Dr.WEB

Enterprise Security Suite

Appendices

Dr.Web Enterprise Security Suite
Version 13.0
Appendices
4/13/2026

For information on regional and international offices of Doctor Web, please visit the official website at https://company.drweb.com/contacts/.

Doctor Web

Doctor Web develops and distributes Dr.Web information security solutions that provide effective protection against malicious software and spam.

Doctor Web customers include home users around the world, government agencies, small businesses, and nationwide corporations.

Since 1992, Dr.Web anti-virus solutions have been known for their continuous excellence in malware detection and compliance with international information security standards.

The state certificates and awards received by Dr.Web solutions, as well as the worldwide use of our products, are the best evidence of exceptional trust in the company products.

We thank all our customers for their support and devotion to Dr.Web products!

Table of Contents

Chapter 1: Introduction

About the Manual

Documentation for the administrator of Dr.Web Enterprise Security Suite anti-virus network is intended to introduce general features of the software suite and provide detailed information on delivering comprehensive anti-virus protection for company computers using Dr.Web Enterprise Security Suite.

Documentation for the anti-virus network administrator contains the following parts:

1. **Installation Manual**

   The Installation Manual will be useful to a company manager who makes a decision to purchase and install a comprehensive anti-virus protection system.

   Installation Manual explains how to build an anti-virus network and install its main components.

2. **Administrator Manual**

   The Administrator Manual is meant for the *anti-virus network administrator,* i. e., an employee of the company who is responsible for anti-virus protection of computers (workstations and servers) of this network.

   The anti-virus network administrator should either have system administrator privileges or work closely with a local network administrator, be competent in anti-virus protection strategy, and have an in-depth knowledge of Dr.Web anti-virus packages for all operating systems that are used on the network.

3. **Appendices**

   The Appendices provide technical information describing configuration parameters for the anti-virus components and the syntax and values of instructions used to work with these modules.

4. **Anti-virus Network Quick Installation Guide**

   It provides brief information on the installation and initial configuration of anti-virus network components. For detailed information refer to the administrator documentation.

The documents listed above contain cross-references of the following two types between them:

- links to local documents. When downloading the documents to your computer, save them under their original names in the same folder.
- links to documents hosted online at https://download.drweb.com/doc/. You can use these links if local versions of the documents are unavailable.

In addition, the following manuals are provided:

1. **Station management manuals**

   They provide information about centralized configuration of anti-virus software on workstations, performed by the anti-virus network administrator using the Dr.Web Security Control Center.

2. **User Manuals**

   They provide information on how to configure Dr.Web anti-virus software directly on protected stations.

3. **Web API Manual**

   It provides technical details on the integration of Dr.Web Enterprise Security Suite with third-party software via Web API.

4. **Dr.Web Server Database Structure Manual**

   It describes the internal structure of the Dr.Web Server database and provides examples of its use.

All the listed manuals are also provided as part of Dr.Web Enterprise Security Suite and can be accessed via Dr.Web Security Control Center.

Before reading these documents, make sure you have the latest version of the corresponding manuals for your product version. The manuals are continuously updated and the latest version can always be found on the official website of Doctor Web at https://download.drweb.com/doc/.

Conventions and Abbreviations

Conventions

The following symbols and text conventions are used in this guide:

Convention	Comment
	An important note or instruction
	A warning about possible errors or important notes that require special attention
Anti-virus network	A new term or an emphasis on a term in descriptions
<IP-address>	Placeholders
Save	Names of buttons, windows, menu items and other program interface elements
CTRL	Names of keyboard keys

Convention	Comment
`C:\Windows\`	Names of files and folders, code examples
Appendix A	Cross-references to document chapters or internal hyperlinks to webpages

Abbreviations

The following abbreviations can be used in the manual without further interpretation:

- ACL—Access Control List,
- CDN—Content Delivery Network,
- DB, DBMS—Database, Database Management System,
- DFS—Distributed File System,
- DN—Distinguished Name,
- DNS—Domain Name System,
- Dr.Web GUS—Dr.Web Global Update System,
- FQDN—Fully Qualified Domain Name,
- GUI—Graphical User Interface, a GUI version of a program—a version using a GUI,
- LAN—Local Area Network,
- MIB—Management Information Base,
- MTU—Maximum Transmission Unit,
- NAP — Network Access Protection,
- OS—Operating System,
- TTL—Time To Live,
- UDS—UNIX Domain Socket.

Chapter 2: Appendices

Appendix A. The Description of the DBMS Settings. The Parameters of the DBMS Driver

Dr.Web Server database structure is available in the form of a separate manual of the same name. The document can be opened from the **Support** section in Dr.Web Security Control Center.

As a database for Dr.Web Server you can use the following variants:

- embedded DB;
- external DBMS.

Embedded DB

When setting up access to the embedded DB for storage and processing of data, use the parameters described in the table below.

Embedded DB parameters

Name	Default value	Description
DBFILE	database.sqlite	Path to the database file
CACHESIZE	2048	Database cache size in pages
PRECOMPILEDCACHE	1048576	Cache size of precompiled SQL operators (in bytes)
MMAPSIZE	• 10485760—for UNIX, • 0—for Windows	Maximum size of the database file (in bytes) that is allowed to be mapped into process address space at a time.
CHECKINTEGRITY	QUICK	Verify integrity of the database image at Dr.Web Server startup: • FULL—full check for any errors concerning UNIQUE, CHECK, and NOT NULL constraints, malformed records, missing pages or index inconsistencies; • QUICK—fast check option, with no regards to constraint errors or index inconsistencies; • NO—integrity check is disabled.
AUTOREPAIR	NO	Automatically restore corrupted database image at Dr.Web Server startup:

		• YES—the database image restoration is initiated every time Dr.Web Server starts, • NO—automatic restoration is disabled.
WAL	YES	Use Write-Ahead Logging: • YES—true, • NO—false.
WAL-MAX-PAGES	1000	Max number of "dirty" pages. When reached, the pages will been written to disk.
WAL-MAX-SECONDS	30	Max time in seconds that the page writing to disk is delayed for.
SYNCHRONOUS	FULL	Mode of synchronous logging of changes in the database to the disk: • FULL—fully synchronous logging to the disk, • NORMAL—synchronous logging of critical data, • OFF—asynchronous logging.

The SQLite3 DB is provided as the embedded DB.

External DBMS

The following database management systems may be used to manage the external database for Dr.Web Server:

- MySQL, MariaDB. The settings are given in A4. Using the MySQL DBMS.
- Oracle. The settings are given in A2. Setting Up the Database Driver for Oracle.
- PostgreSQL. The settings are given in A3. Using the PostgreSQL DBMS.

DBMS based on PostgreSQL (PostgreSQL Pro, Jatoba, and others) are supported.

- Microsoft SQL Server/Microsoft SQL Server Express. To access these DBMSs, an ODBC driver may be used (setup of the parameters of the ODBC driver for Windows is given in A1. Setting Up the ODBC driver).

Microsoft SQL Server 2008 and later is supported. Microsoft SQL Server 2014 and later is recommended to use.

Microsoft SQL Server Express DB is not recommended for an anti-virus network with a large number of stations (100 and more).

If the Microsoft SQL Server is used as an external DB for Dr.Web Server under a Unix-like OS, the proper operation via the ODBC with FreeTDS is not guaranteed.

If warnings or errors occur in Dr.Web Server interaction with Microsoft SQL Server DBMS via the ODBC, please make sure that you are using the latest available DBMS version for this edition.

Learn how to determine the update level on the following Microsoft Corporation page: https://learn.microsoft.com/en-US/troubleshoot/sql/releases/download-and-install-latest-updates.

To reduce the number of deadlocks when using Microsoft SQL Server DBMS with the default transaction isolation level (READ COMMITTED), it is recommended that you enable the READ_COMMITTED_SNAPSHOT option by running the following SQL command:

```
ALTER DATABASE <database_name>
SET READ_COMMITTED_SNAPSHOT ON;
```

The command above must be run in implicit transaction mode and with a single existing connection to the database.

Comparison Characteristics of Embedded DB and External DBMS

The embedded DB is intended for use in networks where about 400–600 stations are connected to Dr.Web Server. If the hardware configuration of the computer on which Dr.Web Server is installed and the load level of other executing tasks permit, between 1000 and 1500 stations can be connected.

Otherwise, you must use an external DB. Depending on the configuration and the load level of the computer running Dr.Web Server, the external DB may be used on the same or on a dedicated computer.

If you use an external DB in an anti-virus network with more than 10 000 stations, it is recommended to follow these minimal system and hardware requirements:

- 3 GHz processor CPU,
- at least 6 CPU cores,
- at least 4 GB RAM for Dr.Web Server and at least 8 GB RAM for the DB server,
- Unix-like OS.

When choosing between the embedded and external database, take into account the following parameters of the database types:

- In large anti-virus networks (of over 400–600 stations), it is recommended to use an external DB, which is more fault-resistant than the embedded DB.
- When using the embedded DB, you do not need to install third party software components. It is recommended mainly for the typical use of databases.

- The embedded database does not require DBMS administration skills and is a good choice for anti-virus networks of small and medium sizes.
- You may use an external database in case it will be necessary to work through a DBMS and access the DB directly. To facilitate access, standard APIs may be used, such as OLE DB, ADO.NET or ODBC.

A1. Setting Up the ODBC driver

When setting up access to an external DBMS for storage and processing of data, use the parameters described in the table below (specific values are given for example).

Parameters for ODBC connection

Name	Value	Description
DSN	drwcs	Data set name
USER	drwcs	User name
PASS	fUqRbrmlvI	Password
TRANSACTION	DEFAULT	Possible values of the TRANSACTION parameter: • SERIALIZABLE • READ_UNCOMMITTED • READ_COMMITTED • REPEATABLE_READ • DEFAULT The DEFAULT value means "use default of the SQL server". For more information on transaction isolation refer to the documentation on the corresponding DBMS.

To exclude encoding problems, you must disable the following parameters of the ODBC driver:

- **Use regional settings when outputting currency, numbers, dates and times**—may cause errors during numerical parameter formatting.
- **Perform translation for character data**—may cause incorrect characters to be displayed in Dr.Web Security Control Center for parameters that came from the DB. This parameter makes character display dependent on the language parameter for programs which do not use Unicode.

When creating a new database in the Microsoft SQL DBMS, you must specify the case-sensitive (has the _CS suffix) and accent-sensitive (has the _AS suffix) collation.

The following characters are undesirable: space, "{", ";" and "}", for more details see https://learn.microsoft.com/en-us/sql/odbc/reference/syntax/sqldriverconnect-function?redirectedfrom=MSDN&view=sql-server-ver15.

The database is initially created on the SQL server with the above mentioned parameters.

It is also necessary to set the ODBC driver parameters on the computer where Dr.Web Server is installed.

Information on ODBC driver setup under a Unix-like OS can be found at https://www.unixodbc.org/ in the **Manuals** section.

ODBC Driver Setup for Windows OS

To configure ODBC driver parameters

1. On the Windows OS **Control Panel**, select **Administrative tools**; then double-click **Data Sources (ODBC)** in the pop-up window. The **ODBC Data Source Administrator** window will open. Go to the **System DSN** tab.
2. Click **Add**. A window for selecting a driver will open.
3. Select the corresponding ODBC driver for this DB on the list and click **Finish**. The first window for setting up access to the DB server will open.

If an external DBMS is used, it is necessary to install the latest version of the ODBC driver delivered with this DBMS. It is strongly recommended not to use the ODBC driver supplied with Windows OS (SQL Server Native Client), except for databases supplied by Microsoft without an ODBC driver.

If you use MySQL, install and use the actual version of the ODBC driver from from the Microsoft website.

4. Specify access parameters to the data source, the same as the parameters in the settings of Dr.Web Server. If the DB server is not installed on the same computer as Dr.Web Server, in the **Server** field, specify the IP address or name of the DB server. Click **Next**.
5. Select the **With SQL Server authentication** option and specify the user credentials to access the DB. Click **Next**.

Dr.Web Server does not support Windows NT authentication.

6. In the **Change the default database to** drop-down list, select the database which is used by Dr.Web Server. The Dr.Web Server database name must be specified, not the **Default** value.

Make sure that the following flags are set: **Use ANSI quoted identifiers** and **Use ANSI nulls, paddings and warnings**. Click **Next**.

If the ODBC driver settings allow you to change the language of SQL server system messages, select **English**.

7. When you complete the configuration, click **Finish**. A window with the summary of the specified parameters wil open.
8. To test the specified settings, click **Test Data Source**. After the notification of a successful test, click **OK**.

A2. Setting Up the Database Driver for Oracle

General Description

The Oracle Database (or Oracle DBMS) is an object-relational DBMS. Oracle may be used as an external DB for Dr.Web Enterprise Security Suite.

The Dr.Web Server may use the Oracle DBMS as an external database on all platforms except FreeBSD (see Installation and supported versions).

To use the Oracle DBMS

1. Install an instance of Oracle DB and set up the `AL32UTF8` encoding. Also you may use existence instance which is configured to use the `AL32UTF8` encoding.
2. Set up the database driver to use the respective external database. You can do this in configuration file or via Dr.Web Security Control Center: **Dr.Web Server configuration**, **Database** tab.

Connection to the Oracle database as the SYS and SYSTEM users, and also with the SYSDBA and SYSOPER privileges is forbidden.

Installation and Supported Versions

To use Oracle as en external DB, you must install the instance of the Oracle DB and set up `AL32UTF8` (`CHARACTER SET AL32UTF8 / NATIONAL CHARACTER SET AL16UTF16`) encoding. This can be done in one of the following ways:

1. Using an Oracle installer (use an external mode of instance installation and configuration).
2. Using the `CREATE DATABASE` SQL command.

To use Dr.Web Server with a connection to an Oracle database on Unix-based operating systems you must run the following command :

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/drwcs/lib/
:/opt/drwcs/lib/plugins/:/opt/drwcs/lib/database/
:/opt/drwcs/lib/database/oracle
```

For more information on creating and configuring Oracle instances, see Oracle documentation.

In case of using a different encoding, national symbols may be displayed incorrectly.

A client to access the database (Oracle Instant Client) is included in the installation package of Dr.Web Enterprise Security Suite.

Platforms supported by the Oracle DBMS are listed on the website of the vendor.

Platforms supported by the Oracle Client are listed on the website of the vendor.

Dr.Web Enterprise Security Suite supports the Oracle DBMS of version 11 and later.

Also, please note the system requirements for Dr.Web Server when operating with the Oracle external database (see the **Installation Manual**, System Requirements (read online).

To use Oracle DB, the `libaio` (Linux kernel AIO access) library is required.

Parameters

To adjust access to the Oracle DBMS, use the parameters described in Table below.

Parameters of the Oracle DBMS

Parameter	Description
`drworacle`	Driver name
`User`	Database user name (obligatory)
`Password`	User password (obligatory)
`ConnectionString`	Database connection string (obligatory)

The format of the connection string to the Oracle DBMS is as follows:

//<host>:<port>/<service name>

where:

- <*host*>—IP address or name of the Oracle server;
- <*port*>—port that the server is 'listening';
- <*service name*>—name of the DB to connect to.

For Example:

```
//myserver111:1521/bjava21
```

where:

- `myserver111`—name of the Oracle server.
- `1521`—port 'listening' to the server.
- `bjava21`—name of the DB to connect to.

Oracle DBMS Driver Configuration

If you deploy Oracle, it is necessary to change the definition and the settings of the database driver by one of the following ways:

- In the Control Center: **Administration** item in the main menu → **Dr.Web Server configuration** item in the control menu → **Database** tab → select in the **Database** drop-down list, the **Oracle** type, and set parameters according to the format listed below.
- In the Dr.Web Server configuration file.

A3. Using the PostgreSQL DBMS

General Description

PostgreSQL is an object-relational DBMS distributed as a freeware unlike such commercial DBMS as Oracle Database, Microsoft SQL Server, etc. The PostgreSQL DBMS may be used to arrange an external DB for Dr.Web Enterprise Security Suite in large anti-virus networks.

To use PostgreSQL as an external database

1. Install the PostgreSQL or Postgres Pro server.
2. Set up Dr.Web Server to use the respective external database. You can do this in configuration file or via Dr.Web Security Control Center: in the **Dr.Web Server configuration** menu, the **Database** tab.

To connect to the PostgreSQL DB you can use only trust, password and MD5 authorization.

Installation and Supported Versions

1. Download the latest available version of this PostgreSQL free product (the PostgreSQL server), otherwise do not use the version earlier than 8.4 or 11.4.1 for Postgres Pro.
2. Create the PostgreSQL database by one of the following ways:
   a) Using the `pgAdmin` graphical interface.
   b) Using the `CREATE DATABASE` SQL command.

Database must be created in the UTF8 encoding.

For more information about conversion to the external database see p. Changing the Database Type.

Also, please note the system requirements for Dr.Web Server when operating with the PostgreSQL external database (see the **Installation Manual**, System Requirements (read online).

Parameters

When setting access to PostgreSQL, use parameters described in the table below.

Parameters of the PostgreSQL DBMS

Name	Default value	Description
`host`	*<UNIX domain socket>*	PostgreSQL server host
`port`		PostgreSQL server port or name extension of the socket file
`dbname`	`drwcs`	Database name
`user`	`drwcs`	User name
`password`	`drwcs`	Password
`options`		Debug/trace options for sending to Dr.Web Server
`requiressl`		• 1 instructs to request a SSL connection • 0 does not instruct to make the request
`temp_tablespaces`		Name space for temporary tables

Name	Default value	Description
`default_transaction_isolation`		Transaction isolation mode (see PostgreSQL documentation)

While you may specify any of the transaction isolation modes supported by PostgreSQL, correct operation of Dr.Web Server in **Repeatable read** mode is not guaranteed.

More information can be found at https://www.postgresql.org/docs/.

Dr.Web Server and PosrtgreSQL DB Interaction via the UDS

If Dr.Web Server and the PostgreSQL DB are installed on the same computer, their interaction can be set via the UDS (UNIX domain socket).

To set interaction via the UDS

1. In the `postgresql.conf` PostgreSQL configuration file, specify the following directory for the UDS:

   ```
   unix_socket_directory = '/var/run/postgresql'
   ```

2. Restart the PostgreSQL.

Configuring the PostgreSQL Database

To increase performance during interaction with the PostgreSQL database, it is recommended to configure it according to the information from the official documentation on the database.

If you use a large database and dispose the appropriate computing resources, it is recommended to configure the following parameters in the `postgresql.conf` configuration file:

Minimal configuration:

```
shared_buffers = 256MB

work_mem = 16MB
```

Extended configuration:

```
shared_buffers = 1GB

work_mem = 32MB

fsync = off
```

```
synchronous_commit = off

wal_sync_method = fdatasync

commit_delay = 1000

max_locks_per_transaction = 256

max_pred_locks_per_transaction = 256
```

The `fsync = off` parameter significantly increases performance but may cause the complete loss of data in case of power failure or system crash. It is recommend to disable the `fsync` parameter only if you have a backup of the database for its full recovery.

Configuration of the `max_locks_per_transaction` parameter can be useful to ensure smooth operation at a mass appeal to the database tables, in particular, when upgrading the database to a new version.

A4. Using the MySQL DBMS

General Description

MySQL—cross-platform relational databases management system. MySQL DBMS may be used as an external DB for Dr.Web Enterprise Security Suite.

To use MySQL as an external database

1. Install the MySQL server.
2. Set up Dr.Web Server to use the respective external database. You can do this in configuration file or via Dr.Web Security Control Center: in the **Dr.Web Server configuration** menu, the **Database** tab.

Installation and Supported Versions

Dr.Web Enterprise Security Suite supports the following versions of MySQL DBMS:

- MySQL—all versions starting from 8.0.12,

Make sure that for user connecting to MySQL version 8.X the athentication type is `caching_sha2_password`. This parameter is set during the installation and becomes the default one for every user, or it is manually configured for every user.

- MariaDB—all versions starting from 10.5.

After DBMS installation, before creating a new database, it is necessary to specify the following settings in its configuration file (see your DBMS documentation for more details):

For MySQL of 8.X versions:

```
[mysqld]

innodb_file_per_table = true

max_allowed_packet = 64M
```

Parameters

To adjust access to the MySQL DBMS, use the parameters described in Table below.

Parameters of the MySQL DBMS

Name	Default value	Description
HOST	localhost	• Database server address—when connecting to database via TCP/IP. • Path to UNIX socket file—when using UDS. If not set, Dr.Web Server tries to locate the file in one of standard mysqld directories.
PORT	3306	• Connection port number—when connecting to database via TCP/IP. • UNIX socket file name—when using UDS.
DBNAME		Database name
USER		Registration name of database user
PASSWORD	QUICK	Database user password
PRECOMPILEDCACHE	1048576	Cache size of precompiled sql operators (in bytes)
SSL	NO	Allow SSL connections only: • YES—connect to database only if SSL protocol is used, • NO—SSL protocol is not necessary to connect to database.

Appendix B. Authentication of Administrators

General information on authentication of administrators at Dr.Web Server is described in **Administrator Manual**, p. Authentication of Administrators (read online).

B1. Active Directory Authentication

Only enabling of using authentication method and the order in authenticators list are configured: in the `<enabled/>` and `<order/>` tags of the `auth-ads.conf` configuration file.

Operation principle:

1. Administrator specifies username and password in one of the following formats:
   - `username`,
   - `domain\username`,
   - `username@domain`,
   - user's LDAP DN.
2. Dr.Web Server registers with these name and password at the default domain controller (or at the domain controller which specified in the username).
3. If registration failed, transition to the next authentication mechanism is performed.
4. LDAP DN of registered user is determined.
5. For the object with determined DN, the `DrWebAdmin` attribute is read. If it has `FALSE` value, authentication is admitted failed and transition to the next authentication mechanism is performed.
6. If any of attributes are not defined at this stage, they are searched in groups to which the user is included to. For each group, its parental groups are checked (search strategy—inward).

If any error occurs, transition to the next authentication mechanism is performed.

The `drweb-modify-ad-schema-<package_version>-<build>-<OS_version>.exe` utility (is included to the Dr.Web Server distribution kit) creates in Active Directory the `DrWebEnterpriseUser` new object class and defines new attributes for this class.

Attributes have the following OID:

```
DrWeb_enterprise_OID "1.3.6.1.4.1" // iso.org.dod.internet.private.enterprise
DrWeb_DrWeb_OID DrWeb_enterprise_OID ".29690" // DrWeb
DrWeb_EnterpriseSuite_OID DrWeb_DrWeb_OID ".1" // EnterpriseSuite
DrWeb_Alerts_OID DrWeb_EnterpriseSuite_OID ".1" // Alerts
```

```
DrWeb_Vars_OID DrWeb_EnterpriseSuite_OID ".2" // Vars
DrWeb_AdminAttrs_OID DrWeb_EnterpriseSuite_OID ".3" // AdminAttrs

// 1.3.6.1.4.1.29690.1.3.1 (AKA
iso.org.dod.internet.private.enterprise.DrWeb.EnterpriseSuite.AdminAttrs.Admin)

DrWeb_Admin_OID DrWeb_AdminAttrs_OID ".1" // R/W admin
DrWeb_AdminReadOnly_OID DrWeb_AdminAttrs_OID ".2" // R/O admin
DrWeb_AdminGroupOnly_OID DrWeb_AdminAttrs_OID ".3" // Group admin
DrWeb_AdminGroup_OID DrWeb_AdminAttrs_OID ".4" // Admin's group
DrWeb_Admin_AttrName "DrWebAdmin"
DrWeb_AdminReadOnly_AttrName "DrWebAdminReadOnly"
DrWeb_AdminGroupOnly_AttrName "DrWebAdminGroupOnly"
DrWeb_AdminGroup_AttrName "DrWebAdminGroup"
```

Editing settings of Active Directory users is implemented manually at the Active Directory server (see **Administrator Manual**, p. Authentication of Administrators (read online ↗).

Assigning permissions to administrators performs according to the general principle of inheriting in the hierarchical structure of groups in which administrator is included.

B2. LDAP Authentication

Settings are stored in the `auth-ldap.conf` configuration file.

General tags of the configuration file:

- `<enabled/>` and `<order/>`—similar to the Active Directory.
- `<server/>` specifies the LDAP server address. Multiple `<server/>` tags with different LDAP server addresses can be added, which would make a list of servers to use for authentication. A main server that is assumed to take the major load should come first, while the remaining addresses of any backup servers should come after. When administrator connects, the first available LDAP server is used. If authentication fails, it will be retried on the next server and so on, following the order in which LDAP server addresses are listed in the configuration file.
- `<user-dn/>` defines rules for translation of name to the DN (Distinguished Name) using DOS-like masks.

  In the `<user-dn/>` tag, the following wildcard characters are allowed:

  - `*` replaces sequence of any characters, except `. , = @ \` and spaces;
  - `#` replaces sequence of any characters.
- `<user-dn-expr/>` defines rules for translation of name to the DN using regular expressions.

  For example, the same rule in different variants:

  ```
  <user-dn user="*@example.com" dn="CN=\1,DC=example,DC=com"/>
  <user-dn-expr user="(.*)@example.com" dn="CN=\1,DC=example,DC=com"/>
  ```

  `\1` .. `\9` defined the substitution place for values of the *, # or expression in brackets at the template.

  According to this principle, if the user name is specified as `login@example.com`, after translation you will get DN: "`CN=login,DC=example,DC=com`".

- <user-dn-extension-enabled/> allows the ldap-user-dn-translate.ds (from the extensions folder) Lua-script execution for translation usernames to DN. This script runs after attempts of using the user-dn, user-dn-expr rules, if appropriate rule is not found. Script has one parameter—specified username. Script returns the string that contains DN or nothing. If appropriate rule is not found and script is disabled or returns nothing, specified username is used as it is.
- Attributes of LDAP object for DN determined as a result of translation and their possible values can be defined by tags (default values are presented):

```
<!-- DrWebAdmin attribute equivalent (OID 1.3.6.1.4.1.29690.1.3.1) -->
<admin-attribute-name value="DrWebAdmin" true-value="^TRUE$" false-value="^FALSE$"/>
```

  As a values of true-value/false-value parameters, regular expressions are specified.
- If undefined values of administrators attributes are present, and the <group-reference-attribute-name value="memberOf"/> tag is set in the configuration file, the value of the memberOf attribute is considered as the list of DN groups, to which this administrator is included, and the search of needed attributes is performed in this groups as for the Active Directory.

B3. LDAP/AD Authentication

Configuration File

Settings are stored in the auth-ldap-rfc4515.conf configuration file.

Configuration files with typical settings are also provided:

- auth-ldap-rfc4515-check-group.conf—configuration file template for administrators external authorization via LDAP using the simplified scheme with verification of belonging to an Active Directory group.
- auth-ldap-rfc4515-check-group-novar.conf—configuration file template for administrators external authorization via LDAP using the simplified scheme with verification of belonging to an Active Directory group and using variables.
- auth-ldap-rfc4515-simple-login.conf—configuration file template for administrators external authorization via LDAP using the simplified scheme.

General tags of the auth-ldap-rfc4515.conf configuration file:

- <server />—LDAP server definition.

Attribute	Description	Default value
base-dn	DN of an object entry relative to which the search is to be performed.	The rootDomainNamingContext attribute value of the Root DSE object
cacertfile	Root certificates files (UNIX only).	–

Attribute	Description	Default value
host	LDAP server address.	• Domain controller for Dr.Web Server running Windows OS. • 127.0.0.1 for Dr.Web Server running a Unix-like OS. • Multiple <server /> tags with different LDAP server addresses can be added. The address of the main server that is supposed to take the main load should be specified first. If the server is unavailable, an attempt to authenticate on the next server will be made, and then in the specified sequence. If the server is available, the account search will be performed only on this server. Regardless of the authentication result in this case, the connection to the following servers will not be made.
scope	Search scope. Allowed values: • sub-tree—whole sub-tree below the base DN • one-level—direct descendants of the base DN • base—base DN.	sub-tree
tls	Use STARTTLS to connect to LDAP.	no
ssl	Use LDAPS to connect to LDAP.	no

- <set />—variables set by LDAP search.

Attribute	Description	Default value
attribute	Attribute name the value of which is assigned to a variable. Cannot be absent.	–
filter	RFC4515 search filter in LDAP.	–
scope	Search scope. Allowed values: • sub-tree—whole sub-tree below the base DN • one-level—direct descendants of the base DN • base—base DN.	sub-tree
search	DN of an object entry relative to which the search is to be performed.	If absent, the base-dn of the <server /> tag is used.

Attribute	Description	Default value
variable	Variable name. Must starts with the letter and contains letters and digits only. Cannot be absent.	–

For correct authentication, all groups for which user membership in groups is configured in the Control Center (**Administration → Authentication → LDAP/AD authentication settings → User membership**), i.e. for which there are corresponding entries in the `<set />` tag, must be actually present in the Active Directory domain.

Variables can be used in the values of the `add` and `user` attributes of the `<mask />` and `<expr />` tags, as well as in the value of the `value` attribute of the `<filter />` tag specified as `\varname`. Allowed recursion level in variables is 16.

If the search returns several found objects, only the first one is used.

- `<mask />`—user name templates.

Attribute	Description
add	String added to a search filter using the AND operation with substitution elements.
user	User name mask using the DOS-like meta symbols * and #. Cannot be absent.

For example:

```
<mask user="*@#"  add="sAMAccountName=\1" />

<mask user="*\*"  add="sAMAccountName=\2" />
```

`\1` and `\2` are the links on matching masks in the `user` attribute.

- `<expr />`—user name templates using regular expressions (attributes are the same as in the `<mask />`).

  For example:

```
<expr user="^(.*)@([^.,=@\s\\]+)$"  add="sAMAccountName=\1" />

<expr user="^(.*)\\(.*)"            add="sAMAccountName=\2" />
```

  Correspondence between masks and regular expressions:

Mask	Regular expression
*	.*
#	[^.,=@\s\\]+

- `<filter />`—LDAP search filter.

Attribute	Description
value	String added to a search filter using the AND operation with substitution elements.

- <user-dn/> defines the rules of name translation to the DN (Distinguished Name) using DOS-like masks.

  In the <user-dn/> tag, the following wildcard characters are allowed:

  - * replaces a sequence of any characters except . , = @ \ and spaces;
  - # replaces a sequence of any characters.

- <user-dn-expr/> defines the rules of name translation to the DN using regular expressions.

  Here is an example of the same name translated to the DN in two ways:

```
<user-dn user="*@example.com" dn="CN=\1,DC=example,DC=com"/>
<user-dn-expr user="(.*)@example.com" dn="CN=\1,DC=example,DC=com"/>
```

  \1 .. \9 define the place where *, # or an expression in brackets is substituted for its value in the template.

  Accordingly, if the username is specified as login@example.com, you will get the following DN after translation: "CN=login,DC=example,DC=com".

- <user-dn-extension-enabled/> allows the ldap_user_dn_translate.ds Lua script (Translating user names to LDAP DN user hook) to be executed to translate usernames to DNs. This script runs after an attempt to apply the user-dn, user-dn-expr rules if no appropriate rule is found. The script has one parameter: the username. The script returns a string that contains the DN or nothing. If an appropriate rule is not found and the script is disabled or returns nothing, the specified username is used as it is.

- <bind dn/> specifies the DN of the user that is used to connect to the LDAP server, and <bind password/> specifies the password of the user that is used to connect to the LDAP server (empty DN and password are possible for anonymous connections).

  For example:

```
<bind dn="CN=some dn,OU=some name,DC=example,DC=com" password="***" />
```

If a separate account for Bind DN search (including anonymous) and DN login translation rules are set at the same time, then priority is given to the prior, while the latter is ignored.

Filters concatenation

```
<set variable="admingrp"  filter="&amp;(objectclass=group)(cn=ESuite Admin)"
attribute="dn" />
<mask user="*\*" add="sAMAccountName=\2" />
<filter value="&amp;(objectClass=user)(memberOf=\admingrp)" />
```

If the admingrp get the "CN=ESuite Admins,OU=some name,DC=example,DC=com" value after the search, and the user input was domain\user, when the result filter is

```
"(&(sAMAccountName=user)(&(objectClass=user)(memberOf=CN=ESuite
Admins,OU=some name,DC=example,DC=com)))"
```

Example of Configuring LDAP/AD Authentication

Here is an example of typical settings for authentication using LDAP. Settings are configured in the Control Center, in the **Administration** → **Authentication** → **LDAP/AD-authentication** section (for the **Advanced settings**).

Initial parameters of administrators who must be authenticated:

- domain: `dc.test.local`
- Active Directory group: `DrWeb_Admins`

Control Center settings:

Setting name		**Value**
Server type		Microsoft Active Directory
Server address		`dc.test.local`
Login templates of users to be authenticated	Account mask	`test\*` or `*@test.local`
	Login	`\1`
Membership of users to be authenticated	Name	`DrWeb_Admins`
	Type	group

B4. PAM Authentication

The settings are stored in the `auth-pam.conf` configuration file.

The main tags of the configuration file:

- `<enabled>`—defines whether PAM authentication is used. Valid values of the `value` attribute: `yes` | `no`.
- `<order>`—sequential number of PAM authentication if multiple authentication methods are used. The value of the `value` attribute is an integer consistent with the values of the other authentication methods.
- `<service>`—name of the service (module) to be used to create a PAM context. The service name is specified as the value of the `name` attribute. PAM can read the policies for this service from the `/etc/pam.d/`<*service_name*> file or from `/etc/pam.conf` if the file does not exist. If the `<service>` tag is missing from the configuration file, the name `drwcs` is used by default.

- <admin-flag>—parameters of the control flag used for identifying a user as an administrator. Changing the attribute values is not recommended.

An example of the configuration file:

```
...
<!-- Enable this authorization module -->
  <enabled value="no" />
<!-- This authorization module number in the stack -->
  <order value="50" />
<!-- PAM service name>" -->
  <service name="drwcs" />
<!-- PAM data to be queried: PAM stack must return INT zero/non-zero -->
  <admin-flag mandatory="no" name="DrWeb_Esuite_Admin" />
...
```

B5. Depended Permissions Sections

The list of administrative rights and their features

Code	Permission	Description	Control Center section
Manage groups of stations			
1*	**View groups of stations properties**	The list of user groups which administrator sees in the anti-virus network. All system groups are also displayed in the anti-virus network tree, but only stations from the specified group list are available inside. If the permission is granted only for some groups as opposed to the entire anti-virus network, the **Administration** → **Installations** → **Network installation** menu item is not available.	Anti-virus Network Anti-virus Network → General → Properties
2*	**Edit groups of stations properties**	The list of user groups, properties of which administrator can edit. Must contain groups from the list of permission 1.	
3	**View groups of stations configuration**	The list of user groups, configuration of which is available to view by administrator. Also, administrator is permitted to view configuration of stations, for which the groups from the list are primary.	Anti-virus Network Anti-virus Network → General → Running components Anti-virus Network → General → Quarantine

Code	Permission	Description	Control Center section
		Must contain groups from the list of permission 1.	Pages from the **Configuration** section
4	**Edit groups of stations configuration**	Same as permission 3, but editing is permitted. Must contain groups from the list of permission 3.	
5	**View stations properties**	The list of user groups that are primary for stations properties of which administrator is permitted to view. Must contain groups from the list of permission 1.	Anti-virus Network Anti-virus Network → General → Properties
6	**Edit stations properties**	Including ACL, blocking, access, etc. Same as permission 5, but editing is permitted. Must contain groups from the list of permission 5.	
8*	**Move stations into groups and remove stations from groups**	The list of user groups. Must contain groups from the list of permission 1.	Anti-virus Network
9	**Delete stations**	The list of user groups that are primary for stations which administrator can delete. Must contain groups from the list of permission 1.	
10	**Remote Agent installation and deinstallation**	The list of user groups, for stations of which administrator is permitted to run remote installation of Dr.Web Agents with selected ID. These groups must be a primary for installing stations. Must contain groups from the list of permission 1. Menu item is not displayed if there are forbidden objects. Network installation is available from the /esuite/network/index.ds only in if 16 permission is allowed.	

Code	Permission	Description	Control Center section
11	**Merge stations**	The list of user groups stations of which can be merged. These groups must be a primary for stations. The icon to merge stations is available on the toolbar. Must contain groups from the list of permission 1.	
12*	**View statistic tables**	The list of user groups statistics of which can be viewed by administrator. The permission allows to create a task in the Dr.Web Server schedule to receive periodically reports. The list of user groups which administrator can be specify in the task is set (groups for stations of which the reports will be received). If **Everyone** is set, reports will be received for all groups from the list. Must contain groups from the list of permission 1.	Anti-virus Network Pages from the **Statistics** section
23	**Edit licensing**	The list of user groups for which administrator can add/change/remove a license key. These groups must be a primary for the stations. Must contain groups from the list of permission 1.	Anti-virus Network → Configuration → License keys
40	**Start and interrupt components**	Starting and stopping components installed on the station.	Anti-virus Network → the **Components management** button on the toolbar → the **Interrupt running components** button Anti-virus Network → General → Protection components → the **Interrupt selected components**, **Launch selected components** buttons on the toolbar Anti-virus Network → the **Scan** button on the toolbar → the **Express scan**, **Full scan**, **Custom scan** buttons

Code	Permission	Description	Control Center section
			Anti-virus Network → General → Quarantine → the **Delete files**, **Export**, **Restore files**, **Scan files** buttons on the toolbar
48	**View technical support reports**	Creating and viewing logs of the components installed on the station.	Anti-virus Network → General → Technical support reports
Manage administrators			
18	**View Dr.Web Server schedule**	View the **Tasks execution log** table. Only tasks created by the administrator as well as by administrators from the selected group(s) can be viewed. If permissions 12 and 18 are denied, the viewing of the Dr.Web Server schedule page is forbidden. If permission 12 is allowed but 18 is denied, when viewing statistics, the schedule is available. The task for sending reports to an administrator is displayed depending on the presence of permission 12 and the **Periodic report** notification, even if permission 18 is denied.	Administration → Configuration → Dr.Web Server Task Scheduler Administration → Statistics → Task execution log
19	**Edit Dr.Web Server schedule**	Only tasks created by the administrator as well as by administrators from the selected group(s) can be edited.	Administration → Configuration → Dr.Web Server Task Scheduler
24	**Edit notifications configuration**		Administration → Notifications → Notifications configuration Administration → Notifications → Unsent notifications Administration → Notifications → Web console notifications
25	**Create administrators, administrative groups**	The corresponding icon in the toolbar is hidden either.	Administration → Configuration →

Code	Permission	Description	Control Center section
26	**Edit administrators accounts**	Administrator from the **Newbies** group sees only a tree of administrators, the root node of which is a group of this administrator, i.e. sees administrators from the own group and its subgroups. Administrator from the **Administrators** group sees all other administrators not depending on their groups. Administrator can edit administrative accounts from the specified groups. At this, the corresponding icon in the toolbar become available.	
27	**Delete administrators accounts**	Same as permission 26.	
28	**View properties and configuration of administrative groups**	Including administrators in groups and subgroups. Administrator is able to select only from a subgroup of own parent group.	
29	**Edit properties and configuration of administrative groups**	Including administrators in groups and subgroups. Administrator is able to select only from a subgroup of own parent group. If this permission is denied, even if permission 26 is allowed for this groups, administrator will not be able to disable inheritance or increase permissions to administrator in the group.	
39	**Display the "Newbies" administrative group**	Allow administrator to view the pre-installed **Newbies** group in the administrators tree. If administrator has not got permissions for viewing the group **Newbies**, and administrator is in this group, then administrator will see only own account.	
41	**Edit administrative notifications**	Editing configuration of the administrator notifications.	
Additional			

Code	Permission	Description	Control Center section
7	**Create stations**	At station creation, only the list of groups with permission 8 is available (group to which stations are placed, must have the 8 permission). At station creation, one of available user groups must become primary.	Anti-virus Network
13	**View audit**	Audit is available for full-rights administrator and for objects with permission 4.	Administration → Statistics → Audit log
16	**Run Network scanner**	If the permission is denied, the network installation for the /esuite/network/index.ds is not available.	Anti-virus Network Administration → Network scanner
17	**Approve newbies**	The groups list from permission 8 is available. This permission cannot be granted if an administrator is allowed to manage only several groups but not all anti-virus network objects. I.e., for the permission 1 (**View groups of stations properties**) the set of groups is specified.	Anti-virus Network
20	**View Dr.Web Server configuration and repository configuration**		Administration → Configuration → Web server configuration Administration → Repository → Repository state Administration → Repository → Delayed updates Administration → Repository → General repository configuration Administration → Repository → Detailed repository configuration Administration → Repository → Repository content

Code	Permission	Description	Control Center section
21	**Edit Dr.Web Server configuration and repository configuration**		Administration → Logs → Log of repository updates Administration → Configuration → User hooks Administration → Dr.Web Server → List of versions
22	**View license information**		Administration → Administration → License Manager
30	**Operation via XML API**		-
31	**View neighborhood connections**		Neighbors
32	**Edit neighborhood connections**		Neighbors
33	**Use additional features**	Limits assess to all subsections of **Additional features** section except the **Utilities** subsection which is always available.	Administration → Additional features
34	**Update repository**	Update Dr.Web Server repository from GUS.	The **Update repository** button in the **Repository state** section
42	**Edit own settings**	Permission to edit settings of own administrative account	Administration → Configuration → Administrators
43	**View Dr.Web Proxy Servers**	View Dr.Web Proxy Server settings (if viewing is denied, then so is editing).	Anti-virus Network → Proxies
44	**Edit Dr.Web Proxy Servers**	Edit Dr.Web Proxy Server settings.	Anti-virus Network → Proxies
45	**View policies properties and configuration**	If viewing is denied, then so is editing.	Anti-virus Network → Policies
46	**Edit policies properties and configuration**		Anti-virus Network → Policies
47	**Edit membership rules**	Edit user group membership rules.	Anti-virus Network

* Permissions 1, 2, 8, 12 are defined for station by the list of groups into which it is included but not by a primary group of the station.

If a station is included into the group and for the group some of this permissions are granted, when administrator will have access to the functions corresponding to these permissions not depending on whether the group is primary for the station or not. At this, granting is in priority: if a station is included into both granted and denied groups, administrator will have access to the functions corresponding to the permissions of granted group.

Appendix C. Notification System Settings

Base information on configuration of administrative notifications is given in the **Administrator Manual**, p. Setting Notifications (read online).

C1. The Description of the Notification System Parameters

The system of alerts for events connected with the anti-virus network components operation, the following types of messages sens are used:

- email notifications,
- notifications via the Web Console,
- notifications via SNMP,
- notifications via the Agent protocol,
- push notifications,
- notifications via the Syslog protocol.

Depending on the notification sens method, the sets of parameters in the key → value format are required. For each method, the following parameters are set:

General parameters

Parameter	Description	Default value	Obligatory
`TO`	The set of notification receivers separated by the | sign		yes
`ENABLED`	Enable or disable notification send	`true` or `false`	yes
`_TIME_TO_LIVE`	The number of notification resend attempts in case of fail	10 attempts	no
`_TRY_PERIOD`	Period in seconds between notification resend attempts	300 sec, (send not often than ones in 300 sec)	no

The tables with parameter lists for different notification send types are given below.

Email notifications

Parameter	Description	Default value
`FROM`	Address of the sender email	-

Parameter	Description	Default value
TO	Address of the receiver email	-
HOST	SMTP server address	127.0.0.1
PORT	SMTP server port number	• 25 if the SSL parameter value is no, • 465 if the SSL parameter value is yes
USER	SMTP server user	"" is specified, at least one authorization method must be enabled, otherwise mail will not be sent.
PASS	password of SMTP server user	""
STARTTLS	Encrypt data transfer. At this, switching to secured connection is performed by using the STARTTLS command. The 25 port is used by default for the connection.	yes
SSL	Encrypt data transfer. At this, a new secured TLS connection is established. The 465 port is used by default for the connection.	no
AUTH-CRAM-MD5	use the CRAM-MD5 authentication	no
AUTH-PLAIN	use the PLAIN authentication	no
AUTH-LOGIN	use the LOGIN authentication	no
AUTH-NTLM	use the NTLM authentication	no
SSL-VERIFYCERT	Validate the server SSL certificate	no
DEBUG	Enable debug mode, e.g., to resolve the problem when authorization failed	-

Notifications via Web console

Parameter	Description	Default value
TO	UUID of administrators, to which this notification will be send	-
SHOW_PERIOD	Time to store the message in seconds starting from the moment of receiving	86400 sec, i.e. one day.

Notifications via SNMP

Parameter	Description	Default value
TO	SNMP receiving entity, e.g., IP address	-
DOMAIN	Domain	• localhost for Windows OS, • "" for Unix-like OS.
COMMUNITY	SNMP community or the context	public
RETRIES	The number of notification resend attempts that the API performed	5 attempts
TIMEOUT	Time in seconds after which the API performs the notification resend attempt	5 sec

Notifications via the Agent protocol

Parameter	Description	Default value
TO	UUID of receiving stations	-
SHOW_PERIOD	Time to store the message in seconds starting from the moment of receiving	86400 sec, i.e. one day.

Push notifications

Parameter	Description	Default value
TO	Devices tokens which applications are get after registration on the vendor server, e.g. Apple	-
SERVER_URL	URL relay of the server, used to send notification to the vendor server	-

Notifications via the Syslog protocol

Parameter	Description	Default value
TO	Address of the Syslog notification receiver. The transfer protocol is TCP or UDP.	UDP, port 514
FORMAT	Notification format: RFC 5424 or CEF (Common Event Format).	RFC 5424

Parameter	Description	Default value
TIMEOUT	Period in seconds during which Dr.Web Server attempts to connect to the notification receiver via TCP.	5 sec
FACILITY	The process which created a message (kernel, mail system, etc.). The value must be between 0 and 23.	14
HOSTNAME	Sender. Dr.Web Server ID (FQDN, host name, IP address).	-

C2. The Parameters of Notification Templates

The text for messages is generated by a Dr.Web Server component named the templates processor on the basis of the templates files.

A template file consists of text and variables enclosed in braces. When editing a template file, the variables listed below can be used.

The variables are written as follows:

- {<*VAR*>}—substitute the current value of the <*VAR*> variable.
- {<*VAR*>:<*N*>}—the first <*N*> characters of the <*VAR*> variable.
- {<*VAR*>:<*first*>:<*N*>}—the value of <*N*> characters of the <*VAR*> variable that go after the first <*first*> characters (beginning from the <*first*>+1 symbol), if the remainder is less, it is supplemented by spaces on the right.
- {<*VAR*>:<*first*>:-<*N*>}—the value of <*N*> characters of the <*VAR*> variable that go after the first <*first*> characters (beginning from the <*first*>+1 symbol), if the remainder is less, it is supplemented by spaces on the left.
- {<*VAR*>/<*original1*>/<*replace1*>[/<*original2*>/<*replace2*>]}—replace specified characters of <*VAR*> variable with given characters: <*original1*> characters are replaced with <*replace1*> characters, <*original2*> characters are replaced with <*replace2*> characters, etc.

  The number of substitution pairs are not limited.
- {<*VAR*>/<*original1*>/<*replace1*[{<*SUB_VAR*>}]>[/<*original2*>/<*replace2*>]}—similarly to the above described replaces to the specified values but the <*SUB_VAR*> nested variable is used. Actions with nested variables are the same as the actions with parent variables.

  Nesting level for recursive substitutions is not limited.
- {<*VAR*>/<*original1*>/<*replace1*>/<*original2*>/<*replace2*>/*/<*replace3*>}—similarly to the above described replaces to the specified values but also the value from <*replace3*> can be substituted, if none of the listed original values match. Also, if either <*original1*>, or <*original2*> have not been found in <*VAR*>, all values will be replaced with the <*replace3*>.

Notation of variables

Variable	Value	Expression	Result
SYS.TIME	10:35:17:456	{SYS.TIME:5}	10:35
SYS.TIME	10:35:17:456	{SYS.TIME:3:5}	35:17
SYS.TIME	10:35:17:456	{SYS.TIME:3:-12}	°°°35:17:456
SYS.TIME	10:35:17:456	{SYS.TIME:3:12}	35:17:456°°°
SYS.TIME	10:35:17:456	{SYS.TIME/10/99/35/77}	99:77:17.456

Conventions

°—whitespace.

Environment Variables

To form messages texts you can use environment variables of the Dr.Web Server process (the **System** user).

Environment variables are available in the Control Center messages editor, in the **ENV** drop-down list. Please note: the variables must be specified with the `ENV.` prefix (the prefix ends with a dot).

System Variables

- `SYS.BRANCH`—system version (Dr.Web Server and Agents),
- `SYS.BUILD`—Dr.Web Server build date,
- `SYS.DATE`—current system date,
- `SYS.DATETIME`—current system date and time,
- `SYS.HOST`—Dr.Web Server DNS name,
- `SYS.MACHINE`—network address of a computer with Dr.Web Server installed,
- `SYS.OS`—operating system name of a computer with Dr.Web Server installed,
- `SYS.PLATFORM`—Server platform,
- `SYS.PLATFORM.SHORT`—short variant of `SYS.PLATFORM`,
- `SYS.SERVER`—product name (Dr.Web Server),
- `SYS.TIME`—current system time,
- `SYS.VERSION`—Dr.Web Server version.

Common Variables for Stations

- `GEN.LoginTime`—station login time,
- `GEN.StationAddress`—station address,
- `GEN.StationDescription`—station description,
- `GEN.StationID`—station unique identifier,
- `GEN.StationLDAPDN`—distinguished name of a station under Windows OS. Relevant for stations included into ADS/LDAP domain,
- `GEN.StationMAC`—stations MAC address,
- `GEN.StationName`—station name,
- `GEN.StationPrimaryGroupID`—identifier of the station primary group,
- `GEN.StationPrimaryGroupName`—name of the station primary group,
- `GEN.StationSID`—security identifier of a station.

Common Variables for Repository

- `GEN.CurrentRevision`—current version identifier,
- `GEN.Folder`—product location folder,
- `GEN.NextRevision`—updated version identifier,
- `GEN.Product`—product description.

Notification Parameters and Variables by Types

Administrators

Administrator authorization failed

Parameter	Value	
Notification sending reason	Sent on error of administrator authorization in the Control Center. The reason of authorization failure is given in the notification text.	
Additional configuration	Not required.	
Variables	`MSG.Login`	login
	`MSG.Address`	Control Center network address
	`MSG.LoginErrorCode`	numeric error code

Unknown administrator

Parameter	Value	
Notification sending reason	Sent on attempt of authorization in the Control Center by administrator with unknown login.	
Additional configuration	Not required.	
Variables	`MSG.Login`	login
	`MSG.Address`	network address of Dr.Web Security Control Center

Installations

For messages of this group, you can also use common variables for stations given above.

Installation on station failed

Parameter	Value	
Notification sending reason	Sent if an error occurred during the Agent installation on a station. The error reason is given in the notification text.	
Additional configuration	Not required.	
Variables	`MSG.Error`	error message

Installation on station successfully completed

Parameter	Value
Notification sending reason	Sent on succeeded Agent installation on a station.
Additional configuration	Not required.
Variables	Absent.

Licenses

Error when trying to request license key renewal data

Parameter	Value	
Notification sending reason	Sent if automatic license key renewal data cannot be obtained because the automatic license renewal server cannot be reached. If the issue persists, please contact the Doctor Web technical support service.	
Additional configuration	Not required.	
Variables	`MSG.Error`	error message

License key automatically updated

Parameter	Value	
Notification sending reason	Sent if a license key has been automatically updated. At this, a new key has been successfully downloaded and propagated on all objects of an old license key.	
Additional configuration	For detailed information on automatic license update, refer the **Administrator Manual**, p. Automatic Licenses Update (read online).	
Variables	`MSG.KeyId`	Identifier of an old license key
	`MSG.KeyName`	Name of an old license key
	`MSG.NewKeyId`	Identifier of a new license key
	`MSG.NewKeyName`	Name of a new license key

License key blocked

Parameter	Value	
Notification sending reason	Sent if during the update from Dr.Web Global Update System, information on the license key blocking has been received. This key can no longer be used.	
Additional configuration	To get detailed information on blocking reason, please contact the technical support service.	
Variables	`MSG.KeyId`	ID of a license key
	`MSG.KeyName`	Name of a user of a license key

License key cannot be automatically updated

Parameter	Value	
Notification sending reason	Sent if a license key cannot be automatically updated, because the compound of licensed components differs in the current and the new keys. At this, a new key successfully downloaded but not propagated on all objects of an old license key. You must replace the license key manually.	
Additional configuration	For detailed information on automatic license update, refer the **Administrator Manual**, p. Automatic Licenses Update (read online).	
Variables	`MSG.ExpirationDate`	date of license expiration
	`MSG.Expired`	• 1—the term has expired • 0—the term has not expired
	`MSG.KeyDifference`	The reason why automatic replacement is impossible: • the compound of licensed components differs in the current and the new license keys • the new license key has fewer licenses than the current license key
	`MSG.KeyId`	Identifier of an old license key
	`MSG.KeyName`	Name of an old license key
	`MSG.NewKeyId`	Identifier of a new license key
	`MSG.NewKeyName`	Name of a new license key

License key expiration

Parameter	Value	
Notification sending reason	Sent if the license key is about to expire, and automatic license update is not available.	
Additional configuration	Not required.	
Variables	`MSG.ExpirationDate`	date of license expiration
	`MSG.Expired`	• 1—the term has expired • 0—the term has not expired

Parameter	Value	
	MSG.KeyId	Identifier of a license key
	MSG.KeyName	Name of a license key

License limitation on a number of online stations is reached

Parameter	Value	
Notification sending reason	Sent if during connection of a station to Dr.Web Server, it was detected that the number of stations in the group into which the connected station is included, reached the limitation in the license key assigned for this group. At this, a new station cannot register on Dr.Web Server.	
Additional configuration	Not required.	
Variables	MSG.ID	station UUID
	MSG.StationName	station name
	Common variables for stations given above are also available.	

Licenses donation has expired

Parameter	Value	
Notification sending reason	Sent if the period of licenses donation to neighbor Dr.Web Servers from the license key of this Dr.Web Server has expired.	
Additional configuration	The period of licenses donation to neighbor Dr.Web Servers is specified in the **Administration** → **Dr.Web Server configuration** → **Licenses** section.	
Variables	MSG.ObjId	license key ID
	MSG.Server	the neighbor Dr.Web Server name

Limitation on a number of donated licenses is reached

Parameter	Value
Notification sending reason	Sent if the number of requested licenses for donation to neighbor Dr.Web Servers exceeds the number of licenses that are available in the license key.

Parameter	Value	
Additional configuration	Not required.	
Variables	`MSG.ObjId`	license key ID

Limitation on a number of licenses in the license key

Parameter	Value	
Notification sending reason	Sent if during the Dr.Web Server startup, it was detected that the number of stations in a group already exceeded the number of licenses in the license key assigned to this group.	
Additional configuration	Not required.	
Variables	`MSG.KeyId`	ID of a license key
	`MSG.KeyName`	license key user name
	`MSG.Licensed`	number of allowed licenses
	`MSG.LicenseLimit`	licenses state: • 1—number of available licenses in the license key is close to the end • 2—there are no available licenses in the license key • 3—the license key has been assigned to more objects than this key allows.
	`MSG.Licensed`	number of objects to which the key has been assigned
	`MSG.Total`	number of licenses in the key

Number of stations in the group is close to the license limit

Parameter	Value	
Notification sending reason	Sent if the number of stations in the group is closing to the license limitation in the key assigned to this group.	
Additional configuration	The number of available licenses left in the key to send the notification is: less than three licenses or less than 5% from the total number of licenses in the key.	
Variables	`MSG.Free`	number of available licenses

Parameter	Value	
	MSG.Licensed	number of stations using licenses of this group
	MSG.Total	Total number of licenses in all keys assigned to the group. Please note: license keys of the group can also be assigned to other licensing objects.
	GEN.StationPrimaryGroupID	primary group ID
	GEN.StationPrimaryGroupName	primary group name

Newbies

For messages of this group, you can also use common variables for stations given above.

Station automatically rejected

Parameter	Value
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server and has been rejected by Dr.Web Server automatically.
Additional configuration	The situation may occur if in the **Administration** → **Dr.Web Server configuration** → **General** section, for the **Newbies registration mode** option, the **Always deny access** value is set.
Variables	Absent.

Station is waiting for approval

Parameter	Value
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server and administrator must approve or reject the station manually.
Additional configuration	The situation may occur if in the **Administration** → **Dr.Web Server configuration** → **General** section, for the **Newbies registration mode** option, the **Approve access manually** value is set.
Variables	Absent.

Station rejected by administrator

Parameter	Value	
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server and has been rejected by administrator manually.	
Additional configuration	The situation may occur if in the **Administration** → **Dr.Web Server configuration** → **General** section, for the **Newbies registration mode** option, the **Approve access manually** value is set and an administrator selected the **Anti-virus Network** →  **Unapproved stations** →  **Reject selected stations** option for this station.	
Variables	`MSG.AdminAddress`	network address of the Control Center
	`MSG.AdminName`	administrator name

Other

Epidemic in the network

Parameter	Value	
Notification sending reason	Sent if an epidemic detected in the anti-virus network. It means that during specified time period, it was detected more than specified number of threats in the network.	
Additional configuration	To sent epidemic notifications, you must set the **Track epidemic** flag in the **Administration** → **Dr.Web Server configuration** → **Statistics** section. Parameters on epidemic detection are set in the same section.	
Variables	`MSG.Infected`	total number of detected threats
	`MSG.Virus`	the most common threats

Large number of abnormally terminated connections detected

Parameter	Value
Notification sending reason	Sent on a large number of abnormally terminated connections with clients: stations, Agent installers, neighbor Dr.Web Servers, Proxy Servers.
Additional configuration	To be able to sent notifications on multiple abnormally terminated connections, you must set the **Abnormally terminated connections** flag in the **Administration** → **Dr.Web Server configuration** → **Statistics** section and configure corresponding parameters in the

Parameter	Value	
	same section.	
Variables	`MSG.Total`	number of terminated connections
	`MSG.AddrsCount`	number of addresses that were disconnected

Large number of blocks by the Application Control detected

Parameter	Value	
Notification sending reason	Sent on a large number of blocked applications at stations by the Application Control component.	
Additional configuration	To be able to sent notifications on multiple blocked applications, you must set the **Multiple blockings by Application Control** flag in the **Administration → Dr.Web Server configuration → Statistics** section and configure corresponding parameters in the same section.	
Variables	`MSG.Total`	total number of blocks
	`MSG.Profile`	most common profiles according to which the block was made

Neighbor server has not connected for a long time

Parameter	Value	
Notification sending reason	Sent according to the task in the Dr.Web Server schedule. Contains information that the neighbor Dr.Web Server has not connected to this Dr.Web Server for a long time. The date of last connection is given in the notification text.	
Additional configuration	The time period during which the neighbor Dr.Web Server should not get connected to send the notification, is set in the **Neighbor server has not connected for a long time** task of the Dr.Web Server schedule configured in the **Administration → Dr.Web Server Task Schedule**.	
Variables	`MSG.LastDisconnectTime`	the time when Dr.Web Server has been connected at the last time
	`MSG.StationName`	the neighbor Dr.Web Server name

Dr.Web Server log rotation error

Parameter	Value	
Notification sending reason	Sent if an error occurred during rotation of the Dr.Web Server operation log. The reason of log rotation error is given in the notification text.	
Additional configuration	Not required.	
Variables	`MSG.Error`	message text

Dr.Web Server log write error

Parameter	Value	
Notification sending reason	Sent when an error occurred during writing an information into the Dr.Web Server operation log. The reason of log write error is given in the notification text.	
Additional configuration	Not required.	
Variables	`MSG.Error`	message text

Statistic report

Parameter	Value	
Notification sending reason	Sent after generation of a periodic report according to the task in the Dr.Web Server schedule. Also, notification contains the path for downloading the report file.	
Additional configuration	The report is generated according to the **Statistic reports** task in the Dr.Web Server schedule configured in the **Administration** → **Dr.Web Server Task Schedule**.	
Variables	`MSG.Attachment`	path to the report
	`MSG.AttachmentType`	MIME type
	`GEN.File`	report file name

Summary report of Preventive protection

Parameter	Value
Notification sending reason	Sent at receiving a lot of reports from the Preventive protection

Parameter	Value	
	component on the network stations.	
Additional configuration	To send a single notification on the Preventive protection report, you must set the **Group reports of Preventive protection** flag in the **Administration** → **Dr.Web Server configuration** → **Statistics** section. Parameters on reports grouping are set in the same section.	
Variables	`MSG.AutoBlockedActCount`	number of processes with suspicious activity that were blocked automatically
	`MSG.AutoBlockedProc`	processes with suspicious activity that were blocked automatically
	`MSG.HipsType`	protected object type
	`MSG.IsShellGuard`	dividing on types of the Preventive protection reactions at automatic blocking: • blocking of unauthorized code • check the access to the protected objects
	`MSG.ShellGuardType`	the most common reason of a blocking of unauthorized code execution at automatic event blocking
	`MSG.Total`	total number of Preventive protection events detected on the network
	`MSG.UserAllowedActCount`	number of processes with suspicious activity that were allowed by user
	`MSG.UserAllowedHipsType`	type of the most common protected objects access to which was allowed by user
	`MSG.UserAllowedIsShellGuard`	dividing on types of the Preventive protection reactions when the access was allowed by user: • blocking of unauthorized code • check the access to the protected objects
	`MSG.UserAllowedProc`	processes with suspicious activity that were allowed by user

Parameter	Value	
	`MSG.UserAllowedShellGuard`	the most common reason of a blocking of unauthorized code execution which was allowed by user
	`MSG.UserBlockedActCount`	number of processes with suspicious activity that were blocked by user
	`MSG.UserBlockedHipsType`	type of the most common protected objects access to which was blocked by user
	`MSG.UserBlockedIsShellGuard`	dividing on types of the Preventive protection reactions when the access was blocked by user: • blocking of unauthorized code • check the access to the protected objects
	`MSG.UserBlockedProc`	processes with suspicious activity that were blocked by user
	`MSG.UserBlockedShellGuard`	the most common reason of a blocking of unauthorized code execution which was blocked by user

Repository

For messages of this group, you can also use common variables for repository given above.

Not enough free space on disk

Parameter	Value	
Notification sending reason	Sent if on a disk where the Dr.Web Server `var` folder with variable data located, is running out of space.	
Additional configuration	Low disk space defined if it is less than 315 MB or less than 1000 nodes (for UNIX system based OS) left, if this values do not redefined by environment variables.	
Variables	Common variables for repository given above are not available.	
	`MSG.FreeInodes`	the number of free inodes file descriptors (has the meaning only

Parameter	Value	
		for some Unix-like OS)
	MSG.FreeSpace	free space in bytes
	MSG.Path	the path to the folder with low free space
	MSG.RequiredInodes	number of free inodes required for operation (has the meaning only for some Unix-like OS)
	MSG.RequiredSpace	free space required for operation

Repository cannot be updated

Parameter	Value	
Notification sending reason	Sent if during update of repository or repository product from the GUS, an error has occurred. Reason of the update error and also the name of the product at product update error, are given in the notification text.	
Additional configuration	Not required.	
Variables	MSG.Error	error message
	MSG.ExtendedError	detailed description of an error

Repository product is up-to-date

Parameter	Value
Notification sending reason	Sent if during repository updates check, it was detected that requested product is up-to-date. At this, update of this product from the GUS is not required.
Additional configuration	Not required.
Variables	Absent.

Repository product is updated

Parameter	Value
Notification sending reason	Sent when repository update from the GUS successfully completed.

Parameter	Value	
Additional configuration	Not required.	
Variables	`MSG.Added`	list of added files (each name in a separate line)
	`MSG.AddedCount`	number of added files
	`MSG.Deleted`	list of deleted files (each name in a separate line)
	`MSG.DeletedCount`	number of deleted files
	`MSG.Replaced`	list of replaced files (each name in a separate line)
	`MSG.ReplacedCount`	number of replaced files

Repository update already running

Parameter	Value
Notification sending reason	Sent if during the Dr.Web Server update, the other update was started.
Additional configuration	Not required.
Variables	Absent.

Update of repository product is frozen

Parameter	Value
Notification sending reason	Sent if the repository product was frozen by administrator. At this, update of this product from the GUS is not performed.
Additional configuration	You can manage repository products including their frozen and unfrozen states in the **Administration** → **Detailed repository configuration** section.
Variables	Absent.

Update of repository product is started

Parameter	Value
Notification sending reason	Sent if during repository updates check, it was detected that for requested products the update is required. At this, the update from

Parameter	Value
	the GUS is launched.
Additional configuration	Not required.
Variables	Absent.

Stations

For messages of this group, you can also use common variables for stations given above.

In multiserver network, it is possible to receive notifications about events on stations of neighbor Dr.Web Servers. You can enable this option when configuring neighbor Dr.Web Server connections (see **Administrator Manual**, the Setting Connections between Several Dr.Web Servers (read online) section).

The following notifications are available to receive on event on the neighbor Dr.Web Server: **Security threat detected**, **Report of Preventive protection**, **Scan error**, **Scan statistics**.

Application Control blocked the process

Parameter	Value	
Notification sending reason	Sent if an application was blocked at station by the Application Control component.	
Additional configuration	Not required.	
Variables	`MSG.AppCtlAction`	applied action: • 0—unknown, • 2—blocked • 3—blocked (not found in the trusted applications list) • 5—blocked by deny rules • 7—blocked by policies settings.
	`MSG.AppCtlType`	event type: • 0—unknown • 1—process launch • 2—host process launch • 3—script interpreter launch • 4—module load • 5—driver load • 6—MSI setup launch

Parameter	Value	
		• 7—new executable file dropped on disk • 8—executable file modified on disk.
	MSG.Path	path to the blocked process
	MSG.Profile	name of the profile according to which the block was made
	MSG.Rule	name of the rule according to which the block was made
	MSG.SHA256	blocked process hash (SHA-256)
	MSG.StationTime	station time when the process was blocked
	MSG.Target	path to the blocked script in case of host process
	MSG.TargetSHA256	hash the blocked script in case of host process (SHA-256)
	MSG.TestMode	whether the test mode is on
	MSG.User	user on behalf of which the blocked object was launched

Application Control blocked the process from the known hashes of threats list

Parameter	Value	
Notification sending reason	Sent if an application from the known hashes of threats was blocked at station by the Application Control component.	
Additional configuration	Notification on detection by the list of known hashes is possible only if the usage of bulletins of known threat hashes is licensed (the license in at least one of the license keys used by Dr.Web Server is sufficient). You can check the license in the information on a license key that can be found in the **License Manager** section, the **Allowed lists of hash bulletins** parameter (If the feature is not licensed, this parameter is absent).	
Variables	MSG.AppCtlAction	applied action: • 0—unknown, • 2—blocked • 3—blocked (not found in the

Parameter	Value	
		trusted applications list) • 5—blocked by deny rules • 7—blocked by policies settings.
	MSG.AppCtlType	event type: • 0—unknown • 1—process launch • 2—host process launch • 3—script interpreter launch • 4—module load • 5—driver load • 6—MSI setup launch • 7—new executable file dropped on disk • 8—executable file modified on disk.
	MSG.Document	bulletin containing the hash
	MSG.Path	path to the blocked process
	MSG.Profile	name of the profile according to which the block was made
	MSG.Rule	name of the rule according to which the block was made
	MSG.SHA256	blocked process hash (SHA-256)
	MSG.StationTime	station time when the process was blocked
	MSG.Target	path to the blocked script in case of host process
	MSG.TargetSHA256	hash the blocked script in case of host process (SHA-256)
	MSG.TestMode	whether the test mode is on
	MSG.User	user on behalf of which the blocked object was launched

Cannot create the station account

Parameter	Value	
Notification sending reason	Sent if a new stations account cannot be created on Dr.Web Server. Error details are given in the Dr.Web Server log file.	
Additional configuration	Not required.	
Variables	`MSG.ID`	station UUID
	`MSG.StationName`	station name

Connection terminated abnormally

Parameter	Value	
Notification sending reason	Sent on abnormal termination of a connection with a client: station, Agent installer, neighbor Dr.Web Server, Proxy Server.	
Additional configuration	To be able to sent notifications on abnormally terminated connections, you must set the **Abnormally terminated connections** flag in the **Administration → Dr.Web Server configuration → Statistics** section and configure corresponding parameters in the same section.	
Variables	`MSG.Total`	number of terminated connections
	`MSG.Type`	client type

Critical error of station update

Parameter	Value	
Notification sending reason	Sent if a notification received from a station reports an error during update of anti-virus components from Dr.Web Server.	
Additional configuration	Not required.	
Variables	`MSG.Product`	updated product
	`MSG.ServerTime`	event receipt time, GMT

Device blocked

Parameter	Value
Notification sending reason	Sent if a notification received from a station reports that a connected

Parameter	Value	
	to the station device has been blocked by Dr.Web anti-virus component.	
Additional configuration	Not required.	
Variables	`MSG.Capabilities`	device characteristics
	`MSG.Class`	device class (the name of a parent group)
	`MSG.Description`	device description
	`MSG.FriendlyName`	user friendly name of the device
	`MSG.InstanceId`	identifier of a device instance
	`MSG.User`	user name

Report of Preventive protection

Parameter	Value	
Notification sending reason	Sent at receiving the report from the Preventive protection component from a station of this or neighbor Dr.Web Server.	
Additional configuration	Not required.	
Variables	`MSG.AdminName`	administrator who initiated the action on a suspicious process
	`MSG.Denied`	action on a suspicious process: • denied • allowed
	`MSG.HipsType`	protected object type
	`MSG.IsShellGuard`	dividing on types of the Preventive protection reactions: • blocking of unauthorized code • check the access to the protected objects
	`MSG.Path`	path to the process with suspicious activity
	`MSG.Pid`	identifier of the process with suspicious activity

Parameter	Value	
	MSG.ShellGuardType	reason of execution of unauthorized code blocking
	MSG.StationTime	time of event occurrence on a station
	MSG.Target	path to the protected object to which the access attempt was made
	MSG.Total	number of denials in case of automatic reaction of the Preventive protection
	MSG.User	user who launched the suspicious process
	MSG.UserAction	initiator of the action on a suspicious process • user • automatic reaction of the Preventive protection
	GEN.ServerRecvLinkID	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerRecvLinkName	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerOriginatorID	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	GEN.ServerOriginatorName	the name of the Dr.Web Server to which the station is connected from which the Preventive protection report was received

Report of Preventive protection on threat detection by known hashes of threats

Parameter	Value	
Notification sending reason	Sent at receiving the report from the Preventive protection component from a station of this or neighbor Dr.Web Server at threat detection from the list of known hashes of threats.	
Additional configuration	Notification on detection by the list of known hashes is possible only if the usage of bulletins of known threat hashes is licensed (the license in at least one of the license keys used by Dr.Web Server is sufficient). You can check the license in the information on a license key that can be found in the **License Manager** section, the **Allowed lists of hash bulletins** parameter (If the feature is not licensed, this parameter is absent).	
Variables	`MSG.AdminName`	administrator who initiated the action on a suspicious process
	`MSG.Denied`	action on a suspicious process: • denied • allowed
	`MSG.Document`	bulletin containing the hash of detected threat
	`MSG.HipsType`	protected object type
	`MSG.IsShellGuard`	dividing on types of the Preventive protection reactions: • blocking of unauthorized code • check the access to the protected objects
	`MSG.Path`	path to the process with suspicious activity
	`MSG.Pid`	identifier of the process with suspicious activity
	`MSG.SHA1`	SHA-1 hash of detected object
	`MSG.SHA256`	SHA-256 hash of detected object
	`MSG.ShellGuardType`	reason of execution of unauthorized code blocking
	`MSG.StationTime`	time of event occurrence on a station

Parameter	Value	
	MSG.Target	path to the protected object to which the access attempt was made
	MSG.Total	number of denials in case of automatic reaction of the Preventive protection
	MSG.User	user who launched the suspicious process
	MSG.UserAction	initiator of the action on a suspicious process • user • automatic reaction of the Preventive protection
	GEN.ServerRecvLinkID	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerRecvLinkName	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerOriginatorID	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	GEN.ServerOriginatorName	the name of Dr.Web Server to which the station is connected from which the Preventive protection report was received

Scan error

Parameter	Value
Notification sending reason	Sent if a notification received from a station reports an error during scanning.

Parameter	Value	
Additional configuration	Not required.	
Variables	MSG.Component	component name
	MSG.Error	error message
	MSG.ObjectName	object name
	MSG.ObjectOwner	object owner
	MSG.RunBy	component is launched by this user
	MSG.ServerTime	event receipt time, GMT
	GEN.ServerRecvLinkID	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerRecvLinkName	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerOriginatorID	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	GEN.ServerOriginatorName	the name of the Dr.Web Server to which the station is connected from which the Preventive protection report was received

Scan error at threat detection by known hashes of threats

Parameter	Value
Notification sending reason	Sent if scan error occurred at threat detection from the list of known hashes of threats.
Additional configuration	Notification on detection by the list of known hashes is possible only if the usage of bulletins of known threat hashes is licensed (the license in at least one of the license keys used by Dr.Web Server is sufficient).

Parameter	Value	
	You can check the license in the information on a license key that can be found in the **License Manager** section, the **Allowed lists of hash bulletins** parameter (If the feature is not licensed, this parameter is absent).	
Variables	`MSG.Component`	component name
	`MSG.Document`	bulletin containing the hash of detected threat
	`MSG.Error`	error message
	`MSG.ObjectName`	object name
	`MSG.ObjectOwner`	object owner
	`MSG.RunBy`	component is launched by this user
	`MSG.SHA1`	SHA-1 hash of detected object
	`MSG.SHA256`	SHA-256 hash of detected object
	`MSG.ServerTime`	event receipt time, GMT
	`GEN.ServerRecvLinkID`	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	`GEN.ServerRecvLinkName`	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	`GEN.ServerOriginatorID`	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	`GEN.ServerOriginatorName`	the name of the Dr.Web Server to which the station is connected from which the Preventive protection report was received

Scan statistics

Parameter	Value	
Notification sending reason	Sent if a notification received from a station reports a scan completion. Administrative notification also contains brief scan statistic.	
Additional configuration	Not required.	
Variables	`MSG.Component`	component name
	`MSG.Cured`	number of cured objects
	`MSG.DeletedObjs`	number of deleted objects
	`MSG.Errors`	number of scan errors
	`MSG.Infected`	number of malicious objects
	`MSG.Locked`	number of blocked objects
	`MSG.Modifications`	number of objects infected with known modifications of viruses
	`MSG.Moved`	number of moved objects
	`MSG.Renamed`	number of renamed objects
	`MSG.RunBy`	component is launched by this user
	`MSG.Scanned`	number of scanned objects
	`MSG.ServerTime`	event receipt time, GMT
	`MSG.Speed`	processing speed in KB/s
	`MSG.Suspicious`	number of suspicious objects
	`MSG.VirusActivity`	number of detected threats
	`GEN.ServerRecvLinkID`	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	`GEN.ServerRecvLinkName`	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received

Parameter	Value	
		(empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerOriginatorID	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	GEN.ServerOriginatorName	the name of the Dr.Web Server to which the station is connected from which the Preventive protection report was received

Security threat detected

Parameter	Value	
Notification sending reason	Sent if a notification received from a station reports the threat detection. Administrative notification also contains detailed information on detected threats.	
Additional configuration	Not required.	
Variables	MSG.Action	action upon a detection
	MSG.Component	component name
	MSG.InfectionType	threat type
	MSG.ObjectName	infected object name
	MSG.ObjectOwner	object owner
	MSG.RunBy	component is launched by this user
	MSG.ServerTime	event receipt time, GMT
	MSG.Virus	threat name
	GEN.ServerRecvLinkID	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerRecvLinkName	the name of the last neighbor Dr.Web Server from which the

Parameter	Value	
		Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	`GEN.ServerOriginatorID`	UUID of Dr.Web Server to which the station is connected from which the Preventive protection report was received
	`GEN.ServerOriginatorName`	the name of Dr.Web Server to which the station is connected from which the Preventive protection report was received

Security threat detected by known hashes of threats

Parameter	Value	
Notification sending reason	Sent if a notification received from a station reports the threat detection from the list of known hashes of threats. Administrative notification also contains detailed information on detected threats.	
Additional configuration	Notification on detection by the list of known hashes is possible only if the usage of bulletins of known threat hashes is licensed (the license in at least one of the license keys used by Dr.Web Server is sufficient). You can check the license in the information on a license key that can be found in the **License Manager** section, the **Allowed lists of hash bulletins** parameter (If the feature is not licensed, this parameter is absent).	
Variables	`MSG.Action`	action upon a detection
	`MSG.Component`	component name
	`MSG.Document`	bulletin containing the hash of detected threat
	`MSG.InfectionType`	threat type
	`MSG.ObjectName`	infected object name
	`MSG.ObjectOwner`	object owner
	`MSG.RunBy`	component is launched by this user
	`MSG.SHA1`	SHA-1 hash of detected object

Parameter	Value	
	MSG.SHA256	SHA-256 hash of detected object
	MSG.ServerTime	event receipt time, GMT
	MSG.Virus	threat name
	GEN.ServerRecvLinkID	UUID of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerRecvLinkName	the name of the last neighbor Dr.Web Server from which the Preventive protection report on connected stations was received (empty value if the report was received about stations connected to this Dr.Web Server)
	GEN.ServerOriginatorID	UUID of the Dr.Web Server to which the station is connected from which the Preventive protection report was received
	GEN.ServerOriginatorName	the name of the Dr.Web Server to which the station is connected from which the Preventive protection report was received

Station already logged in

Parameter	Value	
Notification sending reason	Send on attempt to connect to Dr.Web Server of a station with identifier which matches the identifier of a station already connected to this Dr.Web Server.	
Additional configuration	Not required.	
Variables	MSG.ID	station UUID
	MSG.Server	ID of the Dr.Web Server at which the station is registered
	MSG.StationName	station name

Station approved by administrator

Parameter	Value	
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server and has been approved by administrator manually.	
Additional configuration	The situation may occur if in the **Administration** → **Dr.Web Server configuration** → **General** section, for the **Newbies registration mode** option, the **Approve access manually** value is set and an administrator selected the **Anti-virus Network** →  **Unapproved stations** →  **Approve selected stations and set a primary group** option for this station.	
Variables	`MSG.AdminAddress`	network address of the Control Center
	`MSG.AdminName`	administrator name

Station authorization failed

Parameter	Value	
Notification sending reason	Sent if a station provided incorrect credentials when trying to connect to Dr.Web Server. Further actions that depend on a stations approval policy, are also given in the notification.	
Additional configuration	Stations approval policy is set in the **Newbies registration mode** option of the **Administration** → **Dr.Web Server configuration** → **General** section.	
Variables	`MSG.ID`	station UUID
	`MSG.Rejected`	values: • `rejected`—access to a station is denied • `newbie`—there was an attempt to assign the "newbie" status to a station
	`MSG.StationName`	station name

Station automatically approved

Parameter	Value
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server and has been approved by Dr.Web Server automatically.

Parameter	Value
Additional configuration	The situation may occur if in the **Administration → Dr.Web Server configuration → General** section, for the **Newbies registration mode** option, the **Approve access automatically** value is set.
Variables	Absent.

Station has not connected to Dr.Web Server for a long time

Parameter	Value	
Notification sending reason	Sent according to the task in the Dr.Web Server schedule. Contains information thon the station has not connected to this Dr.Web Server for a long time. The date of last connection is given in the notification text.	
Additional configuration	The time period during which the station should not get connected to send the notification, is set in the **Station has not connected for a long time** task of the Dr.Web Server schedule configured in the **Administration → Dr.Web Server Task Schedule**.	
Variables	Common variables for stations given above are not available.	
	`MSG.DaysAgo`	number of days since the last connection to Dr.Web Server
	`MSG.LastSeenFrom`	address of the station at the last connection to Dr.Web Server
	`MSG.StationDescription`	station description
	`MSG.StationID`	station UUID
	`MSG.StationMAC`	station MAC address
	`MSG.StationName`	station name
	`MSG.StationSID`	station security identifier

Station reboot required

Parameter	Value
Notification sending reason	Sent if a station reboot is required for one of the following reasons: • to complete the cure • to apply the updates • to change the state of hardware virtualization • to complete the cure and apply the updates

Parameter	Value	
	• to complete the cure and change the state of hardware virtualization • to apply the updates and change the state of hardware virtualization • to complete the cure, apply the updates and change the state of hardware virtualization.	
Additional configuration	Not required.	
Variables	`MSG.Reason`	reboot reason the list of possible reboot reasons is given in the predefined template

Station reboot required to apply updates

Parameter	Value	
Notification sending reason	Sent if a notification received from a station reports that the product has been installed or updated, and the station restart is required.	
Additional configuration	Not required.	
Variables	`MSG.Product`	updated product
	`MSG.ServerTime`	event receipt time, GMT

Unknown station

Parameter	Value	
Notification sending reason	Sent if a new station requested a connection to Dr.Web Server, but was not allowed to review for approval or rejection of the registration.	
Additional configuration	Not required.	
Variables	`MSG.ID`	UUID of unknown station
	`MSG.Rejected`	values: • `rejected`—access to a station is denied • `newbie`—there was an attempt to assign the "newbie" status to a station
	`MSG.StationName`	station name

Appendix D. The Specification of Network Addresses

In the specification the following conventions are taken:

- variables (the fields to be substituted by concrete values) are enclosed in angle brackets and written in italic,
- permanent text (remains after substitutions) is written in bold,
- optional elements are enclosed in brackets,
- the defined notion is placed on the left of the `::=` character string, and the definition is placed on the right (as in the Backus-Naur form).

D1. The General Format of Address

The network address looks as follows:

`[`*<protocol>*`://][`*<protocol-specific-part>*`]`

By default, *<protocol>* has the TCP value. The default values of *<protocol-specific-part>* are determined by the application.

The following obsolete addresses format is also allowed:

`[`*<protocol>*`/][`*<protocol-specific-part>*`]`

IP Addresses

- *<interface>*`::=`*<ip-address>*

  <ip-address> can be either a DNS name or an IP address separated by periods (for example, `127.0.0.1`).

- *<socket-address>*`::=`*<interface>*`:`*<port-number>*

  <port-number> must be specified by a decimal number.

When you specify an address of Dr.Web Server or the Agent, you can set the version of the protocol to use. The following variants are available:

- *<protocol>*`://`*<interface>*`:`*<port-number>* – use IPv4 and IPv6.
- *<protocol>*`://(`*<interface>*`):`*<port-number>* – use IPv4 only.
- *<protocol>*`://[`*<interface>*`]:`*<port-number>* – use IPv6 only.

For Example:

1. `tcp://127.0.0.1:2193`

   means a TCP protocol, port `2193` on an interface `127.0.0.1`.

2. `tcp://(example.com):2193`

means a TCP protocol, port `2193` on an IPv4 interface `example.com`.

3. `tcp://[::]:2193`

   means a TCP protocol, port `2193` on an IPv6 interface `0000.0000.0000.0000.0000.0000.0000.0000`

4. `localhost:2193`

   the same.

5. `tcp://:9999`

   value for the server: the default interface depending on the application (usually all available interfaces), port `9999`; value for client: the default connection to the host depending on the application (usually `localhost`), port `9999`.

6. `tcp://`

   TCP protocol, default port.

Connection-Oriented Protocol

<protocol>/<socket-address>

where *<socket-address>* sets the local address of the socket for a server or a remote server for a client.

Datagram-Oriented Protocol

<protocol>://<endpoint-socket-address>[-*<interface>*]

For Example:

1. `udp://231.0.0.1:2193`

   means using a multicast group `231.0.0.1:2193` on an interface depending on the application by default.

2. `udp://[ff18::231.0.0.1]:2193`

   means using a multicast group `[ff18::231.0.0.1]` on an interface depending on the application by default.

3. `udp://`

   application-dependent interface and endpoint.

4. `udp://255.255.255.255:9999-myhost1`

   using broadcasting messages on port `9999` on `myhost1` interface.

UDS Addresses

- Connection-oriented protocol:

  `unx://`*<file_name>*

- Datagram-oriented protocol:

  `udx://<file_name>`

For Example:

```
1. unx://tmp/drwcsd:stream
2. udx://tmp/drwcsd:datagram
```

SRV Addresses

`srv://[<server name>][@<domain name/dot address>]`

D2. Address Formats Used by Dr.Web Agents and Their Installers

This section describes Dr.Web Server address formats used by Dr.Web Agents and Dr.Web Agent installers for connecting to or detecting Dr.Web Servers. The formats and default settings (with empty values) listed below are used in the configurations of Dr.Web Agents and their installers.

Direct connection to Dr.Web Server

`[tcp://]<remote-socket-address>`

For example:

```
tcp://192.168.1.42:2193
```

where `192.168.1.42` is the Dr.Web Server address and `2193` is the port number.

By default (if no address is specified), the following values are applied:

```
tcp://127.0.0.1:2193
```

where `127.0.0.1` is the Dr.Web Server address and `2193` is the port number.

Information on direct connections is provided in the **Administrator Manual**, in the Direct Connections (read online) section.

Search for Dr.Web Server via the Dr.Web Server detection service specifying the protocol family and endpoints

`<drwcs-name>@udp://<endpoint-socket-address>[-<interface>]`

By default (if no address is specified), the following values are applied:

```
drwcs@udp://231.0.0.1:2193-0.0.0.0
```

With these values, a search for a Dr.Web Server named `drwcs` is performed via the UDP protocol using the multicast group `231.0.0.1:2193` on all interfaces.

Information on the Dr.Web Server detection service is provided in the **Administrator Manual**, in the Dr.Web Server Detection Service (read online) section.

Appendix E. Administration of the Repository

It is recommended to manage repository via the corresponding settings of the Control Center. For more details, see **Administrator Manual**, p. Administration of Dr.Web Server Repository (read online).

Repository settings are saved to the following repository configuration files:

- General configuration files reside in the root folder of the repository and specify parameters of update servers.
- Products configuration files reside in the root folders that correspond to concrete repository products and specify the files set and update settings for the product in the folder of which they are located.

After the configuration files have been edited, restart Dr.Web Server.

When setting up interserver links for product mirroring (see **Administrator Manual**, p. Peculiarities of a Network with Several Dr.Web Servers (read online), please remember that configuration files are not the part of the product and therefore are not properly handled by the mirror system. To avoid errors during the in operation of the updating system:

- for peer Dr.Web Servers, use identical configuration,
- for subordinate Dr.Web Servers, disable synchronizing of components through HTTP protocol or keep the configuration identical.

E1. General Configuration Files

.servers

The `.servers` file contains a list of servers for updating the components of Dr.Web Enterprise Security Suite in the Dr.Web Server repository from the GUS servers.

The servers on the list are polled consequently; once the update is successful, the polling procedure is terminated.

For example:

```
esuite.geo.drweb.com
```

.url

The `.url` file contains the basic URI of the update zone—the folder on the update servers that contains updates of a particular Dr.Web product.

For example:

```
update
```

.proto

The `.proto` file contains the name of the protocol which is used to receive updates from the update servers.

May take one of the following values: `http | https | ftp | ftps | sftp | scp | smb | smbs | file`.

> The SMB and SMBS protocols are available only for Dr.Web Servers under Unix-like OSs. The SMB protocol support is implemented using `curl` (supports SMBv1 only).

For example:

```
https
```

.auth

The `.auth` file contains the parameters of user authentication on the update server.

Authentication parameters are specified in the following format:

```
<authentication method>

<user name>

<password>
```

One of the following values can be specified as the authentication method: `none | any | safe | basic | digest | digestie | ntlm | ntlmwb | negotiate`. The values correspond to the values of the **Authorization method** drop-down list in the **Administration** → **General repository configuration** → **Dr.Web GUS** section of the Control Center.

<user name> is mandatory, *<password>* is optional.

For example:

```
safe

admin

root
```

.version

The `.version` file contains the version of the server from whose zone the updates are to be downloaded. It is used for debugging purposes and by default corresponds to the current version of the server in the MM.mm format.

.max-retry

The `.max-retry` file contains the maximum number of attempts for download errors from each of the update servers.

.cdn-mode

The `.cdn-mode` file contains the settings for Content Delivery Network (CDN) when loading the repository:

May take one of the following values:

- `on`—use CDN,
- `off`—do not use CDN.

.cert-mode

The `.cert-mode` file contains the settings for allowed SSL certificates of the update servers that will be automatically accepted:

May take one of the following values:

- `drweb`—accept only the SSL certificate of the Doctor Web company,
- `valid`—accept only valid SSL certificates,
- `any`—accept any certificates,
- `custom`—accept a user-specified certificate.

.cert-file

The `.cert-file` file contains the path to the user-specified certificate if the cert parameter is set to custom.

.ssh-mode

The `.ssh-mode` file contains the settings for authorization mode when using the SCP and SFTP protocols (based on SSH2).

May take one of the following values:

- `pwd`—authorization with the user login and password,
- `pubkey`—authorization with encryption keys.

.ssh-pubkey

The `.ssh-pubkey` file contains the path to the public SSH key of the update server.

.ssh-prikey

The `.ssh-prikey` file contains the path to the private SSH key of the update server.

.sync-ignore-flavoured

The `.sync-ignore-flavored` file contains a list of security components that will not be synchronized from the update zone to the local repository.

E2. Product Configuration Files

.description

The `.description` file specifies the product name. If the file is absent, the name of the respective folder of the product is used as the product name.

For example:

```
Dr.Web Server
```

..languages

The `..languages` file contains a list of language codes for which the update is configured. If the file is missing or empty, no update is configured for any languages, so it will not be performed.

..platforms

The `..platforms` file contains the full platform codes used in the product for which the update is configured. If the file is missing or empty, no update is configured for any platforms, so it will not be performed.

..formats

The `..formats` file contains the file formats that are configured to update, for example, document formats (html, pdf). If the file is missing or empty, no update is configured for any formats, so it will not be performed.

..items

The `..items` file determines which administrative utilities or enterprise products are updated. If the file is missing or empty, no update is configured for any utilities or products, so it will not be performed.

.sync-off

The file disables product updates. Its contents are irrelevant.

.deleted

The `.deleted` file marks the product as removed. All revisions from it will be deleted, synchronization with the GUS is disabled.

Exclusion files for updating Dr.Web Server Repository from GUS

.sync-only

The `.sync-only` file contains regular expressions that define the list of repository files that will be synchronized during a repository update from the GUS. Repository files not specified in the `.sync-only` file will not be synchronized. If the `.sync-only` file is absent, all repository files will be synchronized except for files excluded according to the settings in the `.sync-ignore` file.

.sync-ignore

The `.sync-ignore` file contains regular expressions that define the list of repository files that will be excluded from synchronization during a repository update from the GUS.

Example of the exception file:

```
^windows-nt-x64/

^windows-nt/

^windows/
```

Order of use of configuration files

If the `.sync-only` and `.sync-ignore` files are present for a product, the following procedure is used:

1. `.sync-only` is applied first. Files not matched by `.sync-only` are not handled.
2. `.sync-ignore` is applied to the rest of the files.

Exclusion files for updating Dr.Web Agents from Dr.Web Server

.state-only

The `.state-only` file contains regular expressions that define the list of repository files that will be synchronized when Dr.Web Agents are updated from Dr.Web Server. Repository files not specified in the `.state-only` file will not be synchronized. If the `.state-only` file is absent, all repository files will be synchronized except for files excluded according to the settings in the `.state-ignore` file.

.state-ignore

The `.state-ignore` file contains regular expressions that define the list of repository files that will be excluded from synchronization when Dr.Web Agents are updated from Dr.Web Server.

For example:

- do not receive German, Chinese, and Spanish interface languages (others will be received),
- do not receive any components designed for 64-bit Windows.

```
;^common/ru-.*\.dwl$ this will be updated
```

```
^common/de-.*\.dwl$

^common/cn-.*\.dwl$

^common/es-.*\.dwl$

^win/de-.*

^win/cn-.*

^windows-nt-x64\.*
```

The order of using `.state-only` and `.state-ignore` is the same as for the `.sync-only` and `.sync-ignore` files.

Notification sending configuration

The files of the `notify` group configure notifications sent when the corresponding repository products are successfully updated.

The following settings affect the **Repository product is updated** notification only. The exclusions are not applied to other notification types.

Configuration of the notification system is described in the **Administrator Manual**, the Setting Notifications (read online) section.

.notify-only

The `.notify-only` file contains regular expressions that define the list of repository files whose modification triggers a notification.

.notify-ignore

The `.notify-ignore` file contains regular expressions that define the list of repository files whose modification does not trigger a notification.

Order of use of configuration files

If the `.sync-only` and `.sync-ignore` files are present for a product, the following procedure is used:

1. When the product is being updated, files updated from the GUS are compared with the list of exclusions.
2. Files matched by `.notify-ignore` are excluded first.
3. Files not matched by `.notify-only` are excluded from the remaining files.

4. If any files not excluded in the previous steps remain, a notification is sent.

If the `.notify-only` and `.notify-ignore` files are absent, notifications are always sent (if they are enabled on the **Notifications configuration** page in the Control Center).

For example:

If an exclusion for `^.vdb.lzma$` is set in the `.notify-ignore` file and only virus databases are updated, a notification is not sent. If the `drweb32.dll` engine file is updated alongside the virus databases, a notification is sent.

Freeze settings

.delay-config

The `.delay-config` file contains settings to disable switching the product to a new revision. The repository continues to distribute the previous revision, and synchronization is no longer performed (the state of the product is "frozen"). If the administrator decides that a new revision is fit to be distributed, the administrator must enable its distribution in the Control Center (see the **Administrator Manual**, Administration of Dr.Web Server Repository (read online)).

The file contains two case insensitive parameters which are separated by a semicolon.

File format:

```
Delay [ON|OFF|APPROVAL]; UseFilter [YES|NO]
```

Parameter	Possible values	Description
`Delay`	`ON|OFF|APPROVAL`	• `ON`—product update freeze is enabled. • `OFF`—product update freeze is disabled. • `APPROVAL`—product update freeze is enabled until administrator approval.
`UseFilter`	`YES|NO`	• `Yes`—freeze updates only if update files match the exception list in the `.delay-only` file. • `No`—freeze updates in any case.

For example:

```
Delay ON; UseFilter NO
```

.delay-only

The `.delay-only` file contains regular expressions that define the list of repository files whose modification disables switching the product to a new revision.

If a file from a repository update matches a regular expression from `.delay-only` and the `UseFilter` setting in the `.sync-only` file is enabled, the revision is frozen.

.rev-to-keep

The `.rev-to-keep` file contains the number of product revisions to be stored.

For example:

```
3
```

.on-demand

The presence of the `.on-demand` file means that on-demand synchronization of the product is enabled, i.e. if there are customers requesting this product.

.is-sync-off-in

The presence of the `.is-sync-off-in` file means that receiving product updates via inter-server communications is disabled.

.is-sync-off-out

The presence of the `.is-sync-off-out` file means that the distribution of product updates via inter-server communications is disabled.

.no-platform

The presence of the `.no-platform` file means that the top-level directories in the product revision are not considered platforms, that is, the revision applies to any platform.

Appendix F. Configuration Files Format

This section describes the format of the following files:

File	Description
drwcsd.conf	Dr.Web Server configuration file
webmin.conf	Dr.Web Security Control Center configuration file
download.conf	Configuration file to set up downloaded from the Dr.Web Server data
drwcsd-proxy.conf	Dr.Web Proxy Server configuration file
drwreploader.conf	Repository loader configuration file
share.conf	Shared resource configuration file

If on the computer with corresponding component, Dr.Web Agent with enabled self-protection is installed, before editing configuration files, disable the **Self-protection** component via the Dr.Web Agent settings.

After you save all changes, it is recommended to enable **Self-protection** component.

F1. Dr.Web Server Configuration File

By default, the Dr.Web Server configuration file `drwcsd.conf` is located in the `etc` subfolder of the Dr.Web Server installation folder. Using a command line parameter, you can start Dr.Web Server and specify a custom location and name of its configuration file (for more information, see Appendix G3. Dr.Web Server).

To manually edit the Dr.Web Server configuration file, proceed as follows:

1. Stop the Dr.Web Server (see **Administrator Manual**, Dr.Web Server (read online).
2. Disable Self-Protection (if you installed Dr.Web Agent on the computer where Dr.Web Server is also installed and enabled Self-Protection, you can disable it in the Dr.Web Agent menu).
3. Edit the Dr.Web Server configuration file.
4. Start Dr.Web Server (see **Administrator Manual**, Dr.Web Server (read online).

Dr.Web Server configuration file format

The configuration file of Dr.Web Server is in XML format.

Description of the Dr.Web Server configuration file parameters:

- `<version value="" />`

  Current version of the configuration file.

- `<name value="" />`

  Name of Dr.Web Server or the cluster of Dr.Web Servers to be looked up by Dr.Web Agents, Dr.Web Agent installers, and the Control Center. Leave the value blank ("" is used by default) to use the name of the computer where Dr.Web Server is installed.

  

  It is recommended that you specify the name of Dr.Web Server in FQDN format after registering it with the DNS name server. This will make it easier to configure the anti-virus network when reinstalling Dr.Web Server on another computer. In this case, if you need to change the address of Dr.Web Server, it is enough to change the name of the computer where Dr.Web Server is installed in the DNS server settings, and all Dr.Web Agents will automatically connect to the new server.

  1. If there is a local DNS server in the network, create a separate record for Dr.Web Server and Dr.Web Proxy Server (for example, `drwebes.company.lan`).
  2. Specify the FQDN name of Dr.Web Server in the settings of Dr.Web Agents.
  3. In addition to the FQDN name, it is recommended that you specify the IP address of Dr.Web Server in the Dr.Web Agent settings and update this address when it changes. Thus, if Dr.Web Agent cannot reach Dr.Web Server by its name, it will try to connect to it using its address.

- `<id value="" />`

  Unique Dr.Web Server identifier. Prior to version 10, it was included in the Dr.Web Server license key. It is now specified in the Dr.Web Server configuration file.

- `<passwd-salt value="" />`

  Cryptographic salt. A string of random data that is added to the administrator password. The combined value is hashed by a hash function and stored as a single hash in the database to protect the password from brute force cracking. The salt is generated by default after installing or upgrading Dr.Web Server from previous versions.

  In addition to the static salt, a dynamic salt is generated for each password. The PBKDF2 password-based key generation standard is used as the HMAC authentication code when calculating the salted password fingerprint. Thus, the password is combined with the salt and hashed multiple times. By default, the use of static salt is disabled, dynamic salt is always used.

  

  If the salt is present, it is not possible to view or change the administrator password using the provided database management utility (`drwidbsh3`).

  

  If you are using a cluster of Dr.Web Servers, manually set the same salt value on all Dr.Web Servers in the cluster.

- `<location city="" country="" department="" floor="" latitude="" longitude="" organization="" province="" room="" street="" />`

  Geographic location of Dr.Web Server.

  Attribute description:

Attribute	Description
city	City
country	Country
department	Department name
floor	Floor
latitude	Latitude
longitude	Longitude
organization	Organization name
province	Province name
room	Room number
street	Street name

- `<threads count="" />`

  Number of threads for processing data from Dr.Web Server clients (Dr.Web Agents and their installers, neighboring Dr.Web Servers, Dr.Web proxy servers). The minimum value is 5. The default value is 5.

  This parameter affects the performance of Dr.Web Server. Changing the default value when using the embedded database is not recommended. When using an external database, a larger parameter value may be required (see Dr.Web Server Load and Recommended Configuration Parameters). If Dr.Web Server is used in an anti-virus network with a large number of client connections, it is recommended that you contact the Doctor Web technical support team before changing the value.

- `<newbie approve-to-group="" mode="" />`

  Access mode for new stations.

  Attribute description:

Attribute	Allowed values	Description	Default
approve-to-group	–	Group to be set as the primary group by default for new stations in the **Allow access automatically** mode (`mode='open'`).	Empty value, which means the **Everyone** group is assigned as the primary group.

Attribute	Allowed values	Description	Default
mode	• open—allow access automatically, • closed—always deny access, • approval—approve access manually.	New station approval policy.	–

For more information, see the **Administrator Manual**, New Stations Approval Policy (read online).

- **<emplace-auto** enabled="" />

  This mode is used to create station accounts in the Control Center when installing Dr.Web Agents using a group installation package if the number of already created accounts is insufficient.

Attribute	Allowed values	Default
enabled	• yes—automatically create missing station accounts, • no—installation is possible only for the number of accounts already created in the group whose installation package is being launched.	yes

- **<unauthorized-to-newbie** enabled="" />

  Action policy for unauthorized stations. Allowed values for enabled:

  - yes—stations that failed to be authorized (for instance, if the database is corrupted) will be automatically reset to newbies,
  - no (default)—normal mode of operation.

- **<maximum-authorization-queue** size="" />

  Maximum number of stations in the Dr.Web Server authorization queue. Contact the Doctor Web technical support team before changing the default value.

- **<reverse-resolve** enabled="" />

  Replace IP addresses with DNS names of computers in the fields with workstation addresses in the Control Center. Allowed values for enabled:

  - yes—use DNS names,
  - no (default)—use IP addresses.

- **<replace-netbios-names** enabled="" host="" />

  Replace computer names with their DNS names in the Control Center.

  Attribute description:

Attribute	Allowed values	Description
enabled	• yes—replace, • no—do not replace. The <agent-host-names /> parameter will be used instead.	Name replacement mode.
host	• yes—display partially qualified DNS names (before the trailing dot in FQDN), • no—display fully qualified DNS names (FQDN).	Displayed name format after replacement.

- <agent-host-names mode="" />

  Format in which the names of anti-virus network workstations are sent by Dr.Web Agents to Dr.Web Server and are displayed in the anti-virus network directory of the Control Center. This parameter is ignored if the enabled attribute of the <replace-netbios-names /> parameter is set to yes.

  Allowed values for mode:

  - netbios—NetBIOS names (used by default if the attribute is empty or the parameter is not present),
  - fqdn—fully qualified DNS names (FQDN),
  - host—partially qualified DNS names (before the trailing dot in FQDN).

- <dns>

  DNS settings.

  - <timeout value="" />

    Time-out in seconds for resolving DNS direct/reverse queries. Set the value to 0 to not limit the wait time until the end of the resolution.

  - <retry value="" />

    Maximum number of DNS retries when DNS query resolution fails.

  - <cache enabled="" negative-ttl="" positive-ttl="" />

    Time for storing responses from the DNS server in the cache.

    Attribute description:

Attribute	Allowed values	Description
enabled	• yes—store responses in the cache, • no—do not store responses in the cache.	Storing responses in the cache.
negative-ttl	–	Cache storage time (TTL) for negative responses from the DNS server in minutes.

Attribute	Allowed values	Description
positive-ttl	–	Cache storage time (TTL) for positive responses from the DNS server in minutes.

- <servers>

  List of DNS servers that replaces the default system list. Contains one or several <server address="" /> child elements, where the address parameter specifies the IP address of the server.

- <domains>

  List of DNS domains that replaces the default system list. Contains one or several <domain name="" /> child elements, where the name parameter specifies the domain name.

- <cache>

  Caching settings.

  The <cache> element contains the following child elements:

  - <interval value="" />

    Time interval for starting cache flush (in seconds). The default value is 120 (two minutes).

  - <quarantine ttl="" />

    The storage time (TTL) in the cache of Dr.Web Server quarantine (in seconds). The default value is 604800 (one week).

  - <download ttl="" />

    The storage time (TTL) in the cache of personal and group installation package files (in seconds). The default value is 604800 (one week).

  - <repository ttl="" />

    The storage time (TTL) in the cache of repository files loaded into memory as a result of a download request (in seconds). The default value is 120 (two minutes).

  - <file ttl="" />

    The storage time (TTL) of the file cache (in seconds). The file cache is stored as hard links leading to the Dr.Web Server repository files. The default value is 604800 (one week).

- <replace-station-description enabled="" />

  Synchronize station descriptions on Dr.Web Server with the **Computer description** field on the **System properties** page on stations. Allowed values for enabled:

  - yes—replace the description on Dr.Web Server with the description on the station,
  - no (default)—ignore the description on station.

- <time-discrepancy value="" />

  Allowable difference between the system time on Dr.Web Server and Dr.Web Agents in minutes. If the difference is larger than the specified value, it will be noted in the status of the station on Dr.Web Server. A 3 minute difference is allowed by default. The empty value or 0 means that this check is disabled.

- `<encryption mode="" />`

  Traffic encryption mode. Allowed values for `mode`:

  - `yes`—use encryption,
  - `no`—do not use encryption,
  - `possible`—encryption is possible.

  The default is `yes`.

  For more information, see the **Administrator Manual**, Traffic Encryption and Compression (read online).

- `<compression level="" mode="" />`

  Traffic compression mode.

  Attribute description:

Attribute	Allowed values	Description
`level`	Integer from 1 to 9.	Compression level.
`mode`	• `yes`—use compression, • `no`—do not use compression, • `possible`—compression is possible.	Compression mode.

  For more information, see the **Administrator Manual**, Traffic Encryption and Compression (read online).

- `<track-agent-jobs enabled="" />`

  Allow monitoring and storing the results of task execution on workstations in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<track-agent-status enabled="" />`

  Allow monitoring of changes in the status of stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<track-virus-bases enabled="" />`

  Allow monitoring of changes in the status (composition, modification) of virus databases on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`. The parameter is ignored if `<track-agent-status enabled="no" />`.

- `<track-agent-modules enabled="" />`

  Allow monitoring of module versions on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<track-agent-components enabled="" />`

  Allow monitoring of the list of components installed on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<track-agent-userlogon enabled="" />`

  Allow monitoring of user sessions on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<track-agent-environment enabled="" />`

  Allow monitoring of hardware and software configurations on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-run-information enabled="" />`

  Allow monitoring of information on started and stopped anti-virus components on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-infection enabled="" />`

  Allow monitoring of threat detection on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-scan-errors enabled="" />`

  Allow monitoring of scan errors on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-scan-statistics enabled="" />`

  Allow monitoring of scan statistics on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-installation enabled="" />`

  Allow monitoring of Dr.Web Agent installations on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-blocked-devices enabled="" />`

  Allow monitoring of devices blocked by the Office Control component and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-appcontrol-activity enabled="" />`

  Allow monitoring of process activity on stations detected by Application Control (to populate the Application catalog) and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<keep-appcontrol-block enabled="" />`

  Allow monitoring of blocked processes on stations by Application Control and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<quarantine enabled="" />`

  Allow monitoring of information on the Quarantine status on stations and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<update-bandwidth queue-size="" value="" />`

  Maximum network traffic bandwidth in KB/sec for delivering updates from Dr.Web Server to Dr.Web Agents.

  Attribute description:

Attribute	Allowed values	Description	Default
queue-size	• positive integer, • unlimited.	Maximum allowed number of update distribution sessions running at the same time on Dr.Web Server. When the limit is reached, Dr.Web Agent requests are placed into the waiting queue. The waiting queue size is unlimited.	unlimited
value	• maximum speed in KB/sec, • unlimited.	Maximum total bandwidth for transferring updates.	unlimited

- `<install-bandwidth queue-size="" value="" />`

  Maximum network traffic bandwidth in KB/sec for transferring data when Dr.Web Agents are being installed on stations.

  Attribute description:

Attribute	Allowed values	Description	Default
queue-size	• positive integer, • unlimited.	Maximum allowed number of Dr.Web Agent installation sessions launched simultaneously from Dr.Web Server. When the limit is reached, Dr.Web Agent requests are placed into the waiting queue. The waiting queue size is unlimited.	unlimited
value	• maximum speed in KB/sec, • unlimited.	Maximum total bandwidth for transferring data during Dr.Web Agent installation.	unlimited

- `<geolocation enabled="" startup-sync="" />`

  Enable synchronization of station geolocation data between Dr.Web Servers.

  Attribute description:

Attribute	Allowed values	Description
enabled	• yes—allow synchronization, • no—disable synchronization.	Synchronization mode.
startup-sync	Positive integer.	Number of stations without any geographical coordinates, whose data is requested when establishing a connection between Dr.Web Servers.

- `<audit enabled="" />`

  Allow monitoring of administrator actions in Dr.Web Security Control Center and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<audit-internals enabled="" />`

  Allow monitoring of internal operations on Dr.Web Server and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<audit-xml-api enabled="" />`

  Allow monitoring of operations via Web API on Dr.Web Server and storing the information in the Dr.Web Server database. Allowed values for `enabled`: `yes` or `no`.

- `<proxy auth-list="" enabled="" host="" password="" user="" />`

  Parameters of connections to Dr.Web Server via an HTTP proxy server.

  Attribute description:

Attribute	Allowed values	Description

`auth-list`	• `none`—do not use authentication,
	• `any`—any supported method,
	• `safe`—any secure supported method,
	• if more than one of the following methods are set, they must be separated by a space:
	  ▫ `basic`
	  ▫ `digest`
	  ▫ `digestie`
	  ▫ `ntlmwb`
	  ▫ `ntlm`
	  ▫ `negotiate`
		Proxy server authentication type. The default is `any`.

`enabled`	• `yes`—use proxy server, • `no`—do not use proxy server.	Connecting to Dr.Web Server via HTTP proxy server.
`host`	–	Proxy server address.
`password`	–	Password of the proxy server user if the proxy server requires authentication.
`user`	–	Name of the proxy server user if the proxy server requires authentication.

When specifying the list of allowed authentication methods for the proxy server, you can use the `only` tag (add it to the end of the list with a space) to change the algorithm of selecting the authentication method.

For more information, see https://curl.se/libcurl/c/CURLOPT_HTTPAUTH.html.

- `<statistics enabled="" id="" interval="" />`

  Parameters for sending statistics on events related to threats to Doctor Web (the data will be used to populate the https://stat.drweb.com/ section).

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	• yes—send statistics, • no—do not send statistics.	Sending statistics to Doctor Web.	–
id	–	Identifier (MD5 hash of the Dr.Web Agent license key).	MD5 hash of the current Dr.Web Agent license key.
interval	Positive integer.	Interval of sending statistics in minutes.	30

- `<cluster>`

  Parameters of a Dr.Web Server cluster for exchanging data in a multiserver anti-virus network configuration.

  Contains one or several `<on multicast-group="" port="" interface="" />` child elements.

  Attribute description:

Attribute	Description
multicast-group	IP address of the multicast group that Dr.Web Servers use to exchange information.
port	Port number of the network interface used by the transport protocol to transfer data to the multicast group.
interface	IP address of the network interface used by the transport protocol to transfer data to the multicast group.

- `<multicast-updates enabled="" />`

  Allows to configure the delivery of group updates to workstations using the multicast protocol. Allowed values for `enabled`: `yes` or `no`.

  `<multicast-updates>` contains multiple child elements and attributes:

Child element	Attribute	Description	Default
port `<port value="" />`	value	Port number of the Dr.Web Server network interface used by the multicast transport protocol to deliver updates. This port is used by all multicast groups. For multicast updates, specify any unused port different from the one specified in the Dr.Web Server transport protocol settings.	2197
ttl `<ttl value="" />`	value	Time-to-live of a transferred UDP datagram. This value is used by all multicast groups.	8

Child element	Attribute	Description	Default
group <group address="" />	address	IP address of the multicast group through which stations will receive group updates.	233.192.86.0 for IPv4 FF0E::176 for IPv6
on <on interface="" ttl="" />	interface	IP address of the Dr.Web Server network interface used by the multicast transport protocol to transmit updates.	–
	ttl	Time-to-live of a UDP datagram transferred via a specified network interface. Has a higher priority than the general <ttl value="" /> child element.	8
transfer <transfer datagram-size="" assembly-timeout="" updates-interval="" chunks-interval="" resend-interval="" silence-interval="" accumulate-interval="" announce-send-times="" />	datagram-size	UDP datagram size (bytes)—size of UDP datagrams used by the multicast protocol in bytes. The allowed range is 512–8192. To avoid fragmentation, it is recommended that you set a value less than the MTU (Maximum Transmission Unit) of the network.	1400
	assembly-timeout	File transmission time (ms.)—during the specified time a single update file is transmitted, after which Dr.Web Server starts sending the next file. Any files that could not be transferred during the multicast update phase will be transferred during the standard TCP update session.	180000
	updates-interval	Multicast update duration (ms.)—duration of the update process via the multicast protocol. Any files that could not be transferred during the multicast update phase will be transferred during the standard TCP update session.	600000
	chunks-interval	Packet transmission interval (ms.)—interval for sending packets to the multicast group. A small interval value can cause significant packet loss and overload the network. It is not recommended to change this parameter.	14
	resend-interval	Interval between retransmission requests (ms.)—Dr.Web Agents send requests for retransmission of lost packets with this interval.	1000

Child element	Attribute	Description	Default
		Dr.Web Server accumulates these requests and sends out any lost packets afterwards.	
	silence-interval	"Silence" interval on the line (ms.)—whenever file transmission is over before the allotted time and no requests for retransmission of lost packages are received from Dr.Web Agents during the specified "wait" time interval, Dr.Web Server assumes that all Dr.Web Agents successfully received update files and initiates transfer of the next file.	10000
	accumulate-interval	Retransmission request accumulation interval (ms.)—during the specified time interval, Dr.Web Server accumulates requests for retransmission of lost packets from Dr.Web Agents. Dr.Web Agents request lost packets. Dr.Web Server accumulates these requests throughout the specified time and sends out any lost packets afterwards.	2000
	announce-send-times	Number of file transmission announcements—the number of times Dr.Web Server announces a file transfer to a multicast group before the update transfer starts. When it is announced, a UDP datagram with file metadata is sent to the multicast group. Increasing the number of announcements can improve transmission reliability, but at the same time it may reduce the amount of data that can be transmitted in the time allotted for an update using the multicast protocol.	3

Optionally, the <**multicast-updates**> element can also contain the <**acl**> child element, which is used to create ACL lists. This allows you to limit the number of TCP addresses of workstations that will be able to receive group updates via multicast protocol from this Dr.Web Server. The <**acl**> child element is not present initially, which means no restrictions are applied by default.

<**acl**> as part of <**multicast-updates**> includes the following child elements:

- <**priority** mode="" />

  Sets the list priority. Allowed values for `mode`: `allow` or `deny`. When <**priority** mode="deny" /> is set, the <**deny**> list has a higher priority than the <**allow**> list. Addresses not included in either of the lists or included in both of them are denied. Only addresses included in the <**allow**> list and not included in the <**deny**> list are allowed.

- `<allow>`

  List of TCP addresses that are allowed to receive updates over the multicast protocol. The `<allow>` element contains one or several `<ip address="" />` child elements to specify the allowed addresses in the IPv4 format and `<ip6 address="" />` to specify the allowed addresses in the IPv6 format. The `address` attribute defines network addresses in the following format: <*IP address*>/[<*prefix*>].

- `<deny>`

  List of TCP addresses that are not allowed to receive updates over the multicast protocol. The `<deny>` element contains one or several `<ip address="" />` child elements to specify the denied addresses in the IPv4 format and `<ip6 address="" />` to specify denied addresses in the IPv6 format. The `address` attribute defines network addresses in the following format: <*IP address*>/[<*prefix*>].

- `<database connections="" speedup="" />`

  Database definition.

  Attribute description:

Attribute	Allowed values	Description	Default
`connections`	Positive integer.	Maximum number of connections of the database to Dr.Web Server. Do not change the default value when using the embedded database. When using an external database, a larger parameter value may be required (see Dr.Web Server Load and Recommended Configuration Parameters). If Dr.Web Server is used in an anti-virus network with a large number of client connections, it is recommended that you contact the Doctor Web technical support team before changing the value.	2
`speedup`	`yes | no`	Automatically perform the delayed purging of the database after its initialization, upgrade, and import (see the **Administrator Manual**, Database (read online)).	`yes`

  The `<database>` element contains one of the following child elements:

  

  The `<database>` element can contain only one child element defining a specific database.

  It is recommended that you contact the Doctor Web technical support team before changing database attributes that may be present in the configuration file template but lack their descriptions.

- `<sqlite dbfile="" cache="" cachesize="" readuncommitted="" precompiledcache="" synchronous="" checkintegrity="" autorepair="" mmapsize="" wal="" wal-max-pages="" wal-max-seconds="" />`

  Defines configuration of the embedded SQLite3 database.

  Attribute description:

Attribute	Allowed values	Description	Default
`dbfile`	–	Database file name.	`database.sqlite`
`cache`	`SHARED | PRIVATE`	Caching mode.	`SHARED`
`cachesize`	Positive integer.	Database cache size (in 1.5 KB pages).	2048
`readuncommitted`	`on | off`	Transition to the READ UNCOMMITTED transaction isolation level (access data that has been changed or deleted but not committed by another transaction).	`off`
`precompiledcache`	Positive integer.	Cache size for precompiled SQL statements (in bytes).	1048576
`synchronous`	• `TRUE` or `FULL`—synchronous • `FALSE` or `NORMAL`—regular • `OFF`—asynchronous	Data write mode.	`FULL`
`checkintegrity`	`quick | full | no`	Verify the integrity of the database image at Dr.Web Server startup.	`quick`
`autorepair`	`yes | no`	Automatically restore a corrupted database image at Dr.Web Server startup.	`no`
`mmapsize`	Positive integer.	Maximum number of bytes of the database file that is allowed to be mapped into the process address space at a time.	• for Unix-like OSs—10485760 • for Windows OS—0
`wal`	`yes | no`	Use Write-Ahead Logging.	`yes`
`wal-max-pages`	–	Maximum number of “dirty” pages after which the pages are written to disk.	1000

Attribute	Allowed values	Description	Default
wal-max-seconds	–	Maximum time to delay writing pages to disk (in seconds).	30

