# Service Monitor 0.6.0

## Technical documentation

**Katarzyna Wladyszewska, Hadden Sp.J.**

# Service Monitor 0.6.0: Technical documentation

by Katarzyna Wladyszewska

Published April 2010

# Table of Contents

# List of Tables

# Chapter 1. Conventions

The following typographical conventions are used in this manual:

**Table 1.1. The typographical conventions used in this manual**

| Font | What the font represents | Example |
|---|---|---|
| *Italic* | Environment variables. | The name is kept in environmental variable *$DAVIDPRIVDIR*... |
| *Italic* | Synopsis options. | [*-l,--log-facility log_facility*] |
| **Bold** | Names of programs and products. | **damcsud** is a part of **Operation Manager-a**. |
| `Computer` | Names of options and menus. | There is `Show tool bar` option in `View` menu. |
| `Computer` | Names of files and directories. | ... reads its configuration file `.damadbudrc`. |
| `Computer` | Names of windows and dialog fields. | In `A sessions property` window, in `Sticking string` field, you can write... |
| `Computer` | Names of buttons. | Pressing `Apply` button lets you apply changes. |
| **`Computer Bold`** | Math formulas. | **`exp( -x ), when a = 0 1 / pow( a , a ) * pow( x , a ) * exp( -x + a ), when a > 0`**. |
| **`Computer Bold`** | Terms used in David system terminology. | **`SNMP Data`** - a kind of data... |
| **`Computer Bold`** | Contents of configurations files. | **`action`**<br>**`{`**<br>**`...`**<br>**`}`** |

# Chapter 2. General information about David system

## 2.1. General

**David system**is a network management system. It is a packet of applications (modules) that allows computer network to be monitored and managed in real-time through the Internet. There is only one condition that managed devices must meet. Each device must provide SNMP (Simple Network Management Protocol) service. SNMP is the most common management protocol in the Internet so that requirement shouldn't be difficult to meet. Here is the list of typical devices that can be monitored:

- IP routers,
- ATM switches,
- manageable ethernet switches,
- UPSes with a SNMP adapter,
- TV-SAT modems that allow IP devices to work in TV cable networks,
- computers.

One of the most important feature of **David system** is its architecture. It's built of high level configureable and independent from one another modules. This principle is the most essential rule of the project. In consequences, in th metter of speaking, the same modules may build different management system. Here are the main features of **David system**:

- general thinking in information flow controlling that come form high level independence of modules of the system,
- high level configureability of the system modules that allows a special configuration of **David system** to reach end-user expectations so close as it's only possible,
- the system scalability, so you can build up the system adding additional modules in very easy way; note that these modules needn't to be part of **David system** at all; adding another monitored devices to the system is a very easy procedure,
- using shell scripts in information processing is opportunity for modeling information and influence on processing it,
- all configuration files of **David system**, files with input/output data and log files are text files,

- using SNMPv1, SNMPv2C and SNMPv3 to communicate with monitored devices.

