Logtrend’s Agent installation’s guide

LogTrend

Logtrend’s Agent installation’s guide

by LogTrend

Revision History

Revision 1.0October, 2001
First Docbook version
Revision 1.1November, 2001
Add action configuration section.

Table of Contents

Chapter 1. Introduction

In the LogTrend project, the Agents collect data from systems, networks etc and send them to the StorageServer which stock them into a data base. An Agent is specific to the system it supervises, ths informations it collects are specific, the way to collect them is specific.

The goal of this LogTrend documentation is to explain how to install an Agent on a system.

The explained steps are common to all agents or have common characteristics for every agents. Several times, some specific actions have to be done for the Agent we install currently; in these cases, the reader will refer to the Agent-specific documentation. Example are given for a generic agent called *WidgetAgent*; remplace this patern by the real name of the agent you want install.

Chapter 2. Packages installation

You need Perl (version >= 5.00503) to run LogTrend. Perl is generaly 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
```

Chapter 3. Configuration

You need to configure your agent before to run it.

Edit the `/etc/LogTrend/WidgetAgent.conf` file. This file compotes 2 parts: a generic one (generic to all agent) ans a specific one (specific to the agent you are installing). This documentation describes the generic part of the configuration file.

This file contains the *Configuration* main XML tag. It is made of two tags: *Generic* and *Specific* tags. This last tag is described in agent-specific sub-section. The tag to be described in this section have to be in the *Generic* tag.

The *AgentDescriptionFile* tag contains the name of the file containing the description of data and alarms. This filename is typicaly `/etc/LogTrend/widgetagentdescription.xml`

The *Source* tag contains the number of the source on which the agent is running. This number identifies in an unique way the system. It is given by the `AddSource` utility (see *StorageServer* documentation).

The *Mail* tag contains information for mails. Its *SMTP* attribute defines the SMTP server where to send mails, the *Admin* attribute defines the email address of the administrator for error reporting and the *Sender* attribute defines the email placed in the *From* field of the email.

The *Agent* tag contains informations about the agent. Its attribute *Number* identifies in an unique way the agent in the source. The couple (source number,agent number) have to be unique in a LogTrend installation. The attribute *Version* is not used for the moment (but don't change it after having declared the agent to the database).

The *Time* tag contains information about the time intervals. This tag contains attributes:

- *Between_Collections*: the time between two collections of the data and the alarms by the agent,
- *Between_Deliveries*: the time between two deliveries of the collected data and alarms to the StorageServer,
- *Before_Warn_If_Server_Not_Responding*: the time before the agent warn the administrator if the StorageServer is not responding.

This parameters are in time-format defined by: a number and a unity. For example 30s or 2d. Unity can be s(second, default), m(minute), h(hour), d(day), w(week), M(month), y(year) (see *Visu* documentation section *Formats* for more details).

The *DataFuture* tag contains information about what to do with the collected data and alarms (the way to delivery them). Three choices:

- saving them into a file (usefull for debug or tests): for that use the tag *Save*
- sending them to the StorageServer by direct TCP-connection: for that use the tag *Send*
- sending them to the StorageServer via the *MailBridge* tool (via mail): for that use the tag *Send* and its *MailForBridge* attribute.

The *DataFuture* could contain the following tags:

- *Save* with its *FileName* attribute which contains the name of the file where to save the data and alarms.
- *Send* for TCP/mail sending. Its attributes are:

- *Host* (mandatory): the host of the StorageServer,
- *Port* (mandatory): the port of the StorageServer (9999),
- *GPGHome* (optional): the GnuPG home directory for LogTrend. By default : `/etc/LogTrend/.gnupg`.
- *MailForBridge* (optional): if you need to use the mail to send data and alarms, you can indicate here the e-mail address where to send messages.

This is examples:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Configuration SYSTEM "Configuration.dtd">
<Configuration>
   <Generic>
      <AgentDescriptionFile>/etc/LogTrend/widgetagentdescription.xml</AgentDescriptionFile>
      <Source>12</Source>
      <Agent Number="2" Version="1" />
      <Time Between_Collections="5m" Between_Deliveries="10m"
            Before_Warn_If_Server_Not_Responding="2h" />
      <!-- Mail address of local administrator -->
      <Mail SMTP="smtp.mydomain.com" Admin="admin@mydomain.com" Sender="a.user@mydomain.com" A
      <DataFuture>
         <!-- <Save FileName="cache.xml" /> -->
         <Send Host="logtrend.mydomain.com" Port="9999" GPGHome="/etc/LogTrend/.gnupg"
               MailForBridge="logtrend@mydomain.com"/>
      </DataFuture>
   </Generic>
   <Specific>
       ... specific tags for the agent ...
   </Specific>
</Configuration>

   or:

      <DataFuture>
         <Send Host="logtrend.mydomain.com" Port="9999" Password="mypassword"/>
      </DataFuture>

   or :

      <DataFuture>
         <Save FileName="cache.xml" />
      </DataFuture>
```