```
<pgsql dbname="drwcs" host="localhost" port="5432" options="" requiressl="" user=""
password="" temp_tablespaces="" default_transaction_isolation="" debugproto ="yes" />
```

Defines configuration of the external PostgreSQL database.

Attribute description:

Attribute	Allowed values	Description	Default
dbname	–	Database file name.	–
host	–	PostgreSQL server host or path to a Unix domain socket.	–
port	–	PostgreSQL server port number or the extension of a Unix domain socket file.	–
options	–	Command line parameters to send to the database server. For more details, see chapter 18 at https://www.postgresql.org/docs/9.1/libpq-connect.html	–
requiressl	• 1 | 0 (via the Control Center) • y | n • yes | no • on | off	Allow SSL connections only.	• 0 • y • yes • on
user	–	Database user name.	–
password	–	Database user password.	–
temp_tablespaces	–	Namespace for temporary tables.	–
default_transaction_isolation	• read uncommitted • read committed • repeatable read • serializable	Transaction isolation level.	read committed
debugproto	• yes | no • on | off	Enable debug logging of the DBMS operation.	• yes • on

- `<oracle connectionstring="" user="" password="" client="" prefetch-rows="0" prefetch-mem="0" />`

  Defines configuration of the external Oracle database.

  Attribute description:

Attribute	Allowed values	Description	Default
connectionstring	–	String with the Oracle SQL Connect URL or Oracle Net key-value pairs.	–
user	–	Registration name of the database user.	–
password	–	Database user password.	–
client	–	Path to the Oracle Instant Client for accessing the Oracle DB. Dr.Web Server is supplied with Oracle Instant Client version 11. If Oracle Servers of a later version are used or if the Oracle driver contains errors, you can download a corresponding driver from the Oracle website and set the path to the driver in this field.	–
prefetch-rows	0–65535	Number of rows to be prefetched when executing a query to the database.	0—use the value = 1 (database default)
prefetch-mem	0–65535	Memory allocated for rows to be prefetched when executing a query to the database.	0—unlimited

- `<odbc dsn="drwcs" user="" pass="" limit="" transaction="DEFAULT" />`

  Defines configuration of the connection to an external database via ODBC.

  Attribute description:

Attribute	Allowed values	Description	Default
dsn	–	ODBC data source name.	drwcs
user	–	Registration name of the database user.	drwcs
pass	–	Database user password.	drwcs
limit	Positive integer.	Reconnect to the DBMS after the specified number of transactions.	0—do not reconnect
transaction	• SERIALIZABLE—serializable	Transaction isolation level.	DEFAULT

Attribute	Allowed values	Description	Default
	• READ_UNCOMMITTED—read uncommitted data • READ_COMMITTED—read committed data • REPEATABLE_READ—repeatable read • DEFAULT—equal to ""—depends on the DBMS.	Some DBMS support READ_COMMITTED only.	

- `<mysql dbname="drwcs" host="localhost" port="3306" user="" password="" ssl="no" precompiledcache="" debug="no" />`

  Defines configuration of the external MySQL/MariaDB database.

  Attribute description:

Attribute	Allowed values	Description	Default
dbname	–	Database name.	drwcs
host	Either of the two.	Database server address for TCP/IP connections.	localhost
		Path to the UNIX socket file when using UDS. If not set, Dr.Web Server tries to locate the file in one of the standard mysqld directories.	/var/run/mysqld/
port	Either of the two.	Port number to connect to the database via TCP/IP.	3306
		UNIX socket file name when using UDS.	mysqld.sock
user	–	Registration name of the database user.	""
password	–	Database user password.	""
ssl	yes | any other string	Allow SSL connections only.	no
precompiledcache	Positive integer.	Cache size for precompiled SQL statements (in bytes).	1048576
debug	• yes | no • on | off	Enable debug logging of the DBMS operation.	• no • off

- <acl>

  Access Control List. Allows to configure restrictions for network addresses from which Dr.Web Agents, network installers and other (neighboring) Dr.Web Servers will be able to access Dr.Web Server.

  The <acl> element contains the following child elements for configuring restrictions for the corresponding connection types:

  - <install>—the list of restrictions on IP addresses from which Dr.Web Agent installers can connect to this Dr.Web Server.
  - <agent>—the list of restrictions on IP addresses from which Dr.Web Agents can connect to this Dr.Web Server.
  - <links>—the list of restrictions on IP addresses from which neighboring Dr.Web Servers can connect to this Dr.Web Server.
  - <discovery>—the list of restrictions on IP addresses from which broadcast queries can be received by the Dr.Web Server Detection Service.

  All child elements contain the same structure of nested elements that defines the following restrictions:

  - <priority mode="" />

    List priority. Allowed values for `mode`: `allow` or `deny`. When <priority mode="deny" /> is set, the <deny> list has a higher priority than the <allow> list. Addresses not included in either of the lists or included in both of them are denied. Only addresses included in the <allow> list and not included in the <deny> list are allowed.

  - <allow>

    List of TCP addresses from which access is allowed. The <allow> element contains one or several <ip address="" /> child elements to specify the allowed addresses in the IPv4 format and <ip6 address="" /> to specify the allowed addresses in the IPv6 format. The `address` attribute defines network addresses in the following format: *<IP address>*/[*<prefix>*].

  - <deny>

    List of TCP addresses from which access is denied. The <deny> element contains one or several <ip address="" /> child elements to specify the denied addresses in the IPv4 format and <ip6 address="" /> to specify denied addresses in the IPv6 format. The `address` attribute defines network addresses in the following format: *<IP address>*/[*<prefix>*].

- <scripts profile="" stack="" trace="" />

  Configuration of script profiling parameters.

  Attribute description:

Attribute	Allowed values	Description	Default
`profile`	• `yes`, • `no`.	Log information about Dr.Web Server script profiling. This parameter is used by technical support specialists and	`no`

Attribute	Allowed values	Description	Default
		developers. Changing this parameter unless necessary is not recommended.	
stack		Log information on Dr.Web Server script execution from a call stack. This parameter is used by technical support specialists and developers. Changing this parameter unless necessary is not recommended.	
trace		Log information on Dr.Web Server script execution tracing. This parameter is used by technical support specialists and developers. Changing this parameter unless necessary is not recommended.	

- **<lua-module-path>**

  Lua interpreter paths.

  

  The path order is important.

  The **<lua-module-path>** element contains the following child elements:

  - **<cpath root="" />**—path to the folder containing binary modules. Allowed values for `root`: `home` (default), `var`, `bin`, `lib`.
  - **<path value="" />**—path to the folder containing scripts. If it is not a child of the **<jobs>** or **<hooks>** elements, then it is used by both. Paths specified in the `value` attribute are relative to paths in the `root` attribute of the **<cpath>** element.
  - **<jobs>**—paths to tasks in the Dr.Web Server schedule.

    The **<jobs>** element contains one or several **<path value="" />** child elements to specify the path to the scripts folder.
  - **<hooks>**—paths for the user hooks of Dr.Web Server.

    The **<hooks>** element contains one or several **<path value="" />** child elements to specify the path to the scripts folder.

- **<transports>**

  Configuration of transport protocol parameters used by Dr.Web Server to connect to clients. Contains one or several **<transport** discovery="" ip="" name="" multicast="" multicast-group="" port="" /> child elements.

  Attribute description:

Attribute	Description	Obligatory	Allowed values	Default
discovery	Defines whether the Dr.Web Server Detection Service is	no, specified together with the `ip` attribute	`yes`, `no`	`no`

Attribute	Description	Obligatory	Allowed values	Default
	used or not.	only.		
ip | unix	Defines the family of used protocols (IP or Unix socket) and specifies the interface address.	yes	–	0.0.0.0 | –
name	Specifies the Dr.Web Server name for the Dr.Web Server Detection Service.	no	–	drwcs
multicast	Defines whether Dr.Web Server is in a multicast group or not. This setting is only available when the Dr.Web Server Detection Service (discovery parameter) is enabled: if it is disabled, multicast requests will not be listened to.	no, specified together with the ip attribute only.	yes, no	no
multicast-group	Specifies the address of the multicast group into which Dr.Web Server is included.	no, specified together with the ip attribute only.	–	• 231.0.0.1 • [ff18::231.0.0.1]
port	Port to listen.	no, specified together with the ip attribute only.	–	2193

- <protocols>

  List of disabled protocols. Contains one or several <protocol enabled="" name="" /> child elements.

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	• yes—protocol is enabled, • no—protocol is disabled.	Protocol usage mode.	no
name	• AGENT—protocol that allows interaction of Dr.Web Server with Dr.Web Agents. • MSNAPSHV—protocol that allows interaction of Dr.Web Server with the Microsoft NAP Validator system health check component. • INSTALL—protocol that allows interaction of Dr.Web Server with Dr.Web Agent installers.	Protocol name.	–

Attribute	Allowed values	Description	Default
	• CLUSTER—protocol for interaction between Dr.Web Servers in a cluster system. • SERVER—protocol that allows interaction of Dr.Web Server with other Dr.Web Servers.		

- **<plugins>**

  List of disabled extensions. Contains one or several **<plugin** enabled="" name="" /> child elements.

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	• yes—extension is enabled, • no—extension is disabled.	Extension usage mode.	no
name	• WEBMIN—Dr.Web Security Control Center extension for managing Dr.Web Server and the anti-virus network via the Control Center. • FrontDoor—Dr.Web Server FrontDoor extension that allows connecting a remote diagnostics utility to Dr.Web Server.	Extension name.	–

- **<license>**

  Licensing settings.

  The **<license>** element contains the following child elements:

  - **<limit-notify** min-count="" min-percent="" />

    Options for notifications on a low number of available licenses in the license key.

    Attribute description:

Attribute	Description	Default
min-count	Maximum number of remaining licenses at which the **Limitation on a number of licenses in the license key** notification will be sent.	3
min-percent	Maximum percentage of remaining licenses at which the **Limitation on a number of licenses in the license key** notification will be sent.	5

  - **<license-report** report-period="" active-stations-period="" />

    Options for the report on license usage.

    Attribute description:

Attribute	Description	Default
report-period	Frequency with which Dr.Web Server generates reports on license usage. If a report on license usage is created by a child Dr.Web Server, then this report is immediately sent to the main Dr.Web Server. Created reports are additionally sent each time Dr.Web Server is connected (including after a restart), as well as when the number of issued licenses on the main Dr.Web Server changes.	1440
active-stations-period	Period for counting the number of active stations to create a report on license usage. If set to 0, all stations will be included in the report regardless of their activity status.	0

- <exchange>

  Settings of license propagation between Dr.Web Servers.

  The <exchange> element contains the following child elements:

  - <expiration-interval value="" />
  - <prolong-preact value="" />
  - <check-interval value="" />

  Elements:

Element	Description	Default values, min
expiration-interval	**Validity period of donated licenses**—period of time for which licenses are issued from the key on this Dr.Web Server. The setting is used if the Dr.Web Server donates licenses to neighboring Dr.Web Servers.	1440
prolong-preact	**Period for accepted license renewal**—period of time before the expiration of the automatic license renewal period, when this Dr.Web Server starts requesting preliminary automatic renewal of accepted licenses The setting is used if Dr.Web Server accepts licenses from neighboring Dr.Web Servers.	60
check-interval	**License synchronization period**—time interval for synchronizing information about donated licenses between Dr.Web Servers.	1440

- <auth-flood count="" only-failed="" period="" />

  Authentication settings. If the specified number of attempts is exceeded, authentication will be impossible for a certain period of time.

  Attribute description:

Attribute	Allowed values	Description	Default
count	–	Number of authentication attempts.	5
only-failed	• yes—count only unsuccessful authentication attempts, • no—count both unsuccessful and successful authentication attempts.	Count only unsuccessful attempts.	yes
period	–	Time period during which authentication will be impossible.	60 seconds

- `<email from="" debug="" />`

  Parameters for sending emails from the Control Center, for example, as administrator notifications or when mailing installation packages to stations.

  Attribute description:

Attribute	Allowed values	Description	Default
from	–	Email address that will be set as the email sender.	drwcs@localhost
debug	• yes—use debug mode, • no—do not use debug mode.	Use debug mode to get a detailed log of the SMTP session.	no

  The `<email>` element contains the following child elements:

  - `<smtp server="" user="" pass="" port="" start_tls="" auth_plain="" auth_login="" auth_cram_md5="" auth_digest_md5="" auth_ntlm="" conn_timeout="" />`

    Configuration of SMTP server parameters for sending emails.

    Attribute description:

Attribute	Allowed values	Description	Default
server	–	Address of the SMTP server that is used to send email.	127.0.0.1
user	–	Name of the SMTP server user if the SMTP server requires authentication.	–
pass	–	Password of the SMTP server user if the SMTP server requires authentication.	–

Attribute	Allowed values	Description	Default
port	Positive integer.	Port of the SMTP server that is used to send email.	25
start_tls	• yes—use this authentication type, • no—do not use this authentication type.	Encrypt data transfer. Switching to secure connection is performed by using the STARTTLS command. Port 25 is used for the connection by default.	yes
auth_plain		Use *plain text* authentication on the mail server.	no
auth_login		Use *LOGIN* authentication on the mail server.	no
auth_cram_md5		Use *CRAM-MD5* authentication on the mail server.	no
auth_digest_md5		Use *DIGEST-MD5* authentication on the mail server.	no
auth_ntlm		Use *AUTH-NTLM* authentication on the mail server.	no
conn_timeout	Positive integer.	Connection time-out for the SMTP server.	180

- `<ssl enabled="" verify_cert="" ca_certs="" />`

  Configuration of SSL traffic encryption parameters for sending emails.

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	• yes—use SSL, • no—do not use SSL.	Use of SSL encryption.	no
verify_cert	• yes—check the SSL certificate, • no—do not check the SSL certificate.	Validate the SSL certificate of the mail server.	no
ca_certs	–	Path to the root SSL certificate of Dr.Web Server.	–

- `<track-epidemic enabled="" aggregation-period="" check-period="" threshold="" most-active="" />`

  Configuration of parameters for tracking malware outbreaks in the network.

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	yes | no	Enables monitoring of multiple station infection events and allows to send summary notifications to the administrator.	yes
aggregation-period	Positive integer.	Time period in seconds after sending a notification about an outbreak during which single notifications about infected stations will not be sent.	300
check-period		Time period in seconds during which a specified number of messages on infected stations must be received to send a notification about an outbreak.	3600
threshold		Number of messages on infections that must be received within a specified time period for Dr.Web Server to send a summary notification about an outbreak for all infection cases (the **Epidemic in the network** notification) to the administrator.	100
most-active		Number of the most frequently occurring threats that must be included in the epidemic report.	5

- `<track-hips-storm enabled="" aggregation-period="" check-period="" threshold="" most-active="" />`

  Configuration of parameters for tracking multiple events of the Preventive protection component.

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	yes | no	Enables monitoring of multiple events of Preventive protection and allows to send summary notifications to the administrator.	yes
aggregation-period	Positive integer.	Time interval in seconds after sending a summary report on Preventive protection events during which no single event notifications are sent.	300
check-period		Time period in seconds during which a specified number of Preventive protection events must occur for a summary report to be sent.	3600
threshold		The number of Preventive protection events that must be received in a specified time period, so that Dr.Web Server may send a single summary report on these events (the **Summary report of Preventive protection** notification) to the administrator.	100
most-active		Number of the most active processes that have performed a suspicious action to be included in	5

Attribute	Allowed values	Description	Default
		the Preventive protection report.	

- `<track-appctl-storm enabled="" aggregation-period="" check-period="" threshold="" most-active="" />`

  Configuration of parameters for tracking multiple events of the Application Control component.

  Attribute description:

Attribute	Allowed values	Description	Default
`enabled`	`yes | no`	Enables monitoring of multiple events of Application Control and allows to send summary notifications to the administrator.	`yes`
`aggregation-period`	Positive integer.	Time period in seconds after sending a summary report on processes blocked by Application Control during which notifications about blocked processes will not be sent.	300
`check-period`		Time period in seconds during which a specified number of processes must be blocked for a summary report to be sent.	3600
`threshold`		Number of processes blocked by Application Control in a specified period of time so that Dr.Web Server can send a single summary report on these events to the administrator (the **Large number of blocks by the Application Control detected** notification).	100
`most-active`		Number of the most frequently triggered Application Control profiles to be included in multiple blocked processes notification.	5

- `<track-disconnect enabled="" aggregation-period="" check-period="" single-alert-threshold="" summary-alert-threshold="" min-session-duration="" />`

  Configuration of parameters for tracking multiple abnormally terminated connections to clients.

  Attribute description:

Attribute	Allowed values	Description	Default
`enabled`	`yes | no`	Enables monitoring of abnormally terminated connections to clients and sending corresponding notifications to the administrator.	`yes`
`aggregation-period`	Positive integer.	Time period in seconds after sending a notification about multiple terminated connections during	300

Attribute	Allowed values	Description	Default
		which notifications about single terminated connections are not sent.	
`check-period`		Time period in seconds during which a specified number of connections to clients must be terminated for the corresponding notification to be sent.	3600
`single-alert-threshold`		Minimum number of connections to a single address that must be terminated during the count period for a single abnormally terminated connection notification to be sent (the **Connection terminated abnormally** notification).	10
`summary-alert-threshold`		Minimum number of connections that must be terminated during the count period for a summary notification about multiple abnormally terminated connections to be sent (the **Large number of abnormally terminated connections detected** notification).	1000
`min-session-duration`		If the duration of a terminated connection to a client is less than the specified value, then, when a specified number of connections is reached, a notification about single terminated connections (the **Connection terminated abnormally** notification) will be sent regardless of the count period. The connection must not be terminated later by longer connections, and the notification about multiple abnormally terminated connections must not be sent (the **Large number of abnormally terminated connections detected** notification).	300

- `<default-lang value="" />`

  Default language used by Dr.Web Server components and systems if the language settings could not be obtained from the Dr.Web Server database. In particular, it is used by Dr.Web Security Control Center and the administrator notification system when the database is corrupted and the language settings cannot be obtained.

F2. Dr.Web Security Control Center Configuration File

The `webmin.conf` Dr.Web Security Control Center configuration file is presented in the XML format and located in the `etc` subfolder of the Dr.Web Server installation folder.

Description of Dr.Web Security Control Center configuration file parameters:

- `<version value="">`

  Current version of Dr.Web Server.

- `<server-name value=""/>`

  The name of Dr.Web Server.

  Parameter is specified in the following format:

  <Dr.Web Server IP address or DNS name> `[:` *<port>* `]`

  If the Dr.Web Server address is not specified, computer name returned by the operating system or the Dr.Web Server network address: domain name, if available, otherwise—IP address are used.

  If the port number is not specified, the port from a request is used (e.g., for requests to Dr.Web Server from the Control Center or via the **Web API**). Particularly, for the requests from the Control Center it is the port specified in the address line for connection of the Control Center to Dr.Web Server.

- `<document-root value=""/>`

  Path to web pages root folder. Default is `value="webmin"`.

- `<ds-modules value=""/>`

  Path to modules folder. Default is `value="ds-modules"`.

- `<threads value=""/>`

  Number of parallel requests processed by the web server. This parameter affects server performance. It is not recommended to change this parameter without need.

- `<io-threads value=""/>`

  Number of threads serving data transmitted in network. This parameter affects the Dr.Web Server performance. It is not recommended to change this parameter without need.

- `<compression value="" max-size="" min-size=""/>`

  Traffic compression settings for data transmission over a communication channel with the web server via HTTP/HTTPS.

  Attributes description:

Attribute	Description	Default
`value`	Data compression level from 1 to 9, where the 1 is minimal level and the 9 is maximal compression level.	9
`max-size`	Maximal size of HTTP responses which will be compressed. Specify the 0 value to disable limitation on maximal size of HTTP responses to be compressed.	51200 KB
`min-size`	Minimal size of HTTP responses which will be compressed. Specify the 0 value to disable limitation on minimal size of HTTP responses to be compressed.	32 bytes

- `<keep-alive timeout="" send-rate="" receive-rate=""/>`

  Keep HTTP session active. Allows to establish permanent connection for requests via the HTTP v. 1.X.

  Attributes description:

Attribute	Description	Default
`timeout`	HTTP session time-out. For persistent connections, Dr.Web Server releases the connection, if there are no requests received from a client during specific time slot.	15 sec.
`send-rate`	Minimal acceptable data send rate. If outgoing network speed is lower than this value, connection will be rejected. Specify 0 to ignore this limit.	1024 Bps
`receive-rate`	Minimal acceptable data receive rate. If incoming network speed is lower than this value, connection will be rejected. Specify 0 to ignore this limit.	1024 Bps

- `<buffers-size send="" receive=""/>`

  Configuration of buffers sizes for sending and receiving data.

  Attributes description:

Attribute	Description	Default
`send`	Size of buffers used when sending data. This parameter affects server performance. It is not recommended to change this parameter without need.	8192 bytes
`receive`	Size of buffers used when receiving data. This parameter affects server performance. It is not recommended to change this parameter without need.	2048 bytes

- `<max-request-length value=""/>`

  Maximum allowed size of HTTP request in KB.

- `<xheaders>`

  Configuration parameter that lets you add custom HTTP headers. Three headers present by default are intended to protect from network attacks:

  - `<xheader name="X-XSS-Protection" value="1; mode=block"/>`

    The header controls web browser behavior if it detects any code inlined into attacked web page (the so-called “XSS attack”). Allowed values:

Value	Browser behavior
0	XSS filtering is disabled.
1	XSS filtering is enabled. Web browser will delete inline code if it detects an XSS attack.

Value	Browser behavior
`1; mode=block`	XSS filtering is enabled. Web browser will not render a compromised web page if it detects an XSS attack. This value is used by default.
`1; report=<`*network-address*`>`	XSS filtering is enabled. Web browser will delete inline code and report to specified address if it detects an XSS attack. This value is supported in Chromium-based web browsers only.

- `<xheader name="X-Content-Type-Options" value="nosniff"/>`

  When the default value (`nosniff`) is used, the header prevents the web browser from independently guessing the MIME type regardless of the type specified in `Content-Type` and processing it accordingly (the so-called "MIME sniffing").

- `<xheader name="X-Frame-Options" value="SAMEORIGIN"/>`

  The header controls web browser behavior if it detects an attempt to inline a web page into a frame (the so-called "clickjacking"). Allowed values:

Value	Browser behavior
`DENY`	Prevents web browser from rendering a web page in a frame.
`SAMEORIGIN`	Allows web browser to render a web page in a frame as long as the page and the frame both have the same origin (domain, port, and protocol). This value is used by default.
`ALLOW-FROM <`*network-address*`>`	Allows web browser to render a web page in a frame only if the web page is located at a specified address.

- `<reverse-resolve enabled=""/>`

  Replace IP address with DNS names of computers in the Dr.Web Server log file. Allowed values of enabled: `yes` or `no`.

- `<script-errors-to-browser enabled=""/>`

  Show script errors in browser (error 500). This parameter is used by technical support and developers. It is not recommended to change this parameter without need.

- `<trace-scripts enabled=""/>`

  Enable scripts tracing. This parameter is used by technical support and developers. It is not recommended to change this parameter without need. Allowed values of `enabled`: `yes` or `no`.

- `<profile-scripts enabled="" stack=""/>`

  Profiling configuration. Performance is measuring—execution time of functions and scripts of the web server. This parameter is used by technical support and developers. It is not recommended to change this parameter without need.

  Attributes description:

Attribute	Allowed values	Description
enabled	• yes—enable profiling, • no—disable profiling.	Scripts profiling mode.
stack	• yes—log data, • no—do not log data.	Logging mode of information on profiling (function parameters and returned values) into the Dr.Web Server log.

- `<abort-scripts enabled=""/>`

  Allow aborting of scripts execution if the connection was aborted by client. This parameter is used by technical support and developers. It is not recommended to change this parameter without need. Allowed values of `enabled`: `yes` or `no`.

- `<search-localized-index enabled=""/>`

  Use localized versions of pages. If the flag is set, server searches for localized version of specified page according to the language priority which is set in the `Accept-Language` field of client header. Allowed values of `enabled`: `yes` or `no`.

- `<default-lang value=""/>`

  Language of documents returned by the web server in the absence of the `Accept-Language` header in the HTTP request. The `value` attribute is the ISO language code. Default is `ru`.

- `<ssl certificate="" private-key="" keep-alive="" ciphers="" />`

  SSL certificate settings.

  Description of attributes:

Attribute	Description	Allowed values	Default
certificate	Path to SSL certificate file.	-	certificate.pem
private-key	Path to SSL private key file.	-	private-key.pem
keep-alive	Use keep-alive SSL connection. Older browsers may not work properly with regular SSL connections. Disable this parameter, if you have problems with SSL protocol.	• yes, • no.	yes
ciphers	List and settings of ciphers being used.	See a detailed description in the OpenSSL documentation, section **CIPHER LIST FORMAT**.	HIGH:!aNULL:!RC4:@STRENGTH

- `<listen>`

  Configure parameters to listen for network connections.

  The `<listen />` element contains the following child elements:

  - `<insecure />`

    The list of interfaces to listen for accepting connections via the HTTP protocol for unsecured connections. Default port is 9080.

    The `<insecure />` element contains one or several `<endpoint address=""/>` child elements to specify allowed addresses in the IPv4 or IPv6 format. In the `address` attribute, network addresses are specified in the following format: <*Protocol*>://<*IP address*>.

  - `<secure />`

    The list of interfaces to listen for accepting connections via the HTTPS protocol for secured connections. Default port is 9081.

    The `<secure />` element contains one or several `<endpoint address=""/>` child elements to specify allowed addresses in the IPv4 or IPv6 format. In the `address` attribute, network addresses are specified in the following format: <*Protocol*>://<*IP address*>.

- `<access>`

  Access control lists. Allow to configure limitations on network addresses to listen for accepting incoming HTTP and HTTPS requests by the web server.

  The `<access>` element contains the following child elements, which configuring limitations for corresponding connection types:

  - `<secure priority="">`

    The list of interfaces to listen for accepting secured connections via the HTTPS protocol. Default port is 9081.

    Attributes description:

Attribute	Allowed values	Description	Default
`priority`	`allow`	Allowance priority for HTTPS—addresses not included in any of the lists (or included into both), are allowed.	`deny`
	`deny`	Denial priority for HTTPS—addresses not included in any of the lists (or included into both), are denied.	

    The `<secure />` element contains one or several following child elements: `<allow address=""/>` and `<deny address=""/>`.

    Elements description:

Element	Description	Default value of address attribute
`allow`	Addresses which are allowed to access via the HTTPS protocol for secured connections.	`tcp://127.0.0.1`

Element	Description	Default value of address attribute
deny	Addresses which are denied to access via the HTTPS protocol for secured connections.	-

- <insecure priority="">

  The list of interfaces to listen for accepting unsecured connections via the HTTP protocol. Default port is 9080.

  Attributes description:

Attribute	Allowed values	Description	Default
priority	allow	Allowance priority for HTTP—addresses not included in any of the lists (or included into both), are allowed.	deny
	deny	Denial priority for HTTP—addresses not included in any of the lists (or included into both), are denied.	

  The <insecure /> element contains one or several following child elements: <allow address=""/> and <deny address=""/>.

  Elements description:

Element	Description	Default value of address attribute
allow	Addresses which are allowed to access via the HTTP protocol for unsecured connections.	tcp://127.0.0.1
deny	Addresses which are denied to access via the HTTP protocol for unsecured connections.	-

- <security-through-obscurity="" />

  Configuration of security parameters that enhance security by hiding or deliberately misrepresenting some data.

  The <security-through-obscurity> element contains the following child elements:

  - <server-header enabled="" />
  - <lower-case-uri enabled="" />
  - <hacker-misleading enabled="" />

  Attribute description:

Attribute	Allowed values for enabled	Description	Default value for enabled
`server-header`	`yes | no`	Allows to hide the string with server details (Dr.Web Server version, OS version, loaded libraries), which makes it more difficult to exploit known vulnerabilities. Corresponds to the **Return detailed header** flag in the web server configuration.	`no`
`lower-case-uri`	`yes | no`	Converts all URIs in HTTP requests to lowercase. Corresponds to the **Convert URI to lowercase** flag in the web server configuration.	`no`
`hacker-misleading`	`yes | no`	When enabled, returns fake `passwd`, `hosts`, etc. in response to requests for files such as `/etc/passwd`, `/etc/hosts`, etc. (thus negating the Path/Directory Traversal vulnerability). There is no corresponding option in the current web server configuration.	`yes`

F3. Download.conf Configuration File

The download.conf file purposes:

1. During creating and operating of Dr.Web Servers cluster system, the file allows to distribute the load between the Dr.Web Servers of a clusters when connecting a large number of new stations.
2. If a custom port is used at Dr.Web Server, the file allows to specify this port during generating installation file of Dr.Web Agent.

The `download.conf` file is used during generating the installation file for a new station of the anti-virus network. Parameters of this file allows to specify address of Dr.Web Server and the port, which are used to connect the Dr.Web Agent Installer to Dr.Web Server, in the following format:

```
download = { server = '<Dr.Web_Server_Address>'; port = <port_number> }
```

where:

- *<Dr.Web_Server_Address>*—domain name or IP address of Dr.Web Server.

It is recommended that you use the name of Dr.Web Server in the FQDN format as the Dr.Web Server address.

During generating of the Dr.Web Agent installation file, Dr.Web Server address is taken from the `download.conf` file first. If the Dr.Web Server address is not specified in the `download.conf` file, when value of the `ServerName` parameter from the `webmin.conf` file is taken. Otherwise, the name of the computer, returned by an operating system is used.

- *<port_number>*—port to connect the Dr.Web Agent Installer to Dr.Web Server.

  If the port is not specified in the `download.conf` file, 2193 port is used by default (sets in the **Administration** → **Dr.Web Server configuration** → the **Network** tab → the **Transport** tab in the Control Center).

By default, the `download.conf` configuration file is missing. This file appears in the `etc` subdirectory of the Dr.Web Server root directory only after saving new settings in **Administration** → **Dr.Web Server Configuration** → **Network** tab → **Download** tab.

F4. Dr.Web Proxy Server Configuration File

The `drwcsd-proxy.conf` configuration file of Dr.Web Proxy Server is in the XML format and located in the following folder:

- Windows OS: `C:\ProgramData\Doctor Web\drwcs\etc`
- Linux OS: `/var/opt/drwcs/etc`
- FreeBSD OS: `/var/drwcs/etc`

Description of Dr.Web Proxy Server configuration file parameters:

- `<listen spec="">`

  The `<drwcsd-proxy />` root element contains one or several mandatory `<listen />` elements that define the basic settings of Dr.Web Proxy Server for receiving connections.

  The `<listen />` element contains one mandatory attribute `spec`, the attributes of which define an interface to "listen" on for incoming client connections, and whether the `discovery` mode is enabled on that interface.

  The `spec` element attributes:

Attribute	Mandatory	Allowed values	Description	Default
`ip | unix`	yes	—	Protocol type for receiving incoming connections. The address that Dr.Web Proxy server listens on is specified as a parameter.	`0.0.0.0 | -`
`port`	no	—	Port on which Dr.Web Proxy Server listens.	`2193`
`discovery`	no	`yes, no`	Dr.Web Proxy Server discovery mode. Allows discovery of an available Dr.Web Proxy Server during multicast requests.	`yes`

Attribute	Mandatory	Allowed values	Description	Default
multicast	no	yes, no	Network "listening" mode for receiving multicast requests by Dr.Web Proxy Server.	yes
multicast-group	no	—	Multicast group where Dr.Web Proxy Server is located.	231.0.0.1 [ff18::231.0.0.1]

Depending on the protocol, the list of non-mandatory properties in the spec attribute may vary.

The list of non-mandatory properties, which can be set (+) or cannot be set (-) in the spec attribute, depending on the protocol:

Protocol	Attribute availability			
	port	discovery	multicast	multicast-group
ip	+	+	+	+
unix	+	-	-	-

The **discovery** mode must be enabled explicitly even if the **multicast** mode is already enabled.

The forwarding algorithm if a list of Dr.Web Servers is available is given in the **Administrator Manual**.

▫ <compression mode="" level="">

The <compression /> element is nested in the <listen /> element, it defines compression parameters for the connection between a client and Dr.Web Proxy Server.

Attribute description:

Attribute	Allowed values	Description	Default
mode	yes	Compression enabled.	possible
	no	Compression disabled.	
	possible	Compression possible.	
level	integer from 1 to 9	Compression level. Only for the connection between a client and Dr.Web Proxy Server.	8

- `<encryption mode="">`

The `<encryption />` element is nested in the `<listen />` element, it defines encryption parameters for the connection between a client and Dr.Web Proxy Server.

Attribute description:

Attribute	Allowed values	Description	Default
mode	yes	Encryption enabled.	possible
	no	Encryption disabled.	
	possible	Encryption possible.	

- `<forward to="" master="">`

Specifies the settings for the redirection of incoming connections to Dr.Web Server. The `<forward />` element is mandatory. Several `<forward />` elements with different attribute values can be specified.

Attribute description:

Attribute	Allowed values	Description	Mandatory
to	An address is specified according to the The Specification of Network Addresses: • *<protocol>*://*<interface>*:*<port-number>*—use IPv4 and IPv6. • *<protocol>*://(*<interface>*):*<port-number>*—use only IPv4. • *<protocol>*://[*<interface>*]:*<port-number>*—use only IPv6.	Address of Dr.Web Server to which the connection is redirected.	yes
master	• yes—Dr.Web Server can be used to manage Dr.Web Proxy Server in all scenarios. • no—Dr.Web Server cannot used to manage Dr.Web Proxy under any conditions. • possible—Dr.Web Server can be used to manage Dr.Web Proxy Server only if there are no other Dr.Web Servers that	The attribute defines if Dr.Web Proxy Server settings can be remotely edited via the Control Center of Dr.Web Server specified in the to attribute. You can assign managing roles to any number of Dr.Web Servers (set master="yes"); Dr.Web Proxy Server connects to all managing Dr.Web Servers in their order in the settings until it gets the first valid (not empty) configuration. You can also choose not to assign the managing role to any Dr.Web Server (set the	no

Attribute	Allowed values	Description	Mandatory
	have an explicit management role (with their `master` attribute set to `yes`).	`master="no"`). In this case, Dr.Web Proxy Server parameters (including the assignment of managing Dr.Web Servers) can be configured only locally in Dr.Web Proxy Server configuration file.	

If the `master` attribute is not specified for Dr.Web Server, it is assumed by default that `master="possible"`.

The `master` attribute is not defined for any Dr.Web Server in the configuration file created by the installer during Dr.Web Proxy Server installation.

- `<compression mode="" level="">`

  If the `<compression />` element is nested in the `<forward />` element, it defines compression parameters for the connection between Dr.Web Server and Dr.Web Proxy Server. The attributes are the same as described above.

- `<encryption mode="">`

  If the `<encryption />` element is nested in the `<forward />` element, it defines encryption parameters for the connection between Dr.Web Server and Dr.Web Proxy Server. The attributes are the same as described above.

- `<update-bandwidth value="" queue-size="">`

The `<update-bandwidth />` element specifies the bandwidth limit for the delivery of updates from Dr.Web Server to clients and the number of clients that can download updates at the same time.

Attribute description:

Attribute	Allowed values	Description	Default
`value`	• bandwidth in KB/sec, • `unlimited`	Maximum total bandwidth for update delivery.	`unlimited`
`queue-size`	• positive integer, • `unlimited`	Maximum number of simultaneous update delivery sessions allowed on Dr.Web Server. When the limit is reached, update requests from Dr.Web Agents are placed in the waiting queue. The waiting queue size is unlimited.	`unlimited`

- `<bandwidth value="" time-map="" />`

  The `<update-bandwidth />` element may have one or several nested `<bandwidth />` elements. This element specifies a bandwidth limit for data transfer for the specified time period.

  Attributes description:

Attribute	Allowed values	Description	Default
value	• bandwidth in KB/sec, • unlimited	Maximum total bandwidth for transferring Dr.Web Agent updates.	unlimited
time-map	—	Mask that specifies the time period when the limitations are applied.	—

The time-map attribute value is set automatically once the corresponding setting is configured in the Control Center web interface (see **Administrator Manual**, p. Remote Configuration of Dr.Web Proxy Server (read online). You cannot specify the time-map attribute manually in the configuration file.

- <install-bandwidth value="" queue-size="">

The <install-bandwidth> element specifies the bandwidth limit for data transfer when installing Dr.Web Agents and the number of clients that can download the installation data at the same time.

Attribute description:

Attribute	Allowed values	Description	Default
value	• bandwidth in KB/sec, • unlimited	Maximum total bandwidth for transferring Dr.Web Agent installation data.	unlimited
queue-size	• positive integer, • unlimited	Maximum allowable number of simultaneous Dr.Web Agent installation sessions allowed on Dr.Web Server. When the limit is reached, requests from Dr.Web Agent are placed in the waiting queue. The waiting queue size is unlimited.	unlimited

  - <bandwidth value="" time-map="">

  The <install-bandwidth> element may have one or several nested <bandwidth /> elements. This element specifies the bandwidth limit for data transfer for the specified time period.

  Attribute description:

Attribute	Allowed values	Description	Default
value	• bandwidth in KB/sec, • unlimited	Maximum total bandwidth for transferring Dr.Web Agent installation data.	unlimited
time-map	—	Mask that specifies the time period when the limitations are applied.	—

The `time-map` attribute value is set automatically once the corresponding setting is configured in the Control Center web interface (see **Administrator Manual**, p. Remote Configuration of Dr.Web Proxy Server (read online)). You cannot specify the `time-map` attribute manually in the configuration file.

- `<cache enabled="">`

  Configure the settings of Dr.Web Proxy Server repository cache.

  Attribute description:

Attribute	Allowed values	Description	Default
`enabled`	`yes | no`	Enables/disables caching.	`yes`

  The `<cache>` element contains the following nested elements:

Element	Allowed values	Description	Default
`<maximum-revision-queue size="">`	positive integer	Number of stored revisions.	3
`<clean-interval value="">`	positive integer	Time period to keep old revisions in minutes.	60
`<unload-interval value="">`	positive integer	Time period to keep files in memory in minutes.	10
`<repo-check mode="">`	`idle | sync`	Check cache integrity either at startup (may take a long time) or in the background.	`idle`

  - `<synchronize enabled="" schedule="">`

    Settings for synchronization of Dr.Web Proxy Server and Dr.Web Server repositories.

    Attribute description:

Attribute	Allowed values	Description	Default
`enabled`	`yes | no`	Enables/disables repository synchronization.	`yes`
`schedule`	—	Schedule for synchronization of the specified products.	—

    The `schedule` attribute value is set automatically once the corresponding setting is configured in the Control Center web interface (see **Administrator Manual**, p. Remote Configuration of Dr.Web Proxy Server (read online)). You cannot specify the `schedule` attribute manually in the configuration file.

    The `<product name="" />` nested elements contain products for the synchronization:

    - 05-drwmeta—Dr.Web Server security data,
    - 10-drwbases—virus databases,
    - 10-drwgatedb—SpIDer Gate databases,

- 10-drwspamdb—Anti-spam databases,
- 10-drwupgrade—Dr.Web Updater,
- 15-drwhashdb—known hashes of threats,
- 20-drwagent—Dr.Web Agent for Windows,
- 20-drwandroid11—virus databases for Android,
- 20-drwcs—Dr.Web Server,
- 20-drwunix—content filter databases for UNIX,
- 25-drwcsdoc—documentation,
- 40-drwproxy—Dr.Web Proxy Server,
- 70-drwextra—Dr.Web enterprise products,
- 70-drwutils—Dr.Web administrative utilities,
- 80-drwnews—Doctor Web News.

- `<events enabled="" schedule="">`

  Settings for caching events received from Dr.Web Agents.

  Attribute description:

Attribute	Allowed values	Description	Default
`enabled`	`yes | no`	Enables/disables caching. If enabled, the events are sent to Dr.Web Server according to schedule. If caching is disabled, events will be sent to Dr.Web Server immediately after they are received by Dr.Web Proxy Server.	`yes`
`schedule`	—	Schedule according to which events from Dr.Web Agents will be transmitted.	—

  The `schedule` attribute value is set automatically once the corresponding setting is configured in the Control Center web interface (see **Administrator Manual**, p. Remote Configuration of Dr.Web Proxy Server (read online). You cannot specify the `schedule` attribute manually in the configuration file.

- `<update enabled="" schedule="">`

  Dr.Web Proxy Server automatic update settings.

  For the automatic update, if the synchronization is enabled, Dr.Web Proxy Server updates are downloaded from Dr.Web Server according to the synchronization schedule (see above) and are installed according to the update schedule (by default, with no time limitations). If the synchronization is disabled, then updates are downloaded and installed according to the update schedule (by default, with no time limitations).

  Attributes description:

Attribute	Allowed values	Description	Default
enabled	yes | no	Enables/disables automatic updates.	yes
schedule	—	Schedule for downloading and installing updates (if synchronisation is not set).	—

The schedule attribute value is set automatically once the corresponding setting is configured in the Control Center web interface (see **Administrator Manual**, p. Remote Configuration of Dr.Web Proxy Server (read online). You cannot specify the schedule attribute manually in the configuration file.

By default, the automatic update is allowed with no time limitations.

- <core-dump enabled="" maximum="">

  The collection mode and number of memory dumps when an SEH exception occurs.

  Memory dump configuration is available for OS Windows only.

  To collect a memory dump, the OS must contain the dbghelp.dll library.

  The dump is written to the following folder: %APPDATA%\Doctor Web\drwcsd-proxy\dump\

  Attribute description:

Attribute	Allowed values	Description	Default
enabled	yes | no	Enables/disables collection of memory dumps.	yes
maximum	positive integer	Maximum number of memory dumps. The oldest are deleted.	10

- <dns>

  DNS settings.

  <timeout value="">

  Time-out in seconds for resolving DNS direct/reverse queries. Leave the value blank to disable restriction on wait time until the end of the resolution

  <retry value="">

  Maximum number of DNS retries when DNS query resolution fails.

  <cache enabled="" negative-ttl="" positive-ttl="">

  Time for storing responses from DNS server in the cache.

  Attribute description:

Attribute	Allowed values	Description
enabled	• yes—store responses in the cache, • no—do not store responses in the cache.	Enables/disables caching of responses.
negative-ttl	—	Cache Time To Live (TTL) for negative responses from the DNS server in minutes.
positive-ttl	—	Cache Time To Live (TTL) for positive responses from the DNS server in minutes.

<servers>

List of DNS servers, that replaces the default system list. Contains one or several <server address=""> nested elements, whose address parameter defines IP address of the server.

<domains>

List of private TLD suffixes, that replaces the default system list. Contains one or several <domain name=""> nested elements, whose name parameter defines the domain name.

F5. Repository Loader Configuration File

The drwreploader.conf Repository Loader configuration file is presented in the XML format and located in the etc subfolder of the Dr.Web Server installation folder.

To use the configuration file

- For the console utility, the path to the file must be specified in the --config switch.
- For the graphical utility, the file must reside in the utility folder. If the utility is ran without configuration file, it will be created in the utility folder and will be used at next launches.

Description of the Repository Loader configuration file parameters:

- <mode value="" path="" archive="" key="">

  Attributes description:

Attribute	Description	Allowed values
value	Updates loading mode: • repository—repository is downloaded in the Dr.Web Server repository format. Loaded files can be directly imported via the Control Center as the Dr.Web Server repository updates. • mirror—repository is downloaded in the GUS updates zone format. Loaded files can be placed on the updates mirror in your local network. Further, Dr.Web Servers can be configures to receive updates directly from this updates mirror containing the last version of the repository but not from the GUS servers.	repository | mirror

Attribute	Description	Allowed values
path	The folder for downloading the repository.	–
archive	Pack downloaded repository into a zip archive automatically. This option allows to get prepared archive file for import downloaded repository archive to Dr.Web Server via the Control Center, for the **Administrating** → **Repository Content** section.	yes | no
key	Dr.Web license key file. Instead of a license key you can specify only MD5 hash of a license key, which you can view in the Control Center in the **Administration** → **License Manager** section.	–

- `<log path="" verbosity="" rotate="">`

  Repository Loader log settings.

  Attributes description:

Attribute	Description	Allowed values
path	Path to the log file.	–
verbosity	Log level of detail. TRACE3 is by default.	ALL, DEBUG3, DEBUG2, DEBUG1, DEBUG, TRACE3, TRACE2, TRACE1, TRACE, INFO, NOTICE, WARNING, ERROR, CRIT. The ALL and DEBUG3 values are synonyms.
rotate	Repository Loader log rotation mode in the format: *<N><f>,<M><u>*. Same as the Dr.Web Server log rotation. By default, it is 10,10m, which means storing of 10 files 10 megabytes each, use compression.	–

- `<update url="" proto="" cdn="" update-key="" version="">`

  General repository loading settings.

  Attributes description:

Attribute	Description	Allowed values
url	A GUS servers folder where updates of Dr.Web products are located.	–
proto	The protocol type to receive updates from update servers. For all protocols, updates are downloaded according to the settings of the GUS servers list.	http | https | ftp | ftps | sftp | scp | file

Attribute	Description	Allowed values
cdn	Allow downloading repository from GUS via Content Delivery Network	yes | no
update-key	Path to a public key or to a folder with a public key to validate the signature of updates that are loaded from GUS. The update-key-*.upub public keys to validate updates can be found on Dr.Web Server in the etc folder.	–
version	The Dr.Web Server version to which the updates must be loaded.	–

▫ <servers>

Update servers list. GUS servers are listed in the order the utility contacts them when downloading the repository.

Contains the <server> child elements with update servers.

▫ <auth user="" password="">

User credentials to authenticate on updates server, if the updates server requires authorization.

Attributes description:

Attribute	Description
user	User name at updates server.
password	Password at updates server.

▫ <proxy host="" port="" user="" password="" />

Parameters for connecting to the GUS via the proxy server.

Attributes description:

Attribute	Description
host	The network address of the proxy server.
port	The port number of the proxy server. Default is 3128.
user	User name on the proxy server if used proxy server requests authorization.
password	Password on the proxy server if used proxy server requests authorization.

▫ <ssl cert-mode="" cert-file="">

The type of SSL certificates that will be automatically accepted. This option is used only for secure protocols that support encrypting.

Attributes description:

Attribute	Description	Allowed values
cert-mode	Certificated to accept automatically.	▫ any—accept all certificates, ▫ valid—accept only valid certificates, ▫ drweb—accept only Dr.Web certificates. ▫ custom—accept user-defined certificates.
cert-file	Path to the cert file.	–

▫ <ssh mode="" pubkey="" prikey="">

The type of the authorization on the update server when accessing by SCP/SFTP.

Attributes description:

Attribute	Description	Allowed values
mode	Authorization type.	▫ pwd—authorization using a password. A password is set in the <auth /> tag. ▫ pubkey—authorization using a public key. A public key is set in the pubkey attribute or extracted from the private key specified in the prikey.
pubkey	Public SSH key	–
prikey	Private SSH key	–

- <products>

Loading products settings.

▫ <product name="" update="">

Each product settings separately.

Attributes description:

Attribute	Description	Allowed values
name	Product name.	• 05-drwmeta—Dr.Web Server security data, • 10-drwbases—virus databases, • 10-drwgatedb—SplDer Gate bases, • 10-drwspamdb—Anti-spam bases, • 10-drwupgrade—Dr.Web Updater, • 15-drwhashdb—known hashes of threats, • 20-drwagent—Dr.Web Agent for Windows, • 20-drwandroid11—virus databases for Android, • 20-drwcs—Dr.Web Server, • 20-drwunix—content filter databases for UNIX, • 25-drwcsdoc—documentation, • 40-drwproxy—Dr.Web Proxy Server,

Attribute	Description	Allowed values
		• `70-drwextra`—Dr.Web enterprise products, • `70-drwutils`—Dr.Web administrative utilities, • `80-drwnews`—Doctor Web News.
`update`	Enable this product downloading.	`yes | no`

- **<schedule>**

  The schedule to receive updates periodically. At this, you do not have to launch the utility manually, the repository downloading performed automatically according to the specified time slots.

  - **<job period="" enabled="" min="" hour="" day="" />**

    Settings of scheduled loading execution.

Attribute	Description	Allowed values
`period`	Periodicity for loading task execution.	• `every_n_min`—every N minutes, • `hourly`—every hour, • `daily`—every day, • `weekly`—every week.
`enabled`	Downloading task is enabled.	`yes | no`
`min`	Minute to execute the task.	integers from 0 to 59
`hour`	Hour to execute the task. Relevant for `daily` and `weekly`.	integers from 0 to 23
`day`	Day to execute the task. Relevant for `weekly`.	• `mon`—Monday, • `tue`—Tuesday, • `wed`—Wednesday, • `thu`—Thursday, • `fri`—Friday, • `sat`—Saturday, • `sun`—Sunday.

F6. Share.conf Configuration File

The `DRWESI$` shared resource is created by default when installing Dr.Web Server for Windows.

Its settings are stored in the `/etc/share.conf` file. By default, the contents of the file include two lines:

1. Shared resource address: `%ProgramFiles%\DrWeb Server\webmin\install`.
2. Shared resource name: `DRWESI$` by default (can be changed during installation).

To change the name of the shared resource

1. Create a new shared resource via the **Windows Administration Tools** settings: go to **System Tools** → **Shared Folders** → **Shares** and select **New Share** in the context menu.
2. Specify the new name in the `share.conf` configuration file.

To disable the shared resource

1. Disable the resource via the **Windows Administration Tools** settings: go to **System Tools** → **Shared Folders** → **Shares** and select **Stop Sharing** in the resource context menu.
2. Delete the `share.conf` configuration file.

Appendix G. Command Line Parameters of the Programs Included in Dr.Web Enterprise Security Suite

Command line parameters have a higher priority than the default settings, or other constant settings (set in the Dr.Web Server configuration file, Windows OS registry, etc.). In some cases, the parameters specified at launch also predetermine the constant parameters. Such cases are described below.

When describing the syntax of parameters of separate programs optional parts are enclosed in brackets `[...]`.

Features described below do not apply to the Dr.Web Agent network installer.

Some command line parameters have a form of a switch—they begin with a hyphen. Such parameters are also called switches, or options.

Many switches can be expressed in various equivalent forms. Thus, the switches which imply a logical value (`yes/no`, `disable/enable`) have a negative variant, for example, the `-admin-rights` switch has a pair `-no-admin-rights` with the opposite meaning. They can also be specified with an explicit value, for example, `-admin-rights=yes` and `-admin-rights=no`.

The synonyms of `yes` are `on`, `true`, `OK`. The synonyms of `no` are `off`, `false`.

If a switch value contains spaces or tabs, the whole parameter should be put in quotation marks, for example:

```
"-home=c:\Program Files\DrWeb Server"
```

The names of switches can be abbreviated (by omitting the last letters), unless the abbreviated name is to coincide with the beginning of any other switch.

In order to force run a command as administrator on operating systems of Windows family, you can use the `elevate` parameter. In this case, it shall be placed before all other switches and parameters, for instance: `drwcsd elevate start`.

G1. Network Installer

The start instruction format

`drwinst.exe [<switches>]`

Switches

Command line switches are valid for launching all types of Dr.Web Agent installation files.

Switches to launch the Dr.Web Agent network installer are specified in the following format: `/<switch> <parameter>`.

All parameters values are specified after the space. For example:

```
/silent yes
```

If a switch value contains spaces, tabs or the `/` symbol, the whole parameter should be put in quotation marks. For example:

```
/pubkey "C:\my folder\drwcsd-certificate.pem"
```

Allowed switches

- `/compression` <*mode*>—compression mode of the traffic with Dr.Web Server. The <*mode*> parameter may take one of the following values:
  - `yes`—use compression.

  - `no`—do not use compression.
  - `possible`—compression is possible. The final decision is defined depending on settings on the Dr.Web Server side.

  If the switch is not set, the `possible` value is used by default.
- `/encryption` <*mode*>—encryption mode of the traffic with Dr.Web Server. The <*mode*> parameter may take one of the following values:
  - `yes`—use encryption.
  - `no`—do not use encryption.
  - `possible`—encryption is possible. The final decision is defined depending on settings on the Dr.Web Server side.

  If the switch is not set, the `possible` value is used by default.
- `/excludeFeatures` <*components*>—the list of components, which must be excluded from installation on the station. To set several components, use the "`,`" sign as a divider. Available components:
  - `scanner`—Dr.Web Scanner,
  - `spider-mail`—SpIDer Mail,
  - `spider-g3`—SpIDer Guard,
  - `outlook-plugin`—Dr.Web for Microsoft Outlook,
  - `firewall`—Dr.Web Firewall,
  - `spider-gate`—SpIDer Gate,
  - `parental-control`—Office Control,
  - `antispam-outlook`—Dr.Web Anti-spam for Dr.Web for Microsoft Outlook component.
  - `antispam-spidermail`—Dr.Web Anti-spam for SpIDer Mail component.

  Components that are not set directly, save their default installation status.
- `/id` <*station_id*>—identifier of a station on which Dr.Web Agent will be installed.

  The switch is specifying with the `/pwd` switch for automatic authorization on Dr.Web Server. If authorization parameters are not set, authorization decision is defined on the Dr.Web Server side.
- `/includeFeatures` <*components*>—the list of components, which must be installed on the station. To set several components, use the "`,`" sign as a divider. Available components:
  - `scanner`—Dr.Web Scanner,
  - `spider-mail`—SpIDer Mail,
  - `spider-g3`—SpIDer Guard,
  - `outlook-plugin`—Dr.Web for Microsoft Outlook,
  - `firewall`—Dr.Web Firewall,
  - `spider-gate`—SpIDer Gate,
  - `parental-control`—Office Control,
  - `antispam-outlook`—Dr.Web Anti-spam for Dr.Web for Microsoft Outlook component.

- `antispam-spidermail`—Dr.Web Anti-spam for SpIDer Mail component.

Components that are not set directly, save their default installation status.

- `/installdir` <*folder*>—installation folder.

  If the switch is not set, default installation folder is the "`Program Files\DrWeb`" folder on the system drive.

- `/installtimeout` <*time*>—waiting limit of reply from a station during the remote installation launched in the Control Center. Defined in seconds.

  If the switch is not set, 300 seconds are used by default.

- `/instMode` <*mode*>—installer launch mode. The <*mode*> parameter may take the following values:
  - `change`—change the list of the installed product components;
  - `remove`—remove the installed product;
  - `recovery`—restore the product if some of its components are corrupted.

  If the switch is not set, by default the installer automatically determines the launch mode.

- `/lang` <*language_code*>—installer language. Use the ISO-639-1 format to specify the language code.

  If the switch is not set, the system language is used by default.

- `/pubkey` <*certificate*>—full path to the Dr.Web Server certificate.

  If the certificate is not set, after the launch of the local installation, installer automatically uses the `*.pem` certificate file from own launch folder. If the certificate file is located in the folder other than the installer launch folder, you must manually specify the full path to the certificate file.

  If you launch the installation package generated in the Control Center, the certificate is included into the installation package and additional specifying of the certificate file in the command line switches is not required.

- `/pwd` <*password*>—the Dr.Web Agent password to access Dr.Web Server.

  The switch is specifying with the `/id` switch for automatic authorization on Dr.Web Server. If authorization parameters are not set, authorization decision is defined on the Dr.Web Server side.

- `/regagent` <*mode*>—defines whether Dr.Web Agent will be registered in the list of installed programs. The <*mode*> parameter may take one of the following values:
  - `yes`—register Dr.Web Agent in the list of installed programs.
  - `no`—do not register Dr.Web Agent in the list of installed programs.

  If the switch is not set, the `no` value is used by default.

- `/retry` <*number*>—number of attempts to locate Dr.Web Server by sending multicast requests. If Dr.Web Server has not responded after the specified attempts number is reached, it is assumed that Dr.Web Server is not found.

  If the switch is not set, 3 attempts to find Dr.Web Server are performed.

- `/server [<protocol>/]<server_address>[:<port>]`—the Dr.Web Server address from which the Dr.Web Agent installation will be performed and to which Dr.Web Agent connects after the installation.

  If the switch is not set, by default Dr.Web Server is searched by sending multicast requests.
- `/silent <mode>`—defines whether the installer will be run in the background mode. The *<mode>* parameter may take one of the following values:
  - `yes`—launch the installer in the background mode.
  - `no`—launch the installer in the graphical mode.

  If the switch is not set, by default Dr.Web Agent installation performs in the graphical mode of the installer (see the **Installation Manual**, p. Installing Dr.Web Agent via the Installer (read online)).
- `/timeout <time>`—waiting limit of each reply when searching Dr.Web Server. Defined in seconds. Receiving of response messages continues while the response time is less than the time-out value.

  If the switch is not set, 3 seconds are used by default.

G2. Dr.Web Agent for Windows

The start instruction format

```
es-service.exe [<switches>]
```

Switches

Each switch may be set in one of the following formats (formats are equivalent):

`-<short_switch>[ <argument>]`

or

`--<long_switch>[=<argument>]`

Switches may be used simultaneously including short and long versions.

If an argument contains spaces, it must be enclosed in quotes. Short switches can be used without a space character, for example:

```
es-service -e192.168.1.1:12345
```

All switches can be executed not dependently on permissions granted for the station on Dr.Web Server. I.e. even if permissions to change Dr.Web Agent settings are denied on Dr.Web Server, you can change these settings via the command line switches.

Allowed switches

- Show help:
  - `-?`
  - `--help`
- Change address of Dr.Web Server to which Dr.Web Agent connects:
  - `-e` <*Dr.Web_Server*>
  - `--esserver=`<*Dr.Web_Server*>

  To set several Dr.Web Servers at a time, you must repeat via the space character the `-e` switch for each Dr.Web Server address, e.g.:

  ```
  es-service -e 192.168.1.1:12345 -e 192.168.1.2:12345 -e 10.10.1.1:1223
  ```

  or

  ```
  es-service --esserver=10.3.1.1:123 --esserver=10.3.1.2:123 --
  esserver=10.10.1.1:123
  ```
- Add the public encryption key:
  - `-p` <*key*>
  - `--addpubkey=`<*key*>

  Public key specified as an argument is copied to the Dr.Web Agent folder (the `%ProgramFiles%\DrWeb` folder by default), is renamed to `drwcsd.pub` (if the name differs) and reread by the service. At this, previous public key file, if presented, is renamed to `drwcsd.pub.old` and no longer used.

  All public keys which were used previously (keys transmitted from Dr.Web Server and stored in the registry) are remained and used.
- Add the Dr.Web Server certificate:
  - `-c` <*certificate*>
  - `--addcert=`<*certificate*>

  Dr.Web Server certificate file specified as an argument is copied to the Dr.Web Agent folder (the `%ProgramFiles%\DrWeb` folder by default), is renamed to `drwcsd-certificate.pem` (if the name differs) and reread by the service. At this, previous certificate file, if presented, is renamed to `drwcsd-certificate.pem.old` and no longer used.

  All certificates which were used previously (certificates transmitted from Dr.Web Server and stored in the registry) are remained and used.
- Reconnect to the Server as a newbie:
  - `-w` <*value*>
  - `--newbie=`<*value*>

Valid values: `once, always`. When set to `always`, the authorization parameters of Dr.Web Agent are reset every time the service is started, and Dr.Web Agent always connects to the Server as a newbie (see **Administrator Manual**, the New Stations Approval Policy (read online) section for more details). When set to `once`, the authorization parameters of Dr.Web Agent will be reset the next time the service is started, after which Dr.Web Agent will reconnect to the Server as a newbie just once.

- Change the Dr.Web Agent log details level:

  `--change-loglevel=<level>`

  Allowed values of log details level: `err, wrn, inf, dbg, all.`

  This switch works only under administrator rights and does not require switching off Self-Protection or manually restart of the service or OS.

G3. Dr.Web Server

There are several variants as how to launch Dr.Web Server. These variants will be described separately.

Several commands require `modexec` or `modexecdb` switches specified in command line to execute Lua modules and submit auxiliary parameters, if necessary. Such a command shall be written in the following way:

`drwcsd [<switches>] modexec [<function_name>@]<module_name> [<parameters>]`

- *<function_name>*—name of a specific function to be executed from a Lua module.
- *<module_name>*—name of a Lua module to be executed.

Commands described in p. G3.1. Managing Dr.Web Server—G3.5. Backup of Dr.Web Server Critical Data are cross-platform and enable using in both Windows OS and Unix-like OS, unless it is specified otherwise.

If an error occurred while launching Dr.Web Server management commands, please refer to the Dr.Web Server log file to find possible causes (see **Administrator Manual**, p. Dr.Web Server Log (read online).

G3.1. Managing Dr.Web Server

`drwcsd [<switches>]`—set the parameters for Dr.Web Server operation (the switches are described in more detail in Appendix G3.8).

In Windows OS the `drwcsd.exe` file is located in `C:\Program Files\DrWeb Server\bin` by default.

In Linux and FreeBSD OS it is recommended that you use the scripts located in the following directories by default:

- Linux: `/etc/init.d/drwcsd`
- FreeBSD: `/usr/local/etc/rc.d/drwcsd`

G3.2. Basic Commands

- `drwcsd restart`—restart Dr.Web Server (it is executed as the `stop` and then `start` pair).
- `drwcsd start`—run Dr.Web Server.
- `drwcsd stop`—stop Dr.Web Server.
- `drwcsd stat`—log statistics to a file: CPU time, memory usage, etc. (for Unix-like OS—similar to `send_signal WINCH` or `kill SIGWINCH` commands).
- `drwcsd modexec agent@verify-key` <*full_key_filename*>—verify the license key file (`agent.key`).
- `drwcsd modexec enterprise@verify-key` <*full_key_filename*>—verify the Dr.Web Server license key file (`enterprise.key`). Please note that the Dr.Web Server license key file is no longer used from the version 10.
- `drwcsd verifyconfig` <*full_config_filename*>—verify the syntax of the Dr.Web Server configuration file (`drwcsd.conf`).
- `drwcsd verifycache`—verify content of the Dr.Web Server's file cache.
- `drwcsd syncads`—synchronize the network structure: Active Directory containers that contain computers become anti-virus network groups where workstations are placed into.
- `drwcsd modexec restore@repository-server-update`—restore Dr.Web Server from revision (roll back to the state of the selected revision without saving a backup copy, updating the database and configuration files). If this revision is absent in the repository, nothing will happen.

This action is recorded in the audit log.

- `update@repository-server-update [<revision-to>]`—update Dr.Web Server. If no revision is specified to update to, the server will be updated to the most recent revision.
- `revert@repository-server-update <revision-to>`—roll back Dr.Web Server to a certain revision.

The last two actions are recorded in the audit log.

Update and rollback of Dr.Web Server are described in the **Administrator Manual**, section Updating Dr.Web Server and Restoring it from Backup (read online).

G3.3. Database Commands

Database Initialization

For initialization, the database must be absent or empty.

Flag values must be specified in `%true` and `%false`, or `1` and `0` format.

`drwcsd [<switches>] modexecdb database-init [<license_key> [<password>]]`—database initialization.

- *<license_key>*—path to Dr.Web license key file `agent.key`. If the license key is not specified, it must be added later from the Control Center or get from the neighbor Dr.Web Server via the interserver connection.
- *<password>*—original password of the Dr.Web Server administrator (logged in as **admin**). The default password is **root**.

If you need to skip one or several parameters when writing the command, use the special value `%nil` instead of each skipped parameter.

`%nil` can be omitted, if the next parameters are not set.

Adjusting parameters of database initialization

If embedded database is used, initialization parameters can be set via an external file. The following command is used for this:

`drwcsd.exe modexecdb database-init@<response-file>`

<response-file>—file with initialization parameters written line-by-line in the same order same as parameters of the `database-init` command.

File format:

```
<full_license_key_filename>

<administrator_password>
```

When using a response file under Windows OS, any symbols are allowed in the administrator password.

If a string consists of only the `%nil` value, the default value is used (as in `database-init`).

Database Upgrading

`drwcsd modexecdb database-upgrade [pretend] [upgrade_ver_flag]`—run Dr.Web Server to upgrade the structure of the database to a new version using the internal scripts.

- `pretend=%false`—default value. Instructs to update the database. If `%true`, only checking the relevance of the database will be performed instead of database update.
- `upgrade_ver_flag=%true`—during the update database version and data are recorded on each successful update to the next version of the database schema. Default value is `%false`.

Instead of the `%true` or `%false` parameter values it is allowed to use the equivalent 1 or 0.

Database Export

a) `drwcsd modexecdb database-export` <*file*> `[ignore_tables]`—export the database to a specified file.

- <*file*>—path to the file to which the database will be exported.
- `ignore_tables`—allows to specify a string or a table with strings containing the names of the database tables which are not to be exported. The format of tables with strings is as follows: `%{"table1", "table2"}`. In case of specifying a string—`table1`.

Example for Windows OS:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb Server\var\exportdb.log"
modexecdb database-export "C:\Program Files\DrWeb Server\esbase.es"
alert_unsent
```

Under Unix-like OS, the action is performed on behalf of the `drwcs:drwcs` user to the `$DRWCS_VAR` directory (except **FreeBSD** OS, which by default saves the file to the directory from which the script was run; if the path is specified explicitly, then the directory should have the write access for the <*user*>:<*group*> that had been created at installation, by default it is `drwcs:drwcs`).

b) `drwcsd modexecdb database-export-xml` <*xml_file*> `[ignore_tables]`—export the database to a specified XML file.

