

Vastbase G100

工具参考

(Version 2.2)

【版权声明】

©2007-2020 北京海量数据技术股份有限公司 版权所有

本文档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

北京海量数据技术股份有限公司保留所有的权利。

【商标声明】



及其它海量数据产品和服务相关的商标均为 **北京海量数据技术股份有限公司** 及其关联公司所有。

本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍海量数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本文档内容不做任何明示或模式的承诺或保证。

目录

1. 工具一览表.....	
2. 客户端工具.....	
2.1. vsql.....	
2.1.1. vsql 概述.....	
2.1.2. 使用指导.....	
2.1.3. 获取帮助.....	
2.1.4. 命令参考.....	
2.1.5. 元命令参考.....	
2.1.6. 常见问题处理.....	
3. 服务端工具.....	
3.1. gs_check.....	
3.2. gs_checkos.....	
3.3. gs_checkperf.....	
3.4. gs_collector.....	
3.5. vb_dump.....	
3.6. vb_dumpall.....	
3.7. vb_guc.....	
3.8. gs_om.....	
3.9. vb_restore.....	
3.10. gs_ssh.....	
4. 系统内部使用的工具.....	
4.1. vastbase.....	
4.2. gs_backup.....	
4.3. vb_basebackup.....	
4.4. vb_ctl.....	
4.5. vb_initdb.....	
4.5.1. vb_initdb 简介.....	
4.5.2. 使用指导.....	
4.5.3. 命令参考.....	
4.6. gs_install.....	
4.7. gs_postuninstall.....	
4.8. gs_preinstall.....	
4.9. gs_sshexkey.....	
4.10. gs_uninstall.....	
4.11. gs_upgradectl.....	
4.12. vbtrace.....	
4.13. kdb5_util.....	
4.14. kadmin.local.....	
4.15. kinit.....	
4.16. klist.....	
4.17. krb5kdc.....	
4.18. kdestroy.....	
4.19. vb_config.....	
4.20. vb_controldata.....	
4.21. vb_resetxlog.....	
5. FAQ.....	

5.1. 执行命令报错“Failed to obtain the GPHOME”	
6. gs_collector 工具支持收集的表视图列表.....	

1. 工具一览表

Vastbase 提供客户端和服务端应用程序（工具），帮助用户更好地维护 Vastbase，提供的所有工具如表 1-1 所示。工具位于安装数据库服务器的\$GPHOME/script 和\$GAUSSHOME/bin 路径下。

表 1-1 工具一览表

分类	工具名称	简介
2 客户端工具	2.1 vsql	vsql 是 Vastbase 提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，vsql 还提供了若干 高级特性 ，便于用户使用。
3 服务端工具	3.1 vb_dump	vb_dump 是一款用于导出数据库相关信息的工具，支持导出完整一致的数据库对象（数据库、模式、表、视图等）数据，同时不影响用户对数据库的正常访问。
	3.2 vb_dumpall	vb_dumpall 是一款用于导出数据库相关信息的工具，支持导出完整一致的 Vastbase 数据库所有数据，同时不影响用户对数据库的正常访问。
	3.3 vb_guc	vb_guc 用于设置 Vastbase 配置文件（“postgresql.conf”、“pg_hba.conf”）中的参数，配置文件中参数的默认值是单机的配置模式，您可以使用 vb_guc 来设置适合的参数值。
	3.4 vb_restore	vb_restore 是 Vastbase 提供的针对 vb_dump 导出数据的导入工具。通过此工具可由 vb_dump 生成的导出文件进行导入。

2. 客户端工具

数据库在部署成功后，需要通过一些工具来便捷地连接数据库，对数据库进行各种操作和调试。Vastbase 提供了一些数据库连接工具。通过这些工具可以方便地连接数据库并对数据库进行数据操作。[2.1 vsql](#)

2.1. vsql

vsql 是 Vastbase 提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，vsql 还提供了若干[高级特性](#)，便于用户使用。

2.1.1. vsql 概述

基本功能

❖ 连接数据库：详细操作请参见《开发者指南》中“数据库使用 > 连接数据库 > 使用 vsql 连接 > 远程连接数据库”章节。

说明

vsql 创建连接时，会有 5 分钟超时时间。如果在这个时间内，数据库未正确地接受连接并对身份进行认证，vsql 将超时退出。

针对此问题，可以参考 [2.1.6 常见问题处理](#)。

- ❖ 执行 SQL 语句：支持交互式地键入并执行 SQL 语句，也可以执行一个文件中指定的 SQL 语句。
- ❖ 执行元命令：元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化 SQL 输出结果，以及连接到新的数据库等。元命令的详细说明请参见 [2.1.5 元命令参考](#)。

高级特性

vsql 的高级特性如表 1-2 所示。

表 1-2 vsql 高级特性

特性名称	描述
变量	vsql 提供类似于 Linux 的 shell 命令的变量特性，可以使用 vsql 的元命令 <code>\set</code> 设置一个变量，格式如下： <pre>\set varname value</pre> 要删除一个变量请使用如下方式： <pre>\unset varname</pre> 说明

特性名称	描述
	- 变量只是简单的名称/值对，这里的值可以是任意长度。 - 变量名称必须由字母（包括非拉丁字母）、数字和下划线组成，且对大小写敏感。 - 如果使用\set varname 的格式（不带第二个参数），则只是设置这个变量而没有给变量赋值。 - 可以使用不带参数的\set 来显示所有变量的值。 变量的示例和详细说明请参见 变量 。
SQL 代换	利用 vsql 的变量特性，可以将常用的 SQL 语句设置为变量，以简化操作。 SQL 代换的示例和详细说明请参见 SQL 代换 。
自定义提示符	vsql 使用的提示符支持用户自定义。可以通过修改 vsql 预留的三个变量 PROMPT1、PROMPT2、PROMPT3 来改变提示符。 这三个变量的值可以用户自定义，也可以使用 vsql 预定义的值。详细请参见 提示符 。
客户端操作历史记录	vsql 支持客户端操作历史记录，当客户端连接时指定“-r”参数，此功能被打开。可以通过 \set 设置记录历史的条数，例如，\set HISTSIZE 50，将记录历史的条数设置为 50，\set HISTSIZE 0，不记录历史。 说明 - 客户端操作历史记录条数默认设置为 32 条，最多支持记录 500 条。当客户端交互式输入包含中文字符时，只支持 UTF-8 的编码环境。 - 出于安全考虑，将包含 PASSWORD、IDENTIFIED 敏感词的记录识别为敏感信息，不会记录到历史信息中，即不能通过上下翻回显。

❖ 变量

可以使用 vsql 元命令\set 设置一个变量。例如把变量 foo 的值设置为 bar：

```
vastbase=# \set foo bar
```

要引用变量的值，在变量前面加冒号。例如查看变量的值：

```
vastbase=# \echo :foo
bar
```

这种变量的引用方法适用于规则的 SQL 语句和元命令。

vsql 预定义了一些特殊变量，同时也规划了变量的取值。为了保证和后续版本最大限度地兼容，请避免以其他目的使用这些变量。所有特殊变量见表 2-2。

📖 说明

- 所有特殊变量都由大写字母、数字和下划线组成。
- 要查看特殊变量的默认值，请使用元命令\echo :varname（例如\echo :DBNAME）。

表 1-3 特殊变量设置

变量	设置方法	变量说明
DBNAME	<code>\set DBNAME dbname</code>	当前连接的数据库的名称。每次连接数据库时都会

变量	设置方法	变量说明
		被重新设置。
ECHO	<code>\set ECHO all queries</code>	•如果设置为 all，只显示查询信息。设置为 all 等效于使用 vsql 连接数据库时指定-a 参数。 •如果设置为 queries，显示命令行和查询信息。等效于使用 vsql 连接数据库时指定-e 参数。
ECHO_HIDDEN	<code>\set ECHO_HIDDEN on off noexec</code>	当使用元命令查询数据库信息（例如\dg）时，此变量的取值决定了查询的行为： •设置为 on，先显示元命令实际调用的查询语句，然后显示查询结果。等效于使用 vsql 连接数据库时指定-E 参数。 •设置为 off，则只显示查询结果。 •设置为 noexec，则只显示查询信息，不执行查询操作。
ENCODING	<code>\set ENCODING encoding</code>	当前客户端的字符集编码。
FETCH_COUNT	<code>\set FETCH_COUNT variable</code>	•如果该变量的值为大于 0 的整数，假设为 n，则执行 SELECT 语句时每次从结果集中取 n 行到缓存并显示到屏幕。 •如果不设置此变量，或设置的值小于等于 0，则执行 SELECT 语句时一次性把结果都取到缓存。 说明 设置合理的变量值，将减少内存使用量。一般来说，设为 100 到 1000 之间的值比较合理。
HISTCONTROL	<code>\set HISTCONTROL ignorespace ignoredups ignoreboth none</code>	•ignorespace：以空格开始的行将不会写入历史列表。 •ignoredups：与以前历史记录里匹配的行不会写入历史记录。 •ignoreboth、none 或者其他值：所有以交互模式读入的行都被保存到历史列表。 说明 none 表示不设置 HISTCONTROL。
HISTFILE	<code>\set HISTFILE filename</code>	此文件用于存储历史名列表。缺省值是 ~/.gsq_history

变量	设置方法	变量说明
HISTSIZE	<code>\set HISTSIZE size</code>	保存在历史命令里命令的个数。缺省值是 500。
HOST	<code>\set HOST hostname</code>	已连接的数据库主机名称。
IGNOREEOF	<code>\set IGNOREEOF variable</code>	•若设置此变量为数值，假设为 10，则在 vsql 中输入的前 9 次 EOF 字符（通常是 Ctrl+D 组合键）都会被忽略，在第 10 次按 Ctrl+D 才能退出 vsql 程序。 •若设置此变量为非数值，则缺省为 10。 •若删除此变量，则向交互的 vsql 会话发送一个 EOF 终止应用。
LASTOID	<code>\set LASTOID oid</code>	最后影响的 oid 值，即为从一条 INSERT 或 lo_import 命令返回的值。此变量只保证在下一条 SQL 语句的结果显示之前有效。
ON_ERROR_ROLLBACK	<code>\set ON_ERROR_ROLLBACK on interactive off</code>	•如果是 on，当一个事务块里的语句产生错误的时候，这个错误将被忽略而事务继续。 •如果是 interactive，这样的错误只是在交互的会话里忽略。 •如果是 off（缺省），事务块里一个语句生成的错误将会回滚整个事务。on_error_rollback-on 模式是通过在一个事务块的每个命令前隐含地发出一个 SAVEPOINT 的方式工作的，在发生错误的时候回滚到该事务块。
ON_ERROR_STOP	<code>\set ON_ERROR_STOP on off</code>	•on：命令执行错误时会立即停止，在交互模式下，vsqI 会立即返回已执行命令的结果。 •off（缺省）：命令执行错误时将会跳过错误继续执行。
PORT	<code>\set PORT port</code>	正连接数据库的端口号。
USER	<code>\set USER username</code>	当前用于连接的数据库用户。
VERBOSITY	<code>\set VERBOSITY terse default verbose</code>	这个选项可以设置为值 terse、default、verbose 之一以控制错误报告的冗余行。 •terse：仅返回严重且主要的错误文本以及文本

变量	设置方法	变量说明
		位置（一般适合于单行错误信息）。 •default：返回严重且主要的错误文本及其位置，还包括详细的错误细节、错误提示（可能会跨越多行）。 •verbose：返回所有的错误信息。

❖ SQL 代换

像元命令的参数一样，vsqI 变量的一个关键特性是可以把 vsqI 变量替换成正规的 SQL 语句。此外，vsqI 还提供为变量更换新的别名或其他标识符等功能。使用 SQL 代换方式替换一个变量的值可在变量前加冒号。例如：

```
vastbase=# \set foo 'HR.areaS'
vastbase=# select * from :foo;
 area_id |      area_name
-----+-----
      4 | Middle East and Africa
      3 | Asia
      1 | Europe
      2 | Americas
(4 rows)
```

执行以上命令，将会查询 HR.areaS 表。

须知

变量的值是逐字复制的，甚至可以包含不对称的引号或反斜杠命令。所以必须保证输入的内容有意义。

❖ 提示符

通过表 1-4 的三个变量可以设置 vsqI 的提示符，这些变量是由字符和特殊的转义字符所组成。

表 1-4 提示符变量

变量	描述	示例
PROMPT1	vsqI 请求一个新命令时使用的正常提示符。 PROMPT1 的默认值为： %/%R%#	使用变量 PROMPT1 切换提示符： •提示符变为[local]： <pre>postgres=> \set PROMPT1 %M [local:/tmp/vastbase_mppdb]</pre> •提示符变为 name： <pre>postgres=> \set PROMPT1 name name</pre> •提示符变为=：

变量	描述	示例
		<pre>postgres=> \set PROMPT1 %R =</pre>
PROMPT2	在一个命令输入期待更多输入时（例如，查询没有用一个分号结束或者引号不完整）显示的提示符。	使用变量 PROMPT2 显示提示符： <pre>vastbase=# \set PROMPT2 TEST vastbase=# select * from HR.areaS TEST; area id area name -----+----- 1 Europe 2 Americas 4 Middle East and Africa 3 Asia (4 rows))</pre>
PROMPT3	当执行 COPY 命令，并期望在终端输入数据时（例如，COPY FROM STDIN），显示提示符。	使用变量 PROMPT3 显示 COPY 提示符： <pre>vastbase=# \set PROMPT3 '>>>>' vastbase=# copy HR.areaS from STDIN; Enter data to be copied followed by a newline. End with a backslash and a period on a line by itself. >>>>1 aa >>>>2 bb >>>>\.</pre>

提示符变量的值是按实际字符显示的，但是，当设置提示符的命令中出现 “%” 时，变量的值根据 “%” 后的字符，替换为已定义的内容，已定义的提示符请参见表 1-5。

表 1-5 已定义的替换

符号	符号说明
%M	主机的全名（包含域名），若连接是通过 Unix 域套接字进行的，则全名为[local]，若 Unix 域套接字不是编译的缺省位置，就是[local:/dir/name]。
%m	主机名删去第一个点后面的部分。若通过 Unix 域套接字连接，则为[local]。
%>	主机正在侦听的端口号。
%n	数据库会话的用户名。
%/	当前数据库名称。
%~	类似 %/，如果数据库是缺省数据库时输出的是波浪线~。
%#	如果会话用户是数据库系统管理员，使用#，否则用>。
%R	对于 PROMPT1 通常是“=”，如果是单行模式则是“^”，如果会话与数据库断开（如果 \connect 失败可能发生）则是“!”。

符号	符号说明
	•对于 PROMPT2 该序列被“-”、“*”、单引号、双引号或“\$”(取决于 vsql 是否等待更多的输入：查询没有终止、正在一个 /* ... */ 注释里、正在引号或者美元符扩展里)代替。
%x	事务状态： •如果不在事务块里，则是一个空字符串。 •如果在事务块里，则是“*”。 •如果在一个失败的事务块里则是“!”。 •如果无法判断事务状态时为“?”（比如没有连接）。
%digits	指定字节值的字符将被替换到该位置。
%:name	vsql 变量“name”的值。
%command	command 的输出，类似于使用“^”替换。
%[... %]	提示可以包含终端控制字符，这些字符可以改变颜色、背景、提示文本的风格、终端窗口的标题。例如， potgres=> \set PROMPT1 '%[%033[1;33;40m%]%n@%/%R%[%033[0m%]%#' 这个句式的结果是在 VT100 兼容的可显示彩色的终端上的一个宽体（1;）黑底黄字（33;40） 。

环境变量

表 1-6 与 vsql 相关的环境变量

名称	描述
COLUMNS	如果\set columns 为 0，则由此参数控制 wrapped 格式的宽度。这个宽度用于决定在自动扩展的模式下，是否要把宽输出模式变成竖线的格式。
PAGER	如果查询结果无法在一页显示，它们就会被重定向到这个命令。可以用\pset 命令关闭分页器。典型的是用命令 more 或 less 来实现逐页查看。缺省值是平台相关的。 说明 less 的文本显示，受系统环境变量 LC_CTYPE 影响。
PSQL_EDITOR	\e 和\ef 命令使用环境变量指定的编辑器。变量是按照列出的先后顺序检查的。在 Unix 系统上默认的编辑工具是 vi。
EDITOR	
VISUAL	

名称	描述
PSQL_EDITOR_LINENUMBER_ARG	当\ee和\ef带上一行数字参数使用时,这个变量指定的命令行参数用于向编辑器传递起始行数。像 Emacs 或 vi 这样的编辑器,这只是个加号。如果选项和行号之间需要空白,在变量的值后加一个空格。例如: <pre>PSQL_EDITOR_LINENUMBER_ARG = '+' PSQL_EDITOR_LINENUMBER_ARG='--line '</pre> Unix 系统默认的是+。
PSQLRC	用户的.vsqlrc 文件的交互位置。
SHELL	使用\!命令跟 shell 执行的命令是一样的效果。
TMPDIR	存储临时文件的目录。缺省是/tmp。

2.1.2. 使用指导

前提条件

连接数据库时使用的用户需要具备访问数据库的权限。

背景信息

使用 vsql 命令可以连接远程数据库服务。连接远程数据库服务时,需要在服务器上设置允许远程连接,详细操作请参见《开发者指南》中“数据库使用 > 连接数据库 > 使用 vsql 连接 > 远程连接数据库”章节。

操作步骤

步骤 1 使用 vsql 连接到 Vastbase 服务器。

vsql 工具使用-d 参数指定目标数据库名、-U 参数指定数据库用户名、-h 参数指定主机名、-p 参数指定端口号信息。

说明

若未指定数据库名称,则使用初始化时默认生成的数据库名称;若未指定数据库用户名,则默认使用当前操作系统用户作为数据库用户名;当某个值没有前面的参数(-d、-U等)时,若连接的命令中没有指定数据库名(-d)则该参数会被解释成数据库名;如果已经指定数据库名(-d)而没有指定数据库用户名(-U)时,该参数则会被解释成数据库用户名。

示例 1, 使用 omm 用户连接到本机 postgres 数据库的 5432 端口。

```
vsql -d postgres -p 5432
```

示例 2, 使用 jack 用户连接到远程主机 postgres 数据库的 5432 端口。

```
vsql -h 10.180.123.163 -d postgres -U jack -p 5432
```

示例 3，参数 postgres 和 omm 不属于任何选项时，分别被解释成了数据库名和用户名。

```
vsql postgres omm -p 5432
```

等效于

```
vsql -d postgres -U omm -p 5432
```

详细的 vsql 参数请参见 2.1.4 命令参考。

步骤 2 执行 SQL 语句。

以创建数据库 human_staff 为例。

```
CREATE DATABASE human_staff;  
CREATE DATABASE
```

通常，输入的命令在遇到分号的时候结束。如果输入的命令没有错误，结果就会输出到屏幕上。

步骤 3 执行 vsql 元命令。

以列出 Vastbase 中所有的数据库和描述信息为例。

```
vastbase=# \l  
  
List of databases  
Name | Owner | Encoding | Collate | Ctype | Access privileges  
-----+-----+-----+-----+-----+-----  
human_resource | omm | SQL_ASCII | C | C |  
postgres | omm | SQL_ASCII | C | C |  
template0 | omm | SQL_ASCII | C | C | =c/omm +  
 | | | | | omm=CTc/omm  
template1 | omm | SQL_ASCII | C | C | =c/omm +  
 | | | | | omm=CTc/omm  
human_staff | omm | SQL_ASCII | C | C |  
(5 rows)
```

更多 vsql 元命令请参见 2.1.5 元命令参考。

----结束

示例

以把一个查询分成多行输入为例。注意提示符的变化：

```
vastbase=# CREATE TABLE HR.areaS(  
postgres(# area_ID NUMBER,  
postgres(# area_NAME VARCHAR2(25)  
postgres(# )tablespace EXAMPLE  
CREATE TABLE
```

查看表的定义：

```
vastbase=# \d HR.areaS  
  
Table "hr.areas"  
Column | Type | Modifiers  
-----+-----+-----  
area_id | numeric | not null  
area_name | character varying(25) |
```

向 HR.areaS 表插入四行数据：

```
vastbase=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (1, 'Europe');  
INSERT 0 1  
vastbase=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (2, 'Americas');
```

```

INSERT 0 1
vastbase=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (3, 'Asia');
INSERT 0 1
vastbase=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (4, 'Middle East and Africa');
INSERT 0 1

