

Parallels[®] Plesk Panel

Administrator Guide and Reference

Backup and Restore CLI

Parallels Plesk Panel 9.5

Contents

Preface	3
About This Guide	3
Typographical Conventions	3
Feedback	4
Plesk Backup and Restore Overview	5
Backup Objects: Hierarchy and Volume	7
Backup Logical Structure	10
Specifying Location of Temporary Directory	14
Performing Backup	15
Defining Data for Backup	17
Examples.....	21
Defining Properties of Files That Compose Backup	24
Setting Up Backup Export	26
Defining How the Backup Process is Performed	28
Checking Free Disk Space Before Starting the Backup Process	29
Performing Restore	30
Defining Objects for Restore	31
Defining How the Restore Process is Performed.....	34
Conflict Resolution Rules and Policies.....	35
Reference	38
pleskbackup Commands and Options Reference.....	38
pleskrestore Commands and Options	42
Custom Conflict Resolutions	44
Conflict Description Messages.....	44
Resolutions Description Format.....	45
Samples of Policy Description.....	54
Samples of Conflict Resolution With Rules.....	55

Preface

In this section:

About This Guide.....	3
Typographical Conventions	3
Feedback	4

About This Guide

This guide is intended for administrators of servers with Parallels Plesk Panel who wish to perform Plesk backup and restore operations via command line interface (using the `pleskbackup` and `pleskrestore` utilities).

Typographical Conventions

Before you start using this guide, it is important to understand the documentation conventions used in it.

The following kinds of formatting in the text identify special information.

Formatting convention	Type of Information	Example
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Go to the System tab.
	Titles of chapters, sections, and subsections.	Read the Basic Administration chapter.
<i>Italics</i>	Used to emphasize the importance of a point, to introduce a term or to designate a command line placeholder, which is to be replaced with a real name or value.	The system supports the so called <i>wildcard character</i> search.
Monospace	The names of commands, files, and directories.	The license file is located in the <code>http://docs/common/licenses</code> directory.

Formatting convention	Type of Information	Example
Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	<pre># ls -al /files total 14470</pre>
Preformatted Bold	What you type, contrasted with on-screen computer output.	<pre># cd /root/rpms/php</pre>
CAPITALS	Names of keys on the keyboard.	SHIFT, CTRL, ALT
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another.	CTRL+P, ALT+F4

Feedback

If you have found an error in this guide, or if you have suggestions or ideas on how to improve this guide, please send your feedback using the online form at <http://www.parallels.com/en/support/usersdoc/>. Please include in your report the guide's title, chapter and section titles, and the fragment of text in which you have found an error.

Plesk Backup and Restore Overview

Backup and restoration are among the most critical processes of hosting environment management. Parallels Plesk Panel provides graphical user interface (GUI) and command line interface (CLI) solutions to perform backup and restore tasks.

The main features of the command-line backup/restore solution are the following (marked with asterisk (*) present in GUI, too):

- *Flexible backup and restore*
Supported are not only full backups and restores, but also partial which allow to choose the level of Plesk objects that should be backed up or restored (resellers, clients, domains), and, moreover, to select almost any specific set of the objects.
- *Even more flexible backup: only configuration**
Every backup object is understood as composed of configuration and content: configuration is a set of all object properties and settings, and content is the object's binary data. It is possible to create "light-weight" backups which include only configuration of the selected objects.
- *Mail-oriented backups*
One may create a backup which contains only content and configuration of mail system owned by the selected objects.
- *Web site oriented backups*
One may create a backup which contains only content and configuration of Web sites on domains owned by the selected objects.
- *Various options of storing backups*
Plesk provides a dedicated storage - Backup Repository*, in addition to this, a backup can be exported as a single file directly to a specified FTP server or any location on file system.
- *Calculating domain backups in disk usage statistics**
Plesk disk space usage statistics can be set up so that the domain backup size is counted in overall domain disk usage and in overall disk usage of the domain owner, irrelevant to where the backup is stored.
- *Prudent restore**
Restore adds the data from backup to that existing in destination Plesk, not replaces it.
- *Advanced restore conflicts resolution*
Plesk restore, when encounters conflicts between backup data and data in Plesk that cannot be automatically resolved, returns highly-detailed description of such conflicts. Which allows admin to accurately perform restoration with applying comprehensive set of conflict resolutions

In this chapter:

Backup Objects: Hierarchy and Volume	7
Backup Logical Structure.....	10
Specifying Location of Temporary Directory	14

Backup Objects: Hierarchy and Volume

Plesk provides opportunities for backing up and restoring nearly all hosting data, which includes its major objects: Administrator account, reseller accounts, client accounts, domain accounts, mail accounts, databases, Web sites, and subdomains. These backup objects are organized into a strict hierarchy where parent object is always an owner of its children. The hierarchy is as follows:

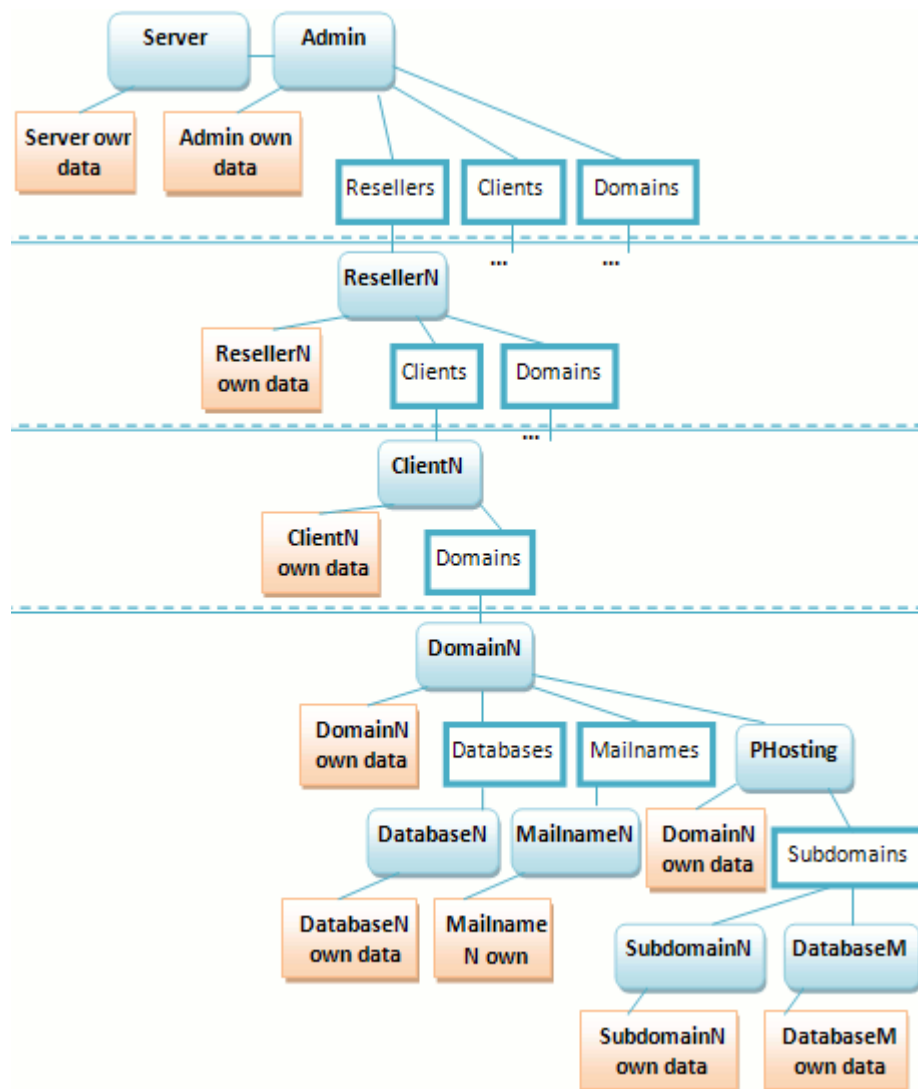


Figure 1: Backup objects hierarchy

As you can see on the diagram, Plesk backup objects are levelled into 4 levels: *server*, *resellers*, *clients* and *domains* levels. The levels are such that a higher level includes objects on the lower levels but a lower level is completely separated from the higher objects.

You can create either full or partial backup. Full backup is the highest-level backup, it includes all Plesk data: *server*, *admin* and all descendant backup objects. Partial backup includes only backup objects you need, of any of the levels. For information on available options when creating partial backup, refer to the **Defining Data for Backup** section (on page 17).

Note: Full backups do not include Plesk modules data.

Restoring a backup, in turn, can also be either full or partial. Full restore revives all data contained in a backup, and partial revives a part. For information on available options when restoring data from backup, refer to the **Defining Objects for Restore** section (on page 31).

As mentioned, each backup object has own data. These data consist of backup object *configuration* and *content*:

- Configuration defines properties of the backup object *and its descendants*.
- Content contains binary data related *only* to the backup object (database backups, mail attachments, etc).

This means that, for instance, client configuration includes configuration of the domain he/she owns, but their content is completely independent.

This table shows what data (configuration and content) is related to each backup object.

Backup Object Type	Own Configuration	Own Content
<i>server</i>	Settings of server-level services (mail and database servers, SSO, application vault, spam filters and antivirus), server preferences, Sitebuilder configuration, Plesk account templates, license keys, certificates, interface preferences, Plesk billing settings.	License keys, custom button icons, Plesk skins, Plesk locales, Web application packages.
<i>admin</i>	Personal Plesk Administrator information.	
<i>reseller</i>	Personal reseller information, limits and permissions on resources, IP pool and Web application pool configuration, personal domain and client templates settings, custom buttons settings.	Virtual host templates, custom button icons.
<i>client</i>	Personal client information, limits and permissions on resources, IP pool and site application pool configuration, personal domain and client templates settings, custom buttons settings.	Virtual host templates, custom button icons.