- <*xml-file*>—path to the XML file to which the database will be exported.
- `ignore_tables`—allows to specify a string or a table with strings containing the names of the database tables which are not to be exported. The format of tables with strings is as follows: `%{"table1", "table2"}`. In case of specifying a string—`table1`.

If you specify the `gz` file extension, during the export, database file will be packed into the `gzip` archive.

If you do not specify any extension or specify an extension other than `gz`, export file will not be archived.

Example for Windows OS:

- To export the database into the XML file with no compression:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -bin-root="C:\Program Files\DrWeb Server" -var-
root="C:\Program Files\DrWeb Server\var" -verbosity=trace -rotate=10,10m -
log="C:\Program Files\DrWeb Server\var\exportxmldb.log" modexecdb
database-export-xml database.db
```

- To export the database into the XML file compressed to an archive:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -bin-root="C:\Program Files\DrWeb Server" -var-
root="C:\Program Files\DrWeb Server\var" -verbosity=trace -rotate=10,10m -
log="C:\Program Files\DrWeb Server\var\exportxmldb.log" modexecdb
database-export-xml content.gz
```

In previous versions, the name of the database backup file is different (`database.gz`).

Example for Unix-like OS:

- To export the database into the XML file with no compression:

```
/etc/init.d/drwcsd modexecdb database-export-xml /test/database.db
```

- To export the database into the XML file compressed to an archive:

```
/etc/init.d/drwcsd modexecdb database-export-xml /es/content.gz
```

Database Import

a) `drwcsd modexecdb database-import` <*file*> `[ignore_tables]`—import the database from a specified file (in this case all records from all tables are deleted, the tables themselves are not deleted; records from the export are inserted into them). An import can be performed only into an initialized database within the same database schema version. To perform an import from an older to a newer database schema version, use the `database-import-and-upgrade` command.

- <*file*>—path to the file from which the database will be imported.
- `ignore_tables`—allows to specify a string or a table with strings containing the names of the database tables which are not to be imported. The format of tables with strings is as follows: `%{"table1", "table2"}`. In case of specifying a string—`table1`.

b) `drwcsd modexecdb database-import-and-upgrade` <*file*> `[import_only_flag] [upgrade_ver_flag] [ignore_tables]`—import and upgrade the database exported from Dr.Web Server of previous version. The tables themselves are deleted, new tables are created according to the description from the export. Records from the export are inserted into new tables, and then the update procedure is started.

- <*file*>—path to the file from which the database will be imported.
- `import_only_flag`—if `%true`, only import is performed, without database update and verification. Default value is `%false`.
- `upgrade_ver_flag`—if `%true`, during a database upgrade, the database version and data are committed each time it is successfully upgraded to the next version of the database schema. Default value is `%false`.
- `ignore_tables`—allows to specify a string or a table with strings containing the names of the database tables which are not to be imported. The format of tables with strings is as follows: `%{"table1", "table2"}`. In case of specifying a string—`table1`.

When specifying parameters, the previous ones must be set, for example: when specifying `ignore_tables`, the `import_only_flag` and `upgrade_ver_flag` parameters must be set (that is, both flags must be specified before `ignore_tables`).

Before using the `database-import-and-upgrade` command, you are strongly advised to back up the database.

Problems during execution of these commands may lead to deletion of all information from the database.

You can use the `database-import-and-upgrade` command to import and upgrade the version of the database only within the same database (for example, when using the MySQL database, the `database-import-and-upgrade` command can be executed to import into the MySQL database only).

Database Verification

`drwcsd modexecdb database-verify [full [ignore-version]]`—run Dr.Web Server to verify the database and restore missing information. In order to record the results into a log file, the command shall be followed by a `-log` key. Details on how to use the key are described in G3.8. The Description of Switches.

- `full=%false`—defines the verification type. By default (`%false`) a quick check is made, if `%true`—a full verification.
- `ignore-version=%false`—defines whether the database schema version needs to be ignored during verification. `%false` by default. If `%true`, verification will continue even if the schema version is wrong.

Example

A full database check even if the database schema version is incorrect.

```
drwcsd modexecdb database-verify %true %true
```

Database verification cannot be performed while Dr.Web Server is running.

Database Speed Up

`drwcsd [<switches>] modexecdb database-speedup`—execute the `VACUUM`, `CLUSTER`, `ANALYZE` commands to speed up the DB operation.

Database Restore

`drwcsd repairdb`—repair malformed disk image of **SQLite3** embedded database or corrupted tables of **MySQL** external database.

SQLite3 may be also automatically recovered on the Dr.Web Server startup if in the **SQLite3** database settings in the Control Center, the **Restore corrupted image automatically** flag is set (see **Administrator Manual**, p. Database Restore (read online).

Database cleanup

`drwcsd modexecdb database-clean`—clean up the Dr.Web Server database by deleting all tables.

Administrator password reset

`drwcsd modexecdb set-admin-password` <*login*> <*new_password*>—set a new password for the specified administrator account.

G3.4. Repository Commands

You must necessarily stop Dr.Web Server before launching the `syncrepository`, `restorerepo` and `saverepo` commands.

- `drwcsd syncrepository`—synchronize the repository with Dr.Web GUS. The command launches the Dr.Web Server process, at this, GUS is called with the following repository update if updates are present.
- `drwcsd rerepository`—reread the repository from the drive. Under Unix-like OS, similar to the `readrepo` command.

- `drwcsd updrepository`—update the repository from Dr.Web GUS. The command sends the signal to the operating Dr.Web Server process to call the GUS and perform the following repository update if updates are present. If Dr.Web Server is not running, the repository update is not performed.
- `drwcsd [<switches>] restorerepo <full_archive_name>`—restore repository of Dr.Web Server from the specified zip archive created via the `saverepo` command.
- `drwcsd [<switches>] saverepo <full_archive_name>`—save all repository of Dr.Web Server to the specified zip archive. Created archive can be imported to Dr.Web Server via the `restorerepo` command.

Archives used by the `restorerepo` and `saverepo` commands are not compatible with archives used for repository export and import via the Control Center.

G3.5. Backup of Dr.Web Server Critical Data

The following command creates backup copies of critical Dr.Web Server data (repository settings, configuration files, encryption keys, certificates, database backup, and hooks):

- Windows OS: `drwcsd -home=<path> backup [<directory> [<quantity>]]`
- Linux, FreeBSD OS: `drwcsd backup [<directory> [<quantity>]]`

where:

- the `-home` switch sets the Dr.Web Server installation folder,
- critical Dr.Web Server data is copied to the specified *<directory>*,
- *<quantity>* is the number of copies of each file.

All files in the backup except the database contents, are ready to use. The database backup copy is stored in the `.gz` format compatible with `gzip` and other archivers. The database contents can be imported from the backup copy to another database of Dr.Web Server, thus restore the data (see p. Restoring the Database).

During the operation, Dr.Web Server regularly stores backup copies of critical data into the following folders:

- for **Windows** OS: *<installation_drive>*`:\DrWeb Backup`
- for **Linux** OS: `/var/opt/drwcs/backup`
- for **FreeBSD** OS: `/var/drwcs/backup`

To perform the back up, a daily task is included into the Dr.Web Server schedule. If such task is missing in the schedule, it is recommended to create it.

The backup folder is the working directory of the Dr.Web Server and can contain only backup copies of critical data.

Example for Windows OS:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program Files\DrWeb
Server" -verbosity=all -log="C:\Program Files\DrWeb Server\var\backup.log"
backup "C:\DrWeb Backup"
```

Example for Linux OS:

```
/etc/init.d/drwcsd backup /tmp/backup/ -verbosity=all -
log=/tmp/backup/backup.log
```

G3.6. Commands for Windows OS Only

- `drwcsd [<switches>] install [<service_name>]`—install the Dr.Web Server service in the system and assign specified switches to launch this service.

  <service_name> is a suffix that is added to the default name of the service; at this, the full name of the service is `DrWebES-<service_name>`. The `install` command creates (edits) the service with specified name and automatically adds the `-service=<service_name>` switch into its arguments. At this, existing services remain unchanged.
- `drwcsd uninstall [<service_name>]`—uninstall the Dr.Web Server service from a system.

  <service_name> is a suffix that is added to the default name of the service; at this, the full name of the service is `DrWebES-<service_name>`.
- `drwcsd kill`—perform emergency shutdown of the Dr.Web Server service (if normal termination failed). This instruction should not be used without extreme necessity.
- `drwcsd reconfigure`—reread and reboot the configuration file (it is performed quicker and without starting a new process).
- `drwcsd silent [<options>] <command>`—disable messages from Dr.Web Server when launching the *<command>*. Used particularly in command files to disable the Dr.Web Server interactivity.

G3.7. Commands for Unix-like OS Only

- `drwcsd cacherepo`—create or recreate the Dr.Web Server repository file cache.
- `drwcsd config`—similar to `reconfigure` or `kill SIGHUP` commands—restart Dr.Web Server.
- `drwcsd interactive`—run Dr.Web Server but do not direct the control to the process.
- `drwcsd newkey`—generate a new encryption keys `drwcsd.pri` and `drwcsd.pub` and the `drwcsd-certificate.pem` certificate.
- `drwcsd readrepo`—reread repository from the drive. Similar to the `rerepository` command.

- `drwcsd selfcert[<hostname>]`—generate a new SSL certificate (`certificate.pem`) and RSA private key (`private-key.pem`). The parameter specifies the host name with Dr.Web Server installed for which the files are generated. If the name is not specified, it will be gotten automatically by a system function.
- `drwcsd shell <file_name>`—run the script file. The command launches `$SHELL` or `/bin/sh` and passes the specified file.
- `drwcsd showpath`—show all program paths, registered in the system.
- `drwcsd status`—show the current status of Dr.Web Server (running, stopped).

G3.8. The Description of Switches

Switches described in this section can be specified in two functionally identical ways: with one hyphen (`-<switch>`) or two (`--<switch>`).

Cross-platform switches

- `-activation-key=<license_key>`—Dr.Web Server license key. By default, it is the `enterprise.key` file located in the `etc` subfolder of the root folder.

  Please note that the Dr.Web Server license key file is no longer used from the version 10. The `-activation-key` switch may be used during the Dr.Web Server upgrade from the previous versions and database initialization: the Dr.Web Server identifier will be taken from the specified license key.
- `-bin-root=<folder>`—the path to executable files. By default, it is the `bin` subfolder of the root folder.
- `-conf=<file>`—name and location of the Dr.Web Server configuration file. By default, it is the `drwcsd.conf` file in the `etc` subfolder of the root folder.
- `-daemon`—for Windows platforms it means to launch as a service; for UNIX platforms —"daemonization of the process" (to go to the root folder, disconnect from the terminal and operate in the background).
- `-db-verify=on`—check database integrity at the Dr.Web Server start. This is the default value. It is not recommended to run with an explicit opposite value, except if run immediately after the database is checked by the `drwcsd modexecdb database-verify` instruction, see above.
- `-help`—displays help. Similar to the programs described above.
- `-hooks`—to permit Dr.Web Server to perform user extension scripts located in the following subfolders of the Dr.Web Server installation folder:
  - for Windows OS: `var\extensions`
  - for FreeBSD OS: `/var/drwcs/extensions`
  - for Linux OS: `/var/opt/drwcs/extensions`

  The scripts are meant for automation of the administrator work enabling quicker performance of certain tasks. All scripts are disabled by default.

- `-home=<folder>`—Dr.Web Server installation folder (root folder). The structure of this folder is described in **Installation Manual**, p. Installing Dr.Web Server for Windows OS (read online). By default, it is the current folder at start.
- `-log=<log_file>`—activate logging to the file at the specified path.

  For Dr.Web Servers under Unix-like OS, the "minus" sign can be used instead of the filename, which defines standard output of the log. Operations are run under Unix-like OS with the permissions of the drwcs user. If this user cannot write to the log folder, an error will occur.

  By default: for Windows OS, it is `drwcsd.log` in the folder specified by the `-var-root` switch, for Unix-like OS, it is set by the `-syslog=user` switch (see below).
- `-private-key=<private_key>`—private encryption key of Dr.Web Server. By default, it is `drwcsd.pri` in the `etc` subfolder of the root folder.
- `-rotate=<N><f>,<M><u>`—Dr.Web Server log rotation mode, where:

Parameter	Description
<*N*>	Total number of log files (including current and archive).
<*f*>	Log files storing format, possible values: • z (gzip)—compress files, used by default, • p (plain)—do not compress files.
<*M*>	Log file size or rotation time, depending on the <*u*> value;

<*u*>	Unit measure, possible values:
	• to set rotation by log file size:
	  ▫ k—Kb,
	  ▫ m—Mb,
	  ▫ g—Gb.
	• to set rotation by time:
	  ▫ H—hours,
	  ▫ D—days,
	  ▫ W—weeks.

If rotation by time is set, synchronization performs independently on command launch time: the H value means synchronization with the beginning of an hour, D—with beginning of a day, W—with beginning of a week (00:00 on Monday) according to the multiplicity specified in the <*u*> parameter.

Initial reference point—January 01, year 01 AD, UTC+0.

The minimum log file size is 1 MB.

By default, it is `10z,10m`, which means storing of 10 files 10 megabytes each, use compression. Alternatively you can use the none format (`-rotate=none`), which means "do not use rotation, always write to the same file of unlimited size".

In the rotation mode, log file names are generated as follows: `file.<N>.log` or `file.<N>.log.gz`, where <*N*>—sequence number: 1, 2, etc.

For example, the log file name is set to `file.log` (see the `-log` switch above), then

- `file.log`—current log file,
- `file.1.log`—previous log file,
- `file.2.log` and so on—the greater the number, the older the version of the log.

The `-rotate` and `-log` switches are position-dependent.

When using these switches together, the `-rotate` switch must go before the `-log` switch: the `-rotate` switch defines the rotation mode of logs located at the paths specified further in the command line.

- `-trace`—to log in detail the location of error origin.
- `-var-root=`<*folder*>—path to a folder to which Dr.Web Server has a write access and which is designed to store modified files (for example, logs and the repository files). By default, it is the `var` subfolder of the root folder.
- `-verbosity=`<*level*>—log level of detail. `TRACE` is by default. Allowed values are: `ALL`, `DEBUG3`, `DEBUG2`, `DEBUG1`, `DEBUG`, `TRACE3`, `TRACE2`, `TRACE1`, `TRACE`, `INFO`, `NOTICE`, `WARNING`, `ERROR`, `CRIT`. The `ALL` and `DEBUG3` values are synonyms.

  If necessary, it is possible to set specific levels of detail for several message sources at once in the following format:

  `-verbosity=`<*message_source1*>`:`<*level1*>`,`<*message_source2*>`:`<*level2*>`,` <*message_source3*>`:`<*level3*> and so on. In this case, <*level*> is inherited on general principle, i.e. it looks for the closest parental source with specified level of detail. The `-verbosity=all:all` switch is equal to the `-verbosity=all` switch (see also Appendix J. Log Files Format).

This switch defines the log detail level set by the subsequent `-log` switch (see above). One instruction can contain several switches of this type.

The `-verbosity` and `-log` switches are position-dependent.

When using these switches together, the `-verbosity` switch must go before the `-log` switch: the `-verbosity` switch defines the verbosity level of logs located at the paths specified further in the command line.

Switches for Windows OS Only

- `-minimized`—(if run not as a service, but in the interactive mode)—minimize a window.

- `-service=<service_name>`—the switch is used by running service process for self-identification and activation of the self-protection for the registry branch of the Dr.Web Server service. *<service_name>* is a suffix that is added to the default name of the service; at this, the full name of the service is `DrWebES-<service_name>`.

  The switch is used by the `install` command; independent use is not provided.
- `-screen-size=<size>`—(if run not as a service, but in the interactive mode)—log size in lines displayed in the Dr.Web Server screen, the default value is 1000.

Switches for Unix-like OS Only

- `-etc=<path>`—path to the `etc` (`<var>/etc`) directory.
- `-keep`—do not remove temporary directory after the Dr.Web Server installation.
- `-pid=<file>`—a file to which Dr.Web Server writes the identifier of its process.
- `-syslog=<mode>`—instructs logging to the system log. Possible modes: `auth`, `cron`, `daemon`, `kern`, `lpr`, `mail`, `news`, `syslog`, `user`, `uucp`, `local0-local7` and for some platforms—`ftp`, `authpriv`.

  

  The `-syslog` and `-log` keys work together. I.e., if you start Dr.Web Server with the `-syslog` key (e.g., `service drwcsd start -syslog=user`), the Dr.Web Server run with specified value for the `-syslog` key and with default value for the `-log` key.

- `-user=<user>`, `-group=<group>`—available for UNIX OS only, if run by the root user; it means to change the user or the group of process and to be executed with the permissions of the specified user (or group).

G3.9. Variables for Unix-like OS Only

To make the administration of the Server under Unix-like OS easier, administrator is provided with variables which reside in the script file:

- For Linux OS: `/etc/init.d/drwcsd`.
- For FreeBSD OS: `/usr/local/etc/rc.d/drwcsd` (symbolic link to the `/usr/local/etc/drweb.com/software/init.d/drwcsd`).

Correspondence between variables and command switches for the `drwcsd` is described in the table below.

drwcsd command line switches and the corresponding variables

Switch	Variable	Default parameters
`-home`	`DRWCS_HOME`	• Linux OS: `/opt/drwcs` • FreeBSD OS: `/usr/local/drwcs`
`-var-root`	`DRWCS_VAR`	• Linux OS: `/var/opt/drwcs`

Switch	Variable	Default parameters
		• FreeBSD OS: `/var/drwcs`
`-etc`	`DRWCS_ETC`	`$DRWCS_VAR/etc`
`-rotate`	`DRWCS_ROT`	`10,10m`
`-verbosity`	`DRWCS_LEV`	`info`
`-log`	`DRWCS_LOG`	`$DRWCS_VAR/log/drwcsd.log`
`-conf`	`DRWCS_CFG`	`$DRWCS_ETC/drwcsd.conf`
`-pid`	`DRWCS_PID`	
`-user`	`DRWCS_USER`	
`-group`	`DRWCS_GROUP`	
`-hooks`	`DRWCS_HOOKS`	
`-trace`	`DRWCS_TRACE`	

`DRWCS_HOOKS` and `DRWCS_TRACE` variables do not have any parameters. If variables have been defined, corresponding switches will be added during the script execution. If variables have not been defined, switches will not be added.

Other variables are described in the table below.

Variables without corresponding command line switches

Variables	Default parameters	Description
`DRWCS_ADDOPT`		Additional command line instructions to deliver to `drwcsd` at start.
`DRWCS_CORE`	`unlimited`	The core file maximal size.
`DRWCS_FILES`	131170	The maximal number of file descriptors, that Dr.Web Server is able to open.
`DRWCS_BIN`	`$DRWCS_HOME/bin`	The directory to start the `drwcsd` from.
`DRWCS_LIB`	`$DRWCS_HOME/lib`	The directory with the Dr.Web Server libraries.

Default values of parameters will be used, if these variables have not been defined in the `drwcsd` script.

`DRWCS_HOME`, `DRWCS_VAR`, `DRWCS_ETC`, `DRWCS_USER`, `DRWCS_GROUP`, `DRWCS_HOOKS` variables are already defined in the `drwcsd` script file.

If the `/var/opt/drwcs/etc/common.conf` file exists, it will be included in `drwcsd`, which could redefine some variables. However, if they are not exported (via the `export` command), they will not take any effect.

To set variables

1. Add variable definition to the `drwcsd` script file.
2. Export this variable using the `export` command (at the same place).
3. When one more process will be run from this script, this process will read values that have been set.

G3.10. Administration of Dr.Web Server Version for UNIX OS with the kill Instruction

The version of Dr.Web Server for UNIX OS is administrated by the signals sent to the Dr.Web Server processor by the `kill` utility.

Use the `man kill` instruction to receive help on the `kill` utility.

Below are listed the utility signals and the actions performed by them:

- `SIGWINCH`—log statistics to a file (CPU time, memory usage, etc.),
- `SIGUSR1`—reread the repository from the drive,
- `SIGUSR2`—reread templates from the drive,
- `SIGHUP`—restart Dr.Web Server,
- `SIGTERM`—shut down Dr.Web Server,
- `SIGQUIT`—shut down Dr.Web Server,
- `SIGINT`—shut down Dr.Web Server.

Similar actions are performed by the switches of the drwcsd instruction for the Windows version of Dr.Web Server, read Appendix G3.3. Database Commands.

G4. Dr.Web Scanner for Windows

This component of the workstation software has the command line parameters which are described in the **Dr.Web Agent for Windows User Manual**. The only difference is that when

the Scanner is run by Dr.Web Agent, the `/go /st` parameters are sent to Dr.Web Server automatically and without fail.

G5. Dr.Web Proxy Server

To configure the Proxy Server parameters, run with corresponding switches the `drwcsd-proxy` executable file, which resides in the `bin` subdirectory of the Proxy Server installation directory.

The Start Instruction Format

`drwcsd-proxy [<switches>] [<commands> [<command_arguments>]]`

Allowed Switches

Cross-platform Switches

- `--console=yes|no`—run the Proxy Server in the interactive mode. At this, the Proxy Server operation log is written to the console.

  Default: `no`.
- `--etc-root=<path>`—path to the configuration files directory (`drwcsd-proxy.conf`, `drwcsd-proxy.auth` and etc.).

  Default: `$var/etc`
- `--home=<path>`—path to the Proxy Server installation directory.

  Default: `$exe-dir/`
- `--log-root=<path>`—path to the directory with the Proxy Server operation log files.

  Default: `$var/log`
- `--pool-size=<N>`—pool size for clients connections.

  Default: core number of the computer with the Proxy Server installed (not less than 2).
- `--rotate=<N><f>,<M><u>`—Proxy Server log rotation mode, where:

Parameter	Description
<*N*>	Total number of log files (including current and archive).
<*f*>	Log files storing format, possible values: • z (gzip)—compress files, used by default, • p (plain)—do not compress files.
<*M*>	Log file size or rotation time, depending on the <*u*> value;
<*u*>	Unit measure, possible values: • to set rotation by log file size:

Parameter	Description
	▫ k—Kb, ▫ m—Mb, ▫ g—Gb. • to set rotation by time: ▫ H—hours, ▫ D—days, ▫ W—weeks.

If rotation by time is set, synchronization performs independently on command launch time: the H value means synchronization with the beginning of an hour, D—with beginning of a day, W—with beginning of a week (00:00 on Monday) according to the multiplicity specified in the <*u*> parameter.

Initial reference point—January 01, year 01 AD, UTC+0.

Default is `10,10m`, which means storing of 10 files of 10 megabytes each, use compression.

- `--trace=yes|no`—enable detailed logging of Proxy Server calls. Available only if the Proxy Server supports calls stack tracing (if an exception occurs, stack is written to the log).

  Default: `no`.

- `--tmp-root=`<*path*>—path to the temporary files directory. Is used at Proxy Server automatic update.

  Default: `$var/tmp`.

- `--var-root=`<*path*>—path to the Proxy Server working directory to store cache and database.

  Default:

  ▫ Windows OS: `%ALLUSERSPROFILE%\Doctor Web\drwcs`

  ▫ Linux OS: `/var/opt/drwcs`

  ▫ FreeBSD OS: `/var/drwcs`

- `--verbosity=`<*details_level*>—log details level. Default is `TRACE`. Allowed values are: `ALL, DEBUG3, DEBUG2, DEBUG1, DEBUG, TRACE3, TRACE2, TRACE1, TRACE, INFO, NOTICE, WARNING, ERROR, CRIT`. The `ALL` and `DEBUG3` values are synonyms.

  If necessary, it is possible to set specific levels of detail for several message sources at once in the following format:

  `-verbosity=`<*message_source1*>`:`<*level1*>`,`<*message_source2*>`:`<*level2*>`,`<*message_source3*>`:`<*level3*> and so on. In this case, <*level*> is inherited on general principle, i.e. it looks for the closest parental source with specified level of detail. The `-verbosity=all:all` switch is equal to the `-verbosity=all` switch (see also Appendix J. Log Files Format).

All switches for setting Proxy Server parameters can be set simultaneously.

Switches for Unix-like OS

- `--user`—set the user identifier. The switch is relevant for both, normal and daemon modes.
- `--group`—set the group identifier. The switch is relevant for both, normal and daemon modes.
- `--pid=`<*path*>—path to the directory with the process identifier.

  Default: `/var/opt/drwcs/run/drwcsd-proxy.pid`

Allowed Commands and their Arguments

If the command is not specified, the `run` command is used.

- `import` <*path*> `[`<*revision*>`]` `[`<*products*>`]`—import files from Dr.Web Server repository to the Proxy Server cache.
  - <*path*>—path to the directory with Dr.Web Server repository. The Dr.Web Server repository must be preliminary downloaded to the computer with the Proxy Server installed.
  - <*revision*>—maximum number of revisions to import. If the value is not set, all revision will be imported.
  - <*products*>—the list of products separated by the space that are intended to import. By default, the list is empty, i.e. import all repository products except Dr.Web Server. If the list is set, only the products from the list will be imported.
- `help`—show help message on switches for Proxy Server configuration.
- `run`—start the Proxy Server in normal mode.

For Windows OS only

- `install`—install the service.
- `start`—start the installed service.
- `stop`—stop the started service.
- `uninstall`—uninstall the service.

Commands available under Unix-like OS only:

- `daemon`—run the Proxy Server as a daemon (see also Switches for Unix-like OS).

Proxy server control script and variables available under Unix-like OS

To make the administration of Dr.Web Proxy Server under Unix-like OS easier, administrator is provided with variables which reside in the `drwcsd-proxy.sh` script file executed with the help of an initialization script:

- **Linux**: `/etc/init.d/dwcp_proxy`
- **FreeBSD**: `/usr/local/etc/rc.d/dwcp_proxy`

The script accepts the following commands:

- `import` <*path*> [<*revision*>] [<*products*>]—import files from Dr.Web Server repository to the Proxy Server cache (same as the command of Proxy Server—see above).
- `interactive`—run the Proxy Server in the interactive mode. At this, the Proxy Server operation log is written to the console.
- `start`—run the Proxy Server as a daemon.
- `status`—check if the daemon is launched.
- `stop`—stop the started daemon.

Correspondence between variables and command switches for the `drwcsd-proxy` is described in the table below.

drwcsd-proxy command line switches and the corresponding variables

Switch	Variable	Default parameters
`--home=`<*path*>	`DRWCS_PROXY_HOME`	• Linux OS: `/opt/drwcs` • FreeBSD OS: `/usr/local/drwcs`
`--var-root=`<*path*>	`DRWCS_PROXY_VAR`	• Linux OS: `/var/opt/drwcs` • FreeBSD OS: `/var/drwcs`
`--etc-root=`<*path*>	`DRWCS_PROXY_ETC`	`$DRWCS_PROXY_VAR/etc`
`--tmp-root=`<*path*>	`DRWCS_PROXY_TMP`	`$DRWCS_PROXY_VAR/tmp`
`--log-root=`<*path*>	`DRWCS_PROXY_LOG`	`$DRWCS_PROXY_VAR/log`
–	`DRWCS_PROXY_LIB`	`$DRWCS_PROXY_HOME/lib`
–	`DRWCS_PROXY_BIN`	`$DRWCS_PROXY_HOME/bin`
`--verbosity=`<*detail_level*>	`DRWCS_PROXY_VERBOSITY`	`info`
`--rotate=`<*N*><*f*>,<*M*><*u*>	`DRWCS_PROXY_ROTATE`	`10,10m`

Switch	Variable	Default parameters
`--pid`	`DRWCS_PROXY_PID`	`$DRWCS_PROXY_VAR/run/drwcsd-proxy.pid`
-	`NO_DRWCS_PROXY_USER`	If any value is assigned, `$DRWCS_PROXY_USER` is ignored.
`-user`	`DRWCS_PROXY_USER`	–
-	`NO_DRWCS_PROXY_GROUP`	If any value is assigned, `$DRWCS_PROXY_GROUP` is ignored.
`--group`	`DRWCS_PROXY_GROUP`	–
-	`DRWCS_PROXY_FILES`	131170 but not less than the current limit.

G6. Dr.Web Server Installer for Unix-like OS

The start instruction format

<*package_name*>.run [<*switches*>] [--] [<*arguments*>]

where:

- `[--]`—separate optional sign, determines the end of the switches list and separates the switches list and the additional arguments list.
- `[`<*arguments*>`]`—additional arguments or embedded scripts.

Switches to get help or information on the package

- `--help`—show the help on switches.
- `--info`—show extended information on the package: the name; destination folder; unpacked size; compression algorithm; compression date; the version of `makeself` which is used for packing; the command user for packing; the script that will be launched after unpacking; whether the archive content will be copied into the temporary folder or not (if no, nothing shown); whether the destination folder stored or will be deleted after the script execution.
- `--lsm`—show contents of embedded LSM entry with basic information about the package: title, version, description, author, etc. A corresponding message will be displayed if the LSM entry is not filled in.
- `--list`—show the list of files in the installation package.
- `--check`—check integrity of the installation package.

Switches to run the package

- `--confirm`—ask before running embedded script.
- `--noexec`—do not run embedded script.
- `--keep`—do not delete the target folder once the embedded script is executed.
- `--nox11`—do not spawn the `xterm` terminal emulator once the installation is complete.
- `--nochown`—do not assign the user who initiated the installation as the owner of extracted files.
- `--log` <*path*>—enable installation logging into a file at a specified path.
- `--nolog`—do not log the installation.
- `--target` <*folder*>—extract the installation package to the specified folder.
- `--tar` <*argument_1*> `[`<*argument_2*> ...`]`—get access the contents of the installation package through the tar command.

Additional arguments

- `--help`—show the help on additional arguments.
- `--quiet`—run the installer in the background mode. The affirmative answer is used for all the following questions of the installer:
  - accept the license agreement,
  - set back up into the default folder,
  - continue the installation provided that extra distribution kit installed in the system will be deleted.
- `--clean`—install the package with the Dr.Web Server default settings not using the backup to restore the settings from the previous installation.
- `--preseed` <*path*>—path to the configuration file with predefined answers on installer questions during the installation.

  Variables to specify the predefined answers in the configuration file:
  - `DEFAULT_BACKUP_DIR=`<*path*>—path to the backup that is used for restoring the settings from the previous version (is not used if you set the installation not applying restore from the backup).
  - `QUIET_INSTALL=[0|1]`—defines the usage of the installer background mode:
    - 0—run the installer in the background mode;
    - 1—run the installer in the regular mode.
  - `CLEAN_INSTALL=[0|1]`—defines the usage of backup during the installation:
    - 0—install not applying backup;
    - 1—install applying restore from the backup located in the folder from the `DEFAULT_BACKUP_DIR` variable. If the `DEFAULT_BACKUP_DIR` variable is not specified, backup is taken from the `/var/tmp/drwcs`.

- `ADMIN_PASSWORD=<password>`—password for default administrator account (**admin**).
  - If the `ADMIN_PASSWORD` variable is specified in the file, its value is used as the administrator password and the following message is displayed at the end of the installation:

    ```
    Password specified in the configuration file for the default administrator (admin): <password>
    ```

  - If the `ADMIN_PASSWORD` variable is not specified in the file, the password is generated automatically and the following message is displayed at the end of the installation:

    ```
    Automatically generated password for the default administrator (admin): <password>
    ```

If you use the `--preseed` argument and do not define the installer background mode in the configuration file via the `QUIET_INSTALL=0` variable, when the values of other variables of the configuration file will be redefined by a user during the installation.

G7. Utilities

G7.1. Digital Keys and Certificates Generation Utility

The following console versions of the digital keys and certificates generation utility are provided:

Executable file	Location	Description
`drweb-sign-<OS>-<bitness>`	Control Center, the **Administration** → **Utilities** section	Independent version of the utility. Can be launched from any directory or on any computer with corresponding operating system.
	The `webmin/utilities` Dr.Web Server directory	
`drwsign`	The `bin` Dr.Web Server directory	Utility version depends on server libraries. Can be launched only from its location directory.

The `drweb-sign-<OS>-<bitness>` and `drwsign` utility versions are similar in their functions. Further in the section, the `drwsign` version is used, but all examples are relevant for both versions.

The start instruction format

- `drwsign check [-public-key=<public_key>] <file>`

  Check the specified file signature using a public key of the person who signed this file.

Switch parameter	Default value
<public_key>	`drwcsd.pub`

- `drwsign extract [-private-key=<private_key>] [-cert=<Dr.Web_Server_certificate>] <public_key>`

  Extract the public key from the private key file or from the certificate and write the public key to the specified file.

  The `-private-key` and `-cert` switches are mutually exclusive, i.e. only one switch can be set; if both switches are set at the same time, the command fails to execute.

  The switch parameters must be specified.

  If none of the switches is set, `-private-key=drwcsd.pri` is used to extract the public key of the `drwcsd.pri` private key.

Switch parameter	Default value
<private_key>	`drwcsd.pri`

- `drwsign genkey [<private_key> [<public_key>]]`

  Generate a public–private pair of keys and write them to the corresponding files.

Switch parameter	Default value
<private_key>	`drwcsd.pri`
<public_key>	`drwcsd.pub`

The utility version for Windows platforms (in contrast to UNIX versions) does not protect private keys from copying.

- `drwsign gencert [-private-key=<private_key>] [-subj=<subject_fields>] [-days=<validity_period>] [<self_signed_certificate>]`

  Generate a self-signed certificate using the Dr.Web Server private key and write it to the corresponding file.

Switch parameter	Default value
<private_key>	`drwcsd.pri`
<subject_fields>	`/CN=<hostname>`
<validity_period>	`1095`
<self_signed_certificate>	`drwcsd-certificate.pem`

- `drwsign gencsr [-private-key=<private_key>] [-subj=<subject_fields>] [<certificate_sign_request>]`

  Generate a request for the certificate signature based on the private key and write this request to the corresponding file.

  Can be used to sign the certificate of another server, e.g. to sign a Dr.Web Proxy Server certificate with the Dr.Web Server key.

  To sign such requests, use the `signcsr` switch.

Switch parameter	Default value
<private_key>	`drwcsd.pri`
<subject_fields>	`/CN=<hostname>`
<certificate_sign_request>	`drwcsd-certificate-sign-request.pem`

- `drwsign genselfsign [-show] [-subj=<subject_fields>] [-days=<validity_period>] [<private_key> [<self_signed_certificate>]]`

  Generate a self-signed RSA certificate and an RSA private key for a web server and write them to the corresponding files.

The `-show` switch prints certificate content in a readable view.

Switch parameter	Default value
<*subject_fields*>	`/CN=`<*hostname*>
<*validity_period*>	`730`
<*private_key*>	`private-key.pem`
<*self_signed_certificate*>	`certificate.pem`

- `drwsign hash-check [-public-key=`<*public_key*>`]` <*hash_file*> <*signature_file*>

  Check the signature of the specified 256-bit number in the client-server protocol format.

  In the <*hash_file*> parameter, the file with the 256-bit number to sign is specified. The <*signature_file*> file is the signature result (two 256-bit numbers).

Switch parameter	Default value
<*public_key*>	`drwcsd.pub`

- `drwsign hash-sign [-private-key=`<*private_key*>`]` <*hash_file*> <*signature_file*>

  Sign the specified 256-bit number in the client-server protocol format.

  In the <*hash_file*> parameter, the file with the 256-bit number to sign is specified. The <*signature_file*> file is the signature result (two 256-bit numbers).

Switch parameter	Default value
<*private_key*>	`drwcsd.pri`

- `drwsign help [`<*command*>`]`

  Print brief information on the program or on the specific command in the command line format.

- `drwsign sign [-private-key=`<*private_key*>`]` <*file*>

  Sign <*file*> using the private key.

Switch parameter	Default value
<*private_key*>	`drwcsd.pri`

- `drwsign signcert [-ca-key=`<*private_key*>`] [-ca-cert=`<*Dr.Web_Server_certificate*>`] [-cert=`<*certificate_to_sign*>`] [-days=`<*validity_period*>`] [-eku=`<*purpose*>`] [`<*signed_certificate*>`]`

  Sign the existing <*certificate_to_sign*> using the private key and the certificate of Dr.Web Server. The signed certificate is saved into a separate file.

  Can be used to sign the Dr.Web Proxy Server certificate with the Dr.Web Server key.

  The following values of the `-eku` switch (Extended Key Usage extension) can be used:

- `drwebServerAuth`—authentication of the Server/Proxy server by Dr.Web Agent,
- `drwebMeshDAuth`—authentication of the Scanning server by the Virtual agent.

Switch parameter	Default value
<*private_key*>	`drwcsd.pri`
<*Dr.Web_Server_certificate*>	`drwcsd-ca-cerificate.pem`
<*certificate_to_sign*>	`drwcsd-certificate.pem`
<*validity_period*>	`365`
<*purpose*>	`drwebServerAuth`
<*signed_certificate*>	`drwcsd-signed-certificate.pem`

- `drwsign signcsr [-ca-key=`<*private_key*>`] [-ca-cert=`<*Dr.Web_Server_certificate*>`] [-csr=`<*certificate_sign_request*>`] [-days=`<*validity_period*>`] [-eku=`<*purpose*>`] [`<*signed_certificate*>`]`

  Sign <*certificate_sign_request*> generated by the `gencsr` command using the private key and the Dr.Web Server certificate. The signed certificate is saved into a separate file.

  Can be used to sign the certificate of another server, e.g. to sign a Dr.Web Proxy Server certificate with the Dr.Web Server key.

  The following values of the `-eku` switch (Extended Key Usage extension) can be used:

  - `drwebServerAuth`—authentication of the Server/Proxy server by Dr.Web Agent,
  - `drwebMeshDAuth`—authentication of the Scanning server by the Virtual agent.

Switch parameter	Default value
<*private_key*>	`drwcsd.pri`
<*Dr.Web_Server_certificate*>	`drwcsd-cerificate.pem`
<*certificate_sign_request*>	`drwcsd-certificate-sign-request.pem`
<*validity_period*>	`365`
<*purpose*>	`drwebServerAuth`
<*signed_certificate*>	`drwcsd-signed-certificate.pem`

- `drwsign tlsticketkey [`<*TLS_ticket*>`]`

  Generate a TLS ticket.

  Can be used in a Server cluster for shared TLS sessions.

Switch parameter	Default value
<*TLS_ticket*>	`tickets-key.bin`

- `drwsign verify [-ss-cert] [-CAfile=`<*Dr.Web_Server_certificate*>`] [`<*certificate_to_check*>`]`

  Check the validity of the certificate with the trusted certificate of the Server.

  The `-ss-cert` switch prescribes to ignore the trusted certificate and validate the self-signed certificate only.

Switch parameter	Default value
<*Dr.Web_Server_certificate*>	`drwcsd-certificate.pem`
<*certificate_to_check*>	`drwcsd-signed-certificate.pem`

- `drwsign x509dump [`<*certificate_to_print*>`]`

  Print the dump of any x509 certificate.

Switch parameter	Default value
<*certificate_to_print*>	`drwcsd-certificate.pem`

- `drwsign version`

  Show the utility version.

G7.2. Administrating Utility of the Embedded Database

For embedded database (SQLite3) management, you are provided with the `drwidbsh3` utility.

`drwidbsh3` resides in the following folders:

- on **Linux** OS: `/opt/drwcs/bin`
- on **FreeBSD** OS: `/usr/local/drwcs/bin`
- on **Windows** OS: <*Dr.Web_Server_installation_folder*>`\bin`

  (default Dr.Web Server installation folder is: `C:\Program Files\DrWeb Server`).

The start instruction format

`drwidbsh3` <*full_DB_filename*>

The program operates in the text dialog mode, meaning it waits for instructions from a user (all instructions shall begin with a period).

To receive help on other instructions, type `.help`.

For more information, use reference manuals for the SQL language.

G7.3. Dr.Web Server Remote Diagnostics Utility

Dr.Web Server remote diagnostics utility allows remotely connect to Dr.Web Server for basic controlling and viewing the operation statistics. Graphical version of the utility is available for Windows OS only.

You can download the utility via the Control Center, the **Administration** item in the main menu, the **Utilities** item in the control menu:

- For Windows OS—graphical version.
- For Unix-like OS—console version.

The following versions of Dr.Web Server remote diagnostics utility are provided:

Executable file	Location	Description
`drweb-cntl-<OS>-<bitness>`	Control Center, the **Administration** → **Utilities** section	Independent version of the utility. Can be launched from any directory or on any computer with corresponding operating system.
	The `webmin/utilities` Dr.Web Server directory	
`drwcntl`	The `bin` Dr.Web Server directory	Utility version depends on server libraries. Can be launched only from its location directory.

The `drweb-cntl-<OS>-<bitness>` and `drwcntl` version of the utility are similar in their functions. Further in the section, the `drwcntl` version is given, but all examples are relevant for both versions.

For connection of the Dr.Web Server remote diagnostics utility, you must enable Dr.Web Server FrontDoor extension. To do this, in the **Dr.Web Server configuration** section, on the **Modules** tab, set the **Dr.Web Server FrontDoor extension** flag.

For connection of the Dr.Web Server remote diagnostics utility, administrator that connects via the utility, must have the **Use additional features** permission. Otherwise, access to Dr.Web Server via the remote diagnostics utility will be forbidden.

For connection of the utility (both graphical and console) using TLS, you must directly specify the protocol when setting the Dr.Web Server address: `ssl://<IP address or DNS name>`.

The Dr.Web Server settings to connect Dr.Web Server remote diagnostics utility are given in the **Administrator Manual**, p. Dr.Web Server Remote Access (read online).

Utility Console Version

The start instruction format

`drwcntl [-?|-h|--help] [+<log_file>] [<server> [<login> [<password>]]]`

where:

- `-? -h --help`—show help message on commands for using the utility.
- <*log_file*>—write all utility actions into the log file by the specified path.
- <*server*>—address string of Dr.Web Server, to which the utility connects, in the following format: `[(tcp|ssl)://]`<*IP address or DNS name*>`[:`<*port*>`]`.

  To be able to connect via the one of the supported protocols, it is necessary to meet the following conditions:

  a) To connect via `ssl`, in the `frontdoor.conf` configuration file, the `<ssl />` tag must be set. At this, the connection can be established via `ssl` only.

  b) To connect via `tcp`, in the `frontdoor.conf` configuration file, the `<ssl />` tag must be disabled (commented). At this, the connection can be established via `tcp` only.

  If connection parameters are not set in the Dr.Web Server address string, the following values are used:

Parameter	Default value
Connection protocol	`tcp` For the TCP connection, the **Use TLS** flag must be cleared in the Control Center, in the **Administration** → **Dr.Web Server remote access** section. This disables the `<ssl />` tag in the `frontdoor.conf` configuration file.
IP address or DNS name of Dr.Web Server	Utility prompts you to specify the Dr.Web Server address in the corresponding format.
Port	`10101` At Dr.Web Server, allowed port is set in the **Dr.Web Server Remote Access** section and saved in the `frontdoor.conf` configuration file. If the alternative port is used in this section, it is necessary to set this port directly when connecting the utility.

- <*login*>—login of the Dr.Web Server administrator.
- <*password*>—administrative password to access Dr.Web Server.

If administrative login and password are not set in the connection string, the utility prompts you to specify corresponding credentials.

Possible commands

- `cache` <*operation*>—operations with file cache. To request the certain operation, use the following commands:
  - `clear`—clear the file cache,
  - `list`—show all file cache content,
  - `matched` <*regular expression*>—show file cache content which matches the specified regular expression,
  - `maxfilesize [`<*size*>`]`—show/set maximal size of preloaded file objects. When launched without additional parameters, shows the current size. To set the size, specify necessary size in bytes after the command name.
  - `statistics`—show statistics of file cache usage.
- `calculate` <*function*>—calculate specified sequence. To request the certain sequence, use the following commands:
  - `hash [`<*standard*>`] [`<*string*>`]`—calculate hash of specified string. To set the certain standard, use the following commands:
    - `gost`—calculate hash of specified string according to the GHOST standard,
    - `md5`—calculate md5 hash of specified string,
    - `sha`—calculate hash of specified string according to the SHA standard,
    - `sha1`—calculate hash of specified string according to the SHA1 standard,
    - `sha224`—calculate hash of specified string according to the SHA224 standard,
    - `sha256`—calculate hash of specified string according to the SHA256 standard,
    - `sha384`—calculate hash of specified string according to the SHA384 standard,
    - `sha512`—calculate hash of specified string according to the SHA512 standard.
  - `hmac [`<*standard*>`] [`<*string*>`]`—calculate HMAC of specified string. To set the certain standard, use the following commands:
    - `md5`—calculate the HMAC-MD5 for the specified string,
    - `sha256`—calculate the HMAC-SHA256 for the specified string.
  - `random`—generate random number,
  - `uuid`—calculate unique identifier.
- `clients` <*operation*>—get information and manage clients connected to Dr.Web Server. To request the certain function, use the following commands:
  - `addresses [`<*regular expression*>`]`—show stations network addresses that match specified regular expression. If the regular expression is not specified, show addresses of all stations.

- `caddresses [<regular expression>]`—show the number of station IP addresses that match specified regular expression. If the regular expression is not specified, show the number of all stations.
- `chosts [<regular expression>]`—show the number of station computer names that match specified regular expression. If the regular expression is not specified, show the number of all stations.
- `cids [<regular expression>]`—show the number of station identifiers that match specified regular expression. If the regular expression is not specified, show the number of all stations.
- `cnames [<regular expression>]`—show the number of station names that match specified regular expression. If the regular expression is not specified, show the number of all stations.
- `disconnect [<regular expression>]`—terminate current active connections with stations whose identifiers match specified regular expression. If the regular expression is not specified, terminate connection with all connected stations.
- `enable [<mode>]`—show/set the mode of accepting clients at Dr.Web Server. When launched without additional parameters, shows the current mode. To set the mode, use the following additional commands:
  - `on`—accept all client connections.
  - `off`—reject all client connections.
- `hosts <regular expression>`—show station computer names that match specified regular expression.
- `ids <regular expression>`—show station identifiers that match specified regular expression.
- `names <regular expression>`—show station names that match specified regular expression.
- `online <regular expression>`—show online time of the stations whose identifier, name or address match specified regular expression. Online time starts from the moment of last connection of the stations to Dr.Web Server.
- `statistics <regular expression>`—show statistics on number of clients that match specified regular expression.
- `traffic <regular expression>`—show traffic information of currently connected clients that match specified regular expression.

- `core`—write the Dr.Web Server process dump.
- `cpu <parameter>`—show statistics of the computer CPU usage on which Dr.Web Server is installed. To request the certain parameter, use the following commands:
  - `clear`—delete all accumulated statistic data,
  - `day`—show CPU loading graph for the current day,
  - `disable`—disable monitoring of CPU loading,
  - `enable`—enable monitoring of CPU loading,
  - `hour`—show CPU loading graph for the current hour,

  - `load`—show average CPU loading,
  - `minute`—show CPU loading graph for the passed minute,
  - `rawd`—show numeric statistic on CPU loading for the day,
  - `rawh`—show numeric statistic on CPU loading for the last hour,
  - `rawl`—show numeric statistic on average CPU loading,
  - `rawm`—show numeric statistic on CPU loading for the last minute,
  - `status`—show the monitoring state of CPU loading.
- `debug` <*parameter*>—debug configuration. To set the certain parameter, use the additional commands. To refine the additional commands list, you can call the help by the `? debug` command.

  

  The `debug signal` command is available for Dr.Web Servers under Unix-like OS only.

- `die`—stop Dr.Web Server and write the Dr.Web Server process dump.

  

  The `die` command is available for Dr.Web Servers under Unix-like OS only.

- `dwcp` <*parameter*>—set/show Dr.Web Control Protocol (includes Dr.Web Server, Dr.Web Agent and Dr.Web Agent installers protocols) options. Allowed parameters:
  - `compression` <*mode*>—set the one of the following traffic compression modes:
    - `on`—compression enabled,
    - `off`—compression disabled,
    - `possible`—compression is possible.
  - `encryption` <*mode*>—set the one of the following traffic encryption modes:
    - `on`—encryption enabled,
    - `off`—encryption disabled,
    - `possible`—encryption is possible.
  - `show`—show current Dr.Web Control Protocol options.
- `io` <*parameter*>—show input/output statistics of the Dr.Web Server process. To request the certain parameter, use the following command:
  - `clear`—delete all accumulated statistic data,
  - `disable`—disable statistics monitoring,
  - `enable`—enable statistics monitoring,
  - `rawdr`—show numeric statistic on data read for the day,
  - `rawdw`—show numeric statistic on data write for the day,
  - `rawh`—show numeric statistic for the last hour,
  - `rawm`—show numeric statistic for the last minute,

  - `rday`—show data read graph for the current day,
  - `rhour`—show data read graph for the last hour,
  - `rminute`—show data read graph for the last minute,
  - `status`—show statistics monitoring state,
  - `wday`—show data write graph for the day,
  - `whour`—show data write graph for the last hour,
  - `wminute`—show data write graph for the last minute.
- `log` <*parameter*>—write the string to the Dr.Web Server log file or set/view the log verbosity level. Depending on the specified parameters, the following actions are performed:
  - `log` <*string*>—write the specified string to the Dr.Web Server log file with the `NOTICE` verbosity level.
  - `log \s [`<*level*>`]`—set/show the log verbosity level. If the command launched with the `\s` command with no level specified, the current verbosity level is shown. Available values of the log verbosity level: `ALL`, `DEBUG3`, `DEBUG2`, `DEBUG1`, `DEBUG`, `TRACE3`, `TRACE2`, `TRACE1`, `TRACE`, `INFO`, `NOTICE`, `WARNING`, `ERROR`, `CRIT`.
- `lua`—execute LUA script.
- `mallopt` <*parameter*>—set the parameters of the memory allocation. To set the certain parameter, use the additional commands. To refine the additional commands list, you can call the help by the `? mallopt` command.

  

  The `mallopt` command is available for Dr.Web Servers under Linux-based OS only.

  To get more details on the command parameters features, refer the description of the `mallopt()` function from the `glibc` library. To get the help on this function, you can use the `man mallopt` command.
- `memory` <*parameter*>—show statistics of the computer memory usage on which Dr.Web Server is installed. To request the certain parameter, use the following commands:
  - `all`—shoe all information and statistic data,
  - `heap`—show information on dynamic memory,
  - `malloc`—show statistic on memory allocation,
  - `sizes`—show statistic on allocated memory sizes,
  - `system`—show information on system memory.

  

  The `memory` command is available for Dr.Web Servers under Windows OS, Linux-based OS and FreeBSD system-based OS only. At this, the following limitations on additional parameters of the `memory` command are active:
  - `system`—for Dr.Web Servers under Windows OS, Linux-based OS only,
  - `heap`—for Dr.Web Servers under Windows OS, Linux-based OS only,
  - `malloc`—for Dr.Web Servers under Linux-based OS and FreeBSD system-based OS only,

- `sizes`—for Dr.Web Servers under Linux-based OS and FreeBSD system-based OS only.

- `monitoring` <*mode*>—set/show monitoring mode of CPU (the `cpu` <*parameter*> command) and I/O (the `io` <*parameter*> command) resources usage by the Dr.Web Server process. Allowed parameters:
  - `disable`—disable monitoring,
  - `enable`—enable monitoring,
  - `show`—show current mode.
- `printstat`—write the Dr.Web Server operation statistics to the log.
- `reload`—reload Dr.Web Server FrontDoor extension.
- `repository` <*parameter*>—repository management. To request the certain function, use the following commands:
  - `all`—show the list of all repository products and the number of all files by products,
  - `clear`—clear cache content not depending on the TTL value of the objects in the cache,
  - `fill`—read all repository files into cache,
  - `keep`—store all repository files currently in the cache forever, not depending on their TTL value,
  - `loaded`—show the list of all repository products and the number of all files by products which are currently in the cache,
  - `reload`—reload repository from disk,
  - `statistics`—show repository updates statistics.
- `restart`—restart Dr.Web Server.
- `show` <*parameter*>—show the information about the system on which Dr.Web Server is installed. To set the certain parameter, use the additional commands. To refine the additional commands list, you can call the help by the `? show` command.

The following limitations on additional parameters of the `show` command are active:

- `memory`—for Dr.Web Servers under Windows OS, Linux-based OS only,
- `mapping`—for Dr.Web Servers under Windows OS, Linux-based OS only,
- `limits`—for Dr.Web Servers under Unix-like OS only,
- `processors`—for Dr.Web Servers under Linux-based OS only.

- `sql`—execute SQL query.
- `stop`—stop Dr.Web Server.
- `traffic` <*parameter*>—show statistics on the Dr.Web Server network traffic. To request the certain parameter, use the following commands:
  - `all`—show all the traffic from the Dr.Web Server start.
  - `incremental`—show traffic incrementation from the last launch of the `traffic incremental` command.

    - `last`—show traffic incrementation from the last stored point.
    - `store`—create the stored point for the `last` command.
- `update` <*parameter*>—get information and manage updates. To request the certain function, use the following commands:
    - `active`—show the list of Dr.Web Agents which are currently updating.
    - `agent [`<*mode*>`]`—show/set the mode of updating Dr.Web Agents from Dr.Web Server. When launched without additional parameters, shows the current mode. To set the mode, use the following additional commands:
        - `on`—enable Dr.Web Agents updates.
        - `off`—disable Dr.Web Agents updates.
    - `gus`—launch the repository update from the GUS ignoring the GUS update state.
    - `http [`<*mode*>`]`—show/set the mode of updating the Dr.Web Server repository from the GUS. When launched without additional parameters, shows the current mode. To set the mode, use the following additional commands:
        - `on`—enable repository updating from the GUS.
        - `off`—disable repository updating from the GUS.
    - `inactive`—show the list of Dr.Web Agents which are not currently updating.
    - `track [`<*mode*>`]`—show/set the mode of tracking Dr.Web Agents update. When launched without additional parameters, shows the current mode. To set the mode, use the following additional commands:
        - `on`—enable Dr.Web Agents update tracking.
        - `off`—disable Dr.Web Agents update tracking. At this, the `update active` command will not show the list of currently updating Dr.Web Agents.
- `version`—show the utility version.

G7.4. Dr.Web Server Remote Scriptable Diagnostics Utility

Dr.Web Server remote diagnostics utility allows you to remotely connect to Dr.Web Server for basic management and viewing of operation statistics. Unlike drwcntl, the `drwcmd` utility can be used for scripting.

The following console versions of the Dr.Web Server remote scriptable diagnostics utility are provided:

Executable file	Location	Description
`drweb-cmd-`<*OS*>`-`<*bitness*>	Control Center, the **Administration** → **Utilities** section	Independent version of the utility. Can be launched from any directory or on any computer with corresponding operating system.
	The `webmin/utilities` Dr.Web Server directory	

Executable file	Location	Description
`drwcmd`	The `bin` Dr.Web Server directory	Utility version depends on server libraries. Can be launched only from its location directory.

The `drweb-cmd-<OS>-<bitness>` and `drwcmd` version of the utility are similar in their functions. Further in the section, the `drwcmd` version is given, but all examples are relevant for both versions.

For connection of the Dr.Web Server remote diagnostics utility, you must enable Dr.Web Server FrontDoor extension. To do this, in the **Dr.Web Server configuration** section, on the **Modules** tab, set the **Dr.Web Server FrontDoor extension** flag.

For connection of the Dr.Web Server remote diagnostics utility, administrator that connects via the utility, must have the **Use additional features** permission. Otherwise, access to Dr.Web Server via the remote diagnostics utility will be forbidden.

The Dr.Web Server settings to connect Dr.Web Server remote diagnostics utility are given in the **Administrator Manual**, p. Dr.Web Server Remote Access (read online).

The start instruction format

`drwcmd [<switches>] [<files>]`

Allowed Switches

The `drwcmd` utility uses switches according the general rules described in the Appendix G. Command Line Parameters of the Programs Included in Dr.Web Enterprise Security Suite.

- `--?`—show help message on switches for using the utility.
- `--help`—show help message on switches for using the utility.
- `--commands=<commands>`—execute specified commands (similar to the drwcntl utility commands). To specify several commands, use the `;` sign as a separator.
- `--debug=yes|no`—log utility operations in the debug mode (the `stderr` standard output stream). Default is `no`.
- `--files=yes|no`—allow execution of the commands (similar to the drwcntl utility commands) from the specified files. Default is `yes`.

  Commands must be set in a file by one on each line. Empty lines are ignored. Use the `#` sign to start a comment.

- `--keep=yes|no`—keep the connection with Dr.Web Server after the last command is executed till the completion of the utility process. Default is `no`.
- `--output=<file>`—output file for the Dr.Web Server response. By default, if the file is not specified, the `stdout` standard output stream is used.

  If the file name starts with the (+), then the result of commands execution will be added to the end to file, otherwise—file will be rewritten.
- `--password=<password>`—password for the authorization at Dr.Web Server. Can be defined in the file set in the `--resource` switch.
- `--read=yes|no`—allow reading the Dr.Web Server connection parameters from the resource file. Default is `yes`.
- `--resource=<file>`—resource file with the Dr.Web Server connection parameters: the Dr.Web Server address and administrator credentials for the authorization at Dr.Web Server. By default, the `.drwcmdrc` file is used from the following directory:
  - For Unix-like OS: `$HOME`
  - For Windows OS: `%LOCALAPPDATA%`

  Each line in the file must contain 3 words separated by spaces: <*Dr.Web_Server*> <*user*> <*password*>.

  To specify the space in the middle of a word, use the `%S`. If you need the percent sign, use `%P`.

  For example:

```
ssl://127.0.0.1 user1 password1

ssl://127.0.0.1 user2 password2

ssl://127.0.0.1 user pass%Sword
```

  

  When using the `--resource` switch, you must also specify the `--server` switch. Utility connects to Dr.Web Server specified in the `--server` switch according to the credentials that corresponds to the address of this Dr.Web Server from the resource file.
- `--server=<Server>`—the Dr.Web Server address. Default is `ssl://127.0.0.1`. Can be defined in the file set in the `--resource` switch.
- `--user=<user>`—user name for the authorization at Dr.Web Server. Can be defined in the file set in the `--resource` switch.
- `--verbose=yes|no`—print detailed response of Dr.Web Server (the `stdout` standard output stream). Default is `no`.
- `--version`—show the utility version.

The procedure for connecting to Dr.Web Server:

1. When defining the data for the Dr.Web Server connection, the priority are given to the values specified in the switches `--server`, `--user` and `--password`.
2. If the `--server` switch is not specified, the default value is used—`ssl://127.0.0.1`.

3. If the `--user` switch is not specified, then the necessary Dr.Web Server is searched in the `.drwcmdrc` file (can be redefined in the `--resource` switch) and first user name is taken in the alphabetical order.
4. If the `--password` switch is not specified, then the search is performed in the `.drwcmdrc` file (can be redefined in the `--resource` switch) by Dr.Web Server and user name.

   

   User name and password will be read from the `.drwcmdrc` file (can be redefined in the `--resource` switch), if it is not forbidden by the `--read` switch.

5. If a user name and a password are not specified via the switches or the resource file, the utility prompts for credentials to be entered via the console.

Commands execution features:

- If the (-) values is set for the files with commands, then the utility reads command entered via the console.
- If both command in the `--commands` switch and the files list are set, then the commands from the `--commands` switch are executed first.
- If neither files of commands in the `--commands` switch are specified, then the commands entered via the console are read.

For example:

To execute the command from the `--command` switch and then a console commands, enter the following:

```
drwcmd --commands=<commands> -- -
```

Completion Codes

- 0—successful execution.
- 1—the help in switches is requested: `--help` or `--?`.
- 2—command line parse error: authorization parameters are not specified, etc.
- 3—cannot create output file for the Dr.Web Server response.
- 4—Dr.Web Server authorization error: wrong administrator's login and/or password.
- 5—Dr.Web Server connection terminated abnormally.
- 127—unknown fatal error.

G7.5. Dr.Web Repository Loader

Graphical version of the Repository Loader utility is described in the **Administrator Manual** document, in the p. GUI Utility (read online).

The following versions of Dr.Web Repository Loader console utility are provided:

Executable file	Location	Description
`drweb-reploader-<OS>-<bitness>`	Control Center, the **Administration** → **Utilities** section	Independent version of the utility. Can be launched from any directory or on any computer with corresponding operating system.
	The `webmin/utilities` Dr.Web Server directory	
`drwreploader`	The `bin` Dr.Web Server directory	Utility version depends on server libraries. Can be launched only from its location directory.

The `drweb-reploader-<OS>-<bitness>` and `drwreploader` version of the utility are similar in their functions. Further in the section, the `drwreploader` version is given, but all examples are relevant for both versions.

To simplify specifying the switches to run the console utility, you can use the Repository Loader configuration file. In the pre-installed configuration file, the switches values correspond to the default values listed below, except the `--ssh-auth` switch: its value is redefined to the `pubkey` in the configuration file.

Possible Switches

You can use a space or the "=" character between a key and its value.

- `--archive`—archive the repository.
- `--auth` <*argument*>—credentials for authorization on the update server in the following format: <*user*>`[:`<*password*>`]`.
- `--cert-file` <*path*>—path to the root certificates storage for SSL authorization.
- `--cert-mode [`<*argument*>`]`—the type of SSL certificates that will be automatically accepted. This option is used only for secure protocols that support encrypting.

  The <*argument*> may take one of the following values:

  - `any`—accept all certificates,
  - `valid`—accept only valid certificates,
  - `drweb`—accept only Dr.Web certificates,
  - `custom`—accept user-defined certificates.

  The `drweb` value is used by default.
- `--config` <*path*>—path to the Repository Loader configuration file.
- `--cwd` <*path*>—path to the current working directory.
- `--ipc`—enable the transfer of data on the utility operation to the standard output stream.

- `--help`—show help message on switches.
- `--license-key` <*path*>—path to the license key file (the key file or its MD5 hash must be specified).
- `--log` <*path*>—path to the log file on the repository downloading process.
- `--mode` <*mode*>—updates loading mode:
  - `repo`—repository is downloaded in the Dr.Web Server repository format. Loaded files can be directly imported via the Control Center as the Dr.Web Server repository updates. Value is used by default.
  - `mirror`—repository is downloaded in the GUS updates zone format. Loaded files can be placed on the updates mirror in your local network. Further, Dr.Web Servers can be configured to receive updates directly from this updates mirror containing the last version of the repository but not from the GUS servers.
- `--only-bases`—download only virus databases. The value of the `--product` key is ignored when this switch is specified.
- `--path` <*argument*>—download the repository from GUS to the folder specified as <*argument*>. When you archive the repository using the `--archive` switch, you can specify the path either to the folder name or to the archive file name. If the archive name is not specified, the `repository.zip` default name is used.
- `--product` <*argument*>—download the specified product. To download more than one product, specify a sequence of key-value pairs, for example: `--product=20-drwcs --product=40-drwproxy`, or the values separated by commas, for example: `--product 20-drwcs,40-drwproxy`. By default, the entire repository is downloaded.
- `--prohibit-cdn`—forbid CDN usage when downloading updates. CDN is allowed to be used by default.
- `--proto` <*protocol*>—updates loading protocol: `file | ftp | ftps | http | https | scp | sftp | smb | smbs`. Default is `https`.
- `--proxy-auth` <*argument*>—data for authentication on the proxy server: user login and password in the following format: <*login*>`[:`<*password*>`]`.
- `--proxy-auth-list` <*argument*>—method of authentication on the proxy server (`any` by default).

  <*argument*> may take one of the following values:
  - `none`—do not use authentication,
  - `any`—use any supported method,
  - `safe`—use any secure supported method,
  - the following methods (if several are used, specify the necessary ones separated by a space, for example: `--proxy-auth-list "digest ntlm"`):
    - `basic`,
    - `digest`,
    - `digestie`,
    - `ntlmwb`,

    - ntlm,
    - negotiate.
- --proxy-host <*argument*>—proxy server address specified in the following format: <*server*>[:<*port*>]. Default is 3128.
- --rotate <*N*><*f*>,<*M*><*u*>—Repository Loader log rotation mode. Same as the Dr.Web Server log rotation.

  By default, it is 10,10m, which means storing of 10 files 10 megabytes each, use compression.
- --server <*argument*>—GUS servers addresses. It is recommended to leave the default value: esuite.geo.drweb.com.
- --show-products—show the list of GUS products. Requires the --ipc switch to be specified as well.
- --ssh-auth <*type*>—type of the authorization on the update server when accessing by SCP/SFTP. For the <*type*> parameter, the one of the following values is allowed:
    - pwd—authorization using a password. A password is set in the --auth switch.
    - pubkey—authorization using a public key. You must specify a private key in the --ssh-prikey switch to extract corresponding public key.
- --ssh-prikey <*path*>—path to the SSH private key.
- --ssh-pubkey <*path*>—path to the SSH public key.
- --strict—terminate downloading if an error occurred.
- --update-key <*path*>—path to a public key or to a folder with a public key to validate the signature of updates that are loaded from GUS. The update-key-*.upub public keys to validate updates can be found on Dr.Web Server in the etc folder.
- --update-url <*argument*>—GUS server folder where updates of Dr.Web products are located. It is recommended that you leave the following values:
    - for non-certified product versions: /update;
    - for certified product versions: **Base URI** parameter value specified in the Control Center in the **Administration** → **General repository configuration** → **Dr.Web GUS** section.
- --V—show the utility version.
- --verbosity <*detail_level*>—log detail level. TRACE3 is by default. Allowed values are: ALL, DEBUG3, DEBUG2, DEBUG1, DEBUG, TRACE3, TRACE2, TRACE1, TRACE, INFO, NOTICE, WARNING, ERROR, CRIT. The ALL and DEBUG3 values are synonyms.
- --version <*version*>—Dr.Web Server version to download the updates for, in the following format: <*major_version*>.<*minor_version*>. For example, for Dr.Web Server of version 13, the <*version*> parameter is 13.00. The availability of updates for specific products may differ depending on Dr.Web Server version. One way to confirm the available products is to check the <**products**> parameter description for Repository loader's configuration file in the manual for desired version.

Switches Usage Features

When launching the Repository Loader, please note the following rules:

Switches must be specified	Condition
`--license-key`	Always
`--update-key`	
`--path`	
`--cert-file`	If the following switches take one of the values: • `--cert-mode valid | drweb | custom` • `--proto https | ftps | smbs`
`--ssh-prikey`	If the following switches take one of the values: • `--proto sftp | scp` • `--ssh-auth pubkey`

Examples of Use

1. To create an imported archive with all products:

```
drwreploader.exe --path C:\Temp --archive --license-key C:\agent.key --
update-key "C:\Program Files\DrWeb Server\etc" --cert-file "C:\Program
Files\DrWeb Server\etc"
```

2. To download only SpIDer Gate databases and Dr.Web Server:

```
drwreploader.exe --path C:\Temp --license-key C:\agent.key --update-key
"C:\Program Files\DrWeb Server\etc" --cert-file "C:\Program Files\DrWeb
Server\etc" --product 10-drwgatedb,20-drwcs
```

3. To download only virus databases to create an update mirror:

```
drwreploader.exe --path C:\Temp --license-key C:\agent.key --update-key
"C:\Program Files\DrWeb Server\etc" --cert-file "C:\Program Files\DrWeb
Server\etc" --mode mirror --only-bases
```

G7.6. Dr.Web Agent for UNIX Remote Installation Utility

Dr.Web Agent for UNIX remote installation utility lets you install Dr.Web Agent on protected workstations running a Unix-like operating system in your anti-virus network. The utility can also be used to install Dr.Web Server Security Suite (Unix), if necessary.

The utility has a command-line interface and it is available in several versions:

Executable file	Location	Description
`drweb-unix-install-<OS>-<bitness>`	Control Center, the **Administration** → **Utilities** section	Independent version, which can be run from any directory and on any computer with a corresponding operating system. It is updated together with the repository or Dr.Web Server.
	The `webmin/utilities` Dr.Web Server directory	
`drwunixinstall`	The `bin` Dr.Web Server directory	A version, which depends on available server libraries and therefore, can be run from its original location only. It is updated together with the Dr.Web Server only.

The `drweb-unix-install-<OS>-<bitness>` and `drwunixinstall` versions are the same in terms of features. Further in this section, the `drwunixinstall` version is used, but the format and allowed switches are the same for both versions.

The start instruction format

```
drwunixinstall [<switches>]
<station_1_IP_address>[:<SSH_port>[^<user_name>[^<password>]]]
<station_2_IP_address>[:<SSH_port>[^<user_name>[^<password>]]] ...
```

Allowed Switches

The `drwunixinstall` utility uses switches according to the general rules described in Appendix G. Command Line Parameters of the Programs Included in Dr.Web Enterprise Security Suite.

- `--help`—show the help message.
- `--ak <authorization_parameters>`—set the alternative parameters of authorization on remote workstations using encryption keys in the following format:

```
<user_name>
^<path_to_Dr.Web_Server_private_key>^<path_to_Dr.Web_Server_public_key>[^<private_key_password>]
```

If you set both the standard authorization parameters (the <*user_name*>^<*password*> pair) and the alternative parameters using encryption keys, the latter will be the used first when the utility is executed.

- `--ap` <*user_name*>^<*password*>—use the keyboard-interactive authorization on remote workstations.
- `--certificate` <*path*>—set a path to the Dr.Web Server certificate file. By default it is `webmin/install/unix/workstation/drwcsd-certificate.pem`.
- `--cpus` <*number*>—set a number of CPU cores to use for remote installation. By default it is `4`.
- `--ctimeout` <*time*>—set the maximum time to wait for installation package transferring to complete on remote workstations. Shall be specified in seconds, by default it is `600`.
- `--debug`—enable logging in debug mode. By default set to `no`.
- `--esuite` <*Dr.Web_Server_address*>—enter address of Dr.Web Server to be used for remote installation. Dr.Web Agent will connect to this Dr.Web Server once the installation is complete. Shall be specified in the following format: `[udp://]`<*IP address or DNS name*>`[:`<*port*>`]`
- `--etimeout` <*time*>—set the maximum time to wait for package installation to complete on remote workstations. Shall be specified in seconds, by default it is `900`.
- `--from` <*path*>—set a path to a directory on Dr.Web Server containing the installation packages. By default it is `webmin/install/unix`.
- `--long`—enable logging with timestamps included. By default set to `no`.
- `--pwd` <*password*>—set a password for authorization on remote workstations while using the `su` and/or `sudo` command.
- `--remote-temp` <*path*>—set a path to a directory on remote workstations to temporarily store installation files and the Dr.Web Server's certificate. By default, the utility uses a temporary directory set in the operating system.
- `--server`—install Dr.Web Server Security Suite (Unix) instead of Dr.Web Agent. By default set to `no`.
- `--simultaneously` <*number*>—set the maximum number of Dr.Web Agent installations on remote workstations at the same time.
- `--sshdebug`—enable logging in debug mode, with extra details on all operations using the SSH protocol. By default set to `no`.
- `--sshwaitdebug`—enable logging in debug mode, with extra detail on all operations using the SSH protocol and timer-related operations. By default set to `no`.
- `--stimeout` <*time*>—set the maximum time to wait for password input allowing the use of the `su` and/or `sudo` commands on remote workstations. Shall be specified in seconds, by default it is `10`.
- `--su`—use the `su` command during the installation to elevate privileges up to the root level on remote workstations. By default set to `no`.

- `--sudo`—use the `sudo` command during the installation to elevate privileges up to the root level on remote workstations. By default set to `no`.
- `--temp` <*path*>—set a path to a directory on Dr.Web Server to temporarily store the certificate. By default, the utility uses a temporary directory set in the operating system.
- `--timeout` <*time*>—set the maximum time to wait to establish connection and authenticate on remote workstations. Shall be specified in seconds, by default it is `30`.
- `--verbosity` <*details_level*>—set a level of detail while logging. By default it is `info`. Allowed values are `all`, `debug3`, `debug2`, `debug1`, `debug`, `trace3`, `trace2`, `trace1`, `trace`, `info`, `notice`, `warning`, `error`, `crit`. The `all` and `debug3` values are synonyms.
- `--version`—show the utility version.

G7.7. Dr.Web Agent for Windows Remote Installation Utility

Dr.Web Agent for Windows remote installation utility lets you install Dr.Web Agent on protected workstations running Windows in your anti-virus network.

The utility has a command-line interface and it is available in several versions:

Executable file	Location	Description
`drweb-windows-install-`<*OS*>`-`<*bitness*>	Control Center, the **Administration** → **Utilities** section	Independent version, which can be run from any directory and on any computer with a corresponding operating system. It is updated together with the repository or Dr.Web Server.
	The `webmin/utilities` Dr.Web Server directory	
`drwwindowsinstall`	The `bin` Dr.Web Server directory	A version, which depends on available server libraries and therefore, can be run from its original location only. It is updated together with the Dr.Web Server only.

The `drweb-windows-install-`<*OS*>`-`<*bitness*> and `drwwindowsinstall` versions are the same in terms of features. Further in this section, the `drwwindowsinstall` version is used, but the format and allowed switches are the same for both versions.

The start instruction format

`drwwindowsinstall` <*switches*>

Allowed Switches

The `drwwindowsinstall` utility uses switches according to the general rules described in Appendix G. Command Line Parameters of the Programs Included in Dr.Web Enterprise Security Suite.

- `--help`—show the help message.
- `--console=yes|no`—print the utility logging to console. By default set to `no`.
- `--disable-v1=yes|no`—disable the SMB version 1 (SMBv1) protocol while the utility is working. By default set to `no`.
- `--distribution` <*file_name*>—manually enter the file name of the Dr.Web Agent installer, which will be run on remote workstations. By default it is `drwinst.exe`.
- `--install-address` <*workstation_IP_address*>—enter address of a remote station, where Dr.Web Agent will be installed to. If you specify several stations at once, make sure to separate their addresses with comma ("`,`") or semicolon ("`;`"), no spaces.
- `--install-certificate` <*path*>—set a path to the Dr.Web Server certificate file.
- `--install-clients` <*number*>—set the maximum number of the Dr.Web Agent installations on remote stations at a time. By default it is `8`.
- `--install-compression` <*mode*>—set the compression mode for traffic between Dr.Web Server and connected workstations: `on` to enable traffic compression, `off` to disable, `possible` to make it possible. The latter means that the mode depends on corresponding settings on Dr.Web Server. By default it is `possible`.
- `--install-encryption` <*mode*>—set the encryption mode for traffic between Dr.Web Server and connected workstations: `on` to enable traffic encryption, `off` to disable, `possible` to make it possible. The latter means that the mode depends on corresponding settings on Dr.Web Server. By default it is `possible`.
- `--install-language` <*language_code*>—set the installed Dr.Web Agent's interface language as a two-letter code according to the ISO 639-1 standard. If the switch is not used or Dr.Web Agent is not available in requested language, the remote station's default system language will be used instead.
- `--install-path` <*path*>—set a path to Dr.Web Agent installation directory on remote stations. By default it is `%ProgramFiles%\DrWeb`.
- `--install-register=yes|no`—register Dr.Web Agent in the system list of installed software once the installation is complete. By default set to `no`.
- `--install-server` <*Dr.Web_Server_address*>—enter address of Dr.Web Server to be used for remote installation. Dr.Web Agent will connect to this Dr.Web Server once the installation is complete. The switch format: `[udp://]`<*IP_address_or_DNS_name*>`[:`<*port*>`]`.
- `--install-timeout` <*time*>—set the maximum time to wait for the Dr.Web Agent installation to complete on remote workstations. Shall be specified in seconds, by default it is `300`.

- `--install-user <user_name>@<domain>:<password>` or `<domain>\<user_name>:<password>`—set user name and password to use for authorization on remote workstations.
- `--log <path>`—set a path to the utility log file. By default it is `drwsmb.log` in the `var` subdirectory of the Dr.Web Server installation directory.
- `--machine <name>`—specify the name to be assigned to a remote workstation in the Dr.Web Enterprise Security Suite anti-virus network once the Dr.Web Agent installation is compete and it is connected to Dr.Web Server. By default, a computer name registered in the operating system is used.
- `--rotate=<N><f>,<M><u>`—set the utility log rotation mode. The switch format is the same as the one used to manage the Dr.Web Server log rotation mode. By default it is `10,10m`.
- `--service-id <name_in_registry>`—set the registry key name to be assigned to the Dr.Web Agent remote installation service in the Windows registry. By default it is `DrWebRsvcRunner`.
- `--service-name <displayed_name>`—set the Dr.Web Agent remote installation service name as it is displayed in the Services snap-in. By default it is `Dr.Web Remote Runner Service`.
- `--target-root <directory_name>`—set the name of a directory in the administrative share on a remote station, which will be used to run the Dr.Web Agent installer copied from Dr.Web Server. By default it is `TEMP`.
- `--target-volume <share_name>`—set the administrative share name, which will contain the Dr.Web Agent installation files. By default it is `ADMIN$`.
- `--threads <number>`—set the number of I/O threads in a pool. By default set to `2`.
- `--verbosity <level>`—set the utility logging level. By default set to `trace`. Allowed values: `all, debug3, debug2, debug1, debug, trace3, trace2, trace1, trace, info, notice, warning, error, crit`. The `all` and `debug3` values are synonyms.
- `--version`—show the utility version.

Appendix H. Environment Variables Exported by Dr.Web Server

To simplify the setting of the processes run by Dr.Web Server on schedule, the data on location of the Dr.Web Server folders is required. To this effect, Dr.Web Server exports the following variables of started processes into the environment:

- `DRWCSD_HOME`—path to the root folder (installation folder). The switch value is `-home`, if it was set at the Dr.Web Server launch; otherwise the current folder at launch.
- `DRWCSD_BIN`—path to the folder with executable files. The switch value is `-bin-root`, if it was set at the Dr.Web Server launch; otherwise it is the `bin` subfolder of the root folder.
- `DRWCSD_VAR`—path to the folder to which Dr.Web Server has a write access and which is designed to store volatile files (for example, logs and repository files). The switchvalue is `-var-root`, if it was set at the Dr.Web Server launch; otherwise it is the var subfolder of the root folder.

Appendix I. Regular Expressions Used in Dr.Web Enterprise Security Suite

Some parameters of Dr.Web Enterprise Security Suite are specified in the form of regular expressions of the following types:

- Regular expressions of Lua language.

  Used for configure an automatic membership of anti-virus network stations into user groups.

  Detailed description of Lua language regular expressions is available at https://www.lua.org/manual/5.1/manual.html.
- Regular expressions of PCRE program library.

  Detailed description of PCRE library syntax is available at https://www.pcre.org/.

  This appendix contains only a brief description of the most common examples for using regular expressions of PCRE library.

I1. Options Used in PCRE Regular Expressions

Regular expressions are used in the configuration file and in Dr.Web Security Control Center when objects to be excluded from scanning in the Scanner settings are specified.

Regular expressions are written as follows:

```
qr{EXP}options
```

where `EXP` is the expression itself; options stands for the sequence of options (a string of letters), and `qr{}` is literal metacharacters. The whole construction looks as follows:

`qr{pagefile\.sys}i`—Windows NT OS swap file

Below goes the description of options and regular expressions. For more details visit https://www.pcre.org/pcre.txt.

- Option `'a'` is equivalent to `PCRE_ANCHORED`

  If this option is set, the pattern is forced to be "anchored", that is, it is constrained to match only at the first matching point in the string that is being searched (the "subject string"). The same result can also be achieved by appropriate constructs in the pattern itself.
- Option `'i'` is equivalent to `PCRE_CASELESS`

  If this option is set, letters in the pattern match both upper and lower case letters. This option can be changed within a pattern by a `(?i)` option setting.
- Option `'x'` is equivalent to `PCRE_EXTENDED`

  If this option is set, whitespace data characters in the pattern are totally ignored except when escaped or inside a character class. Whitespaces do not include the VT character (code 11). In addition, characters between an unescaped `#` outside a character class and a newline character inclusively are ignored. This option can be changed in the pattern by setting a `(?x)`

option. This option enables including comments inside complicated patterns. Note, however, that this applies only to data characters. Whitespaces may not appear in special character sequences in a pattern, for example within the `(?(` sequence which introduces a conditional subpattern.

- Option `'m'` is equivalent to `PCRE_MULTILINE`

  By default, PCRE treats the subject string as consisting of a single line of characters (even if it actually contains newlines). The "*start of line*" metacharacter "`^`" matches only in the beginning of a string, while the "*end of line*" metacharacter "`$`" matches only in the end of a string or before a terminating newline (unless `PCRE_DOLLAR_ENDONLY` is set).

  When PCRE_MULTILINE is set, the "*start of line*" and "*end of line*" metacharacters match any newline characters which immediately follow or precede them in the subject string as well as in the very beginning and end of a subject string. This option can be changed within a pattern by a (`?m`) option setting. If there are no "`\n`" characters in the subject string, or `^` or `$` are not present in the pattern, the PCRE_MULTILINE option has no effect.

- Option `'u'` is equivalent to `PCRE_UNGREEDY`

  This option inverts the "greediness" of the quantifiers so that they are not greedy by default, but become greedy if followed by "`?`". The same result can also be achieved by the (`?U`) option in the pattern.

- Option `'d'` is equivalent to `PCRE_DOTALL`

  If this option is set, a dot metacharacter in the pattern matches all characters, including newlines. Without it, newlines are excluded. This option can be changed within a pattern by a (`?s`) option setting. A negative class such as [`^a`] always matches newline characters, regardless of the settings of this option.

- Option `'e'` is equivalent to `PCRE_DOLLAR_ENDONLY`

  If this option is set, a dollar metacharacter in the pattern matches only at the end of the subject string. Without this option, a dollar also matches immediately before a newline at the end of the string (but not before any other newline characters). The `PCRE_DOLLAR_ENDONLY` option is ignored if `PCRE_MULTILINE` is set.

I2. Peculiarities of PCRE Regular Expressions

A *regular expression* is a pattern that is matched against a subject string from left to right. Most characters stand for themselves in a pattern, and match the corresponding characters in the subject.

The power of regular expressions comes from the ability to include alternatives and repetitions in the pattern. These are encoded in the pattern by the use of metacharacters, which do not stand for themselves but instead are interpreted in a special way.

There are two different sets of metacharacters: those recognized anywhere in a pattern except within square brackets, and those recognized in square brackets. Outside square brackets, the metacharacters are as follows:

Symbol	Value
\	general escape character with several uses
^	assert start of string (or line, in multiline mode)
$	assert end of string (or line, in multiline mode)
.	match any character except newline (by default)
[	start character class definition
]	end character class definition
|	start alternative branch
(	start subpattern
)	end subpattern
?	extends the meaning of (also 0 or 1 quantifier also quantifier minimizer
*	0 or more quantifier
+	1 or more quantifier also "possessive quantifier"
{	start min/max quantifier

Part of a pattern that is in square brackets is called a "character class". In a character class the only metacharacters are:

Symbol	Value
\	general escape character
^	negate the class, but only if the first character
-	indicates character range
[	POSIX character class (only if followed by POSIX syntax)

Symbol	Value
]	terminates the character class

Appendix J. Log Files Format

Events on Dr.Web Server (see **Administrator Manual**, p. Dr.Web Server Log (read online)) and Dr.Web Agent are logged into a text file, where every line is a separate message.

The format of a message line is as follows:

<year><month><day>.<hour><minute><second>.<centisecond> <message_type>
[<process_id>] <thread_name> [<message_source>] <message>

where:

- *<year><month><date>.<hour><minute><second>.<hundredth_of_second>*—exact date of message entry to the log file.
- *<message_type>*—log level:
  - **ftl** (**Fatal error**)—instructs to inform only of the most severe errors;
  - **err** (**Error**)—notify of operation errors;
  - **wrn** (**Warning**)—warn about errors;
  - **ntc** (**Notice**)—display important information messages;
  - **inf** (**Info**)—display information messages;
  - **tr0..3** (trace0..3—tracing)—enable tracing events according to the level of detail. (**Trace 3** instructs to log in the maximum level of detail);
  - **db0..3** (debug0..3—debugging)—instruct to log debugging events according to the level of detail (**Debug 3** instructs to log in the maximum level of detail).

  The **tr0..3** (**trace**) and **db0..3** (**debug**) levels of detail are applicable for messages for Dr.Web Enterprise Security Suite developers only.

- [*<process_id>*]—unique numerical identifier of the process within which the thread that wrote the message to the log file was executed. Under certain OS [*<process_id>*] may be represented as [*<process_id> <thread_id>*].
- *<thread_name>*—character representation of the thread within which the message was logged.
- [*<message_source>*]—name of the system that initiated logging the message. The source is not always present.
- *<message>*—text description according to the log level. It may include both a formal description of the event and the values of certain event-relevant variables.

For example:

1. 20081023.171700.74 inf [001316] mth:12 [Sch] Task "Purge unsent IS events" said OK

   where:

   - 20081023—*<year><month><date>*,

- `171700`—<*hour*><*minute*><*second*>,
- `74`—<*hundredth_of_second*>,
- `inf`—<*message_type*>,
- `[001316]`—`[`<*process_id*>`]`,
- `mth:12`—<*thread_name*>,
- `[Sch]`—`[`<*message_source*>`]`,
- `Task "Purge unsent IS events" said OK`—message about the correct performance of the **Purge unsent events** events task.

2. 20081028.135755.61 inf [001556] srv:0 tcp/10.3.0.55:3575/025D4F80:2: new connection at tcp/10.3.0.75:2193

   where:

   - `20081028`—<*year*><*month*><*date*>,
   - `135755`—<*hour*><*minute*><*second*>,
   - `61`—<*hundredth_of_second*>,
   - `inf`—<*message_type*>,
   - `[001556]`—`[`<*process_id*>`]`,
   - `srv:0`—<*thread_name*>,
   - `tcp/10.3.0.55:3575/025D4F80:2: new connection at tcp/10.3.0.75:2193`—message about having established a new connection through the specified socket.

Appendix K. Integration of Web API and Dr.Web Enterprise Security Suite

The **Web API** is described in the **Web API for Dr.Web Enterprise Security Suite** manual.

Application

Web API, when integrated to Dr.Web Enterprise Security Suite, provides functions for operation of transactions with accounts and automation of service users management. You can use it, for example, to create dynamic pages to receive requests from users and send them installation files.

Authentication

The HTTP(S) protocol is used to interact with Dr.Web Server. **Web API** accepts REST requests and replies with the XML. To get access to the Web API, the Basic HTTP authentication is used (in compliance with RFC 2617 standard). Contrary to RFC 2617 and related standards, the HTTP(S) server does not request credentials (i.e., Dr.Web Enterprise Security Suite administrator account name and its password) from the client.

Appendix L. Licenses

This section contains the list of the third-party software libraries which are used by the Dr.Web Enterprise Security Suite software, information on their licensing, and development project addresses.

Library	License	Project URL
asio	https://www.boost.org/LICENSE_1_0.txt*	https://think-async.com/Asio/
Base58	https://github.com/leafo/lua-base58/blob/master/LICENSE	https://github.com/leafo/lua-base58
boost	https://www.boost.org/LICENSE_1_0.txt*	https://www.boost.org/
brotli	MIT License**	https://github.com/google/brotli
bsdiff	Custom	http://www.daemonology.net/bsdiff/
c-ares	https://c-ares.org/license.html*	https://c-ares.org/
cairo	Mozilla Public License** GNU Lesser General Public License**	https://www.cairographics.org/
CodeMirror	MIT License**	https://codemirror.net/
curl	https://curl.se/docs/copyright.html*	https://curl.se/libcurl/
fontconfig	Custom	https://www.freedesktop.org/wiki/Software/fontconfig/
freetype	GNU General Public License** FreeType Project License (BSD like)	https://freetype.org/
GCC runtime libraries	GNU General Public License** with exception*	https://gcc.gnu.org/
HTMLayout	Custom	https://terrainformatica.com/a-homepage-section/htmlayout/
ICU	https://www.unicode.org/copyright.html#License*	https://icu.unicode.org/home
jemalloc	https://github.com/jemalloc/jemalloc/blob/dev/COPYING*	https://github.com/jemalloc/jemalloc
jQuery	MIT License** GNU General Public License**	https://jquery.com/

Library	License	Project URL
JSON	https://github.com/nlohmann/json/blob/develop/LICENSE.MIT	https://github.com/nlohmann/json
JSON4Lua	MIT License**	https://github.com/craigmj/json4lua
Leaflet	BSD License https://github.com/Leaflet/Leaflet/blob/main/LICENSE*	https://leafletjs.com
libjpeg-turbo	https://github.com/libjpeg-turbo/libjpeg-turbo/blob/main/LICENSE.md	https://libjpeg-turbo.org/
libpng	http://libpng.org/pub/png/src/libpng-LICENSE.txt*	http://libpng.org/pub/png/libpng.html
libradius	Juniper Networks, Inc.*	https://www.freebsd.org
libssh2	3-Clause BSD License https://github.com/libssh2/libssh2/blob/master/COPYING*	https://libssh2.org/
libxml2	MIT License**	https://gitlab.gnome.org/GNOME/libxml2/-/wikis/home
Linenoise NG	BSD license*	https://github.com/arangodb/linenoise-ng
lua	MIT License**	https://www.lua.org/
lua-xmlreader	MIT License**	https://asbradbury.org/projects/lua-xmlreader/
lzma	Public Domain	https://www.7-zip.org/sdk.html
ncurses	MIT License**	https://invisible-island.net/ncurses/announce.html
Net-snmp	http://www.net-snmp.org/about/license.html*	http://www.net-snmp.org/
nghttp2	MIT License**	https://nghttp2.org/
Noto Sans CJK	https://github.com/notofonts/noto-fonts/blob/main/LICENSE*	https://fonts.google.com/noto/use
OpenLDAP	https://www.openldap.org/software/release/license.html*	https://www.openldap.org
OpenSSL	https://openssl-library.org/source/license/index.html*	https://www.openssl.org/

Library	License	Project URL
Oracle Instant Client	https://www.oracle.com/downloads/licenses/instant-client-lic.html*	https://www.oracle.com
ParaType Free Font	https://www.paratype.ru/eula*	https://www.paratype.ru
pcre	https://www.pcre.org/licence.txt*	https://www.pcre.org/
pixman	MIT License**	https://pixman.org/
Prototype JavaScript framework	MIT License**	http://prototypejs.org/assets/2009/8/31/prototype.js
quirc	https://github.com/dlbeer/quirc/blob/master/LICENSE	https://github.com/dlbeer/quirc/
QR Code generator	https://github.com/nayuki/QR-Code-generator	https://www.nayuki.io/page/qr-code-generator-library
script.aculo.us scriptaculous.js	https://madrobby.github.io/scriptaculous/license/*	http://script.aculo.us/
slt	MIT License**	https://code.google.com/archive/p/slt
SQLite	Public Domain https://www.sqlite.org/copyright.html	https://www.sqlite.org/index.html
wtl	Common Public License** Microsoft Public License**	https://sourceforge.net/projects/wtl/
zlib	https://www.zlib.net/zlib_license.html*	https://www.zlib.net/

*—license texts are listed further in this section.

**—basic license texts are found at the following:

License	Address
Common Public License	https://opensource.org/license/cpl1-0-txt
GNU General Public License	https://www.gnu.org/licenses/gpl-3.0.html
GNU Lesser General Public License	https://www.gnu.org/licenses/lgpl-3.0.html
Microsoft Public License	https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff649456(v=pandp.10)

License	Address
MIT License	https://opensource.org/license/mit
Mozilla Public License	https://www.mozilla.org/en-US/MPL/2.0/
3-Clause BSD License	https://opensource.org/license/bsd-3-clause

L1. Base58

```
The MIT License (MIT)


Copyright (c) 2015 leaf


Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:


The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.


THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

L2. Boost

```
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a
copy of the software and accompanying documentation covered by this license (the
"Software") to use, reproduce, display, distribute, execute, and transmit the Software,
and to prepare derivative works of the Software, and to permit third-parties to whom
the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above
license grant, this restriction and the following disclaimer, must be included in all
copies of the Software, in whole or in part, and all derivative works of the Software,
unless such copies or derivative works are solely in the form of machine-executable
object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS
OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY,
WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

L3. C-ares

```
Copyright (c) 2007 - 2018, Daniel Stenberg with many contributors, see AUTHORS file.

Copyright 1998 by the Massachusetts Institute of Technology.

Permission to use, copy, modify, and distribute this software and its documentation for
any purpose and without fee is hereby granted, provided that the above copyright notice
appear in all copies and that both that copyright notice and this permission notice
appear in supporting documentation, and that the name of M.I.T. not be used in
advertising or publicity pertaining to distribution of the software without specific,
written prior permission. M.I.T. makes no representations about the suitability of this
software for any purpose. It is provided "as is" without express or implied warranty.
```

L4. Curl

```
Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or
without fee is hereby granted, provided that the above copyright notice and this
permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

```
Except as contained in this notice, the name of a copyright holder shall not be used in
advertising or otherwise to promote the sale, use or other dealings in this Software
without prior written authorization of the copyright holder.
```

L5. GCC runtime libraries—exception

```
GCC is Copyright (C) 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996,
1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Free Software
Foundation, Inc.

GCC 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
3, or (at your option) any later version.

GCC 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.

Files that have exception clauses are licensed under the terms of the GNU General
Public License; either version 3, or (at your option) any later version.

The following runtime libraries are licensed under the terms of the GNU General Public
License (v3 or later) with version 3.1 of the GCC Runtime Library Exception (included
in this file):

 - libgcc (libgcc/, gcc/libgcc2.[ch], gcc/unwind*, gcc/gthr*, gcc/coretypes.h,
gcc/crtstuff.c, gcc/defaults.h, gcc/dwarf2.h, gcc/emults.c, gcc/gbl-ctors.h, gcc/gcov-
io.h, gcc/libgcov.c,  gcc/tsystem.h, gcc/typeclass.h).

 - libdecnumber

 - libgomp

 - libssp

 - libstdc++-v3

 - libobjc

 - libmudflap

 - libgfortran

 - The libgnat-4.4 Ada support library and libgnatvsn library.

 - Various config files in gcc/config/ used in runtime libraries.

GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

Copyright (C) 2009 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document,
but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under
section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given
```

```
file (the "Runtime Library") that bears a notice placed by the copyright holder of the
file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header
files and runtime libraries with the compiled program. The purpose of this Exception is
to allow compilation of non-GPL (including proprietary) programs to use, in this way,
the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for
execution after a Compilation Process, or makes use of an interface provided by the
Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications,
governed by version 3 (or a specified later version) of the GNU General Public License
(GPL) with the option of using any subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and
use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor
architecture, in executable form or suitable for input to an assembler, loader, linker
and/or execution phase. Notwithstanding that, Target Code does not include data in any
format that is used as a compiler intermediate representation, or used for producing a
compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in non-intermediate
languages designed for human-written code, and/or in Java Virtual Machine byte code,
into Target Code. Thus, for example, use of source code generators and preprocessors
need not be considered part of the Compilation Process, since the Compilation Process
can be understood as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-
compatible software, or if it is done without using any work based on GCC. For example,
using non-GPL-compatible Software to optimize any GCC intermediate representations
would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime
Library with Independent Modules, even if such propagation would otherwise violate the
terms of GPLv3, provided that all Target Code was generated by Eligible Compilation
Processes. You may then convey such a combination under terms of your choice,
consistent with the licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-
party software is unaffected by the copyleft requirements of the license of GCC.
```

L6. ICU

```
Copyright © 1991-2018 Unicode, Inc. All rights reserved.

Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the
Unicode data files and any associated documentation (the "Data Files") or Unicode
```

```
software and any associated documentation (the "Software") to deal in the Data Files or
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, and/or sell copies of the Data Files or Software,
and to permit persons to whom the Data Files or Software are furnished to do so,
provided that either (a) this copyright and permission notice appear with all copies of
the Data Files or Software, or (b) this copyright and permission notice appear in
associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR
A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE
DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in
advertising or otherwise to promote the sale, use or other dealings in these Data Files
or Software without prior written authorization of the copyright holder.
```

L7. Jemalloc

```
Unless otherwise specified, files in the jemalloc source distribution are subject to
the following license:

--------------------------------------------------------------------------------

Copyright (C) 2002-2018 Jason Evans <jasone@canonware.com>.

All rights reserved.

Copyright (C) 2007-2012 Mozilla Foundation.  All rights reserved.

Copyright (C) 2009-2018 Facebook, Inc.  All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice(s), this list
of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice(s), this
list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER(S) ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT HOLDER(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------------------
```

L8. JSON

```
MIT License


Copyright (c) 2013-2022 Niels Lohmann


Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:


The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.


THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

L9. Leaflet

```
Copyright (c) 2010-2018, Vladimir Agafonkin

Copyright (c) 2010-2011, CloudMade

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

```
   1. Redistributions of source code must retain the above copyright notice, this list
of conditions and the following disclaimer.

   2. Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

L10. libjpeg-turbo

```
libjpeg-turbo note:  This file has been modified by The libjpeg-turbo
Project

to include only information relevant to libjpeg-turbo, to wordsmith certain

sections, and to remove impolitic language that existed in the libjpeg v8

README.  It is included only for reference.  Please see README.md for

information specific to libjpeg-turbo.



The Independent JPEG Group's JPEG software

=========================================


This distribution contains a release of the Independent JPEG Group's free
JPEG

software.  You are welcome to redistribute this software and to use it for
any

purpose, subject to the conditions under LEGAL ISSUES, below.


This software is the work of Tom Lane, Guido Vollbeding, Philip Gladstone,

Bill Allombert, Jim Boucher, Lee Crocker, Bob Friesenhahn, Ben Jackson,

Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Ge' Weijers,
```

```
and other members of the Independent JPEG Group.


IJG is not affiliated with the ISO/IEC JTC1/SC29/WG1 standards committee
(also known as JPEG, together with ITU-T SG16).



DOCUMENTATION ROADMAP
=====================

This file contains the following sections:

OVERVIEW            General description of JPEG and the IJG software.
LEGAL ISSUES        Copyright, lack of warranty, terms of distribution.
REFERENCES          Where to learn more about JPEG.
ARCHIVE LOCATIONS   Where to find newer versions of this software.
FILE FORMAT WARS    Software *not* to get.
TO DO               Plans for future IJG releases.

Other documentation files in the distribution are:

User documentation:
  usage.txt         Usage instructions for cjpeg, djpeg, jpegtran,
                    rdjpgcom, and wrjpgcom.
  *.1               Unix-style man pages for programs (same info as
usage.txt).
  wizard.txt        Advanced usage instructions for JPEG wizards only.
  change.log        Version-to-version change highlights.
Programmer and internal documentation:
  libjpeg.txt       How to use the JPEG library in your own programs.
```

```
  example.txt       Sample code for calling the JPEG library.

  structure.txt     Overview of the JPEG library's internal structure.

  coderules.txt     Coding style rules --- please read if you contribute
code.


Please read at least usage.txt.  Some information can also be found in the
JPEG

FAQ (Frequently Asked Questions) article.  See ARCHIVE LOCATIONS below to
find

out where to obtain the FAQ article.


If you want to understand how the JPEG code works, we suggest reading one or

more of the REFERENCES, then looking at the documentation files (in roughly

the order listed) before diving into the code.



OVERVIEW

========


This package contains C software to implement JPEG image encoding, decoding,

and transcoding.  JPEG (pronounced "jay-peg") is a standardized compression

method for full-color and grayscale images.  JPEG's strong suit is
compressing

photographic images or other types of images that have smooth color and

brightness transitions between neighboring pixels.  Images with sharp lines
or

other abrupt features may not compress well with JPEG, and a higher JPEG

quality may have to be used to avoid visible compression artifacts with such

images.
```

```
JPEG is lossy, meaning that the output pixels are not necessarily identical
to

the input pixels.  However, on photographic content and other "smooth"
images,

very good compression ratios can be obtained with no visible compression

artifacts, and extremely high compression ratios are possible if you are

willing to sacrifice image quality (by reducing the "quality" setting in the

compressor.)


This software implements JPEG baseline, extended-sequential, and progressive

compression processes.  Provision is made for supporting all variants of
these

processes, although some uncommon parameter settings aren't implemented yet.

We have made no provision for supporting the hierarchical or lossless

processes defined in the standard.


We provide a set of library routines for reading and writing JPEG image
files,

plus two sample applications "cjpeg" and "djpeg", which use the library to

perform conversion between JPEG and some other popular image file formats.

The library is intended to be reused in other applications.


In order to support file conversion and viewing software, we have included

considerable functionality beyond the bare JPEG coding/decoding capability;

for example, the color quantization modules are not strictly part of JPEG

decoding, but they are essential for output to colormapped file formats or

colormapped displays.  These extra functions can be compiled out of the

library if not required for a particular application.


We have also included "jpegtran", a utility for lossless transcoding between
```

```
different JPEG processes, and "rdjpgcom" and "wrjpgcom", two simple
applications for inserting and extracting textual comments in JFIF files.

The emphasis in designing this software has been on achieving portability
and
flexibility, while also making it fast enough to be useful.  In particular,
the software is not intended to be read as a tutorial on JPEG.  (See the
REFERENCES section for introductory material.)  Rather, it is intended to
be reliable, portable, industrial-strength code.  We do not claim to have
achieved that goal in every aspect of the software, but we strive for it.

We welcome the use of this software as a component of commercial products.
No royalty is required, but we do ask for an acknowledgement in product
documentation, as described under LEGAL ISSUES.



LEGAL ISSUES
============

In plain English:

1. We don't promise that this software works.  (But if you find any bugs,
   please let us know!)
2. You can use this software for whatever you want.  You don't have to pay
us.
3. You may not pretend that you wrote this software.  If you use it in a
   program, you must acknowledge somewhere in your documentation that
   you've used the IJG code.
```

```
In legalese:


The authors make NO WARRANTY or representation, either express or implied,

with respect to this software, its quality, accuracy, merchantability, or

fitness for a particular purpose.  This software is provided "AS IS", and
you,

its user, assume the entire risk as to its quality and accuracy.


This software is copyright (C) 1991-2020, Thomas G. Lane, Guido Vollbeding.

All Rights Reserved except as specified below.


Permission is hereby granted to use, copy, modify, and distribute this

software (or portions thereof) for any purpose, without fee, subject to
these

conditions:

(1) If any part of the source code for this software is distributed, then
this

README file must be included, with this copyright and no-warranty notice

unaltered; and any additions, deletions, or changes to the original files

must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying

documentation must state that "this software is based in part on the work of

the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts

full responsibility for any undesirable consequences; the authors accept

NO LIABILITY for damages of any kind.


These conditions apply to any software derived from or based on the IJG
code,

not just to the unmodified library.  If you use our work, you ought to
```

```
acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company
name
in advertising or publicity relating to this software or products derived
from
it.  This software may be referred to only as "the Independent JPEG Group's
software".

We specifically permit and encourage the use of this software as the basis
of
commercial products, provided that all warranty or liability claims are
assumed by the product vendor.



REFERENCES
==========

We recommend reading one or more of these references before trying to
understand the innards of the JPEG software.

The best short technical introduction to the JPEG compression algorithm is
        Wallace, Gregory K.  "The JPEG Still Picture Compression Standard",
        Communications of the ACM, April 1991 (vol. 34 no. 4), pp. 30-44.
(Adjacent articles in that issue discuss MPEG motion picture compression,
applications of JPEG, and related topics.)  If you don't have the CACM issue
handy, a PDF file containing a revised version of Wallace's article is
available at http://www.ijg.org/files/Wallace.JPEG.pdf.  The file (actually
a preprint for an article that appeared in IEEE Trans. Consumer Electronics)
omits the sample images that appeared in CACM, but it includes corrections
```

and some added material. Note: the Wallace article is copyright ACM and IEEE,

and it may not be used for commercial purposes.

A somewhat less technical, more leisurely introduction to JPEG can be found in

"The Data Compression Book" by Mark Nelson and Jean-loup Gailly, published by

M&T Books (New York), 2nd ed. 1996, ISBN 1-55851-434-1. This book provides good explanations and example C code for a multitude of compression methods including JPEG. It is an excellent source if you are comfortable reading C code but don't know much about data compression in general. The book's JPEG sample code is far from industrial-strength, but when you are ready to look at a full implementation, you've got one here...

The best currently available description of JPEG is the textbook "JPEG Still Image Data Compression Standard" by William B. Pennebaker and Joan L. Mitchell, published by Van Nostrand Reinhold, 1993, ISBN 0-442-01272-1. Price US$59.95, 638 pp. The book includes the complete text of the ISO JPEG standards (DIS 10918-1 and draft DIS 10918-2).

The original JPEG standard is divided into two parts, Part 1 being the actual

specification, while Part 2 covers compliance testing methods. Part 1 is titled "Digital Compression and Coding of Continuous-tone Still Images, Part 1: Requirements and guidelines" and has document numbers ISO/IEC IS 10918-1, ITU-T T.81. Part 2 is titled "Digital Compression and Coding of Continuous-tone Still Images, Part 2: Compliance testing" and has document numbers ISO/IEC IS 10918-2, ITU-T T.83.

```
The JPEG standard does not specify all details of an interchangeable file
format.  For the omitted details, we follow the "JFIF" conventions, revision
1.02.  JFIF version 1 has been adopted as ISO/IEC 10918-5 (05/2013) and
Recommendation ITU-T T.871 (05/2011): Information technology - Digital
compression and coding of continuous-tone still images: JPEG File
Interchange
Format (JFIF).  It is available as a free download in PDF file format from
https://www.iso.org/standard/54989.html and http://www.itu.int/rec/T-REC-
T.871.
A PDF file of the older JFIF 1.02 specification is available at
http://www.w3.org/Graphics/JPEG/jfif3.pdf.


The TIFF 6.0 file format specification can be obtained from
http://mirrors.ctan.org/graphics/tiff/TIFF6.ps.gz.  The JPEG incorporation
scheme found in the TIFF 6.0 spec of 3-June-92 has a number of serious
problems.  IJG does not recommend use of the TIFF 6.0 design (TIFF
Compression
tag 6).  Instead, we recommend the JPEG design proposed by TIFF Technical
Note
#2 (Compression tag 7).  Copies of this Note can be obtained from
http://www.ijg.org/files/.  It is expected that the next revision
of the TIFF spec will replace the 6.0 JPEG design with the Note's design.
Although IJG's own code does not support TIFF/JPEG, the free libtiff library
uses our library to implement TIFF/JPEG per the Note.



ARCHIVE LOCATIONS
=================


The "official" archive site for this software is www.ijg.org.
```

```
The most recent released version can always be found there in

directory "files".


The JPEG FAQ (Frequently Asked Questions) article is a source of some

general information about JPEG.  It is available at

http://www.faqs.org/faqs/jpeg-faq.



FILE FORMAT COMPATIBILITY

=========================


This software implements ITU T.81 | ISO/IEC 10918 with some extensions from

ITU T.871 | ISO/IEC 10918-5 (JPEG File Interchange Format-- see REFERENCES).

Informally, the term "JPEG image" or "JPEG file" most often refers to JFIF
or

a subset thereof, but there are other formats containing the name "JPEG"
that

are incompatible with the DCT-based JPEG standard or with JFIF (for
instance,

JPEG 2000 and JPEG XR).  This software therefore does not support these

formats.  Indeed, one of the original reasons for developing this free
software

was to help force convergence on a common, interoperable format standard for

JPEG files.


JFIF is a minimal or "low end" representation.  TIFF/JPEG (TIFF revision 6.0
as

modified by TIFF Technical Note #2) can be used for "high end" applications

that need to record a lot of additional data about an image.
```

```
TO DO

=====



Please send bug reports, offers of help, etc. to jpeg-info@jpegclub.org.
```

```
Copyright (C)2009-2022 D. R. Commander. All Rights Reserved.

Copyright (C)2015 Viktor Szathmáry. All Rights Reserved.



Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:



Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the libjpeg-turbo Project nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS",
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

```
Copyright (c) <year> <copyright holders>



This software is provided 'as-is', without any express or implied warranty.
In no event will the authors be held liable for any damages arising from the
use of this software.
```

```
Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:



1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software in a
product, an acknowledgment in the product documentation would be appreciated
but is not required.



2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.



3. This notice may not be removed or altered from any source distribution.
```

L11. Libpng

```
If you modify libpng you may insert additional notices immediately following this
sentence.

This code is released under the libpng license.

libpng versions 1.0.7, July 1, 2000 through 1.6.32, August 24, 2017 are Copyright (c)
2000-2002, 2004, 2006-2017 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and
are distributed according to the same disclaimer and license as libpng-1.0.6 with the
following individuals added to the list of Contributing Authors:

   Simon-Pierre Cadieux

   Eric S. Raymond

   Mans Rullgard

   Cosmin Truta

   Gilles Vollant

   James Yu

   Mandar Sahastrabuddhe

   Google Inc.

   Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against
infringement.  There is no warranty that our efforts or the library will fulfill any of
your particular purposes or needs.  This library is provided with all faults, and the
```

```
entire risk of satisfactory quality, performance, accuracy, and effort is with the
user.

Some files in the "contrib" directory and some configure-generated files that are
distributed with libpng have other copyright owners and are released under other open
source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c)
1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed
according to the same disclaimer and license as libpng-0.96, with the following
individuals added to the list of Contributing Authors:

   Tom Lane

   Glenn Randers-Pehrson

   Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997
Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same
disclaimer and license as libpng-0.88, with the following individuals added to the list
of Contributing Authors:

   John Bowler

   Kevin Bracey

   Sam Bushell

   Magnus Holmgren

   Greg Roelofs

   Tom Tanner

Some files in the "scripts" directory have other copyright owners but are released
under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996
Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as
the following set of individuals:

   Andreas Dilger

   Dave Martindale

   Guy Eric Schalnat

   Paul Schmidt

   Tim Wegner

The PNG Reference Library is supplied "AS IS".  The Contributing Authors and Group 42,
Inc. disclaim all warranties, expressed or implied, including, without limitation, the
warranties of merchantability and of fitness for any purpose.  The Contributing Authors
and Group 42, Inc. assume no liability for direct, indirect, incidental, special,
exemplary, or consequential damages, which may result from the use of the PNG Reference
Library, even if advised of the possibility of such damage.
```

```
Permission is hereby granted to use, copy, modify, and distribute this source code, or
portions hereof, for any purpose, without fee, subject to the following restrictions:

  1. The origin of this source code must not be misrepresented.

  2. Altered versions must be plainly marked as such and must not be misrepresented as
being the original source.

  3. This Copyright notice may not be removed or altered from any source or altered
source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and
encourage the use of this source code as a component to supporting the PNG file format
in commercial products.  If you use this source code in a product, acknowledgment is
not required but would be appreciated.



Glenn Randers-Pehrson

glennrp at users.sourceforge.net

April 1, 2017
```

L12. Libradius

```
 Copyright 1998 Juniper Networks, Inc.

 All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

 1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

 2. Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

$FreeBSD: src/lib/libradius/radlib_private.h,v 1.6.30.3 2012/04/21 18:30:48 melifaro
Exp $
```

L13. Libssh2

L14. Linenoise NG

linenoise

```
  * Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

  * Neither the name of Redis nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

wcwidth

```
Markus Kuhn -- 2007-05-26 (Unicode 5.0)

Permission to use, copy, modify, and distribute this software for any purpose and
without fee is hereby granted. The author disclaims all warranties with regard to this
software.
```

ConvertUTF

```
Copyright 2001-2004 Unicode, Inc.

 Disclaimer

 This source code is provided as is by Unicode, Inc. No claims are made as to fitness
for any particular purpose. No warranties of any kind are expressed or implied. The
recipient agrees to determine applicability of information provided. If this file has
been purchased on magnetic or optical media from Unicode, Inc., the sole remedy for any
claim will be exchange of defective media within 90 days of receipt.

 Limitations on Rights to Redistribute This Code

 Unicode, Inc. hereby grants the right to freely use the information supplied in this
file in the creation of products supporting the Unicode Standard, and to make copies of
this file in any form for internal or external distribution as long as this notice
remains attached.
```

L15. Net-snmp

```
Various copyrights apply to this package, listed in various separate parts below.
Please make sure that you read all the parts.

---- Part 1: CMU/UCD copyright notice: (BSD like) -----

 Copyright 1989, 1991, 1992 by Carnegie Mellon University
```

```
Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

* Neither the name of Sparta, Inc nor the names of its contributors may be used to
endorse or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ''AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 6: Cisco/BUPTNIC copyright notice (BSD) -----

Copyright (c) 2004, Cisco, Inc and Information Network

Center of Beijing University of Posts and Telecommunications.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

* Neither the name of Cisco, Inc, Beijing University of Posts and Telecommunications,
nor the names of their contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ''AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 7: Fabasoft R&D Software GmbH & Co KG copyright notice (BSD) -----

Copyright (c) Fabasoft R&D Software GmbH & Co KG, 2003

oss@fabasoft.com
```

```
Author: Bernhard Penz

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

* The name of Fabasoft R&D Software GmbH & Co KG or any of its subsidiaries, brand or
product names may not be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ''AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 8: Apple Inc. copyright notice (BSD) -----

Copyright (c) 2007 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

3. Neither the name of Apple Inc. ("Apple") nor the names of its contributors may be
used to endorse or promote products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY APPLE AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
APPLE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---- Part 9: ScienceLogic, LLC copyright notice (BSD) -----

Copyright (c) 2009, ScienceLogic, LLC

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

```
* Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

* Neither the name of ScienceLogic, LLC nor the names of its contributors may be used
to endorse or promote products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ''AS IS'' AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

L16. Noto Sans CJK

```
Copyright (c) <dates>, <Copyright Holder> (<URL|email>), with Reserved Font Name
<Reserved Font Name>.

Copyright (c) <dates>, <additional Copyright Holder> (<URL|email>), with Reserved Font
Name <additional Reserved Font Name>.

Copyright (c) <dates>, <additional Copyright Holder> (<URL|email>).

This Font Software is licensed under the SIL Open Font License, Version 1.1.

This license is copied below, and is also available with a FAQ at:

http://scripts.sil.org/OFL

-----------------------------------------------------------

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

-----------------------------------------------------------

PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of
collaborative font projects, to support the font creation efforts of academic and
linguistic communities, and to provide a free and open framework in which fonts may be
shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed
freely as long as they are not sold by themselves. The fonts, including any derivative
works, can be bundled, embedded, redistributed and/or sold with any software provided
that any reserved names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The requirement for fonts
to remain under this license does not apply to any document created using the fonts or
their derivatives.
```

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

L17. OpenLDAP

```
The OpenLDAP Public License

  Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation ("Software"), with
or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,

2. Redistributions in binary form must reproduce applicable copyright statements and
notices, this list of conditions, and the following disclaimer in the documentation
and/or other materials provided with the distribution, and

3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is
distinguished by a version number. You may use this Software under terms of this
license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ``AS IS'' AND
ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S)
OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE  POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or
otherwise to promote the sale, use or other dealing in this Software without specific,
written prior permission.  Title to copyright in this Software shall at all times
remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA.  All Rights
Reserved.  Permission to copy and distribute verbatim copies of this document is
granted.
```

L18. OpenSSL

```
Copyright (c) 1998-2018 The OpenSSL Project.  All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.
```

```
3. All advertising materials mentioning features or use of this software must display
the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL
Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or
promote products derived from this software without prior written permission. For
written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL"
appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL
Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL
PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====================================================================

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
 This product includes software written by Tim Hudson (tjh@cryptsoft.com).