```

切换提示符:

```

vastbase=# \set PROMPT1 '%n@%m %~%R%#'
omm@[local] vastbase=#

```

查看表:

```

omm@[local] vastbase=#SELECT * FROM HR.areaS;
 area_id |      area_name
-----+-----
      1 | Europe
      4 | Middle East and Africa
      2 | Americas
      3 | Asia
(4 rows)

```

可以用\pset 命令以不同的方法显示表:

```

omm@[local] vastbase=#\pset border 2
Border style is 2.
omm@[local] vastbase=#SELECT * FROM HR.areaS;
+-----+-----+
| area_id |      area_name      |
+-----+-----+
|      1 | Europe              |
|      2 | Americas            |
|      3 | Asia                |
|      4 | Middle East and Africa |
+-----+-----+
(4 rows)
omm@[local] vastbase=#\pset border 0
Border style is 0.
omm@[local] vastbase=#SELECT * FROM HR.areaS;
area_id      area_name
-----
      1 Europe
      2 Americas
      3 Asia
      4 Middle East and Africa
(4 rows)

```

使用元命令:

```

omm@[local] vastbase=#\a \t \x
Output format is unaligned.
Showing only tuples.
Expanded display is on.
omm@[local] vastbase=#SELECT * FROM HR.areaS;
area_id|2
area_name|Americas

area_id|1
area_name|Europe

area_id|4

```

```
area_name|Middle East and Africa
area_id|3
area_name|Asia
omm@[local] vastbase=#
```

2.1.3. 获取帮助

操作步骤

- ❖ 连接数据库时，可以使用如下命令获取帮助信息。

```
vsql --help
```

显示如下帮助信息：

```
.....
Usage:
  vsql [OPTION]... [DBNAME [USERNAME]]

General options:
  -c, --command=COMMAND  run only single command (SQL or internal) and exit
  -d, --dbname=DBNAME    database name to connect to (default: "postgres")
  -f, --file=FILENAME    execute commands from file, then exit
.....
```

- ❖ 连接到数据库后，可以使用如下命令获取帮助信息。

```
help
```

显示如下帮助信息：

```
You are using vsql, the command-line interface to Vastbase.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with vsql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

任务示例

步骤 1 使用如下命令连接数据库。

```
vsql -d postgres -p 5432
```

postgres 为需要连接的数据库名称，5432 为数据库主节点的端口号。

连接成功后，系统显示类似如下信息：

```
vsql ((Vastbase x.x build 290d125f) compiled at 2020-05-08 02:59:43 commit 2143 last mr 131
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

