



4PSA Total Backup 1.5.5
for Plesk 7.x Reloaded
User's Guide

Manual Version 0.91

For suggestions regarding this manual contact:

docs@4psa.com

© Copyrights 2002 – 2005 Rack-Soft. All rights reserved

Distribution of this work or derivative of this work is prohibited unless prior written permissions is obtained from the copyright holder.

Plesk is a Registered Trademark of SWsoft, Inc.

Linux is a Registered Trademark of Linus Torvalds.

RedHat is a Registered Trademark of Red Hat Software, Inc.

FreeBSD is a Registered Trademark of FreeBSD, Inc.

All other trademarks and copyrights are the property of their respective owners.

Table of Contents

Chapter 1. About 4PSA Total Backup 1.5.5.....	4
4PSA Total Backup 1.5.5 Features	4
Chapter 2. The Administrator Module	5
1. Backup Sessions.....	5
Log Messages Explained	6
2. Log Management.....	8
Search History.....	8
History Results	8
3. Settings	9
Backup Reports	9
Backup Setup.....	9
Server Backup Settings	9
Backup Schedule Period.....	11
Remote Storage Settings	12
Maintenance Settings	13
Interface Settings	14
GnuPG Keys Management	14
4. License Management.....	16
Chapter 3. Backup Tips & Tricks	16
1. Comparison between Tar and Psadump.....	16
2. Why Do I Need the Remote Storage Facility?	17
3. How to Restore?.....	18
Restore Using Tbrestore.....	18
Restore Files from Tar Archives.....	18
Restore the MySQL Databases	22
Restore the PostgreSQL Databases.....	22
4. Low Level Settings.....	23
Appendix A. Server Compatibility.....	24

Chapter 1. About 4PSA Total Backup 1.5.5

4PSA Total Backup 1.5.5 is a server-level application that provides an advanced automated backup system for Plesk 7.x Reloaded servers. The application consists of a single administrator-only module that allows the server administrator to control various backup options and track the history of the backup processes through the integrated logging mechanism.

4PSA Total Backup 1.5.5 Features

- Accessible to server administrator only
- Administrator can schedule backup sessions
- PHP interface, low level ANSI C backup engine for superior performance and minimum resources utilization
- Multi volume backups supported
- Incremental archives capabilities
- Gzip and bzip2 compression tools support
- GPG Backup files encryption
- Backup cycle maintained by the server administrator
- Number of backups stored locally defined by administrator
- Automatically sent alerts to server administrator
- Restore functions available
- Advanced logging features (file + database)
- Quality of Service settings
- Remote backup storage capabilities
 - FTP or SSH protocol
 - Automatic transfer of archives to remote servers
 - Transfer integrity checks
 - Automatic maintenance of remote backups
- Ability to automate psadump backups
- Ability to define exclude backup directories and include backup directories
- Debugging features

- Language packs capabilities

Chapter 2. The Administrator Module

The 4PSA Total Backup administrator module can be accessed after you login to your Plesk 7.x Reloaded server using the admin account. In order to open the 4PSA Total Backup interface click the [4PSA Total Backup](#) link available in the Custom navigation menu located on the left side of the Plesk interface.

The 4PSA Total Backup tool bar is available on top of the application's interface. The tool bar provides an easy method for the server administrator to view details about backup sessions, view the history of the backup operations outcome, view a report about 4PSA Total Backup, modify various parameters that control the behavior of 4PSA Total Backup, manage GnuPG public and private keys, manage the local and remote storage features, change interface settings, and manage the 4PSA Total Backup license.

1. Backup Sessions

4PSA Total Backup keeps a detailed log of all operations performed during a backup. The logging mechanism helps you identify problems that may appear during the backup operation.

In the Backup sessions area (click the **Sessions** button available in the tool bar) the server administrator can view the backup actions of the latest backup cycle performed by the backup engine. This cycle is split in backup sessions.

For every backup session displayed on this page, the session starting and completion date are available.

A log entry consists of three fields:

Date - The system time when the logged action occurred (day month year, hh:mm:ss)

Action taken - Action performed by 4PSA Total Backup

Outcome - The outcome of the backup operation, which can be success or failure.

