

Veritas NetBackup™ for PostgreSQL Administrator's Guide

Windows and Linux

Release 9.1



Veritas NetBackup™ for PostgreSQL Administrator's Guide

Last updated: 2021-06-07

Legal Notice

Copyright © 2021 Veritas Technologies LLC. All rights reserved.

Veritas, the Veritas Logo, and NetBackup are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third party ("Third-party Programs"). Some of the Third-party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the Third-party Legal Notices document accompanying this Veritas product or available at:

<https://www.veritas.com/about/legal/license-agreements>

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. Veritas Technologies LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

<http://www.veritas.com>

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

<https://www.veritas.com/support>

You can manage your Veritas account information at the following URL:

<https://my.veritas.com>

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

<https://sort.veritas.com/documents>

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1	Introduction to NetBackup for PostgreSQL	6
	About NetBackup for PostgreSQL Agent	6
	Features supported by NetBackup for PostgreSQL Agent	8
	The NetBackup for PostgreSQL Agent package	8
	About the License for NetBackup for PostgreSQL Agent	8
Chapter 2	Installing the NetBackup for PostgreSQL Agent	9
	Planning the installation of NetBackup for PostgreSQL Agent	9
	Verifying the operating systems and platforms	10
	Installing prerequisites for NetBackup for PostgreSQL Agent	10
	Post-installation requirements for NetBackup for PostgreSQL Agent	11
	Describing the NetBackup for PostgreSQL Agent package	12
	Installing the NetBackup for PostgreSQL Agent	12
	Authenticating the password	13
	Uninstalling the NetBackup for PostgreSQL Agent	14
Chapter 3	Configuring NetBackup for PostgreSQL	15
	The nbpgsql.conf configuration file	15
	Configuring PostgreSQL backups with DataStore policies	18
Chapter 4	NetBackup for PostgreSQL backup and restore	20
	About PostgreSQL backups	20
	The postgresql.conf configuration file	23
	Performing PostgreSQL backups	23
	Validating the PostgreSQL backups	25
	Querying the PostgreSQL backups	25
	Deleting backup information from the NetBackup catalog files	26
	About PostgreSQL restore	27
	Performing the PostgreSQL restores	29
	Redirected restores	29

	Recovering the restores	31
	Disaster recovery	33
Chapter 5	Troubleshooting for PostgreSQL	34
	Troubleshooting errors when using NetBackup for PostgreSQL Agent	34
Appendix A	NetBackup for PostgreSQL commands and conventions	41
	About NetBackup for PostgreSQL Agent commands	41
	NetBackup for PostgreSQL Agent command conventions	42
Appendix B	NetBackup for PostgreSQL commands	44
	nbpgsql -o backup	45
	nbpgsql -o restore	47
	nbpgsql -o query	48
	nbpgsql -o delete	49
Index		50

Introduction to NetBackup for PostgreSQL

This chapter includes the following topics:

- [About NetBackup for PostgreSQL Agent](#)
- [Features supported by NetBackup for PostgreSQL Agent](#)
- [The NetBackup for PostgreSQL Agent package](#)
- [About the License for NetBackup for PostgreSQL Agent](#)

About NetBackup for PostgreSQL Agent

NetBackup for PostgreSQL Agent extends the capabilities of NetBackup to include backup and restore of the PostgreSQL databases. The NetBackup for PostgreSQL Agent is located on the NetBackup client and supports the operations on standalone setups. The agent supports the PostgreSQL version 9 and later.

The agent also supports to:

- Validate the backup.
- Query a backup and restore.
- Delete the backup information from the catalog files.
- Redirect restores.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

The NetBackup for PostgreSQL workflow

The NetBackup for PostgreSQL Agent reads the parameters from the `nbpgsql.conf` file and initiates the operation. The `nbpgsql.conf` file contains the parameters that you must set for the `nbpgsql` command to run the respective operations.

See [“The nbpgsql.conf configuration file”](#) on page 15.

The agent communicates with the PostgreSQL database to create a snapshot. The Volume Shadow Copy Services (VSS) for Windows or Logical Volume Manager (LVM) for Linux, takes a snapshot of the PostgreSQL database.

The agent then interacts with the NetBackup XBSA interface to update the server name, policy, and schedule type information. The NetBackup master server connects to the NetBackup client to backup or retrieve the data that you want to protect.

The agent mounts the snapshot, copies the file, and sends it to the NetBackup XBSA interface. The NetBackup XBSA interface writes this data to the mounted media or disk storage that is managed by the NetBackup media server.

[Table 1-1](#) lists the configuration files. These files contain the parameters that you must specify for the respective operations.

Table 1-1 NetBackup for PostgreSQL configuration files

Component	Description
<code>nbpgsql.conf</code> file	The <code>nbpgsql.conf</code> file contains the parameters that you must specify before you run the <code>nbpgsql</code> operations. For more information, See “The nbpgsql.conf configuration file” on page 15.
<code>recovery.conf</code> file	The <code>recovery.conf</code> file contains the parameters that you must set for PostgreSQL to perform an archive recovery. You must set the parameters again for the subsequent recovery that you must perform. For more information, See “Recovering the restores” on page 31.
<code>postgresql.conf</code> file	The <code>postgresql.conf</code> file contains the parameters that you must set to archive the Write-Ahead Logs. For more information, See “About PostgreSQL backups” on page 20.
<code>pgpass.conf</code> file	The <code>pgpass.conf</code> file contains the password to connect to the database for Windows. For more information, See “Authenticating the password” on page 13.
<code>.pgpass</code> file	The <code>.pgpass</code> file contains the password to connect to the PostgreSQL database for Linux operating systems. For more information, See “Authenticating the password” on page 13.

Note: You must set the parameters in the `nbpgsql.conf` file before you run the operations or provide them on the `nbpgsql` command line, where command line parameters take precedence.

Features supported by NetBackup for PostgreSQL Agent

Table 1-2 lists the features that are supported by the agent.

Table 1-2 Features of NetBackup for PostgreSQL

Features	Description
Backup	The agent supports full instance backups of the PostgreSQL database.
Restore	The agent supports full instance restores of PostgreSQL backups.
Redirected restore	The agent supports restoring PostgreSQL backups to alternate NetBackup clients .

The NetBackup for PostgreSQL Agent package

The PostgreSQL agent is packaged in `NBPostgreSQLAgent_8.2.zip` file and is available on my.veritas.com site.

The package file contains the following platform files:

- (Windows) `NBPostgreSQLAgent_8.2_AMD64/`
- (Linux RHEL) `NBPostgreSQLAgent_8.2_linuxR_x86/`
- (Linux SLES) `NBPostgreSQLAgent_8.2_linuxS_x86/`

About the License for NetBackup for PostgreSQL Agent

The NetBackup for PostgreSQL Agent is installed on the NetBackup client software and is not a separately licensed option of NetBackup. The NetBackup for PostgreSQL Agent is available to customers who are entitled with a valid license of the Application and Database License Pack. In general, licensing of the NetBackup for PostgreSQL Agent follows the existing capacity licensing models for supported database agents.