vastbase=#
```

步骤 2 查看 vsql 的帮助信息。具体执行命令请参见表 1-7。

表 1-7 使用 vsql 联机帮助

描述	示例
----	----

描述	示例
查看版权信息	<code>\copyright</code>
查看 Vastbase 支持的 SQL 语句的帮助	查看 Vastbase 支持的 SQL 语句的帮助 例如，查看 Vastbase 支持的所有 SQL 语句： <pre>vastbase=# \h Available help: ABORT ALTER APP WORKLOAD GROUP </pre> 例如，查看 CREATE DATABASE 命令的参数可使用下面的命令： <pre>vastbase=# \help CREATE DATABASE Command: CREATE DATABASE Description: create a new database Syntax: CREATE DATABASE database_name [[WITH] {[OWNER [=] user_name]} [TEMPLATE [=] template]} [ENCODING [=] encoding]} [LC_COLLATE [=] lc_collate]} [LC_CTYPE [=] lc_ctype]} [DBCOMPATIBILITY [=] compatibility_type]} [TABLESPACE [=] tablespace_name]} [CONNECTION LIMIT [=] connlimit]}{[...]};</pre>
查看 vsql 命令的帮助	例如，查看 vsql 支持的命令： <pre>vastbase=# \? General \copyright show FusionInsight LibrA usage and distribution terms \g [FILE] or ; execute query (and send results to file or pipe) \h(\help) [NAME] help on syntax of SQL commands, * for all commands \parallel [on [num] off] toggle status of execute (currently off) \q quit vsql </pre>

----结束

2.1.4. 命令参考

详细的 vsql 参数请参见表 1-8、表 1-9、表 1-10 和 表 1-11。

表 1-8 常用参数

参数	参数说明	取值范围
----	------	------

参数	参数说明	取值范围
-c, --command=COMMAND	声明 vsql 要执行一条字符串命令然后退出。	-
-d, --dbname=DBNAME	指定想要连接的数据库名称。	字符串。
-f, --file=FILENAME	使用文件作为命令源而不是交互式输入。vsqI 将在处理完文件后结束。如果 FILENAME 是-（连字符），则从标准输入读取。 说明 环境场景：一主一备（8U32G） 实测：读取 200M 数据文件时，耗时约为 8 分钟 21 秒； 读取 500M 数据文件时，耗时约为 18 分 41 秒。 建议：文件读取的时间随文件数据量逐渐增加，如果文件太大，中间异常时需要重新读取，同时会导致系统 OS 的 IO 过载，建议文件大小控制在 500M 左右。	绝对路径或相对路径，且满足操作系统路径命名规则。
-l, --list	列出所有可用的数据库，然后退出。	-
-v, --set, --variable=NAME=VALUE	设置 vsqI 变量 NAME 为 VALUE。 变量的示例和详细说明请参见变量。	-
-X, --no-vsqrIrc	不读取启动文件（系统范围的 vsqrIrc 或者用户的 ~/.vsqrIrc 都不读取）。 说明 启动文件默认为 ~/.vsqrIrc，或通过 PSQLRC 环境变量指定。	-
-1 ("one"), --single-transaction	当 vsqI 使用-f 选项执行脚本时，会在脚本的开头和结尾分别加上 START TRANSACTION/COMMIT 用以把整个脚本当作一个事务执行。这将保证该脚本完全执行成功，或者脚本无效。 说明 如果脚本中已经使用了 START TRANSACTION, COMMIT, ROLLBACK，则该选项无效。	-
-?, --help	显示关于 vsqI 命令行参数的帮助信息然后退出。	-
-V, --version	打印 vsqI 版本信息然后退出。	-

表 1-9 输入和输出参数

参数	参数说明	取值范围
-a, --echo-all	在读取行时向标准输出打印所有内容。	-

参数	参数说明	取值范围
	注意 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。	
-e, --echo-queries	把所有发送给服务器的查询同时回显到标准输出。 注意 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。	-
-E, --echo-hidden	回显由\d和其他反斜杠命令生成的实际查询。	-
-k, --with-key=KEY	使用 vsql 对导入的加密文件进行解密。 须知 对于本身就是 shell 命令中的关键字符如单引号 (') 或双引号 (")，Linux shell 会检测输入的单引号 (') 或双引号 (") 是否匹配。如果不匹配，shell 认为用户没有输入完毕，会一直等待用户输入，从而不会进入到 vsql 程序。	-
-L, --log-file=FILENAME	除了正常的输出源之外，把所有查询输出记录到文件 FILENAME 中。 注意 - 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。 - 此参数只保留查询结果到相应文件中，主要目标是为了查询结果能够更好更准确地被其他调用者（例如自动化运维脚本）解析；而不是保留 vsql 运行过程中的相关日志信息。	绝对路径或相对路径，且满足操作系统路径命名规则。
-m, --maintenance	允许在两阶段事务恢复期间连接 Vastbase。 说明 该选项是一个开发选项，禁止用户使用，只限专业技术人员使用，功能是：使用该选项时，vsqI 可以连接到备机，用于校验主备机数据的一致性。	-
-n, --no-libedit	关闭命令行编辑。	-
-o, --output=FILENAME	将所有查询输出重定向到文件 FILENAME。	绝对路径或相对路径，且满足操作系统路径命名规则。
-q, --quiet	安静模式，执行时不会打印出额外信息。	缺省时 vsqI 将打印许多其他输出信息。
-s, --single-step	单步模式运行。意味着每个查询在发往服务器之前都要提示用户，用这个选项也可以取消执行。此选项主要用于调试脚本。	-

参数	参数说明	取值范围
	注意 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。	
-S, --single-line	单行运行模式，这时每个命令都将由换行符结束，像分号那样。	-

表 1-10 输出格式参数

参数	参数说明	取值范围
-A, --no-align	切换为非对齐输出模式。	缺省为对齐输出模式。
-F, --field-separator=STRING	设置域分隔符（默认为“ ”）。	-
-H, --html	打开 HTML 格式输出。	-
-P, --pset=VAR[=ARG]	在命令行上以 pset 的风格设置打印选项。 说明 这里必须用等号而不是空格分隔名称和值。例如，把输出格式设置为 LaTeX，可以键入 -P format=latex	-
-R, --record-separator=STRING	设置记录分隔符。	-
-r	开启在客户端操作中可以进行编辑的模式。	缺省为关闭。
-t, --tuples-only	只打印行。	-
-T, --table-attr=TEXT	允许声明放在 HTML table 标签里的选项。 使用时请搭配参数“-H,--html”，指定为 HTML 格式输出。	-
-x, --expanded	打开扩展表格式模式。	-
-z, --field-separator-zero	设置非对齐输出模式的域分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-
-0, --record-separator-zero	设置非对齐输出模式的记录分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-

表 1-11 连接参数

参数	参数说明	取值范围
-h, --host=HOSTNAME	指定正在运行服务器的主机名或者 Unix 域套接字的路径。	如果省略主机名, vsql 将通过 Unix 域套接字与本地主机的服务器相联, 或者在没有 Unix 域套接字的机器上, 通过 TCP/IP 与 localhost 连接。
-p, --port=PORT	指定数据库服务器的端口号。 可以通过 port 参数修改默认端口号。	默认为 5432。
-U, --username=USERNAME	指定连接数据库的用户。 说明 - 通过该参数指定用户连接数据库时, 需要同时提供用户密码用以身份验证。您可以通过交换方式输入密码, 或者通过 -W 参数指定密码。 - 用户名中包含有字符 \$, 需要在字符 \$ 前增加转义字符才可成功连接数据库。	字符串。默认使用与当前操作系统用户同名的用户。
-W, --password=PASSWORD	当使用 -U 参数连接远端数据库时, 可通过该选项指定密码。 说明 数据库主节点所在服务器后连接本地数据库主节点实例时, 默认使用 trust 连接, 会忽略此参数。 用户密码中包含特殊字符“\”和“'”时, 需要增加转义字符才可成功连接数据库。 如果用户未输入该参数, 但是数据库连接需要用户密码, 这时将出现交互式输入, 请用户输入当前连接的密码。该密码最长长度为 999 字节, 受限与 GUC 参数 password_max_length 的最大值。	符合密码复杂度要求。

2.1.5. 元命令参考

介绍使用 Vastbase 数据库命令行交互工具登录数据库后, vsql 所提供的元命令。所谓元命令就是在 vsql 里输入的任何以不带引号的反斜杠开头的命令。

注意事项

- ❖ 一个 vsql 元命令的格式是反斜杠后面紧跟一个动词, 然后是任意参数。参数命令动词和其他参数以任意个空白字符间隔。

- ❖ 要在参数里面包含空白，必须用单引号把它引起来。要在这样的参数里包含单引号，可以在前面加一个反斜杠。任何包含在单引号里的内容都会被进一步进行类似 C 语言的替换：\n（新行）、\t（制表符）、\b（退格）、\r（回车）、\f（换页）、\digits（八进制表示的字符）、\xdigits（十六进制表示的字符）。
- ❖ 用""包围的内容被当做一个命令行传入 shell。该命令的输出（删除了结尾的新行）被当做参数值。
- ❖ 如果不带引号的参数以冒号（:）开头，它会被当做一个 vsql 变量，并且该变量的值最终会成为真正的参数值。
- ❖ 有些命令以一个 SQL 标识的名称（比如一个表）为参数。这些参数遵循 SQL 语法关于双引号的规则：不带双引号的标识强制转换成小写，而双引号保护字母不进行大小写转换，并且允许在标识符中使用空白。在双引号中，成对的双引号在结果名称中分析成一个双引号。比如，FOO"BAR"BAZ 解析成 fooBARbaz；而"Aweird""name"解析成 A weird"name。
- ❖ 对参数的分析在遇到另一个不带引号的反斜杠时停止。这里会认为是一个新的元命令的开始。特殊的双反斜杠序列（\\）标识参数的结尾并将继续分析后面的 SQL 语句（如果存在）。这样 SQL 和 vsql 命令可以自由的在一行里面混合。但是在任何情况下，一条元命令的参数不能延续超过行尾。

元命令

元命令的详细说明请参见表 1-12、表 1-13、表 1-14、表 1-15、表 1-17、表 1-19、表 1-20、表 1-21 和表 1-23。

须知

以下命令中所提到的 FILE 代表文件路径。此路径可以是绝对路径（如/home/vastbase/file.txt），也可以是相对路径（file.txt，file.txt 会默认在用户执行 vsql 命令所在的路径下创建）。

表 1-12 一般的元命令

参数	参数说明	取值范围
\copyright	显示 Vastbase 的版本和版权信息。	-
\g [FILE] or ;	执行查询（并将结果发送到文件或管道）。	-
\h(\help) [NAME]	给出指定 SQL 语句的语法帮助。	如果没有给出 NAME，vsqI 将列出可获得帮助的所有命令。如果 NAME 是一个星号（*），则显示所有 SQL 语句的语法帮助。
\parallel [on [num]][off]	控制并发执行开关。 •on: 打开控制并发执行开关，且最大并发数为 num。	num 的默认值：1024。 须知 • 服务器能接受的最大连接数

参数	参数说明	取值范围
	• off: 关闭控制并发执行开关。 说明 • 不支持事务中开启并发执行以及并发中开启事务。 • 不支持\d这类元命令的并发。 • 并发 select 返回结果混乱问题, 此为客户可接受, core、进程停止响应不可接受。 • 不推荐在并发中使用 set 语句, 否则导致结果与预期不一致。 • 不支持创建临时表! 如需使用临时表, 需要在开启 parallel 之前创建好, 并在 parallel 内部使用。parallel 内部不允许创建临时表。 • \parallel 执行时最多会启动 num 个独立的 vsql 进程连接服务器。 • \parallel 中所有作业的持续时间不能超过 session_timeout, 否则可能会导致并发执行过程中断连。	受 max_connection 及当前已有连接数限制。 • 设置 num 时请考虑服务器当前可接受的连接数合理指定。
\q	退出 vsql 程序。在一个脚本文件里, 只在脚本终止的时候执行。	-

表 1-13 查询缓存区元命令

参数	参数说明
\e [FILE] [LINE]	使用外部编辑器编辑查询缓冲区 (或者文件)。
\ef [FUNCNAME [LINE]]	使用外部编辑器编辑函数定义。如果指定了 LINE (即行号), 则光标会指到函数体的指定行。
\p	打印当前查询缓冲区到标准输出。
\r	重置 (或清空) 查询缓冲区。
\w FILE	将当前查询缓冲区输出到文件。

表 1-14 输入/输出元命令

参数	参数说明
\copy { table [(column_list)] (query) } { from to } { filename stdin stdout pstdin pstdout } [with] [binary] [oids] [delimiter [as] 'character'] [null [as] 'string'] [csv [header] [quote [as] 'character'] [escape [as] 'character'] [force quote column_list *] [force not null column_list]]	在任何 psql 客户端登录数据库成功后可以执行导入导出数据, 这是一个运行 SQL COPY 命令的操作, 但不是读取或写入指定文件的服务器, 而是读取或写入文件, 并在服务器和本地文件系统之间路由数据。这意味着文件的可访问性和权限是本地用户的权限, 而不是服务器的权限, 并且不需

参数	参数说明
	要数据库初始化用户权限。 说明 COPY 只适合小批量，格式良好的数据导入，不会对非法字符进行预处理，也无容错能力。导入数据应优先选择 COPY。
\echo [STRING]	把字符串写到标准输出。
\i FILE	从文件 FILE 中读取内容，并将其当作输入，执行查询。
\i+ FILE KEY	执行加密文件中的命令。
\ir FILE	和 \i 类似，只是相对于存放当前脚本的路径。
\ir+ FILE KEY	和 \i+ 类似，只是相对于存放当前脚本的路径。
\o [FILE]	把所有的查询结果发送到文件里。
\qecho [STRING]	把字符串写到查询结果输出流里。

说明

表 1-15 中的选项 S 表示显示系统对象，+ 表示显示对象附加的描述信息。PATTERN 用来指定要被显示的对象名称。

表 1-15 显示信息元命令

参数	参数说明	取值范围	示例
\d[S+]	列出当前 search_path 中模式下所有的表、视图和序列。当 search_path 中不同模式存在同名对象时，只显示 search_path 中位置靠前模式下的同名对象。	-	列出当前 search_path 中模式下所有的表、视图和序列。 <code>vastbase=# \d</code>
\d[S+] NAME	列出指定表、视图和索引的结构。	-	假设存在表 a，列出指定表 a 的结构。 <code>vastbase=# \dtable+ a</code>
\d+ [PATTERN]	列出所有表、视图和索引。	如果声明了 PATTERN，只显示名称匹配 PATTERN 的表、视图和索引。	列出所有名称以 f 开头的表、视图和索引。 <code>vastbase=# \d+ f*</code>
\da[S] [PATTERN]	列出所有可用的聚集函数，以及它们操作的数据类型和返回值类型。	如果声明了 PATTERN，只显示名称匹配	列出所有名称以 f 开头的聚集函数，以及

参数	参数说明	取值范围	示例
		PATTERN 的聚集函数。	它们操作的数据类型和返回值类型。 vastbase=# \da f*
\db[+] [PATTERN]	列出所有可用的表空间。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的表空间。	列出所有名称以 p 开头的可用表空间。 vastbase=# \db p*
\dc[S+] [PATTERN]	列出所有字符集之间的可用转换。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的转换。	列出所有字符集之间的可用转换。 vastbase=# \dc *
\dC[+] [PATTERN]	列出所有类型转换。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的转换。	列出所有名称以 c 开头的类型转换。 vastbase=# \dC c*
\dd[S] [PATTERN]	显示所有匹配 PATTERN 的描述。	如果没有给出参数, 则显示所有可视对象。“对象”包括: 聚集、函数、操作符、类型、关系(表、视图、索引、序列、大对象)、规则。	列出所有可视对象。 vastbase=# \dd
\ddp [PATTERN]	显示所有默认的使用权限。	如果指定了 PATTERN, 只显示名称匹配 PATTERN 的使用权限。	列出所有默认的使用权限。 vastbase=# \ddp
\dD[S+] [PATTERN]	列出所有可用域。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的域。	列出所有可用域。 vastbase=# \dD
\ded[+] [PATTERN]	列出所有的 Data Source 对象。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的对象。	列出所有的 Data Source 对象。 vastbase=# \ded
\det[+] [PATTERN]	列出所有的外部表。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的表。	列出所有的外部表。 vastbase=# \det
\des[+] [PATTERN]	列出所有的外部服务器。	如果声明了 PATTERN,	列出所有的外部服务

参数	参数说明	取值范围	示例
		只显示名称匹配 PATTERN 的服务器。	器。 vastbase=# \des
\deu[+] [PATTERN]	列出用户映射信息。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的信息。	列出用户映射信息。 vastbase=# \deu
\dew[+] [PATTERN]	列出封装的外部数据。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的数据。	列出封装的外部数据。 vastbase=# \dew
\df[antw][S+] [PATTERN]	列出所有可用函数, 以及它们的参数和返回的数据类型。a 代表聚集函数, n 代表普通函数, t 代表触发器, w 代表窗口函数。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的函数。	列出所有可用函数, 以及它们的参数和返回的数据类型。 vastbase=# \df
\dF[+] [PATTERN]	列出所有的文本搜索配置信息。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的配置信息。	列出所有的文本搜索配置信息。 vastbase=# \dF+
\dFd[+] [PATTERN]	列出所有的文本搜索字典。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的字典。	列出所有的文本搜索字典。 vastbase=# \dFd
\dFp[+] [PATTERN]	列出所有的文本搜索分析器。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的分析器。	列出所有的文本搜索分析器。 vastbase=# \dFp
\dFt[+] [PATTERN]	列出所有的文本搜索模板。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的模板。	列出所有的文本搜索模板。 vastbase=# \dFt
\dg[+] [PATTERN]	列出所有数据库角色。 说明 因为用户和群组的概念被统一为角色, 所以这个命令等价于\du。为了和以前兼容, 所以保留两个命令。	如果指定了 PATTERN, 只显示名称匹配 PATTERN 的角色。	列出名称为'j_e'所有数据库角色。 vastbase=# \dg j?e
\dl	\lo_list 的别名, 显示一个大对象的列表。	-	列出所有的大对象。 vastbase=# \dl
\dL[S+] [PATTERN]	列出可用的程序语言。	如果指定了 PATTERN,	列出可用的程序语言。

参数	参数说明	取值范围	示例
		只列出名称匹配 PATTERN 的语言。	<code>vastbase=# \dL</code>
<code>\dn[S+] [PATTERN]</code>	列出所有的模式（名称空间）。	如果声明了 PATTERN，只列出名称匹配 PATTERN 的模式名。缺省时，只列出用户创建的模式。	列出所有名称以 d 开头的模式以及相关信息。 <code>vastbase=# \dn+ d*</code>
<code>\do[S] [PATTERN]</code>	列出所有可用的操作符，以及它们的操作数和返回的数据类型。	如果声明了 PATTERN，只列出名称匹配 PATTERN 的操作符。缺省时，只列出用户创建的操作符。	列出所有可用的操作符，以及它们的操作数和返回的数据类型。 <code>vastbase=# \do</code>
<code>\dO[S+] [PATTERN]</code>	列出排序规则。	如果声明了 PATTERN，只列出名称匹配 PATTERN 的规则。缺省时，只列出用户创建的规则。	列出排序规则。 <code>vastbase=# \dO</code>
<code>\dp [PATTERN]</code>	列出一列可用的表、视图以及相关的权限信息。 <code>\dp</code> 显示结果如下： <pre>rolename=xxxx/yyyy --赋予一个角色的权限 =xxxx/yyyy --赋予 public 的权限</pre> xxxx 表示赋予的权限，yyyy 表示授予这个权限的角色。权限的参数说明请参见表 2-15。	如果指定了 PATTERN，只列出名称匹配 PATTERN 的表、视图。	列出一列可用的表、视图以及相关的权限信息。 <code>vastbase=# \dp</code>
<code>\drds [PATTERN1 [PATTERN2]]</code>	列出所有修改过的配置参数。这些设置可以是针对角色的、针对数据库的或者同时针对两者的。PATTERN1 和 PATTERN2 表示要列出的角色 PATTERN 和数据库 PATTERN。	如果声明了 PATTERN，只列出名称匹配 PATTERN 的规则。缺省或指定*时，则会列出所有设置。	列出 postgres 数据库所有修改过的配置参数。 <code>vastbase=# \drds * postgres</code>
<code>\dT[S+] [PATTERN]</code>	列出所有的数据类型。	如果指定了 PATTERN，只列出名称匹配	列出所有的数据类型。 <code>vastbase=# \dT</code>

参数	参数说明	取值范围	示例
		PATTERN 的类型。	
\du[+] [PATTERN]	列出所有数据库角色。 说明 因为用户和群组的概念被统一为角色，所以这个命令等价于\dg。为了和以前兼容，所以保留两个命令。	如果指定了 PATTERN，则只列出名称匹配 PATTERN 的角色。	列出所有数据库角色。 <pre>vastbase=# \du</pre>
\dE[S+] [PATTERN] \di[S+] [PATTERN] \ds[S+] [PATTERN] \dt[S+] [PATTERN] \dv[S+] [PATTERN]	这一组命令，字母 E, i, s, t 和 v 分别代表着外部表，索引，序列，表和视图。可以以任意顺序指定其中一个或者它们的组合来列出这些对象。例如：\dit 列出所有的索引和表。在命令名称后面追加+，则每一个对象的物理尺寸以及相关的描述也会被列出。 说明 本版本暂时不支持序列。	如果指定了 PATTERN，只列出名称匹配该 PATTERN 的对象。默认情况下只会显示用户创建的对象。通过 PATTERN 或者 S 修饰符可以把系统对象包括在内。	列出所有的索引和视图。 <pre>vastbase=# \div</pre>
\dx[+] [PATTERN]	列出安装数据库的扩展信息。	如果指定了 PATTERN，则只列出名称匹配 PATTERN 的扩展信息。	列出安装数据库的扩展信息。 <pre>vastbase=# \dx</pre>
\l[+]	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。	-	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。 <pre>vastbase=# \l</pre>
\sf[+] FUNCNAME	显示函数的定义。 说明 对于带圆括号的函数名，需要在函数名两端添加双引号，并且在双引号后面加上参数类型列表。参数类型列表两端添加圆括号。	-	假设存在函数 function_a 和函数名带圆括号的函数 func()name，列出函数的定义。 <pre>vastbase=# \sf function_a vastbase=# \sf "func()name"(argtype1, argtype2)</pre>
\z [PATTERN]	列出数据库中所有表、视图和序列，	如果给出任何 pattern，则被当成一个正则表达	列出数据库中所有表、视图和序列，以及它们

参数	参数说明	取值范围	示例
	以及它们相关的访问特权。	式，只显示匹配的表、视图、序列。	相关的访问特权。 vastbase=# \z

表 1-16 权限的参数说明

参数	参数说明
r	SELECT：允许对指定的表、视图读取数据。
w	UPDATE：允许对指定表更新字段。
a	INSERT：允许对指定表插入数据。
d	DELETE：允许删除指定表中的数据。
D	TRUNCATE：允许清理指定表中的数据。
x	REFERENCES：允许创建外键约束。由于当前不支持外键，所以该参数暂不生效。
t	TRIGGER：允许在指定表上创建触发器。
X	EXECUTE：允许使用指定的函数，以及利用这些函数实现的操作符。
U	USAGE： •对于过程语言，允许用户在创建函数时，指定过程语言。 •对于模式，允许访问包含在指定模式中的对象。 •对于序列，允许使用 nextval 函数。
C	CREATE： •对于数据库，允许在该数据库里创建新的模式。 •对于模式，允许在该模式中创建新的对象。 •对于表空间，允许在其中创建表，以及允许创建数据库和模式的时候把该表空间指定为其缺省表空间。
c	CONNECT：允许用户连接到指定的数据库。
T	TEMPORARY：允许创建临时表。
arwdDxt	ALL PRIVILEGES：一次性给指定用户/角色赋予所有可赋予的权限。
*	给前面权限的授权选项。

表 1-17 格式化元命令

参数	参数说明
\a	对齐模式和非对齐模式之间的切换。
\C [STRING]	把正在打印的表的标题设置为一个查询的结果或者取消这样的设置。
\f [STRING]	对于不对齐的查询输出，显示或者设置域分隔符。
\H	•若当前模式为文本格式，则切换为 HTML 输出格式。 •若当前模式为 HTML 格式，则切换回文本格式。
\pset NAME [VALUE]	设置影响查询结果表输出的选项。NAME 的取值见表 2-17。
\t [on off]	切换输出的字段名的信息和行计数脚注。
\T [STRING]	指定在使用 HTML 输出格式时放在 table 标签里的属性。如果参数为空，不设置。
\x [on off auto]	切换扩展行格式。

表 1-18 可调节的打印选项

选项	选项说明	取值范围
border	value 必须是一个数字。通常这个数字越大，表的边界就越宽线就越多，但是这个取决于特定的格式。	•在 HTML 格式下，取值范围为大于 0 的整数。 •在其他格式下，取值范围： – 0：无边框 – 1：内部分隔线 – 2：台架

选项	选项说明	取值范围
expanded (或 x)	在正常和扩展格式之间切换。	•当打开扩展格式时，查询结果用两列显示，字段名称在左、数据在右。这个模式在数据无法放进通常的"水平"模式的屏幕时很有用。 •在正常格式下，当查询输出的格式比屏幕宽时，用扩展格式。正常格式只对 aligned 和 wrapped 格式有用。
fieldsep	声明域分隔符来实现非对齐输出。这样就可以创建其他程序希望的制表符或逗号分隔的输出。要设置制表符域分隔符，键入\pset fieldsep '\t'。缺省域分隔符是' ' (竖条符)。	-
fieldsep_zero	声明域分隔符来实现非对齐输出到零字节。	-
footer	用来切换脚注。	-
format	设置输出格式。允许使用唯一缩写（这意味着一个字母就够了）。	取值范围： •unaligned：写一行的所有列在一条直线上中，当前活动字段分隔符分隔。 •aligned：此格式是标准的，可读性最好的文本输出。 •wrapped：类似 aligned，但是包装跨行的宽数据值，使其适应目标字段的宽度输出。 •html：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。 •latex：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。 •troff-ms：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。
null	打印一个字符串，用来代替一个 null 值。	缺省是什么都不打印，这样很容易和空字符串混淆。

选项	选项说明	取值范围
numericlocale	切换分隔小数点左边的数值的区域相关的分组符号。	•on: 显示指定的分隔符。 •off: 不显示分隔符。 忽略此参数, 显示默认的分隔符。
pager	控制查询和 vsql 帮助输出的分页器。如果设置了环境变量 PAGER, 输出将被指向到指定程序, 否则使用系统缺省。	•on: 当输出到终端且不适合屏幕显示时, 使用分页器。 •off: 不使用分页器。 •always: 当输出到终端无论是否符合屏幕显示时, 都使用分页器。
recordsep	声明在非对齐输出格式时的记录分隔符。	-
recordsep_zero	声明在非对齐输出到零字节时的记录分隔符。	-
tableattr (或 T)	声明放在 html 输出格式中 HTML table 标签的属性 (例如: cellpadding 或 bgcolor)。注意: 这里可能不需要声明 border, 因为已经在 \pset border 里用过了。如果没有给出 value, 则不设置表的属性。	-
title	为随后打印的表设置标题。这个可以用于给输出一个描述性标签。如果没有给出 value, 不设置标题。	-
tuples_only (或者 t)	在完全显示和只显示实际的表数据之间切换。完全显示将输出像列头、标题、各种脚注等信息。在 tuples_only 模式下, 只显示实际的表数据。	-

表 1-19 连接元命令

参数	参数说明	取值范围
\c[onnect] [DBNAME - USER - HOST - PORT -]	连接到一个新的数据库 (当前数据库为 postgres)。当数据库名称长度超过 63 个字节时, 默认前 63 个字节	-

参数	参数说明	取值范围
	有效，连接到前 63 个字节对应的数据库，但是 vsql 的命令提示符中显示的数据库对象名仍为截断前的名称。 说明 重新建立连接时，如果切换数据库登录用户，将可能会出现交互式输入，要求输入新用户的连接密码。该密码最长长度为 999 字节，受限与 GUC 参数 password_max_length 的最大值。	
\encoding [ENCODING]	设置客户端字符编码格式。	不带参数时，显示当前的编码格式。
\conninfo	输出当前连接的数据库的信息。	-

表 1-20 操作系统元命令

参数	参数说明	取值范围
\cd [DIR]	切换当前的工作目录。	绝对路径或相对路径，且满足操作系统路径命名规则。
\setenv NAME [VALUE]	设置环境变量 NAME 为 VALUE，如果没有给出 VALUE 值，则不设置环境变量。	-
\timing [on off]	以毫秒为单位显示每条 SQL 语句的执行时间。	• on 表示打开显示。 • off 表示关闭显示。
\! [COMMAND]	返回到一个单独的 Unix shell 或者执行 Unix 命令 COMMAND。	-

表 1-21 变量元命令

参数	参数说明
\prompt [TEXT] NAME	提示用户用文本格式来指定变量名称。
\set [NAME [VALUE]]	设置内部变量 NAME 为 VALUE 或者如果给出了多于一个值，设置为所有这些值的连接结果。如果没有给出第二个参数，只设变量不设值。有一些常用变量被 vsql 特殊对待，它们是一些选项设置，通常所有特殊对待的变量都是由大写字母组成(可能还有数字和下划线)。表 2-21 是一个所有特殊对待的变量列表。
\unset NAME	不设置（或删除）vsqI 变量名。

表 1-22 \set 常用命令

名称	命令说明	取值范围
\set VERBOSITY value	这个选项可以设置为值 default, verbose, terse 之一以控制错误报告的冗余行。	value 取值范围: default, verbose, terse
\set ON_ERROR_STOP value	如果设置了这个变量, 脚本处理将马上停止。如果该脚本是从另外一个脚本调用的, 那个脚本也会按同样的方式停止。如果最外层的脚本不是从一次交互的 vsql 会话中调用的而是用-f 选项调用的, vsql 将返回错误代码 3, 以示这个情况与致命错误条件的区别(错误代码为 1)。	value 取值范围 为: on/off, true/false, yes/no, 1/0
\set RETRY [retry_times]	用于控制是否开启语句出错场景下的重试功能, 参数 retry_times 用来指定最大重试次数, 缺省值为 5, 取值范围为 5-10。当重试功能已经开启时, 再次执行\set RETRY 可以关闭该功能。 使用配置文件 retry_errcodes.conf 列举需要重试的错误码列表, 该文件和 vsql 可执行程序位于同一级目录下。该配置文件为系统配置, 非用户定义, 不允许用户直接修改。 当前支持以下出错场景的重试: •YY001: TCP 通信错误, Connection reset by peer •YY002: TCP 通信错误, Connection reset by peer •YY003: 锁超时, Lock wait timeout.../wait transaction xxx sync time exceed xxx •YY004: TCP 通信错误, Connection timed out •YY005: SET 命令发送失败, ERROR SET query •YY006: 内存申请失败, memory is temporarily unavailable •YY007: 通信库错误, Memory allocate error •YY008: 通信库错误, No data in buffer •YY009: 通信库错误, Close because release memory •YY010: 通信库错误, TCP disconnect •YY011: 通信库错误, SCTP disconnect •YY012: 通信库错误, Stream closed by remote •YY013: 通信库错误, Wait poll unknown error •YY014,YY015,53200,08006,08000,57P01,XX003,XX009	retry_times 取值 范围为: 5-10

名称	命令说明	取值范围
	等 同时，出错时 vsql 会查询数据库节点的连接状态，当状态异常时会 sleep 1 分钟再进行重试，能够覆盖大部分主备切换场景下的出错重试。 说明 1. 不支持事务块中的语句错误重试； 2. 不支持通过 ODBC、JDBC 接口查询的出错重试； 3. 含有 unlogged 表的 sql 语句，不支持节点故障后的出错重试； 4. vsql 客户端本身出现的错误，不在重跑考虑范围之内；	

表 1-23 大对象元命令

参数	参数说明
\lo_list	显示一个目前存储在该数据库里的所有 Vastbase 大对象和提供给他们注释。

PATTERN

很多 \d 命令都可以用一个 PATTERN 参数来指定要被显示的对象名称。在最简单的情况下，PATTERN 正好就是该对象的准确名称。在 PATTERN 中的字符通常会被变成小写形式（就像在 SQL 名称中那样），例如 \dt FOO 将会显示名为 foo 的表。就像在 SQL 名称中那样，把 PATTERN 放在双引号中可以阻止它被转换成小写形式。如果需要在 PATTERN 中包括一个真正的双引号字符，则需要把它写成两个相邻的双引号，这同样是符合 SQL 引用标识符的规则。例如，\dt "FOO""BAR" 将显示名为 FOO"BAR (不是 foo"bar) 的表。和普通的 SQL 名称规则不同，不能只在 PATTERN 的一部分周围放上双引号，例如 \dt FOO"FOO"BAR 将会显示名为 fooFOObar 的表。

不使用 PATTERN 参数时，\d 命令会显示当前 schema 搜索路径中可见的全部对象——这等价于用 * 作为 PATTERN。所谓对象可见是指可以直接用名称引用该对象，而不需要用 schema 来进行限定。要查看数据库中所有的对象而不管它们的可见性，可以把 *.* 用作 PATTERN。

如果放在一个 PATTERN 中，* 将匹配任意字符序列（包括空序列），而 ? 会匹配任意的单个字符（这种记号方法就像 Unix shell 的文件名 PATTERN 一样）。例如，\dt int* 会显示名称以 int 开始的表。但是如果被放在双引号内，* 和 ? 就会失去这些特殊含义而变成普通的字符。

包含一个点号 (.) 的 PATTERN 被解释为一个 schema 名称模式后面跟上一个对象名称模式。例如，\dt foo*.bar* 会显示名称以 foo 开始的 schema 中所有名称包括 bar 的表。如果没有出现点号，那么模式将只匹配当前 schema 搜索路径中可见的对象。同样，双引号内的点号会失去其特殊含义并且变成普通的字符。

高级用户可以使用字符类等正则表达式记法，如[0-9]可以匹配任意数字。所有的正则表达式特殊字符都按照 POSIX 正则表达式所说的工作。以下字符除外：

- ❖ .会按照上面所说的作为一种分隔符。
- ❖ *会被翻译成正则表达式记号.*。
- ❖ ?会被翻译成。
- ❖ \$则按字面意思匹配。

根据需要，可以通过书写?、(R+|)、(R|)和 R?来分别模拟 PATTERN 字符.、R*和 R?。\$不需要作为一个正则表达式字符，因为 PATTERN 必须匹配整个名称，而不是像正则表达式的常规用法那样解释（换句话说，\$会被自动地追加到 PATTERN 上）。如果不希望该 PATTERN 的匹配位置被固定，可以在开头或者结尾写上*。注意在双引号内，所有的正则表达式特殊字符会失去其特殊含义并且按照其字面意思进行匹配。另外，在操作符名称 PATTERN 中（即do 的 PATTERN 参数），正则表达式特殊字符也按照字面意思进行匹配。

2.1.6. 常见问题处理

连接性能问题

- ❖ 开启了 log_hostname，但是配置了错误的 DNS 导致的连接性能问题。

在连接上数据库，通过“show log_hostname”语句，检查数据库中是否开启了 log_hostname 参数。

如果开启了相关参数，那么数据库内核会通过 DNS 反查客户端所在机器的主机名。这时如果数据库主节点配置了不正确的/不可达的 DNS 服务器，那么会导致数据库建立连接过程较慢。此参数的更多信息，详见《开发者指南》中“GUC 参数说明 > 错误报告和日志 > 记录日志的内容”章节中关于“log_hostname”的描述。

- ❖ 数据库内核执行初始化语句较慢导致的性能问题。

此种情况定位较难，可以尝试使用 Linux 的跟踪命令：strace。

```
strace vsql -U MyUserName -W MyPassWord -d postgres -h 127.0.0.1 -p 23508 -r -c '\q'
```

此时便会在屏幕上打印出数据库的连接过程。比如较长时间停留在下面的操作上：

```
sendto(3, "Q\0\0\0\25SELECT VERSION()\0", 22, MSG_NOSIGNAL, NULL, 0) = 22
poll([{fd=3, events=POLLIN|POLLERR}], 1, -1) = 1 ([{fd=3, revents=POLLIN}])
```

此时便可以确定是数据库执行"SELECT VERSION()"语句较慢。

在连接上数据库后，便可以通过执行“explain performance select version()”语句来确定初始化语句执行较慢的原因。更多信息，详见《开发者指南》中“性能调优 > SQL 调优指南 > SQL 执行计划介绍”章节。

另外还有一种场景不太常见：由于数据库主节点所在机器的磁盘满或故障，此时所查询等受影响，无法进行用户认证，导致连接过程挂起，表现为假死。解决此问题清理数据库主节点的磁盘空间便可。

❖ TCP 连接创建较慢问题。

此问题可以参考上面的初始化语句较慢排查的做法，通过 strace 跟踪，如果长时间停留在：

```
connect(3, {sa_family=AF_FILE, path="/home/test/tmp/Vastbase_llt1/.s.Pvsql.61052"}, 110) = 0
```

或者

