080 ►     "repairman_shift_spreadsheet": [ ↗ ],
1094 ►     "wip_spreadsheet": [ ↗ ],
1106 ►     "machine_shift_spreadsheet": [ ↗ ],
1130 ▼     "sample_spreadsheet_input": [
1131 ▼       {
1132         "name": "Sample Column"
1133       },
1134 ▼       {
1135         "name": "value 1"
1136       },
1137 ▼       {
1138         "name": "value 2"
1139       },
1140 ▼       {
1141         "name": "value 3"
1142       },
1143 ▼       {
1144         "name": null

```

Application configuration: adding a tab



To add a completely new gadget, locate the gadget folder in dream-repository.
`git/dream/platform/src/dream`

Working dir Menu Machine.py Input_viewAttachDocu

dream-repository.git

dream

KnowledgeExtraction

platform

src

dream

favicon.ico

index.html

index.js

index.less

Input_viewAttachDocu

Input_viewAvailableCa

Input_viewAvailableCa

Input_viewDebugJson.

Input_viewDebugJson.

Input_viewDemandPla

```
4  :     initGadgetMixin) {
5  :   "use strict";
6  :
7  :   var gadget_klass = rJS(window);
8  :   initGadgetMixin(gadget_klass);
9  :   gadget_klass
10 :   ///////////////
11 :   // Acquired methods
12 :   ///////////////
13 :   .declareAcquiredMethod("aq_putAttachment", "jio_putAttachment")
14 :   .declareAcquiredMethod("aq_getAttachment", "jio_getAttachment")
15 :
16 :   ///////////////
17 :   // declared methods
18 :   ///////////////
19 :   .declareMethod("startService", function () {
20 :     var gadget = this,
21 :         encoded_data,
22 :         button;
23 :     return new RSVP.Queue()
24 :       .push(function () {
25 :         return promiseEventListener(
26 :           gadget.props.element.getElementsByClassName("attach_document")[0],
27 :           'submit',
28 :           false
29 :         );
30 :       })
31 :       .push(function (evt) {
32 :         button = evt.target.getElementsByClassName("ui-btn")[0];
33 :         button.disabled = true;
34 :     });
35 :   });
36 : 
```

Application configuration: adding a tab

- Clone existing gadget:
 - html file, usually Input_viewXXXGadget.html
 - javascript file, usually Input_viewXXXGadget.js
- Make sure html file reference the javascript properly
- Debugging tips:
 - code you edit is in dream/platform/src/
 - grunt compiles code in dream/platform/static
 - if syntax error, files in static are not updated !
 - static files have different lines numbers
- use console.log (better with firebug)
- <http://learn.renderjs.org/docs/index.html>



Pre/Post Processing plugins

JSON data goes through :

- pre processing plugins
- simulation engine
- post processing plugins



Pre/Post Processing plugins

Plugins configured in JSON, under *application_configuration*

```

1 ▾
2 ▾
3 ▾
73 ▾
186 ▾
231 ▾
232
233 ▾
234 ▾
235
236
237 ▾
238
239
240
241
242 ▾
243
244
245
246
247
248 ▾
  "application_configuration": {
    "general": { },
    "input": { },
    "output": { },
    "post_processing": {
      "description": "",
      "plugin_list": [
        {
          "_class": "dream.plugins.DefaultTabularExit.DefaultTabularExit"
        },
        {
          "_class": "dream.plugins.PostProcessStationUtilization.PostProcessStationUtilization",
          "family": "Server",
          "output_id": "station_utilization" Parameters for the plugin
        },
        {
          "_class": "dream.plugins.PostProcessQueueStatistics.PostProcessQueueStatistics",
          "output_id": "queue_statistics"
        }
      ],
      "pre_processing": {
        "description": ""
      }
    }
  }
}

```

Python class for plugin

Parameters for the plugin

Pre/Post Processing plugins



Plugins are python class, interface is postprocess(self, data) or preprocess(self, data)

The image shows a code editor interface. On the left, there is a sidebar containing a list of Python files with blue icons next to their names. On the right, the main pane displays the source code for the file `PostProcessStationUtilization.py`.