Backup Object Type	Own Configuration	Own Content
<i>domain</i>	Resource usage limits, permissions, domain-level service settings (IP address, mail system, mailing lists, DNS, tomcat applications, traffic statistics), custom button and domain alias settings, domain administrator personal information, SSL certificates, hosting type settings.	Mailing lists, tomcat applications, custom button icons.
<i>mailuser</i>	Personal mail user information, permissions, mailbox settings, mail aliases, forwarding settings, mailgroup settings (only for mailgroup accounts), autoresponders configuration, addressbook (horde turba), SpamAssassin/drweb settings, custom buttons settings.	Mail box content, autoresponder attachments, SpamAssassin files, custom button icons.
<i>database</i>	Database server settings and database user settings.	Database dump.
<i>phosting</i>	System user account settings, scripting settings, Web applications settings, Frontpage user credentials, log rotation settings, anonymous FTP settings, list of web-protected site locations, list of domain web users, subdomain settings, web statistics settings, hotlinking protection settings, performance and shared SSL settings.	Virtual host content, content of installed Web applications, virtual directories content, web user home directories content.
<i>subdomain</i>	System user account information, scripting options, Web applications list, hotlink protection settings, protected directory settings, and shared SSL settings.	Content of virtual “sub-host”, content of installed Web applications, custom button icons.











Backup Logical Structure




By default, all Plesk backups are created in Plesk backup repository located on Plesk server:

- in Plesk for Linux/Unix, repository location is specified by the `DUMP_D` variable defined in the `/etc/psa/psa.conf` configuration file
- in Plesk for Windows, repository is located in the `%plesk_dir%\Backup\` folder, where `%plesk_dir%` is environment variable specifying directory where Plesk is installed (if installed to default locations, it is “`C:\Program Files\Parallels\Plesk\`”)

The repository is structured as follows, starting with the content of repository root folder (we omit auxiliary files and folders which are irrelevant for backing up/restoring Plesk data using `pleskbackup/pleskrestore` utilities):

 <code><info>.xml</code>	Metadata files of full and server-level backups, one per backup, describe configuration and content of <i>server</i> , <i>admin</i> , and all their descendants.
 <code><content>.<zip tar tgz></code>	Archives with content of <i>server</i> and <i>admin</i> .
 <code>clients/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>clients</i> owned by <i>admin</i> or having no owner ▪ objects owned by the <i>clients</i> Organization of the directory is the same as that of <code><repository>/resellers/<reseller ID>/clients/</code> .
 <code>domains/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>domains</i> owned by <i>admin</i> or having no owner ▪ objects owned by the <i>domains</i> Organization of the directory is the same as that of <code><repository>/resellers/<reseller ID>/clients/<client ID>/domains</code> .
 <code>resellers/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>resellers</i> ▪ objects owned by the <i>resellers</i>

	<code><reseller ID>/</code>	Directories containing backup data of particular <i>resellers</i> , one reseller per directory, and the objects owned by them. The <i>reseller ID</i> stands for the reseller login name.
	<code><info>.xml</code>	Metadata files of the <i>reseller</i> backups, one file per backup, describe configuration and content of the <i>reseller</i> and the objects she owns.
	<code><content>.<zip tar tgz></code>	Archives with the <i>reseller</i> content.
	<code>domains/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>domains</i> owned by the <i>reseller</i> ▪ objects owned by the <i>domains</i> Organization of the directory is the same as that of <code><repository>/resellers/<reseller ID>/clients/<client ID>/domains/</code> .
	<code>clients/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>clients</i> owned by the <i>reseller</i> ▪ objects owned by the <i>clients</i>
	<code><client ID>/</code>	Directories containing backup data of particular <i>clients</i> , one client per directory, and the objects owned by them. The <i>client ID</i> stands for the client login name.
	<code><info>.xml</code>	Metadata files of the <i>client</i> backups, one file per backup, describe configuration and content of the <i>client</i> and the objects he owns.
	<code><content>.<zip tar tgz></code>	Archives with the <i>client</i> content.
	<code>domains/</code>	Directory containing the following backup data: <ul style="list-style-type: none"> ▪ <i>domains</i> owned by the <i>client</i> ▪ objects owned by the <i>domains</i>
	<code><international domain name> <domain ID>/</code>	Directories containing backup data of particular <i>domains</i> , one domain per directory, and the objects owned by them. The domain ID is omitted if the domain IDN is less than 47 symbols.

		<code><info>.xml</code>	Metadata files of the <i>domain</i> backups, one file per backup, describe configuration and content of the <i>domain</i> and the objects it owns.
	 	<code><content></code>	Other files and folders which contain domain contents, and its children contents and configurations.

Files of each backup are placed in the repository folders according to the described structure.

If a partial backup is created, its files will be placed according to the place the backup objects have in the hierarchy. For example, if backing up domain `example.com` owned by reseller `JaneDoe`, its files will be located in the `<repository root directory>/resellers/JaneDoe/domains/example.com/` folder. If backing up reseller `JohnDoe` who owns a domain `joe.info` and has one client `DukeNukem` who owns domain `sample.org`, the backup files will be located in the following folders:

1. `<repository root directory>/resellers/JohnDoe/`
2. `<repository root directory>/resellers/JohnDoe/domains/joe.info/`
3. `<repository root directory>/resellers/JohnDoe/clients/DukeNukem/`
4. `<repository root directory>/resellers/JohnDoe/clients/DukeNukem/domains/sample.org/`

To distinguish files belonging to different backups of the same object, specific prefix and suffix are added to the file names:

- prefix `backup` is added by default, and, if you like, you can change it to your own on a per-backup basis (details (on page 24))
- suffix designating the backup creation date is always added to each backup file, the date format is `<yymdddhmm>`. For example, files of backup created on March 8, 2009, 1:30 am will have suffix `0903080130`.

Plesk is capable of exporting backup as a single file (`.tgz` in Linux/Unix and `.zip` in Windows). Each archive has the same structure as the repository, the only difference is that there is only one `<info>.xml` file on each level.

In case a partial backup is exported, the resulting file structure is reduced from the top so that the highest level corresponds to the level of the highest backup object. For example, if a single client (called, say, `SandyLee`) backup is exported, the resulting file will have the following structure:

```
zip {
  ▪ <sandy lee info>.xml
  ▪ n*<content>.zip
  ▪ domains/
    ▪ domain1/
      ▪ ...
    ▪ domainN/
    ▪ ...
}
```

Specifying Location of Temporary Directory

By default, the following directories on the server are used for storing temporary files during Plesk hosting data backup and restoration:

- `/tmp` on Linux and FreeBSD systems.
- `<operating_system_installation_directory>\PrivateTemp` on Windows systems.

If you have large amounts of data to back up and restore and are not sure that there is enough free disk space in the default temporary directory location, then you can specify another location for storing temporary files.

This is done by setting environment variables in the operating system of the server.

➤ ***To set a new location of the temporary directory on Linux or FreeBSD servers:***

1. Connect to the server over SSH.
2. Open for editing the file `/etc/psa/psa.conf`.
3. Locate the variable `DUMP_TMP_D` and replace the `/tmp` value with a path to the new location of the temporary directory.

The new directory must be present in the file system and writing to this directory must be allowed for all users.

4. Save the file.

➤ ***To set a new location of the temporary directory on Windows servers:***

1. Connect to the server as administrator.
2. Click **Start**, and then click **Run**.
3. In the **Run** dialog box, type `regedit`, and then click **OK**. This will start Registry Editor.
4. Locate the `HKEY_LOCAL_MACHINE\SOFTWARE\PLESK\PSA Config\Config\` and specify the new location of temporary directory in the `DumpTempDir` variable.

The default value of the `DumpTempDir` variable is `%plesk_dir%\PrivateTemp\`.

5. Quit Registry Editor.

Performing Backup

To perform backup of Plesk hosting data, you need to execute the `pleskbackup` utility command composed so that it does the following:

1. defines Plesk data which is going to be backed up
2. defines the way of how the backup process will be performed
3. defines properties of the files that will be contained in backup
4. defines options for exporting backup as a single file

Note: Only first component is obligatory, others are optional.

The following sub-sections explain each component meaning and implementation in detail.

The `pleskbackup` utility location is:

- in Plesk for Linux/Unix: `/usr/local/psa/bin/pleskbackup`
- in Plesk for Windows: `%plesk_dir%\bin\pleskbackup`
where `%plesk_dir%` is environment variable for Plesk installation directory. By default, it is "C:\Program Files\Parallels\Plesk"

To see a complete list of the `pleskbackup` commands and options, refer to the **pleskbackup Reference** (on page 38).

If the command execution succeeds, backup is created in the default server backups location or exported to a file in case exporting options were specified. (For details, refer to the **Setting Up Backup Export** section (on page 26).) If the command execution fails, backup is not created.

In case domain backup files are set to be included in the statistics on disk space usage (in Plesk GUI: **Server Settings** > **System Preferences**) and backups are stored in Plesk backup repository, Plesk works as follows:

- Disk space occupied by a domain backup files is count in the overall domain disk space usage.
- Overall reseller's/client's disk space usage includes disk space occupied by backups of all domains that the reseller/client possesses, stored in the Plesk backup repository.

In this chapter:

Defining Data for Backup..... 17
Defining Properties of Files That Compose Backup..... 24
Setting Up Backup Export 26
Defining How the Backup Process is Performed 28
Checking Free Disk Space Before Starting the Backup Process 29

Defining Data for Backup

Defining data that should be backed up includes the following:

1. Defining backup level and, unless it is server level, optionally, selecting which resellers|clients|domains should be backed up.
2. (optional) Defining which resellers|clients|domains should be excluded from the backup.
3. (optional) Restricting backup to either only mail or only physical hosting, and only to configuration.
4. (optional) Defining that log files are excluded from backup.

Generally speaking, the data that can be backed up with one call of the `pleskbackup` utility is represented by any single cell of the following table.

		(all)		Only physical hosting option: --only-hosting		Only mail option: --only-mail	
		(all)	Only config option: -c	(all)	Only config option: -c	(all)	Only config option: -c
Server command: server	(all)						
	Excluding resellers option: --exclude-reseller or - -exclude-reseller-file						
	Excluding clients option: --exclude-client or --exclude-client-file						
	Excluding domains option: --exclude-domain or --exclude-domain-file						
[Selected] Resellers command: resellers-name or resellers-id	(all)/(all selected)				Example 1		
	Excluding resellers option: --exclude-reseller or - -exclude-reseller-file				Example 1'		
	Excluding clients option: --exclude-client or --exclude-client-file						
	Excluding domains option: --exclude-domain or --exclude-domain-file						
[Selected] Clients command: clients-name or clients-id	(all)/(all selected)						
	Excluding clients option: --exclude-client or --exclude-client-file						
	Excluding domains option: --exclude-domain or --exclude-domain-file						
[Selected] Domains command: domains-name or domains-id	(all)/(all selected)					Example 2	
	Excluding domains option: --exclude-domain or --exclude-domain-file						

Example 1: With one call of `pleskbackup`, you can backup hosting data for several resellers (row 5 or 6 in the table, depending on what's more convenient: to list resellers that should be included or those excluded) and restricting the backup data to configuration of physical hosting on domains owned by the resellers or their clients (column 4 in the table).