3.1. SimpleAgent

SimpleAgent is an empty agent, made to help you to create a real agent from a exemple. There is no real usefull configuration to make for this agent (and no reason to use it without modifications).

3.2. LinuxAgent

The Linux Agent is aimed at collecting information about a Linux system : CPU, memory, swap, system load avg, number of processes, free/used space/inode on disc devices, network devices statistics, HTTP and FTP response time. Some alarms could be thrown when simple conditions are reached.

NB: Many tags use 'Warning' and 'Error' attributes, they are optionals and indicate the thresholds for alarms.

- *LoadAverage* (Optional): Contains the definitions of the threshold for alarms about system-load average. Attributes: Warning/Error (see NB)
- *TooMuchProcesses* (Optional): Contains the definitions of the threshold for alarms about processes number. Attributes: Warning/Error (see NB)
- *TooMuchZombies* (Optional): Contains the definitions of the threshold for alarms about zombies processes number. Attributes: Warning/Error (see NB)
- *FS* (Optional): For filesystems configuration. Attributes:
  - *Dev* (Optional): Specifies the name of the device on which containd instructions will be applicated. Values could be such as /dev/hda1 ...
  - *Type* (Optional): Specifies the type of the devices on which containd instructions will be applicated. Values could be such as ext2, vfat ...

If no attribute is done for FS, the containd instructions will be applicated to other devices (the one not selected by a Dev or Type attribute). Tag FS could contains Space and Inodes tags.

  - *Space* (Optional) Configuration for space information. Containted tags:
    - *PercentUsed* (Optional): If present, the information about percentage used is collected. Attributes: Warning/Error (see NB) thresholds in percent
    - *NumberFree* (Optional): If present, the information about number free is collected. Attributes: Warning/Error (see NB) thresholds in percent
    - *Quota* (Optional), Attributes:
      - *Users* (=yes/no, default=no): information about number of users out of quota is collected.
      - *Groups* (=yes/no, default=no): information about number of groups out of quota is collected.
  - *Inodes* (Optional) Configuration for inodes information. Same tags than Space.
- *HTTP* (Optional): If present, the information about response time of HTTP service is collected. Attributes: Warning/Error (see NB) thresholds in seconds

- *FTP* (Optional): If present, the information about response time of FTP service is collected. Attributes: Warning/Error (see NB) thresholds in seconds
- *Process* (Optional): If present, an alarm is raised when the specified command line is not found in the current running processes list. Attributes : command, the command line to supervise.