```
1 from dream.plugins import plugin
2
3 class PostProcessStationUtilization(plugin.OutputPreparationPlugin):
4     """ Output the station utilization metrics in a format compatible with
5     """
6
7     def postprocess(self, data):
8         result = data['result']['result_list'][-1]
9
10        ticks = []
11        working_data = []
12        waiting_data = []
13        failure_data = []
14        blockage_data = []
15        setup_data = []
16
17        options = {
18            "xaxis": {
19                "minTickSize": 1,
20                "ticks": ticks|
21            },
22            "yaxis": {
23                "max": 100
24            },
25            "series": {
26                "bars": {
27                    "show": True,
28                    "barWidth": 0.8,
29                    "align": "center"
30                }
31            }
32        }
33
34    
```

Pre/Post Processing plugins

Plugin configuration defined in json is in self.configuration_dict

```

49     "shape": "Setup",
50     "data": setup_data
51   }];
52
53   out = result[self.configuration_dict['output_id']] = {
54     "series": series,
55     "options": options
56   }
57
58   i = 0
59   for obj in result['elementList']:
60     if obj.get('family') == self.configuration_dict.get('family'):
61       if obj['results']['working_ratio']:
62         working_data.append((i, obj['results']['working_ratio'][0]))
63       if obj['results']['waiting_ratio']:
64         waiting_data.append((i, obj['results']['waiting_ratio'][0]))
65       if obj['results']['failure_ratio']:
66         failure_data.append((i, obj['results']['failure_ratio'][0]))
67       if obj['results']['blockage_ratio']:
68         blockage_data.append((i, obj['results']['blockage_ratio'][0]))
69       if obj['results']['setup_ratio']:
70         setup_data.append((i, obj['results']['setup_ratio'][0]))
71
72       ticks.append((i, obj.get('name', obj['id'])))
73       i += 1
74
75   return data

```

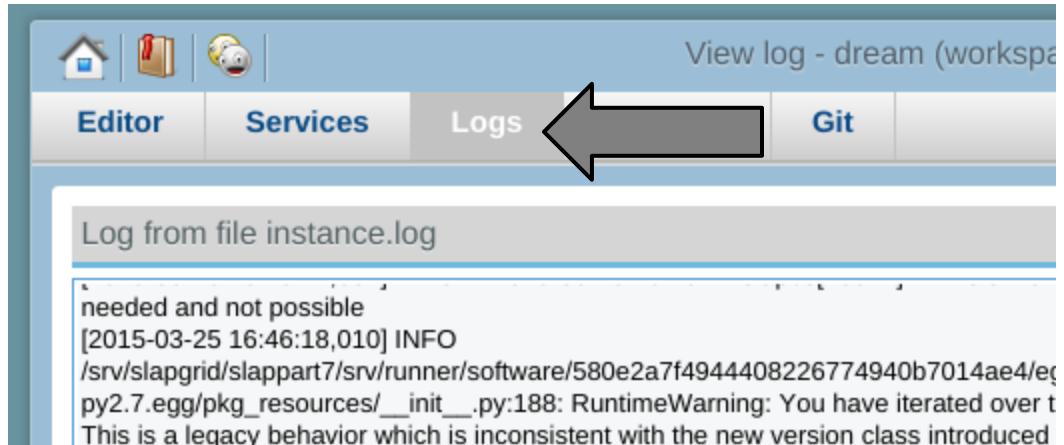
Pre/Post Processing plugins

Plugins debugging tips:

- Use existing Knowledge Extraction Tool objects (from dream/KnowledgeExtraction/ folder)