Example 2: With one call of `pleskbackup`, you can backup mail configuration and content (column 5) for all domains existing on the server (row 12).

The rest of this section explains each option in detail. Possible command syntax and examples of commands are in the **Examples** sub-section (on page 21).

Defining backup level and selecting objects

To define backup level and select backup objects, the commands of `pleskbackup` utility are used.

If performing a selective backup, resellers, clients or domains selected for the backup should be specified by their identifiers which are either logins/names or IDs. The specification can be done in one of the following two ways:

- *Command line specification.* The backup command takes objects identifiers as arguments separated with spaces.
- *File specification.* The backup command takes the `--from-file` option which specifies the file where the identifiers of objects are listed. The file must be in plain text format, and object identifiers are separated by line breaks (i.e., one identifier per line).

Note: If a command contains both specifications, file specification is used and the command line specification is ignored.

Command syntax and samples (on page 21).

Defining which objects should be excluded

Objects that should be excluded from backup are specified by their logins (reseller, client accounts) or names (domain accounts). The specification can be done as follows:

- *Command line specification.* The backup command takes objects identifiers as values of the `--exclude-<reseller|client|domain>` option separated by commas.
- *File specification.* The backup command takes the objects identifiers from the file specified by the `--exclude-<reseller|client|domain>-file` option. The file must be in plain text format, and object identifiers are separated by line breaks (i.e., one identifier per line).

Note: It is acceptable to use both specifications in one command. In such case, all specified objects are excluded from backup.

Command syntax and samples (on page 22).

Restricting backup to only mail or only physical hosting, and to only configuration

The amount of backup data can be further narrowed to backing up either mail or physical hosting content and configuration by using the `—only-mail` or `—only-hosting` options, respectively.

Specifying the `—only-hosting` option results in backing up only Web-site-specific data which includes the following, for each domain with physical hosting:

- Web site content (including protected directories, Web users, MIME types)
- Web hosting configuration (including settings of anonymous FTP, log rotation, hotlink protection, shared SSL, web users)
- installed site applications
- databases
- subdomains

Specifying the `—only-mail` option results in backing up only mail-specific data which includes the following:

- if used for the partial backup, for each domain included in backup:
 - configuration of domain-level mail system (including domain keys)
 - mail accounts
 - mailing lists
- if used for the full backup, *in addition to previous*:
 - SPF spam protection configuration
 - RBL protection settings
 - ACL white and black list configurations

The amount of backup data can also be narrowed in another way: by specifying that only configurations of the selected objects should be backed up. The specification is done by using the `—only-configuration` option.

Such backups are useful when the objects content is backed up by a third-party system.

Command syntax and samples (on page 23).

Excluding log files from back up

In case Plesk log files related to the hosted objects are not required to be backed up, they can be excluded from the backup by using the `—skip-logs` option.

Command syntax and samples (on page 23).

In this section:

Examples 21

Examples

Defining backup level and selecting objects

➤ *To back up the whole Plesk server data:*

```
pleskbackup—server
```

To back up all reseller|client|domain accounts: `pleskbackup --<resellers|clients|domains>--<name|id>`

For example, to back up all client accounts:

```
pleskbackup—clients—name
```

or

```
pleskbackup—clients—id
```

➤ *To back up several reseller|client|domain accounts defined in command line:*

```
pleskbackup --<resellers|clients|domains>--<name|id> [  
<identifier1> [  
<identifier2> ... [<identifier n>]]
```

For example, to back up three resellers defined in the command line:

```
pleskbackup—resellers—name johndoe janedoe josephine
```

if under Linux/Unix, and

```
pleskbackup—resellers—name "johndoe janedoe josephine"
```

if under Windows

To back up several reseller|client|domain accounts listed in file: `pleskbackup --<resellers|clients|domains>--<name|id> --from-file=<file>`

For example,

```
pleskbackup—resellers—name—from-file=/usr/local/backup-lists/j.txt
```

if under Linux/Unix, and

```
pleskbackup—resellers—name—from-file="E:\backup lists\j.txt"
```

if under Windows

Defining which objects should be excluded

- **To back up all reseller accounts except for several selected resellers:**

```
pleskbackup—resellers-name—exclude-reseller=<login1>,<login2>[,<login n>]
```

or

```
pleskbackup—resellers-name—exclude-reseller-file=<file>
```

For example,

```
pleskbackup—resellers-name—exclude-reseller=johndoe,janedoe
```

or

```
pleskbackup—resellers-name—exclude-reseller-file=/usr/local/backup-  
lists/j.txt
```

if under Linux/Unix, and

```
pleskbackup—resellers-name—exclude-reseller-file="E:\backup  
lists\j.txt"
```

if under Windows

- **To back up a selected reseller without several domains belonging to him or her, or his or her clients:**

```
pleskbackup resellers-name <login> --exclude-domain=<name1>,<name2>,<name n>
```

or

```
pleskbackup resellers-name <login> --exclude-domain-file=<file>
```

For example,

```
pleskbackup—resellers-name johndoe—exclude-  
domain=example.com,example.net,example.org
```

or

```
pleskbackup—resellers-name johndoe—exclude-domain-  
file=/usr/local/backup-lists/excl-example-domains.txt
```

if under Linux/Unix, and

```
pleskbackup—resellers-name johndoe—exclude-domain-file="D:\backup-  
lists\excl-example-domains.txt"
```

if under Windows

Narrowing backup data to only hosting, mail, configuration, skipping log files

- **To back up the whole Plesk server configuration without log files:**

```
pleskbackup-server -c-skip-logs
```

To back up mail configuration on domains belonging to a client: `pleskbackup-clients-<name|id> <name|id> --only-mail-configuration`

For example,

```
pleskbackup-clients-id 42 --only-mail-configuration
```

- **To back up web sites content and web hosting configuration on domains belonging to all resellers:**

```
pleskbackup-resellers-id-only-hosting
```

Defining Properties of Files That Compose Backup

Defining properties of the files that will be contained in backup includes the following:

1. Defining that archives with backup object contents should not be compressed.
2. Defining that a prefix should be added to names of the backup files.
3. Defining that backup files should be split into parts of the specified size.

Defining that archives with backup object contents should not be compressed

By default, Plesk backup archives the backup objects content to compressed archives (.tar, .tgz or .zip) in order to save disk space when the backup is stored. However, restoring backups that contain compressed archives requires almost two times more disk space than restoring those with uncompressed files. In case when disk space while the restoring procedure is more critical to you, you may want to create your backups without compression. To do so, use the `-z|--no-gzip` option in your backup command.

Defining that a prefix should be added to names of the backup files

In order to better distinguish files that were created during one backup session from another, `pleskbackup` adds a prefix to backup file name. By default, it is `backup`, so every backup file name looks like `backup_<file-name>.<ext>`. The prefix in names of the files that compose a particular backup can be customized by using the `prefix` option. The option's value will be added as a prefix to names of files of the created backup.

For example, to create a backup of the server mail configuration so that all files in backup have prefix `mail-friday`:

```
pleskbackup-server-only-mail-configuration-prefix="friday"
```

Defining that backup files should be split into parts of the specified size

The `pleskbackup` utility is capable of splitting backup files into parts of a particular size, which is vitally useful in cases when the file size is critical. Such cases can be, for example, the following:

- if backups are burnt to DVDs, file size should not exceed approximately 4 Gbytes
- if backups are stored on the FAT32 file system, file size should not exceed approximately 4 Gbytes

- if backups are stored on FTP, FTP server may have its own restrictions on the size of a single file transferred to the server

To make `pleskbackup` split the backup files to parts of a particular size, use the `-s|--split` option and specify the required size as the option value. For details on the format of size specification, refer to the `pleskbackup` reference (on page 38). The default value used by `pleskbackup` if no custom size is specified is 2 Gbytes. The utility numbers file parts created as a result of split by adding numerical suffixes to the file names starting from `.1`.

For example, to back up Web hosting on a domain splitting backup files into parts of no more than 700 Mbytes:

```
pleskbackup-domains-name example.com-only-hosting-split=700M
```

Setting Up Backup Export

By default, `pleskbackup` stores backups in Plesk backup repository located on the Plesk server (`/var/lib/psa/dumps/` folder in Plesk for Linux/Unix and `%plesk_dir%\Backup\` in Plesk for Windows).

Plesk is capable of exporting the created backup as a single file (`.tar` on Unix and `.zip` on Windows) in one of the following ways:

- to `stdout`
- to local file system
- to FTP server

To export backup as a single file, use the `--output-file` option. Particular export mode requires specific option values.

Important: After a backup is exported, `pleskbackup` removes it from the Plesk backup repository.

The exported file can also be created not compressed and/or split in parts of a particular size, just as the files composing backup in repository (details (on page 24)).

Exporting to stdout

To export a backup as file to `stdout`, use the `--output-file` option with a minus sign as its value.

For example, to create backup of a domain with ID 1 and export it to `stdout`:

```
pleskbackup-domains-id 1 --output-file -
```

Exporting to local file system

To export a backup as a file to local file system, use the `--output-file` option with a `<full-path-to-file>/<file-name>` value.

For example, to create backup of a domain with ID 1 and export it to the file `domain1.tgz` located at `/usr/local/irregular-backups/` folder:

```
pleskbackup-domains-id 1 --output-file=/usr/local/irregular-backups/domain1.tgz
```

Exporting to FTP server

To export a backup as a file to a FTP server, use either of the following options:

- `--output-file=ftp://<login>:<password>@<server>/<filepath>`
- `--output-file=ftp://<server>/<filepath> --ftp-login=<ftp login> --ftp-password=<ftp password>`

You may want to use passive mode FTP connection in case a firewall prevents the export. For this, use the `--ftp-passive-mode` option.