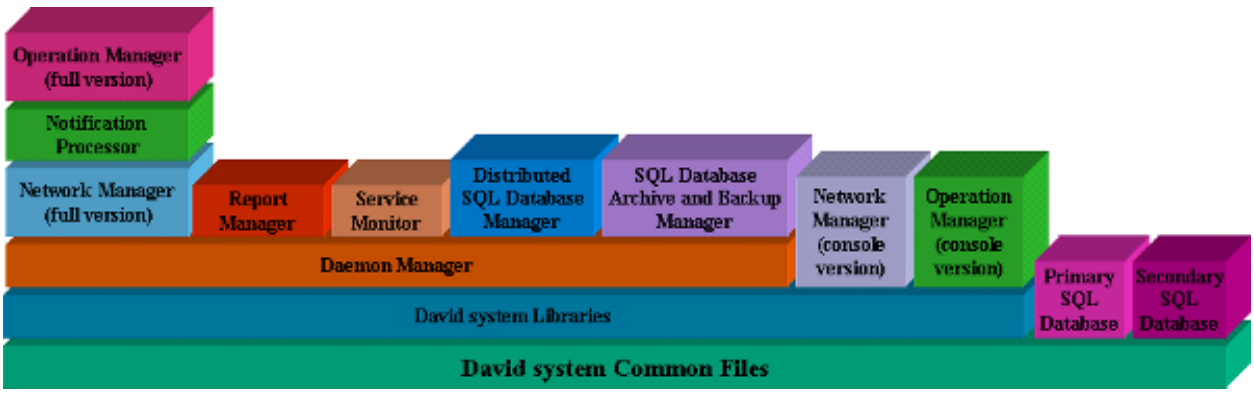
# 2.2. David system architecture

**Table 2.1. David system products**

| Product | Description |
|---|---|
| David system Common Files | The product, during its installation, prepares the rudimentary directory tree for other products of **David system**. It also contains some essential and common files for all the products. Thus, this is a fundamental product of **David system** required by other its products. |
| Primary SQL Database | The product installs the primary SQL database of **David system**. Every single installation of **David system** must have only one the primary database. |
| Secondary SQL Database | The product installs the secondary SQL database of **David system**. Each installation of **David system** may have many secondary databases or none. It allows to distribute the SQL database of **David system** among many servers. |
| David system Libraries | This product provides libraries of **David system** required by its applications. Many other products of **David system** require that one. |
| Daemon Manager | It engages in running and terminating daemons of **David system** as well as monitoring of their work. |
| Network Manager (full version) | The product using SNMP protocol allows to visualise a topology of monitored networks and auto-discover devices in managed networks. The state of monitored devices also is visualized. The product also collects data from monitored devices using SNMP protocol and allows you to manage user accounts. |
| Network Manager (console version) | The product, through a graphic application, allows to visualize a topology of monitored networks and shows states of monitored resources. It allows you to control daemons monitoring devices as well as that ones gathering data. Currently, most of functions of that application is obtainable through web applications. |
| Notification Processor | The product chiefly engages in processing SNMP Trap notifications coming from monitored devices to management stations. The received messages can be formatted to the human readable forms, and then recorded as well. The processed notifications can be passed on to future processing. |
| Operation Manager (full version) | It can run specified actions on the basis of received data. Sophisticated estimation process depends on information coming from other products of **David system** and correlation of that information. It tries to build more intelligent and useful notifications then just simple reactions to incoming |

| Product | Description |
|---|---|
| | events. The graphic application displays notifications about received events and allows to play audio files as well as reading messages by an outer speach synthesizer. |
| Operation Manager (console version) | The product contains a graphic application displaying notifications about events and allowing to play audio files as well as reading messages by an outer speach synthesizer. |
| Report Manager | The product processes recorded SNMP Trap notifications, entries about pending operations and entries about state changes of monitored devices (ping objects, network interfaces and BGP peers), and generates raports on the basis of them. Raports can be viewed using a Web application. |
| Service Monitor | The product monitors selected network services on application level. In order to do this it monitors selected TCP ports of specified hosts. It checks both availability of ports and a correct reaction for a few selected network protocols (HTTP, SMTP, FTP). It also can verify correctness of work of selected services by verification of received data. Results of its work can be viewed as reports and graphs made available by a Web application. |
| SQL Database Archive and Backup Manager | It archives the SQL Database used by **David system** applications. |
| Distributed SQL Database Manager | It allows to devide the database of **David system** into one primary database and many secondary ones. Such step boosts performance of the system and decreases load of the servers where daemons of **David system** work. The migration takes place during the rutine work of the system. Such division may be altered many times. |

Dependences between the **David system** products are shown on the following chart:.




**David system** functionality can be very large and it depends on particular configuration a lot. The most important features of **David system** are:

- discovering and visualization of monitored networks topology including visualization of states of

particular nodes;

- possibility of building control panels to monitored devices (they must support SNMP protocol), regardless of device providers;
- formatting and recording SNMP Traps sent by agents working on monitored devices;
- automatic reaction to specified SNMP Traps received from monitored devices;
- possibility of identification of an operator that has received an alert from the system about a problem;
- collecting data concerning parameters of monitored devices;
- automatic reaction to incorrect values of data that were found during data collecting;
- recording pending cases, processed by the system, which have been created as responses for events detected by the system in a monitored network;
- monitoring selected network services on application level.

# Chapter 3. Terminology

## 3.1. Authorization process made by David system products

The modules of David system which need to do an authorization of message senders (i.e. **damsnmpdaud**, **dnmmsd**, **dgnsd**), use the library, that checks whether an IP address of a sender matches with any record found in the file `.known.host`. The library expects to find the file in a directory pointed by a variable *confdir* in the file `/etc/system-david.conf`.

Records in the file `.known.host` are regular expressions specifying acceptable IP addresses.

## 3.2. David system terminology used in the documentation

There is an explanation of some terms, that are used in David system and its documentation:

- **`massages (information)`** - data received by interfaces of **Operation Manager**, its data analysers and **Cases Database Unit** of the product.
- **`notifications`** - the term often is used in the products: **Notification Processor**, **Operation Manager** and **Report Manager**; There are mostly data, that a source are SNMP agents working on network monitored devices.
- **`events`** - the term often is used in the products: **Operation Manager** and **Report Manager**; and it describes a being, that a source is SNMP Trap or SNMP Data; an **`event`** is always a part of a **`case`**;
- **`cases`** - the term often is used in the products: **Operation Manager** and **Report Manager**; and it describes a group of events connected one another; one **`event`** at last must be included in a **`case`**;
- **`SNMP Trap`** - a kind of data of **Operation Manager** product, which a source are received responses from SNMP agents; SNMP Traps aren't answers on the requests sent by a management station, but they are sent by agents managing network interfaces and processed by **Notification Processor** product;
- **`SNMP Data`** - a kind of data of **Operation Manager** product, which a source are received responses from SNMP agents on request which a management station sent to them by **Network Manager**.

# Chapter 4. Installation

## 4.1. The main configuration file of David system

The essential configuration file of David system in `/etc/david-system.conf`. It contains entries as pairs: key = value. Basically, except the entry `default_email_recipient`, there is no such need to modify any record in that file. All necessary modifications are made during installation processes of particular David system products. Below, there is a list of all entries along with their descriptions that may occur in this basic configuration file.