Installing the NetBackup for PostgreSQL Agent

This chapter includes the following topics:

- [Planning the installation of NetBackup for PostgreSQL Agent](#)
- [Verifying the operating systems and platforms](#)
- [Installing prerequisites for NetBackup for PostgreSQL Agent](#)
- [Post-installation requirements for NetBackup for PostgreSQL Agent](#)
- [Describing the NetBackup for PostgreSQL Agent package](#)
- [Installing the NetBackup for PostgreSQL Agent](#)
- [Authenticating the password](#)
- [Uninstalling the NetBackup for PostgreSQL Agent](#)

Planning the installation of NetBackup for PostgreSQL Agent

[Table 2-1](#) lists the planning steps that are mandatory for installing the agent.

Table 2-1 General steps for installing the agent

Step	Action
Step 1	Verify the operating systems. For more information, See “Verifying the operating systems and platforms” on page 10.

Table 2-1 General steps for installing the agent (*continued*)

Step	Action
Step 2	Verify the prerequisites before you install the NetBackup for PostgreSQL Agent. For more information, See "Installing prerequisites for NetBackup for PostgreSQL Agent" on page 10.
Step 3	Install the NetBackup for PostgreSQL Agent on your operating system. For more information, See "Installing the NetBackup for PostgreSQL Agent" on page 12.
Step 4	Authenticate the database password for PostgreSQL backups. For more information, See "Authenticating the password " on page 13.

Verifying the operating systems and platforms

Verify that the NetBackup for PostgreSQL Agent is supported on your operating system or platforms.

The agent supports the operations on following platforms:

- Red Hat Enterprise Linux (RHEL) 6.8 and later
- SUSE Enterprise Linux Server 11 SP3 and later
- Microsoft Windows Server 2012 and later
- Microsoft Windows 8.1 and later

Installing prerequisites for NetBackup for PostgreSQL Agent

Before you install, ensure that you meet the following prerequisites:

- NetBackup 8.2 or later is installed and operational on the master server, media server, and the client.
- Ensure that versions of the PostgreSQL agent and NetBackup are same. If you upgrade NetBackup to newer version, then you must upgrade the agent version also.
- The PostgreSQL database is installed and operational on the client.

Post-installation requirements for NetBackup for PostgreSQL Agent

After you install

- (Windows) Configure the NetBackup for PostgreSQL Agent to run with administrative privileges.
- (Windows) Add the NetBackup `bin` directory to the `PATH` user environment variable.
- (Linux) If the `nbpgsql.conf` file does not exist, create the default configuration file. For more information, See [“The nbpgsql.conf configuration file”](#) on page 15.
- (Linux) Symbolic link: If a symbolic link does not exists, create `libpq.so` that points to `libpq.so.<n>`, where `<n>` is the PostgreSQL library version. You can create the symbolic link at your chosen directory.
 For example, if the PostgreSQL library version is 5, then the symbolic link `libpq.so` points to `libpq.so.5`.

Note: Ensure that you update the `PGSQL_LIB_INSTALL_PATH` parameter in the `nbpgsql.conf` with the absolute path of the symbolic link.

- (Linux) The user of the agent is a superuser or has superuser privileges.
- Set the following database user privileges:

Table 2-2 User and the privileges

User	Privileges
Backup	LOCK TABLES, SELECT FILE, RELOAD, SUPER, UPDATE, TRIGGER, SHOW, VIEW, EXECUTE, and EVENT.
Restore	CREATE, DROP, INDEX, SHUTDOWN, INSERT, ALTER, DELETE, UPDATE, TRIGGER, SUPER, and CREATE VIEW.

To set the database user privileges, run the following PostgreSQL command:

```
ALTER USER<db_user> with SUPERUSER
```

For more information, see *PostgreSQL Administrator's Guide*.

Describing the NetBackup for PostgreSQL Agent package

The agent is packaged in the `NBPostgreSQLAgent_8.2.zip` file and is available on my.veritas.com site.

The package file contains the following platform files:

- (Windows) `NBPostgreSQLAgent_8.2_AMD64/`
- (Linux RHEL) `NBPostgreSQLAgent_8.2_linuxR_x86/`
- (Linux SUSE) `NBPostgreSQLAgent_8.2_linuxS_x86/`

(Windows) `NBPostgreSQLAgent_8.2_AMD64/` includes the following files:

- `NBPostgreSQLAgent_8.2_AMD64/README.txt`
- `NBPostgreSQLAgent_8.2_AMD64/cab1.cab`
- `NBPostgreSQLAgent_8.2_AMD64/Setup.exe`
- `NBPostgreSQLAgent_8.2_AMD64/NBPgSQLAgent.msi`

(Linux RHEL) `NBPostgreSQLAgent_8.2_linuxR_x86/` includes the following file:

- `VRTSnbpostgresqlagent.rpm`

(Linux SUSE) `NBPostgreSQLAgent_8.2_linuxS_x86/` includes the following files:

- `VRTSnbpostgresqlagent.rpm`

When you install the agent, accept the VERITAS LICENSE AGREEMENT to proceed with installing the agent successfully.

The agent by default, gets installed at the following locations:

- (Windows) `C:\Program Files\VERITAS\NBPostgreSQLAgent`
- (Linux RHEL & SUSE) `/usr/NBPostgreSQLAgent/`

Installing the NetBackup for PostgreSQL Agent

When you install the agent on Windows, Veritas recommends that you run the `NBPostgreSQLAgent_8.2_AMD64/Setup.exe` file.

To install the agent

- 1 Download the `NBPostgreSQLAgent_8.2.zip` file.
- 2 Extract the file that applies to your operating system:
(Windows) `NBPostgreSQLAgent_8.2_AMD64/`
(Linux RHEL) `NBPostgreSQLAgent_8.2_linuxR_x86/`
(Linux SUSE) `NBPostgreSQLAgent_8.2_linuxS_x86/`
- 3 Run the file that applies to your operating system:
(Windows) `NBPostgreSQLAgent_8.2_AMD64/Setup.exe`
(Linux RHEL)
`NBPostgreSQLAgent_8.2_linuxR_x86/VRTSnbpostgresqlagent.rpm`
Use the `rpm -ivh VRTSnbpostgresqlagent.rpm` command.
(Linux SUSE)
`NBPostgreSQLAgent_8.2_linuxS_x86/VRTSnbpostgresqlagent.rpm`
Use the `rpm -ivh VRTSnbpostgresqlagent.rpm` command.
- 4 Type `y` to accept the VERITAS LICENSE AGREEMENT.
The agent gets installed at the default location.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

To accept the EULA agreement automatically (silent installation), create a `/tmp/AgentInstallAnswer.conf` file with following content:

Yes - to accept the EULA agreement

No - to decline EULA agreement

Authenticating the password

Authenticating the password keeps you from specifying the password every time you run a backup. The password file stores the password and the application picks the password every time you run a backup.

The password file

The password file for Windows is `pgpass.conf` and for Linux it is `.pgpass` file.

The password file must contain the lines of the following format:

hostname:port:database:username:password

In Linux, after you edit the `.pgpass` file, change the `.pgpass` file permissions.

Authenticating the password on Windows

To authenticate the password

- 1 Run the following command:

```
>echo%AppData%
```

```
O/P: C:\Users\Administrator\AppData\Roaming
```

- 2 Create `postgresql` directory in `C:\Users\Administrator\AppData\Roaming` path.

