Version 1.00

ZABBIX
Reference
Manual

[for ZABBIX 1.0beta8]

[…unfinished…]

ZABBIX

1 Table of Contents

2 Revision History

Version	**Date**	**Reason**	**Who**
1.0 Release 0001	17/01/03	Initial release	Alexei Vladishev

3 Distribution List

Alexei Vladishev	Author and maintainer of the Manual

4 Introduction

Purpose of this document

The purpose of this document is to provide a comprehensive introduction and overview of ZABBIX, its architecture, the features it offers and their functions.

What you should already know

No particular technical knowledge is required, although an understanding of UNIX is essential.

Who should use this document

Anyone involved in installation and administration of ZABBIX, and anyone else wishing to goin an insight into how it works.

Structure of the document

Introduction

Part1 Overview of the ZABBIX system

Part2 Installation guide

Part3 Configuration guide

Part4 ZABBIX Internals

Appendix

5 Document conventions

The ZABBIX Manual uses the typographical conventions shown in the following table.

Convention	**Description**
ZABBIX	ZABBIX
italic text	Name of file or directory, emphasis
bold text	Programs, applications
NAME	Names
email	Email addresses
script	Shell commands

6 Overview of ZABBIX

What is ZABBIX

ZABBIX was created by Alexei Vladishev, (alex@gobbo.caves.lv). ZABBIX is software that monitors numerous parameters of a network and the servers on that network. ZABBIX is a useful tool for monitoring the health and integrity of servers.

ZABBIX uses a flexible notification mechanism that allows users to configure e-mail based alerts for virtually any event. This allows a fast reaction to server problems. All monitored parameters are stored in a database.

ZABBIX offers excellent reporting and data visualisation features based on the stored data. This makes ZABBIX very useful for capacity planning.

ZABBIX supports both polling and trapping. All ZABBIX reports and statistics, as well as configuration parameters, are accessed through a web-based front end. The web-based front end means that the health of your servers can be assessed from any location. Properly configured, ZABBIX can play an important role in monitoring IT infrastructure. This is equally true for small organisations with a few servers and for large companies with a multitude of servers.

ZABBIX is free of cost. ZABBIX is written and distributed under the GPL General Public License which means that its source code is freely distributed and available for the general public.

What does ZABBIX offer

ZABBIX offers:

- support for both polling and trapping mechanisms
- server software for Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X
- native high performance agents (client software for Linux ,Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X, Windows NT4.0, Windows 2000, Windows XP
- monitoring of server without any agent installed
- secure user authorisation
- flexible user permissions
- web-based interface for viewing and configuration
- flexible e-mail notification of predefined events

Why use ZABBIX

- highly efficient agents for UNIX and WIN32 based platforms
- low learning curve
- high ROI. Downtimes are very expensive.
- Most of other software for monitoring does not have fast agents. Forks, etc, etc ...
- very simple configuration
- all information (configuration, gathered data) is stored in relational database
- high-level service tree
- very easy setup
- support for SNMP (v1,v2)
- visualisation capabilities
- built-in housekeeping procedure

7 About this Manual

This manual is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. This manual is part of ZABBIX software. The latest version of the manual is available at http://zabbix.sourceforge.net.

8 Main Goals and Principles of ZABBIX Development

There are several goals ZABBIX is trying to achieve:

- become recognised Open Source monitoring tool
- create ZABBIX user group, which helps making the software event better

Main principles of ZABBIX development:

- be user friendly
- keep things simple
- use as few resources as possible
- be as fast as possible
- document every single aspect of the software

9 Use of ZABBIX

9.1 Performance monitoring

One of most important uses of ZABBIX is performance monitoring. Processor load, number of running processes, number of processes, disk activity, status of swap space, and memory availability are some of the numerous system parameters ZABBIX is able to monitor. ZABBIX provides a system administrator with timely information about performance of a server. In addition, ZABBIX can produce trend graphs to help identify bottlenecks in system performance.

9.2 Alerting users

Having performance monitoring is good, but it is almost useless without a powerful notification mechanism. With ZABBIX, an administrator can define virtually any possible condition for a trigger, using flexible expressions. Any time these expressions become true (or false), an alert will be emailed to any addresses defined by the administrator.

9.3 Integrity Checking

ZABBIX can easily be used for servers integrity monitoring. All critical configuration files, binaries, kernel, scripts, and web server HTML pages can be monitored by ZABBIX so that the administrator can be alerted to modifications made to these files.

9.4 Logging services

All values of monitored parameters are stored in a database. The collected data can be used later for any purposes.

9.5 Capacity planning

Viewing trends of process load, disk usage, database activity, or other important metrics allows a system administrator to clearly see when the next hardware upgrade should be made.

9.6 High level view of IT resources and services

High level service tree allows create dependencies between various IT resources. Such representation of monitored data makes possible answers to the following questions:

What IT services depends on availability of resource X?

Example: If processor load is too high on server A, then these IT services will be affected: Oracle server, WEB banking, Online transaction processing, etc.

What resources specific IT service depends on?

Example: WEB portal may depend on the following resources:

1. processor load on server A
2. connection to ISP provider
3. disk space on volume /data on server A
4. availability of Oracle DB engine on server B
5. speed of execution of user requests
6. availability of Apache server on server C
7. etc etc

Such dependency tree helps identify weak points in IT infrastructure.

Example: If several critical services offered by IT department depends on, for example, availability of disk space on some server, then it’s time to think about distribution of the volume across different servers or disk arrays to eliminate possible risks.

9.7 Other

- availability analysis
- graphical representation of collected information

10 Installation and Configuration Guide

10.1 How to Get ZABBIX

Check the ZABBIX Home Page at http://zabbix.sourceforge.net for information about the current version and for downloading instructions.

10.2 Requirements

Due to security requirements and mission-critical nature of monitoring server, UNIX is the only operating system that can consistently deliver the necessary high performance, fault tolerance and resilience. ZABBIX operates on market leading versions.

10.2.1 Supported Platforms

ZABBIX is tested on the following platforms:

- Linux 2.xx
- Solaris 2.xx
- HP-UX 11.xx
- AIX 4.3
- Free BSD 4.3
- SCO Open Server 5.0.5

Note: Note that ZABBIX may work on other Unix-like operating systems.

10.2.2 Software Requirements

ZABBIX is built around modern Apache WEB server, MySQL or PostgreSQL database engine, and PHP scripting language.

The following software is required to run ZABBIX:

- Apache

  Version 1.3.12 or later required. Apache can be found at http://www.apache.org

- MySQL (or PostgreSQL)

  Version 3.22 or later required. MySQL can be found at http://www.mysql.com

- PostgreSQL (or MySQL)

Version 7.0.2 or later required. PostgreSQL can be found at http://www.postgresql.org

- MySQL or PostgreSQL development files (headers and libraries).

Note: Usually provided as part of mysql-dev or postgresql-dev packages.

- PHP

  Version 4.0 or later required. PHP and its modules can be found at http://www.php.net. PHP must be compiled as Apache module.

- PHP GD module

  The module is required for displaying graphs and maps.

- PHP 4.0 MySQL or PostgreSQL module

- GNU Make

  Required for building ZABBIX or its agents from sources. In case if pre-built binaries are used, GNU make is not necessary.

- WEB browser on client side

  Support for HTML and PNG images required. MS Explorer (5.xx and 6.xx), Mozilla 1.x and Netscape 4.7x work perfectly. Cookies and JavaScript must be enabled.

- NET-SNMP (or UCD-SNMP) library and header files

  Required for support of SNMP agents. *Optional.*

10.2.3 Required software considerations

ZABBIX may work on previous versions of **Apache**, **MySQL**, **PostgreSQL** as well. It has been tried on **PostgreSQL** 6.5x with no luck.

PostgreSQL crashes after one to three minutes of **zabbix_suckerd** running. So, if using **PostgreSQL**, version 7.0.2 or later is strongly recommended.

PostgreSQL vs MySQL

Regarding the choice between **PostgreSQL** and **MySQL**, **MySQL** is recommended for several reasons:

- recent benchmarks using ZABBIX clearly show that **PostgreSQL** (7.1.x) is at least 10 times slower than **MySQL** (3.23.29)

Note: These results are predictable. **ZABBIX** mainly uses simple SQL statements like single row INSERT, UPDATE, and simple SELECT operators. In such environment, use of advanced SQL engine (like **PostgreSQL**) is overkill.

- no need to constantly run resource-hungry command "*vacuum*" **MySQL**

MySQL MyISAM table type is not a best choice because of lack of row-level locking. So, InnoDB is recommended. InnoDB uses row level locking. Therefore, most of SQL statements can be executed simultaneously without delay. As a result, performance of InnoDB is better than MyISAM. For more information about **MySQL** see http://www.mysql.com.

If you do use **PostgreSQL**, **zabbix_suckerd** will periodically (defined in *HousekeepingFrequency*) execute command "*vacuum analyze*".

For more information about **PostgreSQL** see www.postgresql.org.

10.2.4 Memory Requirements

ZABBIX requires both physical and disk memory. 32 Mb of physical memory and 20 Mb of free disk memory could be good starting point. However, the amount of required disk memory obviously depends on the number of hosts and parameters that are being monitored. If you're planning to keep a long history of monitored parameters, you should be thinking of at least a couple of gigabytes to have enough space to store the history in the database.

Each ZABBIX daemon process requires several connections to a database server. Amount of memory allocated for the connection depends on configuration of the database engine.

Note: **zabbix_trapperd** can be configured so that connection to the database will only be made during process of new value. See description of parameter *DBConnectOnEach*.

Remember, the more physical memory you have, the faster the database (and therefore ZABBIX) works!

10.2.5 Time synchronisation

It is very important to have precise system date on server with ZABBIX running. **timed** is one of most popular daemons that synchronises the host's time with the time of other machines.

10.3 Structure of ZABBIX distribution

ZABBIX distribution:

- *doc*

The directory contains this Manual in different formats

- *src*
- *include*
- *misc*
- *misc/init.d*

The directory contains startup scripts.

- *frontends*
- *create*
- *upgrades*

10.4 Installation procedure

Basic installation of ZABBIX usually takes no more than 15 minutes.

10.4.1 Server side

Create the ZABBIX superuser account

This is the user the server will run as. For production use you should create a dedicated unprivileged account ("zabbix" is commonly used). Running ZABBIX as "*root*," "*bin*," or any other account with special rights is a security risk. Do not do it!

Note: ZABBIX daemon processes (**zabbix_suckerd** and **zabbix_trapperd**) are protected from being run under *root* account.

Untar ZABBIX sources

Use command:

```
gunzip zabbix.tar.gz; tar -xvf zabbix.tar
```

Create the ZABBIX database.

ZABBIX comes with SQL scripts used to create the required database schema and also to insert a default configuration. There are separate scripts for **MySQL** and **PostgreSQL**.

For **MySQL**:

```
mysql -u<username> -p<password>
>create database zabbix;
>quit;
cd create/mysql
cat schema.sql |mysql zabbix -u<username> -p<password>
cd create/data
cat data.sql |mysql zabbix -u<username> -p<password>
```

For **PostgreSQL**:

```
psql -U <username>
create database zabbix;
> \q
cd create/postgresql
cat schema.sql|psql -U <username> zabbix
cd create/data
cat data.sql|psql -U <username> zabbix
```

Configure and compile the source code for your system

The sources must be compiled for both the server (monitoring machine) as well as the clients (monitored machines).

To configure the source for the server, you must specify which database will be used.

```
./configure—with-mysql # for MySQL
```

or

```
./configure—with-pgsql # for PostgreSQL make
```

Note: Use flag *—enable-static* to statically link libraries. If you plan to distribute compiled binaries among different servers, you must use this flag to make these binaries work without required libraries.

However, if you want to compile client binaries only, run:

```
./configure
```

Parameter *—enable-static* may be used to force static linkage.

Make everything:

```
make
```

Copy created binaries from *bin/* to */opt/zabbix/bin* or any other directory
Other common directories are */usr/local/bin* or */usr/local/zabbix/bin*.

Configure /etc/services

The step is not real requirement. However, it is recommended. On the client (monitored) machines, add the following lines to */etc/services*:

```
zabbix_agent 10000/tcp
zabbix_trap 10001/tcp
```

Configure /etc/inetd.conf

If you are going to use non-daemon versions of ZABBIX trapper, add the following line to */etc/inetd.conf*:

zabbix_trap stream tcp nowait.3600 zabbix /opt/zabbix/bin/zabbix_trapper

If you plan to use **zabbix_agent** instead of the recommended **zabbix_agentd**, the following line must be added:

zabbix_agent stream tcp nowait.3600 zabbix /opt/zabbix/bin/zabbix_agent

Restart inetd

```
killall -HUP inetd
```

Configure WEB interface

As user *zabbix* (or whatever you decided to name your dedicated ZABBIX account), do the following:

Change these values in *frontends/php/include/db.inc*:

```
$DB_TYPE =”POSTGRESQL”; /* Or “MYSQL” for MySQL */
$DB_SERVER =”localhost”;
$DB_DATABASE =”zabbix”;
$DB_USER =””;
$DB_PWD =””
```

Copy the **PHP** source files to a place where your web server can get to it. Perhaps */home/zabbix/html* or */home/zabbix/public_html* or */var/www/html/zabbix*, etc. For example,

```
mkdir /home/zabbix/html
cp -R frontends/php/*
/home/zabbix/html/
```

Configure /etc/zabbix/zabbix_agent.conf

You need to configure this file for every host having **zabbix_agent** installed. The file should contain IP address of ZABBIX server. Connections from other hosts will be denied. You may take *misc/conf/zabbix_agent.conf* as example.

Configure /etc/zabbix/zabbix_agentd.conf

You need to configure this file for every host having **zabbix_agentd** installed. The file should contain IP address of ZABBIX server. Connections from other hosts will be denied. You may take *misc/conf/zabbix_agentd.conf* as example.

Configure /etc/zabbix/zabbix_suckerd.conf

For small installations (up to ten monitored hosts), default parameters are fine. However, you should change default parameters to get maximum performance from ZABBIX. See section [Performance tuning] for more details.

You may take *misc/conf/zabbix_suckerd.conf* as example.

Configure /etc/zabbix/zabbix_trapperd.conf

For small installations (up to ten monitored hosts), default parameters are fine. However, you should change default parameters to get maximum performance from ZABBIX. See section [Performance tuning] for more details.

You may take *misc/conf/zabbix_trapperd.conf* as example.

Initial Configuration

Once your ZABBIX installation is complete, you will need to configure ZABBIX. Point your web browser to the location you installed the **PHP** scripts to. (For example, if you copied the PHP files to */home/zabbix/public_html* or */home/zabbix/html*, point your browser to http://localhost/~zabbix.

Once there, click on the CONFIG link. You will be prompted for a username and password. ZABBIX comes pre-configured with an administrative account: Enter the username 'Admin' and leave the password field empty.
Add the configuration of your SMTP server of choice. Next, click on the "USERS" link. Add yourself to list of ZABBIX users. Don't forget to add "read/write"
and "add" default permissions to the user. After you create your account, click the "Media" link beside your account listing, and add your email address.
Next, delete the default Admin user or specify password. Click on the 'HOSTS' link to add new hosts to be monitored. For each host you add, be sure that you've installed the agent software on the host. If you have not yet done this,
set the status to "Not monitored". When a host is added, a list of all possible parameters for the host will be automatically added.
After you've added your desired hosts, click on the "ITEMS" link to modify the details of the monitored parameters. You may disable or delete monitored parameters or change how often the parameters are checked.
Next, click on the 'TRIGGERS' link to change the triggers relating to monitored parameters. You can change threshold values, disable or delete triggers, and set up actions (email notification).
When you set up an email notification, you may modify the subject and message body as you wish. You can use macros in both email subject and email body. For example, the subject for a warning message could look like:

Processor load on www.sf.net is {www.sf.net:system[procload].last(0)}

In this case, you will receive message with subject like "Processor load on www.sf.net is 0.85"

Click on the "MAPS" link to set up a network map. A map will help you quickly identify any problems with your monitored machines.
Click on the "GRAPHS" link to set up a graph. A graph provides quick historical statistics on one or more monitored statistics.
Finally, do not forget to set up IT Services to get high level view of servers and applications.
Once you have an initial configuration created, run zabbix_suckerd on the server machine:

```
/opt/zabbix/bin/zabbix_suckerd
```

and run **zabbix_agentd** on all monitored machines

```
/opt/zabbix/bin/zabbix_agentd
```

You should not run **zabbix_agentd** if you have chosen to use **zabbix_agent**!

10.4.2 Client side

Create the ZABBIX account.

This is the user the agent will run as. For production use you should create a dedicated unprivileged account ("zabbix" is commonly used). ZABBIX agents have protection against running under *root* account.

Untar ZABBIX sources.

Use command:

```
gunzip zabbix.tar.gz; tar xvf zabbix.tar
```

Configure and compile the source code for your system.

The sources must be compiled for the client only.

To configure the source for the client:

```
./configure
```

Note: Use flag *—enable-static* to statically link libraries. If you plan to distribute compiled binaries among different servers, you must use this flag to make these binaries work without required libraries.

Build agent:

```
make
```

Copy created binaries from *bin/* to */opt/zabbix/bin* or any other directory
Other common directories are */usr/local/bin* or */usr/local/zabbix/bin*.

Configure /etc/services

The step is not real requirement. However, it is recommended.

On the client (monitored) machines, add the following lines to */etc/services*:

zabbix_agent 10000/tcp
zabbix_trap 10001/tcp

Configure /etc/inetd.conf

If you plan to use **zabbix_agent** instead of the recommended **zabbix_agentd**, the following line must be added:

zabbix_agent stream tcp nowait.3600 zabbix /opt/zabbix/bin/zabbix_agent

Restart inetd

```
killall -HUP inetd
```

Configure /etc/zabbix/zabbix_agent.conf

You need to configure this file for every host having **zabbix_agent** installed. The file should contain IP address of ZABBIX server. Connections from other hosts will be denied. Note, that no end of line character should present in the file.
You may take *misc/conf/zabbix_agent.conf* as example.

Configure /etc/zabbix/zabbix_agentd.conf

You need to configure this file for every host having zabbix_agentd installed. The file should contain IP address of ZABBIX server. Connections from other hosts will be denied. You may take *misc/conf/zabbix_agentd.conf* as example.

Run zabbix_agentd on all monitored machines

```
/opt/zabbix/bin/zabbix_agentd
```

Not that you should not run **zabbix_agentd** if you have chosen to use **zabbix_agent**!

10.5 Development Environment

Debian Potato Linux and Debian Woody Linux (both on Intel hardware) are used as primary development platform for ZABBIX.

It means that if you have difficulties choosing between Linux and other OS, go for Debian Linux (http://www.debian.org) - you will get better support.

ZABBIX 1.0beta4 was developed and tested on FreeBSD 4.4. Thus, FreeBSD (http://www.freebsd.org) is also safe choice.

10.6 Platform specific notes

10.6.1 FreeBSD

Default FreeBSD settings (at least, FreeBSD 4.4) cannot handle amount of TCP connections ZABBIX is able to generate. To solve this issue, define NoTimeWait parameter in both *zabbix_suckerd.conf* and *zabbix_trapperd.conf*. This will greatly decrease number of sockets in TIME_WAIT state. As an alternative solution, kernel parameters can be tuned using *sysctl* mechanism.

11 Upgrading

The upgrade procedure is quite simple. New binaries and frontend should be installed according to latest installation instructions.
In order to update database structure, the following steps should be performed. Note that before applying database patches, all ZABBIX processes must be stopped. Also, the process can take from 0 seconds (if no patches required) up to several hours. For production installations database backup is required!