- `user` - a name of the user with which rights all daemons of David system works;
- `default_email_recipient` - the default e-mail address where messages from David system applications are sent;
- `bindir` - the directory containing David system applications (default: /usr/bin/david-system);
- `libdir` - the directory containing David system libraries (default: /usr/lib/david-system);
- `incdir` - the directory containing David system headers (default: /usr/include/david);
- `confdir` - the directory containing David system configuration files (default: /etc/david-system);
- `logdir` - the directory containing log files of David system applications (default: /var/log/david-system);
- `sharedir` - the directory containing various files (images, audio files, web files) of David system (default: /usr/share/david-system);
- `docdir` - the directory containing various files (images, audio files, web files) of David system (default: /usr/share/david-system);
- `vardir` - the directory containing archive files of David system SQL database (default: /var/lib/david-system);
- `is_sqldb_installed` - the flag that indicate whether the SQL database of David system has been installed or not.

## 4.2. Dedicated account for service of David system

There is no needs to run any David system module as superuser (usually an account `root` with UID equals 0). Even if some David system daemon requires root rights when starting, there is always possibility to specify, as one of the daemons starting arguments, a user that rights should be taken.

It is a good idea to add a new user to an operating system, under which control David system will work.

# 4.3. Directories of David system

This hierarchy depends on a particular configuration of David system. In the default system configuration, David system contains the following directories:

- `/usr/bin/david-system` - binaries and shell scripts;
- `/etc/david-system` - configuration files;
- `/usr/share/doc/david-system` - the documentation;
- `/usr/share/david-system` - graphic and audio files, web portal;
- `/usr/include/david` - David system header files;
- `/usr/lib/david-system` - David system libraries;
- `/var/log/david-system` - log files;
- `/var/lib/david-system` - archive files of the David system SQL database;

# 4.4. Configuration of syslogd daemon

David system modules use `syslog` subsystem available on UNIX platforms. Default configuration of the system modules causes that log messages are sent with local6 `facility`. It may be changed for every module during its startup. Its recommended to configure `syslogd daemon` to write all messages from David system modules into one place (one or more files with characteristic name i.e.: david.log).

# Chapter 5. Service Manager requirements

The following requirements must be met by a management platform on which **Service Manager** will work:

- installed, compatible version of **Daemon Manager**.

# Chapter 6. Installation

## 6.1. Installation from the RPM package

You must be `root` to install the product. The typical installation looks as this one following below:

- Install the product:

```
rpm -i david-xxx-sm-yyy.rpm
```

## 6.2. Installation from the script

You must be `root` to install the product. The typical installation looks as this one following below:

- Uncompress and unpack the archive:

```
gunzip david-xxx-sm-yyy.i386.tar.gz
tar xf david-xxx-sm-yyy.i386.tar
```

  The operations create david-xxx-sm-yyy.i386 directory in your current directory.

- Change your current directory to david-xxx-sm-yyy.i386:

```
cd david-xxx-sm-yyy.i386
```

- Read LICENSE file from the current directory and CONTINUE THE INSTALLATION, ONLY WHEN YOU ACCEPT ALL CONDITIONS INCLUDED IN THE LICENSE.

- Run the installation script:

```
./install
```

# Chapter 7. General

## 7.1. Functionality

**Service Monitor** makes possible:

- monitoring of availability of application services offered by hosts;
- time monitoring of connections to application services;
- duration monitoring of connections to application services;
- verification of correctness of work of selected services;
- visualization of reports width current and historical data (logs and graphs).

## 7.2. Description

**Service Monitor** monitors selected network services on application level. In other words: TCP ports on remote hosts are monitored. Also connection establish times are monitored, their durations and errors, if the connection can't be established. In special cases verification of correctness of work of selected services is possible by checking of received data.

Only services defined in configuration are monitored. Hosts are specified by a DNS name, an IPv4 or IPv6 address. Each host can have many TCP ports monitored.

On the base of monitored data a special Web application makes reports. The report can be presented as graphs or a list of logs.

## 7.3. Related articles

Network Service Monitor (dnsmd)

Network Service Monitor Configurator

Network Service Browser

# Chapter 8. Network Service Monitor (dnsmd)

## 8.1. General

**dnsmd** is **Network Service Monitor** and it is a part of **Service Monitor**. It is a daemon process which works all the time the system is running. It monitors availability of network services on hosts, on application level during its work. It also records response times of services, but a connection can't be realised, it records an error which follows. A list of hosts with a list of their TCP ports, that are monitored, is configured by a user by Network Service Monitor Configurator.

## 8.2. Synopsis

**dnsmd** can be run with the following options: [*-P,--pid-file filename*] [*-l,--log-facility log_facility*] [*-L,--log-level log_level*] [*-u,--run-as-user username*] [*--background*] [*-v,--version*] [*-h,--help*]

## 8.3. Options

**Table 8.1. dnsmd options**

| Option | Description |
|---|---|
| *-P,--pid-file filename* | Write PID to the specified file. |
| *-l,--log-facility log_facility* | Choose log facility: daemon \| user \| local0 \| ... \| local7 (default: local6). |
| *-L,--log-level log_level* | Choose log level (on stderr and syslog) i.e. messages of selected level and more important levels will be logged: emerg \| alert \| crit \| err \| warning \| notice \| info \| debug0 \| ... \| debug2 (default: notice). |
| *-u,--run-as-user username* | Drop root privileges and run server as the specified user. |
| *--background* | Go to background after startup. |
| *-v,--version* | Display version number on stderr and exit. |
| *-h,--help* | Display this help and exit. |

## 8.4. Description

After startup, the program enters its configuration into databases. It includes a list of hosts, and each host includes a list of TCP ports, that should be moniotred. Network Service Monitor Configurator is a Web application, that allows to prepare such configuration for **Network Service Monitor**. Changing of the configuration doesn't require restart of **Network Service Monitor**, because it checks in the normal course, if its configuration isn't changed, and the program can load it again if necessary.