  This tag can contains Actions (see Section 3.5)

Example :

```
<Specific>
   <LoadAverage      Warning="15"  Error="25" />
   <TooMuchProcesses Warning="750" Error="1000" />
   <TooMuchZombies   Warning="5" Error="10" />
   <FS Dev="/dev/hda12"> <!-- By name -->
      <Space>
         <PercentUsed Warning="90"     Error="98" />
         <NumberFree />
      </Space>
   </FS>
   <FS Type="ext2"> <!-- By type -->
      <Space>
         <PercentUsed Warning="90"     Error="98" />
         <NumberFree  Warning="500000" Error="10000" />
         <Quota Users="yes" Groups="yes" />
      </Space>
      <Inodes>
         <PercentUsed Warning="90"     Error="98" />
         <NumberFree  Warning="50000"  Error="1000" />
         <Quota Users="yes" Groups="yes" />
      </Inodes>
   </FS>
   <FS Type="vfat"> <!-- By type -->
      <Space>
         <PercentUsed />
         <NumberFree />
      </Space>
   </FS>
   <FS>  <!-- Default infos for undiscribed fs : -->
      <Space>
         <PercentUsed Warning="98" />
         <NumberFree />
      </Space>
   </FS>
   <!-- HTTP and FTP timeouts thresholds in s -->
   <HTTP Warning="5" Error="30" />
   <FTP  Warning="5" Error="30" />
   <Process command=/usr/sbin/apache>
      <Action>
```

```
            <Execute>
              /etc/init.d/apache start
            </Execute>
           <Mail Subject="[LogTrend] LinuxAgent Alarm"
                 Address="myaddress@mydomain.com" >
                 The LinuxAgent number 3 on source 13 has report the alarm :
                 "/usr/sbin/apache is not running"
           </Mail>
         </Action>
      </Process>
      <Process>/usr/lib/postgresql/bin/postmaster -D /var/lib/postgres/data</Process>
   </Specific>
```

3.3. FtpAgent

The Ftp Agent is a ProFTPD log analyser agent. It can work in two modes : local and remote.

In local mode (when the agent is running on the server), this agent collect data from log files. In remote mode (when the agent is running on a remote machine), the agent just collects file transfert time.

3.3.1. Data and alarms collected in remote mode

- Data:
  - File transfert time : time needed to download a file from the server.
- Alarms:
  - Can not reach host : the server is not available.

3.3.2. Data and alarms collected in local mode

- Data:
  - File transfert time : time needed to download a file from the server.
  - Connected user : number of user connected to the server.
  - Bytes sent : number of bytes sent by the server.
  - Bytes sent daily : number of bytes sent by the server each day.
  - Bytes sent for /path/ : percentage of data transfert dedicated to /path/.
- Alarms:
  - Can not reach host : the server is not available.

- More than /nbr/ users connected.
- Too Login Failure.

3.3.3. Configuration

Ftp Server tag (local and remote mode, required).

```
<FtpServer host="laurent.orsay.atrid.fr" login="anonymous" password="xxxxx" />
```

Give host name and login information to the agent.

Proxy tag (remote mode only, optional).

```
<Proxy>http://proxy:3128</Proxy>
```

FtpAgent use environment variables like http_proxy, ftp_proxy and no_proxy. However, proxy settings can be specified in the configuration file in the <Proxy> tag.

File To Download Tag (local and remote mode, optional).

```
<FileToDownload>/pub/myfiles.tar.gz</FileToDownload>
```

Give the path of the file to download for file transfert time measurement.

Bytes sent for /PATH/ data (local mode, optional).

```
<Report path="/pub/distributions" />
<Report path="/pub/LogTrend" />
```

Report percentage of bytes sent for secified path.

Login Failure Alarm settings.

```
<LoginFailureAlarm log_path="/var/log/proftp.log"
                   nb_failure="2"
                   time_interval="10" />
```

With a such entry, an alarm is raised when more than 2 Login failure occur in 10 seconds. log_path is the path to the proftpd log file.

