



# CaSchd.rb – QuickStart (20100124)

<http://wdwave.dnsalias.com/caschd>

To install the software you need the archive of the sources and a ruby interpret. The last version of the source code can be found at <http://wdwave.dnsalias.com/caschd> (see **Support** option). Ruby is included in all good operating systems. To verify, type 'ruby -v' in a shell prompt. If the command fails, you have to install ruby (see <http://www.ruby-lang.org/fr/download> ). Unzip the archive 'caschd-nnnnnnnn.zip' (nnnnnnnn is the release number, usually its date) to generate a 'caschd' folder containing the software. You can move the folder where you want. A good choice could be '/usr/bin' on Linux, 'Applications' or '/Users/Applications' on MacOS, 'C:\Program Files' on Windows. You can use a text editor to modify the configuration file 'caschd.conf'. The parameters are described as a JSON expression.

The parameters are organized around two concepts: tests to be operated ("**test**") and users to be informed ("**user**"): { "**test**": [ **tests group** ], "**user**": [ **users group** ] }. The tests group is made of test descriptions separated by a comma («,»). A test description take the form { "**name**": **name of the test**, "**doit**": **true or false**, "**time** ": [ **periods group** ], "**exec**": [ **actions group** ] }. The test name must be use only once. The parameter "**doit**" specifies if the test is active (true) or not (false). This option allows describing a test without using it. The periods group describes the days and associated time periods during which the (active) test must be executed. They are separated by a comma (',') and use the form: { "**days**": "**1234560**", "**from**": "**0830**", "**to**": "**1155**", "**wait**": **3** }. The parameter "**days**" specifies the days of the week from Sunday (0) to Saturday (6). The period is limited to the referred days (all week in the sample). When a day isn't concerned, it can be withdrawn from the list ("**123560**") or replaced by another character ("**123-560**"). The indications "**from**" and "**to**" specify the beginning and the ending of the time period. When the test must proceed continuously twenty-four hours a day, the period expression "**from**": '**0000**', "**to**": "**2400**" must be used. The frequency "**wait**" is in minutes. The expression { "**days**": "**1234560**", "**from**": "**0000**", "**to**": "**2400**", "**wait**": **3** } describes a test that must be operated in 24/7 every 3 minutes. The period between two tests (3 min) is respected even on the day change. If in days 3 (Wednesday), the last test is executed at 23h59, the next one will be done Thursday (4) at 00h02 (23h59+3 min.). The next samples illustrate some current cases.

## Sample 1

```
"time" : [
  { "days" : "123456-", "from" : "0900" , "to" : "1200" , "wait" : 10 },
  { "days" : "12345--", "from" : "1400" , "to" : "1730" , "wait" : 10 }
]
```

The description corresponds to the activation from 9h00 to 12h00 and from 14h to 17h30, Monday to Friday and from 9h to 12h Saturday every 10 minutes. The sample 1 can be formulated differently. The formulation above is less astute but more readable. It corresponds exactly to the literal statement.

```
"time" : [
  { "days" : "12345--", "from" : "0900" , "to" : "1200" , "wait" : 10 },
  { "days" : "12345--", "from" : "1400" , "to" : "1730" , "wait" : 10 },
  { "days" : "-----6-", "from" : "0900" , "to" : "1200" , "wait" : 10 }
]
```

### Sample 2

```
"time" : [
  { "days" : "123-567", "from" : "0000" , "to" : "2400" , "wait" : 10 },
  { "days" : "----4---", "from" : "0900" , "to" : "1230" , "wait" : 15 }
]
```

The sample 2 corresponds to the activation 24/7 every 10 minutes, except Thursday (4) from 9h00 to 12h30 every 15 minutes.