11.1 General Procedure

Go to the *upgrades/dbpatches* directory. In this directory are subdirectories named according to a version upgrade (e.g. *1.0beta3_to_1.0beta4*). Enter the directory corresponding to your upgrade (if you are upgrading through multiple versions, you will need to apply the upgrades one at a time). Depending on which database you use:

```
cd mysql; cat patch.sql |mysql zabbix -u<username> -p<password>
```

or

```
cd postgresql; cat patch.sql|psql -U <username> zabbix
```

Finally, read version specific notes below for any extra procedures and useful information.

11.1 ChangeLog

11.1.5 ZABBIX 1.0beta4

initial support for service tree
vacuuming is made as part of housekeeping process for PostgreSQL
many improvements for PostgreSQL support
added item which reflects host status (0-normal,2-unreachable)
fixed function apply_actions()
added Next100 and Prev100 for alerts and alarms
check permissions for Availability report
added parameter SenderFrequency to zabbix_suckerd configuration file
added frontends/include/defines.inc.php
added column services.algorithm
list of test cases added to the Manual
emails to be sent in case of TRUE-UNKNOWN-FALSE and v.s.
one zabbix_suckerd to send alerts
minimum number of zabbix_suckerd StartSuckers is set to 3

added alerts.retries
added alerts.status
status of triggers changes if host/item status has been changed
status of triggers is set correctly even if ZABBIX is restarted
php/include/*inc renamed to php/include/*inc.php
check of source IP address for trapped values
added include/security.h and include/security.c
added items.trapper_hosts
housekeeping period for items to be set in days
authorisation without user name/password
default user “guest”
added table ‘sessions’
added index triggers.value and trigger.status
added column alarms.value
removed column alarms.istrue
added columns triggers.status
added columns triggers.value
removed column triggers.istrue
support for—enable-static for ./configure
housekeeping period for alarms and alerts to be set in days
created directory upgrades/dbpatches/1.0beta3_to_1.0beta4
more anvanced algorithm for updating disable_until field
problem with “Lost connection to DB” solved
function DBis_empty(), code cleanup
unreachable parameters will not be shown in Queue
support for check_service[service,port]
advanced change of host status in zabbix_suckerd
added support of UserParameter(s) returning non-numeric values
doc/manual.lyx will not be included in ZABBIX release
chnages to make ZABBIX compile in non-GCC environments
unreachable hosts will be clickable in system maps
fixed double posting of the same trigger status into alarms
fixed macros substitution in email alerts
removed option -Wall from all Makefiles
removed “//” comments from ZABBIX code
disabled experimental code by commenting #define PERF
improvements in SNMP handling for zabbix_suckerd
fixed check for SNMP library in configure.in
template host set to empty in Host definition form
support for DisableHousekeeping for zabbix_suckerd
protection from items.delay set to 0
values provided by an agent can be less than zero

11.1.6 ZABBIX 1.0beta5

added parameter DBHost
added upgrades/dbpatches/1.0beta4_to_1.0beta5/ with patches
added support for disktotal[*]
added support for inodetotal[*]
added support for *,/,+, and - in trigger expressions
update item status to UNSUPPORTED in case if SNMP support was not included into zabbix_suckerd
added mysql_init() to DBconnect()

fixed Next100 and Prev100 for case if some hosts are hidden
added select for latest values
delete related services if trigger is deleted
fixed problem with substitution of macros for messages
eliminated DBis_empty() by replacing to DBnum_rows()
MAX_STRING_LEN increased to 4096
fixed zabbix_log(). Possible coredump if data contains
%s, etc
changed grid size for all graphs
added frontends/php/compare.php
redesigned all graphs (changed colors, etc)
host selection for availability report
more fixes to send_email()
fixed crash in case if MySQL server gone (support for signal
PIPE). Thanks to Arturs.
fixed duplicate triggers in Availability Report
fixed division by zero in Availability Report
added frontends/php/chart3.php
small fix for form “Configuration of network map”
for host templates, %s can be used in trigger description
to be substituted with new host name
empty result for DBselect() is always checked
fixed send_email(). According to RFC, DATA should be sent
with followed EOL
fixed memory leak in main_alerter_loop(). Added DBfree_result().
fixed missing close of socket in case if connection to
SMTP server is rejected

11.1.7 ZABBIX 1.0beta6

added support for check_service(service,ip,port)
added support for check_port(ip,port)
fixed check_right(). Thanks to Daniel Higgins.
more changes to send_email(). Thanks to Daniel Higgins.
added file TODO
status will be changed to UNCREACHABLE only in case of
three network errors
added column hosts.network_errors
support for parameters netloadin[*] and netloadout[*]
added src/zabbix_agent/stats.c and stats.h
availability graph for IT Services
improvement in user rights handling routines
if trigger status is UNKNOWN, set function.lastvalue to
NULL
fixed permission check for all *.php
Priority was renamed to Severity
added availability graph for IT Services
if trigger status is UNKNOWN, set function.lastvalue to
NULL
fixed permission check for all *.php
Priority was renamed to Severity
added graph to availability report
added frontends/php/chart4.php
availability report will not show trigger expressions
added creation of table sessions to migration scripts for
1.0beta4
no dependency on register_globals in php.ini

zabbix_suckerd to write “Connection reset by peer” to LogFile

* \n change to \r\n in send_email()
added table service_alarms
fixed problem with parsing of trigger expressions. [.-.]
changed to -
support for function nodata() added
minimum number of zabbix_suckerd that can be started is 4
simplified function update_triggers()
fixed situation when a Service is linked to a trigger.
Service name is correct now.
support for new trigger function str()
changed type of functions.parameter to varchar(255)
additional colors for graphs
in chart3.php: group by <complex expression> changed to
group by 1
added housekeeping procedure for table sessions
changed width=\"X%\" to width=X% in *.php
status of a trigger will blink if it was changes during
last 300 seconds
ZABBIX daemons to update triggers.lastchange when triggers.value
is changed
added misc/init.d/debian/zabbix-[suckerd|trapperd]
removed misc/init.d/debian/zabbix-server
send_email() will timeout after 10 seconds
DBadd_alarm() renamed to add_alarm()
support for SCO OpenServer. Thanks to Alexender Kirhenstein.
./configure will correctly define socklen_t
<center> changed to <div align=center>
reason of problem in screen IT Services
added frontends/php/helpdesk.php
added “...#form” to button Change in form Configuration
of Users
“Just for information” changed to “Information”
fixed ./configure to correctly find zlib
housekeeping to delete no more than HousekeepingFrequency*3600
records from table history and history_str at once (MySQL
only)

changed definition of functions.lastvalue to allow NULL
added upgrades/dbpatches/1.0beta5_to_1.0beta6/ with patches
added misc/init.d/debian/zabbix_agent

11.1.8 ZABBIX 1.0beta7

startup scripts for RedHat 8.0
refresh rate is doubled for user ‘guest’
mass insert and delete for items
support for SNMPv2. New item type introduced.
improvements in housekeeping procedure
improvements for chart.php
added severity to pre-defined triggers
SO_LINGER is not default socket() option for zabbix_sender
recvfrom() and sendto() were changed to read() and write()
fixed housekeeping procedure. Upgrade carefully!
support of template hosts. Remove tables hosts_templates

and triggers_template.
Show-Show-Show renamed to Graph-Trend-Compare in latest.html.
fixed buffer overflow in zabbix_log().
availability reports for trapper-based triggers.
fixed possible zabbix_suckerd crash. It may happen if socket()
returns 0.
added new Host status - Template(3).
fixed group “all” in latest.php
do not dump “Expression cannot be evaluated” in case if
function.lastvalue is NULL. Logic was changed a bit, but
the message is still there.

show who is online in User config form.
when ran first time, some triggers becomes TRUE (like files
was changed).
This is already fixed. This bug does not exist.
fixed bug in queue.php select statement. Trapper type will
not be selected.
when W2000 with SNMP agent restarted, items will becomes
Not Supported. There is no such bug in CVS code anymore.
map will show host status in case of trigger based on trapper
fixed bug in zabbix_suckerd’s select statements. Trapper
type will not be selected.
new protocol for ZABBIX agents, Support for ZBX_ERROR and
ZBX_NOTSUPPORTED
about.php: version of ZABBIX was changed to 1.0beta7
items.php: “Delay” renamed to “Update interval”
fixed precision issues in zabbix_agent(d) and zabbix_suckerd.
All floats converted to double.
fixed selection of icon and host in sysmap.php. Thanks
to Sebastien “slix” Lienard.
monitoring of servers without ZABBIX agent installed. New
item type: Simple Check.
improved performance of pinger.pl
added support of ‘:’,’\’ for flexible parameters (example,
cksum[c:\a.bat])
added copyright notice to source files
pinger.pl will use mass-send of collected data. No unnecessary
forks anymore.
all <? ?> were changed to <?php ?>. No changes in php.ini
required anymore.
added ZABBIX agent for WIN32 platforms. Thanks to Victor
Kirhenstein.
fixed coredump in get_lastvalue(). Thanks to Rickard Plars.
delay renamed to “Update interval (in sec)” in item definition
form
create/mysql/schema.sql will create tables with type=InnoDB
by default

added upgrades/dbpatches/1.0beta6_to_1.0beta7/ with patches
added item type TRAPPER
fixes for PostgreSQL
added validation of host name
default value of StartSuckers is set to 4
fixed Select in Status of Triggers
precision of Change is set to 2 in Latest Values

* --enable-static to work on all platforms
zabbix_sender to be able to get information from stdin
support fo JavaScript required now
automatic resize of all graphs depending on screen size
got rid of message “Bad protocol identification ‘0’” when
doing check_service[ssh]
performance improvements for zabbix_suckerd
added support of CSS
support of processor load on Solaris 2.6. Thanks to Mariusz.
improvements for user-defined graphs. Thanks to Mariusz.
improvements for System Maps. Thanks to Mariusz.
support for host groups
added table hosts_groups
added table groups
./configure to find PostgreSQL’s includes in /usr/local/include
basic support for log rotatation
added additional error checks in get_value_snmp()

11.1.9 ZABBIX 1.0beta8

- very basic support for SNMP trapping
- added misc/snmptrap/snmptrap.sh
- group selection for Item configuration form
- added support for SLA level in IT Services
- added fields services.showsla and services.goodsla
- added field items.snmp_port
- user-defined port number for SNMP requests
- fixed SQL errors when adding host to IT Services
- added chart_sla.php
- fixed incorrect handling of timeouts when sending emails
- fixed selection of recipient in trigger actions
- fixed Plain Text values
- fixed improper handling of special characters i.e. '\/"
- fixed Last100 and Prev100 for both Alarms and Alerts
- fixed "STATUS OF ZABBIX". Command was changed to 'ps aux'
- personalisation (remember default port number for hosts)
- added table 'profiles'
- assured uniqueness of host names
- added upgrades/dbpatches/1.0beta7_to_1.0beta8/ with patches
- mass update for triggers (activate, deactivate, delete)
- all sendto() and recvfrom() were changed to write() and read() in function send_mail()
- fixed bug in host addition procedure when template host is used
- different colors for different status of items, triggers, users

- WIN32 agent. Added support for multiple servers
- WIN32 agent. Changes in communication protocol: on errors now returned special
- WIN32 agent. Support for new client-server protocol
- WIN32 agent. Added proc_info[] parameter
- WIN32 agent. Added memory[cached] parameter (supported only on Windows XP)
- WIN32 agent. Added logging to system Event Log
- WIN32 agent. PDH-related error messages are now more specific
- WIN32 agent. Listener thread now will not exit on any accept() error
- WIN32 agent. Added some parameters for agent statistics monitoring: number of accepted and rejected connections, processing errors, etc. (Victor)
- WIN32 agent. Fixed bug in calculation of md5_hash[] and cksum[] parameters for zero-length files
- WIN32 agent. md5_hash[] and cksum[] now ruturns UNSUPPORTED for files larger than 64MB (due to high performance impact caused by checksum computing for large files).

12 ZABBIX Architecture

12.1 Overview

ZABBIX is structured in a client-server architecture. ZABBIX is capable of running in either standalone or inetd execution mode.

Note: The standalone method is the recommended approach.

The ZABBIX server processes can be run in a polling or listening configuration or both. These methods differ in the approach that they utilise and affect what processes must be run. In the Server Polling Method the ZABBIX Server polls established ZABBIX clients (preferred.) In the Server Listen Method the ZABBIX server awaits connections from active ZABBIX clients.

ZABBIX server processes consist of a standalone process used for the Server Polling Mode (**zabbix_suckerd**) and a choice of standalone or **inetd** processes for the Server Listen Method (**zabbix_trapperd** or **zabbix_trapper**).

ZABBIX agents (clients) consist of a choice of standalone or **inetd** processes for the Server Polling Method and a standalone process for the Server Listen Method (There is no **inetd** process for the Server Listen Method because when using that method the client must initiate the connection.)

If the server is used in the Server Polling Mode then the ZABBIX listening client is used (**zabbix_agentd** or **zabbix_agent**), this client awaits connections from the server. If the server is used in the Server Listen Mode then the ZABBIX active client is used (**zabbix_sender**), this client initiates connections to the server.

ZABBIX is written in **C**. WEB frontend is written in **PHP**.

12.2 Server Approach Modes

ZABBIX supports two methods of server role activity. The server may be used in a polling mode (Server Polling Method) or the server may be using in a listening mode (Server Listen Method) or both.

These methods are not mutually exclusive and can be used in conjunction.

The standard method utilised is the Server Polling Method in most instances. In this method the server is setup and configured to be aware of clients and then the server polls the clients at defined intervals.

The additional Server Active Method can also be used. This is generally used in a complimentary fashion to have clients send information to the server on a predetermined basis as defined by the administrator. For example, the Server Listen Method can be used to have a client machine send information to the server when a backup process has begun and again when it has ended. In this manner the client dictates the interval of the information and sends information when specified events occur rather than the server requesting information at specified intervals.

17.2.1 Server Polling Method (Server connects to Client)

In this method the ZABBIX server, **zabbix_suckerd**, initiates connections to a client agent placed on the monitored host (**zabbix_agentd** or **zabbix_agent**) on a periodic basis. The server requests specific information (processor load, free memory, available inodes, etc, whatever is being monitored.)
The client provides the server with the requested information and the server, in turn, stores the received values in the database.

Note: In this case the traditional sense of client is turned around. The client is actually the server. That is to say that the connection is made from the main ZABBIX server (**zabbix_suckerd**) to the ZABBIX client (**zabbix_agentd**), and therefore the ZABBIX client is acting like a traditional server and listening for connections and responding; and the ZABBIX server is actually acting as a traditional client and initiating a connection.

12.2.2 Server Listen Method (Client connects to Server)

In this method the ZABBIX client (**zabbix_sender**) initiates connections to a ZABBIX server (**zabbix_trapperd** or **zabbix_trapper**) on an event driven basis.

12.3 Theory of operations

17.3.1 Data exchange protocol

zabbix_suckerd zabbix_agent(d)

Server ->[key]->Agent
Agent->[result]->Server

If received result is equal to ‘ZBX_NOTSUPPORTED’ then the agent does not support processing of required parameter. ‘ZBX_ERROR’ indicates that agent was unable to return requested parameter.

zabbix_sender zabbix_trapper(d)

SSender ->[server:key:value]->Trapper
Trapper->[OK]->Sender

12.4 Database

Database use plays a very important role in the ZABBIX application. The entire history of received parameter values is stored in the database. In addition, the database is the sole source of configuration parameters for the application. ZABBIX performance depends highly on the efficiency and speed of the database used by ZABBIX.

12.5 ZABBIX Processes

ZABBIX supports several different processes that perform specific functions depending on the mode of execution that is utilised (**inetd** vs. standalone.)

12.5.1 Standalone Mode Processes

zabbix_suckerd

The **zabbix_suckerd** process resides on the server and periodically connects to agents (either ZABBIX native or SNMP) to get the values of parameters being monitored. After receiving the values, the process recalculates the status of ZABBIX triggers. Also, the process sends alerts to users if required.

If **zabbix_suckerd** is unable to get value from an agent because of network problems, the value will be checked next time only after DELAY_ON_NETWORK_ERROR (defined in *include/common.h*, default value is 60 seconds.) Host status will be changed to UNREACHABLE.

If an agent does not support requested parameter, **zabbix_suckerd** will change status of the parameter to NONSUPPORTED and will not try to get its value anymore.

By default, **zabbix_suckerd** forks five copies of itself. This means that ZABBIX will be able to get four (five minus process used for housekeeping) monitored parameters simultaneously. To modify the number of forked processes, change the parameter *StartSuckers* in */etc/zabbix/zabbix_suckerd.conf*. Note that each forked process requires one connection to the database. Make sure that the database is able to provide this number of connections.

One **zabbix_suckerd** process is used for housekeeping purposes only. The housekeeping process periodically (hourly, by default) deletes outdated information from the tables history, alarms, alerts and sessions.

Housekeeping process will delete all records from table alarms older than value defined in ZABBIX configuration and delete all records from tables alerts and alarms older than value defined in ZABBIX configuration.

Parameter *HousekeepingFrequency* from *zabbix_suckerd.conf* defines how often housekeeping procedure must be executed.

No command line switches are supported by **zabbix_suckerd**.

The **zabbix_suckerd** process is run as a daemon under a non-privileged user account, usually *zabbix*.

Syslog or a file is used to store debug information for the process.

zabbix_agentd

The **zabbix_agentd** process resides on the host being monitored. Its purpose is to provide request information to **zabbix_suckerd**.

When executed, the **zabbix_agentd** process forks itself (five copies by default). Load is balanced between each copy of the process.

When a connection to **zabbix_agentd** is made, the agent determines if the connection is coming from an authorised server. The *zabbix_agentd.conf* file must exist and contain the IP address (or list of IP addresses) of ZABBIX server. Connections from other IP addresses are rejected.

No command-line switches supported by **zabbix_agentd**.

The process is run as daemon under non-priveledged user account, usually *zabbix*.

Syslog or a file is used to store debug information for the process.

List of supported parameters can be extended by using configuration file parameter *UserParameter*.

ZABBIXW32.exe (WIN32 agent)

ZabbixW32 is ZABBIX agent for Win32 systems. It will work on Windows NT 4.0,

Windows 2000 and Windows XP. ZabbixW32 doesn't supposed to work on other Windows platforms.

The ZabbixW32.exe process resides on the host being monitored. Its purpose is to provide request information to zabbix_suckerd.

Installation

Installation is very simple and includes 3 steps:

1. Unpack ZabbixW32.exe
2. Create configuration file c:\zabbix_agentd.conf (it has the same syntax as for UNIX agent). See src/zabbix_agent_win32/README for more detailed information!
3. Run command “ZabbixW32.exe install” to install ZABBIX agent as a service.

Now you can use Control Panel to start agent's service or run

“ZabbixW32.exe start”

Command line syntax

```
Usage: zabbixw32 [options] [command]


Where possible commands are:
standalone
: Run in standalone mode
start
: Start ZABBIX Win32 Agent service
stop
: Stop ZABBIX Win32 Agent service
install
: Install ZABBIX Win32 Agent as service
remove
: Remove previously installed ZABBIX Win32 Agent service
help
: Display help information
version
: Display version information


And possible options are:
config <file> : Specify alternate configuration file


(default is C:\zabbix_agentd.conf)
```

zabbix_trapperd

The process collects information passed by active agents (**zabbix_sender**).