After loading of the configuration, the program periodically connects with TCP ports of specified hosts. A service procedure of the connection depends on a configuration of the given port. Information about each connection are saved in a Database and they are a source of reports, that Network Service Browser prepares. If a connection doesn't work, information about error is saved which follows. Information, that is saved about each connection, the following table presents:

**Table 8.2. Saved information about each connection made by dnsmd program**

| Field | Meaning |
|---|---|
| Start time | Start time of the connection procedure with a remote service (TCP port on a remote host). |
| Connetion time | Time from starting of a connection procedure to closing of system function `connect()`. |
| Total time | Time from starting of a connection procedure to closing of the connection. |
| Operation code | A code of closing of the operation: OK, if the connection has no errors or en error code which follows during establishing of the conncection. |
| A code of action script | A code which was returned by a script, if it was run for the connection. |

## 8.4.1. Built in, serviced types of TCP protocols

**dnsmd** program has built in service of some well-known protocols of application level. There are:

- SMTP
- HTTP
- FTP
- HTTP i FTP Proxy

Each of protocols is serviced in specific way for it, and it was described shortly in farther part of the document.

### 8.4.1.1. Service of SMTP protocol

Service of SMTP protocol is that it sends series of commands and picks up results of their work. A list of operations, that are made during service of SMTP protocol, is presented below:

1. Receiving and verification of a hello code.
2. Sending: `HELO david`.
3. Receiving and verification of a response code.
4. Sending: `QUIT`.
5. Receiving and verification of a response code.
6. Waiting for closing of the connection by a remote host.

### 8.4.1.2. Service of HTTP protocol

Service of HTTP protocol is that it sends series of commands and picks up results of their work. A list of operations, that are made during service of HTTP protocol, is presented below:

1. Sending `GET http://hostname/url HTTP/1.0`.
2. Sending `Host: hostname`
3. Sending optionally `Authorization: Basic coded_user_and_password` (algorithm base64 is used to code a user and a password).
4. Waiting for random data
5. Waiting for closing of the connection by a remote host.

The last command shows only then a user and a password are given in the port configuration. A word `hostname` is replaced with a real name of remote host while `url` is replaced with a path, if it was given in the port configuration.

A result of work of these commands is not veryfied in any way. **dnsmd** program waits for rendom data and closing of the connection by a remote host.

### 8.4.1.3. Service of FTP protocol

Service of FTP protocol is that it sends series of commands and picks up results of their work. A list of operations, that are made during service of FTP protocol, is presented below:

1. Receiving and verification of a hello code.
2. Sending: `QUIT`.
3. Receiving and verification of a response code.

4. Waiting for closing of the connection by a remote host.

### 8.4.1.4. Service of proxy protocol for HTTP i FTP

Service of proxy protocol for HTTP and FTP is similar to service of HTTP protocol, and it is used to send a series of commands and pick up results of their work. A list of operations, that are made during service of proxy protocol for HTTP and FTP, is presented below:

1. Sending `GET url HTTP/1.0`.
2. Sending optonally `Authorization: Basic coded_user_and_password` (algorithm base64 is used to code a user and a password).
3. Waiting for random data
4. Waiting for closing of the connection by a remote host.

The last command shows only then a user and a password are given in the port configuration. A word `url` is replaced with a real location of the file, if it was given in the port configuration. If `url` wasn't given in the configuration, a default file is an address `http://www.w3.org/`.

A result of work of these commands is not veryfied in any way. **dnsmd** program waits for rendom data and closing of the connection by a remote host.

## 8.4.2. Passing an external script on service of conncection

Service of conncetion also can be possible by an external script specified in a given port configuration. Then, after connection of **dnsmd** program with a remote host, remote control is passed on to an external program. Its stdin and stdout are redirected on a connection. In this way it can direct exchanging of data. A code of finishing of work is returned by the script and it is saved in result of a given connection.

The script receives connection parameters as its working parameters:

- A name of a remote host.
- IP of a remote host.
- A number of TCP port, on that the conncetion was done.
- URL which is configured for the port.
- A user whose is configured for the port.
- A password which is configured for the port.

You should remember, that a work of the script is longer then `timeout` parameter for the port, the script work will be disconnected after this time.

### 8.4.3. Errors, that can appear during establishing of connection.

The following errors can appear during establishing of connection.

**Table 8.3. Errors of establishing of the connection.**

| Error | Meaning |
|---|---|
| EADDRESS | Error appers during translation of DNS name of a remote host into its IP address. |
| ECONNECT | `connect()` function isn't finished success. |
| ENETDOWN | A network, that a remote host works, doesn't work. |
| ENETUNREACH | A network, that a remote host works, is unreach. |
| ECONNREFUSED | A remote host refuses connection. |
| EHOSTDOWN | A remote host doesn't work. |
| EHOSTUNREACH | A remote host is unreach. |
| ETIMEOUT | An operation time is exceeded. |
| ESCRIPTFAILED | Running of a given script is failed. |
| EOTHER | Other (unknown) error has appeared. |

## 8.5. Related articles

Network Service Browser

Network Service Monitor Configurator

# Chapter 9. Buttons the most often used in Web applications

## 9.1. The buttons meaning

There are the buttons, in the chart below, that occur the most often in Web applications. Their fucnction in particular applications is similar and even identical sometimes. Some of the buttons can have additional functions, that were described during descriptions of the particular applications.