Once the time periods of the test are defined, the actions to be executed must be specified. The actions group ("**exec**") consists of the description of the actions separated by a comma (','). An action is characterized by its name ('name'), its parameters ("**args**") and consecutive fails number tolerated ("**redo**"). The parameters depend on the name of the action excepted the first which always corresponds to the maximum time assigned its execution (timeout). The examples hereafter specify the usage of TCP ping (TCP socket opening), HTTP access (reading the HMTL body of a Web page), sending SMTP mail, POP3 access, send/receive of SMTP/POP3 mails, 'heart beat' management, DNS control, send/receive of files (passive FTP) and checks the tests results of CaSchd secondary servers.

### Sample 1 - ping

```
{
  "name" : "ping",
  "args" : [ 1, "wdwave.dnsalias.com", 80 ],
  "redo" : 0
}
```

The action to be executed is a TCP ping on port 80. The maximum duration is **1** second. The host can be identified by its IP address or DNS name ("**wdwave.dnsalias.com**").

### Sample 2 - http

```
{
  "name" : "http",
  "args" : [ 5, "wdwave.dnsalias.com" ],
  "redo" : 2
}
```

The action resides in the reading of a Web page at **http://wdwave.dnsalias.com** with a timeout of **2s**. Two consecutive failures are tolerated. In the event of a breakdown (or overload) of the site, the alerts will be propagated at the third unfruitful attempt. When the action is associated with a time period with a frequency ("**wait**") of N minutes, the alert is diffused after a fail duration of 3 \* N minutes.

### Sample 3 - smtp

```
{
  "name" : "smtp",
  "args" : [ 10, "smtp.free.fr", 25, "sujet\tmessage",
            "from@free.fr", "to@free.fr " ],
  "redo" : 0
}
```

The action emits a mail with the subject and message indicated to the recipient ("**to@free.fr**"). A tabulation character ('\t') separates the subject and the message. Send is executed by a SMTP server ("**smtp.free.fr**" listening on port **25**). The sender ("**from@free.fr**") is also defined. A timeout of **10s** is used.

### Sample 4 – pop3

```
{
  "name" : "pop3",
  "args" : [ 30, "pop.laposte.net", "user", "password" ],
  "redo" : 0
}
```

The action consists in reaching a POP3 service ("**pop.laposte.net**") with the following login name and password : "**user**", "**password**".

### Sample 5 - sPPP

```
{
  "name" : "sPPP",
  "args" : [ 30, "smtp.free.fr", 25, "test caption",
            "from@free.fr", "to@laposte.net", "pop.laposte.net",
            "user", "password" ],
  "redo" : 1
}
```

The action consists in emitting a mail starting using the indicated server (see Sample 3 – smtp for the significance of the first 6 parameters) to destination address ("**to@laposte.net**") and controlling the reception of it. The POP3 server used ("**pop.laposte.net**"), the login name ("**user**") and the password ("**password**") associated with the account are also necessary. Each time the action is executed, the target account is consulted. The messages whose subject comprises the marker ("**test caption**") are removed. By default, the action is in anomaly. A message is then

transmitted to the target address (to allow the following test). When the mailbox is empty (first test), the action will be systematically in anomaly waiting for the reception of the first transmitted message. To avoid the detection of this false anomaly, it is appropriate to send a message with the marker before starting CaSchd or to set the parameter 'redo' to 1. The action will then be considered in anomaly only after two consecutive failures.

### Sample 6 - htbt

```
{
  "name" : "htbt",
  "args" : [ 1, "user", "password" ],
  "redo" : 0
}
```

The action fails if the URL ***http://<CaSchd.rb host IP address>:2000/?htbt=user&pswd=password*** is not requested (by an external application to check its state). The associated answer is **JHHMM :OK** or **JHHMM :KO** according to whether the login name and password are accepted or not. **JHHMM** corresponds to the designation of the day and time according to the format used by CaSchd.rb. It is appropriate to choose a period of control twice of for the frequency of URL calls.