The following events are logged:

- disk space exceeded

- backup operation starting and completion time
- outcome of files backup operation
- outcome of databases backup operation
- outcome of remote connection establishment
- file transmission errors
- remote file system operations
- database operations

Logging is conducted in database and on file. The log file is located in **Local archives directory path** field (available in the Settings area) and the file is called `action.log`. You may want to setup a log rotation system for this file or empty it periodically.

Log Messages Explained

backup session [session no] started - Marks the startup of the backup session.

using passive FTP transfer mode - Indicates that passive ftp transfer mode is being used.

using active FTP transfer mode - Indicates that active ftp transfer mode is being used.

do_backup is already running. Backup cannot start twice! - 4PSA Total Backup engine is running. Most likely the old instance didn't end due to an error. Check the logs and eliminate the error cause. 4PSA Total Backup can not continue.

4PSA Total Backup GPG keys not found, backups will be made without encryption - If GPG has to be used and no keys are found the backup process will continue, but the backups will not be encrypted.

disk space absolute limit check failed (available=[avail], required=[req]) - The absolute available disk space (MB) is below the imposed limit.

disk space relative limit check failed (available=[avail] required=[req]) - The relative available disk space (percentage %) is below the imposed limit.

[bktool] backup with no compression started - Indicates the start of a backup using the backup tool [bktool] (tar or psadump) without compression.

[bktool] backup with [comptool] compression started - Indicates the start of a backup using the backup tool [bktool] (tar or psadump) and the compression tool [comptool] (gzip or bzip2)

[bktool] backup with no compression [status] - Indicates the outcome of the backup process ([status] which can be "failed" or "succeeded") using the backup tool [bktool] (tar or psadump) without compression

[bktool] backup with [comptool] compression [status] - Indicates the outcome of the backup process ([status] which can be "failed" or "succeeded") using the backup tool [bktool] (tar or psadump) and the compression tool [comptool] (gzip or bzip2)

an unexpected condition occurred during the backup! - Indicates that one of the issued commands failed to execute.

MySQL databases dump with no compression started - Indicates the start of a MySQL dump without compression process.

MySQL databases dump with [comptool] compression started - Indicates the start of a MySQL dump without compression process. The compression tool [comptool] can be gzip or bzip2.

MySQL databases dump with no compression failed - The mysqldump process could not be started (mysqldump utility is missing or some other error occurred).

MySQL databases dump with [comptool] compression failed - The mysqldump process could not be started (mysqldump utility is missing or some other error occurred).

MySQL databases dump with no compression [status] - Indicates the outcome of the MySQL dump process without compression ([status] can be "failed" or "succeeded").

MySQL databases dump with [comptool] compression [status] - Indicates the outcome of the MySQL dump process ([status] can be "failed" or "succeeded") using the compression tool [comptool] (can be gzip or bzip2).

PostgreSQL databases dump with no compression started - Indicates the start of the PostgreSQL dump process without compression.

PostgreSQL databases dump with [comptool] compression started - Indicates the start of the PostgreSQL dump process using the compression tool [comptool] (can be gzip or bzip2).

PostgreSQL databases dump with no compression [status] - Indicates the outcome of the PostgreSQL dump process without compression ([status] can be "failed" or "succeeded").

PostgreSQL databases dump with [comptool] compression [status] - Indicates the outcome the PostgreSQL dump process ([status] can be "failed" or "succeeded") using the [comptool] compression tool (can be gzip or bzip2).

FTP transfer error while trying to export [path] transfer failed - An error occurred while trying to upload the backup files to the FTP server.

FTP Backups validation failed - The transferred backup files failed the validation check. This can occur only if FTP Transfer Validation is enabled.

FTP Backups validation successful - The transferred backup files were successfully validated. The validation occurs only if FTP Transfer Validation is enabled.

FTP transfer of [path] succeeded - The backup files were successfully uploaded to the FTP site.

post FTP transfer operations succeeded - The post FTP transfer operations were completed successfully.

post FTP transfer operations failed - The post FTP transfer operations failed to complete.

SSH transfer of [path] [status] - Indicates the outcome of the backup files transfer operation to the SSH server ([status] can be "failed" or "succeeded").

post SSH transfer operations [status] - Indicates the outcome of the post SSH transfer operations ([status] can be "failed" or "succeeded").

backup session [session no] ended - Marks the end of the backup session.