```
connect(3, {sa_family=AF_INET, sin_port=htons(61052), sin_addr=inet_addr("127.0.0.1")}, 16) = -1 EINPROGRESS (Operation now in progress)
```

那么说明客户端与数据库端建立物理连接过慢，此时应当检查网络是否存在不稳定、网络吞吐量太大的问题。

创建连接故障

❖ vsql: could not connect to server: No route to host

此问题一般是指定了不可达的地址或者端口导致的。请检查-h 参数与-p 参数是否添加正确。

❖ vsql: FATAL: Invalid username/password,login denied.

此问题一般是输入了错误的用户名和密码导致的，请联系数据库管理员，确认用户名和密码的正确性。

❖ The "libpq.so" loaded mismatch the version of vsql, please check it.

此问题是由于环境中使用的 libpq.so 的版本与 vsql 的版本不匹配导致的，请通过“ldd vsql”命令确认当前加载的 libpq.so 的版本，并通过修改 LD_LIBRARY_PATH 环境变量来加载正确的 libpq.so。

❖ vsql: symbol lookup error: xxx/vsql: undefined symbol: libpqVersionString

此问题是由于环境中使用的 libpq.so 的版本与 vsql 的版本不匹配导致的（也有可能是环境中存在 PostgreSQL 的 libpq.so），请通过“ldd vsql”命令确认当前加载的 libpq.so 的版本，并通过修改 LD_LIBRARY_PATH 环境变量来加载正确的 libpq.so。

❖ vsql: connect to server failed: Connection timed out

Is the server running on host "xx.xxx.xxx.xxx" and accepting TCP/IP connections on port xxxx?

此问题是由于网络连接故障造成。请检查客户端与数据库服务器间的网络连接。如果发现从客户端无法 PING 到数据库服务器端，则说明网络连接出现故障。请联系网络管理人员排查解决。

```
ping -c 4 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
From 10.10.10.1: icmp_seq=2 Destination Host Unreachable
From 10.10.10.1 icmp_seq=2 Destination Host Unreachable
From 10.10.10.1 icmp_seq=3 Destination Host Unreachable
From 10.10.10.1 icmp_seq=4 Destination Host Unreachable
--- 10.10.10.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 2999ms
```

❖ vsql: FATAL: permission denied for database "postgres"

DETAIL: User does not have CONNECT privilege.

此问题是由于用户不具备访问该数据库的权限，可以使用如下方法解决。

- a. 使用管理员用户 dbadmin 连接数据库。

```
vsq1 -d postgres -U dbadmin -p 5432
```

- b. 赋予该用户访问数据库的权限。

```
GRANT CONNECT ON DATABASE postgres TO user1;
```

说明

实际上，常见的许多错误操作也可能产生用户无法连接上数据库的现象。如用户连接的数据库不存在，用户名或密码输入错误等。这些错误操作在客户端工具也有相应的提示信息。

```
vsq1 -d postgres -p 5432
vsq1: FATAL: database "postgres" does not exist

vsq1 -d postgres -U user1 -W vastbase@789 -p 5432
vsq1: FATAL: Invalid username/password,login denied.
```

- ❖ vsq1: FATAL: sorry, too many clients already, active/non-active: 197/3.

此问题是由于系统连接数量超过了最大连接数量。请联系数据库 DBA 进行会话连接数管理，释放无用会话。

关于查看用户会话连接数的方法如表 2-23。

会话状态可以在视图 PG_STAT_ACTIVITY 中查看。无用会话可以使用函数 pg_terminate_backend 进行释放。

```
select datid,pid,state from pg_stat_activity;
 datid |      pid      | state 
-----+-----+-----
 13205 | 139834762094352 | active
 13205 | 139834759993104 | idle
(2 rows)
```

其中 pid 的值即为该会话的线程 ID。根据线程 ID 结束会话。

```
SELECT PG_TERMINATE_BACKEND(139834759993104);
```

显示类似如下信息，表示结束会话成功。

```
PG_TERMINATE_BACKEND
-----
t
(1 row)
```

表 1-24 查看会话连接数

描述	命令
查看指定用户的会话连接数上限。	执行如下命令查看连接到指定用户 USER1 的会话连接数上限。其中-1 表示没有对用户 user1 设置连接数的限制。 <pre>SELECT ROLNAME,ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME='user1'; rolname rolconnlimit -----+-----</pre>

描述	命令
	<pre> user1 -1 (1 row) </pre>
查看指定用户已使用的会话连接数。	执行如下命令查看指定用户 USER1 已使用的会话连接数。其中，1 表示 USER1 已使用的会话连接数。 <pre> SELECT COUNT(*) FROM V\$SESSION WHERE USERNAME='user1'; count ----- 1 (1 row) </pre>
查看指定数据库的会话连接数上限。	执行如下命令查看连接到指定数据库 postgres 的会话连接数上限。其中 -1 表示没有对数据库 postgres 设置连接数的限制。 <pre> SELECT DATNAME,DATCONNLIMIT FROM PG_DATABASE WHERE DATNAME='postgres'; datname datconnlimit -----+----- postgres -1 (1 row) </pre>
查看指定数据库已使用的会话连接数。	执行如下命令查看指定数据库 postgres 上已使用的会话连接数。其中，1 表示数据库 postgres 上已使用的会话连接数。 <pre> SELECT COUNT(*) FROM PG_STAT_ACTIVITY WHERE DATNAME='postgres'; count ----- 1 (1 row) </pre>
查看所有用户已使用会话连接数。	执行如下命令查看所有用户已使用的会话连接数。 <pre> SELECT COUNT(*) FROM V\$SESSION; count ----- 10 (1 row) </pre>

❖ vsql: wait xxx.xxx.xxx.xxx:xxxx timeout expired

vsq1 在向数据库发起连接的时候，会有 5 分钟超时机制，如果在这个超时时间内，数据库未能正常的对客户端请求进行校验和身份认证，那么 vsq1 会退出当前会话的连接过程，并报出如上错误。

一般来说，此问题是由于连接时使用的 -h 参数及 -p 参数指定的连接主机及端口有误（即错误信息中的 xxx 部分），导致通信故障；极少数情况是网络故障导致。要排除此问题，请检查数据库的主机名及端口是否正确。

❖ vsql: could not receive data from server: Connection reset by peer.

同时，检查数据库主节点日志中出现类似如下日志 “ FATAL: cipher file

"/data/dbnode/server.key.cipher" has group or world access” ，一般是由于数据目录或部分关键文件的权限被误操作篡改导致。请参照其他正常实例下的相关文件权限，修改回来便可。

- ❖ vsql: FATAL: GSS authentication method is not allowed because XXXX user password is not disabled.

目标数据库主节点的 pg_hba.conf 里配置了当前客户端 IP 使用"gss"方式来做认证，该认证算法不支持用作客户端的身份认证，请修改到"sha256"后再试。配置方法见《开发者指南》中 “管理数据库安全 > 客户端接入认证 > 配置文件参考” 章节 。

说明

- 请不要修改 pg_hba.conf 中 Vastbase 主机的相关设置，否则可能导致数据库功能故障。
- 建议业务应用部署在 Vastbase 之外，而非 Vastbase 内部。

其他故障

- ❖ 出现因“总线错误”（Bus error）导致的 core dump 或异常退出

一般情况下出现此种问题，是进程运行过程中加载的共享动态库（在 Linux 为.so 文件）出现变化；或者进程二进制文件本身出现变化，导致操作系统加载机器的执行码或者加载依赖库的入口发生变化，操作系统出于保护目的将进程杀死，产生 core dump 文件。

解决此问题，重试便可。同时请尽可能避免在升级等运维操作过程中，在 Vastbase 内部运行业务程序，避免升级时因替换文件产生此问题。

说明

此故障的 core dump 文件的可能堆栈是 dl_main 及其子调用，它是操作系统用来初始化进程做共享动态库加载的。如果进程已经初始化，但是共享动态库还未加载完成，严格意义上来说，进程并未完全启动。

3. 服务端工具

在使用 Vastbase 过程中，经常需要对 Vastbase 进行安装、卸载以及健康管理。为了简单、方便的维护 Vastbase，Vastbase 提供了一系列的管理工具。

- [3.1 vb_dump](#)
- [3.2 vb_dumpall](#)
- [3.3 vb_guc](#)
- [3.4 vb_restore](#)

3.1.vb_dump

背景信息

vb_dump 是 Vastbase 用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等）。支持导出的数据库可以是默认数据库 postgres，也可以是自定义数据库。

vb_dump 工具由操作系统用户 omm 执行。

vb_dump 工具在进行数据导出时，其他用户可以访问 Vastbase 数据库（读或写）。

vb_dump 工具支持导出完整一致的数据。例如，T1 时刻启动 vb_dump 导出 A 数据库，那么导出数据结果将会是 T1 时刻 A 数据库的数据状态，T1 时刻之后对 A 数据库的修改不会被导出。

vb_dump 支持将数据库信息导出至纯文本格式的 SQL 脚本文件或其他归档文件中。

- ❖ 纯文本格式的 SQL 脚本文件：包含将数据库恢复为其保存时的状态所需的 SQL 语句。通过 2.1 vsql 运行该 SQL 脚本文件，可以恢复数据库。即使在其他主机和其他数据库产品上，只要对 SQL 脚本文件稍作修改，也可以用来重建数据库。
- ❖ 归档格式文件：包含将数据库恢复为其保存时的状态所需的数据，可以是 tar 格式、目录归档格式或自定义归档格式，详见表 3-7。该导出结果必须与 3.9 vb_restore 配合使用来恢复数据库，vb_restore 工具在导入时，系统允许用户选择需要导入的内容，甚至可以在导入之前对等待导入的内容进行排序。

主要功能

vb_dump 可以创建四种不同的导出文件格式，通过[-F 或者--format=]选项指定，具体如表 3-7 所示。

表 1-25 导出文件格式

格式名称	-F 的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含 SQL 语句和命令。命令可以由 vsql 命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，一般推荐纯文本格式。	使用 2.1 vsql 工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。

格式名称	-F 的参数值	说明	建议	对应导入工具
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，推荐自定义归档格式。	使用 3.9 vb_restore 可以选择要从自定义归档导出文件中导入相应的数据库对象。
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。	-	
tar 归档格式	t	tar 归档文件支持从导出文件中恢复所有或所选数据库对象。tar 归档格式不支持压缩且对于单独表大小应小于 8GB。	-	

📖 说明

可以使用 vb_dump 程序将文件压缩为纯文本或自定义归档导出文件，减少导出文件的大小。生成纯文本导出文件时，默认不压缩。生成自定义归档导出文件时，默认进行中等级别的压缩。vb_dump 程序无法压缩已归档导出文件。通过压缩方式导出纯文本格式文件，vsqf 无法成功导入数据对象。

注意事项

禁止修改导出的文件和内容，否则可能无法恢复成功。

为了保证数据一致性和完整性，vb_dump 会对需要转储的表设置共享锁。如果表在别的事务中设置了共享锁，vb_dump 会等待锁释放后锁定表。如果无法在指定时间内锁定某个表，转储会失败。用户可以通过指定--lock-wait-timeout 选项，自定义等待锁超时时间。

语法

```
vb_dump [OPTION]... [DBNAME]
```

📖 说明

“dbname”前面不需要加短或长选项。“dbname”指定要连接的数据库。

例如：

不需要-d，直接指定“dbname”。

```
vb_dump -p port_number postgres -f dump1.sql
```

或者

```
export PGDATABASE=postgres
vb_dump -p port_number -f dump1.sql
```

环境变量： PGDATABASE

参数说明

通用参数：

❖ -f, --file=FILENAME

将输出发送至指定文件或目录。如果省略该参数，则使用标准输出。如果输出格式为(-F c/-F d/-F t)时，必须指定-f 参数。如果-f 的参数值含有目录，要求目录对当前用户具有读写权限，并且不能指定已有目录。

❖ -F, --format=c|d|t|p

选择输出格式。格式如下：

- p|plain：输出一个文本 SQL 脚本文件（默认）。
- c|custom：输出一个自定义格式的归档，并且以目录形式输出，作为 vb_restore 输入信息。该格式是最灵活的输出格式，因为能手动选择，而且能在恢复过程中将归档项重新排序。该格式默认状态下会被压缩。
- d|directory：该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。
- t|tar：输出一个 tar 格式的归档形式，作为 vb_restore 输入信息。tar 格式与目录格式兼容；tar 格式归档形式在提取过程中会生成一个有效的目录格式归档形式。但是，tar 格式不支持压缩且对于单独表有 8GB 的大小限制。此外，表数据项的相应排序在恢复过程中不能更改。

输出一个 tar 格式的归档形式，也可以作为 vsql 输入信息。

❖ -v, --verbose

指定 verbose 模式。该选项将导致 vb_dump 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。

❖ -V, --version

打印 vb_dump 版本，然后退出。

❖ -Z, --compress=0-9

指定使用的压缩比级别。

取值范围：0~9

- 0 表示无压缩。
- 1 表示压缩比最小，处理速度最快。
- 9 表示压缩比最大，处理速度最慢。

针对自定义归档格式，该选项指定单个表数据片段的压缩，默认方式是以中等级别进行压缩。p、tar 归档格式目前不支持压缩。

❖ --lock-wait-timeout=TIMEOUT

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 SET statement_timeout 的格式指定超时时间。

❖ -?, --help

显示 vb_dump 命令行参数帮助，然后退出。

转储参数：

❖ -a, --data-only

只输出数据，不输出模式(数据定义)。转储表数据、大对象和序列值。

❖ -b, --blobs

该参数为扩展预留接口，不建议使用。

❖ -c, --clean

在将创建数据库对象的指令输出到备份文件之前，先将清理（删除）数据库对象的指令输出到备份文件中。（如果目标数据库中没有任何对象，vb_restore 工具可能会输出一些提示性的错误信息）

该选项只对文本格式有意义。针对归档格式，可以调用 vb_restore 时指定选项。

❖ -C, --create

备份文件以创建数据库和连接到创建的数据库的命令开始。（如果命令脚本是这种方式执行，无所谓在运行脚本之前连接的是哪个数据库。）

该选项只对文本格式有意义。针对归档格式，可以在调用 vb_restore 时指定选项。

❖ -E, --encoding=ENCODING

以指定的字符集编码创建转储。默认情况下，以数据库编码创建转储。（得到相同结果的另一个办法是将环境变量“PGCLIENTENCODING”设置为所需的转储编码。）

❖ -n, --schema=SCHEMA

只转储与模式名称匹配的模式，此选项包括模式本身和所有它包含的对象。如果该选项没有指定，所有在目标数据库中的非系统模式将会被转储。写入多个-n 选项来选择多个模式。此外，根据 vsql 的 \d 命令所使用的相同规则，模式参数可被理解成一个 pattern，所以多个模式也可以通过在该 pattern 中写入通配符来选择。使用通配符时，注意给 pattern 打引号，防止 shell 扩展通配符。

📖 说明

- 当-n 已指定时，vb_dump 不会转储已选模式所附着的任何其他数据库对象。因此，无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当-n 指定时，非模式对象不会被转储。

转储支持多个模式的转储。多次输入-n schemaname 转储多个模式。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sch12.sql -n sch1 -n sch2
```

在上面这个例子中，sch1 和 sch2 会被转储。

❖ -N, --exclude-schema=SCHEMA

不转储任何与模式 pattern 匹配的模式。Pattern 将参照针对-n 的相同规则来理解。可以通过输入多次-N，不转储与任何 pattern 匹配的模式。

当同时输入-n 和-N 时，会转储与至少一个-n 选项匹配、与-N 选项不匹配的模式。如果有-N 没有-n，则不转储常规转储中与-N 匹配的模式。

转储过程支持排除多个模式。

在转储过程中，输入-N exclude schema name 排除多个模式。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sch12.sql -N sch1 -N sch2
```

在上面这个例子中，sch1 和 sch2 在转储过程中会被排除。

❖ -o, --oids

转储每个表的对象标识符 (OIDs)，作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

❖ -O, --no-owner

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，vb_dump 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的数据库对象的归属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定-O，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限。

该选项只对文本格式有意义。针对归档格式，可以在调用 vb_restore 时指定选项。

❖ -s, --schema-only

只转储对象定义（模式），而非数据。

❖ -S, --sysadmin=NAME

该参数为扩展预留接口，不建议使用。

❖ -t, --table=TABLE

指定转储的表（或视图、或序列、或外表）对象列表，可以使用多个-t 选项来选择多个表，也可以使用通配符指定多个表对象。

当使用通配符指定多个表对象时，注意给 pattern 打引号，防止 shell 扩展通配符。

当使用-t 时，-n 和-N 没有任何效应，这是因为由-t 选择的表的转储不受那些选项的影响。

📖 说明

- -t 参数选项个数必须小于等于 100。
- 如果-t 参数选项个数大于 100，建议使用参数--include-table-file 来替换。
- 当-t 已指定时，vb_dump 不会转储已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。

- -t tablename 只转储在默认搜索路径中可见的表。-t *.tablename 转储数据库下所有模式下的 tablename 表。
- -t schema.table 转储特定模式中的表。
- -t tablename 不会导出表上的触发器信息。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sh12.sql -t schema1.table1 -t schema2.table2
```

在上面这个例子中，schema1.table1 和 schema2.table2 会被转储。

❖ --include-table-file=FILENAME

指定需要 dump 的表文件。

❖ -T, --exclude-table=TABLE

不转储的表（或视图、或序列、或外表）对象列表，可以使用多个 -t 选项来选择多个表，也可以使用通配符指定多个表对象。

当同时输入 -t 和 -T 时，会转储在 -t 列表中，而不在 -T 列表中的表对象。

例如：

```
vb_dump -h host_name -p port_number postgres -f backup/bkp_sh12.sql -T table1 -T table2
```

在上面这个例子中，table1 和 table2 在转储过程中会被排除。

❖ --exclude-table-file=FILENAME

指定不需要 dump 的表文件。

📖 说明

同 --include-table-file，其内容格式如下：

```
schema1.table1
schema2.table2
.....
```

❖ -x, --no-privileges|--no-acl

防止转储访问权限（授权/撤销命令）。

❖ --binary-upgrade

该参数为扩展预留接口，不建议使用。

❖ --binary-upgrade-usermap="USER1=USER2"

该参数为扩展预留接口，不建议使用。

❖ --column-inserts|--attribute-inserts

以 INSERT 命令带列名（INSERT INTO 表（列、...）值...）方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。

❖ --disable-dollar-quoting

该选项将禁止在函数体前使用美元符号 \$，并强制使用 SQL 标准字符串语法对其进行引用。

❖ --disable-triggers

该参数为扩展预留接口，不建议使用。

❖ `--exclude-table-data=TABLE`

指定不转储任何匹配表 pattern 的表这方面的数据。依照针对-t 的相同规则理解该 pattern。

可多次输入--exclude-table-data 来排除匹配任何 pattern 的表。当用户需要特定表的定义但不需要其中的数据时，这个选项很有帮助。

排除数据库中所有表的数据，参见[--schema-only](#)。

❖ `--inserts`

发出 INSERT 命令（而非 COPY 命令）时转储数据。这会导致恢复缓慢。

但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。注意如果重排列顺序，可能会导致恢复整个失败。列顺序改变时，--column-inserts 选项不受影响，虽然会更慢。

❖ `--no-security-labels`

该参数为扩展预留接口，不建议使用。

❖ `--no-tablespaces`

不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。

该选项只对文本格式有意义。针对归档格式，可以在调用 `vb_restore` 时指定选项。

❖ `--no-unlogged-table-data`

该参数为扩展预留接口，不建议使用。

❖ `--non-lock-table`

该参数为扩展预留接口，不建议使用。

❖ `--include-alter-table`

转储表删除列。该选项会记录列的删除。

❖ `--quote-all-identifiers`

强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

❖ `--section=SECTION`

指定已转储的名称区段（pre-data、data、和 post-data）。

❖ `--serializable-deferrable`

转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致；要实现该操作需要在无异常状况的事务流中等待某个点，因为这样才能保证转储成功，避免引起其他事务出现 `serialization_failure` 要重新再做。

但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候，加载一个数据库的拷贝作为报告或其他只读加载共享的转储是有帮助的。没有这个选项，转储会反映一个与任何事务最终提交的序列化执行不一致的状态。

如果当 `vb_dump` 启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始时间可能会延迟一段不确定的时间。

❖ `--use-set-session-authorization`

输出符合 SQL 标准的 `SET SESSION AUTHORIZATION` 命令而不是 `ALTER OWNER` 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 `SET SESSION AUTHORIZATION` 的转储需要数据库系统管理员的权限才能转储成功，而 `ALTER OWNER` 需要的权限则低得多。