### Sample 7 - rdns

```
{
  "name" : "rdns",
  "args" : [ 5, "212.27.40.240", "su3.mcafee.com",
            "192.187.128.17 208.69.152.139" ],
  "redo" : 0
}
```

The DNS action has two uses. The first allows checking the availability of a DNS server (answers or not). The second allows controlling the validity of the returned IP address for a designed host name. The parameters ("**args**") are: timeout (5s), IP address of the DNS server ("**212.27.40.240**"), host name (su3.mcafee.com), list of valid answers ("**192.187.128.17 208.69.152.139**"). In the sample, the host has two public addresses.

### Sample 8 - fpfp

```
{
  "name" : "fpfp",
  "args" : [ 30, "ftp.drivehq.com", "user", "password",
              "remote source file", "local file to compare",
              "ftp.drivehq.com", "user", "password",
              "remote target file", "local file to send" ],
  "redo" : 0
}
```

The action is divided in two parts. The first receives, from the first described FTP server, the "**remote source file**" and compares it to the "**local file to compare**". The second sends the "**local file to send**" to the second FTP server in the "**remote target**

**file".** When "" is used for the description of the **"local file to compare"** or the **"local file to send"**, the file **"caschd.flg"** is used.

### Sample 9 - cns1

```
{
  "name" : "cns1",
  "args" : [ 10, "192.168.0.2:2000", "Monde#Test#France", 2 ],
  "redo" : 0
}
```

Action **"cns1"** works like the remote Windows console (caschd\_cns1). It requests the CaSchd server (eg **192.168.0.2:2000**) and collects the state of tests called (**"Monde", "Test", "France"**). The value 2 specifies the minimum level to be attained to alert: **1** OK?, **2** KO!.

### **NEW:**

The argument "info" (in format HTML) allows specifying the label to be shown for the action linked in Web pages. It is available for all actions.

### Sample 9b - cns1

```
{
  "name" : "cns1",
  "args" : [ 10, "192.168.0.2:2000", "Monde#Test#France", 2 ],
  "redo" : 0,
  "info" : "<b>cns1 10</b>, <a href=\"\"http://192.168.0.2:2000\"\">CaSchd</a>"
}
```

The test hereafter ('TEST1') takes again examples 1 and 2 and locates them in the context previously described. The expression hereafter corresponds to the complete description. It underlines the aggregation of all the characteristics: name, active (or not), time periods, actions to be executed.

```
{ "name" : "TEST1",
  "doit" : true,
  "time" : [
    { "days" : "1234560", "from" : "0830" , "to" : "1155" , "wait" : 3 },
    { "days" : "1234560", "from" : "1210" , "to" : "1800" , "wait" : 3 }
  ],
  "exec" : [
    {
      "name" : "ping",
      "args" : [ 1, "wdwave.dnsalias.com", 80 ],
      "redo" : 0
    },
    {
      "name" : "http",
      "args" : [ 5, "wdwave.dnsalias.com" ],
      "redo" : 2
    }
  ]
}
```

## Sample 10 - icmp

```
{  
  "name" : "icmp",  
  "args" : [ 10, "192.168.0.2 " ],  
  "redo" : 0,  
  "info" : "<b>icmp 10, <font color=\"\"red\"\">ping 192.168.0.2</font></b>"  
}
```

The test tries to ping the address 192.168.0.2.

