Daemon Manager 1.0.0

Technical documentation

Katarzyna Wladyszewska, Hadden Sp.J.

Daemon Manager 1.0.0: 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`** **`{`** **`...`** **`}`**

Chapter 2. General information about David system

2.1. General

David systemis 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.

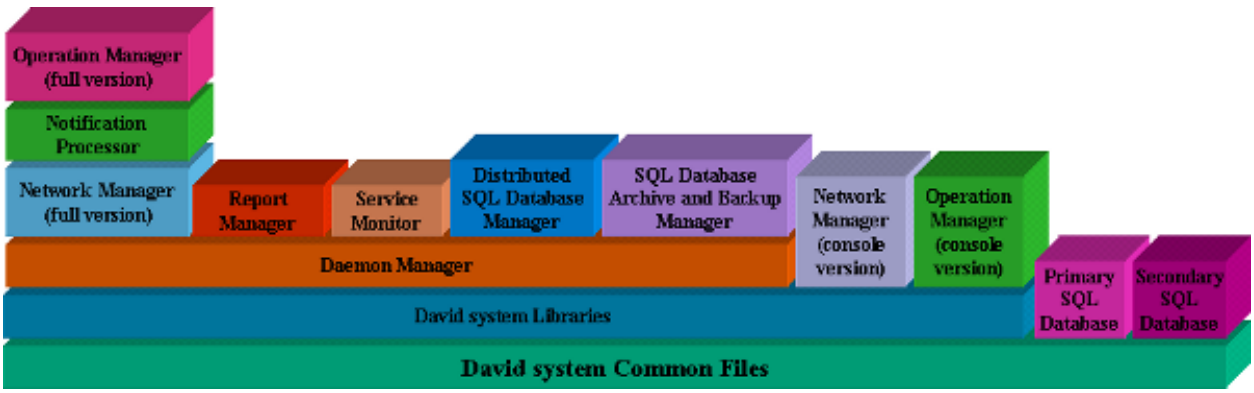
2.2. David system architecture

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. Daemon Manager requirements

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

- installed interpreter PHP 4 (http://www.php.net) (with a support for MySQL and SNMP) as a CLI;
- installed, compatible version of **David system Libraries**.

Chapter 6. Installation

6.1. Installation from RPM package

You must be `root` to install the product. Following steps must be taken in order to install the product:

- Install the product:

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

- Configure `syslog` daemon. Add the following line to the file /etc/syslog.conf:

```
local6.*      /var/log/davidlog
```

  Restart `syslog` daemon:

```
/etc/init.d/syslog restart
```

- Install other products of David system.
- In order to get installated daemons running, type:

```
/etc/init.d/david start
```

  That command runs Service Manager of David system.

6.2. Installation from the script

You must be `root` to install the product. The typical installation requires following steps:

- Uncompress and unpack the archive:

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

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

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

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

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

- Run the installation script:

```
./install
```

- Configure `syslog` daemon. Add the following line to the file /etc/syslog.conf:

```
local6.*      /var/log/davidlog
```

Restart `syslog` daemon:

```
/etc/init.d/syslog restart
```

- Install other products of David system.

- In order to get installated daemons running, type:

```
/etc/init.d/david start
```

That command runs Service Manager of David system.

Chapter 7. General

7.1. Functionality

Daemon Manager makes possible:

- Running and stopping daemons included in David system products;
- Monitoring of run daemons and running them again if unexpected terminated;
- Notifying the administrator about discovered unexpected terminations of daemons.

7.2. Description

The essential assignment of **Daemon Manager** is running and stopping daemons of other installed products of David system. The commands are executed by the administrator or run automatically during startup according to the configuration file.

Moreover, the other goal of **Daemon Manager** is keep the daemons running which means to run daemons found unexpectedly terminated.

This product also sends notifications in case discovering unexpectedly terminated daemons that should work.

7.3. Related articles

Client of Service Manager (dsc)

Service Manager (dsmd)

Chapter 8. Client of Service Manager (dsc)

8.1. General

dsc is **Client of Service Manager** and it is a part of **Daemon Manager**. It make possible to communicate with daemon dsmd from the shell level. The Client allows to list all configured daemons that are managed by dsmd and can run or stop a specified daemon.