**Table 9.1. The buttons the most often used in Web applications**

| Button | Description |
|---|---|
|  | It allows you to recover to a previous page. |
|  | It deletes an item i.e.: it closes a case, sets an event in a passive state etc. |
|  | It allows you to get to an edition of a given item. |
|  | It confirms an operation and makes it (i.e.: generating of a report using selected criterions). |
|  | It allows you to get to a detailed view. |
|  | It allows you to get to a higher level of item hierarchy. |
|  | It opens a new window with data which are prepared for a printout. |
|  | It allows you to get to a presentation of the graph with data for a given item (Collection Browser). |
|  | It reloads a page view. |
|  | It accepts changed values as current one. |

| Button | Description |
|---|---|
|  | It allows you to get to a report for a given item (Node Reporter). |
|  | It lets you get to a Trap browser for a given item (Trap Browser). |
|  | It lets you get to a report browser (about cases) for a given item (Recorded Operation Browser). |
|  | It saves changes, that were done by a user. |

# Chapter 10. Network Service Monitor Configurator

## 10.1. General

**Network Service Monitor Configurator** is a Web application and it is a part of **Service Monitor**. The application allows to control a work of Network Service Monitor providing a host specification and TCP ports, that will be monitored, to it.

## 10.2. Description

### 10.2.1. Default view of the application


**Network Service Monitor Configurator** is an application accessible in `Configuration` tab. A main view of the application presents a list of all defined hosts, that TCP ports are monitored. Links in `Description` and `Host` columns allow to edit a selected host. Meaning of particular columns is presented in the table below:

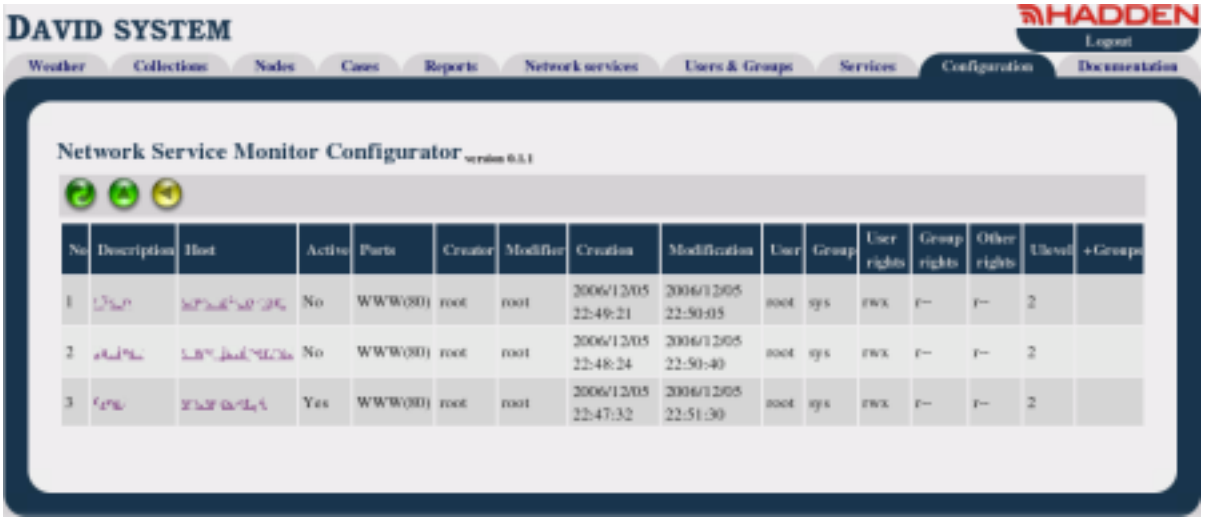
**Table 10.1. Meaning of columns of monitored host list**

| Column | Meaning |
|---|---|
| `Description` | Short description of a host. |
| `Host` | DNS host name or its address IPv4 or IPv6. |
| `Active` | It shows if host can be monitored now. |
| `Source address` | Source address of the connections for all its ports if not empty. |
| `Ports` | A list of TCP ports monitored on this host. |
| `Creator` | A creator of entry. |
| `Modifier` | The last modifier of entry. |
| `Creation` | Entry creation time. |