The second part of the configuration describes the users to whom information of states (defect or return to the normal) will be forwarded. The users group is made of a continuation of user descriptions separated by a comma (','). The user description use the form { **"name": user name, "doit": true or false, "tmhs": [ time periods group ], "tmok": [ time periods group ], "test": [ interested users group ], "exec": action }.** The user is characterized by a univocal name ("**name**"). As for the tests, "**doit**" allows to activate or inhibit a user (without having to remove its description from the configuration file. Time periods "**tmhs**" and "**tmok**" specify the time periods during which, the user agrees to receive the defect and return to normal messages. The period expression is identical to those of the tests with the exception of the frequency ("wait"), that isn't requested. The fact of having time periods for each types of message allows user to be informed 24/7 of any anomaly but limit the sending of the return to the normal message only in working hours. The messages are diffused in real times if they begin with '!' (a two minute delay applies if not). Those that are blocked for reasons of planning are not emitted. The user can adopt this strategy for a push mail device (e.g. Blackberry). It's possible to create a second user who accepts all the messages 24/7 to a classic mailbox. The software journalizes all the events: success or failures of the actions, statute changes of tests. The 100 last events are accessible on the main page. The list of the interesting tests for the user includes the names of each test separated by a comma (','). The last indication "**exec**" describes the action (only one) to be done for the diffusion of the state changes: it requests the name ("**name**") and the argument ("**args**") of the action.

The action hereafter sends a mail (SMTP) that indicates the change of status to **"to@free.fr"** from **"from@free.fr"** using the server **"smtp.free.fr"** on port 25 with a timeout of 10s. The fourth indication "**P!**" corresponds to the type of message to emit. It accepts the value "" or any pattern made of "**P**", "**L**", "**F**" in uppercase (anomaly) or lowercase (return to the normal) corresponding respectively to:

- the test name only ("")
- the web page associated to the test ("P", "p")
- the main page log ("L", "l")
- the content of a text file ("F", "f")

The parameters ("F", "f") allow sending the contents of a defined text file (generic or specific to each test state). For the test named "X", the files are called "\_X.OK" (return to normal), "\_X.KO" (error) and "\_X" (generic). The other characters than alphabetic or numeric are replaced by an underscore character ("\_"), ex: "test" requires "\_\_test.OK", "\_\_test.KO" and / or "\_\_test.

```
"exec" : {
  "name" : "smtp",
  "args" : [ 10, "smtp.free.fr", 25, "P!", "from@free.fr", "to@free.fr" ]
}
```

The actions 'ping' and 'http' can also be attached to the state change of a test. A 'ping' can switch on a « wake-one-Lan » equipment. A 'http' call can execute a program on a Web server. The action 'puts' does nothing (only writes an entry in the log). The example hereafter comprises the integral expression of a configuration file. The two parts **"test"** and **"user"** are easily identifiable. The number of tests and users are not limited. The consultation of the dashboard makes it possible to easily check the good description of the parameters related to the tests.

### caschd.conf

```
{ "conf" :
  {
    "port" : 2000,
    "#drby" : 2001
  },
  "test" : [
    { "name" : "France",
      "doit" : true,
      "time" : [
        { "days" : "1234560", "from" : "0000" , "to" : "2400" , "wait" : 15 }
      ],
      "exec" : [
        {
          "name" : "http",
          "args" : [ 5, "www.google.fr" ],
          "redo" : 0
        },
        {
          "name" : "ping",
          "args" : [ 10, "www.free.fr", 80 ],
          "redo" : 0
        },
        {
          "name" : "ping",
          "args" : [ 10, "www.wanadoo.fr", 80 ],
          "redo" : 0
        },
        {
          "name" : "http",
          "args" : [ 5, "fr.msn.com" ],
          "redo" : 0
        }
      ]
    },
    { "name" : "Monde",
      "doit" : true,
      "time" : [
        { "days" : "1234560", "from" : "0000" , "to" : "2400" , "wait" : 15 }
      ],
      "exec" : [
        {
          "name" : "ping",
          "args" : [ 10, "www.youtube.com", 80 ],
          "redo" : 0
        },
        {
          "name" : "ping",
          "args" : [ 10, "www.facebook.com", 80 ],
          "redo" : 0
        },
        {
          "name" : "http",
          "args" : [ 5, "www.bing.com" ],
          "redo" : 0
        },
        {
          "name" : "http",
          "args" : [ 5, "www.wikipedia.org" ],
          "redo" : 0
        }
      ]
    },
    { "name" : "Test",
      "doit" : true,
```



```

"time" : [
  { "days" : "1234560", "from" : "0830" , "to" : "1155" , "wait" : 3 },
  { "days" : "1234560", "from" : "1210" , "to" : "1800" , "wait" : 3 }
],
"exec" : [
  {
    "name" : "http",
    "args" : [ 5, "192.168.0.1" ],
    "redo" : 0
  }
]
},
"user" : [
  { "name" : "USER1",
    "doit" : false,
    "tmhs" : [
      { "days" : "1234560", "from" : "0000" , "to" : "2400" }
    ],
    "tmok" : [
      { "days" : "1234560", "from" : "0000" , "to" : "2400" }
    ],
    "test" : [ "France", "Monde" ],
    "exec" : {
      "name" : "smtp",
      "args" : [ 10, "smtp.free.fr", 25, "", "from@free.fr", "to@free.fr" ]
    }
  },
  { "name" : "USER2",
    "doit" : false,
    "test" : [ "Test" ],
    "tmhs" : [
      { "days" : "-----60", "from" : "0000" , "to" : "2400" }
    ],
    "tmok" : [
      { "days" : "-----60", "from" : "0000" , "to" : "2400" }
    ],
    "exec" : {
      "name" : "smtp",
      "args" : [ 10, "smtp.free.fr", 25, "", "from@free.fr", "to@free.fr" ]
    }
  }
]
}
}