By default, **zabbix_trapperd** forks ten copies of itself. The parameter TRAPPERD_FORKS in *include/common.h* can be changed to modify the number of forked processes.

No command-line switches are supported by **zabbix_trapperd**.

The process is run as daemon under non-privileged user account, usually *zabbix*.

Syslog or a file is used to store debug information for the process.

12.5.2 Inetd Mode Processes

zabbix_agent

The **zabbix_agent** process resides on the host being monitored. Its purpose it to provide requested information to **zabbix_suckerd**. The **zabbix_agent** process is designed to be used with **inetd** daemon.

Execution time of the process is limited. If the agent is not able to get the requested information within *Timeout* seconds defined in */etc/zabbix/zabbix_agent.conf*, it kills itself.

When a connection to **zabbix_agent** is made, the agent reads */etc/zabbix/zabbix_agent.conf* in order to determine if the connection is coming from an authorised server. The *zabbix_agent.conf* file must exist and contain the IP address of ZABBIX server. Connections from other IP addresses are rejected.

Use of **zabbix_agentd** (see below) instead of **zabbix_agent** is strongly recommended. The **zabbix_agentd** process does not require extra *fork()* and *exec()* calls for every connection, no frequent parsing of configuration file is required as well.

zabbix_trapper

The process provides trapping support for ZABBIX. It constantly waits for connections from **zabbix_sender** agents. It is designed to be used with **inetd** daemon.

If possible, it is recommended that **zabbix_trapperd** is used to provide trapping support. The **zabbix_trapperd** process does not require extra *fork()* and *exec()* calls for every connection.

12.5.3 Command line processes

zabbix_sender

The process, when executed, sends information to ZABBIX trapper (**inetd** or standalone).

zabbix_sender <zabbix_server> <port> <host:key>

zabbix_server Name or IP address of ZABBIX server to connect.

port Port number to connect to ZABBIX. (Port number of **zabbix_trapper** process).

host:key Host name and key for value.

value Value for parameter “host:key”. Positive float number or a character string.

EXAMPLE:

zabbix_sender oracle.company.com 10001 zabbix.company.com:procload 2.34

In this example, **zabbix_sender** will send information about processor load on host *oracle.company.com* to ZABBIX server located at *zabbix.company.com*. Value of the processor load is 2.34.

If no parameters are given, **zabbix_sender** expects list of values from standard input i.e. stdin. Parameters should be given in the following format:

zabbix_server1 port1 host1:key1 value1

zabbix_server2 port2 host2:key2 value2

....

zabbix_serverN portN hostN:keyN valueN

In this case, **zabbix_sender** will be executed only once, therefore no unnecessary forks will be required.

12.6 Extending ZABBIX agents

Using configuration parameter *UserParameter* can easily extend ZABBIX agents. For example, number of emails waiting for delivery may be defined as:

UserParameter: mailq,echo /var/spool/mail/ | wc -w*

See *misc/conf/zabbix_agentd.conf* for already predefined (but commented) parameters.

12.7 Monitoring of MS Windows servers

MS Windows based servers can be monitored by:
using native ZABBIX agent for WIN32 platforms (Windows NT4.0, Windows 2000, Windows XP)
using SNMP
using zabbix_sender compiled under CygWin([footnote][http://www.cygwin.com])

13 ZABBIX Configuration

13.1 Configuration Methods and Execution Modes

Depending on the configuration method and execution mode that is being utilised (polling vs. listening and **inetd** vs. standalone respectively) the ZABBIX server and ZABBIX client have several different processes and names.

13.1.1 Polling mode server

Server must be standalone mode, client can be either standalone or **inetd** mode.

Standalone:

server - **zabbix_suckerd**
client - **zabbix_agentd**

Inetd:

client - **zabbix_agent**

13.1.2 Listen mode serverr

Server can be standalone or **inetd** mode, client must be standalone command line.

Standalone:

server - **zabbix_trapperd**
client - **zabbix_sender** (command line)

Inetd:

server - **zabbix_trapper**
client - **zabbix_sender** (command line)

13.2 Configuration files

All ZABBIX processes can be configured by changing appropriate configuration files. After a change was made, restart of the process required.

All ZABBIX configuration files are stored in directory */etc/zabbix/*.

/etc/zabbix/zabbix_suckerd.conf

The file contains configuration parameters for **zabbix_suckerd**. The file must exist and it should have read permissions for user *zabbix*. Supported parameters:

Parameter	**Mandatory**	**Default value**	**Description**
HousekeepingFrequency	No	1	The parameter defines how often the daemon must perform housekeeping procedure (in hours). If **PostgreSQL** is used set the value to 24 as it will perform command *VACUUM*.
StartSuckers	No	5	Number of **zabbix_suckerd** to

			start (4-255)
SenderFrequency	No	30	The parameter defines how often the daemon must try to send alerts (in seconds)
Timeout	No	5	Do not spend more than *Timeout* seconds on retrieving requested value (1-255)
DisableHousekeeping	No	0	If set to 1, housekeeping procedure will be disabled
DebugLevel	No	3	Debug level (0 - none, 1 - critical, 2 - error, 3 - warnings, 4 - debug)
PidFile	No	*/tmp/zabbix_suckerd.pid*	Name of file to store PID.
LogFile	No	-	Name of log file. If not set, **syslog** is used.
NoTimeWait	No	-	Experimental parameter. If set, no sockets in TIME_WAIT state will exist. Works on Free BSD.
DBUser	No	NULL	User name for connecting to the database
DBHost	No	localhost	DB host name.
DBSocket	No	-	DB socket name. Used for non-TCP connection to **MySQL** database. Example: */tmp/mysql.sock*
DBPassword	No	NULL	User’s password. If password is not used, then this parameter must be commented.
DBName	Yes	-	Database name. Usually “*zabbix*”.

Note: Example of the configuration file can be found at *misc/conf/zabbix_suckerd.conf*

/etc/zabbix/zabbix_agentd.conf

The file contains configuration parameters for **zabbix_agentd**. The file must exist and it should have read permissions for user *zabbix*. Supported parameters:

Parameter	**Mandatory**	**Default value**	**Description**
Server	Yes	-	Comma-delimited list of IP addresses of ZABBIX servers. Connections from other IP addresses will be rejected.
ListenPort	No	10000	Port number to listen

StartAgents	No	5	Number of agents to start
DebugLevel	No	3	Debug level (0 - none, 1 - critical, 2 - error, 3 - warnings, 4 - debug)
PidFile	No	*/tmp/zabbix_suckerd.pid*	Name of file to store PID.
LogFile	No	-	Name of log file. If not set, **syslog** is used.
NoTimeWait	No	-	Experimental parameter. If set, no sockets in TIME_WAIT state will exist. Works on FreeBSD.
DBUser	No	NULL	User name for connecting to the database
DBHost	No	localhost	DB host name.
UserParameter	No	-	User-defined parameter to monitor. There can be several user-defined parameters. Value has form <key>,<shell command> Example: *UserParameter=users,who |wc -l*

Note: Example of the configuration file can be found at *misc/conf/zabbix_agentd.conf*

/etc/zabbix/zabbix_agent.conf

The file contains configuration parameters for **zabbix_agent**. The file must exist and it should have read permissions for user *zabbix*. Supported parameters:

Parameter	**Mandatory**	**Default value**	**Description**
Server	Yes	-	Comma-delimited list of IP addresses of ZABBIX servers. Connections from other IP addresses will be rejected.
Timeout	No	3	Do not spend more that Timeout seconds on getting requested value (1-255).
UserParameter	No	-	User-defined parameter to monitor. There can be several user-defined parameters. Value has form <key>,<shell command> Example: *UserParameter=users,who |wc -l*

Note: Example of the configuration file can be found at *misc/conf/zabbix_agent.conf*

/etc/zabbix/zabbix_trapper.conf

The file contains configuration parameters for **zabbix_trapper**. The file must exist and it should have read permissions for user *zabbix*.
Supported parameters:

Parameter	**Mandatory**	**Default value**	**Description**
Timeout	No	5	Do not spend more than *Timeout* seconds on retrieving requested value (1-255)
DebugLevel	No	3	Debug level (0 - none, 1 - critical, 2 - error, 3 - warnings, 4 - debug)
PidFile	No	*/tmp/zabbix_suckerd.pid*	Name of file to store PID.
LogFile	No	-	Name of log file. If not set, **syslog** is used.
NoTimeWait	No	-	Experimental parameter. If set, no sockets in TIME_WAIT state will exist. Works on FreeBSD.
DBUser	No	NULL	User name for connecting to the database
DBHost	No	localhost	DB host name.
DBSocket	No	-	DB socket name. Used for non-TCP connection to **MySQL** database. Example: */tmp/mysql.sock*
DBPassword	No	NULL	User's password. If password is not used, then this parameter must be commented.
DBName	Yes	-	Database name. Usually "*zabbix*".

Note: Example of the configuration file can be found at *misc/conf/zabbix_trapper.conf*

/etc/zabbix/zabbix_trapperd.conf

The file contains configuration parameters for **zabbix_trapperd**. The file must exist and it should have read permissions for user *zabbix*.
Supported parameters:

Parameter	**Mandatory**	**Default value**	**Description**
Timeout	No	5	Do not spend more than *Timeout* seconds on retrieving requested value (1-255)
StartTrappers	No	5	
ListenPort	No	10001	
DebugLevel	No	3	Debug level (0 - none, 1 - critical, 2 - error, 3 - warnings, 4 - debug)
PidFile	No	*/tmp/zabbix_suckerd.pid*	Name of file to store PID.
LogFile	No	-	Name of log file. If not set, **syslog** is used.
NoTimeWait	No	-	Experimental parameter. If set, no sockets in TIME_WAIT state will exist. Works on FreeBSD.
DBUser	No	NULL	User name for connecting to the database
DBHost	No	localhost	DB host name.
DBSocket	No	-	DB socket name. Used for non-TCP connection to **MySQL** database. Example: */tmp/mysql.sock*
DBPassword	No	NULL	User's password. If password is not used, then this parameter must be commented.
DBName	Yes	-	Database name. Usually "*zabbix*".
DBConnectOnEach	No	0	If 1, **zabbix_trapperd** will connect to the database on each value received.

Note: Example of the configuration file can be found at *misc/conf/zabbix_trapperd.conf*

13.3 Expression for triggers

The expressions used in triggers are very flexible. You can use them to create complex logical tests regarding monitored statistics.

The following operators are supported for triggers:

Operator	**Definition**
-	Arithmetical minus
+	Arithmetical plus

/	Division
*	Multiplication
>	More than
<	Less than
=	Is equal. The operator is defined as A=B <=> (A>B-0.000001) and (A<B+0.000001)
|	Logical OR
&	Logical AND

The following functions are supported:

Function	**Definition**
last	Last (most recent) value. Parameter is ignored.
min	Minimal value for period of time. Parameter defines length of the period in seconds.
max	Maximal value for period of time. Parameter defines length of the period in seconds.
diff	Returns: 1 – if last and previous values differs 0 – otherwise
prev	Returns previous value. Parameter is ignored.
str	Find string in last (most recent) value. Parameter defines string to find. Returns: 1 – found 0 – otherwise Can be used for items having type STRING only.
nodata	Returns: 1 – if no data received during period of time. Parameter defines length of the period in seconds. Can be used for item type TRAPPER only.

Note: Note that all above functions (except **diff** and **str**) cannot be used for non-numeric parameters!

A simple useful expression might look like:

{<server>:<key>.<function>(<parameter>)}<operator><const>

Example 1. Processor load is too high on zabbix.sf.net.

{zabbix.sf.net:system[procload].last(0)}>5

'*zabbix.sf.net:system[procload]*' gives a short name of monitored parameter. It specifies that the server is 'sourceforge.net' and the key being monitored is '*system[procload]*'. By using the function '*last()*', we are specifying we are referring to the most recent value. Finally, '*>5*' means that the trigger is true whenever the most recent processor load measurement from *zabbix.sf.net* is greater than 5.

Example 2. zabbix.sf.net is overloaded.

({zabbix.sf.net:system[procload].last(0)}>5)|*({zabbix.sf.net:system[procload].min(600)}>2)*

The expression is true when either the current processor load is more than 5 or the processor load was more than 2 during last 10 minutes.

Example 3. /etc/passwd has been changed.

Use of *diff* function:

({zabbix.sf.net:cksum[/etc/passwd].diff(0)})>0

The expression is true when the previous value of checksum of */etc/passwd* differs from the most recent one.

Similar expressions could be useful to monitor changes in important files, such as */etc/passwd*, */etc/inetd.conf*, */kernel*, etc.

13.4 Monitored Parameters

13.4.1 Types of parameters

13.4.2 Flexible and non-flexible parameters

Flexible parameter is parameter which accepts argument. For example, *diskfree[*]* is flexible parameter. '*' is any string that will be passed as argument of the parameter. *diskfree[/]*, *diskfree[/opt]* - correct definitions.

13.4.3 List of supported parameters

ALL PLATFORMS

check_port[<ip,>port]

Check, if it is possible to make TCP connection to port number "*". If IP address is missing, 127.0.0.1 is used. IP address is optional.

Result: 0 - can connect

1 - cannot connect

Example: *Use check_port[80] to check either WEB server accepts connections.*

check_service[ftp<,ip><,port>] Check if FTP server is running and accepting connections. If port number is missing, '21' is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - FTP server is down

1 - FTP server is running

check_service[http<,ip><,port>] Check if HTTP server is running and accepting connections. If port number is missing, '80' is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - HTTP server is down

1 - HTTP server is running

check_service[imap<,ip><,port>] Check if IMAP server is running and accepting connections. If port number is missing, '143' is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - IMAP server is down

1 - IMAP server is running

check_service[nntp<,ip><,port>] Check if NNTP server is running and accepting connections. If port number is missing, '119' is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - NNTP server is down

1 - NNTP server is running

check_service[pop<,ip><,port>] Check if POP server is running and accepting connections. If port number is missing, ‘110’ is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - POP server is down

1 - POP server is running

check_service[smtp<,ip><,port>] Check if Email (SMTP) server is running and accepting connections. If port number is missing, ‘25’ is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - SMTP server is down

1 - SMTP server is running

check_service[ssh<,ip><,port>] Check if SSH server is running and accepting connections. If port number is missing, ‘22’ is used. If IP address is missing, 127.0.0.1 is used. Both IP address and port number are optional.

Result: 0 - SSH server is down

1 - SSH server is running

cksum[*] Check sum of a given file.

Example: *cksum[/etc/passwd]*

diskfree[*] Amount of unused disk space for given volume (in Kb). In case of mounted volume, unused disk space for local file system is returned.

Example: *diskfree[/] - returns free disk space for root volume.*

disktotal[*] Amount of total disk space for given volume (in Kb). In case of mounted volume, total disk space for local file system is returned.

Example: *disktotal[/] - returns total disk space for root volume.*

filesize[*] Size of given file.

Example: *filesize[/var/log/syslog]*

inodefree[*] Number of unused inodes space for given volume.

Example: *inodefree[/var] - returns number of free inodes for volume /var*

inodetotal[*] Total number of inodes for given volume.

Example: *inodetotal[/var] - returns number of inodes for volume /var*

kern[maxfiles] Maximum number of opened files supported by OS.

kern[maxproc] Maximum number of processes supported by OS.

memory[buffers] Amount of memory used for kernel buffers.

memory[cached] Amount of cached memory.

memory[free] Amount of unused physical memory.

memory[shared] Amount of used shared memory.

memory[total] Total amount of installed physical memory.

netloadin1[*] Average number of bytes received on interface '*'. Average for 1 minute.

Example: *netloadin1[eth0]*

netloadin5[*] Average number of bytes received on interface '*'. Average for last 5 minutes.

netloadin15[*] Average number of bytes received on interface '*'. Average for last 15 minutes.

netloadout1[*] Average number of bytes sent on interface '*'. Average for 1 minute.

Example: *netloadout1[eth1]*

netloadout5[*] Average number of bytes sent on interface '*'. Average for 5 minute.

netloadout15[*] Average number of bytes sent on interface '*'. Average for 15 minute.

ping Always return 1.

proc_cnt[*] Number of processes called "*" running.

Example: *proc_cnt[inetd]*

status Host status (0 - normal, 2 - unreachable). This parameter is calculated internally by **zabbix_suckerd**. No connection to an agent required.

swap[free] Free swap space.

system[hostname] Returns host name.

Result: string value.

Example of returned value: *www.sf.net*

swap[total] Total swap space.

system[proccount] Number of processes.

system[procload] Load average for last 1 minute.

system[procload5] Load average for last 5 minutes.

system[procload15] Load average for last 15 minutes.

system[procrunning] Number of running processes.

system[uname] Returns detailed host information.

Result:string value.

Example of returned value: *FreeBSD localhost 4.4-RELEASE FreeBSD 4.4-RELEASE #0: Tue Sep 18 11:57:08 PDT 2001 murray@builder.FreeBSD.org:/usr/src/sys/compile/GENERIC i386*

system[uptime] System's uptime in seconds.

system[users] Number of users connected.

tcp_count Number of established TCP connections.

version[zabbix_agent] Returns version of **zabbix_agent**(**d**) running on the server

Result:string value.

Example of returned value: *1.0beta8*

13.4.4 WIN32-SPECIFIC PARAMETERS

cpu_util Average CPU(s) utilization (in percents) for last minute

cpu_util5 Average CPU(s) utilization (in percents) for last 5 minutes

cpu_util15 Average CPU(s) utilization (in percents) for last 15 minutes

cpu_util[*] Average specific CPU utilization (in percents) for last minute, where parameter is zero-based CPU number

cpu_util5[*] Average specific CPU utilization (in percents) for last 5 minutes, where parameter is zero-based CPU number

cpu_util15[*] Average specific CPU utilization (in percents) for last 15 minutes, where parameter is zero-based CPU number

md5_hash[*] MD5 hash of specified file (returned as string).

Example: cksum[c:\autoexec.bat]

perf_counter[*] Value of any performance counter, where <path> is the counter path (you can use Performance Monitor to obtain list of available counters). Note that this parameter will return correct value only for counters which requires just one sample (like "\System\Threads"). It will not work as expected for counters that requires more than one sample - like CPU utilization.

Example: perf_counter[\System\Threads]

service_state[*] State of service. The following states can be returned:

0 - Running

1 - Paused

2 - Start pending

3 - Pause pending

4 - Continue pending

5 - Stop pending

6 - Stopped

7 - Unknown

255 - SCM communication error

Note that parameter should be real service name (as it seen in service properties under “Name:”), not service display name!