 Original SSLeay License

 -----------------------

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following
conditions are aheared to.  The following conditions apply to all code found in this
distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code.  The
SSL documentation included with this distribution is covered by the same copyright
terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not
to be removed.

If this package is used in a product, Eric Young should be given attribution as the
author of the parts of the library used.

This can be in the form of a textual message at program startup or in documentation
(online or textual) provided with the package.
```

```
Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of
conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

3. All advertising materials mentioning features or use of this software must display
the following acknowledgement:

   "This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com)"

   The word 'cryptographic' can be left out if the rouines from the library being used
are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps
directory (application code) you must include an acknowledgement:

   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative
of this code cannot be changed.  i.e. this code cannot simply be copied and put under
another distribution licence [including the GNU Public Licence.]
```

L19. Oracle Instant Client

```
Export Controls on the Programs

Selecting the "Accept License Agreement" button is a confirmation of your agreement
that you comply, now and during the trial term, with each of the following statements:

-You are not a citizen, national, or resident of, and are not under control of, the
government of Cuba, Iran, Sudan, Libya, North Korea, Syria, nor any country to which
the United States has prohibited export.

-You will not download or otherwise export or re-export the Programs, directly or
indirectly, to the above mentioned countries nor to citizens, nationals or residents of
those countries.

-You are not listed on the United States Department of Treasury lists of Specially
Designated Nationals, Specially Designated Terrorists, and Specially Designated
Narcotic Traffickers, nor are you listed on the United States Department of Commerce
Table of Denial Orders.
```

You will not download or otherwise export or re-export the Programs, directly or indirectly, to persons on the above mentioned lists.

You will not use the Programs for, and will not allow the Programs to be used for, any purposes prohibited by United States law, including, without limitation, for the development, design, manufacture or production of nuclear, chemical or biological weapons of mass destruction.

EXPORT RESTRICTIONS

You agree that U.S. export control laws and other applicable export and import laws govern your use of the programs, including technical data; additional information can be found on Oracle®'s Global Trade Compliance web site (http://www.oracle.com/products/export).

You agree that neither the programs nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

Oracle Employees: Under no circumstances are Oracle Employees authorized to download software for the purpose of distributing it to customers. Oracle products are available to employees for internal use or demonstration purposes only. In keeping with Oracle's trade compliance obligations under U.S. and applicable multilateral law, failure to comply with this policy could result in disciplinary action up to and including termination.

Note: You are bound by the Oracle Technology Network ("OTN") License Agreement terms. The OTN License Agreement terms also apply to all updates you receive under your Technology Track subscription.

The OTN License Agreement terms below supercede any shrinkwrap license on the OTN Technology Track software CDs and previous OTN License terms (including the Oracle Program License as modified by the OTN Program Use Certificate).

Oracle Technology Network Development and Distribution License Agreement for Instant Client

"We," "us," and "our" refers to Oracle America, Inc. "You" and "your" refers to the individual or entity that wishes to use the Programs from Oracle under this Agreement. "Programs" refers to the Software Products referenced below that you wish to download and use and Program documentation. "License" refers to your right to use the Programs and Program documentation under the terms of this Agreement. The substantive and procedural laws of California govern this Agreement. You and Oracle agree to submit to the exclusive jurisdiction of, and venue in, the courts of San Francisco, San Mateo, or Santa Clara counties in California in any dispute arising out of or relating to this Agreement.

We are willing to license the Programs to you only upon the condition that you accept all of the terms contained in this Agreement. Read the terms carefully and select the "Accept" button at the bottom of the page to confirm your acceptance. If you are not willing to be bound by these terms, select the "Do Not Accept" button and the registration process will not continue.

Software Product

- Instant Client

License Rights

License.

We grant you a non-exclusive right and license to use the Programs solely for your business purposes and development and testing purposes, subject to the terms of this Agreement. You may allow third parties to use the Programs, subject to the terms of this Agreement, provided such third party use is for your business operations only.

Distribution License

We grant you a non-exclusive right and license to distribute the Programs, provided that you do not charge your end users for use of the Programs. Your distribution of such Programs shall at a minimum include the following terms in an executed license agreement between you and the end user that: (1) restrict the use of the Programs to the business operations of the end user; (2) prohibit (a) the end user from assigning, giving, or transferring the Programs or an interest in them to another individual or entity (and if your end user grants a security interest in the Programs, the secured party has no right to use or transfer the Programs); (b) make the Programs available in any manner to any third party for use in the third party's business operations (unless such access is expressly permitted for the specific program license or materials from the services you have acquired); and (c) title to the Programs from passing to the end user or any other party; (3) prohibit the reverse engineering (unless required by law for interoperability), disassembly or decompilation of the Programs and prohibit duplication of the Programs except for a sufficient number of copies of each Program for the end user's licensed use and one copy of each Program media; (4) disclaim, to the extent permitted by applicable law, our liability for any damages, whether direct, indirect, incidental, or consequential, arising from the use of the Programs; (5) require the end user at the termination of the Agreement, to discontinue use and destroy or return to you all copies of the Programs and documentation; (6) prohibit publication of any results of benchmark tests run on the Programs; (7) require the end user to comply fully with all relevant export laws and regulations of the United States and other applicable export and import laws to assure that neither the Programs, nor any direct product thereof, are exported, directly or indirectly, in violation of applicable laws; (8) do not require us to perform any obligations or incur any liability not previously agreed to between you and us; (9) permit you to audit your end user's use of the Programs or to assign your right to audit the end user's use of the Programs to us; (10) designate us as a third party beneficiary of the end user license agreement; (11) include terms consistent with those contained in the sections of this Agreement entitled "Disclaimer of Warranties and Exclusive Remedies," "No Technical Support," "End of Agreement," "Relationship Between the Parties," and "Open Source"; and (11) exclude the application of the Uniform Computer Information Transactions Act.

You may allow your end users to permit third parties to use the Programs on such end user's behalf for the purposes set forth in the end user license agreement, subject to the terms of such agreement. You shall be financially responsible for all claims and damages to us caused by your failure to include the required contractual terms set forth above in each end user license agreement between you and an end user. We are a third party beneficiary of any end user license agreement between you and the end user, but do not assume any of your obligations thereunder, and you agree that you will not enter into any end user license agreement that excludes us as a third party beneficiary and will inform your end users of our rights.

If you want to use the Programs for any purpose other than as expressly permitted under this Agreement you must contact us to obtain the appropriate license. We may audit your use of the Programs. Program documentation is either shipped with the Programs, or documentation may be accessed online at http://www.oracle.com/technetwork/indexes/documentation/index.html.

You agree to: (a) defend and indemnify us against all claims and damages caused by your distribution of the Programs in breach of this Agreement and/or failure to include the required contractual provisions in your end user agreement as stated above; (b) keep executed end user agreements and records of end user information including name, address, date of distribution and identity of Programs distributed; (c) allow us to inspect your end user agreements and records upon request; and, (d) enforce the terms of your end user agreements so as to effect a timely cure of any end user breach, and to notify us of any breach of the terms.

Ownership and Restrictions

We retain all ownership and intellectual property rights in the Programs. You may make a sufficient number of copies of the Programs for the licensed use and one copy of the Programs for backup purposes.

You may not:

-use the Programs for any purpose other than as provided above;

-charge your end users for use of the Programs;

-remove or modify any Program markings or any notice of our proprietary rights;

-assign this agreement or give the Programs, Program access or an interest in the Programs to any individual or entity except as provided under this agreement;

-cause or permit reverse engineering (unless required by law for interoperability), disassembly or decompilation of the Programs;

-disclose results of any Program benchmark tests without our prior consent.

Export

You agree that U.S. export control laws and other applicable export and import laws govern your use of the Programs, including technical data; additional information can be found on Oracle's Global Trade Compliance web site located at http://www.oracle.com/products/export/index.html. You agree that neither the Programs nor any direct product thereof will be exported, directly, or indirectly, in violation of these laws, or will be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation.

Disclaimer of Warranty and Exclusive Remedies

THE PROGRAMS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. WE FURTHER DISCLAIM ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

IN NO EVENT SHALL WE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. OUR ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND DOLLARS (U.S. $1,000).

No Technical Support

Our technical support organization will not provide technical support, phone support, or updates to you or end users for the Programs licensed under this agreement.

Restricted Rights

If you distribute a license to the United States government, the Programs, including documentation, shall be considered commercial computer software and you will place a legend, in addition to applicable copyright notices, on the documentation, and on the media label, substantially similar to the following:

NOTICE OF RESTRICTED RIGHTS

"Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the

applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065."

End of Agreement

You may terminate this Agreement by destroying all copies of the Programs. We have the right to terminate your right to use the Programs if you fail to comply with any of the terms of this Agreement, in which case you shall destroy all copies of the Programs.

Relationship Between the Parties

The relationship between you and us is that of licensee/licensor. Neither party will represent that it has any authority to assume or create any obligation, express or implied, on behalf of the other party, nor to represent the other party as agent, employee, franchisee, or in any other capacity. Nothing in this Agreement shall be construed to limit either party's right to independently develop or distribute software that is functionally similar to the other party's products, so long as proprietary information of the other party is not included in such software.

Open Source

"Open Source" software - software available without charge for use, modification and distribution - is often licensed under terms that require the user to make the user's modifications to the Open Source software or any software that the user 'combines' with the Open Source software freely available in source code form. If you use Open Source software in conjunction with the Programs, you must ensure that your use does not: (i) create, or purport to create, obligations of us with respect to the Oracle Programs; or (ii) grant, or purport to grant, to any third party any rights to or immunities under our intellectual property or proprietary rights in the Oracle Programs. For example, you may not develop a software program using an Oracle Program and an Open Source program where such use results in a program file(s) that contains code from both the Oracle Program and the Open Source program (including without limitation libraries) if the Open Source program is licensed under a license that requires any "modifications" be made freely available. You also may not combine the Oracle Program with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the Oracle Program or any modifications thereto to become subject to the terms of the GPL.

Entire Agreement

You agree that this Agreement is the complete agreement for the Programs and licenses, and this Agreement supersedes all prior or contemporaneous Agreements or representations. If any term of this Agreement is found to be invalid or unenforceable, the remaining provisions will remain effective.

Last updated: 01/24/08

L20. ParaType Free Font

LICENSING AGREEMENT

for the fonts with Original Name: PT Sans, PT Serif, PT Mono

Version 1.3 - January 20, 2012

GRANT OF LICENSE

```
ParaType Ltd grants you the right to use, copy, modify the fonts and distribute
modified and unmodified copies of the fonts by any means, including placing on Web
servers for free downloading, embedding in documents and Web pages, bundling with
commercial and non commercial products, if it does not conflict with the conditions
listed below:

- You may bundle the fonts with commercial software, but you may not sell the fonts by
themselves. They are free.

- You may distribute the fonts in modified or unmodified versions only together with
this Licensing Agreement and with above copyright notice. You have no right to modify
the text of Licensing Agreement. It can be placed in a separate text file or inserted
into the font file, but it must be easily viewed by users.

- You may not distribute modified version of the font under the Original name or a
combination of Original name with any other words without explicit written permission
from ParaType.

TERMINATION & TERRITORY

This license has no limits on time and territory, but it becomes null and void if any
of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT.
IN NO EVENT SHALL PARATYPE BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY
TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

ParaType Ltd
```

L21. PCRE

```
PCRE2 is a library of functions to support regular expressions whose syntax and
semantics are as close as possible to those of the Perl 5 language.

Release 10 of PCRE2 is distributed under the terms of the "BSD" licence, as specified
below, with one exemption for certain binary redistributions. The documentation for
PCRE2, supplied in the "doc" directory, is distributed under the same terms as the
software itself. The data in the testdata directory is not copyrighted and is in the
public domain.

The basic library functions are written in C and are freestanding. Also included in the
distribution is a just-in-time compiler that can be used to optimize pattern matching.
This is an optional feature that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS

---------------------------

Written by:       Philip Hazel

Email local part: ph10
```

```
Email domain:     cam.ac.uk

University of Cambridge Computing Service,

Cambridge, England.

Copyright (c) 1997-2018 University of Cambridge

All rights reserved.

PCRE2 JUST-IN-TIME COMPILATION SUPPORT

-------------------------------------

Written by:       Zoltan Herczeg

Email local part: hzmester

Email domain:     freemail.hu

Copyright(c) 2010-2018 Zoltan Herczeg

All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER

--------------------------------

Written by:       Zoltan Herczeg

Email local part: hzmester

Email domain:     freemail.hu

Copyright(c) 2009-2018 Zoltan Herczeg

All rights reserved.

THE "BSD" LICENCE

-----------------

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notices, this list of
conditions and the following disclaimer.

 * Redistributions in binary form must reproduce the above copyright notices, this list
of conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution.

  * Neither the name of the University of Cambridge nor the names of any contributors
may be used to endorse or promote products derived from this      software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
```

```
THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

EXEMPTION FOR BINARY LIBRARY-LIKE PACKAGES

------------------------------------------

The second condition in the BSD licence (covering binary redistributions) does not
apply all the way down a chain of software. If binary package A includes PCRE2, it must
respect the condition, but if package B is software that includes package A, the
condition is not imposed on package B unless it uses PCRE2 independently.
```

L22. QR Code Generator

```
Copyright © 2022 Project Nayuki. (MIT License)

https://www.nayuki.io/page/qr-code-generator-library



Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:



The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.



The Software is provided "as is", without warranty of any kind, express or
implied, including but not limited to the warranties of merchantability,
fitness for a particular purpose and noninfringement. In no event shall the
authors or copyright holders be liable for any claim, damages or other
liability, whether in an action of contract, tort or otherwise, arising
from, out of or in connection with the Software or the use or other dealings
in the Software.
```

L23. quirc

```
quirc -- QR-code recognition library

Copyright (C) 2010-2012 Daniel Beer <dlbeer@gmail.com>
```

```
ISC License

==========

Permission to use, copy, modify, and/or distribute this software for
any purpose with or without fee is hereby granted, provided that the
above copyright notice and this permission notice appear in all
copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL
WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE
AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR
PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.
```

L24. Script.aculo.us

```
Copyright © 2005-2008 Thomas Fuchs (http://script.aculo.us, http://mir.aculo.us)

Permission is hereby granted, free of charge, to any person obtaining a copy of this
software and associated documentation files (the “Software”), to deal in the Software
without restriction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be included in all copies
or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

L25. Zlib

```
zlib.h -- interface of the 'zlib' general purpose compression library

  version 1.2.11, January 15th, 2017

  Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

  This software is provided 'as-is', without any express or implied warranty.  In no
event will the authors be held liable for any damages arising from the use of this
software.

  Permission is granted to anyone to use this software for any purpose, including
commercial applications, and to alter it and redistribute it freely, subject to the
following restrictions:

  1. The origin of this software must not be misrepresented; you must not claim that
you wrote the original software. If you use this software in a product, an
acknowledgment in the product documentation would be appreciated but is not required.

  2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.

  3. This notice may not be removed or altered from any source distribution.

  Jean-loup Gailly        Mark Adler

  jloup@gzip.org          madler@alumni.caltech.edu
```

Appendix M. User Hooks

This section describes the following categories of user hooks:

Hook	Description
Administrators	administrator authorization management
Group	group management
Access	access control
Other	other
Newbies	managing new stations
Neighborhood	managing connections with neighboring Dr.Web Servers
Server	Dr.Web Server management
Connections	client connection management
Stations	station management

Hook	Description
Ldap	translating usernames

M1. Administrators

Administrator logged in

Called when administrator successfully authenticated in the Security Control Center.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `address`—network address from which administrator is logged in • `subsys`—Dr.Web Server subsystem (see the adm-subsys.ds file) • `id`—administrator's ID • `authorizer`—authorization module name (database, LDAP, AD) • `language`—language code of administrative account • `date_format`—date format of administrative account	ignored

Procedure text:

```
--[[
Called:
  when Administrator authorize successfully

Database:
  available

Parameters:
  login               Administrator`s login name
  address             Administrator`s network address
  subsys              Server subsystem (see adm-subsys.ds)
  id                  Administrator`s ID
  authorizer          Authorizer name (database, LDAP, AD)
  language            Administrator`s language code
  date_format         Administrator`s date format

Returned value:
  ignored

]]

local args = ... -- args.login, args.address, args.subsys, args.error, args.id,
args.authorizer,
                 -- args.language, args.date_format
```

Administrator logged in using Microsoft Active Directory Service

Called when external administrator successfully authorized using Microsoft Active Directory Service.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `address`—network address administrator used to log in from • is_secure—administrator uses HTTPS connection (`true` | `false`) • `name`—administrator's LDAP name • `DN`—administrator's LDAP distinguished name • `SID`—administrator's Windows security identifier • `GUID`—administrator's GUID • `primary_group`—administrator's primary group name • `primary_group_DN`—administrator's primary group LDAP distinguished name • `primary_group_SID`—administrator's primary group SID • `primary_group_GUID`—administrator's primary group GUID • `groups`—table containing administrator's group names (member of MSAD attribute) • `groups_DN`—table containing administrator's group distinguished names (in the same order as groups) • `groups_SID`—table containing administrator's group SIDs (in the same order as groups) • `groups_GUID`—table containing administrator's group GUIDs (in the same order as groups)	• `nil`—do nothing • `string empty`—do nothing • `not-empty`—set administrator group to this string (group ID)

Procedure text:

```
--[[
Called:
  when the external administrator was authorized successfully using Microsoft Active
Directory Service

Database:
  available

Parameters:
  login              Administrator's login name
  address            Administrator's network address
  is_secure          Is true if administrator uses HTTPS connection
  name               Administrator's LDAP name
  DN                 Administrator's LDAP distinguished name
  SID                Administrator's Windows security identifier
  GUID               Administrator's GUID
  primary_group      Administrator's primary group name
```

```
  primary_group_DN   Administrator's primary group LDAP distinguished name
  primary_group_SID  Administrator's primary group SID
  primary_group_GUID Administrator's primary group GUID
  groups             Table containg Administrator's group names (memberOf MSAD
attribute)
  groups_DN          Table containg Administrator's group distinguished names (in the
same order as groups)
  groups_SID         Table containg Administrator's group SIDs (in the same order as
groups)
  groups_GUID        Table containg Administrator's group GUIDs (in the same order as
groups)

Returned value:
             nil        do nothing
  string     empty      do nothing
             not-empty  set administrator group to this string (group ID)

]]

local args = ... -- args.is_secure, args.login, args.address,
                 -- args.name, args.DN, args.SID, args.GUID,
                 -- args.primary_group, args.primary_group_DN, args.primary_group_SID,
args.primary_group_GUID,
                 -- args.groups, args.groups_DN, args.groups_SID, args.groups_GUID
```

Administrator not logged in

Called when administrator failed to authenticate in the Security Control Center.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `address`—administrator's network address • `subsys`—Dr.Web Server subsystem (see the adm-subsys.ds file) • `error`—error code (see the `auth-error.ds` file)	ignored

Procedure text:

```
--[[
Called:
  when Administrator authorization failed

Database:
  available

Parameters:
  login              Administrator`s login name
  address            Administrator`s network address
  subsys             Server subsystem (see adm-subsys.ds)
  error              Error code (see auth-error.ds)

Returned value:
  ignored

]]

local args = ... -- args.login, args.address, args.subsys, args.error
```

M2. Group

Group created

Called when new group is created.

Database	Parameters	Returned value
available unless hook activated by `drwcs.new_group()` function	• `login`—login name of administrator • `id`—group ID • `name`—group name • `pid`—parent group ID	ignored

Procedure text:

```
--[[
Called:
  when new group created

Database:
  available

Parameters:
  login      administrator`s login name
  id         group ID
  name       group name
  pid        parent group ID

Returned value:
  ignored

]]
```

```
local args = ... -- args.login, args.id, args.name, args.pid
```

Group deleted

Called when group is deleted.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `id`—group ID • `name`—group name	ignored

Procedure text:

```
--[[
Called:
  when group deleted

Database:
  available

Parameters:
  login      administrator`s login name
  id         group ID
  name       group name

Returned value:
  ignored

]]

local args = ... -- args.login, args.id, args.name
```

Group properties changed

Called when group properties are changed.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `id`—group ID • `name`—group name • `descr`—group description • `pid`—parent group ID	ignored

Procedure text:

```
--[[
Called:
  when group properties changed

Database:
  available

Parameters:
  login      administrator`s login name
  id         group ID
  name       group name
  descr      group description
  pid        parent group ID

Returned value:
  ignored

]]

local args = ... -- args.login, args.id, args.name, args.descr, args.pid
```

M3. Access

Access denied

Called when access denied according ACL settings or as a result of the `access_check` procedure.

Database	Parameters	Returned value
available	• `id`—temporary client ID (for newbies/Dr.Web Servers) • `address`—client network address • `station`—NetBIOS name of a client. Undefined for Dr.Web Servers and does not replaced by DNS name • `type`—one of `station`, `installer`, `newbie`, `server`, `proxy` • `description`—station description	ignored

Procedure text:

```
--[[
Called:
  when access denied according ACLs settings or result
  of 'access_check' procedure

Database:
  available

Parameters:
  id             station (temporary for newbie/server) ID
  address        station network address
```

```
  station        station name (undefined for servers)
                 this is NetBIOS station name (not replaced by DNS one)
  type           one of 'station' | 'installer' | 'newbie' | 'server' | 'proxy'
  description    station description

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.type, args.description

-- no return => `nil' value
```

Access check

Called before check access against appropriate ACL (Access Control List).

Database	Parameters	Returned value

available	• `id`—temporary client ID (for newbies/Dr.Web Servers)
	• `address`—client network address
	• `station`—NetBIOS name of a client. Undefined for Dr.Web Servers and does not replaced by DNS name
	• `type`—one of `station`, `installer`, `newbie`, `server`, `proxy`
		• `nil`—check address against configured ACLs
		• `boolean`—do not check against ACLs, for all:
		  ▫ `true`—allow access
		  ▫ `false`—reject access

Procedure text

```
--[[
Called:
  before check access against appropriate ACL

Database:
  available

Parameters:
  id              station ID (temporary for newbie/server)
  address         station network address
  station         station name (undefined for servers)
                  this is NetBIOS station name (not replaced by DNS one)
  type            one of 'station | installer | newbie | server | proxy'

Returned value:
              nil         check address against configured ACLs
  boolean     true        allow access, do not check agains ACLs
              false       reject access, do not check agains ACLs

Procedure from next set will be called if returned nothing.
]]

local args = ... -- args.id, args.address, args.station, args.type

-- no return => `nil' value
```

M4. Other

Automatic update of a license key

Called when a license key is expiring.

Database	Parameters	Returned value

available	• `event`—event type:
	  ▫ `expire`—license key is expiring, automatic update is not available
	  ▫ `diff` —new license key has been downloaded, but the compound of licensed components differs in the current and the new keys. The license key must be replaced manually
	  ▫ `renew`—license key has been automatically updated
	• `old_key`—old license key content
	• `new_key`—new license key content. Available if the event type is `diff` or `renew`
		ignored

Procedure text:

```
--[[
Called:
  when license key expire or have been renewed
```

```
Database:
  available

Parameters:
  event       event type: "expire" - license key expires or have done it
                          "diff"   - received new key, but components differs from
current one
                          "renew"  - current key have been renewed, old one was deleted

  old_key     content of old license key
  new_key     content of renew license key, available at event type "diff" or "renew"

Returned value:
  ignored

]]

local args = ... -- args.event, args.old_key, args.new_key
```

Epidemic detected

Called when a malware outbreak is detected in the network.

Database	Parameters	Returned value
available	• `virus`—most common threat • `total`—total number of detected threats	ignored

Procedure text:

```
--[[
Called:
  when virus epidemic has been detected by the server

Database:
  available

Parameters:
  total            total count of viruses
  virus            most frequently detected virus name

Returned value:
  ignored

]]

local args = ... -- args.total, args.virus
```

Report of Application Control

Called when the Application Control report received from a station.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station network address • `station`—station name • `time`—time of event occurrence (station time) • `sid`—station SID • `user`—user who initiated a process with suspicious activity • `type`—event type • `action`—applied action • `policy_type`—matched policy type • `policy_mask`—matched policy mask • `test_mode`—event occurred in the test mode • `profile_id`—profile UUID used for activity blocking • `profile_name`—profile name used for activity blocking • `rule_id`—rule UUID used for activity blocking (if exists) • `rule_name`—rule name used for activity blocking (if exists) • `process_path`—path to the blocked process • `process_file_sha256`—process file SHA-256 • `process_file_version`—process file version • `process_file_description`—process file description • `process_file_origname`—process file original name • `process_file_prodname`—process file product name • `process_file_prodver`—process file product version • `process_file_company`—process file company name • `process_cert_thumbprint`—process file signing certificate thumbprint (SHA-1) (if exists) • `process_cert_serial`—process file signing certificate serial number (if exists) • `process_cert_issuer`—process file signing certificate issuer (if exists) • `process_cert_subject`—process file signing certificate subject (if exists) • `process_cert_timestamp`—process file signing certificate issuance timestamp (if exists) • `process_cert_not_before`—process file signing certificate start timestamp (if exists)	ignored

Database	Parameters	Returned value
	• process_cert_not_after—process file signing certificate expiration timestamp (if exists) • process_hashdb—bulletin containing the hash of process file • object_path—path to the blocked script or empty • object_file_sha256—script file SHA-256 (if exists) • object_file_version—script file version (if exists) • object_file_description—script file description (if exists) • object_file_origname—script file original name (if exists) • object_file_prodname—script file product name (if exists) • object_file_prodver—script file product version (if exists) • object_file_company—script file company name (if exists) • object_cert_thumbprint—script file signing certificate thumbprint (SHA-1) (if exists) • object_cert_serial—script file signing certificate serial number (if exists) • object_cert_issuer—script file signing certificate issuer (if exists) • object_cert_subject—script file signing certificate subject (if exists) • object_cert_timestamp—script file signing certificate issuance timestamp (if exists) • object_cert_not_before—script file signing certificate start timestamp (if exists) • object_cert_not_after—script file signing certificate expiration timestamp (if exists) • object_hashdb—bulletin containing the hash of script file	

Procedure text:

```
--[[
Called:
  when application control event received from Agent

Database:
  available

Parameters:
```

```
  id                  station ID
  address             station address
  station             station name
  time                station time
  sid                 SID of user initiated activity
  user                name of user initiated activity
  type                event type
  action              applied action
  policy_type         matched policy type
  policy_mask         matched policy mask
  test_mode           event occured in test mode
  profile_id          profile UUID used for activity blocking
  profile_name        profile name used for activity blocking
  rule_id             rule UUID used for activity blocking (if exist)
  rule_name           rule name used for activity blocking (if exist)

  process_path                path to affected process file
  process_file_sha256         process file SHA-256
  process_file_version        process file version
  process_file_description    process file description
  process_file_origname       process file original name
  process_file_prodname       process file product name
  process_file_prodver        process file product version
  process_file_company        process file company name
  process_cert_thumbprint     process file signing certificate thumbprint (SHA-1) (if
exist)
  process_cert_serial         process file signing certificate serial number (if exist)
  process_cert_issuer         process file signing certificate issuer (if exist)
  process_cert_subject        process file signing certificate subject (if exist)
  process_cert_timestamp      process file signing certificate sign issuance timestamp
(if exist)
  process_cert_not_before     process file signing certificate NotBefore timestamp (if
exist)
  process_cert_not_after      process file signing certificate NotAfter timestamp (if
exist)
  process_hashdb              hash database containing process file

  object_path                 path to affected object file (script, etc) or empty
  object_file_sha256          object file SHA-256 (if exist)
  object_file_version         object file version (if exist)
  object_file_description     object file description (if exist)
  object_file_origname        object file original name (if exist)
  object_file_prodname        object file product name (if exist)
  object_file_prodver         object file product version (if exist)
  object_file_company         object file company name (if exist)
  object_cert_thumbprint      object file signing certificate thumbprint (SHA-1) (if
exist)
  object_cert_serial          object file signing certificate serial number (if exist)
  object_cert_issuer          object file signing certificate issuer (if exist)
  object_cert_subject         object file signing certificate subject (if exist)
  object_cert_timestamp       object file signing certificate sign issuance timestamp (if
exist)
  object_cert_not_before      object file signing certificate NotBefore timestamp (if
exist)
  object_cert_not_after       object file signing certificate NotAfter timestamp (if
exist)
  object_hashdb               hash database containing object file

Returned value:
  ignored
```

```
]]

local args = ...
```

Report of Application Control from the neighbor Dr.Web Server

Called when the Application Control report received for a station from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• neighborid—ID of neighbor Dr.Web Server from which the event is received • neighborname—neighbor Dr.Web Server name • originatorid—ID of Dr.Web Server that originated the event • originatorname—name of Dr.Web Server that originated the event • stationid—station ID • stationname—station name • eventid—event ID • event_time—time of event occurrence on a station • sid—station SID • user—user who initiated a process with suspicious activity • type—event type • action—applied action • policy_type—matched policy type • policy_mask—matched policy mask • test_mode—event occurred in the test mode • profile_id—profile UUID used for activity blocking • profile_name—profile name used for activity blocking • rule_id—rule UUID used for activity blocking (if exists) • rule_name—rule name used for activity blocking (if exists) • process_path—path to the blocked process • process_file_sha256—process file SHA-256 • process_file_version—process file version • process_file_description—process file description • process_file_origname—process file original name • process_file_prodname—process file product name	ignored

Database	Parameters	Returned value
	• process_file_prodver—process file product version • process_file_company—process file company name • process_cert_thumbprint—process file signing certificate thumbprint (SHA-1) (if exists) • process_cert_serial—process file signing certificate serial number (if exists) • process_cert_issuer—process file signing certificate issuer (if exists) • process_cert_subject—process file signing certificate subject (if exists) • process_cert_timestamp—process file signing certificate issuance timestamp (if exists) • process_cert_not_before—process file signing certificate start timestamp (if exists) • process_cert_not_after—process file signing certificate expiration timestamp (if exists) • process_hashdb—bulletin containing the hash of process file • object_path—path to the blocked script or empty • object_file_sha256—script file SHA-256 (if exists) • object_file_version—script file version (if exists) • object_file_description—script file description (if exists) • object_file_origname—script file original name (if exists) • object_file_prodname—script file product name (if exists) • object_file_prodver—script file product version (if exists) • object_file_company—script file company name (if exists) • object_cert_thumbprint—script file signing certificate thumbprint (SHA-1) (if exists) • object_cert_serial—script file signing certificate serial number (if exists) • object_cert_issuer—script file signing certificate issuer (if exists) • object_cert_subject—script file signing certificate subject (if exists) • object_cert_timestamp—script file signing certificate issuance timestamp (if exists)	

Database	Parameters	Returned value
	• `object_cert_not_before`—script file signing certificate start timestamp (if exists) • `object_cert_not_after`—script file signing certificate expiration timestamp (if exists) • `object_hashdb`—bulletin containing the hash of script file	

Procedure text:

```
--[[
Called:
  when application control event received from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  event_time          station time
  recv_time           server originator time
  sid                 SID of user initiated activity
  user                name of user initiated activity
  type                event type
  action              applied action
  policy_type         matched policy type
  policy_mask         matched policy mask
  test_mode           event occured in test mode
  profile_id          profile UUID used for activity blocking
  profile_name        profile name used for activity blocking
  rule_id             rule UUID used for activity blocking (if exist)
  rule_name           rule name used for activity blocking (if exist)

  process_path                path to affected process file
  process_file_sha256         process file SHA-256
  process_file_version        process file version
  process_file_description    process file description
  process_file_origname       process file original name
  process_file_prodname       process file product name
  process_file_prodver        process file product version
  process_file_company        process file company name
  process_cert_thumbprint     process file signing certificate thumbprint (SHA-1) (if
exist)
  process_cert_serial         process file signing certificate serial number (if exist)
  process_cert_issuer         process file signing certificate issuer (if exist)
  process_cert_subject        process file signing certificate subject (if exist)
  process_cert_timestamp      process file signing certificate sign issuance timestamp
(if exist)
```

```
  process_cert_not_before    process file signing certificate NotBefore timestamp (if
exist)
  process_cert_not_after     process file signing certificate NotAfter timestamp (if
exist)
  process_hashdb             hash database containing process file

  object_path                path to affected object file (script, etc) or empty
  object_file_sha256         object file SHA-256 (if exist)
  object_file_version        object file version (if exist)
  object_file_description    object file description (if exist)
  object_file_origname       object file original name (if exist)
  object_file_prodname       object file product name (if exist)
  object_file_prodver        object file product version (if exist)
  object_file_company        object file company name (if exist)
  object_cert_thumbprint     object file signing certificate thumbprint (SHA-1) (if
exist)
  object_cert_serial         object file signing certificate serial number (if exist)
  object_cert_issuer         object file signing certificate issuer (if exist)
  object_cert_subject        object file signing certificate subject (if exist)
  object_cert_timestamp      object file signing certificate sign issuance timestamp (if
exist)
  object_cert_not_before     object file signing certificate NotBefore timestamp (if
exist)
  object_cert_not_after      object file signing certificate NotAfter timestamp (if
exist)
  object_hashdb              hash database containing object file

Returned value:
  ignored

]]

local args = ...
```

Dr.Web Proxy Server created

Called when Dr.Web Proxy Server creation is completed.

Database	Parameters	Returned value

available	• `login`—login name of administrator
	• `id`—Dr.Web Proxy Server ID
	• `name`—Dr.Web Proxy Server name
	• `state`—operation completion state:
	  ▫ `0`—created successfully
	  ▫ `1`—operation failed (database error)
	  ▫ `2`—operation timed out (database overloaded)
	  ▫ `4`—Dr.Web Proxy Server already exists
		ignored

Procedure text:

```
--[[
```

```
Called:
  when proxy create completed

Database:
  available

Parameters:
  login         administrator`s login name
  id            proxy ID
  name          proxy name
  state         operation completion state:
                  0  created successfully
                  1  operation failed (database error)
                  2  operation timed out (database overloaded)
                  4  already exists

Returned value:
  ignored

]]

local args = ... -- args.login, args.id, args.name, args.state
```

Dr.Web Proxy Server deleted

Called when Dr.Web Proxy Server deleted.

Database	Parameters	Returned value
availalbe	• `login`—login name of administrator • `id`—Dr.Web Proxy Server ID • `name`—Dr.Web Proxy Server name	ignored

Procedure text:

```
--[[
Called:
  when proxy deleted

Database:
  available

Parameters:
  login     administrator`s login name
  id        proxy id
  name      proxy name

Returned value:
  ignored

]]

local args = ... -- args.login, args.id, args.name
```

M5. Newbies

Newbie registered

Called after granting the access to newbie but before recording information to the database.

Database	Parameters	Returned value
available	• `id` —temporary/permanent station ID • `address`—station network address • `station`—station name • `existing`—approved using existing station	ignored

Procedure text:

```
--[[
Database:
  available

Called:
  when newbie access granted but before information stored in database

Parameters:
  id              station temporary/permanent ID
  address         station network address
  station         station name
  existing        approved using existing station

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station
```

Newbie connects to Dr.Web Server

Called when newbie connects to Dr.Web Server.

Database	Parameters	Returned value
availalbe	• `id`—temporary station ID • `address`—station network address • `station`—station name • `description`—station description (for clients under Windows OS only) • `ldapdn`—station LDAP DN (for clients under Windows OS only)	• `nil`—use default settings as at standard Dr.Web Server operation • `boolean`—Dr.Web Server action:

Database	Parameters	Returned value
	• `sid`—station SID • `mac` —station MAC address	▫ `true`—request manual approval (similarly to `Newbie approval`) ▫ `false`—reject access (similarly to `Newbie closed`) • `string`—primary group on acceptance: ▫ `empty`—accept, set primary group to `Everyone` (similarly to `Newbie open`) ▫ `not-empty`—accept and set as primary the group with ID which corresponds to this string. Attention! existence of this ID will be checked and if it does not exist, the `Everyone` group will be set as primary • `vector`—accept by default, contains the following commands and optional arguments: ▫ `pgroup`—set the primary group, must be followed by group ID ▫ `rate`—set the rate group, must be followed by rate group ID ▫ `id`—set the station ID, must be followed by station ID

Database	Parameters	Returned value
		▫ `approve`— request manual approval instead of automatic accepting ▫ `into`—accept into existing station, must be followed by existing station ID

Procedure text:

```
--[[
Called:
  when newbie connected

Database:
  available

Parameters:
  id              station temporary ID
  address         station network address
  station         station name
  description     station description (Windows client only)
  ldapdn          station LDAP DN (Windows client only)
  sid             station computer SID
  mac             station computer MAC

Returned value:
              nil         default, standard server operation according settings
  boolean     true        request approval (like 'Newbie approval' does)
              false       reject access (like 'Newbie closed' does)
  string      empty       accept, set primary group to 'Everyone'
                            (like 'Newbie open' does)
              not-empty   accept, set primary group to this string (ID) (substring after
space treated as rate group id)
                          Attention! Existence of This ID will be checked and
                          if it does not exist it will be replaced by `Everyone'
  vector                  accept by default, must contain commands and optional arguments:
                          "pgroup"  - set primary group, must be followed by group id
                          "rate"    - set rate group, must be followed by rate group id
                          "id"      - set station id, must be followed by station id
                          "approve" - request approval instead of accepting
                          "into"    - accept into existing station, must be followed by
existing station id


Procedure from next set will be called if returned nothing.
]]

local args = ... -- args.id, args.address, args.station
```

```
-- place my station (named ADMINISTRATOR) into `Everyone' group ignoring newbie policy
and newbie's preference
if string.upper( args.station ) == 'ADMINISTRATOR' then
  return ''
end

-- set new UUID for any station with this id and request for manual approve, useful for
cloned stations
if args.id == '01234567-89ab-cdef-0123-456789abcdef' then
  return { "id", dwcore.get_uuid(), "approve" }
end

-- no return => `nil' value => according server settings
```

Newbie accepted

Called when access to newbie is granted, authorization is successful and station is created in database.

Database	Parameters	Returned value
available	• `id`—temporary station ID • `address`—station network address • `station`—station name • `compsid`—station SID • `compmac`—station MAC address • `description`—station description	ignored

Procedure text:

```
--[[
Called:
  when newbie access granted, authorization is successfull
  and station created in database

Database:
  available

Parameters:
  id              station temporary ID
  address         station network address
  station         station name
  compsid         station UID (SID on Windows)
  compmac         station MAC address
  description     station description

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.compsid, args.compmac,
args.description
```

M6. Neighborhood

Component completed on station of neighbor Dr.Web Server

Called when the `component completed` event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name • `eventid`—event ID • `component`—component number • `pid`—process ID • `infections`—threats detected • `errors`—access errors detected • `exitcode`—component exit code • `time`—end time (station time)	ignored

Procedure text:

```
--[[
Called:
  when "component completed" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  component           component number
  pid                 process ID
  infections          infections found
  errors              access errors detected
  exitcode            component exit code
  time                end time (station time)
```

```
Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname, args.time
                 -- args.component, args.pid, args.infections
                 -- args.errors, args.exitcode
```

Component started on station of neighbor Dr.Web Server

Called when the event `component started` is received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name • `eventid`—event ID • `component`—component number • `pid`—process ID • `engine`—virus-finding engine version • `records`—number of threat records • `user`—user name and group of process owner • `time`—start time (station time)	ignored

Procedure text:

```
--[[
Called:
  when "component started" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
```

```
  originatorid       ID of the event server originator
  originatorname     name of the event server originator
  stationid          station ID
  stationname        station name
  eventid            event ID
  component          component number
  pid                process ID
  engine             virus-finding engine version
  records            virus records number
  user               user name and group (process owner)
  time               start time (station time)

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.component, args.pid, args.engine
                 -- args.records, args.user, args.time
```

Geolocation of neighbor Dr.Web Server or station of neighbor Dr.Web Server changed

Called when `geolocation` event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name • `latitude`—latitude in the DD.DDDDDD format • `longitude`—longitude in the DD.DDDDDD format	ignored

Procedure text:

```
--[[
Called:
  when "geolocation" event received from neighbor server

Database:
  available
```

```
Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  latitude            latitude in DD.DDDDDD format
  longitude           longitude in DD.DDDDDD format

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.latidue,args.longitude
                 -- ...
```

Hardware and software on station of neighbor Dr.Web Server are changed

Called when `environment changed` event received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name • `eventid`—event ID • `group_name`—name of a station primary group • `category`—environment category	ignored

Procedure text:

```
--[[
Called:
  when "environment changed" event recived from neighbor server

Database:
  available
```

```
Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  group_name          station primary group name
  category            environment category

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,args.stationname,
                 -- args.group_name, args.category
```

Security threat detected on a station of neighbor Dr.Web Server

Called when the `virus detected` event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value

available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received
	• `neighborname`—neighbor Dr.Web Server name
	• `originatorid`—ID of Dr.Web Server that originated the event
	• `originatorname`—name of Dr.Web Server that originated the event
	• `stationid`—station ID
	• `stationname`—station name
	• `eventid`—event ID
	• `component`—component number
	• `pid`—process ID
	• `time`—time of event occurrence (station time)
	• `user`—user name and group of process owner
	• `object`—path to the object in the file system
	• `owner`—user name and group of object owner
	• `action`—action code
	• `objecttype`—object type:
	  ▫ `-1`—unknown
	  ▫ `0`—file
		ignored

Database	Parameters	Returned value
	▫ 1—boot sector ▫ 2—memory block or process ▫ 3—malicious activity • infectiontype—threat type (see Dr.Web API) • sha1—SHA-1 hash of detected object	

Procedure text:

```
--[[
Called:
  when "" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  component           component number
  pid                 process ID
  time                event time (station time)
  user                user name and group (process owner)
  object              filesystem object path
  owner               object owner (user name and group)
  action              action code (see Dr.Web API; only errors bit set)
  objecttype          object type
                        -1    unknown
                         0    file
                         1    boot sector
                         2    memory block / process
                         3    virus like activity
  infectiontype       infection type (see Dr.Web API)
  sha1                object SHA-1 hash
  sha256              object SHA-256 hash
  hashdb              hash database containing object

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.component, args.pid, args.time, args.user,
                 -- args.object, args.owner,
                 -- args.action, args.objecttype, args.infectiontype,
                 -- args.sha1, args.sha256, args.hashdb
```

Report of Preventive protection from the neighbor Dr.Web Server

Called when the Preventive protection report received for a station from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name • `eventid`—event ID • `pid`—process ID • `path`—executable path of a process with suspicious activity • `target_path`—path to the protected object to which the access attempt was made • `hips_type`—protected object type (numeric) • `shell_guard_type`—blocking reason of unauthorized code execution (numeric) • `denied`—access was denied (true | false) • `is_user_action`—action was requiested from a user (`true` | `false`) • `event_count`—number of automatically denied events (if the `is_user_action` is `false`) • `event_user`—user who initiated a process with suspicious activity • `action_user`—user who specified the reaction on suspicious activity of a process (if the `is_user_action` is `true`) • `event_time`—time of event occurrense on a station • `recv_time`—report receiving time by neighbor Dr.Web Server • `sha1`—SHA-1 hash of detected object • `sha256`—SHA-256 hash of detected object • `hashdb`—bulletin containing the hash	ignored

Procedure text:

```
--[[
Called:
  when HIPS event received from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  pid                 numeric,process id
  path                process file path
  target_path         affected resource path
  hips_type           numeric, HIPS type
  shell_guard_type    numeric, Shell Guard event type
  denied              boolean, access was denied
  is_user_action      boolean, user was asked
  event_count         event number (for accumulation period - if is_user_action is false)
  event_user          user which initiated the suspicious activity
  action_user         user which allowed or denied the activity (non-empty only if
is_user_action is true)
  event_time          station time
  recv_time           server originator time
  sha1                process file SHA-1 hash
  sha256              process file SHA-256 hash
  hashdb              hash database containing process file

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname, args.originatorid,
args.originatorname,
                 -- args.stationid, args.stationname, args.eventid
                 -- args.pid, args.path, args.target_path, args.hips_type,
args.shell_guard_type,
                 -- args.denied, args.is_user_action, args.event_count, args.event_user,
args.action_user
                 -- args.event_time, args.recv_time, args.sha1, args.sha256, args.hashdb
```

Authorization error on neighbor Dr.Web Server

Called after connection with neighbor Dr.Web Server had been rejected because of authorization error.

Database	Parameters	Returned value
available	• id—Dr.Web Server ID • address—Dr.Web Server address • name—Dr.Web Server name • reason—failure reason	ignored

Procedure text:

```
--[[
Called:
  just after server connection rejected due (authorization) error

Database:
  available

Parameters:
  id          server ID
  address     server address
  name        server name
  reason      failure reason

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.name, args.reason
```

Scan error of a station of neighbor Dr.Web Server

Called when the `scan error` event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• neighborid—ID of neighbor Dr.Web Server from which the event is received • neighborname—neighbor Dr.Web Server name • originatorid—ID of Dr.Web Server that originated the event • originatorname—name of Dr.Web Server that originated the event • stationid—station ID • stationname—station name • eventid—event ID • component—component number • pid—process ID	ignored

Database	Parameters	Returned value
	• `time`—time of event occurrence (station time) • `user`—user name and group of process owner • `object`—path to the object in the file system • `owner`—user name and group of object owner • `action`—action code • `sha1`—SHA-1 hash of detected object • `sha256`—SHA-256 hash of detected object • `hashdb`—bulletin containing the hash	

Procedure text:

```
--[[
Called:
  when "" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  component           component number
  pid                 process ID
  time                event time (station time)
  user                user name and group (process owner)
  object              filesystem object path
  owner               object owner (user name and group)
  action              action code (error bit(s) set)
  sha1                object SHA-1 hash
  sha256              object SHA-256 hash
  hashdb              hash database containing object

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.component, args.pid, args.time, args.user,
                 -- args.object, args.owner, args.action,
                 -- args.sha1, args.sha256, args.hashdb
```

Neighbor Dr.Web Server connected

Called when neighbor Dr.Web Server is connected.

Database	Parameters	Returned value
available	• `id`—Dr.Web Server ID • `address`—Dr.Web Server address • `name`—Dr.Web Server name	ignored

Procedure text:

```
--[[
Called:
  when server connected

Database:
  available

Parameters:
  id          server ID
  address     server address
  name        server name

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.name
```

Station status from neighbor Dr.Web Server

Called when neighbor Dr.Web Server reports station status including states of components, virus databases and some local policies (sending events, receiving updates and tasks).

Database	Parameters	Returned value
available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received • `neighborname`—neighbor Dr.Web Server name • `originatorid`—ID of Dr.Web Server that originated the event • `originatorname`—name of Dr.Web Server that originated the event • `stationid`—station ID • `stationname`—station name	ignored

Database	Parameters	Returned value
	• eventid—event ID • count—number of different status codes • state_0—state value • number_0—number of stations in state_0	

Procedure text:

```
--[[
Called:
  when "" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  count               number of different status code
  state_0             state value
  number_0            number of the stations in 'state_0'

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.count,
                 -- args.state_0, args.number_0
                 -- args.state_1, args.number_1
                 -- ...
```

Station of neighbor Dr.Web Server deleted

Called when station is deleted on neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• neighborid—ID of neighbor Dr.Web Server from which the event is received • neighborname—neighbor Dr.Web Server name	ignored

Database	Parameters	Returned value
	• originatorid—ID of Dr.Web Server that originated the event • originatorname—name of Dr.Web Server that originated the event • stationid—station ID • stationname—station name	

Procedure text:

```
--[[
Called:
  when station was deleted on neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname
                 -- ...
```

Scan statistic of a station of neighbor Dr.Web Server

Called when the scan statistics event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value
available	• neighborid—ID of neighbor Dr.Web Server from which the event is received • neighborname—neighbor Dr.Web Server name • originatorid—ID of Dr.Web Server that originated the event • originatorname—name of Dr.Web Server that originated the event	ignored

Database	Parameters	Returned value
	• `stationid`—station ID • `stationname`—station name • `eventid`—event ID • `component`—component number • `pid`—process ID • `user`—user name and group of process owner • `time`—time of event occurrence (station time) • `size`—summary size of all scanned objects • `elapsedtime`—elapsed time • `scanned`—number of scanned objects • `infected`—number of objects infected by known virus • `modifications`—number of objects infected by virus modification • `suspicious`—number of suspicious objects • `cured`—number of cured files • `deleted`—number of deleted files • `renamed`—number of renamed files • `moved`—number of quarantined files • `locked`—number of locked files (SpIDer Guard only) • `errors`—number of not scanned files due to access error	

Procedure text:

```
--[[
Called:
  when "scan statistics" event recived from neighbor server

Database:
  available

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID
  component           number of component
  pid                 process ID
  user                user name and group (process owner)
  time                event time (station time)
  size                summary size of all scanned objects
  elapsedtime         elapsed time
```

```
  scanned            number of scanned objects
  infected           number of objects infected by known virus
  modifications      number of objects infected by virus modification
  suspicious         number of suspicious objects
  cured              number of cured files
  deleted            number of deleted files
  renamed            number of renamed files
  moved              number of quarantined files
  locked             number of locked files (SpIDer Guard only)
  errors             number of not scanned files (due access error)

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.component, args.pid, args.time, args.user,
                 -- args.scanned, args.infected, args.modifications,
                 -- args.suspicious, args.cured, args.deleted, args.renamed,
                 -- args.moved, args.locked, args.errors, args.size, args.elapsedtime
```

Agent installation from neighbor Dr.Web Server

Called when the `installation` event is received from neighbor Dr.Web Server.

Database	Parameters	Returned value

available	• `neighborid`—ID of neighbor Dr.Web Server from which the event is received
	• `neighborname`—neighbor Dr.Web Server name
	• `originatorid`—ID of Dr.Web Server that originated the event
	• `originatorname`—name of Dr.Web Server that originated the event
	• `stationid`—station ID
	• `stationname`—station name
	• `eventid`—event ID
	• `event`—event type:
	  ▫ `0 stationname`—installation has begun
	  ▫ `1`—installation successfully completed
	  ▫ `2`—rejected
	  ▫ `3`—timed out
	  ▫ `4`—failed
	  ▫ `5`—incomplete
	• `message`—error message (or empty if there is no error)
		ignored

Database	Parameters	Returned value
	• address—station address • begtime—begin time • endtime—end time	

Procedure text:

```
--[[
Called:
  when "installation" event recived from neighbor server

Parameters:
  neighborid          neighbor server ID which the event received from
  neighborname        neighbor server name
  originatorid        ID of the event server originator
  originatorname      name of the event server originator
  stationid           station ID
  stationname         station name
  eventid             event ID5
  event               event type:
                        0   installation begin
                        1   successully completed
                        2   rejected
                        3   timed out
                        4   failed
                        5   incomplete
  message             error message (or empty if there is no error)
  address             station address
  begtime             begin time
  endtime             end time

Returned value:
  ignored

]]

local args = ... -- args.neighborid, args.neighborname,
                 -- args.originatorid, args.originatorname,
                 -- args.eventid, args.stationid,
                 -- args.stationname,
                 -- args.event, args.message, args.address
                 -- args.begtime, args.endtime
```

M7. Server

Dr.Web Server binary file loaded

Called when Dr.Web Server binary file loaded to execute some service functions (Dr.Web Server will not serve clients).

Database	Parameters	Returned value
not available	no	ignored

Procedure text:

```
--[[
Called:
  when server binary file loaded for execute some service function
  (the server will not serve clients)

Database:
  NOT available

Parameters:
  none

Returned value:
  ignored

]]
```

Database verified

Called when database verification completed.

Database	Parameters	Returned value
not available	• `state` —completion state: ▫ `true`—success ▫ `false`—failed	ignored

Procedure text:

```
--[[
Called:
  when database verification completed

Database:
  NOT available
```

```
Parameters:
  state     true   success
                   failed

Returned value:
  ignored

]]

local args = ... -- args.state
```

License limit reached (connection not established)

Called when new client connection cannot be established due to license limitation. After client has been disconnected, the `bad_connection.ds` is called.

Database	Parameters	Returned value
available	• `reason`—reason of connection error: ▫ `connection`—no available license ▫ `database`—cannot create new station in database because there are no available licenses left	ignored

Procedure text:

```
--[[
Called:
  when new client connection cannot be established due license limitation

Database:
  available

Parameters:
  reason     "connection"   no available license
             "database"     cannot create new station in database due
                            no available license

Returned value:
  ignored

]]

local args = ... -- args.reason
```

Some Dr.Web Server functions unloaded

Called when Dr.Web Server completed execution of some service functions (Dr.Web Server did not serve clients).

Database	Parameters	Returned value
not available	no	ignored

Procedure text:

```
--[[
Called:
  when server completed execute some service function
  (the server did not serve clients)

Database:
  NOT available

Parameters:
  none

Returned value:
  ignored

]]
```

Database driver loaded

Called when database driver loading process is completed.

Database	Parameters	Returned value

not available	• `state`—completion state:
	  ▫ `true`—successful load
	  ▫ `false`—load failed
	• `driver`—database driver name
	• `library`—full path to database driver library
	• `message`—error message text when state is `false`
		ignored

Procedure text:

```
--[[
Called:
  when database driver load process completed

Database:
```

```
  NOT available

Parameters:
  state     true     successful load
            false    load failed
  driver             database driver name
  library            full path to database driver library
  message            error message text when state is 'false'

Returned value:
  ignored

]]

local args = ... -- args.state, args.driver, args.library, args.message
```

Dr.Web Server task executed

Called when task executed on Dr.Web Server.

Database	Parameters	Returned value

available	• `id`—Dr.Web Server ID
	• `done`—completion state:
	  ▫ `true`—executed successfully
	  ▫ `false`—execution failed
	• `time`—task completion time
	• `name`—task name
	• `error`—message from execution log
		ignored

Procedure text:

```
--[[
Called:
  when job executed on the server

Database:
  available

Parameters:
  id                server ID
  done      true    executed successfully
            false   execution failed
  time              job completion time
  name              job name
  error             error or other message

Returned value:
  ignored
```

```
]]

local args = ... -- args.id, args.done, args.name, args.time, args.error
```

Protocol module unloaded

Called when protocol module unloaded.

Database	Parameters	Returned value
not available	• `name`—internal protocol name • `path`—path to protocol module file	ignored

Procedure text:

```
--[[
Called:
  when protocol module unloaded

Database:
  NOT available

Parameters:
  name                    protocol name
  path                    path to protocol module file

Returned value:
  ignored

]]

local args = ... -- args.path
```

Protocol loaded

Called when protocol module is loaded.

Database	Parameters	Returned value
unknown	• `path`—path to protocol module file • `name`—internal protocol name • `desc`—protocol module description • `state`—state: ▫ `loaded`—protocol module successfully loaded	ignored

Database	Parameters	Returned value
	▫ `disabled`—protocol module is disabled in the `drwcsd.conf` file • `error`—error message text when state is invalid	

Procedure text:

```
--[[
Called:
  when protocol module loaded

Parameters:
  path                    path to protocol module file
  name                    internal protocol name
  desc                    protocol module description string
  state      "loaded"     protocol module loaded successfully
             "disabled"   protocol module is disabled in drwcsd.conf
             "invalid"    invalid protocol module format
  error                   error message if state is "invalid"

Returned value:
  ignored

]]

local args = ... -- args.state, args.path, args.name
```

Extension unloaded

Called when extension module unloaded.

Database	Parameters	Returned value
not available	• `name`—extension name • `path`—path to extension file	ignored

Procedure text:

```
--[[
Called:
  when plugin module unloaded

Database:
  NOT available

Parameters:
  name                    plugin name
  path                    path to plugin file
```

```
Returned value:
  ignored

]]

local args = ... -- args.name, yargs.path
```

Extension loaded

Called when extension module is loaded.

Database	Parameters	Returned value

not available	• `path`—path to extension file
	• `name`—internal extension name
	• `desc`—extension description
	• `state`—state:
	  ▫ `loaded`—extension successfully loaded
	  ▫ `disabled`—extension usage is disabled in the `drwcsd.conf` file
	  ▫ `invalid`—invalid extension format
	• `error`—error message text when state is invalid
		ignored

Procedure text:

```
--[[
Called:
  when plugin module loaded

Database:
  NOT available

Parameters:
  path                    path to plugin file
  name                    internal plugin name
  desc                    plugin description string
  state       "loaded"    plugin loaded successfully
              "disabled"  plugin is disabled in drwcsd.conf
              "invalid"   invalid plugin format
  error                   error message if state is "invalid"

Returned value:
  ignored

]]
```

```
local args = ... -- args.state, args.path, args.name, args.error
```

Backup

Called after completing backup but before deleting previous backup files.

Database	Parameters	Returned value
available	• `state`—completion state: ▫ `true`—success ▫ `false`—failed	ignored

Procedure text:

```
--[[
Called:
  when backup completed but before deleting previous backup files

Database:
  available

Parameters:
  state  true  successful
         false failed

Returned value:
  ignored

]]

local args = ... -- args.state
```

Dr.Web Server terminating

Called when Dr.Web Server is completing the client servicing.

Database	Parameters	Returned value
not available	no	ignored

Procedure text:

```
--[[
Called:
  when server completed serve clients

Database:
  NOT available
```

```
Parameters:
  none

Returned value:
  ignored

]]
```

Dr.Web Server started and ready

Called when Dr.Web Server started and ready to serve clients.

Database	Parameters	Returned value
not available	no	ignored

Procedure text:

```
--[[
Called:
  when server started and going to serve clients

Database:
  NOT available

Parameters:
  none

Returned value:
  ignored

]]
```

M8. Connections

License limit reached (connection denied)

Called when connection denied according license limitation.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station network address • `station`—NetBIOS name of a station. Does not replaced by DNS name • `type`—the type is `station`	ignored

Procedure text:

```
--[[
Called:
  when connection denied according license limitation

Database:
  available

Parameters:
  id              station ID
  address         station network address
  station         station name
                  this is NetBIOS station name (not replaced by DNS one)
  type            one of 'station'

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.type

-- no return => `nil' value
```

Connection error

Called when new client connection cannot be established. Possible reasons: no available licenses (in this case, the `license_error.ds` is called first), no connection to the database, database error, number of stations that are waiting for authorization is exceeded, Dr.Web Server or DB is overloaded.

Database	Parameters	Returned value

available if reason is `no license` and potentially available if reason is `overload` (it is not recommended to use DB at this time)	• `address`—client address
	• `reason`—reason of connection error:
	  ▫ `no database`—no established database connection
	  ▫ `overload`—database is overloaded
	  ▫ `no license`—no available licenses to accept the connection
		ignored

Procedure text:

```
--[[
Called:
  when new client connection cannot be established

Database:
  available if reason is "no license" and potentialy available if
  reason is "overload" (but it is not recommended to use DB that time)

Parameters:
  address                  client address
  reason   "no database"   no established database connection
           "overload"      database is overloaded
           "no license"    no available license to accept connection

Returned value:
  ignored

]]

local args = ... -- args.address, args.reason
```

The PONG received from client

Called when PONG received from a client.

Database	Parameters	Returned value
available	• `id`—client ID • `address`—client network address • `station`—client name (for Agent, Server, Installer) • `time`—packet round-trip time	ignored

Procedure text:

```
--[[
Called:
  when 'PONG' received from client
```

```
Database:
  available

Parameters:
  id         client ID
  address    network address
  station    station name (for Agent, Server, Installer)
  time       packet round-trip time in milliseconds

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station,
                 -- args.time
```

Client disconnected

Called when client disconnected.

