

Logtrend's Visu installation's guide



LogTrend

Logtrend's Visu installation's guide

by LogTrend

Copyright © 2001 by LogTrend <http://www.logtrend.org>

This software and all affiliated files are Copyright (C) 2001 by Atrid Systèmes under the terms of the GNU General Public License. A copy of this license entitled "GNU General Public License" is included with the software. The original text can be found on <http://www.gnu.org/copyleft/gpl.html>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections , with no Front-Cover Texts , and with no Back-Cover Texts. A copy of the license entitled "GNU Free Documentation License" can be found with this software. The original text can be found on <http://www.gnu.org/copyleft/fdl.html>

Revision History

Revision 1.0 September, 2001

First Docbook version



Table of Contents

1. Introduction	4
2. Web-Visu	5
2.1. Working principles	5
2.2. Packages installation	5
2.3. Apache configuration	5
2.3.1. Apache	6
2.3.2. ApacheModule	7
2.3.3. Restarting Apache	7
2.4. Running the web-Visu	7
3. Report Visu	9
3.1. Configuration	9
3.2. In-line command	9
3.3. In the web-Visu	9
3.4. Generating the report	9
4. Visu configuration	10
4.1. File <code>network.xml</code>	10
4.1.1. Headers	10
4.1.2. <i>Entity's</i>	11
4.1.3. <i>Network</i>	11
4.1.4. <i>System</i>	11
4.1.5. <i>User</i>	13
4.2. File <code>database.xml</code>	14
4.3. Visu classes	14
4.3.1. Zones description	15
4.3.1.1. Label	16
4.3.1.2. TextArea	16
4.3.1.3. LineGraph and BarGraph	16
4.3.1.4. AverageGraph	18
4.3.1.5. Dial	19
4.3.1.6. Led	19
4.3.1.7. Alarms	20
4.3.1.8. Discs	20
4.3.1.9. Link	21
4.3.2. Expressions	21
4.3.3. A full example	22
4.3.4. Placement management	22
4.4. Formats	23
4.4.1. Duration	23
4.4.2. Abolute date	24
4.4.3. Interval	24
4.4.4. Color	24



Chapter 1. Introduction

The goal of this LogTrend module is to permit to visualize via le the web or via a report the data collected by the agents.

LogTrend agents send data and alarms to the StorageServer regularly. This server adds these informations into a database (mostly PostgreSQL). These Visu modules will get data from the database and, in accordance with configuration files and the logged user, will construct:

- for the web-Visu module: HTML pages and PNG images
- for the report--Visu module: SGML file and PS images

representing informations.

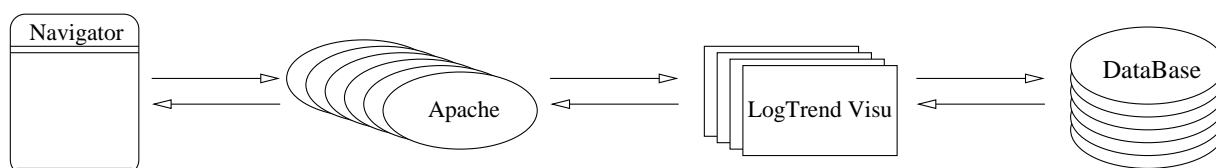


Chapter 2. Web-Visu

The web-Visu LogTrend module permit to visualize via le the web the data collected by the agents.

2.1. Working principles

The principle of the LogTrend web-Visu module is to be detached for the apache processes.



Apache is waiting for connection (as usual). If a connection is for the web-Visu (see chapter 'Apache configuration' for how to recognize such request), an authentication is needed. After authentication, apache contact the Visu server which send to it index pages ; apache re-send these pages to the navigator.

On these pages, some links are available to watch systems informations. The user can access to page containing informations. In this case, the Visu server (contacted by apache) contact the database in the aim to get data. These data arrive in a raw format. The Visu server treats them to make HTML pages which could include PNG images.

Apache, the Visu server and the database can be run on different computers.