13.4.5 SNMP parameters

Example:

Community: *public*
Oid: *1.2.3.45.6.7.8.0*
Key: <Unique string to be used as reference to triggers>

Note that OID can be given in either numeric or string form. However, in some cases, string OID must be converted to numeric representation. Utility **snmpget** may be used for this purpose:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

13.5 Supported Parameters by Platform

Parameter/system	**Windows (NT4.0, W2000,XP)**	**Linux (Debian Potato)**	**FreeBSD 4.3 (i386)**	**Solaris 5.8 (Ultra-60)**	**HP-UX 11.00 (9000/800)**	**AIX 4.3 (Power PC)**
memory[total]	X	X	X	-	X	X
memory[shared]	-	X	X	-	-	-

memory[buffers]	-	X	-	-	-	-
memory[cached]	-	X	-	-	-	-
memory[free]	X	X	X	X	-	X
diskfree[*]	X	X	X	X	X	X
disktotal[*]	X	X	X	X	X	X
inodefree[*]	-	X	X	X	X	X
inodetotal[*]	-	X	X	X	X	X
cksum[*]	X	X	X	X	X	X
filesize[*]	-	X	X	X	X	X
swap[free]	X	X	-	X	X	-
swap[total]	X	X	-	X	X	-
swap[in]	-	X	-	-	-	-
swap[out]	-	X	-	-	-	-
system[interrupts]	-	X	-	-	-	-
system[switches]	-	X	-	-	-	-
proc_cnt[*]	X	X	-	-	-	-
check_port[*]	-	X	X	X	X	X
check_service[ftp]	-	X	X	X	X	X
check_service[ssh]	-	X	X	X	X	X
check_service[smtp]	-	X	X	X	X	X
check_service[http]	-	X	X	X	X	X
check_service[pop]	-	X	X	X	X	X
check_service[nntp]	-	X	X	X	X	X
check_service[imap]	-	X	X	X	X	X
io[disk_io]	-	X	-	-	-	-
io[disk_rio]	-	X	-	-	-	-
io[disk_wio]	-	X	-	-	-	-
io[disk_rblk]	-	X	-	-	-	-
io[disk_wblk]	-	X	-	-	-	-
system[procload]	X	X	X	X	X	-
system[procload5]	X	X	X	X	X	-
system[procload15]	X	X	X	X	X	-
system[proccount]	X	X	-	-	-	-
system[procrunning]	-	X	-	-	-	-
system[uptime]	X	X	X	-	-	-
system[users]	-	X	X	X	X	X
version[zabbix_agent]	X	X	X	X	X	X
ping	X	X	X	X	X	X
kern[maxfiles]	-	-	X	-	-	-
kern[maxproc]	-	-	X	-	-	-
netloadin1[*]	-	X	-	-	-	-

netloadin15[*]	-	X	-	-	-	-
netloadin5[*]	-	X	-	-	-	-
netloadout1[*]	-	X	-	-	-	-
netloadout15[*]	-	X	-	-	-	-
netloadout5[*]	-	X	-	-	-	-
tcp_count	-	X	X	-	-	X
cpu_util	X	-	-	-	-	-
cpu_util5	X	-	-	-	-	-
cpu_util15	X	-	-	-	-	-
cpu_util[*]	X	-	-	-	-	-
cpu_util5[*]	X	-	-	-	-	-
cpu_util15[*]	X	-	-	-	-	-
md5_hash[*]	X	-	-	-	-	-
perf_counter[*]	X	-	-	-	-	-
service_state[*]	X	-	-	-	-	-
tcp_count	-	X	X	-	-	X

14 Monitoring of specific applications

14.1 MySQL

Configuration file misc/conf/zabbix_agentd.conf contains
list of parameters that can be used for monitoring of MySQL.

```
### Set of parameter for monitoring MySQL server (v3.23.42 and later)

### Change -u<username> and add -p<password> if required


#UserParameter=mysql[ping],mysqladmin -uroot ping|grep alive|wc -l

#UserParameter=mysql[uptime],mysqladmin -uroot status|cut f2 -d”:”|cut -f1 -d”T”

#UserParameter=mysql[threads],mysqladmin -uroot status|cut f3 -d”:”|cut -f1 -d”Q”

#UserParameter=mysql[questions],mysqladmin -uroot status|cut f4 -d”:”|cut -f1 -d”S”

#UserParameter=mysql[slowqueries],mysqladmin -uroot status|cut f5 -d”:”|cut -f1 -d”O”

#UserParameter=mysql[qps],mysqladmin -uroot status|cut -f9 d”:”

#UserParameter=version[mysql],mysql -V
```

mysql[ping] Check, if MySQL is alive

Result: 0 - not started
1 - alive

mysql[uptime] Number of seconds MySQL is running *mysql[threads] Number of MySQL threads*

mysql[questions] Number of processed queries

mysql[slowqueries] Number of slow queries

mysql[qps] Queries per second

mysql[version] Version of MySQL

Example: mysql Ver 11.16 Distrib 3.23.49, for pc-linux-gnu (i686)

14.2 IMAP servers

14.4 FTP servers

14.6 POP3 servers

14.8 NNTP servers

15 Getting Maximum Performance from ZABBIX

15.1 Real world configuration

Server with ZABBIX 1.0alpha12 installed (Linux 2.2, MySQL, Celeron/300Mhz, 32Mb, IDE) is able to collect more than 40 parameters per second from servers being monitored. How many servers can be monitored by ZABBIX on the hardware, you may ask. It depends on number of monitored parameters and how often ZABBIX should acquire these parameters. Suppose, each server you monitor has ten parameters to watch for. You want to update these parameters once in a 10 seconds. Doing simple calculation, we see that ZABBIX is able to handle 40 servers. In case if these parameters need to be updated once in a minute, the hardware configuration will be able to handle 40x6=240 servers. These calculations made in assumption that all monitored values are retrieved as soon as required (latency is 0). If this is not a requirement, then number of monitored servers can be increased even up to 5x-10x times.

15.2 Performance tuning

Performance of ZABBIX can be significantly improved by tuning:

Hardware

- Use fastest processor available
- SCSI is better than IDE (performance of IDE disks may be significantly improved by using utility **hdparm**)
- Use fast Ethernet adapter
- Having more memory is always better

Operating System

- Use latest (stable!) version of OS
- Exclude unnecessary functionality from kernel
- Tune kernel parameters

ZABBIX configuration parameters

Many parameters may be tuned to get optimal performance.

zabbix_suckerd

StartSuckers General rule - keep value of this parameter as low as possible. Every additional instance of zabbix_suckerd adds known overhead, in the same time, parallelism is increased. Optimal number of instances is achieved when queue, on average, contains minimum number of parameters (ideally, 0 at any given moment).

DebugLevel Optimal value is ‘3’.

DBSocket **MySQL** only. It is recommended to use *DBSocket* for connection to the database. That is fastest and most secure way.

zabbix_trapperd

DBConnectOnEach Should not be set if you want to get maximum performance.

Database Engine

- use fastest database engine, i.e. **MySQL**

- use stable release of a database engine

- rebuild **MySQL** or **PostgreSQL** from sources to get maximum performance

- follow performance tuning instructions taken from **MySQL** or **PostgreSQL** documentation

- for **MySQL**, use InnoDB table structure

Note: ZABBIX works at least 1.5 times (comparing to MyISAM) if InnoDB is used

16 ZABBIX Frontends

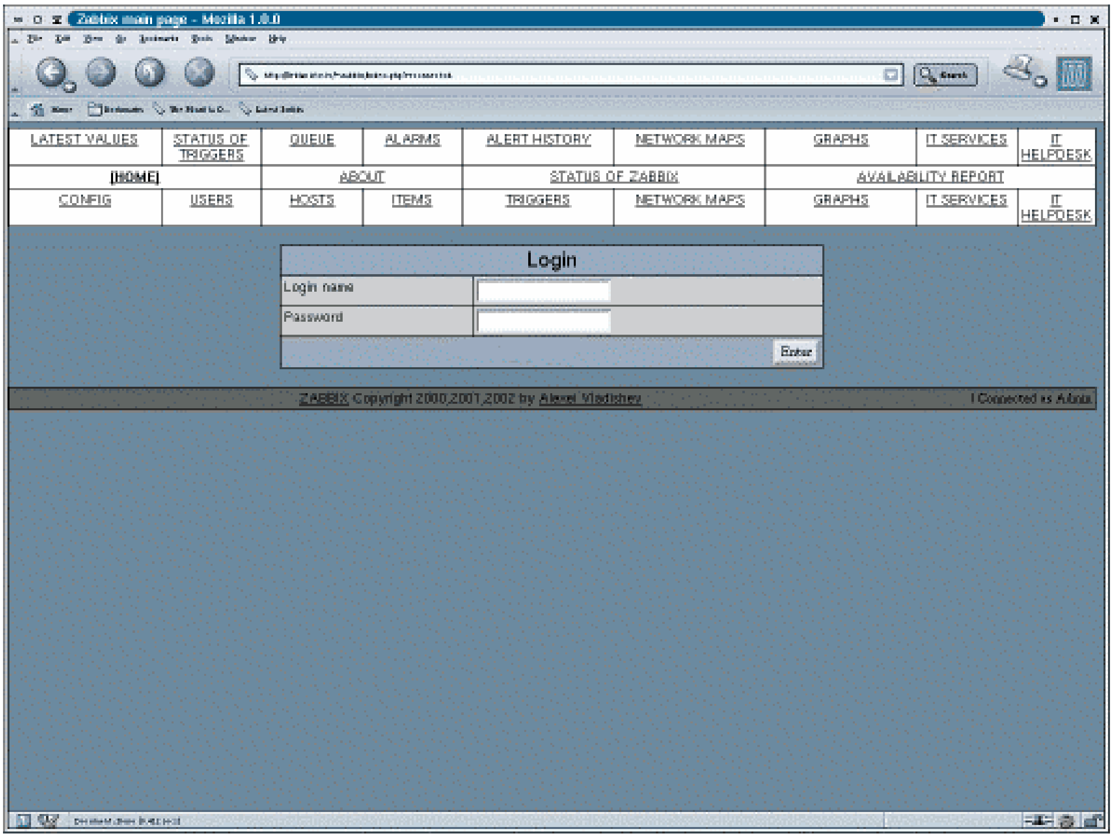
Currently, the only frontend available is WEB frontend written in **PHP**.

16.1 PHP-based Frontend

The frontend provides a convenient and platform-independent method for accessing ZABBIX. Information provided by the frontend can be either graphical or textual. While graphical representations usually provide the easiest method to understand trends, the text representation of monitored parameter is intended to provide an easy way to export ZABBIX data to other analytical tools.

16.1.1 Menu

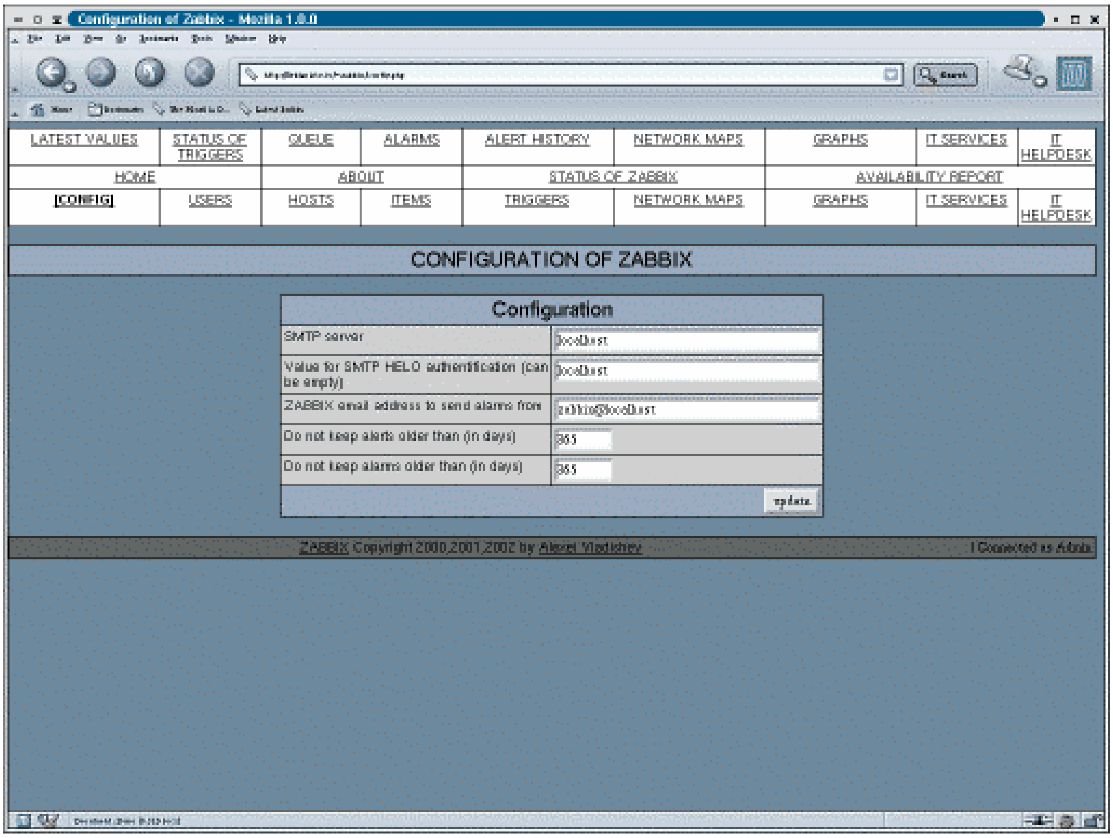
Most of pages contain menu that visibly consists of three rows. First row contains end-user functionality. Second and third row is area for ZABBIX administration.



Main menu -> Home

CONFIG

The menu item is used to set global ZABBIX parameters.



CONFIGURATION OF ZABBIX

Configuration	
SMTP server	localhost
Value for SMTP HELO authentification (can be empty)	localhost
ZABBIX email address to send alarms from	zabbix@localhost
Do not keep alerts older than (in days)	365
Do not keep alarms older than (in days)	365

ZABBIX Copyright 2000,2001,2002 by Alexei Vladishev

Main menu -> CONFIG

SMTP server

IP address or name of SMTP server. ZABBIX sends alert messages via this server.

Value for SMTP HELO authentication

Usually set to domain name. ZABBIX will not use HELO authentication.

Example: *sourceforge.net*

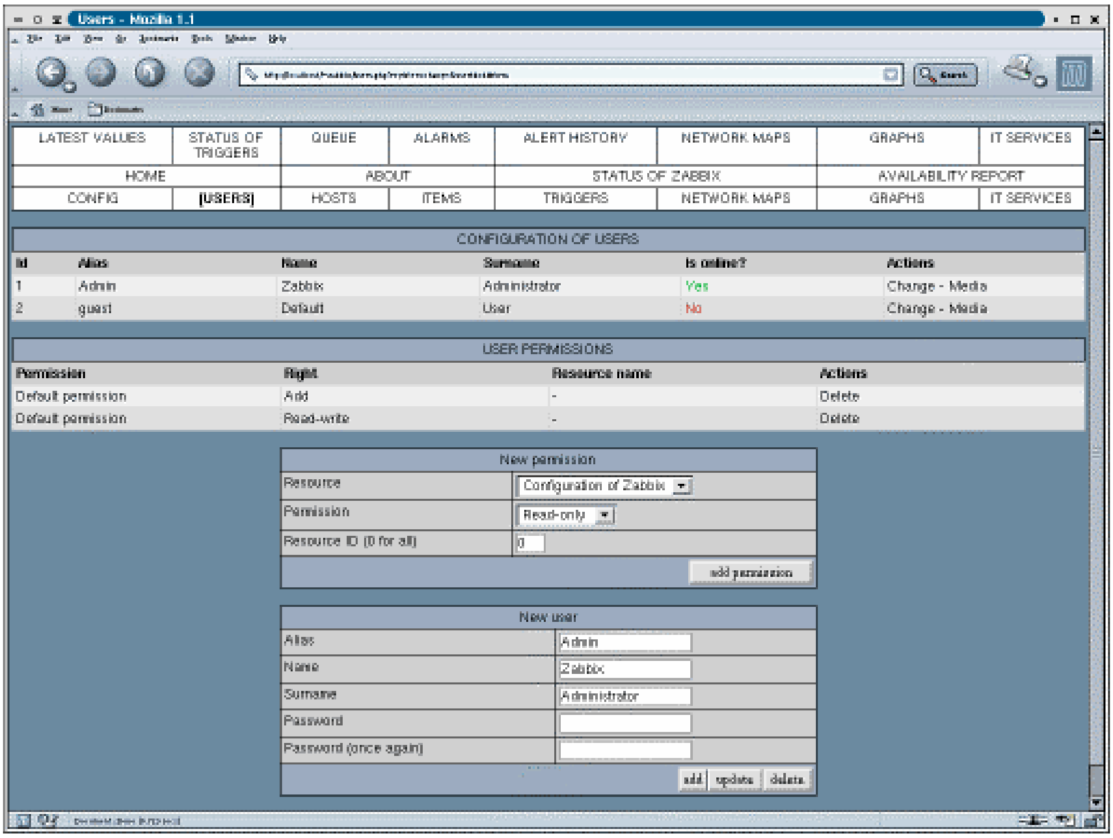
ZABBIX email address to send alerts from

ZABBIX uses this address as source address for all emails.

Example: *zabbix@zabbix.company.com*

USERS

The menu item is used to configure ZABBIX users and theirs access rights.



CONFIGURATION OF USERS

Id	Alias	Name	Surname	Is online?	Actions
1	Admin	Zabbix	Administrator	Yes	Change - Media
2	guest	Default	User	No	Change - Media

USER PERMISSIONS

Permission	Right	Resource name	Actions
Default permission	Add	-	Delete
Default permission	Read-write	-	Delete

New permission

Resource	Configuration of Zabbix
Permission	Read-only
Resource ID (0 for all)	0

New user

Alias	Admin
Name	Zabbix
Surname	Administrator
Password	
Password (once again)	

Main menu -> USERS

USER DETAILS

Alias

Short name of the user.

Example: *alex*

Name

Name of the user.

Example: *Alexei*

Surname

Surname of the user.

Example: *Vladishev*

Password

Password. The field can be left blank.

Example: ********

PERMISSIONS

Resource

ZABBIX resource to apply selected permission.

Example: *“Configuration of Zabbix”*

Permission

It specifies permission for the selected resource.

Example: *Read-Only*

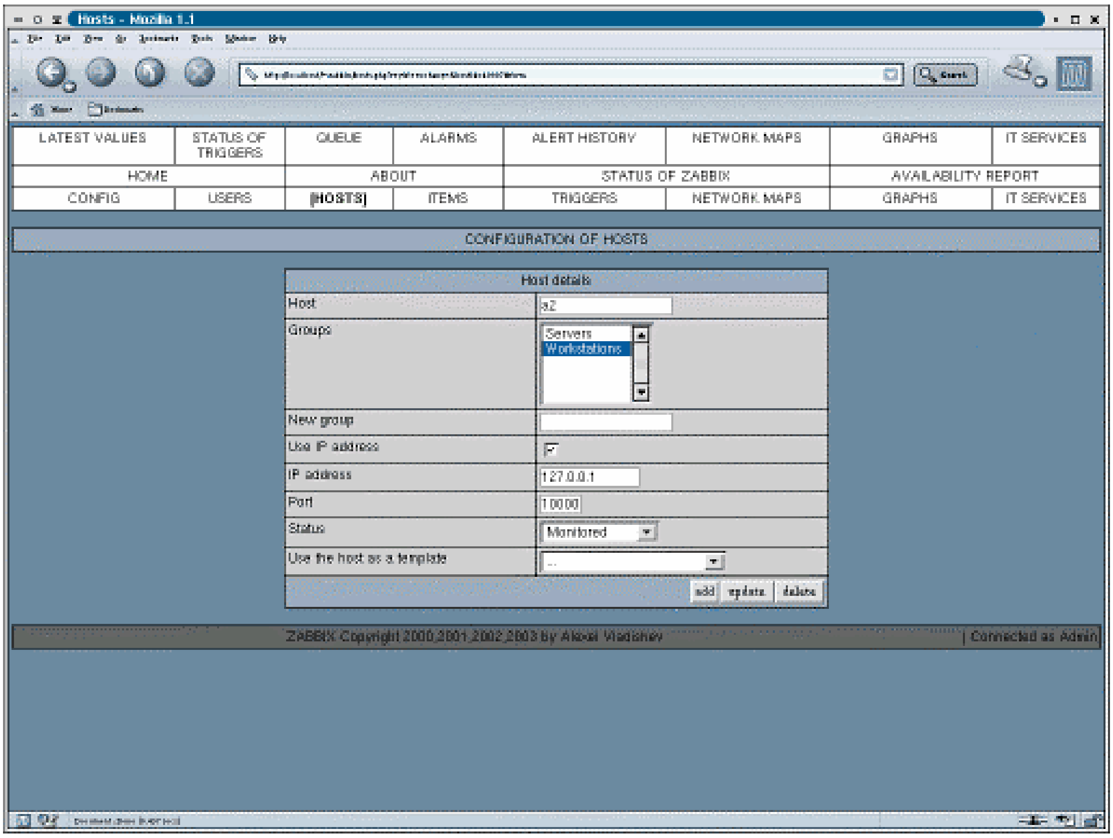
Resource ID (0 for all)

ID of the selected resources. If 0 is specified, permission will be applied to all resources of given type.