2. Log Management

In this area (click the **Logs** button available in the tool bar) the server administrator can view and search backup operations logs based on several criteria.

Search History

In this area the server administrator can search the backup operation logs and choose the search criteria. The following criteria are available:

From and **To** - Search for logs included between two dates in year-month-day format.

Outcome - Search for logs with a specified finish status, which can be success or failure.

Show - Limit the number of results to view in one page.

History Results

Based on the chosen search criteria, 4PSA Total Backup displays the logs that matched these criteria. Every log entry consists of three fields:

Date - The system time when the logged action occurred (day month year, hh:mm:ss)

Action taken - Action performed by 4PSA Total Backup

Outcome – The outcome of the backup operation, which can be success or failure.

3. Settings

In this area (click the **Settings** button available in the tool bar) the server administrator can view a 4PSA Total Backup report, modify various parameters that control the behavior of 4PSA Total Backup, manage the local and remote storage features, manage GnuPG public and private keys, and change interface settings.



Before running 4PSA Total Backup for the first time you must adjust these settings.

Backup Reports

The **Product version** field displays the version of the 4PSA Total Backup installed on the server.

Backup Setup

This area allows the server administrator to enable/disable the backup engine and to setup various available options.

Backup active – When this option is enabled, the backup engine will run as scheduled. When it is disabled, no backups will be performed.

Send email alerts on errors - If this option is enabled, an email message containing a description of the error that has occurred in the backup process will be sent to the email address specified in the **Administrator email** field.

Administrator email - The email address of the server or the backup administrator. It must be a valid address.

Server Backup Settings

This area allows the server administrator to setup various options that control how the backup process is executed.

Backup tool - By default, the tar software is used to perform system backups. Alternatively, if you have installed the Plesk psadump utility, you can use it for the backup operation.

Compression tool - You can choose to compress the archive files generated by the backup operation in order to save disk space. There are two compression tools that can be used, gzip or bzip2. Due to a restriction in the psadump tool you cannot use bzip2 compression for archives created using psadump.



Creating compressed archives is not recommended because a single modified bit in a compressed archive can make the restoration process impossible.

Multi volume archives - You can choose the size of the backup archives. Select a volume size only if you want the backup archives to be split in multiple, equally sized files. These smaller files are better for easy storage on external devices with a limited storage capacity (floppy, CD). You can choose among different preset file sizes.

Backup MySQL databases - If enabled, the mysqldump tool will be used to make a full backup of all your MySQL databases. This setting is valid (and required) only when the backup tool is tar. Psadump integrates databases backup.

Backup PostgreSQL databases - If enabled, the pgdump tool will be used to make a full backup of all your PostgreSQL databases. This setting is valid only when the backup tool is tar. Psadump integrates databases backup.

Local archives directory path - This is the directory where the backup archives will be stored. It should be a local path. We recommend you to use a path located on a different physical disk, especially if you do not use a remote storage facility.



Use an empty directory because the contents of this directory will be erased when the backup archives are copied. Avoid using directory names that contain spaces.

Backup cycle - When the tar software is used to perform backups, setting the backup cycle to a value greater than 1 will result in creating incremental backups with the specified cycle length. Incremental backups are part of a backup technique that consists in creating a full backup of the system at the beginning of the backup cycle


and then in creating backups of the files that were modified after the initial full backup. This backup technique has the advantage of creating smaller backup files, therefore saving disk space and minimizing system load.



Note

When the psadump software is used, the backup cycle represents the number of backup files that will be stored on your machine.

Reset backup cycle - Reset the 4PSA Total Backup backup cycle.

GPG Encryption - Use GnuPG to encrypt the backup files. The administrator can manage the GPG public and private keys by clicking the  **Edit keys** icon.



Note

The GPG encryption is recommended only for EXPERT users who understand what they are doing. It is not recommended to encrypt large backup files because the backup process will be very slow.

Backup Schedule Period

This area allows the server administrator to setup various options that control when the backup process is executed.

Schedule backup tool to run on - You can set the time when the backup operation automatically starts. This is done by adding an entry in your system's crontab manager table in order to schedule the backup operation. The backup operation will be automatically performed when the time specified by the date fields matches the current time. The following time fields are available:

Minute - The accepted values are between 0 and 59.

Hour - The accepted values are between 0 and 23.

Day of the Month - The accepted values are between 1 and 31.