```

Run **ruby CaSchd.rb** from the `caschd` directory to activate `CaSchd.rb`. The call of **http://localhost:2000** gives access to the main page. The WEBrick server is included with the software. In the sample above, the users are disabled. To test, the SMTP indications must be corrected and the parameters **"doit"** must be set to **"true"**. The figures hereafter results from the consultation of the main page with the described parameters.

**http://localhost:2000** or <http://localhost:2000/?page>

<b>CaSchd.rb</b> [Green T.] 20090919	10:14 123456[0] - * -----
---	------------------------------

[France](#)      2009/10/04 10:14:17 : KO : http 5, 192.168.0.1  
 [Monde](#)  
 [Test](#)  
 \*  
 \_

The first part '10:14 123456[0]' indicates the days of the week (0=Sunday), and the hour (10:14) of the generated page. It is followed by the indication of the agent (40) states which execute the actions: '\_' not active, '\*' active. The left part indicates that

the tests 'France' and 'Monde' are correct (green) and that the test 'Test' is in anomaly (red). The right part lists the last state changes (this Sunday October 8th). A clic on the label of each test gives access its dashboard. The page is updated with the indication of the test label, time periods and status. The right part describes the test. It comprises the list of the time periods (those active are indicated in yellow) then the whole of the actions (with their respective states). The duration of the last test and the maximum duration are indicated after each action in the form **<action> (lasted < lasted max.)**. The displayed time period 10:29 (10:29) 1234560 (00:00-24:00) indicates the hour of the next partial control **10:29** (no atomic and in anomaly actions) and the hour of the next control for all actions in parenthesis (**10:29**). Without anomaly, these two indications are identical.

### http://localhost:2000/?page=France

CaSchd.rb [Green T.] 20090919	10:21 123456[0] - France
----------------------------------	--------------------------









---

 <a href="#">France</a>	 10:29 (10:29) 1234560 (00:00-24:00)
 <a href="#">Monde</a>	
 <a href="#">Test</a>	 sample : <a href="http://www.google.fr">www.google.fr</a> (0.6<0.6)
*	 ping 10, <a href="http://www.free.fr">www.free.fr</a> , 80 (0.2<0.2)
-	 ping 10, <a href="http://www.wanadoo.fr">www.wanadoo.fr</a> , 80 (0.3<0.3)
	 http 5, <a href="http://fr.msn.com">fr.msn.com</a> (1.0<1.0)

### http://localhost:2000/?page=Monde

CaSchd.rb [Green T.] 20090919	10:25 123456[0] - Monde
----------------------------------	-------------------------