For example, to create backup of a domain with ID 1 and export it to FTP server `example.com` to the `storage/backups/` directory, using `johndoe` as login and `jjFh6gsm` as password:

```
pleskbackup-domains-id 1 --output-  
file=ftp://johndoe:jjFh6gsm@example.com/storage/backups
```

Or

```
pleskbackup-domains-id 1 --output-  
file=ftp://example.com/storage/backups-ftp-login=johndoe-ftp-  
password=jjFh6gsm
```

Defining How the Backup Process is Performed

Define the way how the backup process will be performed includes the following:

1. Defining backup verbosity
2. Suspending the domains being backed up

Defining level of backup verbosity

Verbose mode of backup process is defined by the `-v` option. The option behavior differs on Linux/Unix and Windows systems.

On Linux/Unix, the level of backup process verbosity is defined by a number of the `-v` options used in the backup command. Depending on the number of used `-v` options, we distinguish the following three levels:

1. *0 to 2 -v used.* The minimum level, only general errors are displayed, like, for example, syntax errors (no or wrong command specified, invalid input parameters), runtime errors and unhandled exceptions, low disk space for backup and so on.
2. *3 -v used.* Normal verbosity level, includes general errors (see above) and information on backup stages (e.g., `09:18:40 INFO Create backup task description`).
3. *4 to 5 -v used.* The maximum verbosity level, in addition to the previous, includes debug information and response/request messages to the internal backup utility.

Note: `pleskbackup` outputs information on its execution to `stdout` only. If you want to have the backup log saved, redirect the utility output to a file with standard command line means.

➤ **To run a task on creating a complete server backup with maximum level of verbosity:**

```
pleskbackup-server -vvvvv
```

On Windows, only two verbosity levels are supported, depending on whether the `-v` option is used in the backup command or not:

1. *No -v option used.* The minimum level, only general errors are displayed, like, for example, syntax errors (no or wrong command specified, invalid input parameters), runtime errors and unhandled exceptions, low disk space for backup and so on.
2. *The -v option used.* Sets up verbosity level equal to 4: in addition to the previous, includes debug information and response/request messages to the internal backup utility.

Suspending domains

In case your backup is going to include domains, it is recommended to use the—`suspend` option to suspend the domains during the backup process. Doing this ensures from errors in backup files that may be caused by changes done to the domain configuration and/or content during the backup.

The suspension is made up to be as short as possible: each domain is suspended only for the time it is being backed up, the domain is started automatically as soon as the domain data is processed.

Checking Free Disk Space Before Starting the Backup Process

It is recommended that you make sure there is enough free disk space before starting the backup process. By default, the amount of free disk space is not checked. If the backup task is started and there is not enough disk space, the task is stalled in the GUI and processes.

To enable the free disk space checking, open for editing the `pmmcli` configuration file located on Parallels Plesk Panel server at

```
%PLESK_DIR%/admin/share/pmmcli/pmmcli-rc and set the  
CHECK_BACKUP_DISK_SPACE option value to 1:
```

CHECK_BACKUP_DISK_SPACE 1

When this option is turned on, free disk space is checked prior to starting the backup process. The free disk space is checked only for the local repository on Plesk server, repository location is specified by the `DUMP_D` variable defined in the `/etc/psa/psa.conf` configuration file. If there is not enough disk space, the backup process is not started and the following error message is displayed:

Not enough free disk space to backup selected objects. At least <estimated-backup-size> free disk space is required.

Note: The free disc space will be checked only before starting the backup process. Thus, this option will not be effective, if the free disc space is occupied during the backup process by other processes, such as a simultaneous upload.

Note: The backup size estimation does not consider possible content compression. Actual size of a backup is usually less than its estimated size.

Performing Restore

To perform a restore of Plesk hosting data, you should execute the `pleskrestore` utility command composed so that it does the following:

1. defining Plesk objects to be restored
2. defining the way of how the restore process will be performed
3. defining conflict resolution rules and policies

The following sub-sections explain each component in detail.

The `pleskrestore` utility location:

- in Plesk for Linux/Unix: `/usr/local/psa/bin/pleskrestore`
- in Plesk for Windows: `%plesk_dir%\bin\pleskrestore`
where `%plesk_dir%` is environment variable for Plesk installation directory. By default, it is `"C:\Program Files\Parallels\Plesk\"`.

To see a list of the `pleskrestore` commands and options, refer to the **pleskrestore Reference** (on page 42).

In this chapter:

Defining Objects for Restore	31
Defining How the Restore Process is Performed.....	34
Conflict Resolution Rules and Policies	35

Defining Objects for Restore

Defining objects for restore includes the following:

1. Defining target backup file
2. Defining the level of restored objects
3. Applying filter on the specified level

Generally speaking, the data that can be restored with one call of the `pleskrestore` utility is represented by any blue cell of the following table.

		Restore level (option: -level)						
		server	resellers		clients		domains	
				selected option: -filter		selected option: -filter		selected option: -filter
Backup file	<server>.<xml zip tar>	Full restore	All resellers	Selected resellers	All clients owned by admin	Selected clients owned by admin	All domains owned by admin	Selected domains owned by admin
	<reseller>.<xml zip tar>		The reseller complete restore		All clients owned by the reseller	Selected clients owned by the reseller	All domains owned by the reseller	Selected domains owned by the reseller
	<client>.<xml zip tar>				The client complete restore		All domains owned by the client	Selected domains owned by the client
	<domain>.<xml zip tar>						The domain complete restore	

Defining Target Backup File

Target backup file defined for restoration can be of one of the following types:

- `<info>.xml` - backup metadata file, in case of restoring from backup located in Plesk repository
- `<backup>.<zip|tar>` - archived backup file, in case of restoring from exported backup

For example, to restore the whole server backup, you choose a `<backup repository root>/<server>.xml` file, or an exported server backup file. To restore a client belonging to a reseller, you choose a `<backup repository root>/resellers/<reseller ID>/clients/<client ID>/<client>.xml` file.

Defining level of restored objects

Defining level of restored objects allows you to narrow the amount of restored data according to your needs. For example, you may want to restore only domains which belong to a client or a reseller, skipping the client's/reseller's own data and objects belonging to him different from domains.

To define the level of restored objects, use the `-level` option with appropriate value. The option is required, so in cases when you don't need any narrowing but just restoring all data from a backup, define the level equal to the level of file.

➤ **To restore entire server:**

```
pleskrestore restore <backup repository root>/<server>.xml -level server
```

Note: When the whole server backup is restored, license keys are not restored by default. To restore license keys along with other server content, use the `-license` option in your restore command.

➤ **To restore entire server with license keys:**

```
pleskrestore-restore <backup repository root>/<server>.xml -level server -license
```

➤ **To restore all domains belonging to a reseller:**

```
pleskrestore-restore <backup repository root>/resellers/<reseller ID>/<reseller>.xml -level domains
```

➤ **To restore all reseller accounts:**

```
pleskrestore-restore <backup repository root>/<server>.xml -level resellers
```

Applying filter on the specified level

To perform a more selective restore, use a filter (the `-filter` option) which selects for restore particular objects of the specified level (resellers, clients, domains). The objects are specified by their names, which are domain names, and logins for resellers and clients. The specification can be done as follows:

- *Command line specification.* The restore command takes objects identifiers as values of the `-filter` option defined in the following string:
`list:<item1>,<item2>,...,<itemN>`.
- *File specification.* The restore command takes the objects identifiers from the file specified as argument of the `-filter` option. The file must be in plain text format, and object identifiers are separated by line breaks (i.e., one identifier per line).

➤ **To restore two resellers from a server backup:**

```
pleskrestore-restore <backup repository root>/<server>.xml -level  
resellers -filter list:JohnDoe,JaneDoe
```

or

```
pleskrestore-restore <Upload directory>/<server backup name>.tar -  
level resellers -filter list:JohnDoe,JaneDoe
```

➤ **To restore two domains owned by Plesk admin:**

```
pleskrestore-restore <backup repository root>/<server>.xml -level  
domains -filter list:example.com,sample.org
```

➤ **To restore client's several domains defined in a file:**

```
pleskrestore-restore <backup repository  
root>/resellers/SandyLee/clients/JaneDow/<client>.xml -level domains -  
filter <path to the file>/restore-domains.txt
```

Defining How the Restore Process is Performed

Define the way how the restore process will be performed includes the following:

1. Defining restore verbosity
2. Suspending the domains being backed up

Defining level of restore verbosity

`pleskrestore` works in one of the following verbosity modes:

1. *Non-verbose mode*. Default mode. The minimum level, only general errors are displayed, like, for example, syntax errors (no or wrong command specified, invalid input parameters), runtime errors and unhandled exceptions, and so on.
2. *Verbose mode*. Restore runs with verbosity level 3, which includes, in addition to the previous level, deployer errors, information about conflicts (read about restore conflicts in the **Conflict Resolution Rules and Policies** section (on page 35)), and so on. Enabled by adding the `-verbose` option to the `pleskrestore` command.
3. *Debug mode*. Restore runs with verbosity level 4, the highest possible, includes the most extensive information on the restore process. Enabled by adding the `-debug` option to the `pleskrestore` command.

Suspending domains

In case you are going to restore domains, it is recommended to use the `-suspend` option to suspend the domains during the restore process. Doing this ensures from errors in the restored domains that may be caused by changes done to the domain configuration and/or content during the restoration.

The suspension is made up to be as short as possible: each domain is suspended only for the time it is being backed up, the domain is started automatically as soon as the domain data is processed.

Conflict Resolution Rules and Policies

Conflict is a situation when settings in a backup and settings in a destination Plesk are such that restoring backup objects leads to an error, or unpredictable Plesk behavior, including misbehavior.

Types of Conflicts

The restoration process can encounter several types of conflicts, which are the following:

- **Timing conflicts.** An object being restored might exist in the system and its last modification date might be more recent than the date of backup. Or an object could be deleted from the system later than the backup was created.
- **Resource usage conflicts.** There are two groups of resource usage conflicts:
 - **Common resource usage conflict:** The total amount of measurable resources after restoration might appear to be over the limits for this particular user (e.g., disk space limit).
 - **Unique resource usage conflict:** An object being restored requires a unique resource which is already used by another object in the system or does not exist (e.g., domain).
- **Configuration conflicts.** It might happen that configuration being restored is not enabled on the destination server. Two types of cases can happen here:
 - Configuration options are not enabled for the domain.
 - Required configuration options are not available (e.g., site applications are not available for the client, database server is not configured on the host, IP address is not in the client's IP pool, etc.)