Month - The accepted values are between 1 and 12.

Day of the Week - The accepted values are between 0 and 7, where 0 and 7 are **Monday and Sunday**.



Note

You can also use an asterisk (*) which indicates that any value is matched.

Lists of numbers are also allowed, e.g. “0,15,30,45” for the **Minute** field.

This is an example: If you want to schedule backups every night at 2:15, the settings must be as below.

Minute: 15

Hour: 2

Day of the Month: *

Month: *

Day of the Week: *

Remote Storage Settings

4PSA Total Backup has the option to keep a copy of the backup archives on a remote machine such as a backup server or a dedicated storage facility. When this option is enabled, the server administrator must provide the connection details required to connect and transfer files on the remote machine.

Remote storage facility – When enabled, 4PSA Total Backup will automatically save a copy of the backup archives on a remote storage facility. The files can be transferred to the remote server using the SSH or the FTP protocol. SSH is the recommended method since it provides increased security and an improved mechanism for error detection / correction.



Note

When using SSH, your machine must be able to login with no password to the remote server using key passing. The public key of you server must be placed in the remote server’s authorized key list. This technique is called key exchange.

Remote machine IP address - The IP address of the remote server

Remote machine hostname - The hostname of the remote server

Use for remote connection - The server administrator must choose how to connect to the remote machine, using the IP address or the hostname. One of the

Remote machine IP address or **Remote machine hostname** options must be filled in based on that choice.

Remote machine login name - The user name required to connect to the remote machine (REQUIRED).

Remote machine password - The password required to connect to the remote machine using the specified user name. (REQUIRED ONLY FOR FTP ignored for SSH connections).

Remote machine storage dir - The directory on the remote machine where the backup files will be stored. Do not place other content on this directory as it will be erased.



Note

In order to minimize the remote storage space, only one backup cycle is kept remotely. You can always restore the system to a previous state using the backup files. However, you are limited by the date of the first backup in the cycle. Avoid using directory names that contain spaces.

FTP transfer mode - This mode can be active or passive. In passive mode, the administrator initiates the data connection by connecting to the data port. Passive mode is often necessary for operations performed from behind firewalls which do not permit incoming connections. The passive mode may need to be disabled, if the server administrator connects to a FTP server that does not support passive operation. Active mode can be used when the server can accept incoming connections.

FTP transfer validation - If enabled, the sizes of the transferred files are compared with the sizes of the original files.



Note

If you have trouble with the storage of the backup files on the FTP server, you can try to disable the FTP transfer validation. Make sure that files are transferred properly to the FTP server.

Maintenance Settings

The server administrator can change the local storage settings.

Don't perform any backup if free disk space is below - The backup operation will fail to start if the free disk space on the **Local archives directory path**

is below the chosen value. You can specify an absolute value (in MB) **e.g.** 1000 MB or a relative value (in %) **e.g.** 10% of the Local archives directory path's full size.

Keep a local copy of # sessions – A local copy of the backup archives can be stored as an emergency recovery source. This option sets the number of backup sessions stored locally.

Delete history records older than # months - In order to prevent the backup history records getting too large, the records that are older than a specified number of months can be deleted.

Interface Settings


In this area the server administrator can choose the interface settings.

Language - Allows the server administrator to select the language that will be used by 4PSA Total Backup's interface.

Custom button title - The name of the custom button in the left panel. The server administrator can change the default 4PSA Total Backup with a more descriptive name for his clients.

Context help - The 4PSA Total Backup application description that will appear in the left navigation panel.

GnuPG Keys Management

In order to manage the public and private GPG keys, the server administrator must click the  **Edit keys** icon available in the Server backup settings area. In the GPG management page the administrator can generate, import and export the GnuPG keys on/from the server. These keys are used when the backup files are encrypted with GPG.

Generate Keys

In this area the server administrator can generate the GPG keys. The following details will be displayed:

Real name – This is the name that will be displayed by the GPG key

Email address – This is the email address displayed by the GPG key

In order to generate the GPG keys with these details, the server administrator must click the **Generate** button.



Note

The generation of the keys may take up to several minutes, depending on the machine speed and randomness sources.

Import Keys

In this area the server administrator can import the GPG keys and upload them on the server. The following fields are available:

Public key - In this field the administrator must enter the name of the file that contains the public key that he wants to import or click the **Browse...** button to locate the desired file.