---







 <a href="#">France</a>	 10:29 (10:29) 1234560 (00:00-24:00)
 <a href="#">Monde</a>	
 <a href="#">Test</a>	 ping 10, <a href="http://www.youtube.com">www.youtube.com</a> , 80 (0.3<0.3)
*	 ping 10, <a href="http://www.facebook.com">www.facebook.com</a> , 80 (0.3<0.3)
-	 http 5, <a href="http://www.bing.com">www.bing.com</a> (0.5<0.5)
	 http 5, <a href="http://www.wikipedia.org">www.wikipedia.org</a> (0.5<0.5)

The test 'Test' comprises two time periods. While consulting, it's the first that was active (in yellow). If the defect, announced in red, is maintained in this state, it will pass purple at 11h56 (message "OK?") and again in red at 12h10 ("KO!").

### http://localhost:2000/?page=Test

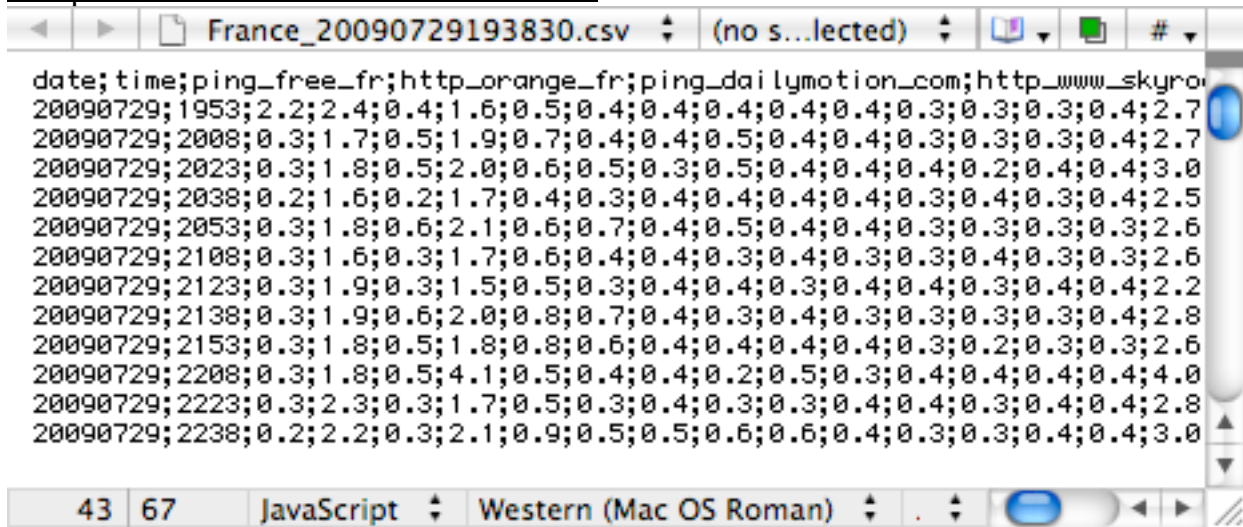
CaSchd.rb [Green T.] 20090919	10:27 123456[0] - Test
----------------------------------	------------------------