Example.

```
<Specific>
   <!-- ******************************** -->
   <!--  FtpAgent specific configuration  -->
   <!-- ******************************** -->
   <FtpServer host="laurent.orsay.atrid.fr"
              login="anonymous"
              password="me@mydomain.fr" />
   <Proxy>http://inet:3128</Proxy>
```

```
    <FileToDownload>/pub/myfiles.tar.gz</FileToDownload>
    <LoginFailureAlarm log_path="/var/log/proftp.log"
                       nb_failure="2"
                       time_interval="10" />
    <Report path="/pub/distributions" />
    <Report path="/pub/LogTrend" />
    <NbUserAlarm limit="3" />
</Specific>
```

3.4. HttpAgent

The Http Agent is a Apache log analyser agent. It can work in two modes : local and remote.

In local mode (when the agent is running on the server), this agent collect data from log files and the server-status apache module. In remote mode (when the agent is running on a remote machine), the agent just collects web page download time.

3.4.1. Data and alarms collected in remote mode

- Data:
  - Download time : time needed to download a web page from the server.
- Alarms:
  - Can not reach host : the server is not available.

3.4.2. Data and alarms collected in local mode

- Data:
  - Download time : time needed to download a web page from the server.
  - Requests per seconds : number of requests received by the server each seconds.
  - Bytes per seconds : number of bytes sent by the server each seconds.
  - Active connections : number of active connections.
  - Memory load : memory used by apache.
- Alarms:
  - Can not reach host : the server is not available.
  - Too Much 403 Forbidden : too much forbidden acces from a source.

- Too Much 404 NotFound : too much not found error on an URL.
- server reached MaxClients setting : Apache has reach MaxClients setting.

3.4.3. Configuration

Apache settings tag (local mode only, optional).

```
<ApacheSettings binary_name="apache"
                access_log="/var/log/apache/access.log"
                error_log="/var/log/apache/error.log"/>
```

Give the path to apache log files and the binary name. Default settings (based on Debian distribution settings) :

- binary="apache"
- access_log="/var/log/apache/access.log"
- error_log="/var/log/apache/error.log"

Proxy tag (remote mode only, optional).

```
<Proxy>http://proxy:3128</Proxy>
```

HttpAgent use environment variables like http_proxy and no_proxy. However, proxy settings can be specified in the configuration file in the <Proxy> tag.

URL tag (local and distant mode, required).

```
<URL>http://myserver.mydomain.com/mypage.html</URL>
```

The URL tag specify the location of the page to download for download time measurement.

Too much 403 forbidden alarm settings (local mode only, optional).

```
<ForbiddenAlarm limit="2" time_interval="10"/>
```

With a such entry, an alarm occur when more than 2 Forbidden errors was returned to a client in a time interval of 10 seconds.

Too much 404 not found alarm settings (local mode only, optional).

```
<NotFoundAlarm limit="2" time_interval="10"/>
```

With a such entry, an alarm occur when more than 2 Not Found errors was returned for an URL in a time interval of 10 seconds.

Example.

```
    <Specific>
```

```
    <!-- ******************************** -->
    <!-- HttpAgent specific configuration -->
    <!-- ******************************** -->
    <!-- Settings for red hat, default settings are ready for Debian -->
    <!--<ApacheSettings binary_name="httpd" access_log="/var/log/httpd/access_log"
                        error_log="/var/log/httpd/error_log"/> -->
    <Proxy>http://inet:3128</Proxy>
    <URL>http://laurent.orsay.atrid.fr</URL>
    <NotFoundAlarm limit="2" time_interval="10"/>
    <ForbiddenAlarm limit="2" time_interval="10"/>
  </Specific>
```

3.4.4. Apache configuration (local mode only)

As I said in the agent description, HttpAgent use the apache server status module. You must enable this module in apache configuration files :

- httpd.conf :

```
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
```

- access.conf :

```
<Location /server-status>
   SetHandler server-status
   order deny,allow
   deny from all
   allow from myserver.mydomain.com  <-- Put your server name here
</Location>
```

3.5. Running actions on agents' alarms

Some agents can run actions when alarms is raised. When this feature is available (see agents specific configuration section), actions is configured with the tag *Action*.

```
<Action>
  <Xmessage Display=":0.0">Alarm</Xmessage>
  <Mail Address="me@mydomain.com" Subject="CPU Overloaded">
    The CPU is overloaded on myclient.mydomain.com
  </Mail>
```

```
</Action>
```

This tag describe actions launched when the alarm is raised. There four standard actions. But you can develop customized action.