8.2. Synopsis

dsc can be run with the following options:: [*-l,--log-facility log_facility*] [*-L,--log-level log_level*] [*--socket-file filename*] [*-r,--run daemon*] [*-s,--stop daemon*] [*-c,--clean-output*] [*-v,--version*] [*-h,--help*]

8.3. Options

Table 8.1. dsc options

Option name	Description
-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).
--socket-file filename	Specify a socket file on which the server is listing (default: /tmp/dsmng.socket).
-r,--run daemon	Run a specified daemon.
-s,--stop daemon	Stop a specified daemon.
-c,--clean-output	Print data without descriptions.
-v,--version	Display version number on stderr and exit.
-h,--help	Display this help and exit.

8.4. Description

Running the program without -r,--run and -s,--stop options causes writing all daemons configured in a configured file of dsmd daemon with information about them..

Running the program with -r,--run option and with the daemon name (a name of the daemon binary file) as its argument causes sending to dsmd daemon a request about running this module.

Running the program with -s,--stop option and with the daemon name (a name of the daemon binary file) as its argument causes sending to dsmd daemon a request about stopping this module.

Running **dsc** module with -c,--clean-output option causes that each piece of information will be written in separated line and without additional explanations.

8.5. Related articles

Service Manager (dsmd)

Chapter 9. Service Manager (dsmd)

9.1. General

dsmd is **Service Manager** and it is a part of **Daemon Manager**. It is a daemon process which works all the time the system is running and it takes care of all specified daemons of David system that they were run. The specification of the daemons managed by **dsmd** is made in its configuration file. During the daemon work its client dsc can change a part of the settings.

During its startup **dsmd** reads its configuration file `.dsmdrc`. The program expects to find it in a directory pointed by the variable *confdir* in the file `/etc/david-system.conf`. The daemon **dsmd** won't be run if it can't read its configuration from `.dsmdrc` file.

9.2. Synopsis

dsmd 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*] [*--out-queue-ttl seconds*] [*--max-clients number*] [*--program4fallen-daemons filename*] [*--program-running-period seconds*] [*-v,--version*] [*-h,--help*]

9.3. Options

Table 9.1. dsmd options

Option name	Description
-P,--pid-file filename	Write PID to a 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.
--out-queue-ttl seconds	Specify a maximum TTL for outgoing messages (default: 300).
--max-clients number	Specify maximum number of client that can be serviced in the same time (default: 5).
--program4fallen-daemons filename	Specify a program which will be run after a fact that a daemon isn't working.
--program-running-period seconds	Specify a minimum time (in seconds) between running of a given program for each of not-working daemons (default: 3600).
-v,--version	Display version number on stderr and exit.

Option name	Description
-h,--help	Display this help and exit.

9.4. Configuration file format

The configuration file consists of objects that type is specified by a keyword used inside their names and their specification is limited by braces `{}`. Only one type of objects is permitted and its keyword is `daemon`. A keyword that describes type of object must be a single word in a whole line. The same rule is applicable to braces that begin and end an object definition. An generic object definition is shown below:

```
daemon

{

...

}
```

Fields occurred between braces are exactly specified like an order of them. Each field from first six ones in an object definition takes a whole line and has got below form:

```
keyword=
```

The first field is `binfile` that must be followed by a name of the daemon binary file. The files are looked for in a directory */usr/bin/david-system*. The next field is `description` describing the daemon in human readable way. The next field is `state` that can accept two values:

```
stop

run
```

The `stop` value shows that the daemon can't be run automatically during the start of **dsmd**. All daemons with `run` value are run in the moment when the **dsmd** is starting.

The next field is `username` describing with whose user's rights the daemon will be run. In next two fields `stdout` and `stderr` you can specify file names that are standard out and standard out for errors of a specified daemon.

The optionally field is `argv` and it can be show many times. The fields specify the next running parameters of a given daemon. Each parameter must be show in a separated `argv` field.

9.5. Description

When **dsmd** daemon starts, it runs all daemons with `state=run` parameter in order of its entries in its configuration file. Next it checks all the time if all daemons, that should be run, are working. When **dsmd** daemon stops its work, it stops all daemons into reverse order to its entries in its configuration file.

If dsmd finds a not-working daemon which should work, it runs its again. It additionally runs a program specified as an argument of --program4fallen-daemons option. The program receives as an argument a daemon name, which doesn't work. For a given daemon the program can't be run more then a number of seconds that will be an argument of --program-running-period option.

A configuration of **dsmd** daemon can be changed by its client dsc. The client can ask for stopping of a working daemon or running of a not-working daemon. The other changes of the configuration are not possible during **dsmd** is working.

9.6. Related articles

Client of Service Manager (dsc)