- 3 Create `pgpass.conf` in the `postgresql` directory.

- 4 In the `pgpass.conf` file update the following and then save the file.

```
hostname:port:database:username:password
```

```
For example, localhost:5432:*:postgres:test_123
```

- 5 Restart the `postgres` services.

Authenticating the password on Linux

To authenticate the password

- 1 Create `.pgpass` file in the user's home directory.

- 2 Edit the `.pgpass` file as:

```
hostname:port:database_name:username:password
```

- 3 To change the `.pgpass` file permissions, run the following command:

```
$ chmod 0600 ~/.pgpass
```

Uninstalling the NetBackup for PostgreSQL Agent

To uninstall the agent

- 1 (Windows) From the **Control Panel**, right-click the `NBPostgreSQLAgent_8.2_AMD64/` file and **Uninstall** the agent.

- 2 (Linux RHEL or SUSE) To uninstall, run the following command:

```
rpm -e VRTSnbpostgresqlagent
```

Configuring NetBackup for PostgreSQL

This chapter includes the following topics:

- [The nbpgsql.conf configuration file](#)
- [Configuring PostgreSQL backups with DataStore policies](#)

The nbpgsql.conf configuration file

The NetBackup for PostgreSQL configuration (`nbpgsql.conf`) file contains the parameters that you must specify for the respective operations. It contains predefined settings and is located on the client. When parameters are not configured in the `nbpgsql.conf` file, then the default parameter value takes precedence. You can configure the parameters in the `nbpgsql.conf` file or set them on the command line, where the command line parameters take precedence. The `nbpgsql.conf` file keeps you from providing the parameters every time you run operations.

The `nbpgsql.conf` file is located in the following locations:

- (Windows)
`install_path\NBPostgreSQLAgent_8.2_AMD64\nbpgsql.conf`
- (Linux RHEL)
`install_path/NBPostgreSQLAgent_8.2_linuxR_x86/nbpgsql.conf`
- (Linux SUSE)
`install_path/NBPostgreSQLAgent_8.2_linuxS_x86/nbpgsql.conf`

Creating the nbpgsql configuration file

Starting from NetBackup 8.2, the `nbpgsql.conf` file is not created by default when you install the agent on RHEL or SUSE. The existing configuration file is prevented

from getting overwritten as the RPM installer simply overwrites any existing files in the destination directory `/usr/NBPostgreSQLAgent/`.

If the `nbpgsql.conf` file does not exist, you can create the file by running the backup utility command without any options. For example, run the `./nbpgsql` command. This command creates the default `nbpgsql.conf` file.

Table 3-1 lists the `nbpgsql.conf` file parameters:

Table 3-1 The `nbpgsql.conf` parameters

Parameters	Description	Required parameter for	Default value
DB_PORT	Configures the PostgreSQL database server port number on which the backup or the restore should be performed. The port number verifies the PostgreSQL service status.	Backups	When you do not specify the port number, the default is 5432.
DB_USER	Configures the PostgreSQL database user name.	Backups	When you do not specify the user name, the default is <code>postgres</code> .
PGSQL_LIB_INSTALL_PATH	(Linux) Configures the <code>libpq.so</code> binary path.	Backups	No default value.
MASTER_SERVER_NAME	Specifies the NetBackup master server for backups and restores.	Backups and restores	No default value.
POLICY_NAME	Specifies the DataStore policy name.	Backups	No default value.
SCHEDULE_NAME	Identifies the backup schedule that you configured while creating the DataStore policy.	Backups	When you do not specify the policy name, the defaults is Application Backup .
CLIENT_NAME	Defines the NetBackup client that has the NetBackup for PostgreSQL Agent.	Redirected restores and query	When you do not specify the client name, the default is the current host.
SNAPSHOT_SIZE	(Linux) Specifies the snapshot size for LVM snapshots. Specify the snapshot size in kilobytes, megabytes, or gigabytes as KB, MB, or GB respectively.	LVM backups	When you do not specify the snapshot size, the default is MB.

Table 3-1 The nbpgsql.conf parameters (*continued*)

Parameters	Description	Required parameter for	Default value
DB_BACKUP_ID	DB_BACKUP_ID represents the backup image name. This parameter configures the backup file when you want to restore or delete a specific backup.	Restoring backups using the backup image name.	No default value.
PGSQL_TARGET_DIRECTORY	Specifies the destination directory where you want to restore the backups.	Restores	No default value.
NBPGSQL_LOG_LEVEL	The NBPGSQL_LOG_LEVEL parameter lets you set the logging level for the <code>nbpgsql</code> logs. For a particular logging level, all details at that level or lower are logged. The <code>nbpgsql</code> debug logs includes the following verbose levels: ■ 1 – ERROR: Conditions that should be corrected, such as configuration errors. ■ 2 – WARN: Conditions that are not errors, but that might require special handling. ■ 3 – INFO: Informational messages. ■ 4 – DEBUG: Debugging the messages that is used for troubleshooting.	The log levels help to control the amount of information that you want to access for troubleshooting errors.	When you do not specify the log level, the default is level 1.
NBPGSQL_LOG_SIZE	Configures the <code>nbpgsql</code> log size in MB. The maximum size is 10MB. When the log size reaches the NBPGSQL_LOG_SIZE , it creates a new log with timestamp.	You can specify the value according to the events that you want to write into the logs.	When you do not specify the log size, the default is 10MB.

Table 3-1 The nbpgsql.conf parameters (*continued*)

Parameters	Description	Required parameter for	Default value
BACKUP_TYPE	Available options: ▪ auto: Default option. Performs an auto discovery backup. ▪ lvm: Agent forces to do a lvm snapshot. ▪ nonlvm: Agent forces to do nonlvm way backup by using <code>pg_start_backup</code> and <code>pg_stop_backup</code> for Postgres with LVM snapshot disabled.	Backup operations	auto

Configuring PostgreSQL backups with DataStore policies

The agent uses the DataStore policies to define the attributes, schedules, clients list, and backup selections.

To configure the PostgreSQL database backups with DataStore policies, complete the following steps:

- 1 Log on to the master server as an administrator (Windows) or root (Linux).
- 2 In the **NetBackup Administration Console**, expand **NetBackup Management**, and then click **Policies**.
- 3 In **All Policies** pane, right-click **Summary of All Policies**, and then click **New Policy**.
- 4 In **Add a New Policy** dialog box, enter the unique policy name.
- 5 In the **Change Policy** dialog box, select **DataStore Policy** from the **Policy Type** drop-down list.
- 6 From the **Policy Storage** list, select a **disk-based storage unit** for storage.
- 7 To select the schedule type, under the **Schedules** tab, click **OK** to select the **Application Backup** schedule type.

Note: The XBSA framework supports the **Application backup** schedule type only.