Private key - In this field the administrator must enter the name of the file which contains the private key that he wants to import or click the **Browse...** button to locate the desired file.

In order to upload the GPG keys, the server administrator must click the **Upload** button. The administrator can also enter the public and private key using the following fields:

Public key - The server administrator can fill in this field the public key

Private key - The server administrator can fill in this field the private key

In order to import the GPG keys, the server administrator must click the **Import** button. These keys must be created using the real name and email address displayed in the Generate keys area.



Note

After importing the keys, the server administrator must login to 4PSA Total Backup and issue the following command:

```
gpg --homedir /usr/local/tbackup/.gnupg --edit-key "Total Backup" trust
```

The administrator should type **5** and **ENTER** when prompted „*Ultimately trust the imported key*”. In order to exit the GPG console he must type **quit** and **ENTER**.

The server administrator **CANNOT USE** the imported key unless he performs the above steps.

Existing Keys

4PSA Total Backup displays in this area the GPG keys available on the server and allows the administrator to export these files on his computer. In order to download these keys to his local machine, the server administrator must click the **Download** button.

4. License Management

In this area you can manage the 4PSA Total Backup license. In order for 4PSA Total Backup to work correctly, a valid license key must be loaded. The license key must be generated by 4PSA based on your server IP and Plesk version installed on your server.

License key - The license key number. This is the key currently loaded on your server.

License key status - The status of the currently loaded license key.

Your server IP - The main IP address of your server. This is the IP for which the license key must be issued in order to work on this server. If the license is issued for another IP, it will not work.

License file - You can use this form in order to upload the license key to the server. The license key can also be executed in command line using the command: `sh keyno.sh`. If you can access other pages in 4PSA Total Backup there is no reason why you should upload a new key.

Chapter 3. Backup Tips & Tricks

In this chapter several general backup concepts are explained which apply to 4PSA Total Backup.

1. Comparison between Tar and Psadump

Most users are undecided what backup tool to use. You have below a comparison between psadump and tar.

Criteria	Psadump	Tar
Backup all server	No, only Plesk files	Yes, except several OS dependent directories
Backup particular directories	Not possible	Yes, you can adjust the list of directories that are backed up
Incremental backup	Not possible	Yes
Disk space required	Equal to the Plesk customer data, for every backup session in the cycle.	Almost equal to the server data, but only once in the cycle (first backup)
Server load	High	Medium
Databases dump	Yes, automatically	Separate option
Time required for the first backup session	Depends on server configuration and load ~ 5Gb/hour	Depends on server configuration and load. Faster than psadump
Time required for the incremental backup sessions	As in the first backup session, no incremental support	When performed daily, the time required by incremental backups is less than 15 min in most cases
Restore difficulty	Easy with psarestore	Easy with tbrestore
Time required by restore	Very slow restore process	Fast restore process

We strongly advise you to use tar because it puts a lower load on the server and the space required to store archives is lower. The restore process is much faster with tar than with psadump, but it might be more difficult for inexperienced administrators.

2. Why Do I Need the Remote Storage Facility?

4PSA Total Backup is able to transfer files to a remote server in order to increase the reliability of the backup process. Although you can store files locally, if you do not use a separate HDD the backups are as exposed as the other server files. Even with a dedicated backup HDD the data is not safe when a hacker breaks in.

Most Data Centers have dedicated backup facilities available for customers. If available, we recommend you to upload files to the remote computer using SSH.

In order to use SSH you must configure the server to login to the remote machine using key exchange. The private and public keys of the backup server should be placed in `/root/.ssh` while the `authorized_keys` file should reside in the `.ssh` directory of the remote server login account (for example `/home/backupaccount/.ssh`).



Tip

Do not use the root account to login to the remote backup server!

3. How to Restore?

Total Backup archives can be restored automatically using `tbrestore` or manually using system level utils. Below you can find several recommendations for this.

Restore Using Trestore

The easiest way to restore system files, directories and MySQL databases is to use the 4PSA Total Backup restore script `tbrestore`. The `tbrestore` script is located in `/usr/local/tbackup` directory and represents an easy, yet advanced way to restore from 4PSA Total Backup archives. In order to start the restore process, launch `tbrestore` and follow the instructions. The script will prompt you for restore options and will guide you through this process.