Example: *1234*

HOSTS

The menu item is used to configure hosts for monitoring.



Host details

Host	a2
Groups	Servers Workstations
New group	
Use IP address	☑
IP address	127.0.0.1
Port	10000
Status	Monitored
Use the host as a template	...

Main menu -> HOSTS

Host

Host name. Full or short DNS name of the host is usually used.

Example: *zabbix.sourceforge.net*

Groups

Host groups. Select zero or several host groups.

New group

New host group.

Example: *Database servers*

Use IP address

Select this checkbox if you want to connect to the host using IP address.

IP address

Host IP address.
`

Example: *127.0.0.1*

Port

This parameter specifies port number of ZABBIX agent on this host.

Example: *10000*

Status

Host status. Possible values:

- *monitored*. ZABBIX keeps this host under control.
- *not monitored*. ZABBIX ignores all events related to the host.

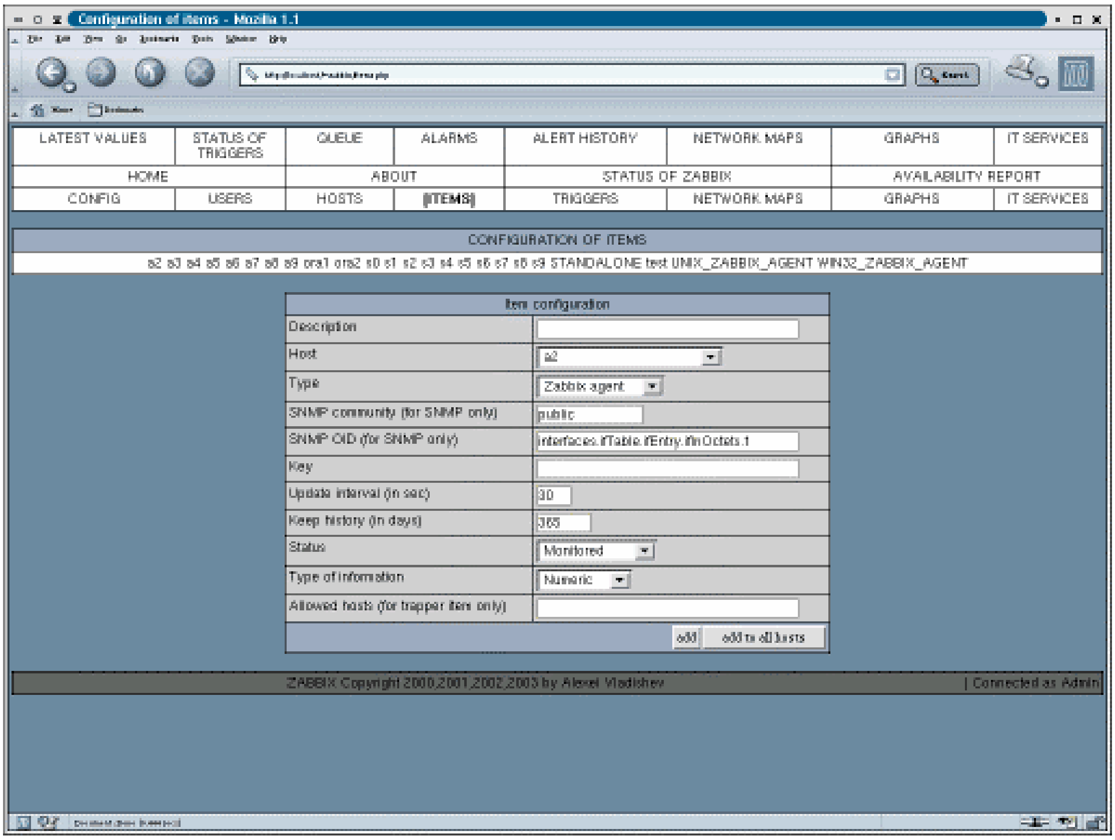
Example: *Monitored*

Use the host as a template

Add items and triggers of selected template host. This will greatly simplify addition of hosts having similar configuration.

ITEMS

The menu item is used to configure parameters for monitoring.



Item configuration

Description	
Host	s2
Type	Zabbix agent
SNMP community (for SNMP only)	public
SNMP OID (for SNMP only)	interfaces.ifTable.ifEntry.ifInOctets.1
Key	
Update interval (in sec)	30
Keep history (in days)	365
Status	Monitored
Type of information	Numeric
Allowed hosts (for trapper item only)	

Description

Description of parameter for monitoring.

Note: Macro %s will be substituted by host name. This is required to make possible using of the item for template host.

Example: *Processor load on host %s*

Host

Host name this item belongs to.

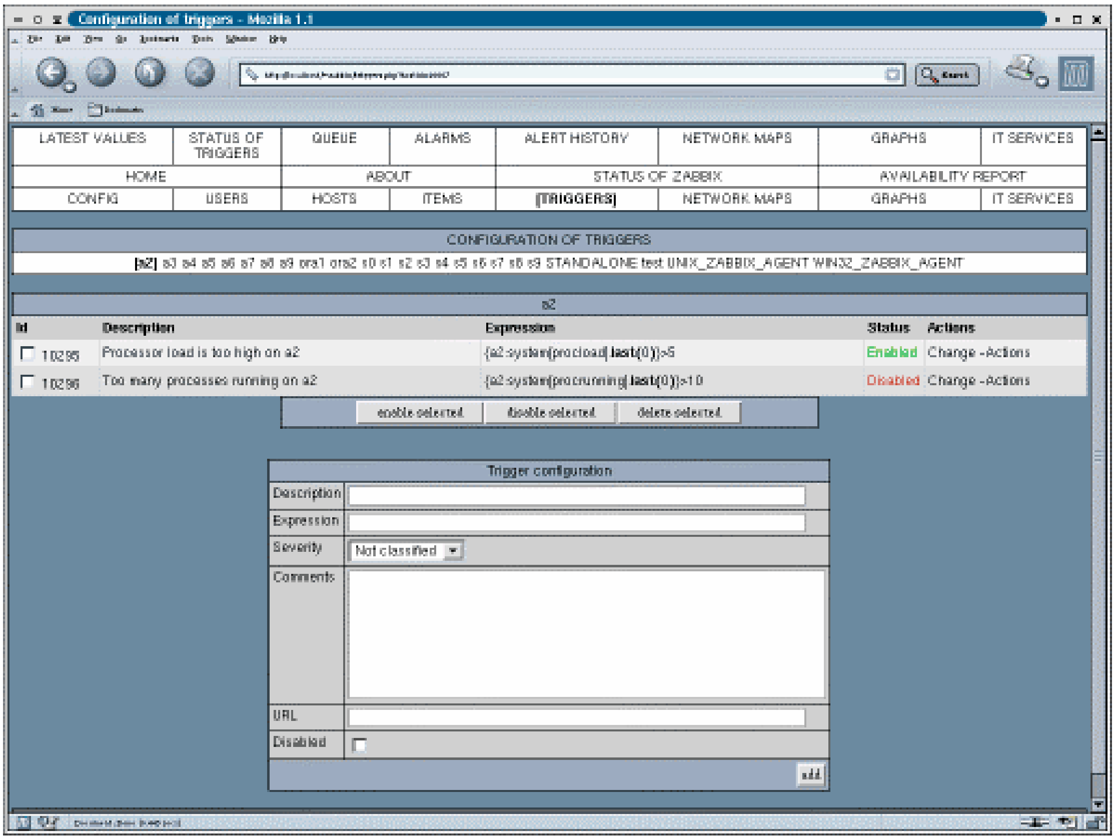
Type

Item type. Possible value:

Zabbix agent. ZABBIX will use ZABBIX agent running on the host.
<UNFINISHED>

TRIGGERS

The menu item is used to configure triggers.



a2

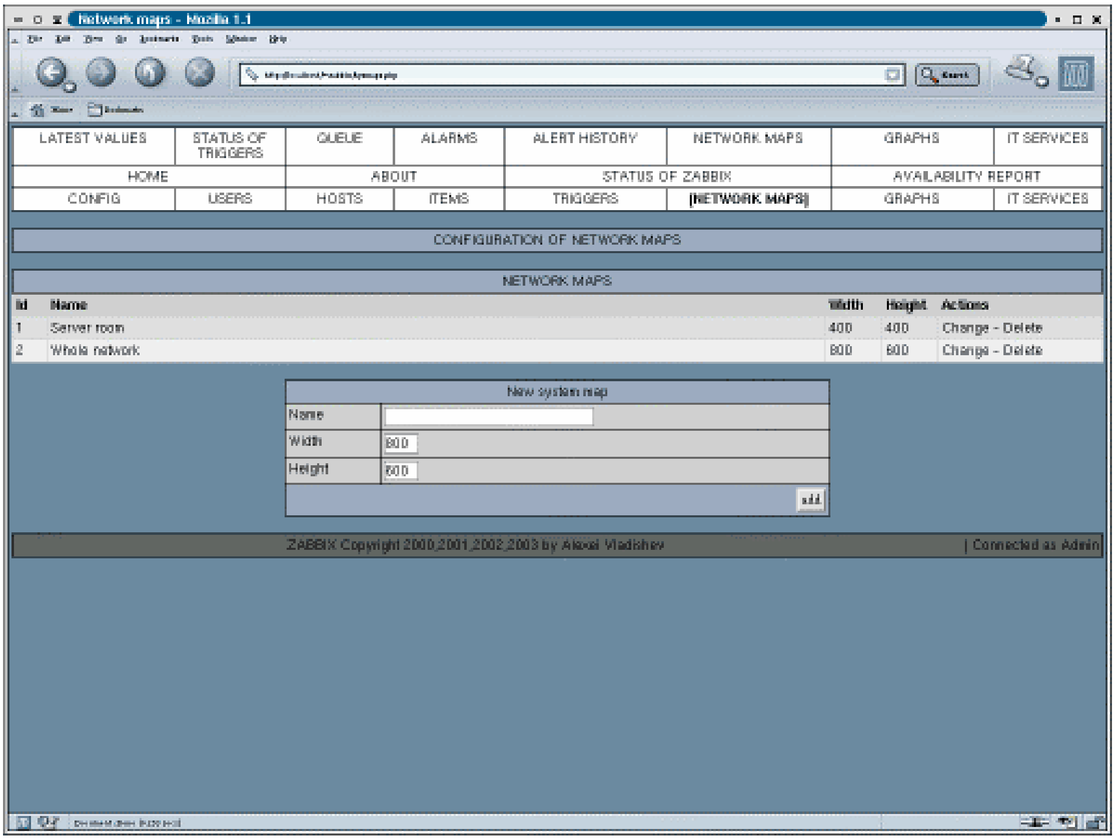
Id	Description	Expression	Status	Actions
☐ 10295	Processor load is too high on a2	{a2:system[procload].last(0)}>5	Enabled	Change - Actions
☐ 10296	Too many processes running on a2	{a2:system[procrunning].last(0)}>10	Disabled	Change - Actions

Trigger configuration

Description	
Expression	
Severity	Not classified
Comments	
URL	
Disabled	☐

NETWORK MAPS

The menu item is used to configure network maps.



NETWORK MAPS

Id	Name	Width	Height	Actions
1	Server room	400	400	Change - Delete
2	Whole network	800	600	Change - Delete

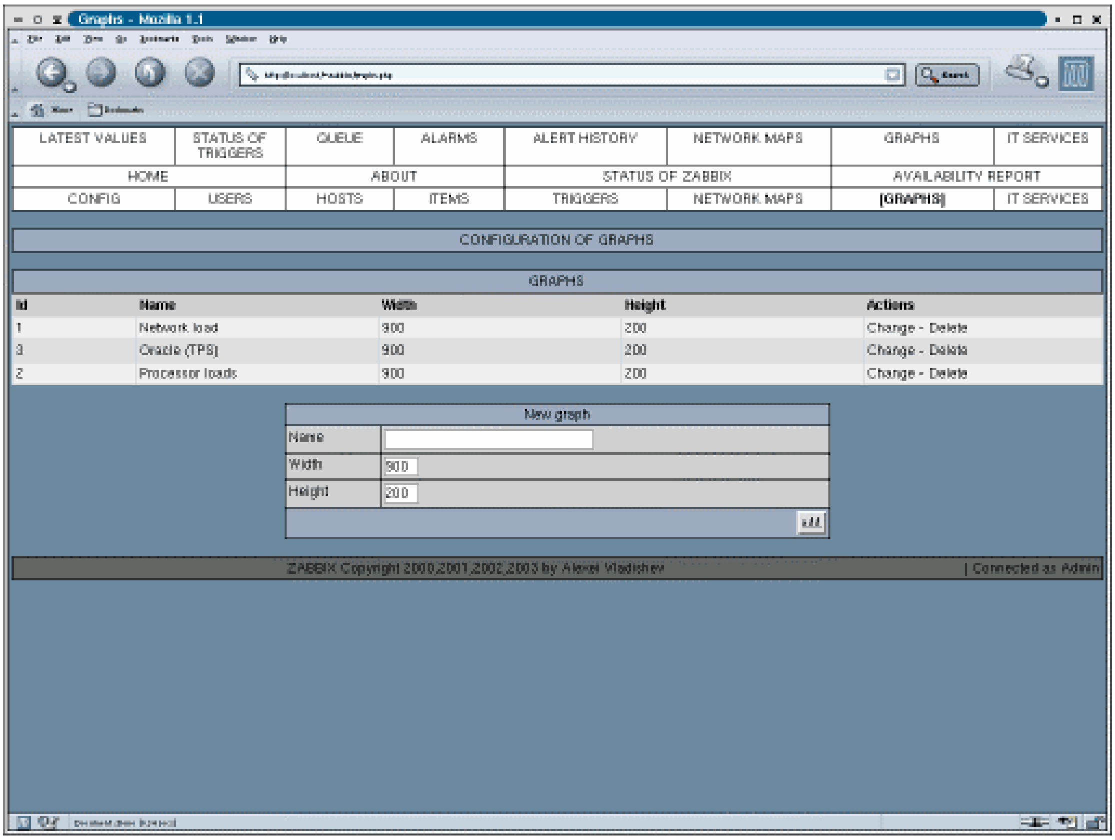
New system map

Name	
Width	800
Height	600

ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev | Connected as Admin

GRAPHS

The menu item is used to configure user-defined (complex) graphs.



CONFIGURATION OF GRAPHS

GRAPHS

Id	Name	Width	Height	Actions
1	Network load	900	200	Change - Delete
3	Oracle (TPS)	900	200	Change - Delete
2	Processor loads	900	200	Change - Delete

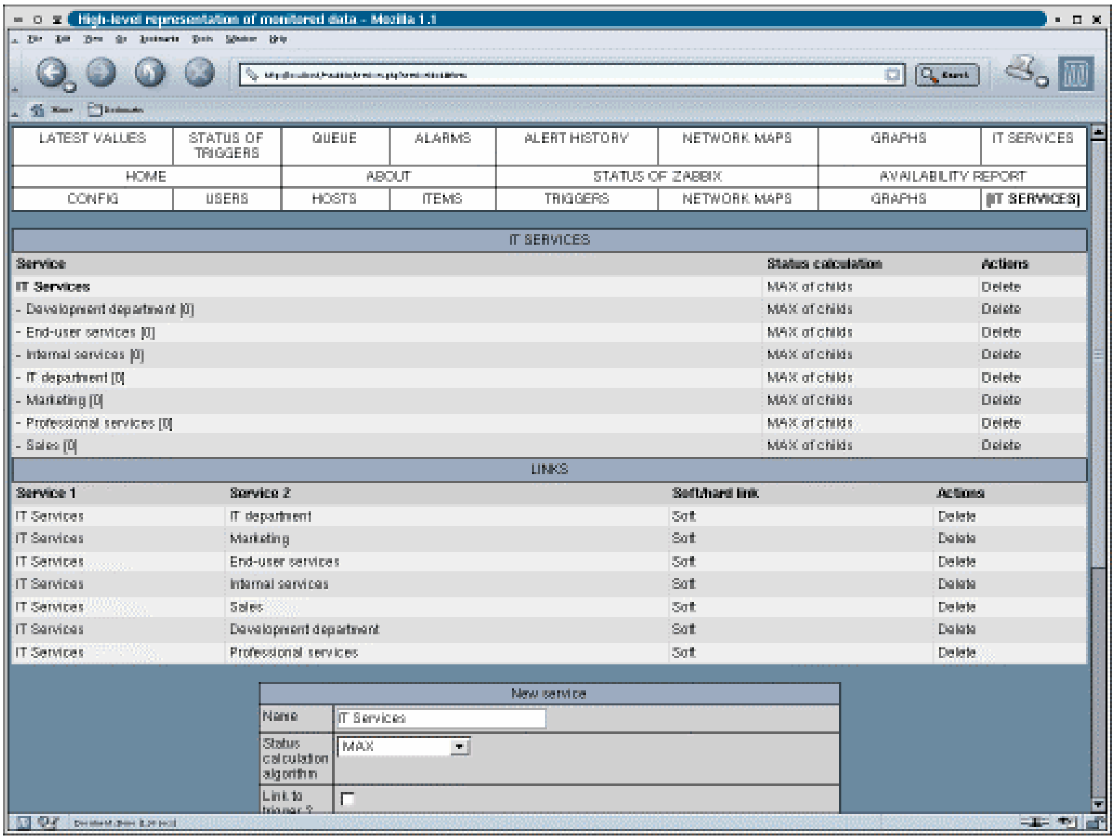
New graph

Name	
Width	900
Height	200

ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev | Connected as Admin

IT SERVICES

The menu item is used to configure structure of high-level IT service tree.