Conflict Resolutions

The following types of conflicts resolutions are possible:

- **Overwrite.** Means that all objects will be restored from the backup files regardless of their current presence in the system. Overwrite works as follows:
 - If an object/setting from backup does not exist in Plesk, it is created.
 - If an object/setting from backup exists in Plesk, it replaces the existing.
 - If an object/setting exists in Plesk but is missed in backup, the existing remains.
- **Proceed with current.** Means that objects which currently present in the system won't be affected by the restoration process. The restoration process will move to the objects belonging to that one, not touching the object itself.
- **Do not restore.** Means that the objects which currently present in the system or were deleted after the backup won't be restored together with the lower level objects belonging to it.

- **Automatic.** Means that configuration option that should be enabled for domain is enabled automatically.
- **Overuse.** Means that objects are restored with the resources overuse. Can be applied only to objects that belong to a reseller who works in the oversell mode.
- **Rename.** Means that unique resources for the restored domain are reassigned with the specified, existing in the system (mapping).

Conflict Resolution Policies and Rules

Depending on the scope of a conflict resolution, we distinguish conflict resolution *rules* and *policies*:

- Rule defines the way of how a specific single conflict should be resolved.
- Policy defines the way of how all conflicts of a particular type should be resolved.

Conflicts Resolving Mechanism: Default Policies, Custom Policies, and Rules

Plesk restore brings a set of default, hard-coded conflict resolution policies, which are as follows:

- for timing conflicts - Overwrite
- for common resource usage conflicts - Overuse
- for unique resource usage conflicts - Do Not Restore
- for configuration conflicts - Automatic

The default policies are always applied during restore and cannot be changed or overridden.

Applying default policies may resolve not all the conflicts occurred. In such cases, those who perform restore should additionally define custom rules and/or policies that resolve the remaining conflicts. Custom rules and policies are defined in an XML format as described in the **Resolutions Description Format** section (on page 45).

Simplified presentation of the conflicts resolving during restore is as follows:

1. Administrator runs `pleskrestore` with specific parameters.
2. `pleskrestore` detects the conflicts occurred and resolves them with the default policies.
3. `pleskrestore` checks if any conflicts remain unresolved.
In case all conflicts are resolved, the restoration continues.
4. `pleskrestore` stops the restoration and, if run in debug or verbose mode, returns detailed description (in XML format) of each remaining conflict.

5. Basing on the returned description of the conflicts, administrator creates a file that defines a resolution for each conflict (with rules) and/or in bulk (with custom policies).
6. Administrator runs the `pleskrestore` utility with the `—conflicts-resolution` option and the file created at the previous step as its argument.
7. `pleskrestore` detects the conflicts occurred and resolves them with the default policies.
8. `pleskrestore` gets to the remaining conflicts:
 - a `pleskrestore` applies resolution rules from the file.
 - b `pleskrestore` applies resolution policies from the file to the rest of the conflicts.
9. `pleskrestore` checks if any conflicts remain unresolved.
 - In case all conflicts are resolved, the restoration continues.
 - In case any conflicts remain unresolved, `pleskrestore` stops the restoration and, if run in debug or verbose mode, returns detailed description (in XML format) of each remaining conflict.

To have such dump restored, admin should add resolution rules for each remaining conflict to the conflict resolution file and repeat the restoration task.

Reference

This chapter contains reference materials on `pleskbackup` and `pleskrestore` utilities.

In this chapter:

<code>pleskbackup</code> Commands and Options Reference	38
<code>pleskrestore</code> Commands and Options	42
Custom Conflict Resolutions	44

`pleskbackup` Commands and Options Reference

Location

- Plesk for Linux/Unix: `/usr/local/psa/bin/pleskbackup`
- Plesk for Windows: `%plesk_dir%\bin\pleskbackup`
where `%plesk_dir%` is environment variable for Plesk installation directory. By default, it is `"C:\Program Files\Parallels\Plesk"`.

Usage

`pleskbackup <command> [<arguments>] [<options>]`

Commands

Command	Argument	Description
<code>--server</code>		Backs up whole Plesk server.

Command	Argument	Description
--resellers-name	[<login-1> <login-2> <...> <login-n>]	<p>Backs up all data for the resellers specified by logins.</p> <p>Logins should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>—from-file</code> option. In such case, resellers specified in the file are backed up and resellers specified as command arguments are ignored.</p> <p>If no logins are specified and the <code>-f</code> option is not used, all resellers are backed up.</p>
--resellers-id	[<ID1> <ID2> <...> <IDn>]	<p>Backs up all data for the resellers specified by IDs.</p> <p>IDs should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>—from-file</code> option. In such case, resellers specified in the file are backed up and resellers specified as command arguments are ignored.</p> <p>If no IDs are specified and the <code>-f</code> option is not used, all resellers are backed up.</p>
--clients-name	[<login-1> <login-2> <...> <login-n>]	<p>Backs up all data for the clients specified by logins.</p> <p>Logins should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>—from-file</code> option. In such case, clients specified in the file are backed up and clients specified as command arguments are ignored.</p> <p>If no logins are specified and the <code>-f</code> option is not used, all clients are backed up.</p>
--clients-id	[<ID1> <ID2> <...> <IDn>]	<p>Backs up all data for the clients specified by IDs.</p> <p>IDs should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>—from-file</code> option. In such case, clients specified in the file are backed up and clients specified as command arguments are ignored.</p> <p>If no IDs are specified and the <code>-f</code> option is not used, all clients are backed up.</p>
--domains-name	[<name-1> <name-2> <...> <name-n>]	<p>Backs up all data for the domains specified by names.</p> <p>Names should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>—from-file</code> option. In such case, domains specified in the file are backed up and domains specified as command arguments are ignored.</p> <p>If no names are specified and the <code>-f</code> option is not used, all domains are backed up.</p>

Command	Argument	Description
<code>--domains-id</code>	<code>[<ID1> <ID2> <...> <IDn>]</code>	<p>Backs up all data for the domains specified by IDs.</p> <p>IDs should be separated by spaces, and, if on Windows, enclosed in quotes.</p> <p>Can be used with the <code>--from-file</code> option. In such case, domains specified in the file are backed up and domains specified as command arguments are ignored.</p> <p>If no IDs are specified and the <code>-f</code> option is not used, all domains are backed up.</p>
<code>--help</code>		Displays help on the utility usage.

Exclude Options

Option	Description
<code>--exclude-reseller[=<login1>,<login2>,. ..]</code>	Skips resellers with the specified logins during backup.
<code>--exclude-reseller-file[=<file>]</code>	Skips resellers listed in the specified file during backup.
<code>--exclude-client=[<login1>,<login2>,. . .]</code>	Skips clients with the specified logins during backup.
<code>--exclude-client-file=<file></code>	Skips clients listed in the specified file during backup.
<code>--exclude-domain[=<name1>,<name2>,. . .]</code>	Skips domain with the specified names during backup.
<code>--exclude-domain-file=<file></code>	Skips domains listed in the specified file during backup.

General Options

Option	Description
<code>-v --verbose</code>	Shows more information about backup process. Multiple <code>-v</code> options increase verbosity, for the maximum verbosity level, define 5 options.
<code>-c --configuration</code>	Backs up only configurations of Plesk objects, excluding their content.
<code>-s --split[=<integer>[K M G]]</code>	<p>Splits the backup files into parts of the specified size. The parts are numbered by appending numerical suffixes starting with <code>.1</code>.</p> <p>Size is specified in Kbytes, Mbytes or Gbytes. If none is defined, then interpreted as being in bytes.</p> <p>If no argument is specified, default value of 2 Gbytes is used.</p>
<code>-z --no-gzip</code>	Sets that objects content is archived without compressing.

Option	Description
<code>--only-mail</code>	<p>Backs up only mail configuration and content.</p> <p>When used with the <code>resellers clients domains-login id</code> commands, backs up configuration of domain-level mail system, and content and configuration of mail accounts.</p> <p>When used with the <code>server</code> command, backs up also server-wide mail configuration.</p> <p>Cannot be used together with the <code>—only-hosting</code> option.</p>
<code>--only-hosting</code>	<p>Backs up only physical hosting configuration and Web site content, including site applications, databases and subdomains.</p> <p>Cannot be used together with the <code>—only-mail</code> option.</p>
<code>--suspend</code>	Suspends domains during backup operation.
<code>-f --from-file=<file></code>	<p>Backs up resellers clients domains listed in the specified file, ignoring those specified in the command line as arguments.</p> <p>The file should be in plain text format and should contain a list of resellers clients domains, one per line.</p> <p>Used only with the <code>resellers-name, resellers-id, clients-name, clients-id, domains-name, domains-id</code> commands.</p> <p>Depending on the command, resellers clients domains are listed in the file by either logins or IDs.</p>
<code>--skip-logs</code>	Sets that log files are not saved to backup.
<code>--prefix=<string></code>	<p>Adds specified prefix to the backup files names.</p> <p>Used to customize backup file name which is created with the <code>backup</code> prefix by default.</p>

FTP Options

Option	Description
<code>--ftp-login=<ftp_login></code>	Specifies FTP login that will be used for uploading backup file to the FTP server.
<code>--ftp-password=<ftp_password></code>	Specifies password that will be used for uploading backup file to the FTP server.
<code>--ftp-passive-mode</code>	Specifies that a passive mode FTP connection should be used.

Output File Option

Option	Description
<code>--output-file</code>	Exports backup as a single file to <code>stdout</code> and removes backup from Plesk repository.

Option	Description
<code>--output-file=<fullpath/></code> <code>filename></code>	Exports backup as a single file with the specified name to a local file system and removes backup from Plesk repository.
<code>--output-file=<ftp://[<login>[:<password>]@]></code> <code><server>/<filepath>></code>	Exports backup as a single file to the specified FTP server and removes backup from Plesk repository. The <code>FTP_PASSWORD</code> environment variable can be used for setting password. The <code>--ftp-login</code> and <code>--ftp-password</code> FTP options can be used for setting login and password.

pleskrestore Commands and Options

Location

- Plesk for Linux/Unix: `/usr/local/psa/bin/pleskrestore`
- Plesk for Windows: `%plesk_dir%\bin\pleskrestore`
where `%plesk_dir%` is environment variable for Plesk installation directory. By default, it is `"C:\Program Files\Parallels\Plesk\"`.

Usage

`pleskrestore <command> [<arguments>] [<options>]`

Commands

Command	Argument	Description
<code>--restore</code>	<code><backup_file></code>	Restores data from the specified backup. Requires the <code>-level</code> option.
<code>--check-backup</code>	<code><backup_file></code>	Checks integrity of the specified backup file, which is: <ul style="list-style-type: none"> ▪ backup digital sign match ▪ backup file format ▪ content files integrity
<code>-i --info</code>	<code><backup_file></code>	Shows the backup file description.
<code>-h --help</code>		Displays help on the utility usage.