Restore Files from Tar Archives

To restore files from a tar archive, you can consult the tar documentation with “`man tar`”. Tar is a very flexible archiving tool and has a lot of features that can be used to restore specific files, directories, etc.

Example 1: Restore from a multivolume tar backup that was compressed using `gzip`.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: `root@localhost ~ # cd /tbackup.backups`

Change to the directory that contains the desired backup session:

Example: `root@localhost tbackup.backups # cd bak-2-tar-2005-3-10`

Assuming that you have in this directory files like `backup.tar.gz*` and `mysql.dmp.gz*`, issue the following commands:

```
ls backup.tar.gz* | grep -v "md5" | xargs cat | tar -C [dest_dir] -zxf -  
  
ls mysql.dmp.gz* | grep -v "md5" | xargs cat | gzip -q -c -d >  
[dest_dir]/mysql.dmp
```

Example 2: Restore a directory from a multivolume tar backup that was compressed using `bzip2`.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: `root@localhost ~ # cd /tbackup.backups`

Change to the directory that contains the desired backup session:

Example: `root@localhost tbackup.backups # cd bak-2-tar-2005-3-10`

Assuming that you have in this directory files like `backup.tar.bz2*` and `mysql.dmp.bz2*`, issue the following commands:

```
ls backup.tar.bz2* | grep -v "md5" | xargs cat | tar -C [dest_dir] -jxf -  
path/to/directory  
  
ls mysql.dmp.bz2* | grep -v "md5" | xargs cat | bunzip2 -q -c -d >  
[dest_dir]/mysql.dmp
```

Example 3: Restore from a tar backup file that was compressed using `gzip` and encrypted using `GnuPG`.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: root@localhost ~ # cd /tbackup.backups

Change to the directory that contains the desired backup session:

Example: root@localhost tbackup.backups # cd bak-2-tar-2005-3-10

Assuming that you have in this directory the files backup.tar.gz.gpg and mysql.dmp.gz.gpg, issue the following commands:

```
gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -o - -d backup.tar.gz.gpg | gzip -q -c -d | tar -C ./ -xf -
```

```
gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -o - -d mysql.dmp.gz.gpg | gzip -q -c -d > [dest_dir]/mysql.dmp
```

Example 4: Restore from a tar backup file that was compressed using bzip2 and encrypted using GnuPG.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: root@localhost ~ # cd /tbackup.backups

Change to the directory that contains the desired backup session:

Example: root@localhost tbackup.backups # cd bak-2-tar-2005-3-10

Assuming that you have in this directory the files backup.tar.bz2.gpg and mysql.dmp.bz2.gpg, issue the following commands:

```
gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -o - -d backup.tar.bz2.gpg | bunzip2 -q -c -d | tar -C [dest_dir] -xf -
```

```
gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -o - -d mysql.dmp.bz2.gpg | bunzip2 -q -c -d > [dest_dir]/mysql.dmp
```

Example 5: Restore a multivolume tar backup that was compressed using gzip and encrypted using GnuPG.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: root@localhost ~ # cd /tbackup.backups

Change to the directory that contains the desired backup session:

Example: root@localhost tbackup.backups # cd bak-2-tar-2005-3-10

Assuming that you have in this directory files like backup.tar.gz.gpg* and mysql.dmp.gz.gpg*, issue the following commands:

```
ls backup.tar.gz.gpg* | grep -v "md5" | xargs cat | gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -d - | gzip -q -c -d | tar -C [dest_dir] -xf -
```

```
ls mysql.dmp.gz.gpg* | grep -v "md5" | xargs cat | gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -d - | gzip -q -c -d > [dest_dir]/mysql.dmp
```

Example 6: Restore a multivolume tar backup that was compressed using bzip2 and encrypted using GnuPG.