IT SERVICES

Service	Status calculation	Actions
IT Services	MAX of childs	Delete
- Development department [0]	MAX of childs	Delete
- End-user services [0]	MAX of childs	Delete
- Internal services [0]	MAX of childs	Delete
- IT department [0]	MAX of childs	Delete
- Marketing [0]	MAX of childs	Delete
- Professional services [0]	MAX of childs	Delete
- Sales [0]	MAX of childs	Delete

LINKS

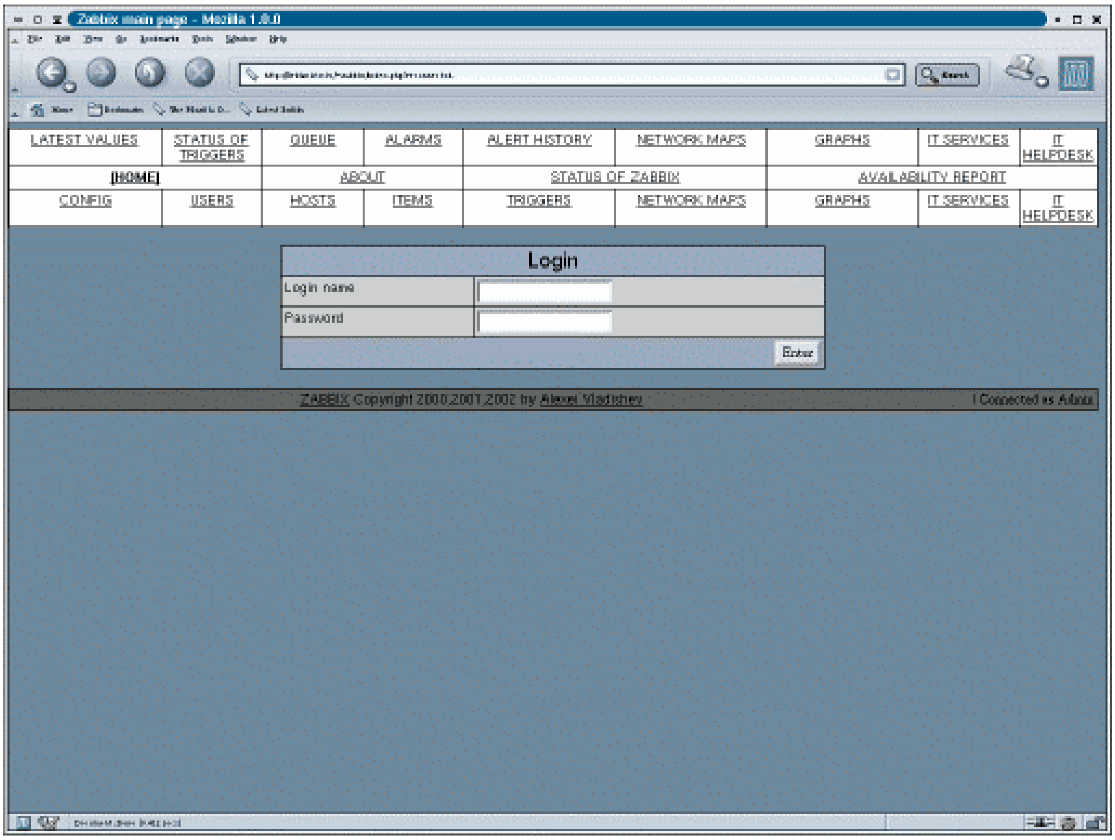
Service 1	Service 2	Soft/hard link	Actions
IT Services	IT department	Soft	Delete
IT Services	Marketing	Soft	Delete
IT Services	End-user services	Soft	Delete
IT Services	Internal services	Soft	Delete
IT Services	Sales	Soft	Delete
IT Services	Development department	Soft	Delete
IT Services	Professional services	Soft	Delete

New service

Name	IT Services
Status calculation algorithm	MAX
Link to trigger ?	☐

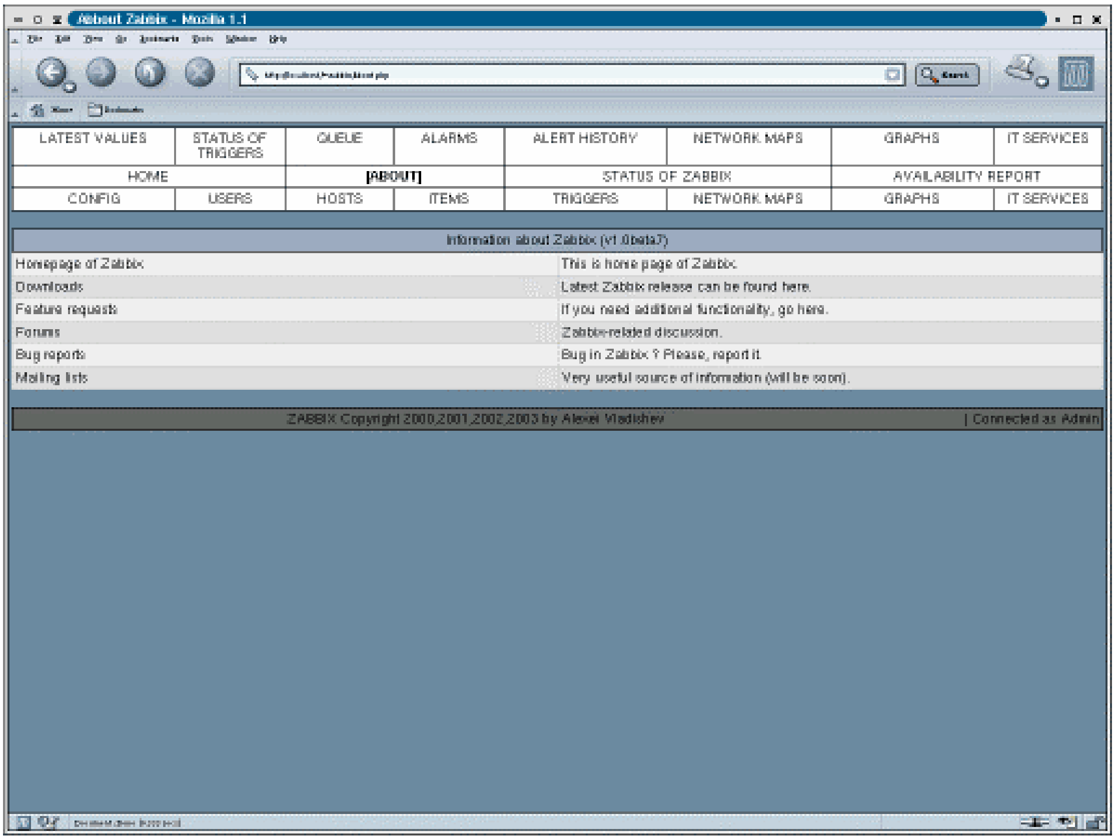
HOME

This is login screen.



ABOUT

This screen provides pointers to ZABBIX related information.

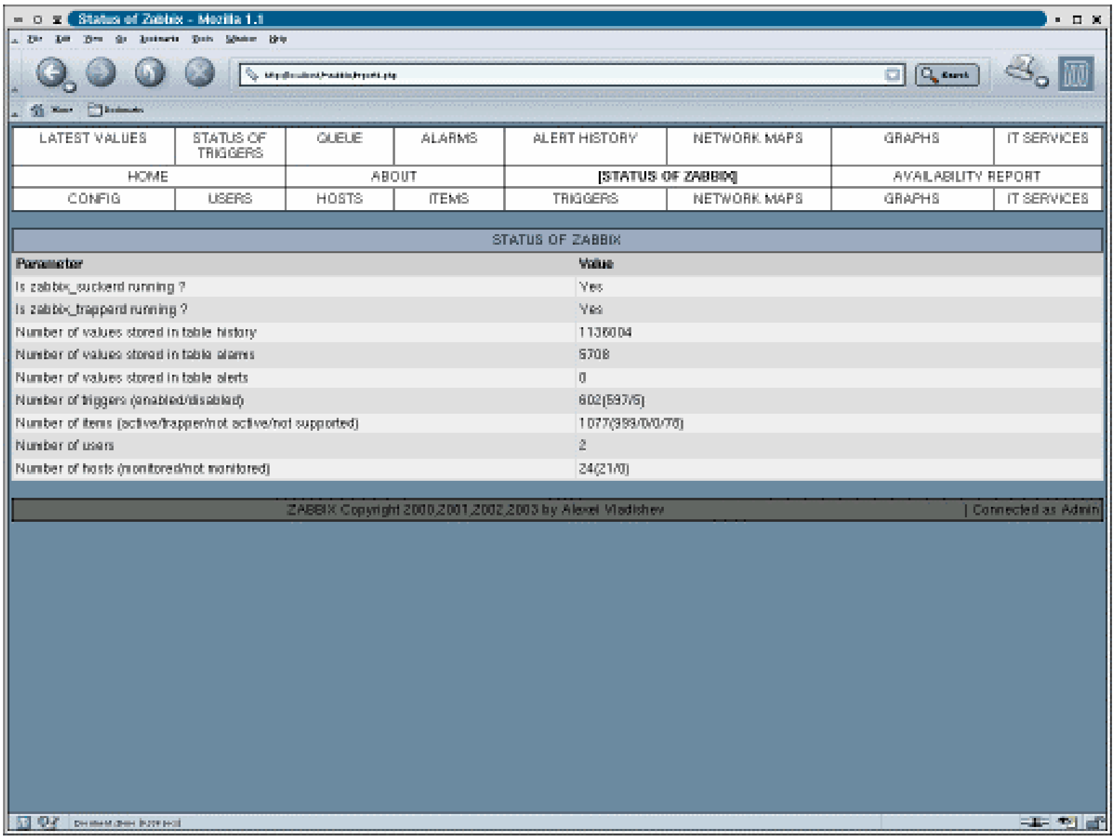


Information about Zabbix (v1.0beta7)	
Homepage of Zabbix	This is home page of Zabbix.
Downloads	Latest Zabbix release can be found here.
Feature requests	If you need additional functionality, go here.
Forums	Zabbix-related discussion.
Bug reports	Bug in Zabbix ? Please, report it.
Mailing lists	Very useful source of information (will be soon).

ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev | Connected as Admin

STATUS OF ZABBIX

This report gives general information about the ZABBIX server.

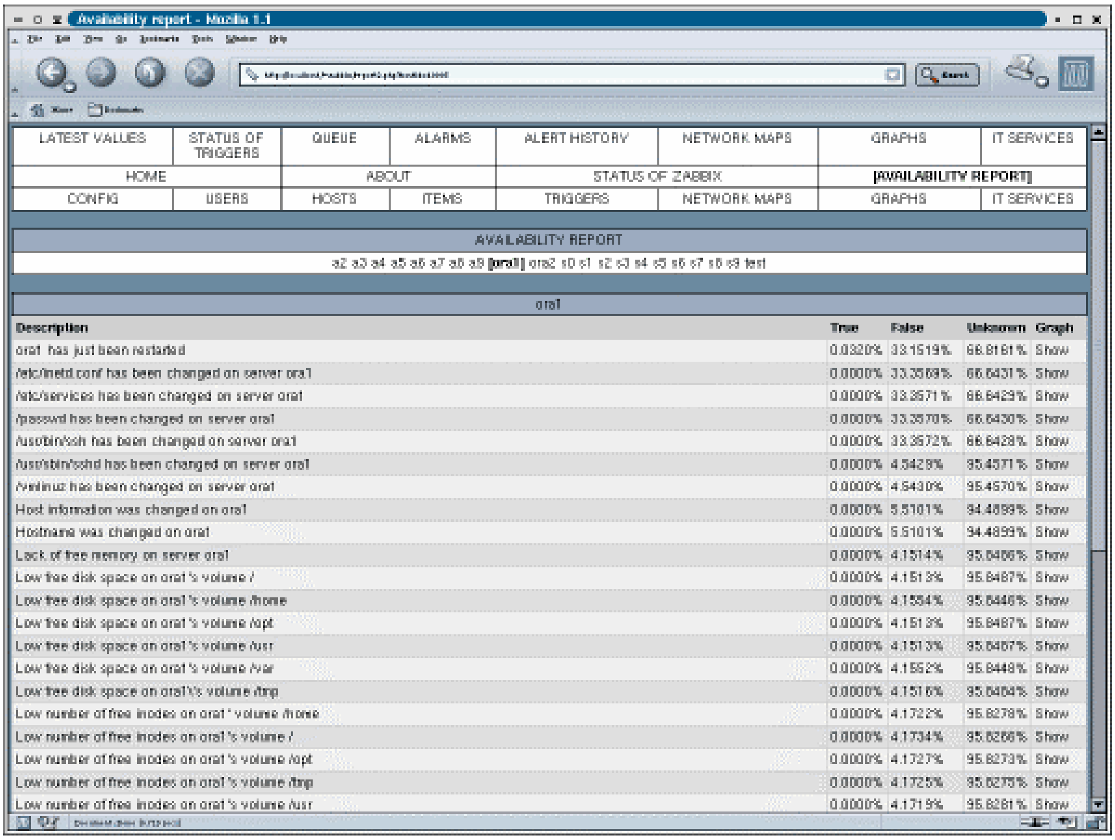


STATUS OF ZABBIX

Parameter	Value
Is zabbix_suckerd running ?	Yes
Is zabbix_trapperd running ?	Yes
Number of values stored in table history	1136004
Number of values stored in table alarms	5708
Number of values stored in table alerts	0
Number of triggers (enabled/disabled)	602(597/5)
Number of items (active/trapper/not active/not supported)	1077(999/0/0/78)
Number of users	2
Number of hosts (monitored/not monitored)	24(21/0)

AVAILABILITY REPORT

This report calculates amount of time a trigger has been in true and false states. For example, suppose you want to know what percentage of the time your web server is down. Assuming you have been monitoring the web server and have the “WEB server is down” trigger enabled, the availability report will display what percentage of the time the trigger was in the true state (web server down) and in the false state (web server running).



AVAILABILITY REPORT

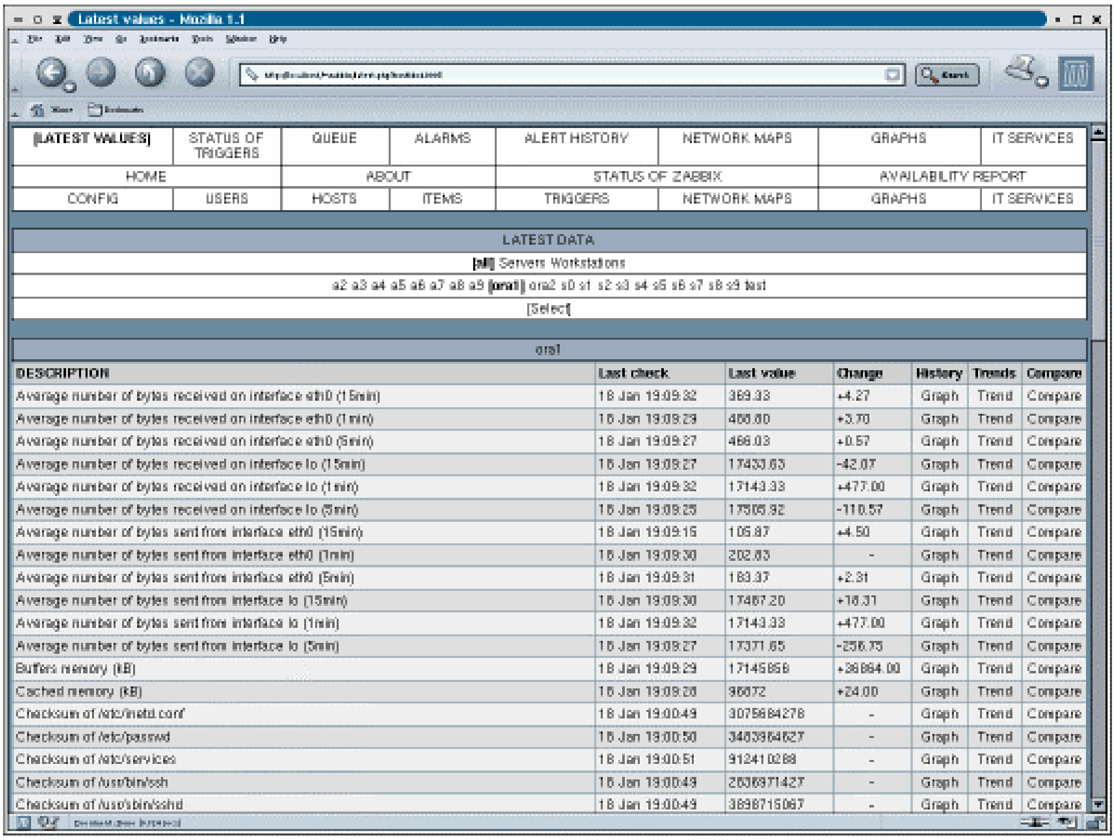
a2 a3 a4 a5 a6 a7 a8 a9 **[ora1]** ora2 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 test

ora1

Description	**True**	**False**	**Unknown**	**Graph**
ora1 has just been restarted	0.0320%	33.1519%	66.8161%	Show
/etc/inetd.conf has been changed on server ora1	0.0000%	33.3569%	66.6431%	Show
/etc/services has been changed on server ora1	0.0000%	33.3571%	66.6429%	Show
/passwd has been changed on server ora1	0.0000%	33.3570%	66.6430%	Show
/usr/bin/ssh has been changed on server ora1	0.0000%	33.3572%	66.6428%	Show
/usr/sbin/sshd has been changed on server ora1	0.0000%	4.5429%	95.4571%	Show
/vmlinuz has been changed on server ora1	0.0000%	4.5430%	95.4570%	Show
Host information was changed on ora1	0.0000%	5.5101%	94.4899%	Show
Hostname was changed on ora1	0.0000%	5.5101%	94.4899%	Show
Lack of free memory on server ora1	0.0000%	4.1514%	95.8486%	Show
Low free disk space on ora1's volume /	0.0000%	4.1513%	95.8487%	Show
Low free disk space on ora1's volume /home	0.0000%	4.1554%	95.8446%	Show
Low free disk space on ora1's volume /opt	0.0000%	4.1513%	95.8487%	Show
Low free disk space on ora1's volume /usr	0.0000%	4.1513%	95.8487%	Show
Low free disk space on ora1's volume /var	0.0000%	4.1552%	95.8448%	Show
Low free disk space on ora1\'s volume /tmp	0.0000%	4.1516%	95.8484%	Show
Low number of free inodes on ora1' volume /home	0.0000%	4.1722%	95.8278%	Show
Low number of free inodes on ora1's volume /	0.0000%	4.1734%	95.8266%	Show
Low number of free inodes on ora1's volume /opt	0.0000%	4.1727%	95.8273%	Show
Low number of free inodes on ora1's volume /tmp	0.0000%	4.1725%	95.8275%	Show
Low number of free inodes on ora1's volume /usr	0.0000%	4.1719%	95.8281%	Show

LATEST VALUES

The screen shows latest values of monitored parameters. The screen opens access to historical data, graphs and trends of collected information.