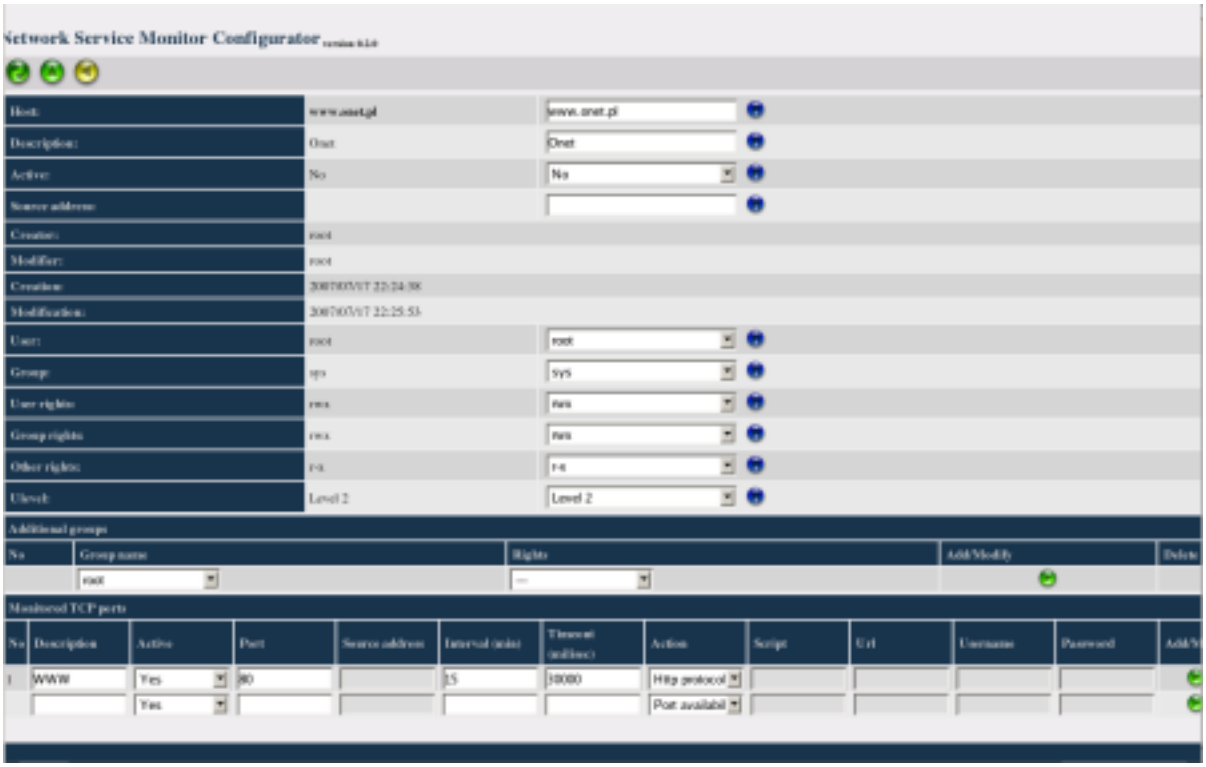
| Column | Meaning |
|---|---|
| Modification | Entry modification time. |
| User | A user of entry. |
| Group | A group of entry. |
| User rights | User rights of entry. |
| Group rights | Group rights of entry. |
| Other rights | Rights to write for other users. |
| Ulevel | A minimum level of a user, that can modify entry. |
| +Groups | Additionl groups of entry. |
| Add/Delete | Buttons, that can add or delete an existed entry. |

A user of 0 level can add a new item to the list of hosts, that will be monitored. A basic configuration of the new entry, you can do in edition fields, that are placed below a list of hosts.

A user of higher level then 0 only can see existed hosts according to his access rights and he can't add a new item to the list.




## 10.2.2. An edition of entry describing a monitored host



A top part of the view shows fields specifing a selected item of the list. They are compatible with columns of a default view of the application. Below, there are a list of additional user groups to which entry defining host belongs to. At the bottom a list of monitored TCP ports of a given host is placed. Meaning of columns of TCP port list is presented in the table below:

**Table 10.2. Meaning of columns of TCP port list**

| Column | Meaning |
|---|---|
| `Description` | Short description of a port. |
| `Active` | It tells if the port is active or not. |
| `Port` | TCP port number. |
| `Source address` | Source address of the connections for that port if not empty. |
| `Interval (min)` | Time interval describing every how minutes a given port will be chacked. |
| `Timeout (milisec)` | A maximum period of time in miliseconds, that can have a whole, single operation monitoring a given port. After that time, the operation will be interrupted. |
| `Action` | A kind of action of port checking. The parameter can have the following values:<br>• `Port availability only` - checking only availability of connecting with a port; |

| Column | Meaning |
|---|---|
| | • `Custom script` - a selected script running;<br>• `Smtp protocol` - built into SMTP service;<br>• `Http protocol` - built into HTTP service;<br>• `Ftp protocol` - built into FTP service;<br>• `Http/Ftp proxy` - built into HTTP service with proxy option. |
| `Script` | A path of the script if `Action` has a value `Custom script`. |
| `Url` | Optional parameter specifing a file used only to read if `Action` has a value `Http protocol` or `Http/Ftp proxy`. |
| `Username` | Optional user name only is used if `Action` has a value `Http protocol` or `Http/Ftp proxy`. |
| `Password` | Optional password is used if `Action` has a value `Http protocol` or `Http/Ftp proxy` and `Username` field is not empty. |
| `Add/Modify` | A button, that allows to add a new entry or edit an existed one. |
| `Delete` | A button deleting an existed entry. |

# 10.3. Related articles

Network Service Monitor (dnsmd)