- 8 In the **Clients** tab, click **New** and then add the NetBackup client that has the **NetBackup for PostgreSQL Agent**.
- 9 In the **Add Client** screen, click **New**, and then in the **Client Name** field, type the name of the client.
- 10 In the **NetBackup Administration Console**, click **NetBackup Management > Policies** to view the policy in the existing policies list.
- 11 Before performing the backup, review the settings in the `nbpgsql.conf` file.
For more information, see See [“The nbpgsql.conf configuration file”](#) on page 15.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

NetBackup for PostgreSQL backup and restore

This chapter includes the following topics:

- [About PostgreSQL backups](#)
- [Performing PostgreSQL backups](#)
- [Validating the PostgreSQL backups](#)
- [Querying the PostgreSQL backups](#)
- [Deleting backup information from the NetBackup catalog files](#)
- [About PostgreSQL restore](#)
- [Performing the PostgreSQL restores](#)
- [Redirected restores](#)
- [Recovering the restores](#)
- [Disaster recovery](#)

About PostgreSQL backups

The `nbpgsql -o backup` command for backup initiates the backup operation using the `-S`, `-P`, and `-s` as the required parameters. The parameters `-l` and `-z` are the required parameters for Linux operating systems.

The parameters `portnum` and `-u` are optional parameters. You can configure these parameters in the `nbppgsql.conf` file or specify on the `nbppgsql` command line.

After you set the parameters for the backup, the agent reads the parameters and starts the backup according to the specified parameters. The agent writes the data that you want to protect into the WAL files.

These WAL files are then archived in the archive directory that you can create at your chosen location.

When you create the archive or WAL directory, Veritas recommends that you create outside the data directory.

Ensure that before you run a backup, set the parameters in the `postgresql.conf` file to enable WAL archiving.

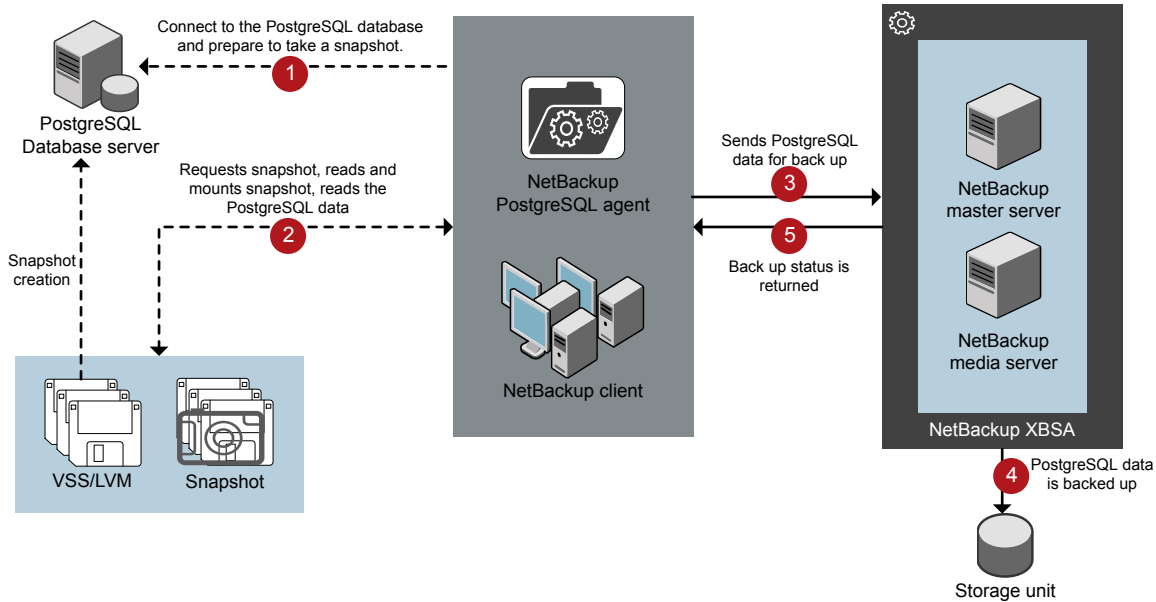
For more information, see See [“The postgresql.conf configuration file”](#) on page 23.

The agent protects the following files:

- Schema files that are associated with all database tables.
- Files that are associated with the database tables.
- Data and index files.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

Figure 4-1 NetBackup for PostgreSQL backup workflow



The NetBackup for PostgreSQL workflow

When you run the backup, the agent connects to the PostgreSQL database to execute a flush and read only lock on all tables. The agent then reads the associated PostgreSQL database files from the mounted directory and initiates the backup.

The LVM or VSS, creates a snapshot, and mounts the snapshot. The associated files (whole instance) are archived into file. The agent copies the archived file into the XBSA data object and sends to the NetBackup XBSA interface.

The NetBackup XBSA interface writes this data to the mounted media or disk storage managed by the NetBackup media server.

The command prompt displays the successful completion status of the backup. The **Activity Monitor** also displays the status for the backup job.

The postgresql.conf configuration file

The `postgresql.conf` file contains the parameters that you must set to enable WAL archiving before you run a backup.

[Table 4-1](#) table lists the parameters that you must set to archive the WAL logs.

Table 4-1 The `postgres.conf` parameters to enable WAL archiving

Parameters	Description
wal_level	This parameter determines how much information is written to the WAL files.
archive_mode	This parameter enables the archive mode so that the WAL logs get stored in the archive directory using the <code>archive_command</code> .
archive_timeout	This parameter sets the number of seconds after which the log file segment will switch to a new segment.
statement_timeout	This parameter aborts any statement that takes more than the set number of milliseconds.

Performing PostgreSQL backups

This topic lists the prerequisites for the backup, describes the procedure to run a backup, and the information to schedule the backup from NetBackup.

Prerequisites

Before you run the backup, ensure that you meet the following prerequisites:

- Ensure that versions of the PostgreSQL agent and NetBackup are same. If you upgrade NetBackup to newer version, then you must upgrade the agent version also.
- Ensure that the user has administrator (Windows) or root (Linux) access.
- (Windows) Set the `NetBackup\bin` directory in the environment variable.
For example, `Path =C:\Program Files\Veritas\Netbackup\bin`
- (Windows) Set the `PostgreSQL\bin` directory in user environment variable.
For example, `Path=C:\Program Files\PostgreSQL\bin`
- (Linux) Symbolic link: If a symbolic link does not exists, create the symbolic link `libpq.so` and ensure that it points to the valid `libpq .so.<n>`, where `n` is the PostgreSQL library version.
For more information, See [“Post-installation requirements for NetBackup for PostgreSQL Agent”](#) on page 11.

- Set the following parameters in the `nbpgsql.conf` file:
 - `DB_USER`
 - `DB_PORT`
 - `MASTER_SERVER_NAME`
 - `POLICY_NAME`
 - `SCHEDULE_NAME`
 - (Linux) `PGSQL_LIB_INSTALL_PATH`
 - (Linux) `SNAPSHOT_SIZE`
- Create the `archivedir` directory and then set the following parameters in the `postgresql.conf` file:
 - `wal_level = archive`
 - `archive_mode = on`
 - `archive_timeout = 0`
 - `statement_timeout = 0`

Note: Ensure that you add the time in *milliseconds*. The recommended time is 30000 milliseconds (30 seconds).