LATEST DATA

[all] Servers Workstations

a2 a3 a4 a5 a6 a7 a8 a9 [ora1] ora2 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 test

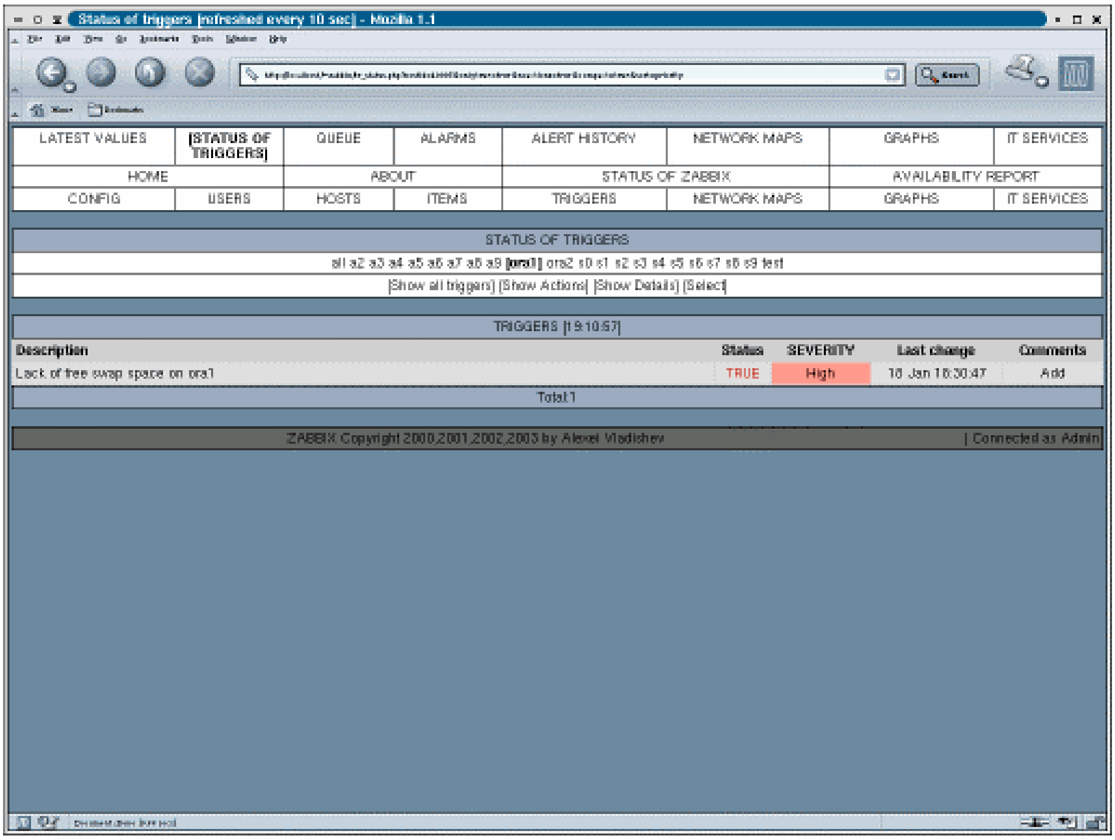
[Select]

ora1

DESCRIPTION	Last check	Last value	Change	History	Trends	Compare
Average number of bytes received on interface eth0 (15min)	18 Jan 19:09:32	369.33	+4.27	Graph	Trend	Compare
Average number of bytes received on interface eth0 (1min)	18 Jan 19:09:29	466.80	+3.70	Graph	Trend	Compare
Average number of bytes received on interface eth0 (5min)	18 Jan 19:09:27	466.03	+0.57	Graph	Trend	Compare
Average number of bytes received on interface lo (15min)	18 Jan 19:09:27	17433.63	-42.07	Graph	Trend	Compare
Average number of bytes received on interface lo (1min)	18 Jan 19:09:32	17143.33	+477.00	Graph	Trend	Compare
Average number of bytes received on interface lo (5min)	18 Jan 19:09:25	17505.92	-110.57	Graph	Trend	Compare
Average number of bytes sent from interface eth0 (15min)	18 Jan 19:09:15	105.87	+4.50	Graph	Trend	Compare
Average number of bytes sent from interface eth0 (1min)	18 Jan 19:09:30	202.83	-	Graph	Trend	Compare
Average number of bytes sent from interface eth0 (5min)	18 Jan 19:09:31	183.37	+2.31	Graph	Trend	Compare
Average number of bytes sent from interface lo (15min)	18 Jan 19:09:30	17487.20	+18.31	Graph	Trend	Compare
Average number of bytes sent from interface lo (1min)	18 Jan 19:09:32	17143.33	+477.00	Graph	Trend	Compare
Average number of bytes sent from interface lo (5min)	18 Jan 19:09:27	17371.65	-256.75	Graph	Trend	Compare
Buffers memory (kB)	18 Jan 19:09:29	17145856	+36864.00	Graph	Trend	Compare
Cached memory (kB)	18 Jan 19:09:28	96872	+24.00	Graph	Trend	Compare
Checksum of /etc/inetd.conf	18 Jan 19:00:49	3075684276	-	Graph	Trend	Compare
Checksum of /etc/passwd	18 Jan 19:00:50	3463964627	-	Graph	Trend	Compare
Checksum of /etc/services	18 Jan 19:00:51	912410288	-	Graph	Trend	Compare
Checksum of /usr/bin/ssh	18 Jan 19:00:49	2636971427	-	Graph	Trend	Compare
Checksum of /usr/sbin/sshd	18 Jan 19:00:49	3898715067	-	Graph	Trend	Compare

STATUS OF TRIGGERS

The screen shows latest status of triggers.



STATUS OF TRIGGERS

all a2 a3 a4 a5 a6 a7 a8 a9 **[ora1]** ora2 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 test

[Show all triggers] [Show Actions] [Show Details] [Select]

TRIGGERS [19:10:57]

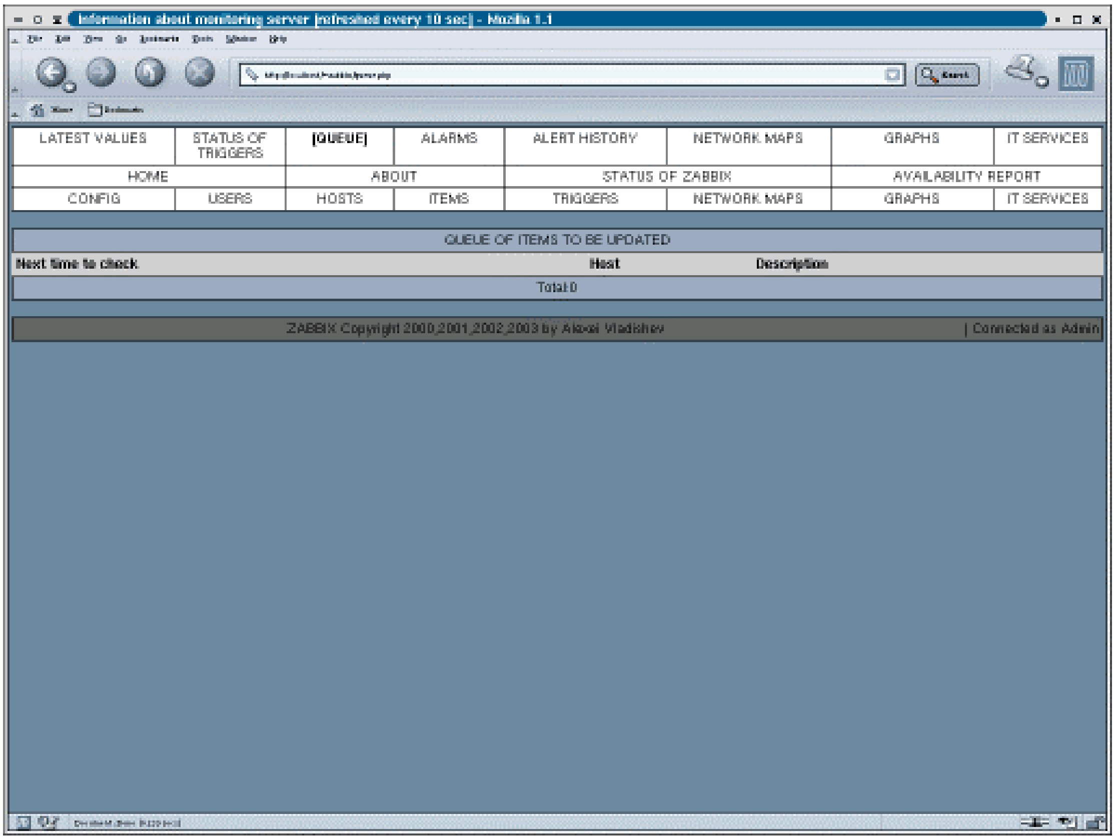
Description	Status	SEVERITY	Last change	Comments
Lack of free swap space on ora1	TRUE	High	18 Jan 18:30:47	Add

Total:1

ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev | Connected as Admin

QUEUE

The screen shows list of parameters that must be immediately retrieved.



QUEUE OF ITEMS TO BE UPDATED

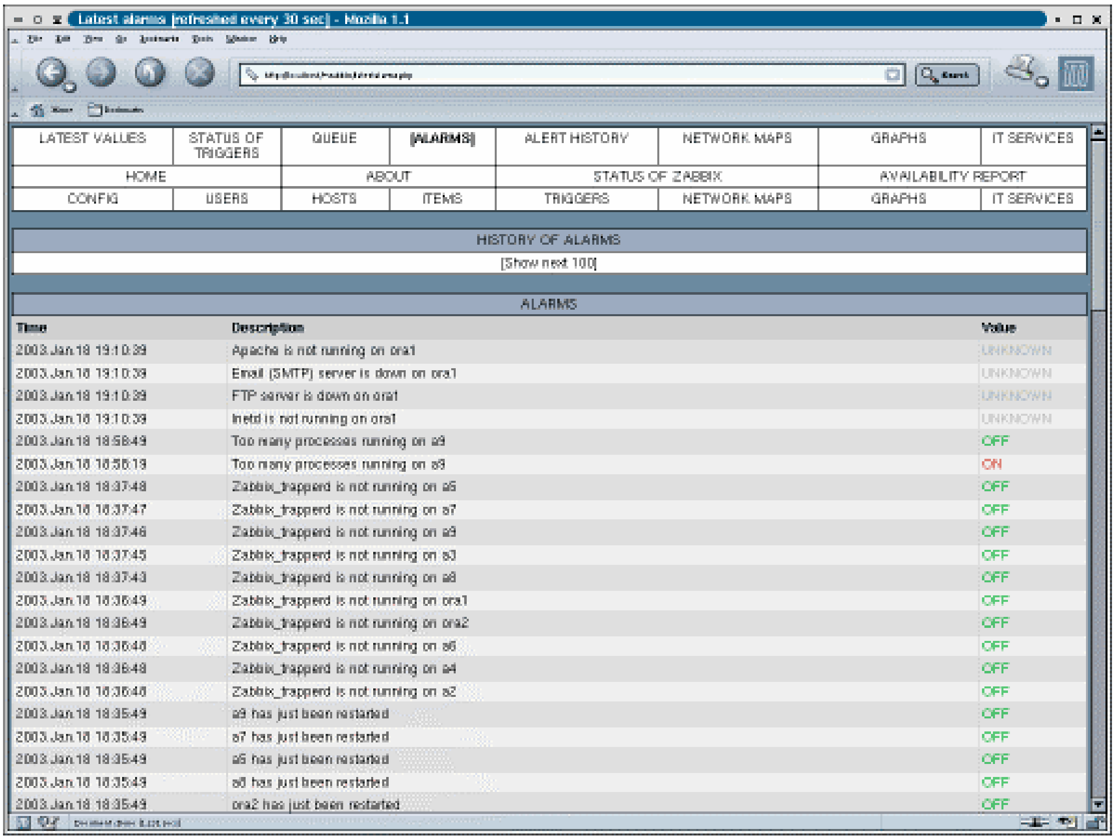
Next time to check	Host	Description

Total:0

ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev | Connected as Admin

ALARMS

The screen shows list of alarms.

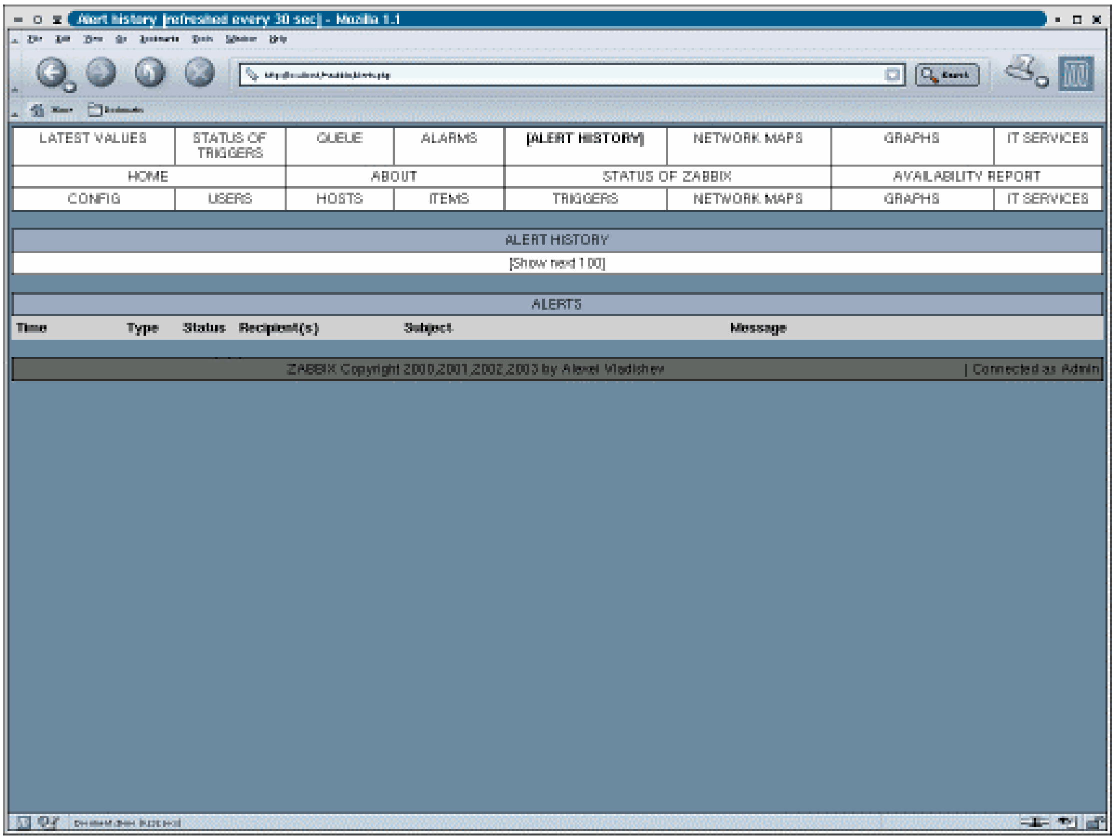


ALARMS

Time	Description	Value
2003.Jan.18 19:10:39	Apache is not running on ora1	UNKNOWN
2003.Jan.18 19:10:39	Email (SMTP) server is down on ora1	UNKNOWN
2003.Jan.18 19:10:39	FTP server is down on ora1	UNKNOWN
2003.Jan.18 19:10:39	Inetd is not running on ora1	UNKNOWN
2003.Jan.18 18:58:49	Too many processes running on a9	OFF
2003.Jan.18 18:56:19	Too many processes running on a9	ON
2003.Jan.18 18:37:48	Zabbix_trapperd is not running on a5	OFF
2003.Jan.18 18:37:47	Zabbix_trapperd is not running on a7	OFF
2003.Jan.18 18:37:46	Zabbix_trapperd is not running on a9	OFF
2003.Jan.18 18:37:45	Zabbix_trapperd is not running on a3	OFF
2003.Jan.18 18:37:43	Zabbix_trapperd is not running on a8	OFF
2003.Jan.18 18:36:49	Zabbix_trapperd is not running on ora1	OFF
2003.Jan.18 18:36:49	Zabbix_trapperd is not running on ora2	OFF
2003.Jan.18 18:36:48	Zabbix_trapperd is not running on a6	OFF
2003.Jan.18 18:36:48	Zabbix_trapperd is not running on a4	OFF
2003.Jan.18 18:36:48	Zabbix_trapperd is not running on a2	OFF
2003.Jan.18 18:35:49	a9 has just been restarted	OFF
2003.Jan.18 18:35:49	a7 has just been restarted	OFF
2003.Jan.18 18:35:49	a5 has just been restarted	OFF
2003.Jan.18 18:35:49	a8 has just been restarted	OFF
2003.Jan.18 18:35:49	ora2 has just been restarted	OFF

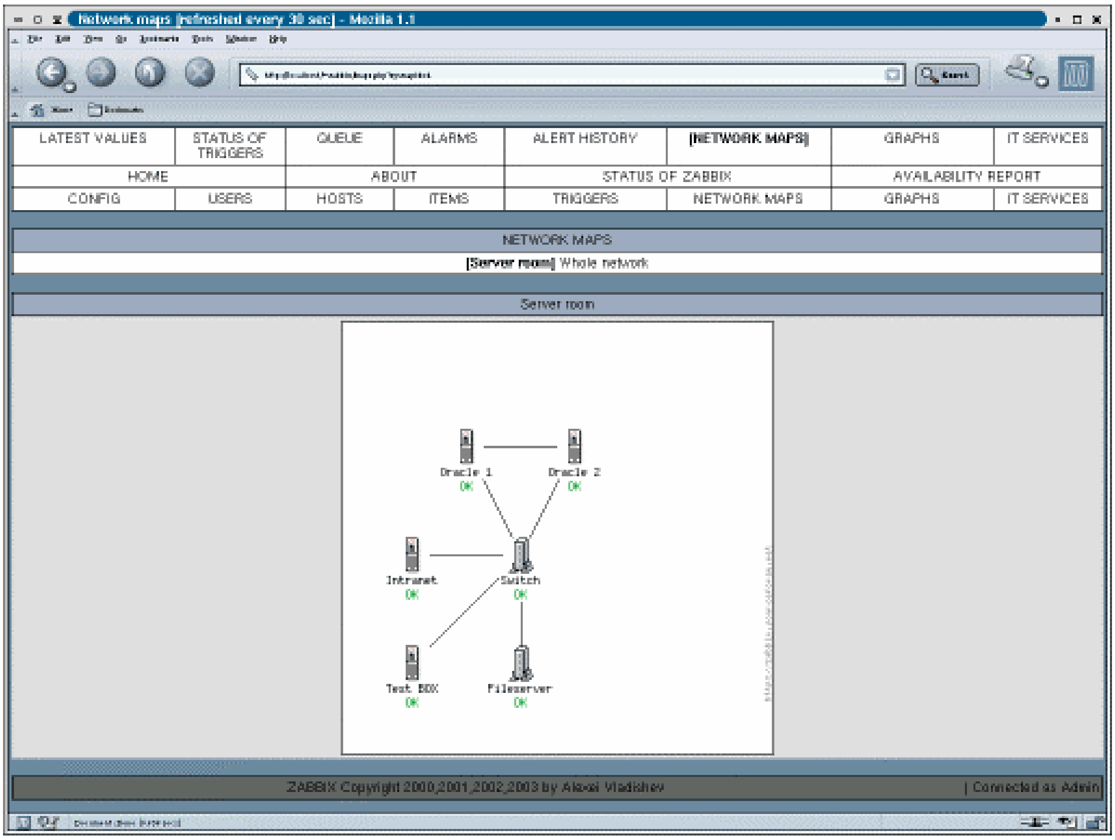
ALERT HISTORY

The screen shows list of alerts.