- Use dream.plugin.TimeSupport to convert from and to simulation clock time
- Log using self.logger.info (standard python logger <https://docs.python.org/2/library/logging.html>)
- Use dream.plugin.Debug in the chain to dump the json
- Write unit tests (examples in dream/tests/testGUIModels.py)

Accessing the application logs

Add a file in log panel (1/2)

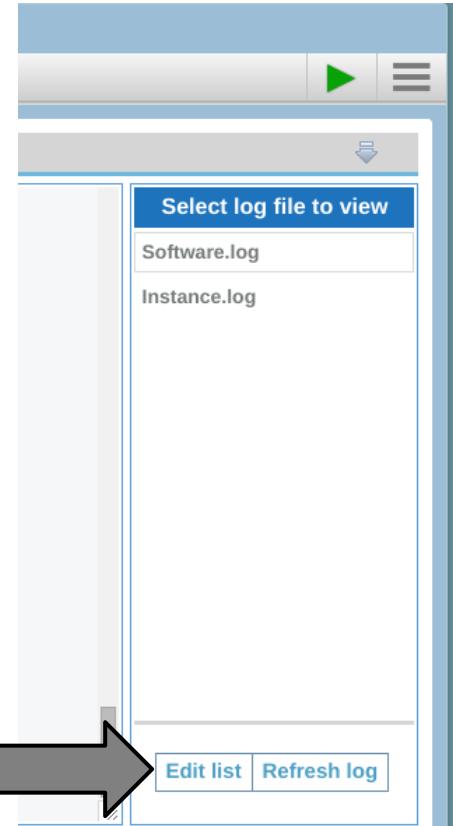


View log - dream (workspace)

Editor Services Logs Git

Log from file instance.log

```
needed and not possible
[2015-03-25 16:46:18,010] INFO
/srv/slappgrid/slappart7/srv/runner/software/580e2a7f4944408226774940b7014ae4/equ
py2.7.egg/pkg_resources/_init__.py:188: RuntimeWarning: You have iterated over t
This is a legacy behavior which is inconsistent with the new version class introduced
```



Select log file to view

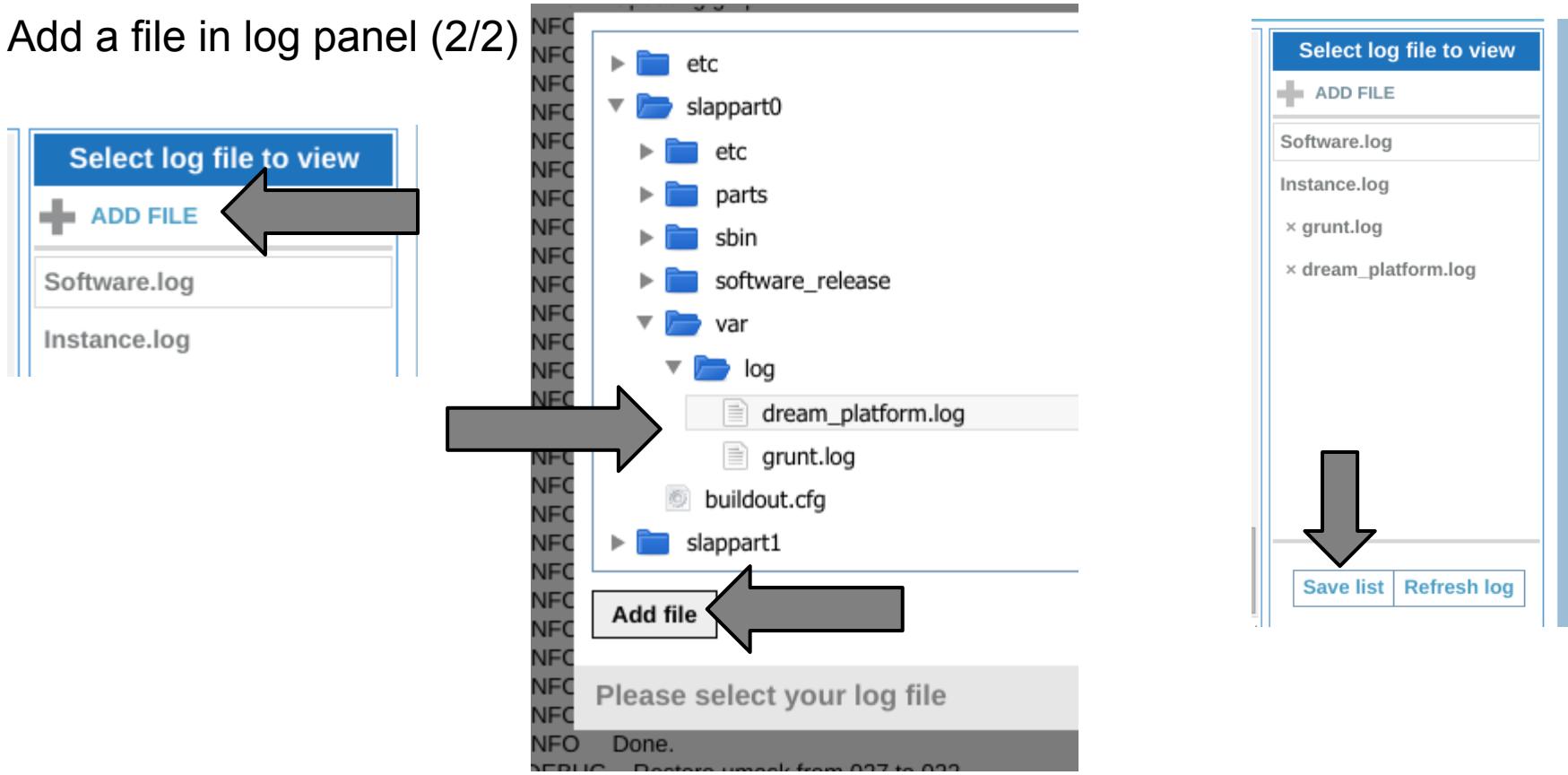
Software.log

Instance.log

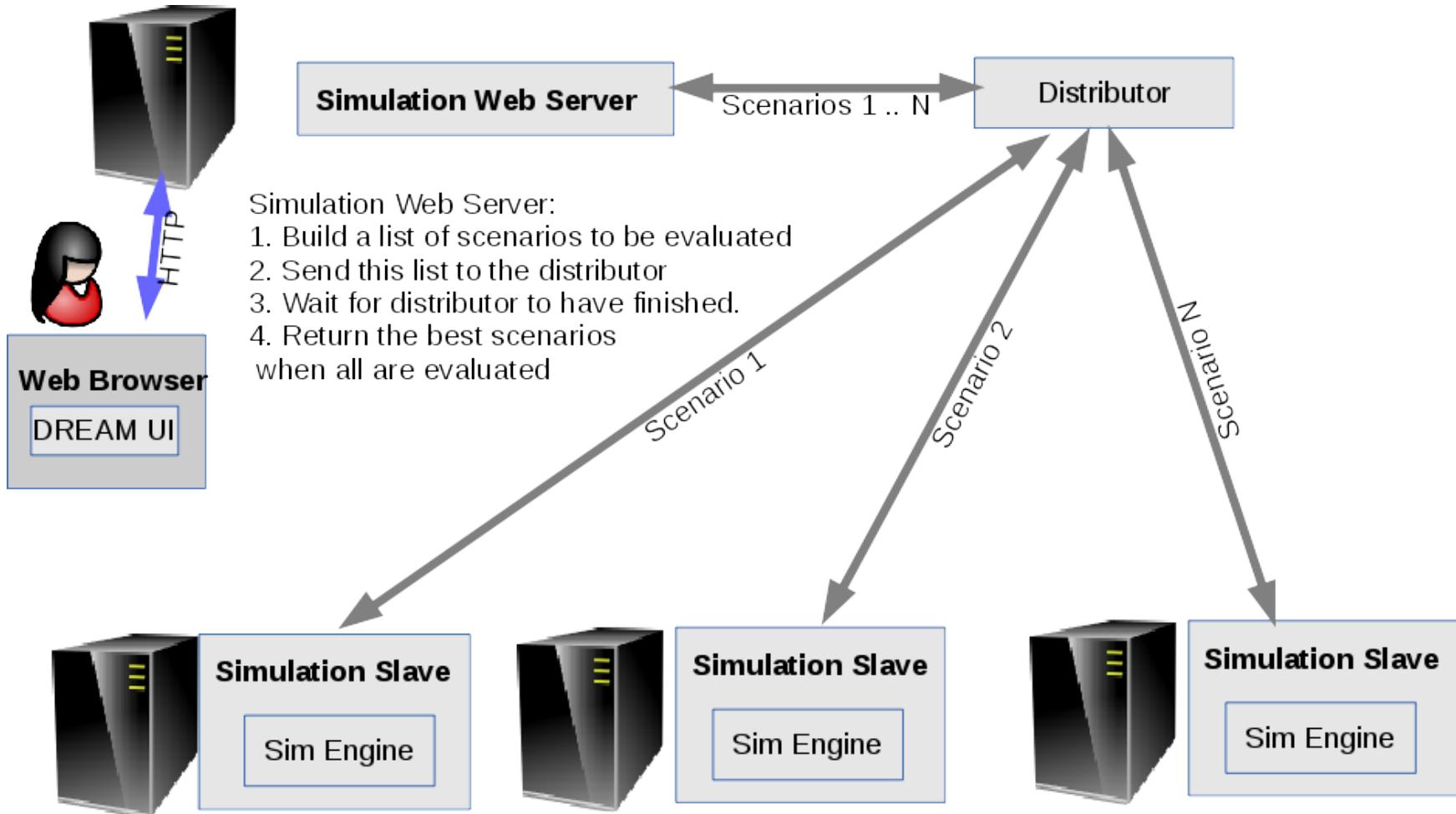
Edit list Refresh log

Accessing the application logs

Add a file in log panel (2/2)



Cloud Execution



Cloud Execution

- Install DREAM distributor [http://git.erp5.org/gitweb/slapos.