Options

Option	Argument	Description
-level	clients resellers domains server	Specifies restoring level. Required with the—restore command.
-filter	<file> <list:<item1_name>[,<item2_name>[,...]]>	Specifies list of domain, client or reseller names for restore. The object names are listed either in a specified file, one per line, or as the option argument, separated by commas.
-license		Restores Plesk license key from the backup.
-verbose		Enables verbose restore mode.
-debug		Enables debugging restore mode.
-conflicts-resolution	<file>	Specifies file which describes conflict resolution policies and rules.
-suspend		Suspends the domains being restored.

Custom Conflict Resolutions

This Reference section is devoted to implementing custom conflict resolutions during restore.

In this section:

Conflict Description Messages	44
Resolutions Description Format.....	45
Samples of Policy Description	54
Samples of Conflict Resolution With Rules.....	55

Conflict Description Messages

Conflict descriptions returned by `pleskrestore` utility contain `message` elements included for the GUI generation purposes. Despite of the self-explaining character of XML conflict descriptions, values of the `message` elements may be confusing, so this section references the meanings of these messages as they are displayed in Plesk GUI.

Value of message element	Message displayed in Plesk GUI
<code>backup__restore__object_vhost</code>	Virtual host
<code>backup__restore__object_plesk_admin</code>	Plesk server administrator
<code>backup__restore__conflict__object__name</code>	<object name>
<code>backup__restore__conflict__object__complex_name</code>	<object name> of <group name>
<code>backup__restore__conflict__object__mailname</code>	<mail name>@<domain name>
<code>backup__restore__object_ftpuser</code>	FTP account
<code>backup__restore__object_frontpageuser</code>	Frontpage account
<code>backup__restore__object_webuser</code>	web user
<code>backup__restore__object_domain</code>	domain
<code>backup__restore__object_subdomain</code>	subdomain
<code>backup__restore__object_domainaliases</code>	domain alias
<code>backup__restore__object_client</code>	client
<code>backup__restore__object_reseller</code>	reseller
<code>backup__restore__object_autoresponder</code>	autoresponder
<code>backup__restore__object_mailalias</code>	mail alias

Value of message element	Message displayed in Plesk GUI
backup__restore__object_database	database
backup__restore__object_mailname	mail account
backup__restore__conflict_timing_reason_owner_absent	Cannot restore object: object owner is not specified
backup__restore__conflict_timing_reason_wrong_owner	Cannot restore object: object owner does not exist in Plesk
backup__restore__conflict_timing_reason_object_already_exists	Cannot restore <object name>: <object name> <object type> already exists in Plesk
backup__restore__conflict_configuration_reason_ip	Cannot restore object: required IP address <IP> not found in owner's IP pool
backup__restore__conflict_configuration_reason_db	Cannot restore database: required database server <host> is not registered in Plesk
backup__restore__conflict_configuration_reason_site_app	Cannot restore web application: required web application <application name> not found in owner's web application pool
backup__restore__conflict_unique_reason_name_already_used	Cannot restore <object>: name <unique resource name> is already used in Plesk by another <object>
backup__restore__conflict_resource_usage_reason	Cannot restore object: resource limit <limit name> will be exceeded (required: <value>, available: <value>)

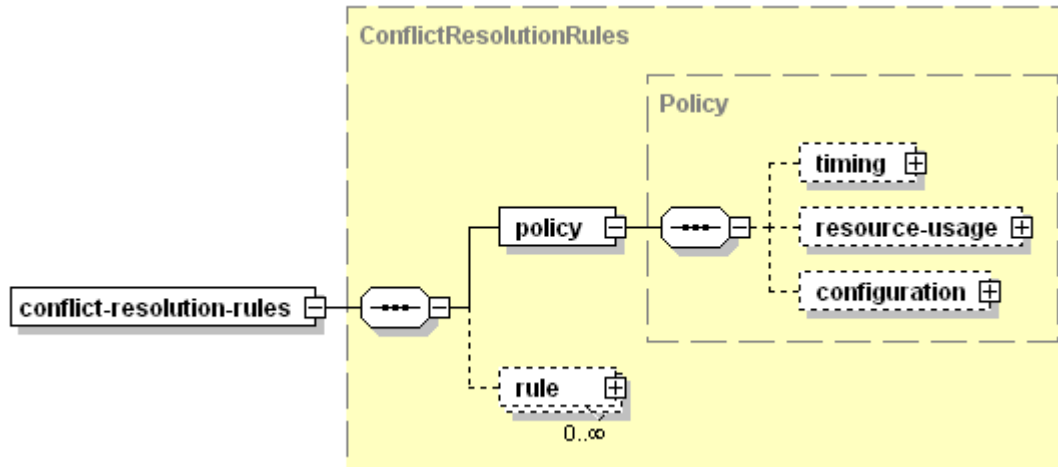
Resolutions Description Format

In this section:

Policies.....	46
Rules.....	49

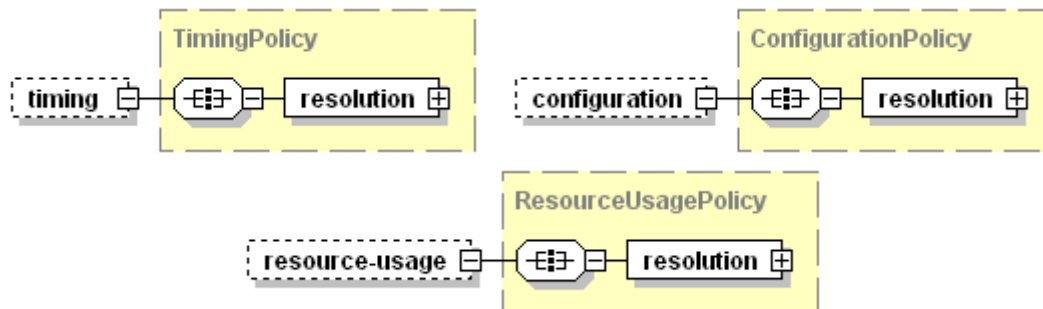
Policies

The file should be structure as follows.

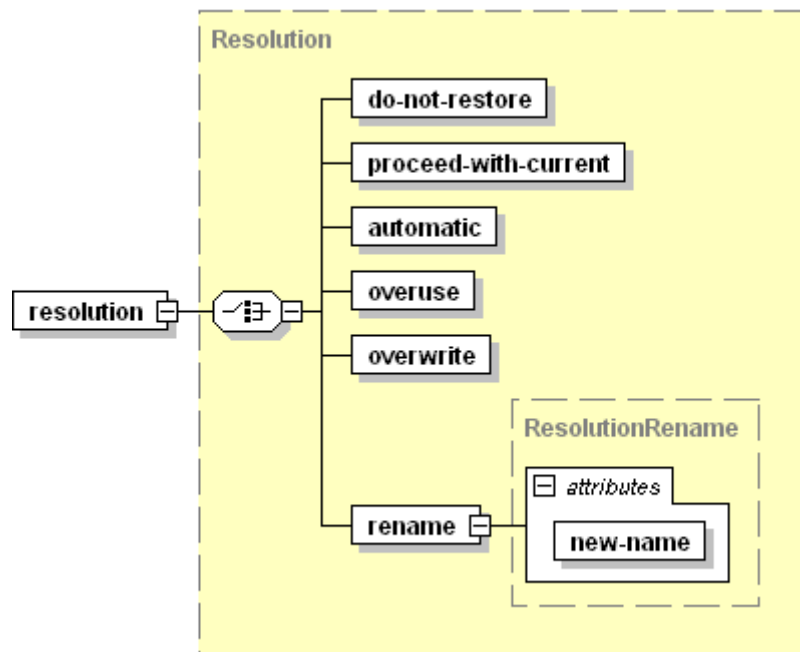


- `conflict-resolution-rules`
Required, document root element.
- `policy`
- Required, contains the policies descriptions. Children, if present, *must* be placed in order shown on the scheme.
 - `timing`
Optional, contains description of policy on resolving timing conflicts. See the structure below.
Must present in the document if a timing policy should be used during the restore.
May not present in the document if no policy required for timing conflicts.
 - `resource-usage`
Optional, contains description of policy on resolving resource usage conflicts. See the structure below.
Must present in the document if a resource usage policy should be used during the restore.
May not present in the document if no policy required for resource usage conflicts.
 - `configuration`
Optional, contains description of policy on resolving configuration conflicts. See the structure below.
Must present in the document if a configuration policy should be used during the restore.
May not present in the document if no policy required for configuration conflicts.
- `rule`
Optional, contains the rule descriptions. For details on the node structure, refer to the **Resolutions Description Format: Rules** section (see page 49).

The policy elements have the same structure:



- `resolution`
Required, contains a definition of conflict resolution. Structured as follows:

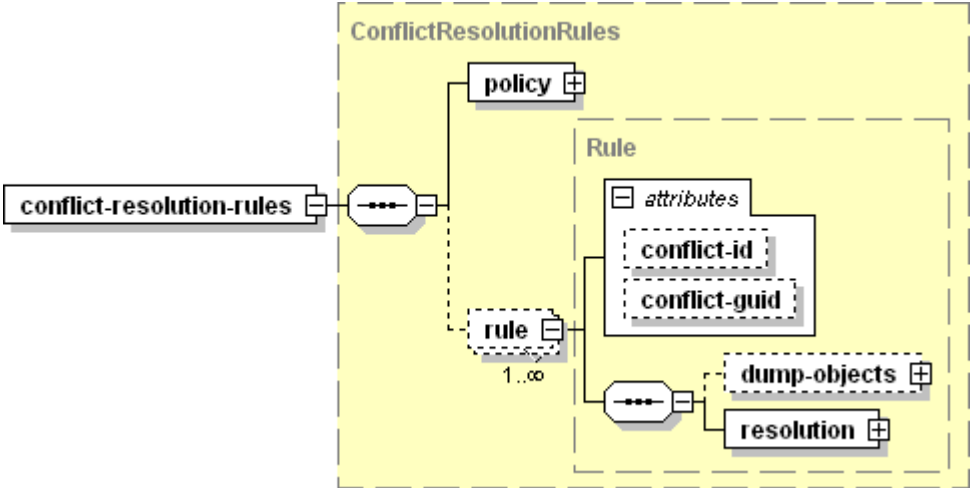


The `resolution` element *must not* be empty, it is *required* that it contains one, and only one of its children elements:

- `do-not-restore`
Sets the Do Not Restore resolution, *empty value*.
- `proceed-with-current`
Sets the Proceed With Current resolution, *empty value*.
- `automatic`
Sets the Automatic resolution, *empty value*.
- `overuse`
Sets the Overuse resolution, *empty value*.
- `overwrite`
Sets the Overwrite resolution, *empty value*.
- `rename`
Sets the Rename resolution, *empty value*.
 - `new-name`
Required, makes sense only if defined for configuration conflicts. Specifies a name of new configuration that should be assigned to all conflict objects. Value must be a string.