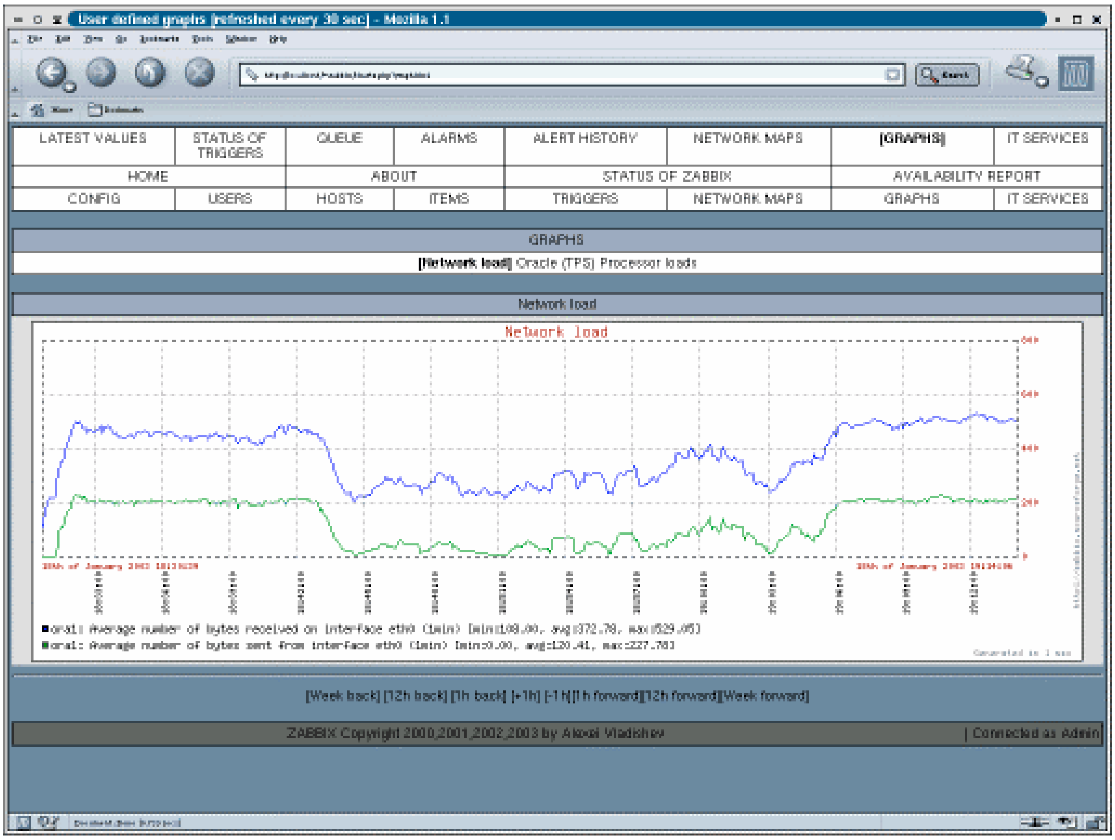
NETWORK MAPS

The screen shows list of network maps.



GRAPHS

The screen shows list of user-defined graphs.

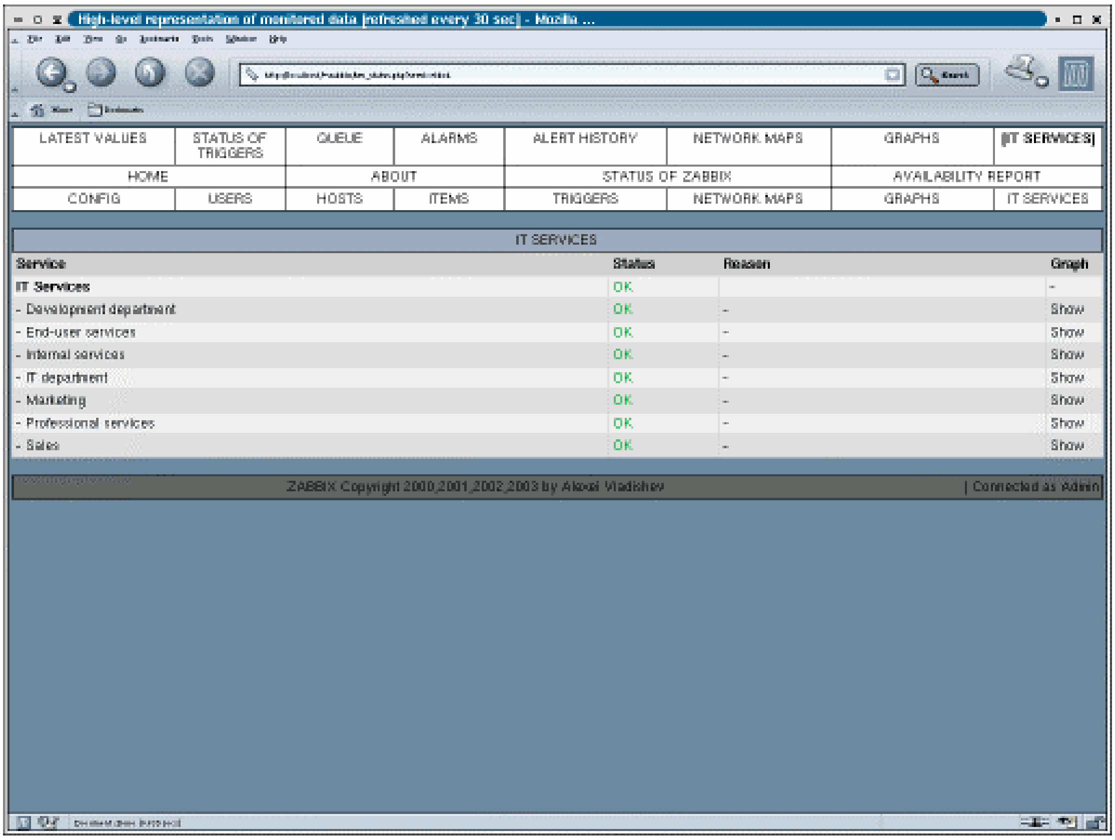

User defined graphs [refreshed every 30 sec] - Mozilla 1.1
LATEST VALUES
STATUS OF TRIGGERS
QUEUE
ALARMS
ALERT HISTORY
NETWORK MAPS
[GRAPHS]
IT SERVICES
HOME
ABOUT
STATUS OF ZABBIX
AVAILABILITY REPORT
CONFIG
USERS
HOSTS
ITEMS
TRIGGERS
NETWORK MAPS
GRAPHS
IT SERVICES
GRAPHS
[Network load] Oracle (TPS) Processor loads
Network load
Network load
[Week back] [12h back] [1h back] [+1h] [-1 h][1 h forward][12h forward][Week forward]
ZABBIX Copyright 2000,2001,2002,2003 by Alexei Vladishev
| Connected as Admin

oral: Average number of bytes received on interface eth0 (1min) [min:108.00, avg:372.78, max:529.05]

oral: Average number of bytes sent from interface eth0 (1min) [min:0.00, avg:120.41, max:227.78]

IT SERVICES

The screen shows high-level tree.



IT SERVICES

Service	Status	Reason	Graph
IT Services	OK		-
- Development department	OK	-	Show
- End-user services	OK	-	Show
- Internal services	OK	-	Show
- IT department	OK	-	Show
- Marketing	OK	-	Show
- Professional services	OK	-	Show
- Sales	OK	-	Show

17 ZABBIX utilities

17.1 Startup scripts

17.2 pinger.pl

The script can be found in *misc/pinger*. The script is used to periodically send ICMP pings to list of servers. In case, if it does not receive response from host, it executes:

```
zabbix_sender <server> <port> <host>:alive 0
```

otherwise:

```
zabbix_sender <server> <port> <host>:alive 1
```

The script is usually called from cron. Configure the script before use. Change location for “*hosts*” file. You must have **fping** installed to successfully run the script.

18 Database structure

The section describes structure of tables of ZABBIX database.

ACTIONS

The table contains list of actions to be applied if a trigger changes its state.

Column name	Type	Description
actionid	int	Unique action ID
triggerid	int	Trigger ID
userid	int	User ID
good	int	
subject	varchar	Subject of the message
message	varchar	The message itself
nextcheck	int	Time when the message should be sent next time

ALARMS

This table contains history of changes of trigger states. When a trigger changes its state, new record is added to the table.

Column name	Type	Description
triggerid	int	Trigger ID
clock	int	Time when trigger changed its status
value	int	Value of the alarm: 0 – trigger became FALSE 1 – trigger became TRUE 2 – trigger became UNKNOWN

ALERTS

This table contains history of alerts sent to users.

Column name	**Type**	**Description**
alertid	int	Unique alert ID
actionid	int	Action ID that generated this alert
clock	int	Time when this alert was generated
type	varchar	Time of alert: EMAIL – email based alert
sendto	varchar	Recipient(s). In case of EMAIL, this specifies email address.
subject	varchar	In case of email, specifies subject of the email.
message	blob	Message
status	int	o – not sent 1 – successfully sent
retries	int	Number of retries. zabbix_suckerd will not send message if the value is more that 2. Increased after unsuccessful try.

CONFIG

The table contains global configuration parameters.

Column name	**Type**	**Description**
smtp_server	varchar	Name of SNMP server
smtp_helo	varchar	HELO string used for SMTP server
smtp_email	varchar	Email address used by ZABBIX
alarm_history	int	ZABBIX will delete records in table alarms older than this value (in days)
alert_history	int	ZABBIX will delete record in table alerts older than this value (in days)

FUNCTIONS

The table contains list of simple functions used in trigger expressions.

Column name	Type	Description
functionid	int	Unique function ID
itemid	int	Item ID
triggerid	int	Trigger ID
lastvalue	double	Last (most recent) value
function	varchar	Function name: LAST, MIN, MAX, PREV, DIFF
parameter	int	Parameter to the function. Ignored if function does not need any parameters

GRAPHS

The table contains list of user-defined graphs (charts).

Column name	**Type**	**Description**
graphid	int	Unique graph ID
name	varchar	Name of the graph
width	int	Graph width
height	int	Graph height

GRAPHS_ITEMS

The table contains list of monitored items belonging to graph.

Column name	**Type**	**Description**
gitemid	int	Unique ID
graphid	int	Graph ID
itemid	int	Item ID
color	int	Color

GROUPS

The table contains list of host groups.

Column name	**Type**	**Description**
groupid	int	Unique ID
name	varchar	Name of the group

HISTORY

This table contains history of values of items.

Column name	**Type**	**Description**
itemid	int	Item ID
clock	int	Timestamp
value	double	Value of the item

HOSTS

The table contains list of monitored Hosts.

Column name	**Type**	**Description**
hostid	int	Unique host ID
host	varchar	Host name or IP address
status	int	Host status: 0 – monitored 1 – not monitored 2 – unreachable 3 – template
disable_until	int	In case if status is *unreachable*, do not check host and its items until this time

HOSTS_GROUPS

The table is used to define linkage between hosts and host groups.

Column name	Type	Description
hostid	int	Host ID
groupid	int	Group ID

ITEMS

This table contains definitions of monitored items.

Column name	**Type**	**Description**
itemid	int	Unique item ID
type	int	Type of the item: 0 – ZABBIX agent 1 – SNMPv1 2 – Trapper 3 – Simple check 4 – SNMPv2
value_type	int	Type of received values: 0 – float 1 – string
snmp_community	varchar	Name of community for SNMP request. Example: public
snmp_oid	varchar	Object ID for SNMP request
hostid	int	Host ID
description	varchar	Description of the item
key_	varchar	Key to be sent to monitored host
delay	int	Update interval. Defines how often retrieve this item (in sec)
history	int	Do not store more than *history* days of history data
lastdelete	int	Time of last deletion from table *history*
nextcheck	int	Time when next value should be retrieved
lastvalue	double	Last value retrieved from host
lastclock	int	Time when last value was retrieved
prevvalue	double	Previous retrieved value
trapper_hosts	varchar	Comma-delimited list of IP addresses of hosts allowed sending data of the item. For item type *Trapper* only.
status	int	Status of the item: 0 – active 1 – disabled 3 – not supported by agent

MEDIA

This table contains list of available medias for given user.

Column name	Type	Description
mediaid	int	Unique media ID
userid	int	User ID
type	int	Type of media: 0 – EMAIL
sendto	varchar	Address to send alert to
active	int	Status of the media: 0 – active 1 – disabled

SERVICES

This table contains list of defined IT services.

Column name	Type	Description
serviceid	int	Unique service ID
name	int	Description of the service
status	int	Status of the service
algorithm	int	Algorithm used to calculate status of the service: 0 – none 1 – MAX status of child services
triggerid	int	Is the service linked to the trigger: NULL – it is not otherwise, trigger ID

SERVICES_LINKS

This table is used to define connections between different services to form IT Service tree.

Column name	**Type**	**Description**
linkid	int	Unique link ID
serviceupid	int	ID of higher level service
servicedownid	int	ID of lower level service
soft	int	0 – hard link 1 – soft link

RIGHTS

The table contains list of user permissions.

Column name	Type	Description
rightid	int	Unique ID
userid	int	User ID
name	varchar	Resource name
permission	char	Permission: R – read only U – read and write A – add H – restrict access (hide)
id	int	Resource ID

SYSMAPS

The table contains list of defined network maps.

Column name	**Type**	**Description**
sysmapid	int	Unique network map ID
name	varchar	Name of the network map
width	int	Width of the map
height	int	Height of the map

SYSMAPS_HOSTS

The table contains definition of host displayed on network map.

Column name	Type	Description
shostid	int	Unique ID
sysmapid	int	Network map ID
hostid	int	Host ID
label	varchar	Label displayed under host icon
x	int	X
y	int	Y

SYSMAPS_LINKS

The table contains list of connectors between hosts displayed on network map.

Column name	**Type**	**Description**
linkid	int	Unique link ID
sysmapid	int	Network map ID
shostid1	int	ID of first host
shostid2	int	ID of second host

TRIGGERS

The table contains list of triggers.

Column name	**Type**	**Description**
triggerid	int	Unique trigger ID
expression	varchar	Trigger’s expression
url	varchar	URL
description	varchar	Description of the trigger
status	int	Status of the trigger: 0 – enabled 1 – disabled
value	int	Value of the trigger: 0 – FALSE 1 – TRUE 2 – UNKNOWN
priority	int	
lastchange	int	Time when trigger value was changed
dep_level	int	Dependency level. The number shows how many triggers depend on this trigger.

SERVICES

The table contains user-defined list of services.

Column name	Type	Description
serviceid	int	Unique service ID
name	varchar	Service name
status	int	Status of the servide
triggerid	int	NULL – service is not linked to a trigger otherwise, trigger ID

SERVICES_LINKS

The table contains links between services.

Column name	**Type**	**Description**
serviceupdid	int	ID of higher level service
servicedownid	int	ID of lower level service
soft	int	0 – hard link 1 – soft link

SESSIONS

The table contains list of user sessions.

Column name	**Type**	**Description**
sessionid	varchar	User ID
userid	int	User ID
lastaccess	int	Last time the session was used

USERS

The table contains list of ZABBIX’s users.

Column name	**Type**	**Description**
userid	int	Unique user ID
name	varchar	Name of the user
surname	varchar	Surname of the user
alias	varchar	User alias(short name)
passwd	varchar	MD5 hash of user password

19 Problems and Common Errors

19.1 Authentication failed in case if user name and password is correct

During initial installation, no *data.sql* loaded into database.

Check system date on both ZABBIX server and client machine where WEB browser is running. The difference should not be more than 20 minutes.

Browser should be configured to accept cookies. ZABBIX uses cookies to store session-related information.

19.2 Undefined references to compress and uncompress

```
/usr/lib/libmysqlclient.a(my_compress.o): In function ‘my_uncompress’:

my_compress.o(.text+0xa1): undefined reference to ‘uncompress’

/usr/lib/libmysqlclient.a(my_compress.o): In function ‘my_compress_alloc’:

my_compress.o(.text+0x138): undefined reference to ‘compress’
```

These messages indicate that *./configure* is unable to find **libz** installed. The problem should appear only in case if flag *enable-static* is specified. Install package **libz-dev** package to get rid of the problem.

20 How ZABBIX Compares to Other Monitoring Software

20.1 Open Source monitoring software

BigBrother
Nagios
MRTG

* ...

20.2 Commercial monitoring software

SysOrb
HP OpenView
WhatsUp Gold
Windows SMS

* ...

21 ZABBIX in the future

The sections contains list of improvements planned to be included into future releases of ZABBIX.

21.1 This manual TODO

Add “To be finished...” to all unfinished sections

21.2 TODO (1.1 stable)

- Distributed monitoring
- Get rid of Apache and PHP (2.0?)
- Data encryption (SSL)
- Centralised monitoring of log files
- ZABBIX to periodically generate and send by email set of standard reports in PDF/HTML/PS format
- Graphs to be included into alert’s email

21.3 TODO (1.0 stable)

- support for IT helpdesk
- *./configure* to correctly find SNMP libraries and headers (RedHat)
- more reports
- all ZABBIX errors and warnings to be numbered and documented and explained
- better support for SNMP (send and receive SNMP traps)
- full set of regression tests
- same time navigator for all graphs
- add safe wrapper functions for strncat, strcpy, etc
- change trigger status to UNKNOWN when status of a host is changed
- network map. Connect an icon to a trigger/service
- create one universal *chart.php* to be used for graphs, charts, and trends
- add MRTG-like graphs
- add support for *check_service[ldap]*
- add support for Kb,bytes,kps, etc
- improve functionality of IT Services

- dlopen/dlsym for agents
- **zabbix_suckerd** to send ICMP pings

22 Support of ZABBIX

22.1 Free support

Mailing lists at http://sourceforge.net/mail/?group_id=23494
ZABBIX related discussions at http://sourceforge.net/forum/?group_id=23494
Feature requests can be viewed and registered at http://sourceforge.net/tracker/?atid=378686&group_id=23494&func=browse

Email address of Alexei Vladishev, author and main developer of ZABBIX is alex@gobbo.caves.lv

Commercial support is currently available to Baltic States (Estonia, Latvia and Lithuania) only. Hopefully, other countries will be included in the list as soon as ZABBIX 1.0 (stable) is ready.

If support is an issue for you, several options are available as well. Please, contact alex@gobbo.caves.lv to get additional information on available commercial support levels.

Benefits of commercial support may include:

- guaranteed response time
- direct contact to ZABBIX developers
- customer-specific development made by ZABBIX developers
- feature requests are considered top priority
- printed ZABBIX manual plus installation on CD
- other benefits

23 Credits

ZABBIX team wants to thank the guys from *http://sourceforge.net* for providing hosting for the project. Our team also wants to thank all the ZABBIX users who have sent corrections and suggestions. This sort of feedback helps us make the software better.

23.1 Developers of ZABBIX

Alexei Vladishev

Has written most of ZABBIX code including PHP frontend.

23.2 Contributors to ZABBIX

In alphabetical order:

ALEXANDER KALIMULIN

Help with various issues related to C, C functions, etc

ALEXANDER KIRHENSTEIN

Suggested fixes to make ZABBIX work under SCO.

ARTURS ABOLTINS

Patch to allow connection to MySQL using UNIX socket. Support for graceful shutdown in case if MySQL server goes down.

CHARLIE COLLINS

Start-up scripts. Significant improvements of the Manual. Thanks Charlie!

DENIS USTIMENKO

Support for querying SNMP parameters by IP address.

DANIEL HIGGINS

Improvements for email sending routines. Other changes.

ERIK CARLSEEN

Many excellent ideas.

EUGENY BACULA

Many suggestions for improvements.

JEFF REDDING

Support for non-GCC compilers

JOHN CRUNK

Start-up scripts for RedHat 8.0

JOSH KONKOL

Help with testing

KASPARS CIKMACS

Lots of new ideas based on real experience of using ZABBIX.

LAURIS STIGLICS

Select criteria in for “Status of Triggers”

LUKAS MACURA

Many ideas

MARIUSZ ...

Support for system[procload] on Solaris 2.6. Improvements for graphs. Improvements for system maps.

MICHAL SUSZYCKI

Help with autoconf and automake issues.

MIKE HOOLEHAN

Help with making the ZABBIX Manual correct and understandable.

OLIVER SIEGMAR

Fixes in SQL statements of WEB frontend.

RICKARD PLARS

Help with fixing coredump for zabbix_suckerd.

SEBASTIEN “SLIX” LIENARD

Fixed selection of hosts and icons in sysmap.php

VIKTOR KIRHENSTEIN

Native ZABBIX agent for WIN32 platforms.

24 Glossary of terms

Term	Description
GNU	
SNMP	

25 GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 30, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

26 References

MySQL Reference Manual for version 3.23.39

References

PostgreSQL User's Guide

References

PHP Manual

References

The ZABBIX home page at http://zabbix.sourceforge.net

27 Appendix

27.1 Warnings and Errors