Network Service Browser

# Chapter 11. Network Service Browser

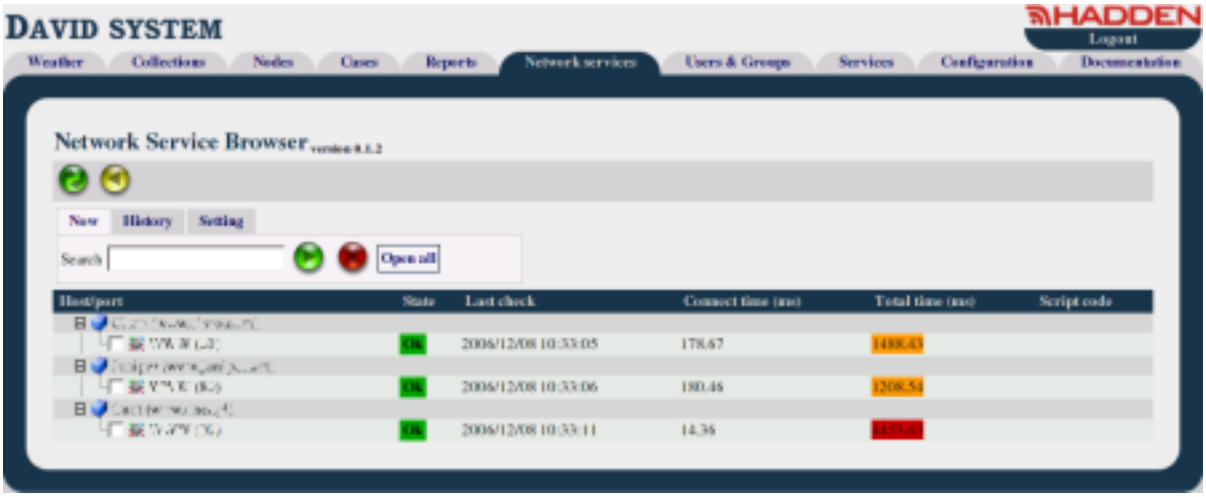
## 11.1. General

**Network Service Browser** is a Web application and it is a part of **Service Monitor**. It allows to browse reports concerning availability of network services on the base of data collected by Network Service Monitor.

## 11.2. Description

**Network Service Browser** application is available in `Network Services` tab. The top part of the application is a toolbar characteristic of all Web applications. The view, below the toolbar, depends on a current mode of the application. **Network Service Browser** works in two modes: as a borwser of current (last) data and as a browser of historical data, the data in selected period of time.

### 11.2.1. Current data mode




If you want to show current, last data of moniotred network services, you should choose `Now` tab of the panel below the toolbar. Below the panel with tabs, a list of hosts and their monitored TCP ports is placed. The tab makes possible a filtration of the presented list of results. `Open all` button pulls the whole list presenting ports of all visible hosts.

The list is divided into columns presenting monitoring results of particular ports. Meaning of columns shows the table below:

**Table 11.1. Meaning of columns of results list of monitored TCP ports**

| Column | Meaning |
|---|---|
| `Host/port` | A short description of the monitored host or TCP port. |

| Column | Meaning |
|---|---|
| State | A result of port monitoring: OK or an error description. |
| Last check | A date of the last port monitoring. |
| Connect time (ms) | Time from a moment of running of the connection. |
| Total time (ms) | Total time of monitoring of the port. |
| Script code | Return code of external script, if it was run for a given port. |