- Mention the following changes for `archive_command`
 - (Windows) `'copy ' "%p" "C:\\archivedir\\%f"'`
 - (Linux) `test ! -f <archive_path>/%f && cp %p <archive_path>/%f`
- (Linux) After creating the `archivedir` directory, change the group and ownership to PostgreSQL user.
- Restart the PostgreSQL services.
- Configure PostgreSQL backups with **DataStore** policies.
For more information, See [“Configuring PostgreSQL backups with DataStore policies”](#) on page 18.
- Verify the installation prerequisites and the post-installation requirements.
For more information, See [“Installing prerequisites for NetBackup for PostgreSQL Agent”](#) on page 10.
For more information, See [“Post-installation requirements for NetBackup for PostgreSQL Agent”](#) on page 11.

To run the backup

- 1 Run the following command:

```
nbpgsql -o backup  
  
-S master_server  
  
-P policy_name  
  
-s schedule_name  
  
(Linux)-z snapshot_size  
  
(Linux)-l postgresql_library_path  
  
[-portnum db_port]  
  
[-u dbuser]  
  
(Linux)[-b backup_type]
```

- 2 (Optional) Type the database password, when the command line prompts for a password . NetBackup connects to the database and initiates the backup.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

Scheduling PostgreSQL backups from NetBackup

You can schedule the PostgreSQL backups from the **NetBackup Administration Console** using the **DataStore** policy to call a backup script.

For more information, see https://www.veritas.com/support/en_US/article.100041371

Validating the PostgreSQL backups

After a successful backup, you can view and verify the backup information using the following command:

```
nbpgsql -o query
```

Querying the PostgreSQL backups

The `nbpgsql -o query` command lists previously backed up files according to the options that you specify. You can configure these parameters from the `nbpgsql.conf` file or set the parameters using the `nbpgsql` command.

The parameter `-s` is the required parameter. You can use the `-c` and `-P` options to define a different client and policy.

By default, NetBackup uses the values that you have configured in the `nbpgsql.conf` file.

Before you run a query, you must set the following parameter in the `nbpgsql.conf` file or provide on the command line:

- `CLIENT_NAME`
- `POLICY_NAME`

To query a backup

- 1 Configure the parameters, in the `postgresql.conf` file or provide on the command line.
- 2 Run the following command:

```
nbpgsql -o query -S master_server [-C client_name] [-P
policy_name]
```

For example, to query a backup from client `ClientA`, run the following command:

```
nbpgsql -o query -S master_server [-C ClientA]
```

For example, to list backup files with the policy name `policy_name`, run the following command:

```
nbpgsql -o query -S master_server [-P policy_name]
```

Deleting backup information from the NetBackup catalog files

The `nbpgsql -o delete` command, removes the backup information from the catalog files but retains the backup files on the NetBackup media server. The parameter `-s` is required parameter. You can use the `-id` option to delete a backup by specifying its backup image name.

Specify the following parameter in the `nbpgsql.conf` file or on the command line:

- `DB_BACKUP_ID`

To delete the backup information

- 1 Configure the parameters, in the `postgresql.conf` file or provide on the command line.
- 2 Run the following command:

```
nbpgsql -o delete -S master_server [-id db_backup_id].
```

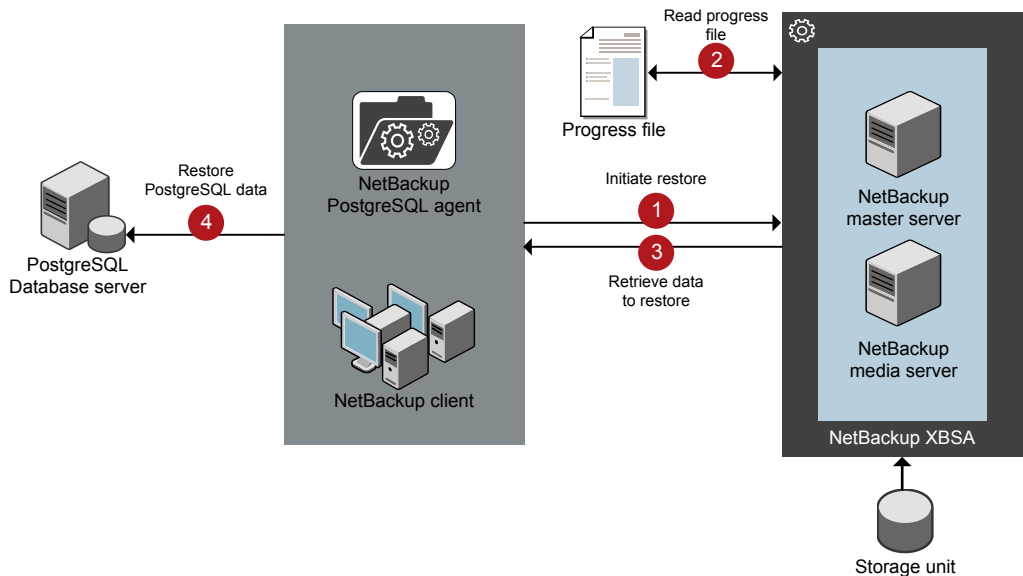
About PostgreSQL restore

The `nbpgsql -o restore` command for restore initiates the restore operation using `-s` and `-t` as the required parameters. The parameters `-id` and `-c` are optional parameters.

The parameter `-id` restores the backup using the specified backup image name. The parameter `-c` lists all the backups that exists on the specified client. When you do not specify `-c`, it defaults to NetBackup master server.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

Figure 4-2 NetBackup for PostgreSQL restore workflow



The NetBackup for PostgreSQL restore workflow

After you specify the parameters, the agent reads the command line arguments and parses the `nbpgsql.conf` configuration file. The agent then interacts with the NetBackup XBSA interface to retrieve the backup using the specified parameters.

The NetBackup XBSA interface reads the progress files to receive the PostgreSQL backup files to restore them to the target directory.

The command prompt indicates the successful completion status of the restore. The **Activity Monitor** displays the status for the restore job.

In Linux operating systems, after a successful restore, the owner and group of the restored data defaults to `postgres`. You must change the ownership to PostgreSQL user and modify the settings according to your environment.

In Linux operating system, if the data directory contains symbolic link, the backup does not retain the link information. The symbolic link gets backed up as a normal data directory and is restored as a normal directory. To restore the link, you must reconfigure the symbolic link.

Note: Ensure that the target directory is valid and empty.

Prerequisites

Before you run a restore, ensure that you meet the following prerequisites:

- Ensure that versions of the PostgreSQL agent and NetBackup are same. If you upgrade NetBackup to newer version, then you must upgrade the agent version also.
- Ensure that the user has administrator (Windows) or root (Linux) access.
- (LVM users) Ensure that data logs and the logs directory reside on the logical volume.
- Set the following parameters in the `nbpgsql.conf` file:
 - `PGSQL_TARGET_DIRECTORY`
 - `MASTER_SERVER_NAME`
 - `DB_PORT`
 - `CLIENT_NAME`
 - `DB_BACKUP_ID`

Performing the PostgreSQL restores

To restore the backup

- 1 Configure the parameters in the `nbpgsql.conf` file, or provide them on the command line.
- 2 Run the following command:

```
nbpgsql -o restore -S master_server -t target_directory [-id  
db_backup_id] [-C client_name]
```

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

Redirected restores

Redirected restores lets you restore backup files to a client different from the client that originally performed the backup. The new location can be a different host or a different file path using a different name for the redirected restore. To redirect a

restore to a different host, include the destination client name in the `install_path\NetBackup\db\altnames` directory.