Database	Parameters	Returned value
available	• `id`—client ID • `address`—client network address • `type`—client type: `unknown`, `station`, `console`, `server`, `installer`, `newbie` • `station`—station name (only for Agent) • `bytesin`—bytes received • `bytesout`—bytes sent • `totalbytesin`—compressed bytes received • `totalbytesout`—compressed bytes sent • `reason`—disconnect reason	ignored

Procedure text:

```
--[[
Called:
  when client disconnected

Database:
  available

Parameters:
  id              client ID
  address         network address
  type            client type: "unknown", "station", "proxy",
                               "server", "installer", "newbie"
  station         station name (only for Agent)
  bytesin         bytes received
  bytesout        bytes sent
  totalbytesin    compressed bytes received
  totalbytesout   compressed bytes sent
  reason          disconnect reason

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.type, args.station
                 -- args.bytesin, args.bytesout
                 -- args.totalbytesin, args.totalbytesout
                 -- args.reason
```

M9. Stations

Agent deinstalled

Called when deinstallation of Agent is completed.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `state`—completion state: • `true`—success • `false`—failed • `id`—station ID • `address`—station address • `station`—station name • `message`—empty if state is `true`, otherwise contains error message	ignored

Procedure text:

```
--[[
Called:
  when deinstallation of Agent completed

Database:
  available

Parameters:
  login           login name of administrator
  state           true    success
                  false   failed
  id                      station ID
  address                 station address
  station                 station name
  message                 empty if state is 'true' or contains error message

Returned value:
  ignored

]]

local args = ... -- args.login, args.state, args.id
                 -- args.address, args.station, args.message
```

Component completed at station

Called when the `component completed` event is received from Agent.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `component`—component number • `pid`—process ID • `infections`—threats detected • `errors`—access errors detected • `exitcode`—component exit code	ignored

Procedure text:

```
--[[
Called:
  when "component completed" event received from Agent

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  component   component number
  pid         process ID
  infections  infections found
  errors      access errors detected
  exitcode    component exit code

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.component,
                 -- args.pid, args.exitcode, args.infections, args.errors
```

Task executed

Called when `job executed` event is received from Agent.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address	ignored

Database	Parameters	Returned value
	• `station`—station name • `done`—execution state: • `true`—executed successfully • `false`—execution failed • `time`—task completion time • `name`—task name • `error`—error or status message	

Procedure text:

```
--[[
Called:
  when "job executed" event received from Agent

Database:
  available

Parameters:
  id                station ID
  address           station address
  station           station name
  done      true    executed successfully
            false   execution failed
  time              job completion time
  name              job name
  job               job ID (empty for Agent prior version 11 (protocol 3.1+))
  error             error or other message

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.done,
                 -- args.name, args.job, args.time, args.error
```

Component started at station

Called when the `component started` event is received from Agent.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name	ignored

Database	Parameters	Returned value
	• `component`—component number • `pid`—process ID • `engine`—virus-finding engine version • `records`—number of threat records • `user`—user name and group of process owner • `time`—start time (station time)	

Procedure text:

```
--[[
Called:
  when "component started" event received from Agent

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  component   component number
  pid         process ID
  engine      virus-finding engine version
  records     virus records number
  user        user name and group (process owner)
  time        start time (station time)

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.component,
                 -- args.pid, args.records, args.user, args.time, args.engine
```

Station geolocation changed

Called when station geolocation is changed.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name	ignored

Database	Parameters	Returned value
	• `latitude`—station latitude in the DD.DDDDDD format • `longitude`—station longitude in the DD.DDDDDD format	

Procedure text:

```
--[[
Called:
  when agent geolocation changed

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  latitude    station latitude in DD.DDDDDD format
  longitude   station longitude in DD.DDDDDD format

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.name, args.latitude, args.longitude
```

Station must be rebooted

Called after Dr.Web Server received the `reboot required` message from a station.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station network address • `station`—NetBIOS name of a station. Does not replaced by DNS name • `product`—product ID • `description`—product description • `from_revision`—current revision number • `to_revision`—new revision number	ignored

Database	Parameters	Returned value
	• from_revision_date—current revision date • to_revision_date—new revision date	

Procedure text:

```
--[[
Called:
  after server received 'reboot required' station message.

Database:
  available

Parameters:
  id                  station ID
  address             station network address
  station             station name (this is NetBIOS station name not replaced by DNS
one)
  product             product ID
  description         product description
  from_revision       current revision number
  to_revision         new revision number
  from_revision_date  current revision date
  to_revision_date    new revision date

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.product, args.description,
args.from_revision, args.to_revision, args.from_revision_date, args.to_revision_date
```

Threat to a station security detected

Called when the `virus detected` event is received from Agent.

Database	Parameters	Returned value
available	• id—station ID • address—station address • station—station name • component—component number • pid—process ID • time—time of event occurrence (station time)	ignored

Database	Parameters	Returned value

	• `user`—user name and group of process owner
	• `object`—path to the object in the file system
	• `owner`—user name and group of object owner
	• `virus`—threat name
	• `action`—action code
	• `objecttype`—object type:
	  ▫ -1—unknown
	  ▫ 0—file
	  ▫ 1—boot sector
	  ▫ 2—memory block or process
	  ▫ 3—malicious activity
	• `infectiontype`—threat type (see Dr.Web API)
	• `compsid`—station SID
	• `compmac`—station MAC address
	• `description`—station description
	• `compdn`—station LDAP DN (for clients under Windows OS only)
	• `sha1`—SHA-1 hash of detected object
	• `sha256`—SHA-256 hash of detected object
	• `hashdb`—bulletin containing the hash	

Procedure text:

```
--[[
Called:
  when "virus detected" event received from Agent

Database:
  available

Parameters:
  id                station ID
  address           station address
  station           station name
```

```
  component         component number
  pid               process ID
  time              event time (station time)
  user              user name and group (process owner)
  object            filesystem object path
  owner             object owner (user name and group)
  virus             virus name
  action            action code (see Dr.Web API; only errors bit set)
  objecttype        object type
                      -1    unknown
                       0    file
                       1    boot sector
                       2    memory block / process
                       3    virus like activity
  infectiontype     infection type (see Dr.Web API)
  compsid           computer sid
  compmac           computer MAC
  description       computer description
  compdn            computer LDAP DN
  sha1              object SHA-1 hash
  sha256            object SHA-256 hash
  hashdb            hash database containing object

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.component,
                 -- args.pid, args.time, args.user, args.object, args.owner,
                 -- args.virus, args.action, args.objecttype, args.infectiontype
                 -- args.compsid, args.compmac, args.description, args.compdn
                 -- args.sha1, args.sha256, args.hashdb
```

Report of Preventive protection

Called when the Preventive protection report received from a station.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `time`—time of event occurrence on a station • `pid`—process ID • `path`—executable path of a process with suspicious activity • `target_path`—path to the protected object to which the access attempt was made • `hips_type`—protected object type (numeric)	ignored

Database	Parameters	Returned value
	• shell_guard_type—blocking reason of unauthorized code execution (numeric) • denied—access was denied (true | false) • is_user_action—action was requested from a user (true | false) • event_count—number of automatically denied events (if the is_user_action is false) • event_user—user who initiated a process with suspicious activity • action_user—user who specified the reaction on suspicious activity of a process (if the is_user_action is true) • sha1—SHA-1 hash of detected object • sha256—SHA-256 hash of detected object • hashdb—bulletin containing the hash	

Procedure text:

```
--[[
Called:
  when HIPS event received from Agent

Database:
  available

Parameters:
  id                  station ID
  address             station address
  station             station name
  time                station time
  pid                 numeric,process id
  path                process file path
  target_path         affected resource path
  hips_type           numeric, HIPS type
  shell_guard_type    numeric, Shell Guard event type
  denied              boolean, access was denied
  is_user_action      boolean, user was asked
  event_count         event number (for accumulation period - if is_user_action is false)
```

```
  event_user         user which initiated the suspicious activity
  action_user        user which allowed or denied the activity (non-empty only if
is_user_action is true)
  sha1               process file SHA-1 hash
  sha256             process file SHA-256 hash
  hashdb             hash database containing process file

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.time,
                 -- args.pid, args.path, args.target_path, args.hips_type,
args.shell_guard_type,
                 -- args.denied, args.is_user_action, args.event_count, args.event_user,
args.action_user
                 -- args.sha1, args.sha256, args.hashdb
```

Station authorization failed

Called after Agent connection rejected because of authorization error.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `reason`—failure reason • `type`—one of `station`, `installer`, `proxy` • `compsid`—station SID • `compmac`—station MAC address • `description`—station description	ignored

Procedure text:

```
--[[
Called:
  just after Agent connection rejected due authorization error

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  reason      failure reason
```

```
  type        one of 'station' | 'installer' | 'proxy'
  compsid     station UID (SID on Windows)
  compmac     station MAC address
  description station description

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.reason, args.type,
args.compsid, args.compmac, args.description
```

Station date/time error

Called when invalid station time/date detected.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `now`—server time (in milliseconds) • `time`—station time (in milliseconds) • `valid_delta`—valid time delta (in milliseconds)	ignored

Procedure text:

```
--[[
Called:
  when invalid station time/date detected

Database:
  available

Parameters:
  id              station ID
  address         station address
  station         station name
  now             server time (in milliseconds)
  time            station time (in milliseconds)
  valid_delta     valid time delta (in milliseconds)

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station
```

```
                -- args.now, args.date, args.valid_delta
```

Station update failed

Called after Dr.Web Server received the `update failed` message from a station.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station network address • `station`—NetBIOS name of a station. Does not replaced by DNS name • `product`—product ID • `description`—product description • `from_revision`—current revision number • `to_revision`—new revision number • `from_revision_date`—current revision date • `to_revision_date`—new revision date	ignored

Procedure text:

```
--[[
Called:
  after server received 'update failed' station message.

Database:
  available

Parameters:
  id                  station ID
  address             station network address
  station             station name (this is NetBIOS station name not replaced by DNS
one)
  product             product ID
  description         product description
  from_revision       current revision number
  to_revision         new revision number
  from_revision_date  current revision date
  to_revision_date    new revision date

Returned value:
  ignored

]]
```

```
local args = ... -- args.id, args.address, args.station, args.product, args.description,
args.from_revision, args.to_revision, args.from_revision_date, args.to_revision_date
```

Station scan error

Called when the `scan error` event is received from Agent.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `component`—component number • `pid` —process ID • `time`—time of event occurrence (station time) • `user`—user name and group of process owner • `object`—path to the object in the file system • `owner`—user name and group of object owner • `action`—action code • `compsid`—station SID • `compmac`—station MAC address • `description`—station description • `ldapdn`—station LDAP DN (for clients under Windows OS only) • `sha1`—SHA-1 hash of detected object • `sha256`—SHA-256 hash of detected object • `hashdb`—bulletin containing the hash	ignored

Procedure text:

```
--[[
Called:
  when "scan error" event received from Agent
```

```
Database:
  available

Parameters:
  id                station ID
  address           station address
  station           station name
  component         component number
  pid               process ID
  time              event time (station time)
  user              user name and group (process owner)
  object            filesystem object path
  owner             object owner (user name and group)
  action            action code (error bit(s) set)
  compsid           computer SID
  compmac           computer MAC
  description       computer description
  ldapdn            computer LDAP DN
  sha1              object SHA-1 hash
  sha256            object SHA-256 hash
  hashdb            hash database containing object

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.component,
                 -- args.pid, args.time, args.user, args.object, args.owner,
                 -- args.action, args.compsid, args.compmac, args.description,
args.ldapdn
                 -- args.sha1, args.sha256, args.hashdb
```

List of components received

Called when Agent reports installed components list.

Database	Parameters	Returned value
available	• `id`—station address • `station`—station name • `count`—number of components reported • `component_0`—component name • `time_0`—installation time • `from_0`—installation source (Dr.Web Server address, MSI, etc.) • `path_0`—installation path	ignored

Procedure text:

```
--[[
Called:
  when Agent reported installed components

Database:
  available

Parameters:
  id                  station ID
  address             station address
  station             station name
  count               number of components reported
  component_0         component name
  time_0              installation time
  from_0              installation source (server address, MSI, etc)
  path_0              installation path

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.count
                 -- args.component_0, args.time_0, args.from_0, args.path_0
                 -- args.component_1, args.time_1, args.from_1, args.path_1
                 -- ...
```

Information on virus databases received

Called when Agent sends virus database information.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `station`—station name • `count`—number of virus databases • `name_0`—virus database file name • `md5_0`—virus database file MD5 • `version_0`—virus database version • `issued_0`—virus database issue date and time • `records_0`—number of records in virus database • `type_0`—virus database type	ignored

Procedure text:

```
--[[
Called:
  when Agent sent virus bases information

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  count       number of found virus bases
  name_0      virus base file name
  md5_0       virus base file MD5
  version_0   virus base version
  issued_0    virus base issue date and time
  records_0   number of records
  type_0      virus base type

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.count,
                 -- args.name_0, args.md5_0, args.version_0,
                 -- args.issued_0, args.records_0, args.type_0,
                 -- args.name_1, args.md5_1, args.version_1,
                 -- args.issued_1, args.records_1, args.type_1,
                 -- ...
```

Station status

Called when Agent reports state of components, virus databases and some local policies (sending events, receiving updates and tasks).

Database	Parameters	Returned value

available	• `events`—reporting on events:
	  ▫ `true`—Agent sends information on events
	  ▫ `false`—Agent does not send information on events
	• `jobs`—accepting tasks (scheduled and remote scans):
	  ▫ `true`—Agent accepts tasks
	  ▫ `false`—Agent does not accept tasks
	• `updates`—accepting updates:
		ignored

Database	Parameters	Returned value
	• `true`—Agent accepts updates • `false`—Agent does not accept updates	

Procedure text:

```
--[[
Called:
  when Agent report its local policy

Database:
  available

Parameters:
  events    true     Agent send events
            false    Agent do not send events
  jobs      true     Agent accept jobs (schedule & remote scan)
            false    Agent do not accept jobs
  updates   true     Agent accept updates
            false    Agent do not accept updates


Returned value:
  ignored

]]

local args = ... -- args.events, args.jobs, args.updates
```

Station authorization in progress

Called when station tries to authorize (ID and password already checked, valid and known).

Database	Parameters	Returned value

available	• `id`—station ID
	• `connected`—checking for stations with this ID already connected to Dr.Web Server:
	  ▫ `true`—other station with this ID already connected to Dr.Web Server
	  ▫ `false`—no other stations with this ID connected
	• `current_address`—network address of already connected station (not empty only if `connected` is true)
		`string`—result of a request to connect the station
		
		`nil`—default Dr.Web Server behavior
		
		`deny`—deny authorization for station
		
		`force`—allow authorization even if other station with this ID already connected (disconnect connected station)
		
		`newbie`—reset station to newbie

Database	Parameters	Returned value
	• `current_name`—name of already connected station • `last_address`—network address of a station with this ID at its last connection • `last_time`—last seen time of a station with this ID • `last_server`—Dr.Web Server of a station with this ID at its last connection • `new_name`—name of connecting station • `new_address`—network address of connecting station	

Procedure text:

```
--[[
Called:
  when station tries to authorize (id and password already checked, valid and known)

Database:
  available

Parameters:
  id                      station ID
  connected       true    station with same ID already connected to server
                  false   no any station with same ID connected
  current_address         already connected station network address (not empty only if
'connected' is true)
  current_name            last connected station name
  last_address            last disconnected station network address
  last_time               last disconnected station seen time
  last_server             last connected station server
  new_name                now connecting station name
  new_address             now connecting station network address

Returned value:
          nil             default server behavior
  string  'deny'          deny authorization for station
          'force'         allow authorization even if other station with same ID already
connected (by disconnecting it)
          'newbie'        reset station to newbie

Procedure from next set will be called if returned nothing.
]]

local args = ... -- args.id, args.connected, args.current_address, args.current_name,
args.last_address,
                 -- args.last_time, args.last_server, args.new_name, args.new_address

-- no return => `nil' value
```

Station connected

Called when Agent connected successfully.

Database	Parameters	Returned value
available	• `id` —station ID • `address`—station address • `station`—station name • `os`—station OS • `platform`—station platform • `compsid`—station SID • `compmac`—station MAC address • `description`—station description	ignored

Procedure text:

```
--[[
Called:
  when Agent connected successfully

Database:
  available

Parameters:
  id          station ID
  address     station address
  station     station name
  os          station os
  platform    station platform
  compsid     station UID (Security ID on Windows)
  compmac     station MAC address
  description station description

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.name, args.os, args.platform,
args.compsid, args.compmac, args.description
```

Station created

Called when station creation is completed.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `id`—station ID • `name`—station name • `state`—operation completion state: ▫ `0` —created successfully ▫ `1`—operation failed (database error) ▫ `2`—operation timed out (database overloaded) ▫ `3`—no available license ▫ `4`—station already exists	ignored

Procedure text:

```
--[[
Called:
  when station create completed

Database:
  available

Parameters:
  login         administrator`s login name
  id            station ID
  name          station name
  state         operation completion state:
                  0  created successfully
                  1  operation failed (database error)
                  2  operation timed out (database overloaded)
                  3  no available license
                  4  already exists

Returned value:
  ignored

]]

local args = ... -- args.login, args.id, args.name, args.state
```

Station deleted

Called when station deleted.

Database	Parameters	Returned value
available	• `login`—login name of administrator • `id`—station ID	ignored

Procedure text:

```
--[[
Called:
  when station deleted

Database:
  available

Parameters:
  login     administrator`s login name
  id        station id

Returned value:
  ignored

]]

local args = ... -- args.login, args.id
```

Station scan statistic

Called when the `scan statistics` event is received from Agent.

Database	Parameters	Returned value
available	• `id` —station ID • address—station address • `station`—station name • `component`—component number • `pid` —process ID • `user`—user name and group of process owner • `time`—time of event occurrence (station time) • `size`—summary size of all scanned objects • `elapsedtime`—elapsed time • `scanned`—number of scanned objects	ignored

Database	Parameters	Returned value
	• `infected`—number of objects infected by known virus • `modifications`—number of objects infected by virus modification • `suspicious`—number of suspicious objects • `cured`—number of cured files • `deleted`—number of deleted files • `renamed`—number of renamed files • `moved`—number of quarantined files • `locked`—number of locked files (SpIDer Guard only) • `errors`—number of not scanned files due to access error	

Procedure text:

```
--[[
Called:
  when "scan statistics" event received from Agent

Database:
  available

Parameters:
  id                station ID
  address           station address
  station           station name
  component         number of component
  pid               process ID
  user              user name and group (process owner)
  time              event time (station time)
  size              summary size of all scanned objects
  elapsedtime       elapsed time
  scanned           number of scanned objects
  infected          number of objects infected by known virus
  modifications     number of objects infected by virus modification
  suspicious        number of suspicious objects
  cured             number of cured files
  deleted           number of deleted files
  renamed           number of renamed files
  moved             number of quarantined files
  locked            number of locked files (SpIDer Guard only)
  errors            number of not scanned files (due access error)
```

```
Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station, args.component,
                 -- args.pid, args.time, args.user, args.scanned,
                 -- args.infected, args.modifications, args.suspicious,
                 -- args.cured, args.deleted, args.renamed, args.moved,
                 -- args.locked, args.errors, args.size, args.elapsedtime
```

Agent installation

Called when the `installation` event occurred.

Database	Parameters	Returned value

available	• `id` —installation ID (attention: it is not station ID)
	• `address`—station address
	• `station`—station name
	• `event`—event type:
	  ▫ `0` —installation begin
	  ▫ `1`—successfully completed
	  ▫ `2`—rejected
	  ▫ `3`—timed out
	  ▫ `4` —failed
	  ▫ `5`—incomplete
	• `message`—error message (or empty if there is no error)
	• `sessionid`—installation session ID
		ignored

Procedure text:

```
--[[
Called:
  when "installation" event occured

Database:
  available

Parameters:
  id                installation ID (not station!)
  address           station address
  station           station name
  event             event type:
                      0   installation begin
                      1   successully completed
```

```
                      2   rejected
                      3   timed out
                      4   failed
                      5   incomplete
  message           error message (or empty if there is no error)
  sessionid         installation session ID

Returned value:
  ignored

]]

local args = ... -- args.id, args.address, args.station
                 -- args.event, args.message, args.sessionid
```

Device blocked

Called when device on station has been blocked.

Database	Parameters	Returned value
available	• `id`—station ID • `address`—station address • `name`—station name • `user`—user name • `instance_id`—device instance ID • `friendly_name`—device friendly name • `description`—device description • `guid`—device GUID • `class`—device class (parent group name)	ignored

Procedure text:

```
--[[
Called:
  when device on station blocked

Database:
  available

Parameters:
  id              station ID
  address         station address
  station         station name
  user            user name
  instance_id     device instance id
  friendly_name   device friendly name
  description     device description
  guid            device guid
  class           device group class guid
  blocktime       time when station was blocked
  blockrcvtime    time when server received alert

Returned value:
  ignored

]]

local args = ... -- args.id args.address args.station args.user args.instance_id
                 -- args.friendly_name args.description args.guid args.class
                 -- args.station_time args.args.recv_time
```

M10. Ldap

Translating user names to LDAP DN

Called: when translating user names to LDAP.

Database	Parameters	Returned value

available	• `user`—user name	• `nil`—unknown user
		• `string` —user information:
		  ▫ `empty`—unknown user
		  ▫ `non-empty`—DN of user

Procedure text

```
--[[

Called:
  when AuthLDAP module translates user name to DN

Database:
  available

Parameters:
  user           username

Returned value:
              nil         unknown user
  string      empty       unknown user
              non-empty   DN of user

Procedure from next set will be called if returned nothing.
]]

local args = ... -- args.user

-- example code:
-- if args.user == 'super-admin' then return 'CN=super,DC=example,DC=com' end

-- no return => `nil' value
```

Chapter 3: Frequently Asked Questions

Moving Dr.Web Server to Another Computer (Windows OS)

The procedure for moving Dr.Web Server described below requires that the same major version of Dr.Web Server as on the original computer will be installed on the computer it is moved to.

When moving Dr.Web Server to another computer, pay attention to the settings of the transport protocols and, if necessary, change them in the **Transport** tab of the **Administration** → **Dr.Web Server configuration** section.

If you want to restore Dr.Web Server from a backup copy after its uninstallation or fatal failure, please refer to the Restoring from Dr.Web Server Backup section.

The procedure for starting and stopping Dr.Web Server is described in the **Administrator Manual**, in the following subsections:

- Starting and Stopping Dr.Web Server (read online) for Windows OS,
- Starting and Stopping Dr.Web Server (read online) for Unix-like OSs.

The procedures for moving Dr.Web Server differ depending on the OS of the original computer:

- Moving from a computer running Windows to another computer running Windows.
- Moving from a computer running a Unix-like OS to a computer running Windows.

After completing either of the procedures, check that the new Dr.Web Server is working properly and ensure that stations are connected to it.

To move Dr.Web Server from a computer running Windows to another computer running Windows

1. Install a new Dr.Web Server (empty configuration, with a new database) of the same major version on the target computer (detailed instructions are given in the **Installation Manual**, section Installing Dr.Web Server for Windows OS (read online).

   If you plan to keep the old IP address of your Dr.Web Server, assign a temporary IP address to the new Dr.Web Server so that the stations can communicate with the old Dr.Web Server during the move.

2. In the web interface of the new Dr.Web Server, navigate to the **Administration** → **License Manager** section, add your valid `agent.key` license key, and propagate it to the **Everyone** group.

3. Open the **Repository state** section in the web interface of the new Dr.Web Server and make sure that the repository is updated correctly.

   If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.

4. Open the **Dr.Web Server** section and make sure that the date displayed in this section matches the date of the current Dr.Web Server revision in the **Repository state** section. If the date does not match and there is a message about the availability of updates, click the **View list of versions** button and update Dr.Web Server to the latest version.
5. Stop the new Dr.Web Server using Windows OS service management tools, Dr.Web Security Control Center, or the **Start** → **All programs** → **Dr.Web Server** → **Stop** menu.
6. Replace the contents of the `%programfiles%\DrWeb Server\etc` and `%programfiles%\DrWeb Server\var\extensions` folders on the new Dr.Web Server with the contents of the same folders on the old Dr.Web Server.
7. Replace the `drwcsd-certificate.pem` file in the `%programfiles%\DrWeb Server\webmin\install\windows` subfolder on the new Dr.Web Server with the same file from the old Dr.Web Server.
8. Stop the old Dr.Web Server.
9. Verify the integrity of the database of the old Dr.Web Server using the command `drwcsd modexecdb database-verify`. The full command line for database verification looks as follows:

   ```
   "C:\Program Files\DrWeb Server\bin\drwcsd.exe" -verbosity=trace -log="C:\Program Files\DrWeb Server\var\verifydb.log" modexecdb database-verify
   ```

   If an error message appears in the `verifydb.log` file after running this command, please contact technical support.

10. Replace the database used by the new Dr.Web Server.

    

    Before proceeding with this step, make sure that the old and the new Dr.Web Servers are stopped.

    - If the old Dr.Web Server uses an embedded database: replace the `database.sqlite` database file in the `%programfiles%\DrWeb Server\var` folder on the new Dr.Web Server with the same file from the old Dr.Web Server.
    - If the old Dr.Web Server uses an external database:
      a) Clone the Dr.Web Server database using DBMS tools.
      b) Restore the clone as a new database using DBMS tools.
      c) Open the `drwcsd.conf` configuration file on the new Dr.Web Server and edit the values in the child element of the `database` tag to match the new database parameters (see F1. Dr.Web Server Configuration File).

11. Update the database structure on the new Dr.Web Server using the command:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -verbosity=trace -log="C:
\Program Files\DrWeb Server\var\upgradedb.log" modexecdb database-upgrade
```

12. Start the new Dr.Web Server.
13. Log in to the web interface of the new Dr.Web Server using the same credentials as for the old Dr.Web Server.
14. Open the **Administration** → **Dr.Web Server Task Scheduler** section and select the **Backup sensitive data** task. Click the icon and select the **Action** tab in the task editing window. Make sure that the path to the folder on the old Dr.Web Server is not specified in the **Path** field. Either clear this field and leave it empty (in this case, the default `%programfiles%\DrWeb Server\var\backup` folder will be used to store backup copies) or specify the path to a folder on the new Dr.Web Server. Do the same for the **Backup repository** task, as well as for any other tasks with the **Back up critical Dr.Web Server data** and **Back up repository** actions (if such tasks are present).

    If there are any tasks with the **Run program** action in the schedule, either edit them according to the software configuration of the new computer or remove them from the schedule.
15. Check that the new Dr.Web Server is working properly and ensure that stations are connected to it.
16. Stop the old Dr.Web Server and uninstall it (see the **Installation Manual**, section Removing Dr.Web Server for Windows OS (read online).

To move Dr.Web Server from a computer running a Unix-like OS to a computer running Windows

The instructions contain examples of commands for Linux OS. Note the difference in the paths in FreeBSD OS:

- `/etc/init.d/ → /usr/local/etc/rc.d/`
- `/var/opt/drwcs/ → /var/drwcs/`
- `/opt/drwcs/ → /usr/local/drwcs/`

1. Install a new Dr.Web Server (empty configuration, with a new database) of the same major version on the target computer (detailed instructions are given in the **Installation Manual**, section Installing Dr.Web Server for Windows OS (read online).

    If you plan to keep the old IP address of your Dr.Web Server, assign a temporary IP address to the new Dr.Web Server so that the stations can communicate with the old Dr.Web Server during the move.

2. In the web interface of the new Dr.Web Server, navigate to the **Administration** → **License Manager** section, add your valid `agent.key` license key, and propagate it to the **Everyone** group.

3. Open the **Repository state** section in the web interface of the new Dr.Web Server and make sure that the repository is updated correctly.

   If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.

4. Open the **Dr.Web Server** section and make sure that the date displayed in this section matches the date of the current Dr.Web Server revision in the **Repository state** section. If the date does not match and there is a message about the availability of updates, click the **View list of versions** button and update Dr.Web Server to the latest version.
5. Stop the new Dr.Web Server using Windows OS service management tools, Dr.Web Security Control Center, or the **Start** → **All programs** → **Dr.Web Server** → **Stop** menu.
6. Replace the following files on the new Dr.Web Server with their equivalents from the old Dr.Web Server:
   - `drwcsd-certificate.pem` certificate file in the `%programfiles%\DrWeb Server\webmin\install\windows` folder on the new Dr.Web Server with the corresponding file from the `/opt/drwcs/webmin/install/windows/` folder on the old Dr.Web Server;
   - `%programfiles%\DrWeb Server\var\extensions` (user hooks) folder contents on the new Dr.Web Server with the `/var/opt/drwcs/extensions/` folder contents on the old Dr.Web Server;
   - `drwcsd.conf` configuration file, `drwcsd.pri` private key file, and a copy of the `drwcsd-certificate.pem` certificate in the `%programfiles%\DrWeb Server\etc` folder on the new Dr.Web Server with the corresponding files from the `/var/opt/drwcs/etc/` folder on the old Dr.Web Server.

   

   Check if there are any Unix paths in the `drwcsd.conf` file. If so, fix them manually before the move.

7. Stop the old Dr.Web Server using the command line with the command:

```
# /etc/init.d/drwcsd stop
```

8. Verify the integrity of the database on the old Dr.Web Server using the command:

```
# /etc/init.d/drwcsd modexecdb database-verify
```

   If an error message appears in the `drwcsd.log` file after running this command, please contact technical support.

9. Replace the database used by the new Dr.Web Server.

   

   Before proceeding with this step, make sure that the old and the new Dr.Web Servers are stopped.

- If the old Dr.Web Server uses an embedded database: replace the `database.sqlite` database file in the `%programfiles%\DrWeb Server\var` folder on the new Dr.Web Server with the same file from the folder `/var/opt/drwcs/` on the old Dr.Web Server.
- If the old Dr.Web Server uses an external database:
  a) Clone the Dr.Web Server database using DBMS tools.
  b) Restore the clone as a new database using DBMS tools.
  c) Open the `drwcsd.conf` configuration file on the new Dr.Web Server and edit the values in the child element of the `database` tag to match the new database parameters (see F1. Dr.Web Server Configuration File).

10. Update the database structure on the new Dr.Web Server using the command:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -verbosity=trace -log="C:
\Program Files\DrWeb Server\var\upgradedb.log" modexecdb database-upgrade
```

11. Start the new Dr.Web Server using Windows OS service management tools or the **Start** → **All programs** → **Dr.Web Server** → **Start** menu.
12. Log in to the web interface of the new Dr.Web Server using the same credentials as for the old Dr.Web Server.
13. Open the **Administration** → **Dr.Web Server Task Scheduler** section and select the **Backup sensitive data** task. Click the icon and select the **Action** tab in the task editing window. Make sure that the path to the folder on the old Dr.Web Server is not specified in the **Path** field. Either clear this field and leave it empty (in this case, the default `%programfiles%\DrWeb Server\var\backup` folder will be used to store backup copies) or specify the path to a folder on the new Dr.Web Server. Do the same for the **Backup repository** task, as well as for any other tasks with the **Back up critical Dr.Web Server data** and **Back up repository** actions (if such tasks are present).

    If there are any tasks with the **Run program** action in the schedule, either edit them according to the software configuration of the new computer or remove them from the schedule.
14. Check that the new Dr.Web Server is working properly and ensure that stations are connected to it.
15. Stop the old Dr.Web Server and uninstall it (see the **Installation Manual**, section Removing Dr.Web Server for Unix-like OS (read online).

To check if Dr.Web Server is working properly after the move

1. Log in to the web interface of the new Dr.Web Server. Make sure that all Dr.Web Agents are displayed correctly in the anti-virus network list.
2. Go to the **Administration** → **Repository state** section and make sure that the repository of the new Dr.Web Server is updated without errors. If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.

To ensure that stations are connected to the new Dr.Web Server

To allow those Dr.Web Agents to migrate for which the new Dr.Web Server address is set via the Control Center instead of the Dr.Web Agent settings on the station, keep both Dr.Web Servers operating until the procedure is completed.

It is recommended that you use the Dr.Web Server name in the FQDN format.

- If the new Dr.Web Server has its own IP address:
  a) Specify the address of the new Dr.Web Server according to the procedure described in the Connecting Dr.Web Agent to Another Dr.Web Server section for all Dr.Web Agents serviced by the old Dr.Web Server.

     The new Dr.Web Server IP address must be specified on both Dr.Web Servers in the Dr.Web Agent settings section for those Dr.Web Agents for which the new Dr.Web Server address was set via the Control Center instead of in the Dr.Web Agent settings on the station.
  b) Wait until all Dr.Web Agents are connected to the new Dr.Web Server.
- If you need the new Dr.Web Server to keep the old IP address:
  a) Stop the old Dr.Web Server.
  b) Assign the IP address of the old Dr.Web Server to the new one.
  c) Restart the new Dr.Web Server for the changed network settings to take effect.

Moving Dr.Web Server to Another Computer (Unix-like OS)

The procedure for moving Dr.Web Server described below requires that the same major version of Dr.Web Server as on the original computer will be installed on the computer it is moved to.

When moving Dr.Web Server to another computer, pay attention to the settings of the transport protocols and, if necessary, change them in the **Transport** tab of the **Administration** → **Dr.Web Server configuration** section.

If you want to restore Dr.Web Server from a backup copy after its uninstallation or fatal failure, please refer to the Restoring from Dr.Web Server Backup section.

The procedure for starting and stopping Dr.Web Server is described in the **Administrator Manual**, in the following subsections:

- Starting and Stopping Dr.Web Server (read online) for Windows OS,
- Starting and Stopping Dr.Web Server (read online) for Unix-like OSs.

The procedures for moving Dr.Web Server differ depending on the OS of the original computer:

- Moving from a computer running a Unix-like OS to another computer running a Unix-like OS.
- Moving from a computer running Windows OS to a computer running a Unix-like OS.

After completing either of the procedures, check that the new Dr.Web Server is working properly and ensure that stations are connected to it.

To move Dr.Web Server from a computer running a Unix-like OS to another computer running a Unix-like OS

The instructions contain examples of commands for Linux OS. Note the difference in the paths in FreeBSD OS:

- `/etc/init.d/ → /usr/local/etc/rc.d/`
- `/var/opt/drwcs/ → /var/drwcs/`
- `/opt/drwcs/ → /usr/local/drwcs/`

1. Install a new Dr.Web Server (empty configuration, with a new database) on the target computer according to the instructions in the **Installation Manual**, section Installing Dr.Web Server for Unix-like OS (read online).

If you plan to keep the old IP address of your Dr.Web Server, assign a temporary IP address to the new Dr.Web Server so that the stations can communicate with the old

Dr.Web Server during the move.

2. In the web interface of the new Dr.Web Server, navigate to the **Administration** → **License Manager** section, add your valid `agent.key` license key, and propagate it to the **Everyone** group.
3. Open the **Repository state** section in the web interface of the new Dr.Web Server and make sure that the repository is updated correctly.

   If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.
4. Open the **Dr.Web Server** section and make sure that the date displayed in this section matches the date of the current Dr.Web Server revision in the **Repository state** section. If the date does not match and there is a message about the availability of updates, click the **View list of versions** button and update Dr.Web Server to the latest version.
5. Stop the new Dr.Web Server via the web interface or via the command line using the following command:

   ```
   # /etc/init.d/drwcsd stop
   ```
6. To keep your user hooks, replace the contents of the `/var/opt/drwcs/extensions/` folder on the new Dr.Web Server with the contents of the same folder on the old Dr.Web Server.
7. Remove the `drwcsd-certificate.pem` certificate file from the `/opt/drwcs/webmin/install/windows/` folder on the new Dr.Web Server:

   ```
   # rm /opt/drwcs/webmin/install/windows/drwcsd-certificate.pem
   ```
8. Replace the `drwcsd.conf` configuration file, `drwcsd.pri` private key file, and a copy of the `drwcsd-certificate.pem` certificate in the `/var/opt/drwcs/etc/` folder on the new Dr.Web Server with the corresponding files from the old Dr.Web Server.
9. Stop the old Dr.Web Server via the web interface or via the command line using the following command:

   ```
   # /etc/init.d/drwcsd stop
   ```
10. Verify the integrity of the database on the old Dr.Web Server using the command:

    ```
    # /etc/init.d/drwcsd modexecdb database-verify
    ```

    If an error message appears in the `drwcsd.log` file after running this command, please contact technical support.
11. Replace the database used by the new Dr.Web Server.

Before proceeding with this step, make sure that the old and the new Dr.Web Servers are stopped.

- If the old Dr.Web Server uses an embedded database: replace the `database.sqlite` database file in the `/var/opt/drwcs/` folder on the new Dr.Web Server with the same file from the old Dr.Web Server.
- If the old Dr.Web Server uses an external database:
  a) Clone the Dr.Web Server database using DBMS tools.
  b) Restore the clone as a new database using DBMS tools.
  c) Open the `drwcsd.conf` configuration file on the new Dr.Web Server and edit the values in the child element of the `database` tag to match the new database parameters (see F1. Dr.Web Server Configuration File).

12. Update the database structure on the new Dr.Web Server using the command:

```
# /etc/init.d/drwcsd modexecdb database-upgrade
```

13. Start the old Dr.Web Server so that it could keep servicing its clients via the command line using the command:

```
# /etc/init.d/drwcsd start
```

14. Make the `drwcs` user the owner of the `/var/opt/drwcs/` database folder and the `/var/opt/drwcs/etc/drwcsd.pri`, `/var/opt/drwcs/etc/drwcsd.conf`, and `/var/opt/drwcs/etc/drwcsd-certificate.pem` files on the new Dr.Web Server:

```
# chown -R drwcs:drwcs /var/opt/drwcs

# chown drwcs:drwcs /var/opt/drwcs/etc/drwcsd*
```

15. Copy the `drwcsd-certificate.pem` certificate to the `/opt/drwcs/webmin/install/windows/` folder.

```
# cp /var/opt/drwcs/etc/drwcsd-
certificate.pem /opt/drwcs/webmin/install/windows
```

16. Start the new Dr.Web Server using the command:

```
# /etc/init.d/drwcsd start
```

17. Log in to the web interface of the new Dr.Web Server using the same credentials as for the old Dr.Web Server.
18. Open the **Administration** → **Dr.Web Server Task Scheduler** section and select the **Backup sensitive data** task. Click the icon and select the **Action** tab in the task editing window. Make sure that the path to the folder on the old Dr.Web Server is not specified in the **Path** field. Either clear this field and leave it empty (in this case, the default `/var/opt/drwcs/backup` folder will be used to store backup copies) or specify

the path to a folder on the new Dr.Web Server. Do the same for the **Backup repository** task, as well as for any other tasks with the **Back up critical Dr.Web Server data** and **Back up repository** actions (if such tasks are present).

If there are any tasks with the **Run program** action in the schedule, either edit them according to the software configuration of the new computer or remove them from the schedule.

19. Check that the new Dr.Web Server is working properly and ensure that stations are connected to it.
20. Stop the old Dr.Web Server and uninstall it (see the **Installation Manual**, section Removing Dr.Web Server for Unix-like OS (read online ↗).

To move Dr.Web Server from a computer running Windows OS to a computer running a Unix-like OS

The instructions contain examples of commands for Linux OS. Note the difference in the paths in FreeBSD OS:

- `/etc/init.d/ → /usr/local/etc/rc.d/`
- `/var/opt/drwcs/ → /var/drwcs/`
- `/opt/drwcs/ → /usr/local/drwcs/`

1. Install a new Dr.Web Server (empty configuration, with a new database) of the same major version on the target computer according to the instructions in the **Installation Manual**, section Installing Dr.Web Server for Unix-like OS (read online ↗).

If you plan to keep the old IP address of your Dr.Web Server, assign a temporary IP address to the new Dr.Web Server so that the stations can communicate with the old Dr.Web Server during the move.

2. In the web interface of the new Dr.Web Server, navigate to the **Administration** → **License Manager** section, add your valid `agent.key` license key, and propagate it to the **Everyone** group.
3. Open the **Repository state** section in the web interface of the new Dr.Web Server and make sure that the repository is updated correctly.

If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.

4. Open the **Dr.Web Server** section and make sure that the date displayed in this section matches the date of the current Dr.Web Server revision in the **Repository state** section. If the date does not match and there is a message about the availability of updates, click the **View list of versions** button and update Dr.Web Server to the latest version.

5. Stop the new Dr.Web Server via the web interface or via the command line using the following command:

```
# /etc/init.d/drwcsd stop
```

6. Delete the following files on the new Dr.Web Server:

- `drwcsd-certificate.pem` certificate file:

```
# rm /opt/drwcs/webmin/install/windows/drwcsd-certificate.pem
```

- `drwcsd.conf` configuration file, `drwcsd.pri` private key file, and a copy of the `drwcsd-certificate.pem` certificate:

```
#
rm /var/opt/drwcs/etc/drwcsd.conf /var/opt/drwcs/etc/drwcsd.pri /var/opt/d
rwcs/etc/drwcsd-certificate.pem
```

- user hooks from the `/var/opt/drwcs/extensions/` folder.

7. Copy the following files from the old Dr.Web Server to the new one:

- the `%programfiles%\DrWeb Server\var\extensions` folder contents (user hooks) of the old Dr.Web Server to the `/var/opt/drwcs/extensions/` folder of the new Dr.Web Server;
- the `drwcsd.conf` configuration file, `drwcsd.pri` private key, and `drwcsd-certificate.pem` certificate file from the `%programfiles%\DrWeb Server\etc` folder of the old Dr.Web to the `/var/opt/drwcs/etc/` folder of the new Dr.Web Server.

Check if there are any Windows paths in the `drwcsd.conf` file. If so, fix them manually before moving the file.

8. Stop the old Dr.Web Server via the web interface or the **Start** → **All programs** → **Dr.Web Server**→ **Stop** menu.
9. Verify the integrity of the database of the old Dr.Web Server using the command `drwcsd modexecdb database-verify`. The full command line for database verification looks as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -verbosity=trace -log="C:
\Program Files\DrWeb Server\var\verifydb.log" modexecdb database-verify
```

If an error message appears in the `verifydb.log` file after running this command, please contact technical support.

10. Replace the database used by the new Dr.Web Server.

Before proceeding with this step, make sure that the old and the new Dr.Web Servers are stopped.

- If the old Dr.Web Server uses an embedded database: replace the `database.sqlite` database file in the `/var/opt/drwcs/` folder on the new Dr.Web Server with the same file from the folder `%programfiles%\DrWeb Server\var` on the old Dr.Web Server.
- If the old Dr.Web Server uses an external database:
  a) Clone the Dr.Web Server database using DBMS tools.
  b) Restore the clone as a new database using DBMS tools.
  c) Open the `drwcsd.conf` configuration file on the new Dr.Web Server and edit the values in the child element of the `database` tag to match the new database parameters (see F1. Dr.Web Server Configuration File).

11. Update the database structure on the new Dr.Web Server using the command:

```
# /etc/init.d/drwcsd modexecdb database-upgrade
```

12. Start the old Dr.Web Server so that it could keep servicing its clients using Windows OS service management tools or the **Start → All programs → Dr.Web Server → Start** menu.
13. Make the `drwcs` user the owner of the `/var/opt/drwcs/` database folder and the `/var/opt/drwcs/etc/drwcsd.pri`, `/var/opt/drwcs/etc/drwcsd.conf`, and `/var/opt/drwcs/etc/drwcsd-certificate.pem` files on the new Dr.Web Server:

```
# chown -R drwcs:drwcs /var/opt/drwcs

# chown drwcs:drwcs /var/opt/drwcs/etc/drwcsd*
```

14. Copy the `drwcsd-certificate.pem` certificate to the `/opt/drwcs/webmin/install/windows` folder.

```
# cp /var/opt/drwcs/etc/drwcsd-
certificate.pem /opt/drwcs/webmin/install/windows
```

15. Start the new Dr.Web Server using the command:

```
# /etc/init.d/drwcsd start
```

16. Log in to the web interface of the new Dr.Web Server using the same credentials as for the old Dr.Web Server.
17. Open the **Administration → Dr.Web Server Task Scheduler** section and select the **Backup sensitive data** task. Click the icon and select the **Action** tab in the task editing window. Make sure that the path to the folder on the old Dr.Web Server is not specified in the **Path** field. Either clear this field and leave it empty (in this case, the default `/var/opt/drwcs/backup` folder will be used to store backup copies) or specify the path to a folder on the new Dr.Web Server. Do the same for the **Backup repository** task, as well as for any other tasks with the **Back up critical Dr.Web Server data** and **Back up repository** actions (if such tasks are present).

If there are any tasks with the **Run program** action in the schedule, either edit them according to the software configuration of the new computer or remove them from the schedule.

18. Check that the new Dr.Web Server is working properly and ensure that stations are connected to it.
19. Stop the old Dr.Web Server and uninstall it (see the **Installation Manual**, section Removing Dr.Web Server for Windows OS (read online).

To check if Dr.Web Server is working properly after the move

1. Log in to the web interface of the new Dr.Web Server. Make sure that all Dr.Web Agents are displayed correctly in the anti-virus network list.
2. Go to the **Administration** → **Repository state** section and make sure that the repository of the new Dr.Web Server is updated without errors. If there are any error messages in the product list table in the **State** column, please contact technical support. Attach the `drwcsd.log` file to the request. Do not take any further action until you receive feedback on the request.

To ensure that stations are connected to the new Dr.Web Server

To allow those Dr.Web Agents to migrate for which the new Dr.Web Server address is set via the Control Center instead of the Dr.Web Agent settings on the station, keep both Dr.Web Servers operating until the procedure is completed.

It is recommended that you use the Dr.Web Server name in the FQDN format.

- If the new Dr.Web Server has its own IP address:
  a) Specify the address of the new Dr.Web Server according to the procedure described in the Connecting Dr.Web Agent to Another Dr.Web Server section for all Dr.Web Agents serviced by the old Dr.Web Server.

     The new Dr.Web Server IP address must be specified on both Dr.Web Servers in the Dr.Web Agent settings section for those Dr.Web Agents for which the new Dr.Web Server address was set via the Control Center instead of in the Dr.Web Agent settings on the station.

  b) Wait until all Dr.Web Agents are connected to the new Dr.Web Server.
- If you need the new Dr.Web Server to keep the old IP address:
  a) Stop the old Dr.Web Server.
  b) Assign the IP address of the old Dr.Web Server to the new one.
  c) Restart the new Dr.Web Server for the changed network settings to take effect.

Connecting Dr.Web Agent to Another Dr.Web Server

Connecting a Dr.Web Agent to another Dr.Web Server may be necessary if agents are connected to Dr.Web Server whose address has changed or if Dr.Web Server was reinstalled without using the certificate of the previous server.

You can connect Dr.Web Agent to another Dr.Web Server in one of the following ways:

1. Via the Control Center.

   Remote management without direct access to the station is possible if the station is still connected to the previous Dr.Web Server. You need the access to the Control Center both of the previous and the new servers.

2. Directly on the station.

   To perform the actions directly on the station, you must have administrative privileges on the station and, if you want to edit configuration via the agent UI, permissions to edit the Dr.Web Agent properties, which are set on Dr.Web Server (section **Anti-virus Network** → **Configuration** → **Permissions** → **General** tab → **Change Dr.Web Agent configuration**). If you do not have these permissions, you can reconnect to another Dr.Web Server locally on the station only after removing the installed Dr.Web Agent via Control Center and installing the new Dr.Web Agent with the new Dr.Web Server settings.

You can change some of connection settings from the command line as a station administrator:

- Windows OS:

```
"%ProgramFiles%\DrWeb\es-service.exe" --esserver=<server_address> --addcert=<certificate_path>
```

- UNIX OS:

```
drweb-ctl esdisconnect && drweb-ctl esconnect <server_address> --Certificate <certificate_path>
```

<*server_address*>—if the server address has changed.

<*certificate_path*>—if the server was reinstalled without using the certificate from the previous server.

The server certificate is available in the Control Center, under **Administration** → **Encryption keys**.

Reconnecting Dr.Web Agent to another Dr.Web Server via the Control Center

1. On the new Dr.Web Server, allow the stations with incorrect authorization parameters to request new authorization parameters as newbies. To do this, in the Control Center, select **Administration** from the main menu → **Dr.Web Server configuration** from the control menu → **General** tab:

   a) Set the **Reset unauthorized to newbie** flag if it is cleared.

b) If the **Always deny access** option is selected in the **Newbie registration** drop-down list, change it to the **Approve access manually** or **Allow access automatically**.

c) To apply these settings, click **Save** and restart Dr.Web Server.

If your company policy does not allow you to change the settings from step 1, then you need to set the parameters of the station authorization directly on the station, in accordance with the account you created in advance in the Control Center.

2. Set the parameters of the new Dr.Web Server on the old Dr.Web Server to which Dr.Web Agent is connected. To do this, select the **Anti-Virus Network** item in the main menu of the Control Center → select the required station (or a group thereof to reconnect all the stations of this group) in the hierarchical list of the network → select the **Connection settings** item in the control menu:

a) If the new Dr.Web Server certificate is different from the previous Dr.Web Server certificate, set the path to the new Dr.Web Server certificate in the **Certificate** field.

b) Enter the new Dr.Web Server address in the **Server** field.

It is recommended that you use the name of Dr.Web Server in the FQDN format as the Dr.Web Server address.

c) Click **Save**.

Reconnecting Dr.Web Agent to another Dr.Web Server directly on the station

1. Set the new Dr.Web Server parameters in the Dr.Web Agent settings. To do this, in the context menu of the Dr.Web Agent icon, select **Security Center** → click the lock (if it is not already unlocked) to get access to the advanced settings → click the button to access the settings → navigate to **Server** menu → **Connection parameters** → click the **Change settings** button:

a) If the new Dr.Web Server certificate is different from the previous Dr.Web Server certificate, set the path to the new Dr.Web Server certificate using the **List of certificates** button.

b) Set the appropriate parameters of the new Dr.Web Server using the **Add** button.

2. Make the station a newbie (reset the authorization parameters on Dr.Web Server). To do this, in the connection settings section from step 1, click the following buttons: **Station connection parameters** → **Reset the parameters and connect as a newbie** → **Reset the parameters**.

If you already know the ID and the password for connecting the new Dr.Web Server, you can enter them in the **Station ID** and the **Password** fields. In this case, it is not necessary to make the station a newbie.

Dr.Web Server Load and Recommended Configuration Parameters

When using an external database in an anti-virus network of significant size, you may need to adjust the values of the following parameters to ensure proper functioning of Dr.Web Server:

- **Administration** → **Dr.Web Server configuration** → **Database** → **Number of connections** (`database connections=""` parameter in the Dr.Web Server configuration file)—maximum allowed number of Dr.Web Server connections with the database.
- **Administration** → **Dr.Web Server configuration** → **General** → **Number of parallel requests from clients** (`threads count=""` parameter in the Dr.Web Server configuration file)—number of threads for processing incoming data from clients: Dr.Web Agents, Agent installers, neighboring Dr.Web Servers, Dr.Web Proxy Servers.

These parameters affect the performance of Dr.Web Server. Do not change them without need. To determine whether the values need to be adjusted, consult the table of relationships between the size of the anti-virus network, hardware capabilities of the Dr.Web Server and the optimal parameter values:

Number of client connections to Dr.Web Server	Number of CPU cores of Dr.Web Server	Number of connections to the database ('database connections')	Number of parallel requests from clients ('threads count')
up to 500	2	2 (default value)	5 (default value)
500–1500	3–5	3–5	6–10
over 1500	5–8	5–8	10–14

The **Number of parallel requests from clients** (`threads count`) parameter value must not be higher than twice the number of CPU cores on the computer performing the functions of Dr.Web Server.

Do not change the default values when using the embedded database.

When working in an anti-virus network with a large number of connections, it is recommended that you consult with the Doctor Web technical support team before changing the values of these parameters.

Increasing Disk Space for Dr.Web Server

The `/var/opt/drwcs/` folder in Linux OS or `/var/drwcs/` folder in FreeBSD OS (collectively referred to as the **var** folder further in this section) stores backup copies of the database, the main configuration files of the anti-virus network components, the installation file repository, and other files necessary for Dr.Web Server. The size of this folder may increase significantly in the course of Dr.Web Server operation. If free space is limited on the Dr.Web Server installation disk, it is recommended that you mount this folder on a separate disk. This can be done either before or after installing Dr.Web Server.

The instructions contain examples of commands for Linux OS. Note the difference in the paths in FreeBSD OS:

- `/var/opt/drwcs/ → /var/drwcs/`
- `/etc/init.d/ → /usr/local/etc/rc.d/`

When initially installing Dr.Web Server

To prevent potential issues with lack of disk space, mount a new disk device as the **var** folder before installing Dr.Web Server.

To mount a new disk as the var folder in Linux OS

1. Create the **var** folder:

   ```
   # mkdir /var/opt/drwcs/
   ```

2. Mount a new disk device (for example, `/dev/sdXY`) to the **var** folder:

   ```
   # mount /dev/sdXY /var/opt/drwcs/
   ```

   where `/var/opt/drwcs/` is the mount point of the new disk device.

3. If necessary, make the appropriate changes to the `/etc/fstab` file.

After following these steps, install Dr.Web Server according to the instructions from the **Installation Manual**, the Installing Dr.Web Server for Unix-like OS (read online) section.

After installing Dr.Web Server

To avoid running out of free space on the Dr.Web Server installation disk, you can move the **var** folder to a separate disk.

To move the var folder to a separate disk in Linux OS

1. Stop Dr.Web Server using the command:

```
# /etc/init.d/drwcsd stop
```

2. Create a new folder to move the contents of the **var** folder to, for example:

```
# mkdir /mnt/var/opt/drwcs/
```

3. Mount a new disk device (for example, `/dev/sdXY`) to the folder you created:

```
# mount /dev/sdXY /mnt/var/opt/drwcs/
```

where `/mnt/var/opt/drwcs/` is the mount point of the new disk device.

4. Copy the contents of the **var** folder to the mounted disk:

```
# cp -apx /var/opt/drwcs/* /mnt/var/opt/drwcs/
```

5. Remove the contents of the original **var** folder:

```
# rm -R /var/opt/drwcs/*
```

6. Unmount the disk mounted to the folder you created:

```
# umount /dev/sdXY
```

7. Mount the disk to the **var** folder:

```
# mount /dev/sdXY /var/opt/drwcs/
```

8. Set the `drwcs` user of the `drwcs` group as the owner of the **var** folder:

```
# chown -R drwcs:drwcs /var/opt/drwcs
```

9. Start Dr.Web Server:

```
# /etc/init.d/drwcsd start
```

10. If necessary, make the appropriate changes to the `/etc/fstab` file.

Changing the Database Type

For Windows OS

The procedure for starting and stopping Dr.Web Server is described in the **Administrator Manual**, section Starting and Stopping Dr.Web Server (read online).

1. Stop Dr.Web Server.
2. Run `drwcsd.exe` with the `modexecdb database-export` switch to export the content of the database to a file. The complete command (for Windows) looks as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb Server\var\exportdb.log"
modexecdb database-export D:\esbase.es
```

It is assumed that Dr.Web Server is installed in the `C:\Program Files\DrWeb Server` folder and the database is exported to the `esbase.es` file , located in the root directory of disc `D`.

If the path to a file (or a file name) contains spaces or national characters, the path should be put in quotation marks:

```
"D:\<long name>\esbase.es"
```

3. Start Dr.Web Server, connect Dr.Web Security Control Center to Dr.Web Server and configure Dr.Web Server to use a different DB. Cancel the prompt to restart Dr.Web Server.
4. Stop Dr.Web Server.
5. Move the database file to a temporary directory until you are sure that changing the database type was successful.
6. Run `drwcsd.exe` with the `modexecdb database-init` switch to initialize a new database. The command line will look as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=all -log="C:\Program Files\DrWeb Server\var\initdb.log"
modexecdb database-init
```

It is assumed that Dr.Web Server is installed in the `C:\Program Files\DrWeb Server` folder.

7. Run `drwcsd.exe` with the `modexecdb database-import` switch to import the database from the file. The command line will look as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb Server\var\importdb.log"
modexecdb database-import D:\esbase.es
```

8. Start Dr.Web Server.

For UNIX OS

1. Stop Dr.Web Server using the script
   - for **Linux** OS:

```
/etc/init.d/drwcsd stop
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd stop
```

or using Dr.Web Security Control Center.

2. Start Dr.Web Server with the `modexecdb database-export` switch to export the database to a file. The command line initiated from the Dr.Web Server installation folder will look as follows:
   - for **Linux** OS:

```
/etc/init.d/drwcsd modexecdb database-export /var/esbase.es
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-export /var/drwcs/esbase.es
```

   It is assumed that the database is exported to `esbase.es`, which is located in the specified folder.
3. Start Dr.Web Server using the script
   - for **Linux** OS:

```
/etc/init.d/drwcsd start
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd start
```

   connect Dr.Web Security Control Center to Dr.Web Server and configure Dr.Web Server to use a different database through Dr.Web Security Control Center menu: **Administration** → **Dr.Web Server configuration** → **Database** tab.

   

   You can also reconfigure Dr.Web Server to use a different database by editing the Dr.Web Server configuration file `drwcsd.conf` directly. To do this, you should comment out/delete the entry about the current database and enter the new database (for more details see F1. Dr.Web Server Configuration File).

   You will be prompted to restart Dr.Web Server. Cancel the restart prompt.
4. Stop Dr.Web Server (see step 1).
5. Move the database file to a temporary directory until you are sure that changing the database type was successful.
6. Run `drwcsd` with the `modexecdb database-init` switch to initialize a new database. The command will look as follows:
   - for **Linux** OS:

```
/etc/init.d/drwcsd modexecdb database-init
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-init
```

7. Run `drwcsd` using the `modexecdb database-import` switch to import the database from a file. The command will look as follows:
   - for **Linux** OS:

```
/etc/init.d/drwcsd modexecdb database-import /var/esbase.es
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-import /var/esbase.es
```

8. Start Dr.Web Server (see step 3).

If you want to change the Dr.Web Server start-up parameters (for example, specify the Dr.Web Server installation folder, change the log level, etc.), you will have to edit the start-up script:

- for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd
```

- for **Linux** OS:

```
/etc/init.d/drwcsd
```

Restoring the Database

During its operation, Dr.Web Server regularly saves backup copies of critical Dr.Web Server data: repository settings, configuration files, encryption keys, certificates, database backup, and hooks.

The backup files are stored in the following folders:

- for **Windows** OS: <*installation_drive*>`:\DrWeb Backup`
- for **Linux** OS: `/var/opt/drwcs/backup`
- for **FreeBSD** OS: `/var/drwcs/backup`

To perform the back-up, a daily task is included in the Dr.Web Server schedule. If such a task is missing in the schedule, it is recommended to create it.

All files in the backup except the database contents, are ready to use. The backup copy of the database is stored in the `.gz` format compatible with `gzip` and other archivers. The contents of the database can be imported from the backup copy to another Dr.Web Server database using the `modexecdb database-import` command. Thus the data can be restored.

To restore the database you can also use a backup created manually by the administrator via the Control Center in the **Administration** → **Database management** → **Export** (for the **Export entire database** mode only).

If for some reason the DB was corrupted during the Dr.Web Server upgrade from previous versions to version 13, please proceed as follows:

1. Remove Dr.Web Server version 13. Backup copies of files, used by Dr.Web Server, will be saved automatically.
2. Install the version of Dr.Web Server that was installed before the upgrade and that was used to create a backup copy.

   According to the general upgrade procedure, you should use all saved Dr.Web Server files except the DB file.

   Create a new DB during the Dr.Web Server installation.
3. Restore the DB from the backup according to the general procedure (see below).
4. Disable Dr.Web Agent, Dr.Web Server and the Network Installer protocols in the Dr.Web Server settings. To do this, select the **Administration** item in the main menu and click **Dr.Web Server configuration** in the control menu, go to the **Modules** tab and clear corresponding flags.
5. Upgrade Dr.Web Server to version 13 according to general procedure (see **Administrator Manual**, Chapter 11: Updating Dr.Web Enterprise Security Suite Software and Its Components (read online).
6. Enable Dr.Web Agent, Dr.Web Server and the Network Installer protocols, disabled in step 4.

Restoring the DB under Windows OS

The procedure for starting and stopping Dr.Web Server is described in the **Administrator Manual**, section Starting and Stopping Dr.Web Server (read online).

To restore the DB from a backup copy

1. Stop Dr.Web Server if it is running.
2. Import the contents of the database from the corresponding backup file. The command will look as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb Server\var\importdb.log"
modexecdb database-import-and-upgrade "<path_to_the_backup_file>\content.gz"
```

In previous versions, the name of the database backup file is different (`database.gz`).