2.2. Packages installation

You need Perl (version ≥ 5.00503) to run LogTrend. Perl is generally installed by default on all popular GNU/Linux and Unix distributions. But you can found more recent Perl release on <http://www.perl.org/>

LogTrend is released in CPAN perl packages. To install a LogTrend package, run the script :

```
$ tar xvzf LogTrend-packagename-version.tar.gz
$ cd LogTrend-packagename-version
$ perl Makefile.PL
    This command gives you the list of missing dependencies.

$ make
$ su
password :
# make install
```

2.3. Apache configuration



This is the instruction for configuration of the system on which Apache will run.

On this computer, you must have an apache, the apache mod_perl module and the LogTrend module called *Logtrend-Visu-Apache* installed.

2.3.1. Apache

An Apache specific configuration is need in aim to make Visu work. You must load perl_module and add a *Location* directive which will call an LogTrend Visu Apache module. For a standard configuration, you can simply add the line

```
Include /usr/share/doc/LogTrend/Visu-Apache/access.conf
```

in `/etc/apache/access.conf`. Otherwise, you must add the following lines to apache's configuration files.

In `httpd.conf` :

You must load the mod_perl module :

```
LoadModule perl_module path/mod_perl.so
```

where *path* is substituted by the full path to the `mod_perl.so` file ; for example `/usr/lib/apache/1.3`

In `access.conf` :

The *Location* directive start like this:

```
<Location /LogTrendVisu>
```

and end like that:

```
</Location>
```

In this directive, the next line are needed:

Call the LogTrend Visu Apache module:

```
SetHandler perl-script
PerlHandler LogTrend::Visu::ApacheModule
```

This instruction make that Apache calls the LogTrend Visu module when one good URL (starting by *LogTrend-Visu*) is requested.

The authentification mechanism need this lines:

```
AuthUserFile file
AuthName "LogTrendVisu"
AuthType Basic
<limit GET POST>
    require valid-user
</limit>
```



where *file* have to be the Apache password file (see *Visu configuration*), typically */etc/LogTrend/Visu/htpasswd*

You could add a directive for name resolution (to get the canonical name of the computer connection instead of its IP adress):

```
HostNameLookups on
```

In substance, you should add to your *access.conf* the following in order to make apache works with LogTrend Visu:

```
LoadModule perl_module /usr/lib/apache/1.3/mod_perl.so
<Location /LogTrendVisu>
    SetHandler perl-script
    PerlHandler LogTrend::Visu::ApacheModule
    #HostNameLookups on
    AuthUserFile /etc/LogTrend/Visu/htpasswd
    AuthName "LogTrendVisu"
    AuthType Basic
    <limit GET POST>
        require valid-user
    </limit>
</Location>
```

2.3.2. ApacheModule

If you install Visu and Apache on separated system, you have to edit the *LogTrend/Visu/ApacheModule.pm* in your Perl architecture in lines 111 and 113 (begining of the *handler* method). The variable *visuHost* contains the name of the Visu-system (by default: the same system that where Apache is running) and the variable *visuPort* the port of the Visu process (by default: 10101).

2.3.3. Restarting Apache

After this configuration, you need to restart Apache. Do it by your usual way: */etc/.../init.d/apache restart* or something else.

2.4. Running the web-Visu

See in the *Visu configuration* chapter who to configure the visu.

The Visu configuration is done in the */etc/LogTrend/Visu* directory. In this directory, the file *htpasswd* contains the logins and the passwords of the users allowd to connect to the web-Visu. This file is generated by a software usually provided in Apache packages: *htpasswd* (see *man 1 htpasswd*).

To run the Visu, you have the choise between to methods:



- run it in direct command line: just run *LogTrendVisu*
- instal an *init.d* file, and start it (or put the good links into the way to be run a system start).



Chapter 3. Report Visu

The report-Visu LogTrend module permit to visualize via in a paper-report the data collected by the agents. This module generate a tarball file (tar'ed and gzip'ed) containing the report in SGML format and postscript files for the figures.