Change to the directory where the 4PSA Total Backup backup files are kept:

Example: root@localhost ~ # cd /tbackup.backups

Change to the directory that contains the desired backup session:

Example: root@localhost tbackup.backups # cd bak-2-tar-2005-3-10

Assuming that you have in this directory files like backup.tar.bz2.gpg* and mysql.dmp.bz2.gpg*, issue the following commands:

```
ls backup.tar.bz2.gpg* | grep -v "md5" | xargs cat | gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -d - | bzip2 -q -c -d | tar -C [dest_dir] -xf -
```

```
ls mysql.dmp.bz2.gpg* | grep -v "md5" | xargs cat | gpg --homedir /usr/local/tbackup/.gnupg/ -r "Total Backup" -d - | bzip2 -q -c -d > [dest_dir]/mysql.dmp
```

Let's assume that you have a backup cycle of seven days and the first backup day is Sunday. To restore the files on server as they were on Wednesday, you have to restore the Sunday archive, the Monday archive and the Wednesday archive.

In order to restore the server to the latest state you must restore all the backup archives in the backup order. 4PSA Total Backup keeps the backup session's files indexed in name incremented directories, so all you have to do is to change to directories in the backup order (1,2,3, ...n) and untar files as described above.



The incremental backup features allow you to restore the server to any previous state saved during the backup cycle.

Restore the MySQL Databases

To restore the MySQL databases you must have the mysql.sql file. If it was compressed and/or encrypted you must first decompress and/or decrypt the mysql dump as described above. In order to load the sql file into Mysql you must issue the command:

```
cat mysqlfile.sql* | /path/to/mysql -uadmin -p`cat  
/etc/psa/.psa.shadow`
```

You can use the `-force` option when errors are encountered during the MySQL files restore. However, this is not recommended.

Restore the PostgreSQL Databases

To restore the PostgreSQL databases you must have the postgresql.sql file. If it was compressed and/or encrypted you must first decompress and/or decrypt the postgresql dump as described above. In order to load the sql file into PostgreSQL you must issue the command:

```
/path/to/psql -f postgresql.sql* template1
```

Due to the nature of SQL databases, they can not be backed up incrementally, so every backup session in the cycle will contain the full MySQL and PostgreSQL database dumps.

4. Low Level Settings

The file `/usr/local/tbackup/paths.cfg` contains five directives. These directives **CANNOT** be modified using the browser interface:

tar_path – The path to tar tool. If the directive is empty or commented the default tar path on the operating system is used. (Example: `tar_path /usr/bin/tar`)

gpg_path – The path to gpg tool. If the directive is empty or commented the default tar path on the operating system is used. (Example: `tar_path /usr/bin/gpg`)

md5_path – The path to md5sum (under RedHat) or md5 (under FreeBSD). If the directive is empty or commented the default path on the operating system is used. (Example: `md5_path /usr/local/md5sum`)

exclude_path – Paths that are excluded during the backup operation, when tar is the used backup tool. Some paths are excluded from the backup in the default installation. If you have an additional HDD or another path that you want to exclude from the server backup in order to save space you must add it here. (Example: `exclude_path /opt /mnt /proc /sbin`)

include_path – Use this setting when you do not want the entire server to be backed up. Only the directories in the `include_path` are saved in the tar file. The exclude path is also considered. The directories required to restore Plesk are `/etc`, `/usr/lib`, `/usr/qmail` and `/usr/local/psa`.

tar_debug_file – In this file all tar operations are logged. This setting is used for debugging purposes and it is recommended to leave it empty.

log_level – Syslog log level used by Total Backup backup agent. All Total Backup logs are written to `/var/log/messages`.



Note

The Local archives directory path is automatically excluded. It is not necessary to add it here!

Appendix A. Server Compatibility

4PSA Total Backup for Plesk 7.x Reloaded is compatible with Plesk 7.x Reloaded installations only.

You have to download the build based on the operating system installed on your machine.

The file `total_backup_buildRedHat7xXXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- RedHat Linux 7.3
- RedHat Enterprise Linux 2.1

The file `total_backup_buildRedHat9xXXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- RedHat Linux 9
- RedHat Enterprise Linux 3.0
- Fedora Linux Core 1
- Fedora Linux Core 2

The file `total_backup_buildFreeBSD4XXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- FreeBSD 4.8
- FreeBSD 4.9

The file `total_backup_buildFreeBSD5XXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- FreeBSD 5.2.1
- FreeBSD 5.3

The file `total_backup_buildSuseXXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- Suse Linux 9
- Suse Linux 9.1

The file `total_backup_buildMandrakeXXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- Mandrake 10

The file `total_backup_buildDebianXXX_Plesk7x.tar.gz` provides compatibility with the following operating systems:

- Debian 3.1