git/blob_plain/refs/heads/dream:/software/erp5/software.cfg](http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/dream:/software/erp5/software.cfg)
- Note the custom frontend URL for later reference
- Install some DREAM simulation nodes [http://git.erp5.org/gitweb/slapos.
git/blob_plain/refs/heads/dream:/software/testnode/software.cfg](http://git.erp5.org/gitweb/slapos.git/blob_plain/refs/heads/dream:/software/testnode/software.cfg) set the distributor URL as “test-suite-master-url” parameter. Or using the following XML Parameter for web runner:

```
<instance>
  <parameter id="parameter-test-node-title">Simulation Node 1</parameter>
  <parameter id="parameter-test-suite-master-url">https://login:
password@softinstXXX.host.vifib.
net/erp5/portal_task_distribution/dream_distributor</parameter>
  <parameter id="slapos-software">software/erp5testnode</parameter>
  <parameter id="slapos-reference">dream</parameter>
</instance>
```

Cloud Execution

Be sure to use dream.plugin.ACO.ACO as processing plugin class in your model

```

763▶   "output": { },
839▼   "pre_processing": {
840▼     "plugin_list": [
841▼       {
842         "_class": "dream.plugins.Debug.Debug",
843         "argument": "Argument Value"
844       },
845▼       {
846         "_class": "dream.plugins.OldStylePartJobShopWIP.OldStylePartJobShopWIP",
847         "input_id": "old_style_part_jobshop_spreadsheet"
848       }
849     ]
850   },
851▼   "processing_plugin": {
852     "_class": "dream.plugins.ACO.ACO"
853   },
854▼   "post_processing": {
855     "plugin_list": []
856   },
857▼   "general": {
858     "properties": {
859       "numberOfReplications": {
860         "normal": "Number of replications"
861       }
862     }
863   }
864 }
```



Cloud Execution

Be sure to specify the distributor URL in general properties

Seed for random number generator

Number of generations

Number of ants per generation

Number of solutions

Distributor URL

Number of simulation nodes

Cloud Execution

View progress in “Test Result Module” of the distributor

ERP5 / Test Results / DREAM Simulation Run / Logged In as : dream

View **Nodes** **Graph** **Consistency** **History** **Metadata**

Title	DREAM Simulation Run	Project	<input type="text"/>
Reference	SIMPAT_20_orders_20_ants_Deterministic	All Tests	0
Test Report ID	928	Failures	0
Launch Date	2015/01/25 22:18	Errors	0
Completion Date	2015/01/25 22:20	Skips	0
		Test Result	PASS
		Test Status	Completed

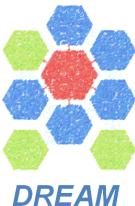
Comment

Total processing time: 138.312806
 Total execution time (including protocol): 312.004467
 Total execution time (only manpy): 9.2277841568

Test Results : 1 - 0 of 20 records

Test Case ▲▼	Start Date ▲▼	Duration	Manpy Execution Time	Status ▲▼
(u'QIM': 'EDD', u'QStart': 'RPC', u'QMASS': 'Priority', u'QMILL': 'SPT', u'QCAM': 'FIFO', u'QEDM': 'Priority', u'QTURN': 'SPT')	2015/01/25 22:19:42.885798 UTC	15.208942	0.489723920822	Completed