3.1. Configuration

The file `/etc/LogTrend/Visu/report.conf` must contain:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ReportConf>
  <Dsssl Path="/usr/share/LogTrend/Visu/dsssl"/>
</ReportConf>
```

where `/usr/share/LogTrend/Visu/dsssl` is an example of directory name containing the dsssl files for Report generation.

In visu-class file, the tag *Class* accepts a *Report* attribute which indicates the file to use for report generation.

3.2. In-line command

You can get you report by an in-line command `LogTrendReport`. This program accepts these arguments:

- `--user` or `-u` (mandatory): the logtrend user to read the configuration (see *Visu configuration* chapter),
- `--output` or `-o` (optional): the output filename (default: on the standard output).

For exemple: `LogTrendReport -u logtrend -o report.tar.gz`

3.3. In the web-Visu

In the left frame of the web-Visu, a link named *Report Generation* gives access to a page containing a link for report generation. Save the file on you disc.

3.4. Generating the report

Whatever is the method you used to generate the report, you have a tarball file. Uncompress it with `tar` (and perhaps with `gunzip`). Go in the *report* directory and type `make`.

If all is correct, it will generate a `report.ps` file. You can visualize and print it.



Chapter 4. Visu configuration

The Visu configuration is done in the `/etc/LogTrend/Visu` directory. In this directory, a directory `users` contains all the information about the configuration of the Visu for each user. Each user has his own directory; the directory and the user have the same name. For example, user *peter* has his Visu configuration in the directory `/etc/LogTrend/Visu/users/peter`

In the following, will be described the files contained in these users directories:

4.1. File `network.xml`

When an user is logging, the first file to be read by the Visu server is the file named `network.xml` in his directory. This file contains information about the network(s) and the system(s) whom informations the user want visualize.

The DTD could be found in documentation directory (file `network.dtd`).

Its format is the following:

4.1.1. Headers

The file start by an XML declaration:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

The XML root tag:

```
<SystemsDeclaration>
```

Then, the name and the reference of the Visu configuration and a list of contacts will follow with firstname, surname and email (optional) (these informations will appear in the index page):

```
<Name>System and network supervision</Name>
<Reference>ref AR062001</Reference>
<Contact Firstname="Paul" Surname="Mart" Email="Paul.Mart@MaSociete.com" />
<Contact Firstname="Katy" Surname="Waldone" Email="Katy.Waldone@MaSociete.com" />
```

An XML tag for general colors:

```
<Colors BackGround="#FFFFFF" CellGround="#F0E8F0"
Text="#000000" Links="#084E86" VisitedLinks="#9999FF" />
```

used for HTML and PNG generation in web-Visu (in color format, see *Formats* section).

Then will be given the name of the file containing the parameters for data base connection:

```
<DataBaseFile>database.xml</DataBaseFile>
```

The format of this file will be described in the next section.



4.1.2. *Entity's*

Then some *Entity's* will be described. An *Entity* has a unique number and correspond to an Agent (ie a source and an agent number).

```
<Entity Number="1">
  <Source>16</Source>
  <Agent>2</Agent>
  <Name>mars.widget.org</Name>
  <IP_Address>192.168.10.12</IP_Address>
  <Owner>Widget Corp</Owner>
  <Situation>2nd floor</Situation>
  <Description>Linux server</Description>
</Entity>
```

Respectively:

- the number of the source on which the agent is running,
- the number of the agent on this source,
- the name of the controlled system (unused for the moment),
- the name of the owner (unused for the moment),
- the situation of the system (unused for the moment),
- a description of the system (unused for the moment).

4.1.3. *Network*

Then the network structure has to be described in the file. For that, one or several *Network* tags follow in the file. A *Network* can contain *Networks* and/or *Systems*.

A *Network* has some personal informations: its name, its IP address, its net-mask and a description.

```
<Network>
  <Name>Widget Corp</Name>
  <IP_Address>192.168.0.0</IP_Address>
  <Netmask>255.255.0.0</Netmask>
  <Description>Internal network</Description>
  ... some Network and System ...
</Network>
```

4.1.4. *System*

A *System* tag doesn't correspond with a system in common sense: it is not a computer representation. But it could. It is the application of an visu-class (see *visu-class* section) on one or several *Entity's*.



A *System* is name up of a name, a visu-class and one or several canvas. A canvas has an own-number (used in the visu-class) and an *Entity* number; the visu-class use the canvas number to identify the agent, this is a way to put agents as parameters of the visu-class.

In most case, a *System* looks like this:

```
<System>
  <Name>Zeus</Name>
  <Description>web server</Description>
  <Class Web="Linux_server"/>
  <Canvas Number="1" Entity_Number="8"/>
</System>
```

We apply the visu-class *Linux_server* to the entity number 8 (previously defined as the agent supervising the web server called *Zeus*). The visu-class know what to do with this entity. The attribute *Web* of tag *Class* is for the web-visu, an attribute *Report* could be set for report-visu.

Some more complicated cases could appear:

```
<System>
  <Name>Comparison</Name>
  <Description>mars/junon</Description>
  <Class Web="2Linux"/>
  <Canvas Number="1" Entity_Number="4"/>
  <Canvas Number="2" Entity_Number="12"/>
</System>
```

In this example, the visu-class *2Linux* needs 2 agents to make a Comparison, so it needs 2 canvas. In this case, we say that it must take the entity number 4 as canvas 1 and the entity number 12 as canvas 2. Entities are just (ordered) parameters for the visu-class.

This is a full example of *Network* and *Systems* tree:

```
<Network>
  <Name>Widget Corp</Name>
  <IP_Address>192.168.0.0</IP_Address>
  <Netmask>255.255.0.0</Netmask>
  <Description>Internal network</Description>
  <Network>
    <Name>Paris</Name>
    <IP_Address>192.168.10.0</IP_Address>
    <Netmask>255.255.255.0</Netmask>
    <Description>Head office</Description>
    <System>
      <Name>zeus</Name>
      <Description>Central server</Description>
      <Class Web="Linux_Server"/>
      <Canvas Number="1" Entity_Number="1"/>
    </System>
  </Network>
</System>
```



```
<Name>aphrodite</Name>
<Description>Mail relay</Description>
<Class Web="Linux_Server"/>
<Canvas Number="1" Entity_Number="2"/>
</System>
</Network>
<Network>
  <Name>Bordeaux</Name>
  <IP_Address>192.168.30.0</IP_Address>
  <Netmask>255.255.255.0</Netmask>
  <Description>Hip!</Description>
  <System>
    <Name>mars</Name>
    <Description>Central server</Description>
    <Class Web="Linux_Server">
    <Canvas Number="1" Entity_Number="3"/>
  </System>
</Network>
<System>
  <Name>Zeus/Héphaïstos</Name>
  <Class Web="2Linux"/>
  <Description>Comparison zeus/mars</Description>
  <Canvas Number="1" Entity_Number="1"/>
  <Canvas Number="2" Entity_Number="3"/>
</System>
</Network>
<Network>
  <Name>FooBar client</Name>
  <IP_Address>178.34.10.0</IP_Address>
  <Netmask>255.255.255.0</Netmask>
  <Description>Client number 351T</Description>
  <System>
    <Name>Web server</Name>
    <Description>www.foobar.com</Description>
    <Class Web="Linux_Server"/>
    <Canvas Number="1" Entity_Number="4"/>
  </System>
</Network>
```

4.1.5. User

A *User* tag corresponds with a pre-existing user configuration. This allows some users to supervise several companies with the same login. This tag is allowed only for user who's name starts by admin; for other users, a nice error page is prepared.

This tag has one attribute : *Name*

```
<User Name="client1"/>
```



```
<User Name="client2"/>
```

where client1 and client2 are existing user for LogTrend-WebVisu.

In the case of the admin user has no *Network* nor *System*, tags *Entity* and *DataBaseFile* are not need.

4.2. File database.xml

The file named in *DataBaseFile* tag in file *network.xml* (typically *database.xml*) contains informations about the data base and the way to connect it.

This file contains the name of the data base, the name of the system on which the data base run, the port of the data base, the user and password to connect.

For example, the file could look like this:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<Infos>
  <DataBase>logtrend</DataBase>
  <Host>db.widget.com</Host>
  <Port>5432</Port>
  <User>logtrend</User>
  <Password>password</Password>
</Infos>
```

The DTD could be found in documentation directory (file *database.dtd*).

4.3. Visu classes

In the *network.xml* file, some visu-classes are called to be applied to entity/ies (agent). Such a file contains the way to display data of the agent(s): texts, graphes, etc.

A visu-class could contains several screens, corresponding to the HTML pages or to the Report pages. At least one screen is needed (otherwise, why having such a visu-class?).