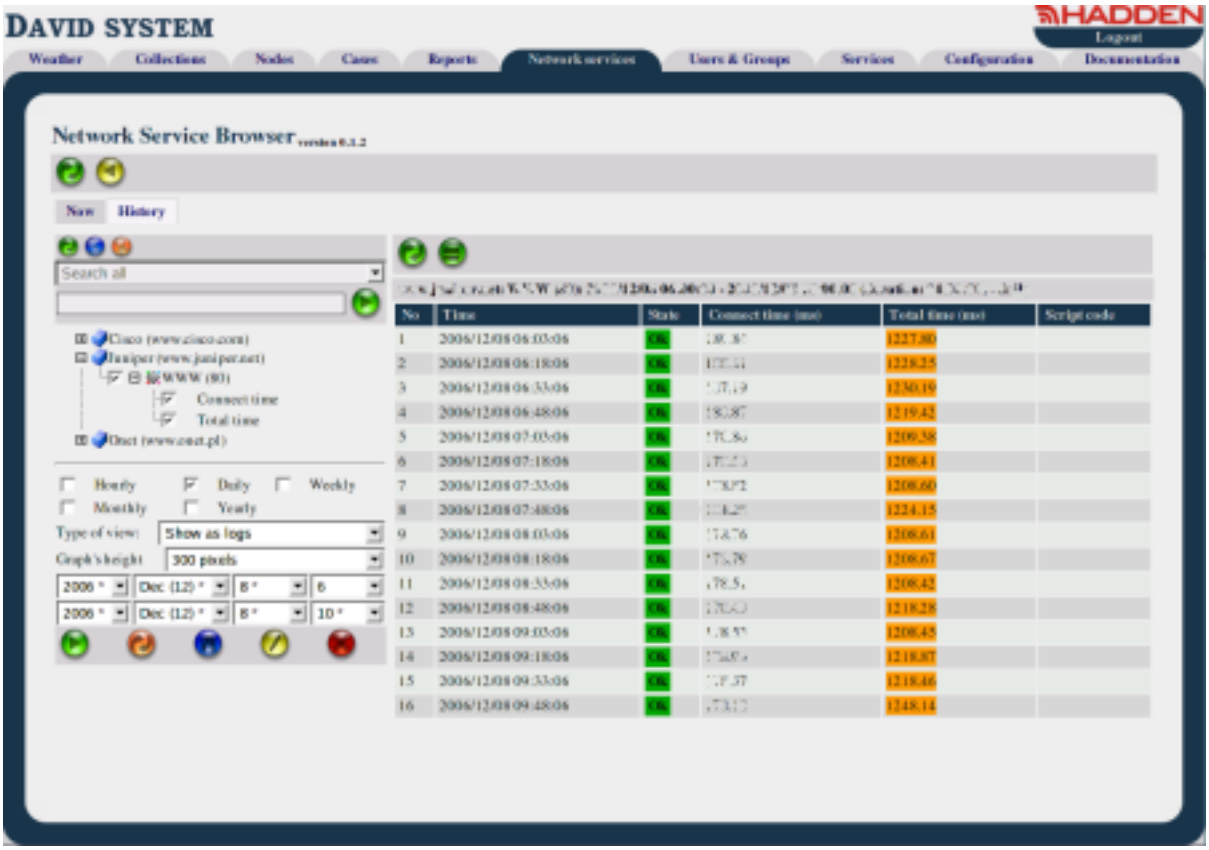
## 11.2.2. Historical data mode


If you want to show historical data of monitored network services, you should to choose History tab of the panel below the toolbar. A left panel of the view includes a list of hosts with monitored TCP ports, and the right one shows data of selected services.

The left panel, besides a pulled list of services, includes elements allowing to choose a way of data presenting, their time range etc. There is a searching parameter of service list describing its range, on the top of the panel. It is also accesible by Setting tab. Below the searching option, an edited filed is palced, that allows to give a searching string, and after them there is the service list according to searching criterions. Below the list elements are placed, that allow to select a graph type (hourly, daily, weekly, monthly, yearly), if Type of view option has a value Show as graphs. Then the data are shown as graphs of connection time and as graphs of duration of the operation. If option Type

of view is set on Show as logs, the data are presented as a list of logs. In the case of presenting data as graphs, height of graph accesible as Graph's height option is additional parameter.

Common parameters for two type of presenting data is their time range. If you want to set time range, you can use two identical set of options, that are placed one set above the second one, below Graph's height parameter. The top options show beginning, while the bottom options show the end of range. Below there is a toolbar with buttons that enable to save data selected to the report.




When data are shown as logs, the list is divided into columns similarly to view of Now tab. Below table described meaning of particular columns:

**Table 11.2. Meaning of columns of historical results list of monitored TCP ports**

| Column | Meaning |
|---|---|
| No | The next number of entry for a given service. |
| Time | Time of service measurement. |
| State | A result of port monitoring: OK or en error description. |
| Connect time (ms) | Time from the moment of running of the connection. |
| Total time (ms) | Total time of monitoring of the service. |
| Script code | Return code of external script, if it was run for a given service. |

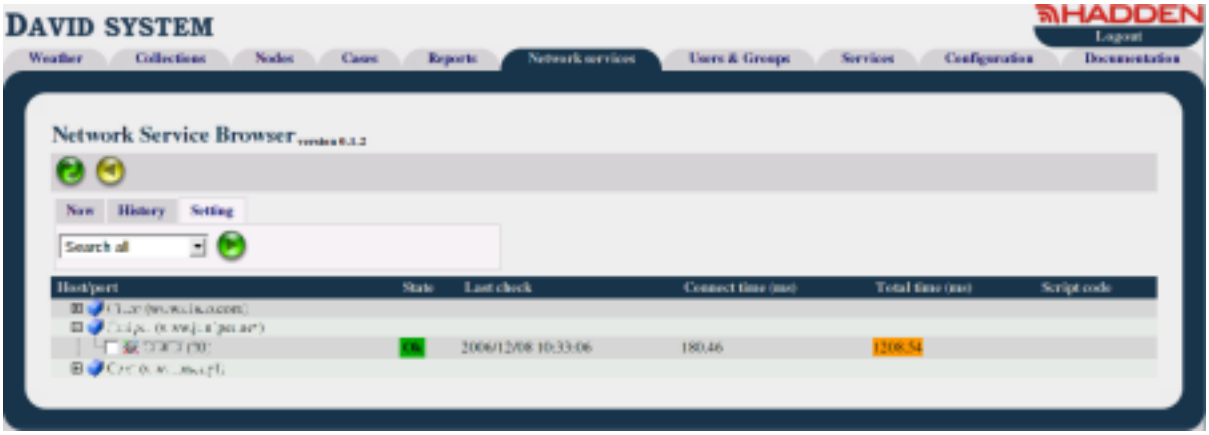
### 11.2.2.1. Buttons of the index panel toolbar

Meaning of buttons placed on the bottom of the monitored service list is as follows:

**Table 11.3. Network Service Browser - description of buttons on the index panel**

| Button | Description |
|---|---|
|  | Add selected services to previously saved ones and show them all. |
|  | Save only currently visible and checked services and show them. |
|  | Save currently visible and checked services. |
|  | Add checked services to previously saved and show a list of all saved services. |
|  | Delete all saved selections of services. |

## 11.2.3. Changing working parameters of the application




Only working parameter of the application, that can be set by `Setting` tab, is a way of searching of hosts list and their monitored services. The option can have one of three possibilities:

- Search hosts only
- Search ports only

- Search all

# 11.3. Related articles

Network Service Monitor (dnsmd)

Network Service Monitor Configurator