   The command must be entered in a single line. It is assumed that Dr.Web Server is installed in the `C:\Program Files\DrWeb Server` folder.
3. Start Dr.Web Server.

To restore the DB from a backup copy when the Dr.Web Server version is changed or the current DB version is corrupted

1. Stop Dr.Web Server if it is running.
2. Initialize a new database.
   - For the embedded DB:
     a) Move the `database.sqlite` file to a temporary directory until you are sure that the database has been successfully restored.
     b) In Windows the database initialization command will look as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=all -log="C:\Program Files\DrWeb Server\var\initdb.log"
modexecdb database-init
```

     It is assumed that Dr.Web Server is installed to the `C:\Program Files\DrWeb Server` folder.
     c) Once this command is executed, a new `database.sqlite` will be generated in the `var` subfolder of Dr.Web Server installation directory.
   - For the external DB:

a) Export the database file to a temporary directory until you are sure that the database has been successfully restored.

b) Clean up the DB using `modexecdb database-clean` command (see Appendix G3.3. Database Commands).

3. Import the contents of the database from the corresponding backup file. The command will look as follows:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb Server\var\importdb.log"
modexecdb database-import-and-upgrade "<path_to_the_backup_file>\content.gz"
```

The command must be entered in a single line. It is assumed that Dr.Web Server is installed in the `C:\Program Files\DrWeb Server` folder.

4. Start Dr.Web Server.

Restoring the DB under UNIX OS

1. Stop Dr.Web Server (if it is running):
   - for **Linux** OS:

```
/etc/init.d/drwcsd stop
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd stop
```

2. Move `database.sqlite` to a temporary directory until you are sure that the database has been successfully restored. Database file is located in the following subfolder of the Dr.Web Server installation directory:
   - for **Linux** OS: `/var/opt/drwcs/`
   - for **FreeBSD** OS: `/var/drwcs/`

To clean an external DB, use the `modexecdb database-clean` command (see Appendix G3.3. Database Commands).

3. Initialize the Dr.Web Server database. The command will look as follows:
   - for **Linux** OS:

```
/etc/init.d/drwcsd modexecdb database-init
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-init
```

4. After running this command, a new `database.sqlite` database will be generated in the `var` subfolder of the Dr.Web Server installation directory.

5. Import the contents of the database from the corresponding backup file. The command will look as follows:
   - for **Linux** OS:

```
/etc/init.d/drwcsd modexecdb database-import-and-upgrade
"<path_to_the_backup_file>/content.gz"
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-import-and-upgrade
"<path_to_the_backup_file>/content.gz"
```

6. Start Dr.Web Server:
   - for **Linux** OS:

```
/etc/init.d/drwcsd start
```

   - for **FreeBSD** OS:

```
/usr/local/etc/rc.d/drwcsd start
```

If you want to run the script with parameters (for example, specify the Dr.Web Server installation directory, etc.), change the appropriate values in the start script:

- for FreeBSD OS: `/usr/local/etc/rc.d/drwcsd`
- for Linux OS: `/etc/init.d/drwcsd`

If you need to change the log verbosity level of Dr.Web Server, use the `local.conf` file:

- for Linux OS: `/var/opt/drwcs/etc/local.conf`;
- for FreeBSD OS: `/var/drwcs/etc/local.conf`.

If some Dr.Web Agents were installed after the last backup had been made, they will not connect to Dr.Web Server after the database is restored from the backup. You should remotely reset them to Newbie mode. In the **Administration** menu → **Dr.Web Server configuration** on the **General** tab, set the **Reset unauthorized to newbie** flag and in the **Newbies registration mode** drop-down list, select **Allow access automatically**. Click **Save** and restart Dr.Web Server.

After all stations have been successfully connected to the new Dr.Web Server, change the settings of this Dr.Web Server to the settings adopted by your company policy.

As soon as the database is restored from the backup it is recommended to connect Dr.Web Security Control Center to Dr.Web Server. In the **Administration** menu, select **Dr.Web Server Task Scheduler** and make sure that the **Back up critical Dr.Web Server data** task is included in the list. If this task is missing, add it to the list.

Upgrading Dr.Web Agents on LAN Servers

When upgrading Dr.Web Agents installed on LAN servers, restarting stations or stopping a network software on such stations may be undesirable.

To avoid the loss of functionality of stations that perform important network functions due to downtime, it is recommended to upgrade Dr.Web Agents and anti-virus software as follows:

1. In the Dr.Web Server schedule, change the default upgrade tasks from all components to virus databases only.
2. Create a new upgrade task for all components to run at an appropriate time, when it does not negatively impact the functioning of LAN servers.

   The process of creating and editing tasks in the Dr.Web Server schedule is described in the **Administrator Manual**, section Setting Dr.Web Server Schedule (read online).

It is not recommended to install SpIDer Gate, SpIDer Mail and Dr.Web Firewall components on servers which perform important network functions (domain controllers, license distribution servers and etc.) to avoid probable conflicts between network services and internal components of Dr.Web anti-virus.

Using DFS when Installing Dr.Web Agent via Active Directory

You can use Distributed File System (DFS) when installing Dr.Web Agent via the Active Directory service.

It can be useful, for example, when your LAN has multiple domain controllers.

Installing Dr.Web Agent in a LAN with multiple domain controllers

1. Create a directory with the same name on each domain controller.
2. Using the DFS, map the created directories to one root destination directory.
3. Perform the administrative installation of the `*.msi` package to the created destination directory (see **Installation Manual**, section Installing Dr.Web Agent Software via Active Directory (read online).
4. Use this destination directory when assigning the package in the group policy object editor.

   Use the network address as: `\\<domain>\<folder>`

   where: <*domain*> is the domain name, <*folder*> is the name of destination directory.

Restoring the Anti-virus Network after Dr.Web Server Failure

In case of a Dr.Web Server fatal failure, it is recommended to use the following procedures to restore the anti-virus network operation without reinstalling Dr.Web Agents on stations.

The new Dr.Web Server should be installed on a computer with the same IP address and DNS name.

Restoring from Dr.Web Server Backup

During its operation, Dr.Web Server regularly stores backup copies of critical Dr.Web Server data: repository settings, configuration files, encryption keys, certificates, database backup, and hooks.

The backup files are stored in the following folders:

- for **Windows** OS: <*installation_drive*>`:\DrWeb Backup`
- for **Linux** OS: `/var/opt/drwcs/backup`
- for **FreeBSD** OS: `/var/drwcs/backup`

To perform the backup, a daily task is included in the Dr.Web Server schedule. If such a task is missing in the schedule, it is recommended to create it.

Restoring from backup is possible within one major version.

All files in the backup except the database contents, are ready to use. The backup copy of the database is stored in the `.gz` format compatible with `gzip` and other archivers. The contents of the database can be imported from the backup copy to another Dr.Web Server database using the `modexecdb database-import-and-upgrade` command. Thus the data can be fully restored.

To restore the database you can also use a backup created by the administrator manually via the Control Center in the **Administration** → **Database management** → **Export** (for the **Export entire database** mode only).

It is also recommended to store copies of created backups and other important files on another computer. Thus, you will be able to avoid data loss should the computer on which Dr.Web Server is installed be damaged and to fully restore the data and the functionality of Dr.Web Server. If license keys are lost, they may be requested once again, as specified in the **Administrator Manual**, section Licensing (read online).

Restoring Dr.Web Server after a failure if a backup is available

1. Select a computer to install the new Dr.Web Server. Isolate this computer from the online Dr.Web Agents: disconnect it from the network where Dr.Web Agents are installed or temporarily change its IP address, or use any other method you prefer.
2. Install the new Dr.Web Server. Installation must be performed with the empty database of the same major version that needs to be restored.
3. In the **Administration** → **License manager** section, add the license key from the previous Dr.Web Server installation and propagate it to the appropriate groups, particularly to the **Everyone** group. This step is mandatory if the license key was not set during Dr.Web Server installation.
4. Update the repository of the installed Dr.Web Server from the GUS:
   a) Open the **Administration** → **Repository state** section of the Control Center.
   b) Click the **Check for updates** button to check whether updates for all of products are available on the GUS servers and download them, if necessary.
5. If new versions of the Dr.Web Server software are available, perform the update to the latest version (within one major version):
   a) Open **Administration** → **Dr.Web Server** section of the Control Center.
   b) To open the Dr.Web Server version list, click on the current version of Dr.Web Server or click the **List of versions** button. This will open the **Dr.Web Server Updates** section with the list of available Dr.Web Server updates and backups.
   c) To update the Dr.Web Server software, set the flag next to the last version in the **All versions** list. Click **Apply**.
   d) Wait for the Dr.Web Server update process to complete.
6. Stop Dr.Web Server.
7. Replace the critical Dr.Web Server data with the saved data from the backup by copying the contents of the backup folders to the following folders:

Folder in backup	Folder to which copying is performed
`drwcsd`	Windows: <*installation_folder*>`\etc` Linux: `/var/opt/drwcs/etc` FreeBSD: `/var/drwcs/etc`
`webmin`	Windows: <*installation_folder*>`\etc` Linux: `/var/opt/drwcs/etc` FreeBSD: `/var/drwcs/etc`
`extensions`	Windows: <*installation_folder*>`\var\extensions`

Folder in backup	Folder to which copying is performed
	Linux: `/var/opt/drwcs/etc/extensions` FreeBSD: `/var/drwcs/etc/extensions`
`repository`	Windows: *<installation_folder>*`\var\repository` Linux: `/var/opt/drwcs/etc/repository` FreeBSD: `/var/drwcs/etc/repository`

8. Configure the database.
   a) Using external database:

   No further actions are required to connect the database to Dr.Web Server (as long as the Dr.Web Server configuration file was saved).

   If the version of Dr.Web Server installed using the latest updates is greater than the version of the faulty Dr.Web Server, update the external database using the `modexecdb database-upgrade` command:

   - for Windows OS:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -verbosity=all -log="C:
\Program Files\DrWeb Server\var\upgradedb.log" modexecdb database-
upgrade
```

   - for Linux OS:

```
/etc/init.d/drwcsd modexecdb database-upgrade
```

   - for FreeBSD OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-upgrade
```

   b) Using backup of external or embedded database:

   If you are using the external database, first clean it up using the `modexecdb database-clean` command (see Appendix G3.3. Database Commands).

   Import the database contents from the corresponding backup file and simultaneously update the database format to comply with the installed Dr.Web Server version using the `modexecdb database-import-and-upgrade` command:

   - for Windows OS:

```
"C:\Program Files\DrWeb Server\bin\drwcsd.exe" -home="C:\Program
Files\DrWeb Server" -var-root="C:\Program Files\DrWeb Server\var" -
verbosity=trace -log="C:\Program Files\DrWeb
Server\var\importupgradedb.log" modexecdb database-import-and-upgrade
"<path_to_the_backup_file>\content.gz"
```

In previous versions, the name of the database backup file is different (`database.gz`).

- for Linux OS:

```
/etc/init.d/drwcsd modexecdb database-import-and-upgrade
"<path_to_the_backup_file>/content.gz"
```

- for FreeBSD OS:

```
/usr/local/etc/rc.d/drwcsd modexecdb database-import-and-upgrade
"<path_to_the_backup_file>/content.gz"
```

Assign the same permissions to all replaced files as in the previous (corrupted) installation of Dr.Web Server.

For Unix-like OS: `rw` for `drwcs:drwcs`.

9. Start Dr.Web Server.
10. Make sure that the data from the database backup is safe and up-to-date: the Dr.Web Agent settings, anti-virus network tree state and etc.
11. Restore the access to Dr.Web Server for Dr.Web Agents according to the Dr.Web Server isolation method selected in step 1.

If some Dr.Web Agents were installed after the last backup had been made, they will not connect to Dr.Web Server after the database is restored from the backup. You should remotely reset them to Newbie mode. In the **Administration** → **Dr.Web Server configuration** on the **General** tab, set the **Reset unauthorized to newbie** flag and in the **Newbies registration mode** drop-down list, select **Allow access automatically**. Click **Save** and restart Dr.Web Server.

After all stations have successfully connected to the new Dr.Web Server, change these Dr.Web Server settings to the settings adopted by your company policy.

Restoring without Dr.Web Server Backup

Restoring Dr.Web Server after a failure if no backup had been saved

1. Select a computer to install the new Dr.Web Server. You can restore the connection to the server either on the same computer (with the same address) or on another computer (by changing the address for connecting to Dr.Web Server in the settings of Dr.Web Agents).

   To be able to perform initial settings of Dr.Web Server and anti-virus network before connecting Dr.Web Agents, isolate this computer from the online Dr.Web Agents: temporarily change the port to a port other than the default port 2193 (you can change it

back to the default later). After the initial setup, test the connection to Dr.Web Server using one or two Dr.Web Agents as an example.

It is possible to restore Dr.Web Server to the same computer (with the same address) or to another computer by changing the server connection address in the Dr.Web Agents settings.

2. Install the new Dr.Web Server.
3. In the **Administration** → **License manager** section, add the license key from the previous Dr.Web Server installation and propagate it to the corresponding groups, particularly to the **Everyone** group. This step is mandatory if the license key was not set during the Dr.Web Server installation.
4. Update the repository of the installed Dr.Web Server from the GUS:
   a) Open the **Administration** → **Repository state** section of the Control Center.
   b) Click the **Check for updates** button to make sure if updates for all products are available on the GUS servers and download them, if necessary.
5. If newer versions of the Dr.Web Server software are available, upgrade to the latest versions:
   a) Open **Administration** → **Dr.Web Server** section of the Control Center.
   b) To open the Dr.Web Server version list, click on the current version of Dr.Web Server or click the **List of versions** button. This will open the **Dr.Web Server Updates** section with the list of available Dr.Web Server updates and backups.
   c) To update the Dr.Web Server software, set the flag next to the latest version in the **All versions** list. Click **Save**.
   d) Wait for the Dr.Web Server update process to complete.
6. Change the station connection settings in the Dr.Web Server configuration:
   a) Open **Administration** → **Dr.Web Server configuration**.
   b) On the **General** tab, set the **Reset unauthorized to newbie** flag.
   c) On the **General** tab in the **Newbies registration mode** drop-down list, select **Allow access automatically**.
   d) Click **Save** and restart Dr.Web Server.
7. In the **Anti-virus Network** section of the Control Center, create user groups in the anti-virus network tree in the same way as in the previous version. If necessary, create automatic membership rules for stations in the created user groups.
8. If necessary, specify the Dr.Web Agent settings and the Dr.Web Server settings (except for the temporary settings from step 5) similarly to the previous version.
9. If necessary, change the repository settings in the **Administration** → **Detailed repository configuration** section.
10. Restore the access to Dr.Web Server for Dr.Web Agents according to the Dr.Web Server isolation method selected in step 1.
11. Replace the certificate on all stations of the network that will be connected to the new Dr.Web Server. You can download the certificate in the **Administration** → **Encryption keys** section.

- If the self-protection is enabled, copy the certificate created during the new Dr.Web Server installation to a station and run the following command:

  `"%ProgramFiles%\DrWeb\es-service.exe" -c <certificate>`

  or

  `"%ProgramFiles%\DrWeb\es-service.exe" --addcert=<certificate>`

  Substitute the *<certificate>* with the path to the server certificate copied to the station.

  As a result, the certificate will be copied to the Dr.Web Agent installation folder. By default, it is the `%ProgramFiles%\DrWeb` folder (for more details, see the Appendix G2. Dr.Web Agent for Windows).

- If the self-protection is disabled on a station, you can take the certificate created during the installation of the new Dr.Web Server and place it in the folder specified above.

12. If you restored connection to Dr.Web Server on another computer, change the address for connecting to the server in the Dr.Web Agents settings.

To perform the actions directly on the station, you must have administrative privileges on the station and, if you want to edit the configuration via the Dr.Web Agent interface, permission to edit Dr.Web Agent properties, which is granted on Dr.Web Server (section **Anti-virus Network** → **Configuration** → **Permissions** → **General** tab → **Change Dr.Web Agent configuration**). If you do not have these permissions, you can reconnect to another Dr.Web Server locally on the station only after removing the installed Dr.Web Agent via the Control Center and installing Dr.Web Agent again with the new Dr.Web Server settings.

It is recommended that you use the name of Dr.Web Server in the FQDN format as the Dr.Web Server address.

13. After all stations have been successfully connected to the new Dr.Web Server, change the Dr.Web Server settings specified in step 6 to the settings adopted by your company policy.

Restoring a Dr.Web Server Cluster Node

If one of the Dr.Web Servers in a cluster stops working, Dr.Web Agents that lost the connection to it will connect to another cluster node. In case of a fatal Dr.Web Server failure, you need to restore it: install Dr.Web Server software and add the Dr.Web Server to the cluster again.

Installing a Dr.Web Server to be added to a cluster

To be able to work with a common database, all Dr.Web Servers must be of the same version.

If you have a backup copy of the failed Dr.Web Server, you can use it to facilitate the cluster node restoration process. A backup copy contains repository settings, configuration files, encryption keys, certificates, database backup, and hooks.

When installing a Dr.Web Server to replace a cluster node, follow the directions below.

Installing Dr.Web Server for Unix-like OS

Follow the instructions from the **Installation Manual**, section Installing Dr.Web Server for Unix-like OS (read online) . You can install a new Dr.Web Server or use a backup copy of the Dr.Web Server that was part of the cluster and failed.

By default, backup copies are stored in the following folders:

- for Dr.Web Server under Linux—`/var/opt/drwcs/backup`,
- for Dr.Web Server under FreeBSD—`/var/drwcs/backup`.

A Dr.Web Server that was installed from a backup copy will use the common database accessed by the cluster nodes.

A Dr.Web Server installed without a backup copy will use the embedded database. Once the installation process is finished, the Dr.Web Server must be switched to the common database used by the cluster.

Installing Dr.Web Server for Windows OS

You can install a new Dr.Web Server or use a backup copy of the Dr.Web Server that was part of the cluster and failed. By default, backup copies are stored in the *<installation_drive>*`:\DrWeb Backup` folder.

Follow the instructions from the **Installation Manual**, section Installing Dr.Web Server for Windows OS (read online), taking note of the following points:

- It is inadvisable to connect Dr.Web Server to an external database during the installation process due to the risk of database data loss once the Dr.Web Server is connected to the cluster.
- When installing from a backup copy, set the following configuration parameters:
  - **Database driver**: select the **SQLite (embedded database)** option. It is not recommended to proceed with the external database as it can lead to data loss once Dr.Web Server connects to the cluster. Dr.Web Server needs to be reconnected to the external database used within the cluster only after the Dr.Web Server installation is finished.
  - **Network configuration**: set the **Interface** and **Port** values. If the **Enable Dr.Web Server detection service** flag is not set, set it to allow Dr.Web Server to communicate with the other cluster nodes via a multicast group. Specify the parameters of the multicast group used by Dr.Web Servers in the cluster below.

  The values of the other parameters can be specified according to the **Installing Dr.Web Server for Windows OS** section.
- When installing a new Dr.Web Server, specify the following configuration parameters:
  - **Dr.Web Server configuration**: specify the paths to the `drwcsd.conf` configuration file and the private encryption key used by the Dr.Web Server that failed, if you have them. Otherwise, leave the fields empty to create new encryption keys, certificate, and

configuration file with the default settings. The values of these parameters must be changed after the Dr.Web Server installation.

- **Database driver**: select the **SQLite (embedded database)** option. The Dr.Web Server needs to be connected to the external database used within the cluster only after the Dr.Web Server installation is finished.
- **Network configuration**: specify the **Interface** and **Port** values. Set the **Enable Dr.Web Server detection service** flag to allow the Dr.Web Server to communicate with other cluster nodes via a multicast group. Specify the multicast group parameters to be used by Dr.Web Servers in the cluster.

The values of other parameters can be set as described in the **Installing Dr.Web Server for Windows OS** section.

Adding a Dr.Web Server to a cluster

To add the installed Dr.Web Server to a cluster, follow the instructions from the **Administrator Manual**, section Dr.Web Server Cluster (read online) .

Managing the Logging Level of Dr.Web Server for Windows OS

You can change the level of verbosity for Dr.Web Server under Windows OS in one of the following ways:

- Using the **Dr.Web Server configuration** → **Log** section of the Control Center.

  This is the preferred method. In the **Log** section, you can specify any allowed level of logging detail for Dr.Web Server, as well as some other settings.

  Detailed information is given in the **Administrator Manual**, section Log (read online).

- Using the console command:

  ```
  drwcsd [<switches>] install
  ```

  You can specify any allowed level of verbosity for Dr.Web Server using the `--verbosity` switch.

  Detailed information on command line switches for the Dr.Web Server management is given in the G3.8. The Description of Switches section.

  Command example to set the logging level to **Trace**:

  ```
  drwcsd --daemon "--home=C:\Program Files\DrWeb Server" "--bin-root=C:
  \Program Files\DrWeb Server" "--var-root=C:\Program Files\DrWeb Server\var"
  --verbosity=trace --log="C:\Program Files\DrWeb Server\var\install.log" --
  rotate=10,50m install
  ```

  Other switches are mandatory, particularly, if you have redefined the default paths of the Dr.Web Server installation and working folders.

  After changing the log verbosity level, restart Dr.Web Server:

```
drwcsd restart
```

- Using the commands in the **Start** main menu of Windows OS.

  In this case, only two possible levels of logging detail are available: **Detailed** or **Default**:

  a) **Programs** → **Dr.Web Server** → **Detailed logging** or **Programs** → **Dr.Web Server** → **Default logging**

  b) **Programs** → **Dr.Web Server** → **Restart**.

Automatic Location of Stations under Android OS

Dr.Web Enterprise Security Suite can automatically provide administrators with information about the geographical location of protected mobile devices running Android OS.

Locating a mobile device

1. Configure transmission of location data from a mobile device to Dr.Web Server:
   a) In the Dr.Web Security Control Center, navigate to the **Anti-virus network** section, select the required station or a group of stations running Android OS in the network tree.
   b) Select the **Dr.Web Mobile Security Suite (Android)** item in the control menu.
   c) On the **General** tab, set the **Track location** flag. In the **Period of coordinates transmission** drop-down list, select a value according to which the device location data will be updated.
   d) Save the changes.
2. Automatic location tracking is performed in one of the following ways:
   - If location providers (GPS, cellular networks) are enabled on a user device and the signal is stable, the location is monitored using the mobile device itself.
   - If location providers (GPS, cellular networks) are disabled on a user device or there is no GPS signal, Dr.Web Enterprise Security Suite uses the Yandex.Locator technology to locate a mobile device. This technology is based on the coordinates of the cell phone towers (GSM, 3D, LTE) and WiFi ID.

     To configure the Yandex.Locator technology, you must activate and set up the **Yandex.Locator Extension**:

     a) Get the API key from the Yandex website at https://yandex.ru/dev/locator/keys/.
     b) In the Dr.Web Security Control Center navigate to the **Administration** → **Dr.Web Server configuration** → **Modules** section, set the **Yandex.Locator Extension** flag.
     c) In the **API key** field, specify the key received in step a).
     d) Save the changes and restart Dr.Web Server.

WiFi ID can only be used on mobile devices running Android 5.1 or earlier.

3. To view the location of a station in the Dr.Web Security Control Center:
   a) In the **Anti-virus network** section navigate in the network tree to the station for which the appropriate settings were specified in step 1.
   b) In the **Location** section of the station properties, the geographical coordinates received from a mobile device will be displayed automatically.
   c) Click **Show on map** to view the geographical location of a mobile device on OpenStreetMap according to the received coordinates.

Functional Analysis Criteria

When setting up functional analysis, it is recommended to set the functional analysis criteria to achieve the maximum level of protection.

The **Functional analysis criteria** section contains information on the categories that you can set to protect your profile. You can choose a category based on the level of protection you need. The default value for all parameters is **Disabled**.

Categories of functional analysis criteria

Application launch

Enables control of running processes for the list of trusted applications.

- *Prevent running of applications signed by certificates known in Doctor Web as certificates for adware*.

  This criterion blocks the launch of applications that may distribute advertising.
- *Prevent running of applications signed by certificates known in Doctor Web as gray*.

  This criterion blocks the launch of applications signed by "gray" certificates. These certificates are often used to sign insecure applications.
- *Prevent running of applications signed by certificates known in Doctor Web as certificates for hacktools*.

  This criterion blocks the launch of applications that threaten the system security. It is recommended to set this criterion.
- *Prevent running of applications signed by fake/malformed certificates*.

  This criterion blocks the launch of malicious applications signed with invalid certificates (corrupted or attached to a binary file in order to prevent the threat from being detected—for example, legitimate software certificates). It can also help when trying to modify a legitimate file or infect it with a virus. It is recommended to set this criterion.
- *Prevent running of applications signed by certificates known in Doctor Web as certificates for malware*.

  This criterion blocks the launch of applications signed with compromised certificates. It is recommended to set this criterion.
- *Prevent running of applications signed by revoked certificates*.

  This criterion blocks the launch of applications signed with stolen or compromised certificates. This criterion allows to prevent the launch of potentially malicious applications. It is recommended to set this criterion.
- *Prevent running of applications signed by self-signed certificates*.

  This criterion blocks unlicensed software that may be malicious. Malicious programs can add a fake signature with a well-known name (for example, Microsoft) to their binary files and/or

add a root certificate to the system so that this file is shown and recognized by the OS as legally signed.

- *Prevent running of unsigned applications*.

  This criterion blocks the launch of potentially malicious and untrusted applications of unknown origin.

- *Prevent running of Sysinternals utilities*.

  This criterion blocks the launch of Sysinternals utilities which are often used to compromise the system.

> If the **Allow running of system applications and Microsoft company applications** flag is set in the **Permissions** section, Sysinternals utilities will run even if they are not allowed to run.

- *Prevent running of applications from NTFS (ADS) alternate threads*.

  Applications saved in NTFS alternate data streams (ADS) are often malicious. It is recommended to set this criterion.

- *Prevent running of applications from network and shares*.

  Launching applications from the network and shared resources is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of applications from removable media*.

  Launching applications from removable media is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of applications from temporary folders*.

  Launching applications from temporary folders is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of Windows/Microsoft Store applications (only for Windows 8 and later)*.

  This criterion blocks the launch of applications downloaded from Windows/Microsoft Store.

- *Prevent running of applications with double/non-typical extension*.

  This criterion blocks the launch of suspicious files with an atypical extension (for example, *.jpg.exe).

- *Prevent running of bash shells and WSL applications (only for Windows 10 and later)*.

  This criterion blocks the launch of bash command shells and WSL applications.

- *Prevent running of child processes of office applications (Microsoft Word, Excel, Access, etc.)*.

  This criterion blocks the launch of child processes of office applications, such as Microsoft Word, Excel, Access, PowerPoint.

- *Prevent running of child processes of mail clients*.

  This criterion blocks the launch of child processes from email clients, such as Microsoft Outlook.

- *Prevent running of child processes of PDF viewers*.

This criterion blocks the launch of child processes of PDF viewers.

Exceptions:

- *Allow running of system applications and Microsoft company applications*.
- *Allow running of known/trusted by Doctor Web applications*.

  If enabled, applications signed with a trusted certificate are allowed to launch.

  

  If this option is enabled, applications signed with a trusted certificate are allowed to run. This feature allows you not to create an excessive number of rules and base on the data already checked by Dr.Web. Trust in this case is based on cryptography, an extensive and constantly updated database.

  

  These exceptions do not apply to the following categories:

  - Prevent running of child processes of office applications (Microsoft Word, Excel, Access, etc.).
  - Prevent running of child processes of mail clients.

Modules load and execution

Enables control of loaded modules. In this category, you can specify two operating modes:

- *Control all modules load and execution*.

  A global option that enables control for the modules in the list of allowed objects. This mode is resource-intensive. It is recommended to use it only if you need enhanced protection.
- *Control modules load and execution in host applications*.

  This mode is less resource-intensive. This mode controls the operation of modules only in processes that can be used to compromise the system or in cases when malware can get into a system by masquerading as a system or trusted file. Use this mode if there is no need for enhanced protection. In this mode you can:
  - *Prevent loading and execution of modules signed by certificates known in Doctor Web as certificates for adware*.

    This criterion blocks the launch of applications that may distribute advertising.
  - *Prevent loading and execution of modules signed by certificates known in Doctor Web as gray*.

    This criterion blocks the launch of applications signed by "gray" certificates. These certificates are often used to sign insecure applications.
  - *Prevent loading and execution of modules signed by certificates known in Doctor Web as certificates for hacktools*.

    This criterion blocks the launch of applications that threaten the system security. It is recommended to set this criterion.
  - *Prevent loading and execution of modules signed by fake/malformed certificates*.

This criterion blocks the launch of malicious applications signed with invalid certificates (corrupted or attached to a binary file in order to prevent the threat from being detected —for example, legitimate software certificates). It can also help when trying to modify a legitimate file or infect it with a virus. It is recommended to set this criterion.

- *Prevent loading and execution of modules signed by certificates known in Doctor Web as certificates for malware*.

  This criterion blocks the launch of applications signed with compromised certificates. It is recommended to set this criterion.

- *Prevent loading and execution of modules signed by revoked certificates*.

  This criterion blocks the launch of applications signed with stolen or compromised certificates. This criterion allows to prevent the launch of potentially malicious applications. It is recommended to set this criterion.

- *Prevent loading and execution of modules signed by self-signed certificates*.

  This criterion blocks unlicensed software that may be malicious. Malicious programs can add a fake signature with a well-known name (for example, Microsoft) to their binary files and/or add a root certificate to the system so that this file is shown and recognized by the OS as legally signed.

- *Prevent running of unsigned modules*.

  This criterion blocks the launch of potentially malicious and untrusted applications of unknown origin.

- *Prevent loading and execution of modules from NTFS (ADS) alternate threads*.

  Applications saved in NTFS alternate data streams (ADS) are often malicious. It is recommended to set this criterion.

- *Prevent loading and execution of modules from network and shares*.

  Launching applications from the network and shared resources is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading and execution of modules from removable media*.

  Launching applications from removable media is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading and execution of modules from temporary folders*.

  Launching applications from temporary folders is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading and execution of modules with double/non-typical extension*.

  Blocks the launch of suspicious applications with a non-standard extension (for example, *.jpg.exe).

Exceptions:

- *Allow loading and execution system and Microsoft modules*.

- *Allow loading and execution modules known/trusted by Doctor Web*.

  If enabled, modules signed with a trusted certificate are allowed to launch.

Launch of script interpreters

Enables control of launched script interpreters for the list of trusted applications.

- *Prevent running of CMD/BAT scripts*.

  This criterion blocks the launch of cmd and bat files.

- *Prevent running of HTA scripts*.

  This criterion blocks the launch of HTA scripts. Such scripts can process malicious scenarios and download executable files which can be harmful to your computer.

- *Prevent running of VBScript/JavaScript*.

  This criterion blocks the launch of applications written in the VBScript and JavaScript languages. Such applications can process malicious scripts and download executable files which can be harmful to your computer.

- *Prevent running of PowerShell scripts*.

  This criterion blocks the launch of scripts written in the PowerShell language. Such scripts can process malicious scripts and download executable files to your computer.

- *Prevent running of REG scripts*.

  This criterion blocks the launch of registry scripts (files with the reg extension). These files can be used to add or change values in the registry.

- *Prevent running of scripts from NTFS (ADS) alternate threads*.

  Applications saved in NTFS alternate data streams (ADS) are often malicious. It is recommended to set this criterion.

- *Prevent running of scripts from network and shares*.

  Launching applications from the network and shared resources is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of scripts from removable media*.

  Launching applications from removable media is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of scripts from temporary folders*.

  Launching applications from temporary folders is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent running of scripts from office applications (Microsoft Word, Excel, Access, etc.)*.

  This criterion blocks the launch of scripts from office applications, such as Microsoft Word, Excel, Access, PowerPoint.

- *Prevent running of scripts from mail clients*.

  This criterion blocks the launch of scripts from email clients, such as Microsoft Outlook.

- *Prevent running of AutoIt scripts*.

  This criterion blocks the launch of scripts written in the AutoIt language.

- *Prevent running of Java applications*.

This criterion blocks the launch of scripts written in the Java language. Such scripts may be dangerous due to the peculiarities of the Java platform, vulnerabilities in the program code, and in application components.

- *Prevent running of Python scripts*.

  This criterion blocks the launch of scripts written in the Python language.

- *Prevent scripts from PDF viewers from launching*.

  This criterion blocks the launch of scripts from PDF viewers.

Exceptions:

- *Allow running of system scripts and Microsoft company scripts*.
- *Allow running of known/trusted by Doctor Web scripts*.

  If enabled, scripts signed with a trusted certificate are allowed to run.

These exceptions do not apply to the following categories:

- Prevent running of scripts from office applications (Microsoft Word, Excel, Access, etc.).
- Prevent running of scripts from mail clients.

Drivers loading

Control driver loading for the list of trusted applications.

- *Prevent loading of drivers signed by certificates known in Doctor Web as certificates for adware*.

  This criterion blocks the launch of applications that may distribute advertising.

- *Prevent loading of drivers signed by certificates known in Doctor Web as gray*.

  This criterion blocks the launch of applications signed by "gray" certificates. These certificates are often used to sign insecure applications.

- *Prevent loading of drivers signed by certificates known in Doctor Web as certificates for hacktools*.

  This criterion blocks the launch of applications that threaten the system security. It is recommended to set this criterion.

- *Prevent loading of drivers signed by fake/malformed certificates*.

  This criterion blocks the launch of malicious applications signed with invalid certificates (corrupted or attached to a binary file in order to prevent the threat from being detected—for example, legitimate software certificates). It can also help when trying to modify a legitimate file or infect it with a virus. It is recommended to set this criterion.

- *Prevent loading of drivers signed by certificates known in Doctor Web as certificates for malware*.

  This criterion blocks the launch of applications signed with compromised certificates. It is recommended to set this criterion.

- *Prevent loading of drivers signed by revoked certificates*.

This criterion blocks the launch of applications signed with stolen or compromised certificates. This criterion allows to prevent the launch of potentially malicious applications. It is recommended to set this criterion.

- *Prevent loading of drivers signed by self-signed certificates*.

  This criterion blocks unlicensed software that may be malicious. Malicious programs can add a fake signature with a well-known name (for example, Microsoft) to their binary files and/or add a root certificate to the system so that this file is shown and recognized by the OS as legally signed.

- *Prevent loading of unsigned drivers*.

  This criterion blocks the launch of potentially malicious and untrusted applications of unknown origin.

- *Prevent loading of drivers from NTFS (ADS) alternate threads*.

  Applications saved in NTFS alternate data streams (ADS) are often malicious. It is recommended to set this criterion.

- *Prevent loading of drivers from network and shares*.

  Launching applications from the network and shared resources is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading of drivers from removable media*.

  Launching applications from removable media is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading of drivers from temporary folders*.

  Launching applications from temporary folders is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent loading of vulnerable driver versions of popular software*.

  This criterion blocks the loading of vulnerable driver versions of popular software. Legitimate software drivers, such as VirtualBox, Asus, etc. can be used to get into the system via RDP. When this option is enabled, unsafe versions of these drivers will be blocked from loading. This policy cannot be blocked by exceptions.

  

  The **Prevent loading vulnerable driver versions of popular software** criterion cannot be overridden by exclusions.

- *Prevent loading of drivers without a WHQL signature*.

  This criterion blocks the launch of drivers without a digital signature confirming that they have passed Windows Hardware Quality Lab testing.

Exceptions:

- *Allow loading of system drivers and Microsoft company drivers*.
- *Allow loading of known/trusted by Doctor Web drivers*.

  If enabled, applications signed with a trusted certificate are allowed to launch.

MSI packages installation

Enables control of the launched MSI packages for the list of trusted applications.

- *Prevent installation of packages signed by certificates known in Doctor Web as certificates for adware*.

  This criterion blocks the launch of applications that may distribute advertising.

- *Prevent installation of packages signed by certificates known in Doctor Web as gray*.

  This criterion blocks the launch of applications signed by "gray" certificates. These certificates are often used to sign insecure applications.

- *Prevent installation of packages signed by certificates known in Doctor Web as certificates for hacktools*.

  This criterion blocks the launch of applications that threaten the system security. It is recommended to set this criterion.

- *Prevent installation of packages signed by fake/malformed certificates*.

  This criterion blocks the launch of malicious applications signed with invalid certificates (corrupted or attached to a binary file in order to prevent the threat from being detected—for example, legitimate software certificates). It can also help when trying to modify a legitimate file or infect it with a virus. It is recommended to set this criterion.

- *Prevent installation of packages signed by certificates known in Doctor Web as certificates for malware*.

  This criterion blocks the launch of applications signed with compromised certificates. It is recommended to set this criterion.

- *Prevent installation of packages signed by revoked certificates*.

  This criterion blocks the launch of applications signed with stolen or compromised certificates. This criterion allows to prevent the launch of potentially malicious applications. It is recommended to set this criterion.

- *Prevent installation of packages signed by self-signed certificates*.

  This criterion blocks unlicensed software that may be malicious. Malicious programs can add a fake signature with a well-known name (for example, Microsoft) to their binary files and/or add a root certificate to the system so that this file is shown and recognized by the OS as legally signed.

- *Prevent installation of unsigned packages*.

  This criterion blocks the launch of potentially malicious and untrusted applications of unknown origin.

- *Prevent installation of packages from NTFS (ADS) alternate threads*.

  Applications saved in NTFS alternate data streams (ADS) are often malicious. It is recommended to set this criterion.

- *Prevent installation of packages from network and shares*.

Launching applications from the network and shared resources is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent installation of packages from removable media*.

  Launching applications from removable media is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

- *Prevent installation of packages from temporary folders*.

  Launching applications from temporary folders is an atypical scenario that can threaten the system security. It is recommended to set this criterion.

Exceptions:

- *Allow installation of system packages and Microsoft company packages*.
- *Allow installation of known/trusted by Doctor Web packages*.

  If enabled, applications signed with a trusted certificate are allowed to launch.

Executable files integrity

Enables control of executable files integrity. These criteria are only used in systems that run in a trusted execution environment. In such systems, all processes are controlled by an administrator (for example, ATMs and other systems). If this criteria is used in other systems, its behavior is unpredictable. In this case, the risk of station failure is high.

- *Prevent creating new executable files*.

  This criterion blocks attempts to create new executable files on disk.

- *Prevent modification of executable files*.

  This criterion blocks attempts to modify existing executable files on disk.

Exceptions:

- *Allow creation and modification of executable files by signed system applications and Microsoft company applications*.
- *Allow creation and modification of executable files by signed known/trusted by Doctor Web applications*.

  If enabled, applications signed with a trusted certificate are allowed to launch.

Criteria of the **Executable files integrity** category cannot be overridden by the allow/deny rules.

Examples of Dr.Web Server Database Queries

Provided below are a few examples of SQL queries to a PostgreSQL database. Queries to other databases may contain some differences due to the features of the database itself and the subtleties of its use.

For ease of understanding, the examples do not take the hierarchy of groups and stations into account and use less performance-optimal but easier-to-understand syntax.

To query the database directly

1. Open the Control Center of your Dr.Web Server.
2. Go to the **Administration** → **SQL console** section.
3. Write an SQL query. Examples of queries are given below.
4. Click **Execute**.

Examples of SQL queries

1. Find stations running server versions of Windows OS that are using virus databases issued before 2024.07.16-00:00:00 UTC.

```
SELECT
  stations.name Station,
  groups_list.name OS,
  station_products.crev Bases
FROM
  stations
  INNER JOIN groups_list ON groups_list.platform =(
    CAST(stations.lastos AS INTEGER) & ~15728640
  )
  AND (
    (
      CAST(stations.lastos AS INTEGER) & 2130706560
    ) = 33554560
  )
  INNER JOIN station_products ON station_products.id = stations.id
  AND station_products.product = '10-drwbases'
  AND station_products.crev < 13020240716000000;
```

Data on the platforms of stations in the anti-virus network is stored as numeric codes.

To find out the correspondence between the code and the name of the station platform

1. Open the Control Center of your Dr.Web Server.
2. Navigate to the **Administration** → **SQL console** section.
3. Run the following command:
   ```
   SELECT DISTINCT lastos FROM stations
   ```
4. Click **Execute**.

   You will see a list of codes corresponding to platforms that have been identified on the stations of this anti-virus network.
5. Save the result of running the query to a file using the buttons on the toolbar or leave the tab open for further use of the obtained data.
6. Open the **Administration** → **Lua console** section.

7. Enter a command of the following form:

   ```
   return { ['<code>'] = dwcore.platform_str(<code>)}
   ```

   where <*code*> is one of the numeric codes obtained with the SQL query.
8. Click **Execute**.

2. Find stations with **High** and **Maximum** severity records in the **Anti-virus network** → **Statistics** → **Status** section.

```
SELECT
  stations.name Station
FROM
  stations
WHERE
  id IN (
    SELECT
      DISTINCT id
    FROM
      station_status
    WHERE
      severity >= 1342177280
  );
```

Data on station status severity is stored as numeric codes:

`0`—minimum,

`268435456`—low,

`805306368`—medium,

`1342177280`—high,

`1879048192`—maximum.

You can also check the station status severity codes the following way:

1. Open the Control Center of your Dr.Web Server.
2. Navigate to the **Administration** → **Lua console** section.
3. Run the following command:

   ```
   require 'statusmessages' return statusmessages.severity
   ```
4. Click **Execute**.

3. Get the correspondence between station status messages and the number of stations with each of these statuses.

```
SELECT
  code Code,
  COUNT(code) Num
FROM
  (
    SELECT
      DISTINCT id,
      code
    FROM
      station_status
  ) AS t
```

```
GROUP BY
  Code
ORDER BY
  Code;
```

Data on station status messages is stored as numeric codes.

To find out the correspondence between the codes and the station status messages

1. Open the Control Center of your Dr.Web Server.
2. Navigate to the **Administration** → **Lua console** section.
3. Enter the following code:

```
local sm = require 'statusmessages'
local messageByCode = require 'stats/message_by_code'
local codes = {}
for _,v in pairs(sm) do
  local n = tonumber(v)
  if n ~= nil then
    code = sm.get_code( n )
    codes[code] = messageByCode( code )
  end
end
return codes
```

4. Click **Execute**.

4. Get the top 10 threats detected from 2024.06.01 through 2024.07.01 on stations included in the group with the identifier '373a9afb-9c9a-4d4d-b9b1-de817b96bcc5' or in any nested groups.

```
SELECT
  cat_virus.str Threat,
  COUNT(cat_virus.str) Num
FROM
  station_infection
  INNER JOIN cat_virus ON cat_virus.id = station_infection.virus
WHERE
  station_infection.infectiontime BETWEEN 20240601000000000
  AND 20240701000000000
  AND station_infection.id IN (
    SELECT
      sid
    FROM
      station_groups
    WHERE
      gid = '373a9afb-9c9a-4d4d-b9b1-de817b96bcc5'
      OR gid IN (
        SELECT
          child
        FROM
          group_children
        WHERE
          id = '373a9afb-9c9a-4d4d-b9b1-de817b96bcc5'
      )
  )
```

```
GROUP BY
  cat_virus.str
ORDER BY
  Num DESC
LIMIT
  10;
```

5. Get the top 10 infected stations.

```
SELECT
  Station,
  Grp,
  Num
FROM
  (
    SELECT
      stations.id,
      groups_list.id,
      stations.name Station,
      groups_list.name Grp,
      COUNT(stations.id) Num
    FROM
      station_infection
      INNER JOIN stations ON station_infection.id = stations.id
      INNER JOIN groups_list ON groups_list.id = stations.gid
    GROUP BY
      stations.id,
      groups_list.id,
      stations.name,
      groups_list.name
    ORDER BY
      Num DESC
    LIMIT
      10
  ) AS t;
```

6. Remove all stations from any user groups that are not primary for those stations.

```
DELETE FROM
  station_groups;
INSERT INTO station_groups(sid, gid)
SELECT
  stations.id,
  groups_list.id
FROM
  stations
  INNER JOIN groups_list ON stations.gid = groups_list.id
  AND groups_list.type NOT IN(1, 4);
```

7. Find objects on the anti-virus network that have the specified domain whitelisted in the personal settings of the SpIDer Gate component.

```
SELECT
  stations.name Station
FROM
  station_cfg
  INNER JOIN stations ON stations.id = station_cfg.id
WHERE
  station_cfg.component = 38
  AND station_cfg.name = 'WhiteVirUrlList'
  AND station_cfg.value = 'domain.tld';
```

```
SELECT
  groups_list.name Grp
FROM
  group_cfg
  INNER JOIN groups_list ON groups_list.id = group_cfg.id
WHERE
  group_cfg.component = 38
  AND group_cfg.name = 'WhiteVirUrlList'
  AND group_cfg.value = 'domain.tld';
SELECT
  policy_list.name Policy
FROM
  policy_cfg
  INNER JOIN policy_list ON policy_list.id = policy_cfg.id
WHERE
  policy_cfg.component = 38
  AND policy_cfg.name = 'WhiteVirUrlList'
  AND policy_cfg.value = 'domain.tld';
```

8. Get the events of unsuccessful administrator logins to the Control Center with the corresponding authorization error codes from the audit log.

```
SELECT
  admin_activity.login Login,
  admin_activity.address Address,
  activity_data.value ErrorCode,
  admin_activity.createtime EventTimestamp
FROM
  admin_activity
  INNER JOIN activity_data ON admin_activity.record = activity_data.record
WHERE
  admin_activity.oper = 10100
  AND admin_activity.status != 1
  AND activity_data.item = 'Error';
```

9. Find stations running Windows OS that do not have the required security patches installed.

   Option 1:

```
SELECT
  stations.name Station
FROM
  stations
WHERE
  id NOT IN (
    SELECT
      station_env_kb.id
    FROM
      station_env_kb
      INNER JOIN stations ON stations.id = station_env_kb.id
    WHERE
      (
        CAST(stations.lastos AS INTEGER) & 2130706432
      )= 33554432
      AND station_env_kb.name IN (
        SELECT
          id
        FROM
          env_strings
        WHERE
```

```
          str LIKE 'KB4012212:%'
          OR str LIKE 'KB4012213:%'
          OR str LIKE 'KB4012214:%'
          OR str LIKE 'KB4012215:%'
          OR str LIKE 'KB4012216:%'
          OR str LIKE 'KB4012217:%'
          OR str LIKE 'KB4012598:%'
          OR str IN(
            'KB4012212', 'KB4012213', 'KB4012214',
            'KB4012215', 'KB4012216', 'KB4012217',
            'KB4012598'
          )
      )
  );
```

Data on the platforms of stations in the anti-virus network is stored as numeric codes.

Option 2, where '24e27d73-d21d-b211-a78c-85419c46f0e6' is the UUID of the **Windows** system group:

```
SELECT
  stations.name Station
FROM
  stations
WHERE
  id NOT IN (
    SELECT
      station_env_kb.id
    FROM
      station_env_kb
      INNER JOIN stations ON stations.id = station_env_kb.id
    WHERE
      stations.osgroup IN
      (
        SELECT
          child
        FROM
          group_children
        WHERE
          id='24e27d73-d21d-b211-a78c-85419c46f0e6'
        UNION ALL
        SELECT
          '24e27d73-d21d-b211-a78c-85419c46f0e6'
      )
      AND station_env_kb.name IN (
        SELECT
          id
        FROM
          env_strings
        WHERE
          str LIKE 'KB4012212:%'
          OR str LIKE 'KB4012213:%'
          OR str LIKE 'KB4012214:%'
          OR str LIKE 'KB4012215:%'
          OR str LIKE 'KB4012216:%'
          OR str LIKE 'KB4012217:%'
```

```
            OR str LIKE 'KB4012598:%'
            OR str IN(
              'KB4012212', 'KB4012213', 'KB4012214',
              'KB4012215', 'KB4012216', 'KB4012217',
              'KB4012598'
            )
        )
    );
```

Replacing the Web Server Private Key and Certificate

The Control Center web server uses a self-signed certificate to establish secure HTTPS connections. Your company's security policy may require you to replace the web server private key and certificate in order to use Dr.Web Enterprise Security Suite. In this case you can replace the original web server files with your own or use the `drwsign` utility shipped with Dr.Web Enterprise Security Suite to generate a new private key and self-signed certificate and replace the files.

To replace the web server private key and certificate with your own files

1. Make sure that you have separate private key and certificate files in the PEM format at your disposal.

   Only PEM files with no empty lines between the header and body of the certificate are allowed. The file may be a certificate or a certificate chain.

   You can use the `openssl` toolkit to extract the private key and certificate from a PFX file, as well as to convert your private key and certificate in another format to PEM. Detailed information on using the toolkit is provided on the official OpenSSL website at https://docs.openssl.org/3.6/man1/openssl/.

2. Rename your private key file to `private-key.pem` and your certificate file to `certificate.pem`.
3. Copy the original web server files named `private-key.pem` and `certificate.pem` from the following folder to another one as a backup copy:
   - Windows: `%programfiles%\DrWeb Server\etc`
   - Linux: `/var/opt/drwcs/etc`
   - FreeBSD: `/var/drwcs/etc`
4. Replace the original `private-key.pem` and `certificate.pem` web server files in the `etc` subfolder with your own files.

   If your computer is running Windows OS and is under the protection of a Dr.Web product, you will need to temporarily disable the **Self-Protection** component in order to replace the files.

   1. Open the Dr.Web menu and select **Security Center**.

2. Make sure that Dr.Web is operating in administrative mode (the lock in the bottom part of the program is "open"). Otherwise click the lock icon.
3. Click **Settings** in the upper part of the program window.
4. A window with the main settings of the program will open. Click the **Self-Protection** item in the left part of the window.
5. Use the **Enable Self-Protection (recommended)** toggle button to disable Self-Protection.

Once you are done manipulating the files, re-enable the component to allow it to continue protecting the integrity of files in Dr.Web products.

5. In a Unix-like OS: assign the `drwcs` user as the owner of the files you replaced in the `etc` subfolder.
   - Linux:

```
# chown drwcs: /var/opt/drwcs/etc/private-
key.pem /var/opt/drwcs/etc/certificate.pem
```

   - FreeBSD:

```
# chown drwcs: /var/drwcs/etc/private-
key.pem /var/drwcs/etc/certificate.pem
```

6. Restart Dr.Web Server.
   - Windows: using Windows OS service management tools or the **Start** → **All programs** → **Dr.Web Server** → **Restart** menu.
   - Linux:

```
# /etc/init.d/drwcsd restart
```

   - FreeBSD:

```
# /usr/local/etc/rc.d/drwcsd restart
```

7. Wait a few minutes and try to connect to the Control Center.

You can also use the `drwsign` digital key and certificate generation utility in order to generate a new private key and self-signed certificate and place them in the appropriate folder.

Detailed information on the features and command format of the `drwsign` utility is provided in the G7.1. Digital Keys and Certificates Generation Utility section.

To generate and replace the web server private key and certificate using the drwsign utility

1. Copy the original `private-key.pem` private key and `certificate.pem` certificate files used by the web server from the following folder to another one as a backup copy:

- Windows: `%programfiles%\DrWeb Server\etc`
- Linux: `/var/opt/drwcs/etc`
- FreeBSD: `/var/drwcs/etc`

2. Run the following command to generate a new `private-key.pem` private key file and a `certificate.pem` self-signed certificate file and replace the original files in the `etc` subfolder.
   - Windows:

```
"C:\Program Files\DrWeb Server\bin\drwsign.exe" genselfsign "C:\Program
Files\DrWeb Server\etc\private-key.pem" "C:\Program Files\DrWeb
Server\etc\certificate.pem"
```

   - Linux:

```
# /opt/drwcs/bin/drwsign genselfsign /var/opt/drwcs/etc/private-
key.pem /var/opt/drwcs/etc/certificate.pem
```

   - FreeBSD:

```
# /usr/local/drwcs/bin/drwsign genselfsign /var/drwcs/etc/private-
key.pem /var/drwcs/etc/certificate.pem
```

3. In a Unix-like OS: assign the `drwcs` user as the owner of the generated files.
   - Linux:

```
# chown drwcs: /var/opt/drwcs/etc/private-
key.pem /var/opt/drwcs/etc/certificate.pem
```

   - FreeBSD:

```
# chown drwcs: /var/drwcs/etc/private-
key.pem /var/drwcs/etc/certificate.pem
```

4. Restart Dr.Web Server.
   - Windows: using Windows OS service management tools or the **Start** → **All programs** → **Dr.Web Server** → **Restart** menu.
   - Linux:

```
# /etc/init.d/drwcsd restart
```

   - FreeBSD:

```
# /usr/local/etc/rc.d/drwcsd restart
```

5. Wait a few minutes and try to connect to the Control Center.

Chapter 4: Troubleshooting

Remote Installation Troubleshooting

Principle of installation:

1. Dr.Web Server connects to the `ADMIN$` resource on the remote station (`\\<remote_station>\ADMIN$\Temp`) and copies the network installer `drwinst.exe` that is located in the `webmin\install\windows` folder of the Dr.Web Server installation folder and SSL certificate `drwcsd-certificate.pem` located in the `etc` folder of the Dr.Web Server installation directory, to the `\\<remote_station>\ADMIN$\Temp` folder.
2. Dr.Web Server runs the `drwinst.exe` file on the remote station with the command line switches according to the Control Center settings.

Successful installation requires the following to be available to Dr.Web Server from which the installation is performed:

1. The `ADMIN$\Temp` resource must be available on the remote station.

   The availability can be checked as follows:

   In the address bar of `Windows Explorer` enter the following:

   `\\<remote_station>\ADMIN$\Temp`

   You will be prompted to enter the login and password to assess this resource. Enter the account information provided on the installation page.

   The `ADMIN$\Temp` resource may be unavailable for the following reasons:

   a) the account does not have administrative privileges;

   b) the station is turned off or the firewall blocks access to port 455;

   c) the access to the `ADMIN$\Temp` resource may be limited on Windows Vista and later operating systems, if the station is outside a domain;

   d) the folder owner is not specified or the user or group has insufficient permissions to access the folder.
2. The `drwinst.exe` file and the `*.pem` certificate must be available.

Dr.Web Security Control Center displays detailed information (step and error code), that can help to diagnose the cause of the error.

List of Dr.Web Agent remote installation errors

Step	Error	Reason
Connecting via SMB to the <*host*> station	Invalid address of the <*host*> station	Station IP address that is specified for the Dr.Web Agent installation is not a valid IPv4/IPv6 address or conversion of DNS name to address failed: there is no such DNS name or the name server is incorrectly configured.

	Error connecting via SMB to the <*host*> station	Unable to connect to a station via SMB. Possible reasons are:
		• The Server service is disabled on the station;
		• Port 445 port is not available on the remote station, possible reasons are:
		  ▫ station is turned off;
		  ▫ firewall blocks the specified port;
		  ▫ the operating system on the remote station is not Windows.
		• Sharing and security model for local accounts is not configured;
		• The authorization server (domain controller) is not available;
		• Unknown user name or incorrect password.

	Insufficient privileges to open the <*share*> shared resource on the <*host*> station	The `ADMIN$` resource does not exist on a remote station, or you do not have sufficient privileges to open it.
Sending files to the <*host*> station	The <*path*> path in the <*share*> shared resource is not found on the <*host*> station	The `ADMIN$/TEMP` directory does not exist.
	Unable to create the <*path*> temporary folder in the <*share*> shared resource on the <*host*> station	Unable to create the temporary directory in `ADMIN$/TEMP`, for example, due to insufficient write permissions.
	Unable to delete the <*path*> temporary folder in the <*share*> shared resource on the <*host*> station	Unable to delete the temporary directory in `ADMIN$/TEMP` after the procedure is complete. For example, if the service was force stopped, or someone opened a file in that directory.

Step	Error	Reason
	Unable to open the <*path*> file for reading on Dr.Web Server Unable to read the <*path*> file on Dr.Web Server	The installer file was not found on Dr.Web Server, or insufficient permissions are set on the installer file.
	Unable to open the <*path*> file for writing in the <*share*> shared folder on the <*host*> station Unable to write the <*path*> file in the <*share*> shared folder on the <*host*> station	Insufficient privileges to read/write the appropriate files or directories.
Creating service on the <*host*> station	Error connecting to the server service (RPC srvsvc) on the <*host*> station	Remote management of services is not available.
	Error connecting to SCM on the <*host*> station Unable to create the service on the <*host*> station Unable to start the service on the <*host*> station Unable to stop the service on the <*host*> station Unable to delete the service on the <*host*> station	Insufficient privileges to control services.
Running service on the <*host*> station	Unable to get the service status on the <*host*> station	Possible SCM error.
	Installation timed out on the <*host*> station	The installer failed to install Dr.Web Agent within the specified time. Possible reasons: slow channel between the station and Dr.Web Server, insufficient time to download necessary data.
	Unable to get the local path to the <*share*> shared resource on the <*host*> station	Unable to locate the path to the `ADMIN$` resource on the station.
	The service has failed on the <*host*> station. Completion status: <*share*>. Error code: <*rc*>.	Dr.Web Agent installer errors.

Resolving BFE Errors during Dr.Web Agent for Windows Installation

Some components of Dr.Web Anti-virus for Windows require the Base Filtering Engine (BFE) service to be running. If this service is missing or corrupted, Dr.Web Agent for Windows cannot be installed. Corruption or absence of the BFE service can indicate security threats on the station.

If the attempt to install Dr.Web Agent for Windows fails with a BFE error, please perform the following actions:

1. Scan the station system using the CureNet! utility provided by Doctor Web company.

   You can request the demo version of the utility (which is used to diagnose, but not to cure) here: https://download.drweb.com/curenet/.

   You can read the terms of use and the cost of the full version of the utility here: https://estore.drweb.com/utilities/.
2. Enable or restart the BFE service. If you cannot restart the BFE service or it is missing from the list of services, please contact Microsoft technical support.
3. Run the Dr.Web Agent for Windows installer and perform the installation according to the general procedure given in the **Installation Manual**.

   If the problem persists, please contact Doctor Web technical support.

Technical Support

If you have a problem installing or using Doctor Web products, please try the following before contacting technical support:

1. Download and review the latest manuals and guides at https://download.drweb.com/doc/.
2. See the Frequently Asked Questions section at https://support.drweb.com/show_faq/.
3. Browse the official Doctor Web forum at https://forum.drweb.com/.

If you haven't found a solution to your problem, you can fill out a web form in the appropriate section at https://support.drweb.com/ to request direct assistance from Doctor Web technical support specialists.

For information on regional and international offices of Doctor Web, please visit the official website at https://company.drweb.com/contacts/.