A screen is made of zones. A zone is a way to display one or several given data. There is several types of zones (see *Zones description* section). They contain information about the way to place them one in relation to the others in the screen (for the moment, a screen in a Report could only contains one zone).

This is an example of a visu-class file:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<Class>
  <Name>Linux_Server</Name>
  <Screen Id="1" ColSize="50">
    <Name>Main screen</Name>
    <Description>Main infos for Linux server</Description>
```



```
<Zone>
  ... a zone ...
</Zone>
... other zones ...
</Screen>
... other screens ...
</Class>
```

The visu-class has a name (*Name* tag / not used for the moment). Each screen has:

- an unique identifier (*Id* attribute) used for making links to this screen,
- a size for columns in HTML tables (*ColSize* attribute) (see *Placement management* section),
- a name (*Name* tag),
- a description (*Description* tag).

A screen contains zones (*Zone* tags).

The DTD could be found in documentation directory (file `visu-class.dtd`).

4.3.1. Zones description

A zone is a rectangular area in the screen displaying informations given by the entity (see *Entity's* section). There is several type of zones but all have some common characteristics.

this is an example of a zone:

```
<Zone>
  <Title>title</Title>
  <Position x="4" y="6" width="1" height="2"/>
  <Type>
    ... the type and informations which depend of the type ...
  </Type>
  <Data>
    ... some Expression tags ...
  </Data>
</Zone>
```

A zone has:

- a title (*Title* tag),
- some position and size informations (*Position* tag) ; *x* and *y* attributes are about the position in the page, and *width* and *height* attributes are about the size of the zone (see *Placement management* section),
- the *Type* tag will contain a tag indicating the type of the zone and informations specific to the type of the zone (see below),



- the *Type* tag will contain some *Expression* tags which indicate the exact data to display (see *Expressions* section).

4.3.1.1. Label

Usage: web-Visu only

A *Label* is the more simple zone: it just print its title. It contains no more information.

Its *Type* tag could be like this:

```
<Type>
  <Label></Label>
</Type>
```

Its *Data* tag will be empty.

4.3.1.2. TextArea

Usage: web-Visu only

A *TextArea* is a zone displaying in a text way the last values of the given expression. The number of last values to display is to be declare in an optional attribute named *Lines*.

If you consider your value as a percentage, you can add the optional attribute *PercentRound* which multiplies the value by 100 and round it.

Its *Type* tag could be like this:

```
<Type>
  <TextArea Lines="2" PercentRound="0.1" />
</Type>
```

In this case the last two values of each expression will be displayed, multiplied by 100 and rounded at the first decimal.

4.3.1.3. LineGraph and BarGraph

Usage: web-Visu and report-Visu

A *(Line/Bar)Graph* is a graphical representation of numeric data: the time on x axis (horizontal), the values on y axis (vertical). A *LineGraph* is a graph joining value points by lines; a *BarGraph* is a graph showing values by a rectangular colored area from the bottom of the graph.

The *(Line/Bar)Graph* tag can have attributes:

- *PixelHeight* (optional): indicate the height of the generated image in pixel (see *Placement management* section),
- *LabelsPerLine* (optional): indicate the number of labels to be displayed by line.



- *Kilo* (optional): could be 1000 or 1024 (default). If 1024 is set, a kilo is 1024, a mega 1024x1024 etc and letters are K, M, G and T. If 1000 is set, a kilo is 1000, a mega 1000x1000 etc and letters are k, m, g and t.

The *(Line/Bar)Graph* tag can contain tags:

- *X* (mandatory): informations about the time. For a graph, we need to know the interval of time of the data to display. This tag must have *Start* and *Stop* attributes or *Interval* attribute or *Duration* attribute.
 - *Start*: the earlier boundary of the interval of time (in duration or absolute format, see *Formats* section),
 - *Stop*: the latest boundary of the interval of time (in duration or absolute format) (*Start* and *Stop* must be together in duration or absolute format in the same time),
 - *Interval*: the time-interval (in interval format),
 - *Duration*: the moment before now to display information (in duration format) *Duration*="x" is equivalent to *Start*="x" *Stop*="0"

It can have also these attributes:

- *Graduation* (optional): the interval of time between two graduations (in duration format).
- *TooBigGap* (optional, default=1h): the interval of time after which two values are too far to be linked by line or bar.
- *Y* (optional): informations about the values. It can have attributes:
 - *LowerBoundary* (optional): the minimum value to display,
 - *UpperBoundary* (optional): the maximum value to display,
 - *Graduation* (optional): the interval of value between two graduations.
- *Colors* (optional): indicate the colors for the image (in color format, see *Formats* section):
 - *PictureBG* (optional): the image background color,
 - *GraphBG* (optional): the graph background color,
 - *Grid* (optional): the color for the grid,
 - *Text* (optional): the color for the textes.

The *BarGraph* tag can contain additional tags:

- *Thresholds* (optional): indicate the thresholds for changing the color when reaching given values; attributes are:
 - *Warning* (optional): the threshold for warning color,
 - *Critical* (optional): the threshold for critical color.
- *Colors3* (optional): indicate the colors for thresholds. Attributes are (in color format, see *Formats* section):



- *Warning* (optional): the color for warning color,
- *Critical* (optional): the color for critical color.

To change the color of the line/bars, see the *Expression* section.

A wholly configured BarGraph could look like this:

```
<Type>
  <BarGraph PixelHeight="180">
    <X Graduation="1h" Start="30h" Stop="24h"/>
    <Y Graduation="20000000" LowerBoundary="0" UpperBoundary="150000000"/>
    <Colors PictureBG="#AAAAFF" GraphBG="#FFFFFF" Grid="#0000FF" Text="#AAFFAA"/>
    <Thresholds Warning="120000000" Critical="125000000"/>
    <Colors3 Warning="#FFFF00" Critical="#FF0000"/>
  </BarGraph>
</Type>
```

4.3.1.4. AverageGraph

Usage: web-Visu and report-Visu

The AverageGraph is a zone based on LineGraph. The X axis represent one day from midnight to 23:59:59. The curves represent:

- Max values on the defined time interval.
- Min values on the defined time interval.
- The average on the defined time interval.

The AverageGraph tag can have attribute *PixelHeight* (see *LineGraph* section).

The AverageGraph tag may contains tags:

- *Time*: the time interval to analyse. Attributes are:
 - *Start/Stop* or *Interval* or *Duration*: (same attributes than X tag of LineGraph)
 - *Period*: The average period. The default value is 10 minutes
 - *IncludeWeekEnd*: Indicate if the week end must be included in average or not.
- *Y* and *Colors*: see *LineGraph* for more details.

A wholly configured AverageGraph could look like this:

```
<Type>
  <AverageGraph PixelHeight="180">
    <Time Duration="last_month" Period="10m" IncludeWeekEnd="0" />
  </AverageGraph>
</Type>
```



```
<Colors PictureBG="#AAAAFF" GraphBG="#FFFFFF" Grid="#0000FF" Text="#AAFFAA"/>
</AverageGraph>
</Type>
```

4.3.1.5. Dial

Usage: web-Visu only

This zone generate a picture for one expression and for a period of time. It shows the minimum value, the maximum value and the last value (called the current value).

The *Dial* tag can have one attribute:

- *PixelHeight* (optional): indicate the height of the generated image in pixel (see *Placement management* section),

The *Dial* tag can contain tags:

- *X* (mandatory): informations about the time. See *LineGraph and BarGraph* section; attributes *Interval* is not used, and *Start/Stop* can only have duration format.
- *Y* (optional): informations about the values. See *LineGraph and BarGraph* section.
- *Colors* (optional): indicate the colors for the image. See *LineGraph and BarGraph* section.
- *ArrowsColors* (optional): indicate the colors for the arrows. Allowed attributes are (in color format, see *Formats* section):
 - *Min* (optional): the minimum arrow,
 - *Current* (optional): the current arrow (last value),
 - *Max* (optional): the maximum arrow.

A wholly configured Dial could look like this:

```
<Type>
  <Dial PixelHeight="180">
    <X Graduation="1h" Start="30h" Stop="24h"/>
    <Y Graduation="20000000" LowerBoundary="0" UpperBoundary="150000000"/>
    <Colors PictureBG="#AAAAFF" GraphBG="#FFFFFF" Grid="#0000FF" Text="#AAFFAA"/>
    <Formats Min="#00FF00" Current="#000000" Max="#FF0000"/>
  </Dial>
</Type>
```

4.3.1.6. Led

Usage: web-Visu only

This zone shows an label whom color changes according to the value of an expression and thresholds.



The *Led* tag can contain tags:

- *Thresholds* (mandatory): informations about the thresholds. See *LineGraph* and *BarGraph* section; attributes are mandatory.
- *Colors3* (optional): indicate the colors for thresholds. See *LineGraph* and *BarGraph* section.

This is an example:

```
<Type>
  <Led>
    <Thresholds Warning="0.1" Critical="0.5"/>
    <Colors3 Warning="#FFFF00" Critical="#FF0000"/>
  </Led>
</Type>
```

4.3.1.7. Alarms

Usage: web-Visu and report-Visu

This zone prints in a text format the alarms trowed by the entity (agent). Such a zone can have a *Colors3* tag (see *LineGraph* and *BarGraph* section).

An additional tag named *Time* is required for report-Visu. This tag contains Start/Stop or Duration attributes (same attributes than *X* tag of *LineGraph*)

This is an example:

```
<Type>
  <Alarms>
    <Colors3 Warning="#FFFF00" Critical="#FF0000"/>
  </Alarms>
</Type>
```

4.3.1.8. Discs

Usage: web-Visu and report-Visu

This zone prints in a text format information about discs of one entity (agent).

The *Discs* tag can contain tags:

- *Thresholds* (mandatory): informations about the thresholds for disc space usage (in percentage of occupation). See *LineGraph* and *BarGraph* section; attributes are mandatory.
- *Colors3* (optional): indicate the colors for thresholds. See *LineGraph* and *BarGraph* section. One attribute is added for *Discs*: *Ok* for the color of disc under limit.



An additional tag named *Time* is required for report-Visu. This tag contains Start/Stop or Duration attributes (same attributes than *X* tag of *LineGraph*)

This is an example:

```
<Type>
  <Discs>
    <Thresholds Warning="80" Critical="95"/>
    <Colors3 Ok="#00FF00" Warning="#FFFF00" Critical="#FF0000"/>
  </Discs>
</Type>
```

4.3.1.9. Link

Usage: web-Visu only

This zone permit to make an HTML link to another screen. The *Link* tag have an mandatory attribute *ScreenId* which indicate the identificator of the screen. In the HTML page, the title will be the text to click on.

This is an example:

```
<Type>
  <Link ScreenId="3">
  </Link>
</Type>
```

4.3.2. Expressions

The way to name data in the data base in the visu is to use Expressions. An Expression is a combinaison of data in an arithmetic expression.

A data is identified by a string containing:

- the number of the canvas (ie the entity is the agent)
- a dot (.)
- the name of the data for this agent, in lower case, with spaces changed into underscores (_) and without the optional text in parentheses

For exemple *1.cpu_user* represents the data name "Cpu User" for the canvas number 1.

For zones Alarms and Discs, only the the number of the canvas is needed.

Arithmetic operations could be applied to this data: the 4 classical operations, parentheses and numerical constants are allowed.

For exemple *1.cpu_user+(1.cpu_nice-1.cpu_idle)/10* is a correct expression.



Variables containing special characters (such as / or * etc) have to be put into singles-quotes (') in order to that the global expression will not ambiguous. For example: ('1.used_space_on_/'+'1.used_space_on/tmp')/2

Such an arithmetic expression have to be put into an XML tag called *Expression* (isn't it amazing?) in its *Value* attribute. Other attributes can be used in Graphs:

- *Label* (optional): label to be print
- *Color* (optional): color of the line/bars

This is a full example of *Data* tag (in *Zone* tag):

```
<Data>
  <Expression Value="1.cpu_user"    Label="User" Color="#FF0000"/>
  <Expression Value="1.cpu_nice"    Label="Nice" Color="#00FF00"/>
  <Expression Value="1.cpu_idle"    Label="Idle" Color="#0000FF"/>
  <Expression Value="1.cpu_system"  Label="System" Color="#00FFFF"/>
  <Expression Value="1.cpu_user+(1.cpu_nice-1.cpu_idle)/10"
    Label="Test" Color="#FF00FF"/>
</Data>
```

4.3.3. A full example

This is a full example of a *Zone* (here a *LineGraph*):