❖ `--with-encryption=AES128`

指定转储数据需用 AES128 进行加密。

❖ `--with-key=KEY`

AES128 密钥长度必须是 16 字节。

❖ `--include-depend-objs`

备份结果包含依赖于指定对象的对象信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

❖ `--exclude-self`

备份结果不包含指定对象自身的信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

❖ `--dont-overwrite-file`

文本、tar、以及自定义格式情况下会重写现有文件。这对目录格式不适用。

例如：

设想这样一种情景，即当前目录下 `backup.sql` 已存在。如果在输入命令中输入 `-f backup.sql` 选项时，当前目录恰好也生成 `backup.sql`，文件就会被重写。

如果备份文件已存在，且输入 `--dont-overwrite-file` 选项，则会报告附带‘转储文件已经存在’信息的错误。

```
vb_dump -p port_number postgres -f backup.sql -F plain --dont-overwrite-file
```

📖 说明

- `-s/--schema-only` 和 `-a/--data-only` 不能同时使用。
- `-c/--clean` 和 `-a/--data-only` 不能同时使用。
- `--inserts/--column-inserts` 和 `-o/--oids` 不能同时使用，因为 `INSERT` 命令不能设置 OIDS。
- `--role` 和 `--rolepassword` 必须一起使用。
- `--binary-upgrade-usermap` 和 `--binary-upgrade` 必须一起使用。
- `--include-depend-objs/--exclude-self` 需要同 `-t/--include-table-file` 参数关联使用才会生效
- `--exclude-self` 必须同 `--include-depend-objs` 一起使用。

连接参数：

❖ `-h, --host=HOSTNAME`

指定主机名称。如果数值以斜杠开头，则被用作到 Unix 域套接字的路径。缺省从 PGHOST 环境变量中获取（如果已设置），否则，尝试一个 Unix 域套接字连接。

该参数只针对 Vastbase 外，对 Vastbase 内本机只能用 127.0.0.1。

例如：主机名

环境变量：PGHOST

❖ -p, --port=PORT

指定主机端口。在开启线程池情况下，建议使用 pooler port，即主机端口+1。

环境变量：PGPORT

❖ -U, --username=NAME

指定所连接主机的用户名。

不指定连接主机的用户名时，用户默认系统管理员。

环境变量：PGUSER

❖ -w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是 trust，则不会对系统管理员进行密码验证，即无需输入-W 选项；如果没有-W 选项，并且不是系统管理员，“Dump Restore 工具”会提示用户输入密码。

❖ --role=ROLENAME

指定创建转储使用的角色名。选择该选项，会使 vb_dump 连接数据库后，发起一个 SET ROLE 角色名命令。当所授权用户（由-U 指定）没有 vb_dump 要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

❖ --rolepassword=ROLEPASSWORD

指定角色名的密码。

说明

如果 Vastbase 有任何本地数据要添加到 template1 数据库，请谨慎将 vb_dump 的输出恢复到一个真正的空数据库中，否则可能会因为被添加对象的定义被复制，出现错误。要创建一个无本地添加的空数据库，需从 template0 而非 template1 复制，例如：

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

tar 归档形式的文件大小不得超过 8GB（tar 文件格式的固有限制）。tar 文档整体大小和任何其他输出格式没有限制，操作系统可能对此有要求。

由 vb_dump 生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此，最好从某转储文件恢复之后运行 ANALYZE 以确保最佳效果。转储文件不包含任何 ALTER DATABASE...SET 命令，这些设置由 vb_dumpall 转储，还有数据库用户和其他完成安装设置。

示例

使用 vb_dump 转储数据库为 SQL 文本文件或其它格式的操作，如下所示。

示例中“Bigdata@123”表示数据库用户密码；“backup/MPPDB_backup.sql”表示导出的文件，其中 backup 表示相对于当前目录的相对目录；“37300”表示数据库服务器端口；“postgres”表示要访问的数据库名。

说明

导出操作时，请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例 1：执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.sql 文件格式为纯文本格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.sql -p 37300 postgres -F p
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: The total objects number is 356.
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: [100.00%] 356 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 09:49:17]: total time: 1274 ms
```

使用 vsql 程序从纯文本导出文件中导入数据。

示例 2：执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.tar 文件格式为 tar 格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.tar -p 37300 postgres -F t
vb_dump[port='37300'][postgres][2018-06-27 10:02:24]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:02:53]: total time: 50086 ms
```

示例 3：执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup.dmp 文件格式为自定义归档格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.dmp -p 37300 postgres -F c
vb_dump[port='37300'][postgres][2018-06-27 10:05:40]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:06:03]: total time: 36620 ms
```

示例 4：执行 vb_dump，导出 postgres 数据库全量信息，导出的 MPPDB_backup 文件格式为目录格式。

```
vb_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup -p 37300 postgres -F d
vb_dump[port='37300'][postgres][2018-06-27 10:16:04]: The total objects number is 1369.
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: [100.00%] 1369 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:16:23]: total time: 33977 ms
```

示例 5：执行 vb_dump，导出 postgres 数据库信息，但不导出/home/MPPDB_temp.sql 中指定的表信息。导出的 MPPDB_backup.sql 文件格式为纯文本格式。

```
vb_dump -U omm -W Bigdata@123 -p 37300 postgres --exclude-table-file=/home/MPPDB_temp.sql -f
backup/MPPDB_backup.sql
vb_dump[port='37300'][postgres][2018-06-27 10:37:01]: The total objects number is 1367.
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: [100.00%] 1367 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-27 10:37:22]: total time: 37017 ms
```

示例 6：执行 vb_dump，仅导出依赖于指定表 testtable 的视图信息。然后创建新的 testtable 表，再恢复依赖其上的视图。

- ❖ 备份仅依赖于 testtable 的视图。

```
vb_dump -s -p 37300 postgres -t PUBLIC.testtable --include-depend-objs --exclude-self -f
backup/MPPDB_backup.sql -F p
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: The total objects number is 331.
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: [100.00%] 331 objects have been dumped.
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: dump database postgres successfully
vb_dump[port='37300'][postgres][2018-06-15 14:12:54]: total time: 327 ms
```

- ❖ 修改 testtable 名称。

```
vsq1 -p 37300 postgres -r -c "ALTER TABLE PUBLIC.testtable RENAME TO testtable_bak;"
```

- ❖ 创建新的 testtable 表。

```
CREATE TABLE PUBLIC.testtable(a int, b int, c int);
```

- ❖ 还原依赖于 testtable 的视图。

```
vsq1 -p 37300 postgres -r -f backup/MPPDB_backup.sql
```

相关命令

3.6 vb_dumpall, 3.9 vb_restore

3.2. vb_dumpall

背景信息

vb_dumpall 是 Vastbase 用于导出所有数据库相关信息工具，它可以导出 Vastbase 数据库的所有数据，包括默认数据库 postgres 的数据、自定义数据库的数据、以及 Vastbase 所有数据库公共的全局对象。

vb_dumpall 工具由操作系统用户 omm 执行。

vb_dumpall 工具在进行数据导出时，其他用户可以访问 Vastbase 数据库（读或写）。

vb_dumpall 工具支持导出完整一致的数据。例如，T1 时刻启动 vb_dumpall 导出 Vastbase 数据库，那么导出数据结果将会是 T1 时刻该 Vastbase 数据库的数据状态，T1 时刻之后对 Vastbase 的修改不会被导出。

vb_dumpall 在导出 Vastbase 所有数据库时分为两部分：

- ❖ vb_dumpall 自身对所有数据库公共的全局对象进行导出，包括有关数据库用户和组，表空间以及属性（例如，适用于数据库整体的访问权限）信息。
- ❖ vb_dumpall 通过调用 vb_dump 来完成 Vastbase 中各数据库的 SQL 脚本文件导出，该脚本文件包含将数据库恢复为其保存时的状态所需要全部 SQL 语句。

以上两部分导出的结果为纯文本格式的 SQL 脚本文件，使用 2.1 vsq1 运行该脚本文件可以恢复 Vastbase 数据库。

注意事项

- ❖ 禁止修改导出的文件和内容，否则可能无法恢复成功。
- ❖ 为了保证数据一致性和完整性，vb_dumpall 会对需要转储的表设置共享锁。如果某张表在别的事务中设置了共享锁，vb_dumpall 会等待此表的锁释放后锁定此表。如果无法在指定时间内锁定某张表，转储会失败。用户可以通过指定--lock-wait-timeout 选项，自定义等待锁超时时间。
- ❖ 由于 vb_dumpall 读取所有数据库中的表，因此必须以 Vastbase 管理员身份进行连接，才能导出完整文件。在使用 vsql 执行脚本文件导入时，同样需要管理员权限，以便添加用户和组，以及创建数据库。

语法

```
vb_dumpall [OPTION]...
```

参数说明

通用参数：

- ❖ -f, --filename=FILENAME
将输出发送至指定文件。如果这里省略，则使用标准输出。
- ❖ -v, --verbose
指定 verbose 模式。该选项将导致 vb_dumpall 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。
- ❖ -V, --version
打印 vb_dumpall 版本，然后退出。
- ❖ --lock-wait-timeout=TIMEOUT
请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 SET statement_timeout 的格式指定超时时间。
- ❖ -?, --help
显示 vb_dumpall 命令行参数帮助，然后退出。

转储参数：

- ❖ -a, --data-only
只转储数据，不转储模式（数据定义）。
- ❖ -c, --clean
在重新创建数据库之前，执行 SQL 语句清理（删除）这些数据库。针对角色和表空间的转储命令已添加。
- ❖ -g, --globals-only
只转储全局对象（角色和表空间），无数据库。
- ❖ -o, --oids
转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

- ❖ `-O, --no-owner`
不输出设置对象的归属这样的命令,以匹配原始数据库。默认情况下, `vb_dumpall` 会发出 `ALTER OWNER` 或 `SET SESSION AUTHORIZATION` 语句设置所创建的模式元素的所属。如果脚本正在运行,该语句不会执行成功,除非是由系统管理员触发(或是拥有脚本中所有对象的同一个用户)。通过指定 `-O`, 编写一个任何用户都能存储的脚本,且该脚本会授予该用户拥有所有对象的权限。
- ❖ `-r, --roles-only`
只转储角色,不转储数据库或表空间。
- ❖ `-s, --schema-only`
只转储对象定义(模式),而非数据。
- ❖ `-S, --sysadmin=NAME`
在转储过程中使用的系统管理员名称。
- ❖ `-t, --tablespaces-only`
只转储表空间,不转储数据库或角色。
- ❖ `-x, --no-privileges`
防止转储访问权限(授权/撤销命令)。
- ❖ `--column-inserts|--attribute-inserts`
以 `INSERT` 命令带列名(`INSERT INTO 表(列、...) 值...`)方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致该行丢失,而非整个表内容。
- ❖ `--disable-dollar-quoting`
该选项将禁止在函数体前使用美元符号\$,并强制使用 SQL 标准字符串语法对其进行引用。
- ❖ `--disable-triggers`
该参数为扩展预留接口,不建议使用。
- ❖ `--inserts`
发出 `INSERT` 命令(而非 `COPY` 命令)时转储数据。这会导致恢复缓慢。注意如果重排列顺序,可能会导致恢复整个失败。`--column-inserts` 选项更加安全,虽然可能更慢些。
- ❖ `--no-security-labels`
该参数为扩展预留接口,不建议使用。
- ❖ `--no-tablespaces`
请勿输出创建表空间的命令,也请勿针对对象选择表空间。使用该选项,无论默认表空间是哪个,在恢复过程中所有对象都会被创建。
- ❖ `--no-unlogged-table-data`
该参数为扩展预留接口,不建议使用。
- ❖ `--quote-all-identifiers`

强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

❖ --dont-overwrite-file

不重写当前文件。

❖ --use-set-session-authorization

输出符合 SQL 标准的 SET SESSION AUTHORIZATION 命令而不是 ALTER OWNER 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 SET SESSION AUTHORIZATION 的转储需要数据库系统管理员的权限才能转储成功，而 ALTER OWNER 需要的权限则低得多。

❖ --with-encryption=AES128

指定转储数据需用 AES128 进行加密。

❖ --with-key=KEY

AES128 密钥长度必须是 16 字节。

❖ --include-templatedb

转储过程中包含模板库。

❖ --binary-upgrade

该参数为扩展预留接口，不建议使用。

❖ --binary-upgrade-usermap="USER1=USER2"

该参数为扩展预留接口，不建议使用。

❖ --tablespaces-postfix

该参数为扩展预留接口，不建议使用。

❖ --parallel-jobs

指定备份进程并发数，取值范围为 1~1000。

📖 说明

- -g/--globals-only 和 -r/--roles-only 不能同时使用。
- -g/--globals-only 和 -t/--tablespaces-only 不能同时使用。
- -r/--roles-only 和 -t/--tablespaces-only 不能同时使用。
- -s/--schema-only 和 -a/--data-only 不能同时使用。
- -r/--roles-only 和 -a/--data-only 不能同时使用。
- -t/--tablespaces-only 和 -a/--data-only 不能同时使用。
- -g/--globals-only 和 -a/--data-only 不能同时使用。
- --tablespaces-postfix 和 --binary-upgrade 必须一起使用。
- --binary-upgrade-usermap 和 --binary-upgrade 必须一起使用。
- --parallel-jobs 和 -f/--file 必须一起使用。

连接参数：

❖ -h, --host

指定主机的名称。如果取值是以斜线开头,它将用作 Unix 域套接字的目录。默认值取自 PGHOST 环境变量;如果没有设置,将启动某个 Unix 域套接字建立连接。

该参数只针对 Vastbase 外,对 Vastbase 内本机只能用 127.0.0.1。

环境变量: PGHOST

❖ -l, --database

指定所连接的转储全局对象的数据库名称,并去寻找还有其他哪些数据库需要被转储。如果没有指定,会使用 postgres 数据库,如果 postgres 数据库不存在,会使用 template1。

❖ -p, --port

指定服务器所监听的 TCP 端口或本地 Unix 域套接字后缀,以确保连接。默认值设置为 PGPORT 环境变量。

在开启线程池情况下,建议使用 pooler port,即监听端口+1。

环境变量: PGPORT

❖ -U, --username

所连接的用户名。

环境变量: PGUSER

❖ -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出,则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password

指定用户连接的密码。如果主机的认证策略是 trust,则不会对系统管理员进行密码验证,即无需输入 -W 选项;如果没有 -W 选项,并且不是系统管理员,“Dump Restore 工具”会提示用户输入密码。

❖ --role

指定创建转储使用的角色名。选择该选项,会使 vb_dumpall 连接数据库后,发起一个 SET ROLE 角色名命令。当所授权用户(由 -U 指定)没有 vb_dumpall 要求的权限时,该选项会起到作用,即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录,而使用该选项能够在不违反该规定的情况下完成转储。

❖ --rolepassword

指定具体角色用户的角色密码。

说明

由于 vb_dumpall 内部调用 vb_dump,所以一些诊断信息参见 3.5 vb_dump。

一旦恢复,最好在每个数据库上运行 ANALYZE,优化程序提供有用的统计数据。

vb_dumpall 恢复前需要所有必要的表空间目录才能退出;否则,对于处在非默认位置的数据库,数据库创建会失败。

示例

使用 vb_dumpall 一次导出 Vastbase 的所有数据库。

说明

vb_dumpall 仅支持纯文本格式导出。所以只能使用 vsql 恢复 vb_dumpall 导出的转储内容。

```
vb_dumpall -f backup/bkp2.sql -p 37300
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:09]: The total objects number is 2371.
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:35]: [100.00%] 2371 objects have been
dumped.
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: dump database dbname='postgres'
successfully
vb_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: total time: 55567 ms
vb_dumpall[port='37300'][2018-06-27 09:55:46]: dumpall operation successful
vb_dumpall[port='37300'][2018-06-27 09:55:46]: total time: 56088 ms
```

相关命令

3.5 vb_dump, 3.9 vb_restore

3.3. vb_guc

背景信息

目前 Vastbase 配置文件 (“postgresql.conf”、“pg_hba.conf”) 中的参数默认值都是单机的配置模式。应用程序可以通过调用 vb_guc 来设置适合自己的参数。

vb_guc 工具由操作系统用户 omm 执行。

❖ “vb_guc-current.log”

这个文件记录 vb_guc 产生的日志。

默认路径: \$GAUSSLOG/bin/vb_guc。

❖ “vb_guc-年-月-日_HHMMSS.log”

当 “vb_guc-current.log” 文件大小满 16M 后, 会根据当前时刻生成一个备份文件。

❖ “server.key.cipher”, “server.key.rand”

使用 vb_guc encrypt 命令加密用户密码时, 如果指定的-M 的值为 server, 则会生成这两个文件。

其中 “server.key.cipher” 存储用户密码的密文, “server.key.rand” 存储的是加密因子。

❖ “client.key.cipher”, “client.key.rand”

使用 vb_guc encrypt 命令加密用户密码时, 如果指定的-M 的值为 client, 则会生成这两个文件。

其中 “client.key.cipher” 存储用户密码的密文, “client.key.rand” 存储的是加密因子。

说明

“client.key.cipher”和“client.key.rand”是不指定-U 选项时生成的密文文件和加密因子文件。如果通过-U 指定了用户名, 则会生成以用户名开头的文件。以指定-U test 为例, 生成的文件名为:“test.key.cipher、test.key.rand”。

- ❖ “datasource.key.cipher”，“datasource.key.rand”

使用 `vb_guc encrypt` 命令加密用户密码时，如果指定的-M 的值为 source，则会生成这两个文件。

其中 “datasource.key.cipher” 存储用户密码的密文，“datasource.key.rand” 存储的是加密因子。

📖 说明

“datasource.key.cipher”和“datasource.key.rand”是创建 Data Source 对象时调用的密钥文件。vb_guc 生成时即有读权限。使用前需将这两个文件放入各节点目录\$GAUSSHOME/bin，且确保具有读权限。

语法

- ❖ 检查配置文件中参数

```
vb_guc check [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter"
```

- ❖ 修改配置文件中参数

```
vb_guc set [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter = value"
```

- ❖ 将已设置的参数值修改为默认值

```
vb_guc [ set | reload ] [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter"
```

- ❖ 修改配置文件中参数，同时发送信号量到 postgresql.conf

```
vb_guc reload [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c parameter=value
```

- ❖ 修改客户端认证策略，同时发送信号量到 pg_hba.conf

```
vb_guc [ set | reload ] [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -h "HOSTTYPE DATABASE  
USERNAME IPADDR-WITH-IPMASK AUTHMEHOD authentication-options option"
```

📖 说明

authmehod-options 支持以下选项：

- trust：不验密，禁止远程主机使用 trust 方式访问 Vastbase
- reject：拒绝访问
- md5：md5 认证，默认不支持
- sha256：sha256 认证（推荐使用）
- cert：客户端证书认证
- gss：kerberos 认证

- ❖ 注释已经设置的客户端认证策略，同时发送信号量到 pg_hba.conf

```
vb_guc [ set | reload ] [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -h "HOSTTYPE DATABASE  
USERNAME IPADDR-WITH-IPMASK AUTHMEHOD"
```

📖 说明

如果需要注释已经设置的认证策略，请使用-h "HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK AUTHMEHOD"，不需要指定 authmehod-options。

- ❖ 显示帮助信息