Available actions are :

3.5.1. Mail

```
<Mail Address="me@mydomain.com" Subject="CPU Overloaded" Sender="sender@mydomain.com" SMTPServ
  The CPU is overloaded on myclient.mydomain.com
</Mail>
```

Send a mail to `Address` with subject `subject`. If specified, SMTPServer is used. Sender is used in the From field of the mail. By default, Sender is equal to Address.

3.5.2. SMS

```
<SMS NationalPhone="0612345678" InternationalPhone="+33612345678">
   System overloaded
</SMS>
```

Send a message to a cellular phone. Phone number must be specified in national and international format.

> **Note:** Currently, SMS Sender uses free web sites to send messages. This method does not really work because it depends of web site structure that change frequently. This action will be soon reimplemented to used more reliable methods like GSM modem.

3.5.3. Syslog

```
<Syslog>System overloaded</Syslog>
```

Add a message in system logs with Syslog

3.5.4. Execute

```
<Execute>/usr/local/bin/myscript.pl</Execute>
```

Run an external command line.

3.5.5. Xmessage

```
<Xmessage Display=":0.0">Alarm</Xmessage>
```

Display a message in a X Window. This action is very usefull while debugging.

Chapter 4. Agent description generation

After configuration of the agent, you need to generate the description of the agent. For that you will run the agent with the *--description*.

This process will generate the file which name is given by the *AgentDescriptionFile* tag.

Chapter 5. Declaring the agent to the database

After configuration and description generation of the agent, you need to declare the agent to the database. For that, you need to run `AgentDescriptionToDB`.

Its parameters are:

- *-f* (mandatory): the name of the agent description file (typicaly `/etc/LogTrend/widgetagentdescription.xml`),
- *-g* (optional): The GnuPG home dir for LogTrend. By default : `/etc/LogTrend/.gnupg`
- *-d* (mandatory): the name of the database (typicaly *logtrend*),
- *-H* (optional): the host name of the database (default localhost),
- *-P* (optional): the post of the database (default 5432),
- *-u* (mandatory): the user for database connection (typicaly *logtrend*),
- *-p* (mandatory): the password for this user,
- *-M* (optional): the MailBridge email address for a declaration via mail (need *-a* option),
- *-a* (optional): the email address of the administrator for error reports,
- *-m* (optional): the email address of the sender (from field of the mail)
- *-s* (optional): the SMTP serveur to use to send mail (default localhost),
- *-S* (optional): agent-to-upgrade source
- *-N* (optional): agent-to-upgrade number

With options *-S old_source_number* and *-N old_agent_number*, you can upgrade an agent. In other words, values continuity is ensure for data having the same name.

Chapter 6. Agent crash and server-side supervisor

To avoid data lost, agents are designed to cache data when the StorageServer is not available. When the agent crashed or is stopped, it try to send data to StorageServer. If the server is not available, the agent send a mail to the agent administrator with not sent data and an error message explaining the problem.

Agents warn their administrator as soon as they have a connection problem. But if the mail can't be deliver to the administrator, you can detect agents' problems with the Agent Superisor complex alarm, which was a server-side agent supervisor. This complex alarm use the detection function *last_data_older_than*, and can detect agent which do not have recent data. You can found an example of a such Complex Alarm config file in the file `AgentSupervisor.xml` in the complex alarm documentation directory. See Complex Alarm's documentation for more details on complex alarm's configuration.

Chapter 7. Running the Agent

To run an agent, you have the choise between to methods:

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