```
<Zone>
  <Title>Memory</Title>
  <Position x="3" y="3" width="3" height="1"/>
  <Type>
    <LineGraph PixelHeight="250">
      <X Graduation="15m" Duration="2h"/>
      <Y Graduation="25000000" LowerBoundary="0" UpperBoundary="150000000"/>
    </LineGraph>
  </Type>
  <Data>
    <Expression Value="1.memory_total"    Label="Total"    />
    <Expression Value="1.memory_used"     Label="Used"     />
    <Expression Value="1.memory_free"     Label="Free"     />
    <Expression Value="1.memory_shared"   Label="Shared"   />
    <Expression Value="1.memory_buffers"  Label="Buffers"  />
    <Expression Value="1.memory_cached"   Label="Cached"   />
  </Data>
</Zone>
```

4.3.4. Placement management



This section is for the moment valid only for the web-Visu.

There is two problems (almost independant): the position and the size of the Zones into a Screen.

The screen is divided in columns and in lines (an HTML table). It is split into a grid, where the cell $x=0,y=0$ is at the left-upper corner of the screen.

All the line have not the same size. The columns could have the same size, but it can happen that not (depending of images or long texts contained in cells).

The cell of a *Zone* is determined by its attributes x and y (determining the left-upper cell of the HTML table) and *width* and *height* (determining the number of lines and columns in the HTML table allocated for this *Zone*).

The size of an image is computed such a way: its width is the value of the *Screen*'s *ColSize* attribute multiplied by the number of columns for this zone; its height is indicated in *PixelHeight*'s *(Line|Bar)Graph* attribute. This dimensions can modify the size of lines and columns of the HTML table.

4.4. Formats

Some formats are common to several attributes or tags.

4.4.1. Duration

This format permits to exprim time duration. It is made of two parts: a number and an unity. The unity can be:

- *s*: seconds (default)
- *m*: minutes
- *h*: hours
- *d*: days
- *w*: weeks
- *M*: month (1 year/12)
- *y*: years (365.242 days)

Examples:

- *15s*: 15 seconds
- *2d*: 2 days

You are allowed to use several unity in the same expression:

- *2d6h*: 2 days + 6 hours
- *1M4d8h*: 1 month + 4 days + 8 hours



4.4.2. Absolute date

This format permits to exprim absolute date. The absolute dates format is: DD/MM/YYYY [hh:mm[:ss]]. The time is optional, the default value is midnight (00:00:00). The seconds are optionals, the default value is 00.

Examples:

- 23/10/2001 12:35:26
- 23/10/2001
- 24/1/2002 1:06

4.4.3. Interval

The Interval format is used to specify a time interval. Its format is : *(last/current) n? (month/year)(es/s)?*.

The third part *(month/year)* specifies the time-unit we are talking about. It can be in plural (es/s)? but it is not mandatory. Week is not available for the moment.

The second part *n?* is optional. It allows you to specify the number of time-units to take. The default value is 1.

The first part *(last/current)* specifies whatever the curent time-unit is included in the interval.

If the current time is 18/10/2001 12:00:00,

- *last month* is equivalent to Start="01/09/2001 00:00:00" Stop="30/09/2001 23:59:59".
- *last 2 monthes* is equivalent to Start="01/08/2001 00:00:00" Stop="30/09/2001 23:59:59".
- *current month* is equivalent to Start="01/10/2001 00:00:00" Stop="18/10/2001 12:00:00".
- *current 2 monthes* is equivalent to Start="01/09/2001 00:00:00" Stop="18/10/2001 12:00:00".

4.4.4. Color

This format permits to exprim coloration for textes/images ... Two formats are allowed.

The first format is a sharp (#) followed by 3 hexa code on 2 digits for red, green and blue.

Examples:

- #FF0000: red
- #00FFFF: yellow
- #000089: dark blue

The second format is a string representing the color depending of the /etc/LogTrend/Visu/rgb.txt instructions (this file have the same format than an /etc/X11/rgb.txt standard file).

Strings like *white*, *grey*, *red*, *blue* ... are accepted.