```
vb_guc -? | --help
```

- ❖ 显示版本号信息

```
vb_guc -V | --version
```

- ❖ 生成加密密码文件

```
vb_guc encrypt [-M keymode] -K password [-U username] -D DATADIR
```

📖 说明

-K 是用户指定的密码，vb_guc 会对该密码进行长度 ($8 \leq \text{len} \leq 16$) 和密码复杂度要求，如果不满足，将会报错。

-M 是加密类型，当前仅支持 server、client 和 source。默认值为 server。

📖 说明

- vb_guc 工具不支持参数值中包含 '#' 的设置。可以使用 vi 工具通过手工修改配置文件来设置。
- 如果已经在环境变量中设置 PGDATA，则可以省略 -D 参数。否则提示设置参数失败。
- 环境变量 PGDATA 设置方法：先将参数 PGDATA 设置为 /Vastbase/data，然后在执行 "export PGDATA" 使设置生效。
- authpolicy 包含一串认证参数：HOSTTYPE DATABASE USERNAME IPADDR IPMASK，或者 HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK，或者 HOSTTYPE DATABASE USERNAME HOSTNAME。
- 如果设置 GUC 参数时使用 -c "parameter"，则会将已设置的 GUC 参数值设置成该参数的内核默认值（注意 log_directory 和 audit_directory 不会被设置为内核参数默认值，而是设为 \$GAUSSLOG/pg_audit/instance_name）。因 GUC 参数间存在依赖关系，因此请慎用该功能。
- 设置 -c 参数时，参数都可以省略双引号。
- 如果 value 中含有特殊字符（如 \$），请转义后使用。
- 如果同一个配置参数在配置文件里面出现多行，且有两行或多于两行同时生效（即没有用 '#' 注释掉），那么只有最后一个配置参数会被设置，而前面的都会被忽略。
- 通过 reload 模式设置或修改 Vastbase 节点配置文件（postgresql.conf）的参数，生效存在短暂延迟，有可能导致配置后 Vastbase 各实例参数极短时间不一致。

命令参考

❖ set

表示只修改配置文件中的参数。

❖ check

表示只检查配置文件中的参数。

❖ reload

表示修改配置文件中的参数，同时发送信号量给数据库进程，使其重新加载配置文件。

❖ encrypt/generate

生成加密密码文件。

❖ -N

需要设置的主机名称。

取值范围：已有主机名称。

当参数取值为 ALL 时，表示设置 Vastbase 中所有的主机。

❖ -I INSTANCE-NAME

需要设置的实例名称。

取值范围：已有实例名称。

当参数取值为 ALL 时，表示设置主机中所有的实例。

❖ -D

需要执行命令的 Vastbase 实例路径。使用 encrypt 命令时，此参数表示指定的密码文件生成的路径。

📖 说明

- 与“-l”不能一块使用

❖ -c parameter=value

要设定的 Vastbase 配置参数的名称和参数值。

📖 说明

- 如果参数是一个字符串变量，则使用-c parameter="value"或者使用-c "parameter = 'value'"。
- 当使用 vb_guc set/reload 为"log_directory" 恢复默认值时，其默认值会被置为具体的 data 目录。
- 当使用 vb_guc reload 进行参数设定，并指定-N 参数时，当指定的节点为主节点时，主备节点的参数值都会被修改；当指定节点为备节点时，只会修改备节点的参数值，不会修改主节点的参数值。
- 当使用 vb_guc reload 进行参数设定，未指定-N 参数时，当在主节点上执行时，主备节点的参数值都会被修改；当在备节点上执行时，只会修改备节点的参数值，不会修改主节点的参数值。

取值范围：postgresql.conf 中的所有参数。

❖ -c parameter

当进行 check 操作时，表示需要检查的参数名称；当进行 set/reload 操作时，表示需要恢复为数据库默认值的参数名称。

❖ -h host-auth-policy

指定需要在“pg_hba.conf”增加的客户端认证策略。

取值范围：

- HOSTTYPE DATABASE USERNAME IPADDR IPMASK [authmethod-options]
- HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK [authmethod-options]
- HOSTTYPE DATABASE USERNAME HOSTNAME [authmethod-options]

HOSTTYPE 是必输参数，取值：

- local
- host
- hostssl
- hostnossl

其中“local”是一个 Unix 域套接字，“host”是普通或 SSL 加密的 TCP / IP 套接字，“hostssl”是 SSL 加密的 TCP / IP 套接字，“hostnossl”是一个纯 TCP / IP 套接字。

authmethod-options 是可选输入，取值：

- trust
- reject
- md5
- sha256
- cert
- gss

详细的参数说明请参见“pg_hba.conf”配置文件中的描述。

- ❖ `-?, --help`
显示帮助信息。
- ❖ `-V, --version`
显示版本信息。
- ❖ `-M, --keymode=MODE`
设定该密码在数据库运行过程中，用于服务端(server)、客户端(client)还是创建 Data Source 对象时调用。

📖 说明

在使用 ssl 功能时，会涉及到服务端证书和私钥文件的加密密码和客户端证书和私钥文件的加密密码，在加密存储加密密码时，需要指定模式，否则默认是生成服务端的密码文件。

取值范围：

- server (缺省)
- client
- source

- ❖ `-K PASSWORD`
指定需要加密的密码。
取值范围：字符串，符合密码复杂度的要求。

- ❖ `-U, --keyuser=USER`
指定要加密的用户，隶属于 OS 用户。

📖 说明

比如，Vastbase 可以为每个用户配置不同的 ssl 证书和私钥文件，指定该选项，可以生成按用户名区分的密码文件。

- ❖ `-S CIPHERKEY`
指定需要加密的字符串。
取值范围：字符串。

示例

示例 1：修改数据库允许的最大连接数为 800。修改后需要重启数据库才能生效。

```
vb_guc set -D /Vastbase/data/coordinator -c "max_connections = 800" -N plat1

Total instances: 1. Failed instances: 0.
Success to perform vb_guc!
```

示例 2：将已设置的数据库允许的最大连接数值修改为默认值。修改后需要重启数据库才能生效。

```
vb_guc set -D /Vastbase/data/dbnode -c "max_connections" -N plat1

Total instances: 1. Failed instances: 0.
Success to perform vb_guc!
```

示例 3：设置客户端认证策略

```
vb_guc set -N all -I all -h "host replication testuser 10.252.95.191/32 sha256"
```

```
Total instances: 2. Failed instances: 0.  
Success to perform vb_guc!
```

示例 4：注释清理已经设置的客户端认证策略

```
vb_guc set -N all -I all -h "host replication testuser 10.252.95.191/32"
```

```
Total instances: 2. Failed instances: 0.  
Success to perform vb_guc!
```

示例 5：加密服务器端存储用户密码的明文，避免密码泄露。

```
vb_guc encrypt -K Vastbase@123 -D /data/xzg/wisquery/Vastbase  
vb_guc encrypt -K ***
```

示例 6：检查 Vastbase 中各个实例的参数配置情况。

```
vb_guc check -D /Vastbase/data/dbnode -c "max_connections" -N plat1  
  
Total GUC values: 1. Failed GUC values: 0.  
The value of parameter max_connections is same on all instances.  
max_connections=800
```

示例 7：修改 archive_command 归档命令，由用户自己保证归档路径的有效性。修改后需要重启数据库才能生效。推荐使用此方法。

```
vb_guc set -N all -D /data/datanode -c "archive_command= 'cp %p  
/data/datanode/pg_xlog/archive_status/%f'"  
Total instances: 3. Failed instances: 0.  
Success to perform vb_guc!
```

示例 8：通过用户输入的明文密码，生自定义的 test.key.cipher 和 test.key.rand 文件。

```
vb_guc generate -S Mppdb@123 -D data/xzg/wisquery/Vastbase -o test  
vb_guc encrypt -S ***
```

3.4. vb_restore

背景信息

vb_restore 是 Vastbase 提供的针对 vb_dump 导出数据的导入工具。通过此工具可由 vb_dump 生成的导出文件进行导入。

vb_restore 工具由操作系统用户 omm 执行。

主要功能包含：

- ❖ 导入到数据库

如果连接参数中指定了数据库，则数据将被导入到指定的数据库中。其中，并行导入必须指定连接的密码。

- ❖ 导入到脚本文件

如果未指定导入数据库，则创建包含重建数据库所必须的 SQL 语句脚本并写入到文件或者标准输出。等效于直接使用 vb_dump 导出为纯文本格式。

命令格式

```
vb_restore [OPTION]... FILE
```

📖 说明

- FILE 没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件，需输入 dbname 或-l 选项。不允许用户同时输入 dbname 和-l 选项。
- vb_restore 默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议使用"-c" 参数，在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。
- 日志打印无开关，若需隐藏日志，请将日志重定向到日志文件。

参数说明

通用参数：

- ❖ -d, --dbname=NAME
连接数据库 dbname 并直接导入到该数据库中。
- ❖ -f, --file=FILENAME
指定生成脚本的输出文件，或使用-l 时列表的输出文件。
默认是标准输出。

📖 说明

-f 不能同-d 一起使用。

- ❖ -F, --format=c|d|t
指定归档格式。由于 vb_restore 会自动决定格式，因此不需要指定格式。
取值范围：
 - c/custom：该归档形式为 4.21-vb_dump 的自定义格式。
 - d/directory：该归档形式是一个目录归档形式。
 - t/tar：该归档形式是一个 tar 归档形式。
- ❖ -l, --list
列出归档形式内容。这一操作的输出可用作-L 选项的输入。注意如果像-n 或-t 的过滤选项与-l 使用，过滤选项将会限制列举的项目（即归档形式内容）。
- ❖ -v, --verbose
指定 verbose 模式。
- ❖ -V, --version
打印 vb_restore 版本，然后退出。
- ❖ -?, --help
显示 vb_restore 命令行参数帮助，然后退出。

导入参数：

- ❖ -a, --data-only
只导入数据，不导入模式（数据定义）。vb_restore 的导入是以追加方式进行的。
- ❖ -c, --clean
在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象

❖ **-C, --create**

导入到数据库之前请创建数据库。（选择该选项后，以-d 打头的数据库将被用作发布首个 CREATE DATABASE 命令。所有数据将被导入到出现在归档文件的数据库中。）

❖ **-e, --exit-on-error**

当发送 SQL 语句到数据库时如果出现错误，请退出。默认状态下会继续，且在导入后会显示一系列错误信息。

❖ **-I, --index=NAME**

只导入已列举的 index 的定义。允许导入多个 index。如果多次输入-I index 导入多个 index。

例如：

```
vb_restore -h host_name -p port_number -d postgres -I Index1 -I Index2 backup/MPPDB_backup.tar
```

在上面这个例子中，Index1 和 Index2 会被导入。

❖ **-j, --jobs=NUM**

运行 vb_restore 最耗时的部分（如加载数据、创建 index、或创建约束）使用并发任务。该选项能大幅缩短导入时间，即将一个大型数据库导入到某一多处理器的服务器上。

每个任务可能是一个进程或一个线程，这由操作系统决定；每个任务与服务器进行单独连接。

该选项的最优值取决于服务器的硬件设置、客户端、以及网络。还包括这些因素，如 CPU 核数量、硬盘设置。建议是从增加服务器上的 CPU 核数量入手，更大的值（服务器上 CPU 核数量）在很多情况下也能导致数据文件更快的被导入。当然，过高的值会由于超负荷反而导致性能降低。

该选项只支持自定义归档格式。输入文件必须是常规文件（不能是像 pipe 的文件）。如果是通过脚本文件，而非直接连接数据库服务器，该选项可忽略。而且，多任务不能与--single-transaction 选项一起使用。

❖ **-L, --use-list=FILENAME**

只导入列举在 list-file 中的那些归档形式元素，导入顺序以它们在文件中的顺序为准。注意如果像-n 或-t 的过滤选项与-L 使用，它们将会进一步限制导入的项目。

一般情况下，list-file 是通过编辑前面提到的某个-l 参数的输出创建的。文件行的位置可更改或直接删除行，也可使用分号（;）在行的开始注出。见下文的举例。

❖ **-n, --schema=NAME**

只导入已列举的模式中的对象。

该选项可与-t 选项一起用以导入某个指定的表。

多次输入-n schemaname 可以导入多个模式。

例如：

```
vb_restore -h host_name -p port_number -d postgres -n sch1 -n sch2 backup/MPPDB_backup.tar
```

在上面这个例子中，sch1 和 sch2 会被导入。

❖ -O, --no-owner

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，vb_restore 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的模式元素的所属。除非是由系统管理员（或是拥有脚本中所有对象的同一个用户）进行数据库首次连接的操作，否则语句会失败。使用-O 选项，任何用户名都可用于首次连接，且该用户拥有所有已创建的对象。

❖ -P, --function=NAME(args)

只导入已列举的函数。请按照函数所在转储文件中的目录，准确拼写函数名称和参数。

当-P 单独使用时，表示导入文件中所有'function-name(args)'函数；当-P 同-n 一起使用时，表示导入指定模式下的'function-name(args)'函数；多次输入-P，而仅指定一次-n，表示所有导入的函数默认都是位于-n 模式下的。

可以多次输入-n schema-name -P 'function-name(args)'同时导入多个指定模式下的函数。

例如：

```
./vb_restore -h host_name -p port_number -d postgres -n test1 -P 'Func1(integer)' -n test2 -P 'Func2(integer)' backup/MPPDB_backup.tar
```

在上面这个例子中，test1 模式下的函数 Func1(i integer)和 test2 模式下的函数 Func2(j integer)会被一起导入。

❖ -s, --schema-only

只导入模式（数据定义），不导入数据（表内容）。当前的序列值也不会导入。

❖ -S, --sysadmin=NAME

该参数为扩展预留接口，不建议使用。

❖ -t, --table=NAME

只导入已列举的表定义、数据或定义和数据。该选项与-n 选项同时使用时，用来指定某个模式下的表对象。-n 参数不输入时，默认为 PUBLIC 模式。多次输入-n <schemaname> -t <tablename> 可以导入指定模式下的多个表。

例如：

导入 PUBLIC 模式下的 table1

```
vb_restore -h host_name -p port_number -d postgres -t table1 backup/MPPDB_backup.tar
```

导入 test1 模式下的 test1 和 test2 模式下的 test2

```
vb_restore -h host_name -p port_number -d postgres -n test1 -t test1 -n test2 -t test2 backup/MPPDB_backup.tar
```

导入 PUBLIC 模式下的 table1 和 test1 模式下的 test1

```
vb_restore -h host_name -p port_number -d postgres -n PUBLIC -t table1 -n test1 -t table1 backup/MPPDB_backup.tar
```

须知

-t 不支持 schema_name.table_name 的输入格式。

- ❖ -T, --trigger=NAME
该参数为扩展预留接口。
- ❖ -x, --no-privileges/--no-acl
防止导入访问权限（grant/revoke 命令）。
- ❖ -1, --single-transaction
执行导入作为一个单独事务（即把命令包围在 BEGIN/COMMIT 中）。
该选项确保要么所有命令成功完成，要么没有改变应用。该选项意为--exit-on-error。
- ❖ --disable-triggers
该参数为扩展预留接口，不建议使用。
- ❖ --no-data-for-failed-tables
默认状态下，即使创建表的命令失败（如表已经存在），表数据仍会被导入。使用该选项，像这种表的数据会被跳过。如果目标数据库已包含想要的表内容，这种行为会有帮助。
该选项只有在直接导入到某数据库中时有效，不针对生成 SQL 脚本文件输出。
- ❖ --no-security-labels
该参数为扩展预留接口，不建议使用。
- ❖ --no-tablespaces
不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在导入过程中所有对象都会被创建。
- ❖ --section=SECTION
导入已列举的区段（如 pre-data、data、或 post-data）。
- ❖ --use-set-session-authorization
该选项用来进行文本格式的备份。

输出 SET SESSION AUTHORIZATION 命令，而非 ALTER OWNER 命令，用以决定对象归属。该选项使转储更加兼容标准，但通过参考转储中对象的记录，导入过程可能会有问题。使用 SET SESSION AUTHORIZATION 的转储要求必须是系统管理员，同时在导入前还需参考"SET SESSION AUTHORIZATION"，手工对导出文件的密码进行修改验证，只有这样才能进行正确的导入操作，相比之下，ALTER OWNER 对权限要求较低。
- ❖ --with-key=KEY
AES128 密钥长度必须是 16 字节。

📖 说明

如果转储被加密，则必须在 vb_restore 命令中输入--with-key <keyname>选项。如果未输入，用户会收到错误信息。应该输入转储时所输入的相同的 key。

转储格式为 c 或 t 时，转储内容已被处理，因此转储格式为 c 或 t 是输入不受加密限制。

须知

- 如果安装过程中有任何本地数据要添加到 template1 数据库，请谨慎将 vb_restore 的输出载入到一个真正的空数据库中；否则可能会因为被添加对象的定义被复制，而出现错误。要创建一个无本地添加的空数据库，需从 template0 而非 template1 复制，例如：

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

- vb_restore 不能选择性地导入大对象；例如只能导入那些指定表的对象。如果某个归档形式包含大对象，那所有大对象都会被导入，或一个都不会被导入，如果它们通过 -L、-t 或其他选项被排除。

说明

1. -d/--dbname 和 -f/--file 不能同时使用；
2. -s/--schema-only 和 -a/--data-only 不能同时使用；
3. -c/--clean 和 -a/--data-only 不能同时使用；
4. 使用 --single-transaction 时，-j/--jobs 必须为单任务；
5. --role 和 --rolepassword 必须一起使用。

连接参数：

❖ -h, --host=HOSTNAME

指定的主机名称。如果取值是以斜线开头，他将用作 Unix 域套接字的目录。默认值取自 PGHOST 环境变量；如果没有设置，将启动某个 Unix 域套接字建立连接。

该参数只针对 Vastbase 外，对 Vastbase 内本机只能用 127.0.0.1。

❖ -p, --port=PORT

指定服务器所监听的 TCP 端口或本地 Unix 域套接字后缀，以确保连接。默认值设置为 PGPORT 环境变量。

在开启线程池情况下，建议使用 pooler port，即监听端口+1。

❖ -U, --username=NAME

所连接的用户名。

❖ -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是 trust，则不会对系统管理员进行密码验证，即无需输入 -W 参数；如果没有 -W 参数，并且不是系统管理员，“vb_restore”会提示用户输入密码。

❖ --role=ROLENAME

指定导入操作使用的角色名。选择该参数，会使 vb_restore 连接数据库后，发起一个 SET ROLE 角色名命令。当所授权用户（由 -U 指定）没有 vb_restore 要求的权限时，该参数会起到作用，

即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录，而使用该参数能够在不违反该规定的情况下完成导入。

❖ --rolepassword=ROLEPASSWORD

指定具体角色用户的角色密码。

示例

特例：执行 vsql 程序，使用如下选项导入由 vb_dump/vb_dumpall 生成导出文件夹（纯文本格式）的 MPPDB_backup.sql 文件到 postgres 数据库。

```
vsq1 -d postgres -p 5432 -W Bigdata@123 -f /home/omm/test/MPPDB_backup.sql
SET
SET
SET
SET
SET
SET
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
SET
CREATE INDEX
REVOKE
REVOKE
GRANT
GRANT
total time: 30476 ms
```

vb_restore 用来导入由 vb_dump 生成的导出文件。

示例 1：执行 vb_restore，将导出的 MPPDB_backup.dmp 文件（自定义归档格式）导入到 postgres 数据库。

```
vb_restore -W Bigdata@123 backup/MPPDB_backup.dmp -p 5432 -d postgres
vb_restore: restore operation successful
vb_restore: total time: 13053 ms
```

示例 2：执行 vb_restore，将导出的 MPPDB_backup.tar 文件（tar 格式）导入到 postgres 数据库。

```
vb_restore backup/MPPDB_backup.tar -p 5432 -d postgres
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21203 ms
```

示例 3：执行 vb_restore，将导出的 MPPDB_backup 文件（目录格式）导入到 postgres 数据库。

```
vb_restore backup/MPPDB_backup -p 5432 -d postgres
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21003 ms
```

示例 4：执行 vb_restore，使用自定义归档格式的 MPPDB_backup.dmp 文件来进行如下导入操作。导入 PUBLIC 模式下所有对象的定义和数据。在导入时会先删除已经存在的对象，如果原对象存在跨模式的依赖则需手工强制干预。

```

vb_restore backup/MPPDB_backup.dmp -p 5432 -d postgres -e -c -n PUBLIC
vb_restore: [archiver (db)] Error while PROCESSING TOC:
vb_restore: [archiver (db)] Error from TOC entry 313; 1259 337399 TABLE table1 Vastbasea
vb_restore: [archiver (db)] could not execute query: ERROR: cannot drop table table1 because other
objects depend on it
DETAIL: view t1.v1 depends on table table1
HINT: Use DROP ... CASCADE to drop the dependent objects too.
Command was: DROP TABLE public.table1;

```

手工删除依赖，导入完成后再重新创建。

```

vb_restore backup/MPPDB_backup.dmp -p 5432 -d postgres -e -c -n PUBLIC
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 2203 ms

```

示例 5：执行 vb_restore，使用自定义归档格式的 MPPDB_backup.dmp 文件来进行如下导入操作。
只导入 PUBLIC 模式下表 table1 的定义。

```

vb_restore backup/MPPDB_backup.dmp -p 5432 -d postgres -e -c -s -n PUBLIC -t table1
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 21000 ms

```

示例 6：执行 vb_restore，使用自定义归档格式的 MPPDB_backup.dmp 文件来进行如下导入操作。
只导入 PUBLIC 模式下表 table1 的数据。