---

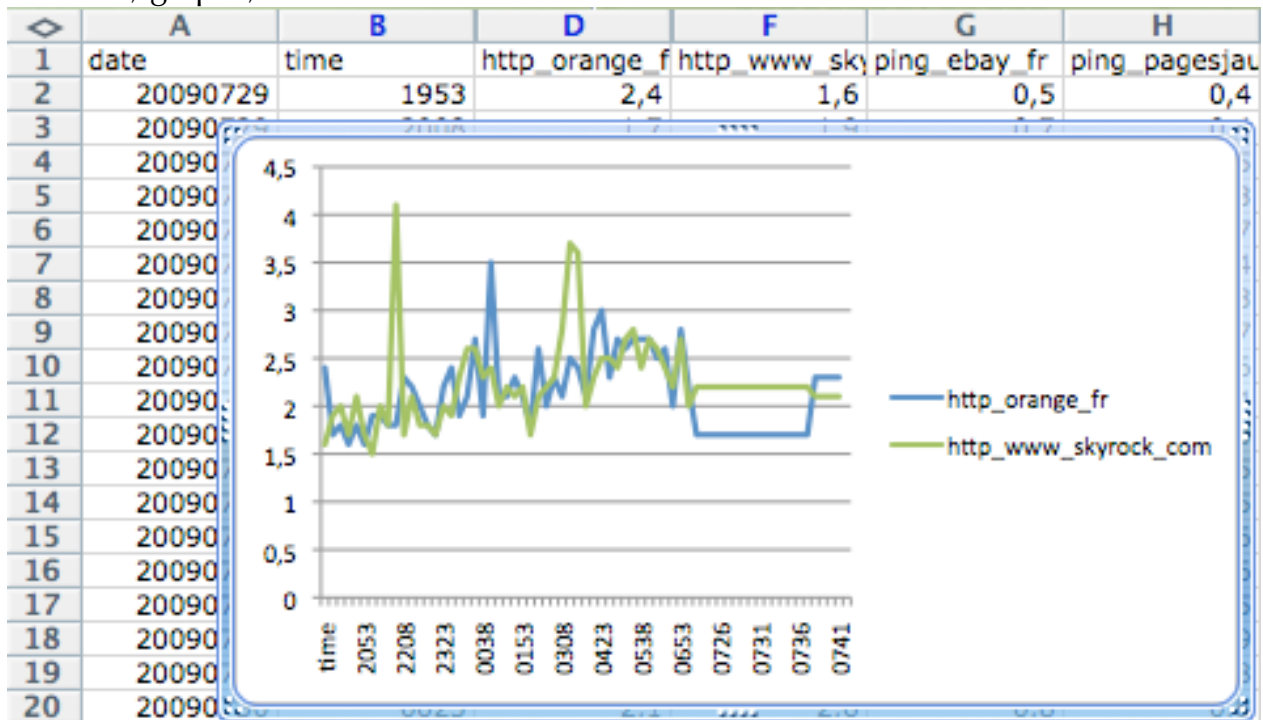
 <a href="#">France</a>	 10:27 (10:29) 1234560 (08:30-11:55)
 <a href="#">Monde</a>	 12:10 (12:10) 1234560 (12:10-18:00)
 <a href="#">Test</a>	
*	 http 5, 192.168.0.1 (0.0<0.1)
-	

CaSchd.rb consigns the response times of each action in a CSV file specific to each test. The file is named according to the test name, the date and the hour of the application starting.

Sample – France\_20090729193830.csv



The file can be imported in a spreadsheet (e.g. OpenOffice Calc or Excel) to carry out statistics, graphs, etc.



The field separator is the semicolon. The point is the decimal separator (to be replaced by a comma according to linguistic preferences).

[\*] The context of execution of the application of supervision conditions its performances and its straightness. The use of an operating system powerful, reliable and stable is highly recommended. Linux and Mac OS (native platform of the solution) are highly recommended.

## WARNINGS:

20090828 release introduces the red threads model (using distributed object system). You can remove '#' before the drby key in caschd.conf to use the red threads model. Red threads mode increase cpu load (up to 41 simultaneous process). It's recommended using only green threads mode on slow or not dedicated computer for better performance.

Some additional tools (Windows console, Windows heartbeat) are available to display the status of selected tests in the tray-bar or to generate heartbeat. See their respective documentation and configuration file before usage.

The sources of the CaSchd.rb project are under LPG v3 license. Documentations are under Creative Commons Paternity-Not of Commercial-Not Use of Modification 2.0 France license.

<http://wdwave.dnsalias.com>

Jean-Marc "*jihem*" QUERE

45935