Rules

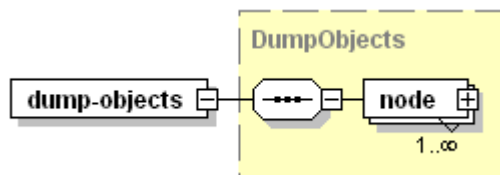
The file should be structure as follows.



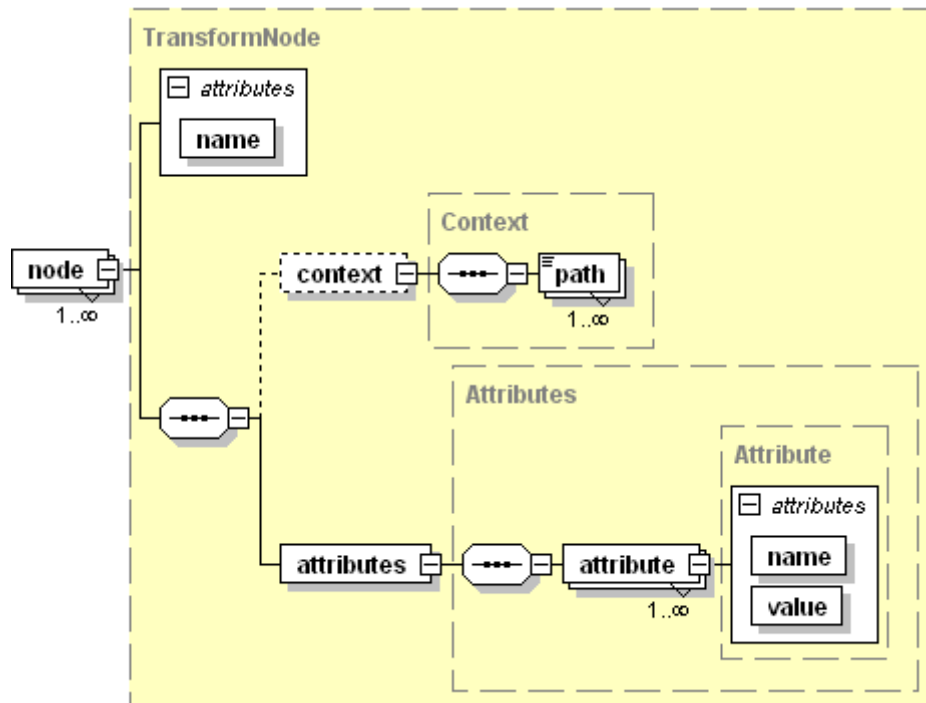
- `conflict-resolution-rules`
Required, document root element.
 - `policy`
Required, contains the policies descriptions. For details on the node format, refer to the **Resolutions Description Format: Policies** section (on page 46).
The element content *must* reflect the conditions under which the conflicts were detected.
 - `rule`
Optional, contains a rule description.
Must present in the document when defining conflict resolution rules. Should present as many times as the number of unresolved conflicts.

At least one of the attributes (`conflict-id`, `conflict-guid`) **MUST** present.
 - `conflict-id`
Optional, defines ID of the conflict being resolved. Value is integer.
The ID should be obtained from the conflict description returned by `pleskrestore` (the `"/conflicts-description/conflict[@id]"` attribute value)
 - `conflict-guid`
Optional, defines global ID of the conflict being resolved. Value is string.
The GUID should be obtained from the conflict description returned by `pleskrestore` (the `"/conflicts-description/conflict[@guid]"` attribute value).
If omitted, the conflict for resolution is identified by ID.
 - `dump-objects`
Optional, holds a collection of descriptions of backup objects involved into the conflict and taking the same conflict resolution
Must present in the document in case when different objects involved in the same conflict should be resolved in different ways.
May not present in the document in case when all objects involved in the conflict should be resolved the same way.
See the structure below.
 - `resolution`
Required, contains definition of resolution for the conflict, see the structure below.

`dump-objects` structure:

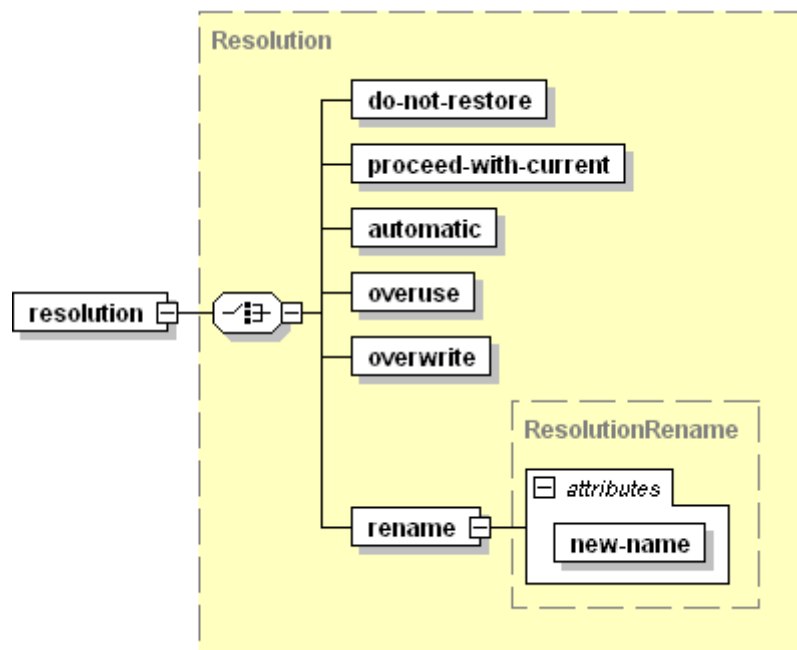


- `node`
Required, contains a description of backup object involved in the conflict.
The element contents *must* be taken from the conflict description returned by `pleskrestore` (the `/conflicts-description/conflict/conflicting-objects/node` element).
Structured as follows:



- `name`
Required, specifies the object type, value must be a string.
- `context`
Optional, holds a collection of data specifying the object position in backup.
 - `path`
Required if the `context` element is present in the document, specifies the location of object definition in the backup metadata. Value must be a string conforming to the XPath notation.
- `attributes`
Required, holds a collection of the object properties.
 - `attribute`
Required, specifies a particular property of the object (e.g., login, ID, GUID, etc.), *empty value*.
 - `name`
Required, specifies the property name, value must be a string.
 - `value`
Required, specifies the property value, value must be a string.

resolution structure:



The `resolution` element *must not* be empty, it is *required* that it contains one, and only one of its children elements:

- `do-not-restore`
Sets the Do Not Restore resolution for the conflict, *empty value*.
- `proceed-with-current`
Sets the Proceed With Current resolution for the conflict, *empty value*.
- `automatic`
Sets the Automatic resolution for the conflict, *empty value*.
- `overuse`
Sets the Overuse resolution for the conflict, *empty value*.
- `overwrite`
Sets the Overwrite resolution for the conflict, *empty value*.
- `rename`
Sets the Rename resolution for the conflict, *empty value*.
 - `new-name`
Required, specifies a name of unique resource that should be assigned to the conflict object(s), value must be a string.
Makes sense only for unique resource usage conflicts (mapping of IP, database server, object owner).

Samples of Policy Description

Default Plesk conflict resolution policies are described in the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<conflict-resolution-rules>
  <policy>
    <timing>
      <resolution>
        <proceed-with-current />
      </resolution>
    </timing>
    <resource-usage>
      <resolution>
        <do-not-restore />
      </resolution>
    </resource-usage>
    <configuration>
      <resolution>
        <automatic />
      </resolution>
    </configuration>
  </policy>
</conflict-resolution-rules>
```

The following conflict resolution file resolves all configuration conflicts with database mapping. This can be done in case all configuration conflicts beyond default policies appear because a database server defined in the backup is missed on the target Plesk.

```
<?xml version="1.0" encoding="UTF-8"?>
<conflict-resolution-rules>
  <policy>
    <configuration>
      <resolution>
        <rename new-name="host:192.0.2.12:port:3306"/>
      </resolution>
    </configuration>
  </policy>
</conflict-resolution-rules>
```

Samples of Conflict Resolution With Rules

This reference section contains format specification of conflict resolution rules description, and several examples of conflicts that may appear and their possible resolutions.

In this section:

Sample 1: Configuration Conflict With Missing IP Address	55
Sample 2: Configuration Conflict With Missing Database Server	59

Sample 1: Configuration Conflict With Missing IP Address

This sample represents descriptions of a conflict which appeared unresolved upon using default policies, and its resolution.

The conflict appears because of the following mismatch in backup data and destination Plesk configuration:

Backup	Destination Plesk
Domain example.com owned by the client with ID 30 has a physical hosting configured on shared IP 192.0.2.200.	Client with ID 30 does not have shared IP 192.0.2.200 in his/her IP pool.

The conflict is resolved with IP mapping suggesting that the restored domain will be hosted on shared IP 192.0.2.34 which is in the owner's IP pool.

Note that the conflict resolution XML contains no conflict resolution policies.

In this section:

Conflicts Description.....	56
Conflicts Resolution.....	58

Conflicts Description

```

<conflicts-description>
  <conflict id="0">
    <type>
      <configuration>
        <reason-description>

          <required-resource-description>
            <ip type="shared" value="192.0.2.200"></ip>
          </required-resource-description>

          <plesk-object-identifier>

            <!-- beginning of definition of Plesk object that conflicts with an
            object in the backup -->

            <!-- In resource usage conflicts, the plesk-object-identifier
            element specifies Plesk object which is an owner of the conflicting
            resource. In this example, the conflicting resource is IP, and its
            owner is a described client with ID 30. -->