```

vb_restore backup/MPPDB_backup.dmp -p 5432 -d postgres -e -a -n PUBLIC -t table1
vb_restore[2017-07-21 19:16:26]: restore operation successful
vb_restore[2017-07-21 19:16:26]: total time: 20203 ms

```

相关命令

3.1 vb_dump, 3.2 vb_dumpall

4. 系统内部使用的工具

本节所涉及的工具为 Vastbase 进程间运行使用的工具，或者说模块间调用的工具。这些工具只用于满足内部调用场景，未对其他场景下的使用正确性做过验证。因此，不建议用户直接使用这些工具执行业务，以防对系统运行带来影响。

[4.1 Vastbase](#)

[4.2 gs_backup](#)

[4.3 vb_basebackup](#)

[4.4 vb_ctl](#)

[4.5 vb_initdb](#)

[4.6 gs_install](#)

[4.7 gs_postuninstall](#)

[4.8 gs_preinstall](#)

[4.9 gs_sshexkey](#)

[4.10 gs_uninstall](#)

[4.11 gs_upgradectl](#)

[4.12 vbtrace](#)

[4.13 kdb5_util](#)

[4.14 kadmin.local](#)

[4.15 kinit](#)

[4.16 klist](#)

[4.17 krb5kdc](#)

[4.18 kdestroy](#)

[4.19 vb_config](#)

[4.20 vb_controldata](#)

[4.21 vb_resetxlog](#)

4.1.vastbase

背景信息

vastbase 是 Vastbase 数据库的主进程，也是一个可执行的命令，能够通过其启动一个数据库进程。

语法

```
Vastbase [OPTION]...
```

描述

客户端应用程序为了访问数据库，将连接（通过网络或本地）到一个正在运行的 vastbase 进程。然后该进程实例会启动一个独立的线程来处理这个连接。

须知

通过 vastbase 启动数据库时，需要再开一个窗口来连接到数据库，也可以使用&符号使程序在后台执行。

一个 `vastbase` 进程总是管理来自同一个数据库的数据。一个系统上可以同时运行多个 `Vastbase` 进程，只要使用不同的数据目录和不同的端口号。`vastbase` 启动时需要知道数据目录的位置，该位置必须通过 `-D` 指定。通常，`-D` 直接指向由 `vb_initdb` 创建的数据库目录。

缺省时 `vastbase` 在前台启动并将日志信息输出到标准错误。但在实际应用中，`vastbase` 应当作为后台进程启动，而且多数是在系统启动时自动启动。

`vastbase` 还能以单用户模式运行，这种用法主要用于 `vb_initdb` 的初始化过程中，有时候也被用于调试灾难性恢复。不过，单用户模式运行的服务器并不适合于调试，因为没有实际的进程间通讯和锁动作发生。当从 `shell` 上以单用户模式调用时，用户可以输入查询，然后结果会在屏幕上以一种更适合开发者阅读（不适合普通用户）的格式显示出来。在单用户模式下，将把会话用户 ID 设为 1 并赋予系统管理员权限。该用户不必实际存在，因此单用户模式运行的服务器可以用于对某些意外损坏的系统表进行手工恢复。

选项

`vastbase` 接受表 1-34 中所示的命令行参数。

表 1-26 参数说明

参数	参数说明	取值范围
-A 1 0	运行时断言检查,是检测编程错误的调试帮助。	-
-B NBUFFERS	设置服务器进程使用的共享内存缓冲区的数量。	-
-b BINARY UPGRADES	binary upgrade 标志。	-
-c NAME=VALUE	给一个正在运行的参数命名。-c 可以出现多次从而设置多个参数。	-
-C NAME	打印一个命名的运行时的参数的值然后退出。可以用在正在运行的服务器上，从 <code>postgresql.conf</code> 返回值。	取值为 <code>postgresql.conf</code> 中的所有参数。
-d 1-5	设置调试级别, 1-5 记录对应 debug 级别信息。级别越高，写到服务器日志的调试输出越多。	取值范围：1-5。 说明 如果取值小于等于 0，则只会记录 notice 级别的信息。
-D DATADIR	声明数据目录或者配置文件的文件系统路径。	用户自定义。
-e	把缺省日期风格设置为"European", 也就是说用 DMY 规则解释日期输入，并且在一些日期输出格式里日在月份前面打印。	-

参数	参数说明	取值范围
-F	关闭 fsync 调用以提高性能，但是要冒系统崩溃时数据毁坏的风险。声明这个选项等效关闭了 fsync 参数。	-
-h HOSTNAME	指定 Vastbase 侦听来自前端应用 TCP/IP 连接的主机名或 IP 地址。	前端存在的主机名或 IP 地址。
-i	该选项允许远程客户通过 TCP/IP (网际域套接字) 与服务器通讯。	-
-k DIRECTORY	指定 Vastbase 侦听来自前端应用连接的 Unix 域套接字的目录。	缺省通常是/tmp，但是可以在编译的时候修改。
-l	该选项允许远程客户通过 SSL (安全套接层) 与服务器通讯。	-
-N MAX-CONNECT	设置服务器接受的客户端连接的最大数。缺省时由 vb_initdb 自动选择。声明这个选项等价于声明 max_connections 配置参数。	取值范围：正整数。
-M SERVERMODE	在启动时指定数据库的启动模式。	SERVERMODE 可以取下面三个值： •primary：本端以主机模式启动。 •standby：本端以备机模式启动。 •pending：本端处于等待状态，等待提升为主机或者备机。
-o OPTIONS	向每个服务器进程传递 "OPTIONS"。保留语法，已没有实际功能。	-
-p PORT	指定 Vastbase 侦听客户端连接的 TCP/IP 端口或本地 Unix domain socket 文件的扩展。默认端口号为 5432。	正整数，在操作系统支持的端口范围内。
-s	在每条命令结束时打印时间信息和其他统计信息。	-

参数	参数说明	取值范围
-S WORK-MEM	声明内部排序和散列在求助于临时磁盘文件之前可以使用的内存数量。	单位为 KB。
--single_node	拉起单机数据库。	-
-V, --version	打印 Vastbase 的版本信息然后退出。	-
--NAME=VALUE	给一个正在运行的参数命名。	-
--describe-config	描述配置参数然后退出。	-
-, --help	显示关于 Vastbase 命令行参数的帮助信息，然后退出。	-

表 1-35 中的参数主要是便于开发人员调试使用，有时也用于帮助恢复严重损坏的数据库。在应用程序使用数据库提供服务时，请不要使用这些参数进行调试。

表 1-27 开发者选项

参数	参数说明	取值范围
-f s i n m h	禁止某种扫描和连接方法的使用。	取值范围： •s: 关闭顺序 •i: 索引扫描 •n: 关闭嵌套循环 •m: 融合 (merge) 连接 •h: Hash 连接
-n	主要用于调试导致服务器进程异常崩溃的问题。一般策略是通知所有其他服务器进程终止并重新初始化共享内存和信号灯。该选项指定 Vastbase 不重新初始化共享内存。	-
-O	允许修改系统表的内部。	-
-P	读系统表时忽略系统索引，但在修改表时仍然更新索引。	-
-t pa plex	打印与每个主要系统模块相关的查询记时统计。	-
-T	主要用于调试导致服务器进程异常崩溃的问题。该选项指定 Vastbase 通过发送 SIGSTOP 信号停止其他所有服务器进程，但是并不让它们退出。这样就允许系统程序员手动从所有服务器	-

参数	参数说明	取值范围
	进程搜集内核转储。	
-W NUM	指定一个新的服务器进程开始需要等待的秒数。	单位：秒
--localxid	使用本地事务 ID，而不是全局事务 ID。 须知 此选项仅用于 vb_initdb。使用此选项可能会导致数据库不一致。	已存在的本地事务 ID

下面的选项表 1-36 仅在单用户模式下可用。

表 1-28 单用户模式选项

参数	参数说明	取值范围
--single	启动单用户模式。必须是命令行中的第一个选项。	-
DBNAME	要访问的数据库的名称。	字符串。默认为用户名。
-d 0-5	重新指定调试级别。	0-5
-E	回显所有命令。	-
-j	禁止使用新行作为语句分隔符。	-
-r FILENAME	将所有服务器标准输出和标准错误保存到文件 filename 中。	-

表 1-29 自启动模式选项

参数	参数说明	取值范围
--boot	启动自启动模式。必须是命令行中的第一个选项。该参数是对数据库中的参数变量及相关配置初始化，常用在数据库安装的流程中，直接使用该参数，无明显行为感知。	-
-r FILENAME	将所有服务器标准输出和标准错误保存到文件 filename 中。	-
-x NUM	指定一个新的服务器线程的类型。在初始化数据库时，会用到自启动模式，通过设置这个参数，启动不同线程来执行一些逻辑，正常情况下，不会用到，因为自启动模式很少会用到。	-

表 1-30 升级模式选项

参数	参数说明	取值范围
-u NUM	指定升级前数据库内核版本号。	-

除以上参数外，出现在 postgres.conf 中的所有参数，均可以用--guc_name=guc_value 的方式在 Vastbase 启动时传给 Vastbase。例如，--enable_fast_query_shipping=off

环境变量

- ❖ PGLINETENCODING：客户端使用的缺省字符编码。客户端可以独立地覆盖它。这个值也可以在配置文件里设置。
- ❖ PGDATESTYLE：运行时参数 DateStyle 的缺省值。现在反对使用该环境变量。
- ❖ TZ：服务器的时区。

错误处理

一个提到了 semget 或 shmget 的错误信息可能意味着需要重新配置内核，提供足够的共享内存和信号灯。可以通过降低 shared_buffers 值以减少 Vastbase 的共享内存的消耗，或者降低 max_connections 值减少 Vastbase 的信号灯的消耗。

如果发现类似“另外一个服务器正在运行”的错误信息，可以根据系统使用不同的命令：

```
ps ax | grep Vastbase
```

或

```
ps -ef | grep Vastbase
```

如果确信没有冲突的服务器正在运行，可以删除消息里提到的锁文件然后再次运行。

无法绑定端口的错误信息可能表明该端口已经被其他非 Vastbase 进程使用。如果终止 Vastbase 后又马上用同一端口号运行它，也可能得到错误信息。这时，必须多等几秒，等操作系统关闭了该端口再试。最后，如果使用了一个操作系统认为是保留的端口，也可能导致这个错误信息。例如：Unix 版本认为低于 1024 的端口号是“可信任的”，因而只有 Unix 系统管理员可以使用它们。

须知

- 如果有可能，不要使用 SIGKILL 杀死主进程。这样会阻止 Vastbase 在退出前释放它持有的系统资源（例如共享内存和信号灯）。这样可能会影响到将来启动新的进程。
- 可以使用 SIGTERM, SIGINT, SIGQUIT 信号正常结束服务器进程。第一个信号将等待所有的客户端退出后才退出。第二个将强制断开所有客户端，而第三个将不停止立刻退出，导致在重启时的恢复运行。
- 信号 SIGHUP 将会重新加载服务器配置文件。它也可能给单个服务器进程发送 SIGHUP 信号，但是这通常是不明显的。
- 要取消一个正在执行的查询，向正在运行的进程发送 SIGINT 信号。
- 主服务器进程向子进程发送 SIGTERM 信号让它们正常退出；发送 SIGQUIT 信号立即退出且不做清理工作。用户有三种信号不能用。同时，发送 SIGKILL 信号也是不明智的：主进程将把这个信号当作崩溃信号，然后会强制其他兄弟进程作为标准的崩溃回复过程退出。

用法

启动一个单用户模式的服务器：

```
Vastbase --single -D /usr/local/pvsql/data other-options my_database
```

用-D 给服务器提供正确的数据库目录的路径。同时还要声明已存在的特定数据库名称。

须知

- 通常，独立运行的服务器把换行符当做命令输入完成字符；要想把一行分成多行写，必需在除最后一个换行符以外的每个换行符前面敲一个反斜杠。
- 如果使用了-j 命令行选项，新行将不被当作命令结束符。此时服务器将从标准输入一直读取到 EOF 标志为止，然后把所有读到的内容当作一个完整的命令字符串看待，并且反斜杠与换行符也被当作普通字符来看待。
- 输入 EOF (Control+D) 即可退出会话。如果已经使用了-j 则必须连续使用两个 EOF 才行。
- 单用户模式运行的服务器不会提供复杂的行编辑功能（比如没有命令历史）。单用户模式也不会做任何后台处理，像自动检查点。

示例

1. 用缺省值在后台启动 Vastbase：

```
nohup Vastbase >logfile 2>&1 </dev/null &
```

2. 在指定端口启动 Vastbase，如 1234：

```
Vastbase --singlenode -p -D data_dir
```

3. 使用 vsql 与服务器连接，用-p 指定端口号：

```
vsq1 -d postgres -p 1234
```

4.2. vb_basebackup

背景信息

Vastbase 部署成功后，在数据库运行的过程中，会遇到各种问题及异常状态。Vastbase 提供了 vb_basebackup 工具做基础的物理备份。vb_basebackup 的实现目标是对服务器数据库文件的二进制进行拷贝，其实现原理使用了复制协议。远程执行 vb_basebackup 时，需要使用系统管理员账户。

vb_basebackup 当前仅支持热备份模式，不支持压缩格式备份。

说明

- vb_basebackup 仅支持全量备份，不支持增量。
- vb_basebackup 当前仅支持热备份模式，不支持压缩格式备份。
- vb_basebackup 在备份包含绝对路径的表空间时，不能在同一台机器上进行备份。对于同一台机器，绝对路径是唯一的，因此会产生冲突。可以在不同的机器上备份含绝对路径的表空间。
- 若打开增量检测点功能且打开双写，vb_basebackup 也会备份双写文件

前提条件

- ❖ 可以正常连接 Vastbase 数据库,pg_hba.conf 中需要配置允许复制链接, 需要配置 max_wal_senders 的数量, 至少有一个可用。
- ❖ 在进行还原时, 需要保证各节点备份目录中存在备份文件, 若备份文件丢失, 则需要从其他节点进行拷贝。

语法

- ❖ 显示帮助信息

```
vb_basebackup -? | --help
```

- ❖ 显示版本号信息

```
vb_basebackup -V | --version
```

参数说明

vb_basebackup 参数可以分为如下几类:

- ❖ -D directory

备份文件输出的目录, 必选项。

- ❖ 常用参数:

- -c, --checkpoint=fast|spread

设置检查点模式为 fast 或者 spread(默认)

- -l, --label=LABEL

为备份设置标签

- -P, --progress

启用进展报告

- -v, --verbose

启用冗长模式

- -V, --version

打印版本后退出

- -?, --help

显示 vb_basebackup 命令行参数

- -F, --format=plain|tar

指定输出格式

- Z
将输出的 tar 格式文件进一步压缩为.gz 格式
- T, 需备份的表空间路径=表空间备份输出路径
指定表空间目录映射

❖ 连接参数

- h, --host=HOSTNAME
指定正在运行服务器的主机名或者 Unix 域套接字的路径。
- p, --port=PORT
指定数据库服务器的端口号。
可以通过 port 参数修改默认端口号。
- U, --username=USERNAME
指定连接数据库的用户。
- s, --status-interval=INTERVAL
发送到服务器的状态包的时间(以秒为单位)
- w, --no-password
不出现输入密码提示。
- W, --password
当使用-U 参数连接本地数据库或者连接远端数据库时, 可通过指定该选项出现输入密码提示。

示例

```
vb_basebackup -D /home/test/trunk/install/data/backup -h 127.0.0.1 -p 21233 -F plain
INFO: The starting position of the xlog copy of the full build is: 0/1B800000. The slot minimum LSN
is: 0/1B800000.
```

从备份文件恢复数据

当数据库发生故障需要从备份文件进行恢复。因为 vb_basebackup 是对数据库按二进制进行备份, 因此恢复时可以直接拷贝替换原有的文件, 或者直接在备份的库上启动数据库。

📖 说明

- 若当前数据库实例正在运行, 直接从备份文件启动数据库可能会存在端口冲突, 这是需要修配置文件的 port 参数, 或者在启动数据库时指定一下端口。

- 若当前备份文件为主备数据库，可能需要修改一下主备之间的复制连接。即配置文件中的 `postgre.conf` 中的 `replconninfo1`，`replconninfo2` 等

若要在原库的地方恢复数据库，参考步骤如下：

1. 停止数据库服务器，具体操作请参见<管理员指南>
2. 将原数据库库和所有表空间复制到另外一个位置，以备后面需要
3. 清理原库中的所有或部分文件
4. 使用数据库系统用户权限从备份中还原需要的数据库文件
5. 若数据库中存在链接文件，需要修改使其连接到正确的文件。
6. 重启数据库服务器，并检查数据库内容，确保数据库已经恢复到所需的状态。

📖 说明

- 暂不支持备份文件增量恢复
- 恢复后需要检查数据库中的连接文件是否连接到正确的文件

4.2.1.PITR 支持恢复到指定 LSN

功能描述

支持 Oracle 的 DBMS_UTILITY 包使用，提供了各种实用子程序。

示例

```
1.数据库开启归档模式
修改配置文件 postgresql.conf，并重启数据库
archive_mode = on
archive_command = 'cp %p /home/vastbase_1st/data/vastbase/arch/%f'
mkdir/home/vastbase_1st/data/vastbase/arch/
vb_ctl restart
2.进行备份
mkdir/home/vastbase_1st/data/vastbase/bak
mkdir /home/vastbase_1st/data/tbs_test1_bak
vb_basebackup -D /home/vastbase_1st/data/vastbase/bak -h 127.0.0.1 -p 6012 -Fplain -P -v
-T/home/vastbase_1st/data/tbs_test1=/home/vastbase_1st/data/tbs_test1_bak
3.进入数据库，创建表空间跟表
create tablespace tbs_test1 location '/home/vastbase_1st/data/tbs_test1';
create table t_tbs_backup1(id int)TABLESPACE tbs_test1;
insert into t_tbs_backup1 values(1);
insert into t_tbs_backup1 values(2);
insert into t_tbs_backup1 values(3);
insert into t_tbs_backup1 values(4);
insert into t_tbs_backup1 values(5);
insert into t_tbs_backup1 values(6);
insert into t_tbs_backup1 values(7);
insert into t_tbs_backup1 values(8);
insert into t_tbs_backup1 values(9);
insert into t_tbs_backup1 values(10);
4.查看当前 LSN 号
select pg_current_xlog_location();
```