(u'QIM': 'EDD', u'QStart': 'SPT', u'QMASS': 'WINQ', u'QMILL': 'SPT', u'QCAM': 'EOD', u'QEDM': 'NumStages', u'QTURN': 'WINQ')	2015/01/25 22:19:45.376398 UTC	14.172767	0.328832149506	Completed
(u'QIM': 'EDD', u'QStart': 'WINQ', u'QMASS': 'LPT', u'QMILL': 'EOD', u'QCAM': 'RPC', u'QEDM': 'WINQ', u'QTURN': 'EOD')	2015/01/25 22:19:25.806784 UTC	15.131353	0.496667146683	Completed
(u'QIM': 'EDD', u'QStart': 'Priority', u'QMASS': 'EDD', u'QMILL': 'EDD', u'QCAM': 'EOD', u'QEDM': 'NumStages', u'QTURN': 'RPC')	2015/01/25 22:20:1.496204 UTC	14.408654	0.369523048401	Completed
(u'QIM': 'FIFO', u'QStart': 'MS', u'QMASS': 'NumStages', u'QMILL': 'Priority', u'QCAM': 'EOD', u'QEDM': 'EOD', u'QTURN': 'RPC')	2015/01/25 22:18:51.861788 UTC	15.180787	0.495496034622	Completed
(u'QIM': 'LPT', u'QStart': 'Priority', u'QMASS': 'FIFO', u'QMILL': 'EOD', u'QCAM': 'LPT', u'QEDM': 'LPT', u'QTURN': 'EDD')	2015/01/25 22:19:26.947794 UTC	16.429031	0.550364017487	Completed
(u'QIM': 'LPT', u'QStart': 'SPT', u'QMASS': 'RPC', u'QMILL': 'MS', u'QCAM': 'Priority', u'QEDM': 'FIFO', u'QTURN': 'Priority')	2015/01/25 22:18:50.794476 UTC	16.041735	0.555281162262	Completed
(u'QIM': 'MS', u'QStart': 'MS', u'QMASS': 'EOD', u'QMILL': 'RPC', u'QCAM': 'MS', u'QEDM': 'Priority', u'QTURN': 'WINQ')	2015/01/25 22:20:5.955112 UTC	16.498534	0.573115825653	Completed
(u'QIM': 'NumStages', u'QStart': 'NumStages', u'QMASS': 'SPT', u'QMILL': 'NumStages', u'QCAM': 'MS', u'QEDM': 'RPC', u'QTURN': 'RPC')	2015/01/25 22:20:24.754760 UTC	16.344587	0.552254915237	Completed
(u'QIM': 'Priority', u'QStart': 'MS', u'QMASS': 'LPT', u'QMILL': 'EDD', u'QCAM': 'WINQ', u'QEDM': 'SPT', u'QTURN': 'EDD')	2015/01/25 22:18:50.853945 UTC	15.010426	0.339298009872	Completed
(u'QIM': 'Priority', u'QStart': 'Priority', u'QMASS': 'RPC', u'QMILL': 'NumStages', u'QCAM': 'Priority', u'QEDM': 'Priority', u'QTURN': 'RPC')	2015/01/25 22:20:18.677495 UTC	14.220671	0.325212001801	Completed
(u'QIM': 'RPC', u'QStart': 'MS', u'QMASS': 'NumStages', u'QMILL': 'MS', u'QCAM': 'FIFO', u'QEDM': 'WINQ', u'QTURN': 'Priority')	2015/01/25 22:19:8.728046 UTC	16.402627	0.558176994324	Completed
(u'QIM': 'RPC', u'QStart': 'RPC', u'QMASS': 'WINQ', u'QMILL': 'RPC', u'QCAM': 'MS', u'QEDM': 'EDD', u'QTURN': 'FIFO')	2015/01/25 22:19:45.312931 UTC	18.448665	0.552452087402	Completed
(u'QIM': 'RPC', u'QStart': 'WINQ', u'QMASS': 'RPC', u'QMILL': 'MS', u'QCAM': 'SPT', u'QEDM': 'LPT', u'QTURN': 'EDD')	2015/01/25 22:19:8.876556 UTC	15.107798	0.485086917877	Completed
(u'QIM': 'SPT', u'QStart': 'LPT', u'QMASS': 'MS', u'QMILL': 'WINQ', u'QCAM': 'SPT', u'QEDM': 'RPC', u'QTURN': 'LPT')	2015/01/25 22:20:34.908766 UTC	14.25004	0.331388950348	Completed
(u'QIM': 'SPT', u'QStart': 'NumStages', u'QMASS': 'LPT', u'QMILL': 'RPC', u'QCAM': 'EDD', u'QEDM': 'MS', u'QTURN': 'EDD')	2015/01/25 22:20:1.21873 UTC	15.211300	0.469630946665	Completed

Cloud Execution



Cloud execution only has benefit for long running scenarios