Note: Ensure that the PostgreSQL agent and NetBackup are of same version for successful backup and restore operations.

Performing redirected restores

To redirect a restore to a different host

- 1 Update the `nbpgsql.conf` file with the NetBackup client name as the host and the PostgreSQL target directory as the directory where you want to redirect the restore.
- 2 On the NetBackup master server, create an `altnames` directory for the host that you want to have permission to perform the redirected restore. For example, to give Host B permissions to restore from another host, create the following file:
 - (Windows) `install_path\NetBackup\db\altnames\HostB`
 - (Linux RHEL and SLES) `/usr/openv/netbackup/db/altnames/HostB`
- 3 In the `altnames` directory, add the names of the client(s) whose files the requesting client wants to restore. For example, if you want Host B to have permissions to redirect restores from Host A, add Host A to the Host B file.
- 4 Run the following command:


```
nbpgsql -o restore -S master_server_name -t target_directory
-portnum db_port [-id db_backup_id] [-C client_name]
```
- 5 After a successful redirected restore, undo the changes that you made on the master server and the client.

To redirect a restore to a different file path

- 1 Run the following command:


```
nbpgsql -o restore -S master_server_name -t target_directory
-portnum database_server_port [-id db_backup_id] [-C client_name]
```
- 2 Copy the restore data to the data directory.
- 3 After a successful restore, change the ownership of data directory to PostgreSQL user and modify the settings according to your environment.

Recovering the restores

Select a recovery workflow based on the version of PostgreSQL you are using. If you are using PostgreSQL version 12 or later, go directly to the section [Recovering for PostgreSQL version 12 or later](#).

Recovering for PostgreSQL version 11 or earlier

After a successful restore, to recover the restore, copy the `recovery.conf.sample` file to the PostgreSQL data directory. The `recovery.conf.sample` is available at the PostgreSQL install path. Ensure that after you copy the recovery file, remove the `.sample` extension.

When you set the parameters and restart the PostgreSQL services, the server goes into the recovery mode and reads the archived WAL files. If the recovery gets terminated, you can restart the server to continue the recovery process.

After successful completion, the server renames the `recovery.conf` file to `recovery.done` to prevent re-entering into the recovery mode.

In Linux operating system, the owner and group defaults to `postgres` after a successful restore. You must change the ownership to PostgreSQL owner and modify the settings according to your environment.

The recovery.conf configuration file

The `recovery.conf` file contains the parameters that you must set to enable archive recovery or act as a replication standby. The parameters must be set again for the subsequent recovery that you must perform.

[Table 4-2](#) lists the parameters that you must set to enable archive recovery.

Table 4-2 The `recovery.conf` file parameters

Parameters	Description
<code>restore_command</code>	This parameter specifies the shell command that is executed to copy log files back from archival storage. This parameter is required for archival storage but is optional for streaming replication. The command string may contain <code>%f</code> that is replaced by the name of the desired log file and <code>%p</code> is replaced by the absolute path to copy the log file to.
<code>recovery_target</code>	This parameter stops the roll-forward at a specific point. By default, the recovery rolls forward to the end of the WAL log.

Performing the restore recovery

To recover the restore

- 1 Stop the PostgreSQL services.
- 2 Copy the restore data into the PostgreSQL data directory.
- 3 (Linux) Change the ownership to PostgreSQL user.
- 4 Copy the `recovery.conf` file to the PostgreSQL data directory and remove the `.sample` extension .
- 5 (Linux) Change the ownership to the PostgreSQL user and modify the settings according to your environment.
- 6 Edit the `recovery.conf` file to set the following:
 - (Windows) Mention the `restore_command` parameter as `cp "<PostgreSQL-data-directory>\pgarchive\\%f" "%p"`
 - (Linux) Mention the `restore_command` as `cp <PostgreSQL-data-directory>/pgarchive/%f %p`
 - Remove the `pause_recovery_target` parameter.
- 7 Start the PostgreSQL services.
- 8 After successful recovery, delete the `pgarchive` directory and the `recovery.done` file.

Recovering for PostgreSQL version 12 or later

If you are using PostgreSQL version 12 or a higher version, perform the following steps.

To recover the restore

- 1 Stop the PostgreSQL services.
- 2 Copy the restore data into the PostgreSQL data directory.
- 3 (Linux) Change the ownership to PostgreSQL user.
- 4 (Windows) Mention the `restore_command` parameter as `cp "<PostgreSQL-data-directory>\pgarchive\\%f" "%p"`
- 5 (Linux) Mention the `restore_command` as `cp <PostgreSQL-data-directory>/pgarchive/%f %p`
- 6 Create an empty file named `recovery.signal` in the PostgreSQL data directory. For example, # `touch /<target_restored_directory>/recovery.signal`

- 7 (Linux) Change ownership of the **recovery.signal** file to the PostgreSQL user.
For example: `chown postgres:postgres
/<PostgreSQL-data-directory>/recovery.signal`
- 8 Start the PostgreSQL services.

Disaster recovery

Disaster recovery is a plan to recover the data that can get lost in a disaster event. The agent supports redirected restore as a disaster recovery strategy.

For more information, See [“Redirected restores”](#) on page 29.

Troubleshooting for PostgreSQL

This chapter includes the following topics:

- [Troubleshooting errors when using NetBackup for PostgreSQL Agent](#)

Troubleshooting errors when using NetBackup for PostgreSQL Agent

General guidelines to resolve problems

The following table includes steps that help you resolve problems you may encounter while using NetBackup for PostgreSQL Agent.

Table 5-1 General steps to resolve problems

Steps	Action	Description
Step1	Remember the error message	Error messages are usually the vehicles for telling you something went wrong. If you do not see an error on the command line, but still suspect a problem, check the logs and the reports. These can provide an error message that directly points to the problem. The logs and reports are essential troubleshooting tools.

Table 5-1 General steps to resolve problems (*continued*)

Steps	Action	Description
Step 2	Identify what you were doing when the problem occurred.	Ask the following questions: ■ What operation was tried? ■ What method did you use? ■ What type of server platform and operating system was involved? ■ If your site uses both master server and media server, was it a master server or a media server? ■ If a client was involved, what type of client was it? ■ Have you performed the operation successfully in the past? If so, what is different now? ■ What is the service pack level? ■ Do you use operating system software with the latest fixes supplied, especially those required for use with NetBackup? ■ Is your device firmware at a level, or higher than the level, at which it has been tested according to the posted device compatibility lists?
Step 3	Record all information	Capture potentially valuable information. ■ The NetBackup logs. ■ The logs specific to NetBackup for PostgreSQL Agent logs. ■ The logs specific to NetBackup XBSA .
Step 4	Correct the problem	After you define the problem, use the information to correct it.
Step 5	Contact Technical Support	If you cannot solve the troubleshooting, contact the Technical support.