            <type>client</type>
            <database-id>30</database-id>
            <guid>93dbe1b1-cff5-430f-8466-5b810099772f</guid>
            </plesk-object-identifier>

            <!-- end of definition of Plesk object that conflicts with an
            object in the backup -->

            </reason-description>
            <resolve-options>

              <option name="do-not-restore"></option>
            <option name="rename"></option>
            <option name="automatic"></option>
            </resolve-options>

            <!-- resolve-options element lists all resolutions that are
            possible for this particular conflict. When composing the conflict
            resolution rule, you should choose one of these resolutions. -->

            </configuration>
          </type>
        <conflicting-objects>

          <!-- beginning of definition of backup objects that conflict with
          destination Plesk objects. Here, it is a domain example.com -->

          <node children-processing-type="" name="domain">
            <attributes>

              <attribute name="id" value="25"></attribute>
              <attribute name="guid" value="0822c175-a10d-459e-bd3a-
e5cbc497e1f0"></attribute>

```



```
<attribute name="owner-guid" value="93dbelb1-cff5-430f-8466-5b810099772f"></attribute>
  <attribute name="name" value="example.com"></attribute>
  </attributes>
</node>

</conflicting-objects>
<!-- end of definition of backup objects that conflict with
destination Plesk objects -->

<overview>

<!-- beginning of more detailed conflict overview. Here, the
conflict appears because the required IP 192.0.2.200 is not in the
owner s IP pool -->
  <object>
    <message>backup__restore__conflict_object_name</message>
    <name>example.com</name>
    <type>domain</type>
    <reasons>
      <reason>

<message>backup__restore__conflict_configuration_reason_ip</message>
      <param name="ip-address" value="192.0.2.200"></param>
      <param name="ip-type" value="shared"></param>
      <param name="type" value="client"></param>
    </reason>
  </reasons>
</object>
</overview>
  <!-- end of detailed conflict overview -->
</conflict>

</conflicts-description>
```

Conflicts Resolution

```
<?xml version="1.0" encoding="UTF-8"?>
<resolve-conflicts-task-description>
  <conflict-resolution-rules>
    <policy />
    <rule conflict-id="0">
      <dump-objects>
        <node name="domain">
          <attributes>
            <attribute name="id" value="25"></attribute>
            <attribute name="guid" value="0822c175-a10d-459e-bd3a-
e5cbc497e1f0"></attribute>
            <attribute name="owner-guid" value="93dbelb1-cff5-430f-8466-
5b810099772f"></attribute>
            <attribute name="name" value="example.com"></attribute>
          </attributes>
        </node>
      </dump-objects>
      <resolution>
        <!-- beginning of the conflict resolution definition: IP mapping:
upon restore, the conflicting domain example.com should have hosting
configured on IP 192.0.2.34 -->
        <rename new-name="ip-type:shared:ip-address:192.0.2.34"/>
      </resolution>
      <!-- end of the conflict resolution definition -->
    </rule>
  </conflict-resolution-rules>
</resolve-conflicts-task-description>
```

Sample 2: Configuration Conflict With Missing Database Server

This sample represents description and resolution of configuration conflicts which appeared unresolved due to the lack of the required database server on the destination PPP.

The conflicts appear because of the following mismatches in backup data and destination Plesk configuration:

Backup	Destination Plesk
Domain sample.net has database mysql_db2_7469 on the MySQL database server with host name 192.0.2.15 listening on port 3306.	No MySQL servers configured on host 192.0.2.15.
Domain 69.sample.net has database mysql_db1_6319 on the MySQL database server with host name 192.0.2.15 listening on port 3306.	

These conflicts are resolved with database mapping (Rename resolution) suggesting that the first databases will be restored on the MySQL server with host name 192.0.2.12, and the second to the local MySQL database server.

In this section:

Conflicts Description.....	60
Conflicts Resolution.....	64

Conflicts Description

```

<conflicts-description>
  <conflict id="0">
    <type>
      <configuration>
        <reason-description>
          <required-resource-description>
            <db-server host="192.0.2.15" type="mysql" port="3306"></db-server>
          </required-resource-description>
          <plesk-object-identifier>
            <!-- beginning of definition of Plesk object that conflicts with an
            object in the backup. In resource usage conflicts it is owner of the
            conflicting resource. Here, it is Plesk administrator who is the
            owner of all database servers -->
            <type>admin</type>
            <database-id>1</database-id>
            <guid>00000000-0000-0000-0000-000000000000</guid>
            </plesk-object-identifier>
            <!-- end of definition of Plesk object that conflicts with an
            object in the backup -->
          </reason-description>
          <resolve-options>
            <option name="do-not-restore"></option>
            <option name="rename"></option>
            <option name="automatic"></option>
          </resolve-options>
        </configuration>
      </type>
      <conflicting-objects>
        <!-- beginning of definition of backup objects that conflict with
        destination Plesk objects. Here, it is database mysql_db2_7469 -->
        <node children-processing-type="" name="database">
          <attributes>
            <attribute name="guid" value="86124f4a-5935-48c4-80df-
            6d3e9c645378_db_20"></attribute>
            <attribute name="owner-guid" value="86124f4a-5935-48c4-80df-
            6d3e9c645378"></attribute>
            <attribute name="name" value="mysql_db2_7469"></attribute>
          </attributes>
        </node>
      </conflicting-objects>
    </conflict>
  </type>
</conflicts-description>

```

```

    </node>
    </conflicting-objects>
    <!-- end of definition of backup objects that conflict with
destination Plesk objects -->
    <overview>
    <!-- beginning of detailed overview of the conflict. This conflict
appears because database mysql_db2_7469 requires MySQL database
server with host name 192.0.2.15 listening on port 3306, which is
not configured on the destination Plesk. -->
        <object>
        <message>backup__restore__conflict_object_complex_name</message>
        <name>mysql_db2_7469</name>
        <type>database</type>
        <owner-name>sample.net</owner-name>
        <reasons>
        <reason>
<message>backup__restore__conflict_configuration_reason_db</message>
            <param name="db-type" value="mysql"></param>
            <param name="db-host" value="192.0.2.15"></param>
            <param name="db-port" value="3306"></param>
            <param name="type" value="admin"></param>
            <param name="name"
value="backup__restore__object_plesk_admin"></param>
        </reason>
        </reasons>
        </object>
    </overview>
    <!-- end of detailed overview of the conflict -->
</conflict>

<!-- ===== begin new conflict description =====
-->
<conflict id="1">
    <type>
    <configuration>
        <reason-description>
        <required-resource-description>
            <db-server host="192.0.2.15" type="mysql" port="3306">
</db-server>
        </required-resource-description>
        <plesk-object-identifier>

```

```
<!-- beginning of definition of Plesk object that conflicts with an
object in the backup In resource usage conflicts it is owner of the
conflicting resource. Here, it is Plesk administrator who is the
owner of all database servers -->
```

```
<type>admin</type>
```

```
<database-id>1</database-id>
```

```
<guid>00000000-0000-0000-0000-000000000000</guid>
```

```
</plesk-object-identifier>
```

```
<!-- end of definition of Plesk object that conflicts with an
object in the backup -->
```

```
</reason-description>
```

```
<resolve-options>
```

```
  <option name="do-not-restore"></option>
```

```
  <option name="rename"></option>
```

```
  <option name="automatic"></option>
```

```
</resolve-options>
```

```
</configuration>
```

```
</type>
```

```
<conflicting-objects>
```

```
<!-- beginning of definition of backup objects that conflict with
destination Plesk objects. Here, it is database mysql_db1_6319 -->
```

```
  <node children-processing-type="" name="database">
    <attributes>
```

```
      <attribute name="guid" value="e1fbb4b2-538b-4542-9220-
56808741a3d3_db_19"></attribute>
```

```
      <attribute name="owner-guid" value="e1fbb4b2-538b-4542-9220-
56808741a3d3"></attribute>
```

```
      <attribute name="name" value="mysql_db1_6319"></attribute>
```

```
    </attributes>
```

```
  </node>
```

```
</conflicting-objects>
```

```
<!-- end of definition of backup objects that conflict with
destination Plesk objects -->
```

```
<overview>
```

```
<!-- beginning of detailed overview of the conflict. This conflict
appears because database mysql_db1_6319 requires MySQL database
server with host name 192.0.2.15 listening on port 3306, which is
not configured on the destination Plesk. -->
```

```
  <object>
```

```
    <message>backup__restore__conflict_object_complex_name</message>
```

```
    <name>mysql_db1_6319</name>
```

```
    <type>database</type>
```

```
<owner-name>69.sample.net</owner-name>
<reasons>
  <reason>
<message>backup__restore__conflict_configuration_reason_db</message>
  <param name="db-type" value="mysql"></param>
  <param name="db-host" value="192.0.2.15"></param>
  <param name="db-port" value="3306"></param>
  <param name="type" value="admin"></param>
  <param name="name"
value="backup__restore__object_plesk_admin"></param>
  </reason>
</reasons>
</object>
</overview>
<!-- end of detailed overview of the conflict -->
</conflict>
</conflicts-description>
```

Conflicts Resolution

```
<?xml version="1.0" encoding="UTF-8"?>
<resolve-conflicts-task-description>
  <conflict-resolution-rules>
    <policy />
    <rule conflict-id="0">
      <!-- beginning of the first conflict resolution rule: restore the
database described in the node element on local MySQL server
listening on the port 3306 -->
      <dump-objects>
        <node name="database">
          <attributes>
            <attribute name="name" value="mysql_db2_7469"/>
          </attributes>
        </node>
      </dump-objects>
      <resolution>
        <rename new-name="host:192.0.2.12:port:3306"/>
      </resolution>
    </rule>
    <!-- end of the first conflict resolution rule -->
    <rule conflict-id="1">
      <!-- beginning of the second conflict resolution rule: restore the
database described in the node element on local MySQL server
listening on the port 3306 -->
      <dump-objects>
        <node name="database">
          <attributes>
            <attribute name="name" value="mysql_db1_6319"/>
          </attributes>
        </node>
      </dump-objects>
      <resolution>
        <rename new-name="host:localhost:port:3306"/>
      </resolution>
    </rule>
    <!-- end of the second conflict resolution rule -->
  </conflict-resolution-rules>
</resolve-conflicts-task-description>
```


</conflict-resolution-rules>

</resolve-conflicts-task-description>