Troubleshooting errors using logs and reports

To troubleshoot the errors, you can refer to the NetBackup logs, NetBackup for PostgreSQL Agent logs, and the NetBackup XBSA logs. These logs are located at the following locations:

The NetBackup logs are located at:

- `install_path\NetBackup\logs\bprd`
- `install_path\NetBackup\logs\bpcd`
- `install_path\NetBackup\logs\user_ops\dbext\logs`

You must enable the `bprd` and the `bpcd` log files. For more information, see the *NetBackup Troubleshooting Guide*.

The logs that are specific to NetBackup for PostgreSQL Agent are located at:

- `install_path\nbpgsql.log`

The logs that are specific to NetBackup XBSA are located at:

- <NetBackup_install_path>/netbackup/logs/exten_client

Troubleshooting NetBackup errors

For troubleshooting NetBackup errors, see *Veritas NetBackup Troubleshooting Guide* and the *Veritas NetBackup Commands Reference Guide*

Troubleshooting NetBackup for PostgreSQL Agent errors

[Table 5-2](#) lists and describes the errors and the solutions to troubleshoot the problems while running the operations.

Table 5-2 Troubleshooting NetBackup for PostgreSQL errors

Problems	Description	Solution
The nbpgsql backup fails with the following error: <i>Unable to load postgresql library</i>	You may encounter this problem when the nbpgsql.conf file is not updated with the following ■ postgresql library file location ■ PGSQL_LIB_INSTALL_PATH does not point to correct libpq.so library version.	Verify the following and then run the backup again: ■ Add or update the PostgreSQL library file location in the nbpgsql.conf file. ■ Ensure that the PostgreSQL_LIB_INSTALL_PATH is set to the correct path. It should point to libpq.so library version. ■ Create a symbolic link libpq.so that points to the libpq.so.<n> where n is the PostgreSQL library version. For more information, See “Post-installation requirements for NetBackup for PostgreSQL Agent” on page 11.
The nbpgsql backup fails with the following error: <i>Unable to connect to the database</i>	The nbpgsql backup fails when the nbpgsql.conf file is updated with invalid username or port number.	To add the appropriate database user name and port number ■ Configure the appropriate database user name and port number in the nbpgsql.conf file or provide the appropriate options with the nbpgsqlcommand. For more information, See “The nbpgsql.conf configuration file” on page 15.
The nbpgsql backup fails with the following error: <i>Unable to load xbsa.dll</i>	The nbpgsql backup fails if the environment variable path is not updated with NetBackup bin directory.	To run a nbpgsql backup successfully ■ Update the environment variable path with NetBackup_install_path/bin.

Table 5-2 Troubleshooting NetBackup for PostgreSQL errors (*continued*)

Problems	Description	Solution
The <code>nbpgsql</code> backup fails with the following error: <i>XBSA initiation failed</i>	The <code>nbpgsql</code> backup fails if the <code>nbpgsql.conf</code> file is not updated with the required parameters.	To run the <code>nbpgsql</code> backup successfully ■ Configure the valid master server name, policy name, schedule type in the <code>nbpgsql.conf</code> file or from the command line. ■ Verify if there are communication errors between the <code>nbpgsql</code> agent and the NetBackup master server. For more information see the <i>NetBackup Administration guide</i>.
(Windows) <i>VSS snapshot creation failed</i>	The <code>nbpgsql</code> backup may fail when the user does not have the privileges to run the <code>nbpgsql</code> operations.	Run <code>cmd.exe</code> in Administrator mode.
The <code>nbpgsql</code> restore operation does not restore any data from the target NetBackup client.	The <code>nbpgsql</code> restore fails if the <code>nbpgsql.conf</code> file is not updated with the NetBackup client name and the target directory.	For a successful restore ■ Verify that the target directory is valid and empty. ■ Initiate the restore from the NetBackup source client. ■ Set the NetBackup client name and target directory parameters in the <code>nbpgsql.conf</code> file.
The <code>nbpgsql</code> backup fails with the following error: (Linux) <i>Error creating LVM snapshot</i>	The <code>nbpgsql</code> backup may fail when the volume group does not have sufficient space for the snapshot.	To verify the space in the volume group 1 Run the following command: <pre>\$vgs</pre> The command displays the volume group details. 2 Update the <code>nbpgsql.conf</code> file with the appropriate snapshot size. The snapshot should be equivalent to or more than the instance size.

Table 5-2 Troubleshooting NetBackup for PostgreSQL errors (*continued*)

Problems	Description	Solution
Error messages after a successful backup: <pre><volume_group>/<snapshot_name> Read failure after 0 of 4096 at 29393616896: input or output error. OR <pre><volume_group>/<snapshot_name>: read failure after 0 of 4096 at 4096: input or output error.</pre> </pre>	The <code>nbpgsql</code> backup gives these errors when the volume group contains the snapshots. You can list the snapshots and then remove them before you run the backup again. Note: <code>nbpgsql</code> created LVM snapshot names are prefixed with <code>pgsqlsnap</code>	To remove the snapshots 1 Run the following command to list the existing snapshot: <pre>\$lvs</pre> The command displays the snapshot details. 2 To remove the snapshots, run the following command: <pre>\$ lvremove -f <volume_group>/<snapshot_name></pre>
The <code>nbpgsql</code> backup fails with the following error: <i>"Failed to load PostgreSQL Library"</i>	You may encounter this problem when the <code>nbpgsql.conf</code> file is not updated with the following: ■ PostgreSQL library file location. ■ The <code>PGSQL_LIB_INSTALL_PATH</code> does not point to <code>libpq.so</code>.	Verify the following and then run the backup again: ■ Add or update the PostgreSQL library file location in the <code>nbpgsql.conf</code> file. For more information, See "The nbpgsql.conf configuration file" on page 15. ■ Ensure that the <code>PGSQL_LIB_INSTALL_PATH</code> is set to the correct path. It should point to <code>libpq.so.<n></code>, where <code>n</code> is the PostgreSQL library version. ■ Create a symbolic link <code>libpq.so</code> that points to the <code>libpq.so.<n></code>, where <code><n></code> is the PostgreSQL library version.

Table 5-2 Troubleshooting NetBackup for PostgreSQL errors (*continued*)

Problems	Description	Solution
The <code>nbpgsql</code> backup on Linux (LVM), fails with the following error: <i>Error unmounting the snapshot-Device or resource busy</i> OR <i>Error removing the snapshot-pgsqlsnap_<timestamp></i> Note: <timestamp> is the LVM snapshot time	The <code>nbpgsql</code> backup fails during an attempt to unmount the snapshot, the device, or when you remove the existing snapshots.	To unmount the snapshot 1 Run the following command to list all mounted file systems: <pre>\$ mount-l</pre> 2 If the snapshot still exists, create a mount directory using the following command: <pre>\$mount<mount_directory></pre> Note: This directory is created in <code>/mnt/<snapshot_name></code>. The prefix names for snapshot are <code>pgsqlsnap</code>. 3 Run the following command to remove the mount directory: <pre>\$rm -rf <mount_directory></pre> 4 Run the following command to remove the snapshot manually : <pre>lvremove -f <volume_group>/<snapshot_name></pre>
Even after a successful restore, the PostgreSQL services, failed to start.	The restore operation is successful, only when you restore the backup on a machine that has the same minor version of PostgreSQL. For example, if you back up a file from PostgreSQL version 9.6.x, then you must restore the file to a computer with PostgreSQL version 9.6.x.	■ Verify that the PostgreSQL agent and NetBackup are of same version for successful restore operations. ■ Verify that the PostgreSQL version from the backed up data is same as the PostgreSQL version on the computer where you want to restore the data.

Table 5-2 Troubleshooting NetBackup for PostgreSQL errors (*continued*)

Problems	Description	Solution
The <code>nbpgsql.conf</code> file is missing after installing the agent on RHEL or SUSE	Starting from NetBackup 8.2, the <code>nbpgsql.conf</code> file is not created by default when you install the agent on RHEL or SUSE. The existing configuration file is prevented from getting overwritten as the RPM installer simply overwrites any existing files in the destination directory <code>/usr/NBPostgreSQLAgent/.</code>	If the <code>nbpgsql.conf</code> file does not exist, you can create the file by running the backup utility command without any options. For example, <code>./nbpgsql</code> . This command creates the default <code>nbpgsql.conf</code> file.

NetBackup for PostgreSQL commands and conventions

This appendix includes the following topics:

- [About NetBackup for PostgreSQL Agent commands](#)
- [NetBackup for PostgreSQL Agent command conventions](#)

About NetBackup for PostgreSQL Agent commands

This section describes the commands, options, and parameters that are available to run the `nbpgsql` operations. Each command contains a brief description, required parameters, and optional parameters for the respective operations. The NetBackup for PostgreSQL Agent supports only those commands, options, and parameters that are mentioned in this document.

Observe for the following:

- Specify the parameters in the `nbpgsql.conf` file or provide them on the command line.
- The parameter on the command line take precedence over the `nbpgsql.conf` file.
- Specify the operation type `-o` on the `nbpgsql` command line.
- Specify the parameters and options for the respective operations in the `nbpgsql.conf` file or on the command line.

The NetBackup for PostgreSQL Agent command options

Table A-1 The `nbpgsql` command options

Options	Description
-C	Configures the NetBackup client name for redirected restores.
-h	Displays the Help usage, when it is the only option on the <code>nbpgsql</code> command line.
-id	Configures the specified backup.
-l	(Linux) Configures the PostgreSQL library path.
-o	Configures the operation type (backup, restore, query, and delete).
-P	Configures the DataStore policy.
-portnum	Configures the database server port number that identifies the PostgreSQL instance on which the backup or restore is performed.
-s	Configures the NetBackup schedule.
-S	Configures the NetBackup master server.
-t	Configures the target directory to restore the data.
-u	Configures the database user name.
-z	Configures the LVM snapshot size.
-b	Configures the backup type as LVM or non-LVM

NetBackup for PostgreSQL Agent command conventions

This document uses the following conventions when describing commands that are specific to the agent.

Run the following commands in the command line interface to see the results:

- The `-help` command (`-h`) option prints a command-line usage message when it is the only option on the command line. For example,

```
nbpgsql -h
```

- Brackets [] indicate that the enclosed component of the command line is optional. Other parameters are required.

- Italics indicate that the information is user supplied. For example, you may provide the client name and the schedule name for a backup operation.

```
nbpgsql -o backup -S master_server -P policy_name -s schedule_name
```

NetBackup for PostgreSQL commands

This appendix includes the following topics:

- [nbpgsql -o backup](#)
- [nbpgsql -o restore](#)
- [nbpgsql -o query](#)
- [nbpgsql -o delete](#)

nbpgsql -o backup

`nbpgsql -o backup` – runs the backup operation from the NetBackup client.

SYNOPSIS

```
nbpgsql -o backup
-S master_server
-P policy_name
-s schedule_name
(Linux) -l postgresql_library_path
(Linux) -z snapshot_size
[(Linux) -b backup_type auto, lvm, and nonlvm]
[-portnum db_port]
[-udatabase_user]
```

Description

This command invokes the backup operation from the NetBackup client using the NetBackup DataStore policy name and the schedule type. The parameter `-s` and `-P` are required parameters for Windows. The parameters `-b`, `-l`, and `-z` are required parameters for Linux. The `-portnum` and `-u` are the optional parameters.

On Linux systems, the directory path is `/usr/NBPostgreSQLAgent/`.

On Windows, the directory path is `install_path\NBPostgreSQLAgent\`.

Options

`-l`
(Linux) Configures the PostgreSQL library directory

`-portnum`
Configures the database port number that identifies the PostgreSQL instance on which the backup is performed.

`-P`
Configures the NetBackup DataStore policy name.

- S
Configures the NetBackup server name.
- s
Specifies the schedule name that you have configure for the **DataStore** policy.
- u
Configures the database user name.
- z
(Linux) Specifies the LVM snapshot size.
- b Configures the backup type as LVM or non-LVM.

nbpgsql -o restore

`nbpgsql -o restore` – restores the backup files from the NetBackup server.

SYNOPSIS

```
nbpgsql -o restore -S master_server -t target_directory [-id  
db_backup_id] [-Cclient_name]
```

Description

The `nbpgsql` command restores the backup file using `-t` and `-S` as the required parameters. The `-id` and `-C` are optional parameters.

On Linux systems, the directory path to this command is `/usr/NBPostgreSQLAgent/`

On Windows systems, the directory path to this command is
`install_path/NBPostgreSQLAgent/`

Options

- `-C`
Specifies the client name.
- `-id`
Specifies the backup image name.
- `-S`
Configures the NetBackup master server.
- `-t`
Configures the target directory where the backups are restored.

nbpgsql -o query

`nbpgsql -o query` – query the backup.

SYNOPSIS

```
nbpgsql -o query -S master_server [-C NetBackup_client_name] [-P  
policy_name]
```

Description

The `nbpgsql -o query` command gets the backup using `-s` as the required parameter and `-c` and `-p` as optional parameters.

On Linux systems, the directory path to this command is `/usr/PostgreSQLAgent/`

On Windows systems, the directory path to this command is
`install_path/NBPostgreSQLAgent/`

Options

- `-c` Retrieves and lists all the backups of the specified client.
- `-p` Retrieves and lists all backups with the specified policy name.
- `-s` Configures the NetBackup master server.

nbpgsql -o delete

`nbpgsql -o delete` – deletes the backup information from the NetBackup catalog files.

SYNOPSIS

```
nbpgsql -o delete -S master_server[-id db_backup_id]
```

Description

The `nbpgsql -o delete` command deletes the backup information from the NetBackup catalog files, but retains the backups in the storage media.

The parameter `-s` is a required parameter and `-id` is the optional parameter.

Options

`-id`

Specifies the backup image name to delete the specified backup information.

`-S`

Configures the NetBackup master server.

Index

A

authenticating password 9

B

backups

archive directory 21

delete 21

query 21

schema files 21

symbolic link 21

Write-Ahead logs level 21

I

installation

privileges 9

symbolic link 9

N

nbpgsql file

configuration file 15

configuration parameters 15

database server port 15

log level 15

log size 15

P

PostgreSQL Agent

components 6

features 6

license 6

package 6

R

redirected restore

altnames 29

different file path 29

different host 29

restore

target directory 29

restores

recovery 29

redirected restore 29

U

uninstall 9