```

5. 继续插入数据
insert into t_tbs_backup1 values(11);
insert into t_tbs_backup1 values(12);
insert into t_tbs_backup1 values(13);
insert into t_tbs_backup1 values(14);
insert into t_tbs_backup1 values(15);
insert into t_tbs_backup1 values(16);
insert into t_tbs_backup1 values(17);
insert into t_tbs_backup1 values(18);
insert into t_tbs_backup1 values(19);
insert into t_tbs_backup1 values(20);
6. 进入到备份目录/home/vastbase_1st/data/vastbase/bak, 创建 recovery.conf 文件并写入
restore_command = 'cp /home/vastbase_1st/data/vastbase/arch/%f %p'
recovery_target_lsn='$LSN'
(填写第 4 步查看到的 LSN 号)
7. 关闭原库
vb_ctl stop
8. 启动数据库, 完成恢复
vb_ctl start -D /home/vastbase_1st/data/vastbase/bak
9. 进入数据库, 查询数据
select * from t_tbs_backup1;

```

4.3. vb_ctl

功能介绍

vb_ctl 是 Vastbase 提供的数据库服务控制工具, 可以用来启停数据库服务和查询数据库状态。主要供 Vastbase 管理模块调用。

vb_ctl 工具由操作系统用户 omm 执行。

- ❖ 启动、停止、重启 Vastbase 节点。
- ❖ 在不停止数据库的情况下, 重新加载配置文件 (postgresql.conf, pg_hba.conf)。
- ❖ 主备切换、主备状态查询、重建和重建状态查询。

参数说明

vb_ctl 参数可分为如下几类:

- ❖ option 参数, 详细请参见表 1-39。
- ❖ 公共参数, 详细请参见表 1-40。
- ❖ start 和 restart 模式的参数, 详细请参见表 1-41。
- ❖ stop 和 restart 模式的参数, 详细请参见表 1-42。
- ❖ switchover 模式的参数, 详细请参见表 1-43。
- ❖ build 参数, 详细请参见表 1-44。
- ❖ restore 参数, 详细请参见表 1-45。

表 1-31 option 参数

参数	参数说明
init[db]	初始化数据库。
start	启动数据库。
restart	重启数据库。
build	在主机上重建备实例。
stop	停止数据库。
reload	重载配置文件 (postgresql.conf, pg_hba.conf) 。
status	显示数据库运行状态。
promote	在指定的数据目录运行的备用服务器被命令退出故障恢复，并开始进行读写操作。
failover	在主机异常时，将备机切换为主机。 切换成功后，需要执行 <code>gs_om -t refreshconf</code> 命令记录当前主备机信息。
switchover	在主备机正常时，出于维护的需要，将备机切换为主机，可保证切换过程中数据不丢失。 切换成功后，需要执行 <code>gs_om -t refreshconf</code> 命令记录当前主备机信息。 switchover 命令下发后，命令如果超时返回，后台进程的执行状态可能处于不可确定状态；如果备机在 standby wait 状态，可以通过重复下发 switchover 命令消除，使集群恢复到正常状态。
query	查询主备机之间的状态信息。
notify	启动后再指定主备机。
kill	给指定的进程发送信号量。
querybuild	查看数据库的重建进度。
restore	恢复增量备份集，需要在 Vastbase 进程停止的情况下执行。

表 1-32 公共参数

参数	参数说明	取值范围
-D, --pgdata=DATADIR	指定数据目录的位置。	DATADIR 的取值必须为有效的数据目录。
-s, --silent	仅打印错误，没有参考消息。	-

参数	参数说明	取值范围
-t, --timeout=SECS	这个参数同时控制如下两个功能： •等待数据库启动或者关闭完成的最大秒数。	取值范围：整型（秒）。 默认值： •等待数据库启动或者停止完成：60 秒。
-V, --version	打印 vb_ctl 的版本信息然后退出。	-
-w	用户执行启动或者关闭数据库的操作后，会等待事务操作的完成后再执行。	默认值：省略此参数时，默认会等待事务操作完成。
-W	用户执行启动或者关闭数据库的操作后，会强制完成启动或者关闭，不等待事务操作完成。	-
-M	-M 后面需要跟 SERVERMODE 参数，表示在启动时指定数据库的启动模式。 说明 当前主机处于一主零备，或单机环境时，不支持-M 参数。	SERVERMODE 的取值范围： •primary：本端以主机模式启动。 •standby：本端以备机模式启动。 •pending：本端处于等待状态，等待提升为主机或者备机。
-T	-T 后面跟 term，升主命令时会用此 term 作为主机 term，build 时会连接大于等于此 term 的主机进行 build	term 的取值范围，无符号整型。
-P PASSWORD	与-U 参数配合使用，指定连接本地数据库的用户密码。 说明 当省略-U 参数，且认证方法为 trust 时，-P 参数无效。	-
-U USERNAME	指定连接数据库的用户。此参数只能与 notify、query 和 querybuild 参数配合使用。 对于域用户，使用 DOMAIN\username 格式。	取值范围：Vastbase 中存在的用户。 默认值：省略此参数则使用与当前操作系统用户同名的用户。

参数	参数说明	取值范围
-?, --help	显示关于 vb_ctl 命令行参数的帮助信息。	-

表 1-33 start 或 restart 共有参数

参数	参数说明	取值范围
-c, --core-file	允许 Vastbase 进程产生内核文件。	-
-l, --log=FILENAME	把服务器日志输出附加在 FILENAME 文件上。	FILENAME 的取值为启动数据库服务的用户有权限的文件。 例如，data 目录下的某文件。
-o OPTIONS	声明要直接传递给由 Vastbase 执行的 Vastbase 的命令行选项。 参数通常都用单或者双引号包围以保证它们作为一个整体传递。	4.1 Vastbase 支持的参数。
-p PATH-TO-POSTGRES	声明 Vastbase 可执行文件的位置。	缺省位于 vb_ctl 自身所在目录，一般不需要此选项。

表 1-34 stop 和 restart 共有参数

参数	参数说明	取值范围
-m, --mode=MODE	声明关闭模式。	mode 的取值： •smart：单机模式下主机立即断开，同 fast。 •fast：不等待客户端中断连接，所有活跃事务都被回滚并且客户端都被强制断开，然后服务器将被关闭。 •immediate：强行关闭，在下次重新启动的时候将导致故障恢复。 默认值：fast

表 1-35 switchover 参数

参数	参数说明	取值范围
-m SWITCHOVER-MODE	声明双机主备实例正常时切换的切换模式。	mode 的取值： •fast：不等待客户端中断连接，所有活跃事务都被回滚并且客户端都被强制断开，然后服务器将被切换。 •smart：本版本暂不支持该模式 •immediate：本版本暂不支持该模式 默认值：fast

表 1-36 build 参数

参数	参数说明	取值范围
-b MODE	指定重建备机的模式。	mode 的取值： • full：通过全量镜像的方式重新同步 主机的数据目录。 • incremental：通过解析 Xlog 日志获 取主备机差异的数据进行增量修复备机。 警告 • 增量重建适用于主备双主等因日志造成的不一致场景。 • 备机数据文件损坏、数据目录丢失 等故障通过增量重建的方式无法修复，此时可通过全量重建的方式重 新修复备机。 • auto(不指定)：先增量，根据失败 后是否可以再增量选择继续增量或 者全量，三次增量失败后进行全 量。 默认值：auto
-r, --recvtimeout=INTERVAL	指定重建过程中备机日志接收等待主机响应的超时时间。	取值范围：整型（秒）。 默认值：120 秒
-q	重建结束后，是否自动重启。 指定参数场合，不自动重启。 未指定场合，自动重启。	无参数

表 1-37 restore 参数

参数	参数说明	取值范围
--remove-backup	指定是否删除增量备份集。	无参数

4.4. vb_initdb

4.4.1. vb_initdb 简介

背景信息

vb_initdb 初始化数据库时，会创建数据库目录、生成系统表、创建默认数据库和模板数据库。

系统表

初始化数据库时会生成大量的系统表和视图，其中绝大部分都对任何数据库用户开放查看权限。

📖 说明

pg_user_status、pg_auth_history 系统表权限只对初始化数据库用户和 sysadmin 用户开放。

生成的数据库

- ❖ template1：是一个模板数据库，当以后再创建一个新的数据库时，template1 数据库里的所有内容都会拷贝到新数据库中。通过 vb_initdb 的参数可以决定 template1 数据库的设置。
- ❖ template0：是 Vastbase 提供的最初级的备份数据库，当需要时可用 template0 作为模板生成“干净”的数据库。
- ❖ postgres：是一个提供给用户、工具和第三方应用的缺省数据库。

4.4.2. 使用指导

背景信息

在安装时，推荐使用-D 参数调用 vb_initdb 初始化数据库。如果由于故障恢复等原因，需要重新初始化一个数据库，可以通过执行 vb_initdb 来完成。

- ❖ 尽管 vb_initdb 会尝试创建相应的数据目录，但可能没有权限执行此操作，因为要创建目录的父目录通常被 root 所拥有。如果要创建数据目录，首先用 root 用户创建一个空数据目录，然后用 chown 把该目录的所有权交给数据库用户。
- ❖ vb_initdb 决定 template1 数据库的设置，而该设置将会成为其他数据库的默认设置。

- ❖ vb_initdb 初始化数据库的缺省区域和字符集编码。字符集编码、字符编码排序 (LC_COLLATE) 和字符集类 (LC_CTYPE, 如大写、小写数字等) 可以在创建数据库时独立设置。

操作步骤

- 步骤 1 以操作系统用户 omm 登录数据库主节点。
- 步骤 2 规划数据库目录。

1. 切换到 root 用户。

```
su - root
```

2. 根据提示输入用户名和密码。

- 步骤 3 进入“/opt/Vastbase”目录，并创建“data1”目录。

```
cd /opt/Vastbase
mkdir data1
```

3. 将 data1 目录的所有权分配给数据库用户 omm，dbgrp 是 omm 所属的用户组。

```
chown omm:dbgrp data1
```

4. 退出 root 用户。

```
exit
```

- 步骤 4 执行 vb_initdb 命令初始化数据库。

```
vb_initdb /opt/Vastbase/data1 -w "Vastbase@123"
```

vb_initdb 的语法：

```
vb_initdb [OPTION]... [DATADIR]
```

OPTION 参数请参见 4.5.3 命令参考。

----结束

4.4.3. 命令参考

命令参考

vb_initdb 支持的常用参数和不常用参数请分别参见表 4-13 和表 4-14。

表 1-38 常用参数说明

参数	参数说明	取值范围
-A, --auth=METHOD	指定本地用户连接数据库时的认证方法，即“pg_hba.conf”配置文件中 host 和 local 所在行的认证方法。 除非用户对本地用户都是信任的，否则不要使用默认值 trust。	METHOD 的取值： • trust • reject • md5（不安全的算法，为了兼容老版本而存在） • sha256 默认值：trust

参数	参数说明	取值范围
	须知 若取值为 md5，则需手动修改参数文件 postgresql.conf.sample 中的密码存储类型 password_encryption_type 参数的值，修改为 0，且放开注释使之生效。vb_initdb 工具需同时配合 -W 的使用。	
--auth-host=METHOD	指定本地用户通过 TCP/IP 连接数据库时的认证方法，即：“pg_hba.conf”配置文件中 host 所在行的认证方法。 指定此参数则会覆盖-A 参数的值。	METHOD 的取值： • trust • reject • md5（不安全的算法，为了兼容老版本而存在） • sha256 默认值：trust
--auth-local=METHOD	指定本地用户通过 Unix 域套接字连接数据库时的认证方法，即“pg_hba.conf”配置文件中 local 所在行的认证方法。 指定此参数则会覆盖-A 参数的值。	METHOD 的取值： • trust • reject • md5（不安全的算法，为了兼容老版本而存在） • sha256 默认值：trust
[-D, --pgdata=]DATADIR	指定数据目录的位置。	DATADIR 的取值：用户自定义。
--nodename=NODENAME	初始化的节点名称。	-
-E, --encoding=ENCODING	为新数据库设置编码格式。	• 如果使用此参数，需要加上--locale 选项指定支持此编码格式的区域。如果不加--locale 选项，则采用系统默认的区域，如果系统默认区域的编码格式和用此参数指定的编码格式不匹配则会导致数据库初始化失败。 • 如果不指定此参数，则使用系统默

参数	参数说明	取值范围
		认区域的编码格式。系统默认区域和编码格式可以使用 locale 命令查看，如下： <pre>omm@linux:~> locale grep LC_CTYPE LC_CTYPE="en_US.UTF-8"</pre> 其中 UTF-8 表示系统默认区域的编码格式。
--locale=LOCALE	为新数据库设置缺省的区域。 须知 如果用户设置了数据库的编码格式，则用户选择区域的编码格式必须与用户设置的编码格式一致，否则数据库初始化会失败。	例如用户要将数据库编码格式初始化为 GBK，可以采用如下步骤： 1. 用 locale -a grep gbk 命令查看系统支持 gbk 编码的区域，如下： <pre>omm@linux:~> locale -a grep gbk zh_CN.gbk zh_SG.gbk</pre> 2. 初始化数据库时加入 --locale=zh_CN.gbk 选项。
--dbcompatibility=DBCMPATIBILITY	指定兼容的数据库的类型。	取值范围：A、B、C。分别表示兼容 O、MY 和 TD。
--lc-collate=LOCALE --lc-ctype=LOCALE --lc-messages=LOCALE --lc-monetary=LOCALE --lc-numeric=LOCALE --lc-time=LOCALE	为新数据库设置指定范畴的区域。	各参数的取值必须是操作系统支持的值。 说明 如果用户在数据库安装的时候没有指定 --lc-collate 参数，则--lc-collate 参数的默认值为 C。
--no-locale	和--locale=C 等价。	-
--pwfile=FILE	vb_initdb 时从文件 FILE 中读取数据库中系统管理员的密码。该文件的第一行将被当作密码使用。	FILE 可以是“相对路径+文件”的形式，也可以是“绝对路径+文件”的形式。相对路径是相对当前路径的。
-T, --text-search-config=CFG	设置缺省的文本搜索方式。此配置项的值不会做正确性校验，配置成功后，有日志记录提醒当前配置项的取值。	text-search-config 的取值： • pg_catalog.english 全文搜索 • pg_catalog.simple 普通文本搜索 默认值：pg_catalog.simple

参数	参数说明	取值范围
-U, --username=NAME	选择数据库系统管理员的用户名。	取值范围：正常的数据库用户。 默认值：运行 vb_initdb 的操作系统用户。
-W, --pwprompt	vb_initdb 时强制交互式输入数据库管理员的密码。	-
-w, --pwpasswd=PASSWD	vb_initdb 时通过命令行指定的管理员用户的密码，而不是交互式输入。	设置的密码要符合复杂度要求： • 最少包含 8 个字符； • 不能和用户名和当前密码（ALTER）相同，或和当前密码反序； • 至少包含大写字母（A-Z），小写字母（a-z），数字，非字母数字字符（限定为 ~!@#\$\$%^&*()-_+=+\\[{}];,:.<.>/?) 四类字符中的三类字符。
-C, --enpwdfidir=DIR	vb_initdb 时指定的经 AES128 加密算法加密过的密码文件所在目录。 vb_initdb 会将该目录下密码文件进行解密，同时把解密后的密码做密码复杂度校验，校验通过的话会将此密码作为用户的密码。 说明 • 加密密码文件需使用 vb_guc 工具生成： vb_guc encrypt -K Vastbase@123 -D Dir。 • 如果用户指定多个 -w 和 -C 参数，vb_initdb 会将用户输入的最后一个 -w 或 -C 参数作为用户的需求，即输入密码的明文或经过 AES128 加密后的密码。	-

参数	参数说明	取值范围
-X, --xlogdir=XLOGDIR	声明事务日志存储的目录。 所设置的目录，必须满足运行 Vastbase 的用户有读写权限。	只支持绝对路径。
-S, --security	安全方式初始化数据库。	以-S 方式初始化的数据库后，创建的数据库用户权限受到限制，默认不再具有 public schema 的使用权限。

表 1-39 不常用参数说明

参数	参数说明	取值范围
-d, --debug	从初始化后端打印调试输出信息。初始化后端是 vb_initdb 用于创建系统表的程序。	-
-L DIRECTORY	vb_initdb 初始化数据库所需要的输入文件的路径。通常是不必要的。如果需要明确声明的话，程序会提示输入。因该参数意为创建指定配置信息的数据库，建议将 share/postgresql 下的所有涉及启动相关的子目录及文件全部复制过来，避免其他因素的影响。	初始化数据库所需的输入文件路径。
-n, --noclean	不指定该参数时，当 vb_initdb 发现一些错误妨碍它完成创建数据库的工作时，它将在检测到不能结束工作之前将其创建的所有文件删除。这个选项禁止任何清理动作，因而对调试很有用。	-
-s, --show	显示内部设置。	-
-V, --version	打印 vb_initdb 版本信息，然后退出。	-
-?, --help	显示关于 vb_initdb 命令行参数的帮助信息，然后退出。	-

4.5. vbtrace

功能介绍

vbtrace 是 Vastbase 提供的用来跟踪内核代码执行路径，记录内核数据结构，分析代码性能的工具。Trace 的有限点位和数据在版本中被固化，无法动态添加和删除。

警告

1. 对内核 dump 指定内存变量的数据用于诊断分析，不存在直接指定任意寄存器或者内存地址的行为。读取的内存地址均是在开发阶段硬编码，没有任意地读取或者修改寄存器或内存的操作。
2. Trace 点可能涉及敏感数据，收集 trace 信息前需要同用户协商，授权和许可后方可收集。
3. Vastbase 不会在敏感信息上打点，不会 TRACE 和用户相关的数据。
4. Trace 仅用于诊断目的，开启 trace 将对性能产生一定影响，影响的大小视负载的高低，trace 的模块而不同。
5. Trace 工具的权限为 0700，仅限于数据库用户读、写和执行。

说明

如果进程异常终止，/dev/shm/ 目录下将会有 vbtrace_trace_cfg_* 残留，可以手动清除。

语法

```
vbtrace [start|stop|config|dump|detail|codepath|analyze] [-p <pid>] [-s <BUFFER_SIZE>] [-f <DUMP_FILE>] [-o <OUTPUT_FILE>] [-t <STEP_SIZE>]
```

参数说明

表 1-40 gs_trace 参数说明

参数	说明
start	开始记录 trace
stop	停止 trace, 释放存储信息的共享内存 trace buffer。注意: trace buffer 中已捕获 trace 信息会丢失。
config	显示 trace 的配置信息
dump	将共享内存中的 trace 信息写入指定文件。若没有启动 trace 将报错。
detail	将 dump 出来的二进制文件解析成文本文件，显示 trace 点的线程、时间信息。
codepath	提取 dump 文件中的函数调用信息，按照调用栈的方式显示。
analyze	统计各个函数的执行次数、总耗时、平均耗时、最长耗时、最短耗时
-p PID	指定启动 trace 功能的实例进程号
-f DUMP_FILE	指定 dump 导出的 trace 文件
-o OUTPUT_FILE	指定写入的文件
-t STEP_SIZE	指定分片分析的时间跨度（秒），可选。将生成单独的{OUTPUT_FILE}.step 文件。
-s BUFFER_SIZE	指定用于 trace 功能的共享内存大小，默认为 1G，该值需要为 2 的倍数，如果不是

参数	说明
	则会补齐为 2 的倍数。

示例

1. 启动 trace

```
vbtrace start -p 207787
```

2. 停止 trace

```
gstarce stop -p 207787
```

3. 查看 trace 配置

```
vbtrace config -p 207787
```

4. dump trace

```
vbtrace dump -p 207787 -o /data/207787.dump
```

5. 解析 dump detail 信息

```
vbtrace detail -f /data/207787.dump -o /data/207787.detail
```

6. 解析 dump codepath

```
vbtrace codepath -f /data/207787.dump -o /data/207787.codepath
```

7. 分析全局性能

```
vbtrace analyze -f /data/207787.dump -o /data/207787.perf
```

8. 分析分片性能

```
vbtrace analyze -f /data/207787.dump -o /data/207787.perf -t 1
```

4.6. kdb5_util

功能介绍

为 Vastbase 认证服务 Kerberos 提供数据库管理操作的工具，可以新建、销毁、导入和导出数据库。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.7. kadmin.local

功能介绍

为 Vastbase 认证服务 Kerberos 提供命令行接口，直接接入 kdc 服务的数据库，进行增加、删除、修改 Kerberos 用户等操作。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.8. kinit

功能介绍

为 Vastbase 认证服务 Kerberos 提供特定用户获取和缓存 TGT 能力。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.9. klist

功能介绍

为 Vastbase 认证服务 Kerberos 提供列出凭证缓存中的用户和票据能力。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.10. krb5kdc

功能介绍

为 Vastbase 认证服务 Kerberos 提供列认证和密钥管理服务。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.11. kdestroy

功能介绍

为 Vastbase 认证服务 Kerberos 提供销毁用户的授权票据操作。

参数说明

Kerberos 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/index.html

4.12. vb_config

功能介绍

通过 vb_config 工具打印的当前所安装数据库的配置参数信息，可用于找到与数据库相关的软件包所对应的头文件和库。

语法

```
vb_config [option...]
```

参数说明

vb_config 工具支持下面一个或多个选项，如果给出多于一个选项，那么信息以该顺序打印，每行一条。如果没有给出选项，则打印所有可获得的信息，连着标签一起打印。

- ❖ --bindir
打印用户可执行文件的路径。 比如，可以用这个找 psql 程序。 通常这也是 vb_config 程序存在的路径。
- ❖ --docdir
打印文档文件的位置。
- ❖ --htmldir
输出 HTML 文档文件的位置。
- ❖ --includedir
打印客户端接口 C 头文件的路径。
- ❖ --pkgincludedir
打印其它 C 头文件的位置。
- ❖ --includedir-server
打印那些做服务器编程时使用的 C 头文件的位置。
- ❖ --libdir
打印目标代码库的路径。
- ❖ --pkglibdir
打印动态加载的模块的位置，或者服务器在哪里寻找他们。 其它体系相关的数据文件可能也放在这个目录里。
- ❖ --localedir
打印区域支持文件的位置。如果在编译数据库的时候没有配置区域支持， 则其值为空字符串。
- ❖ --mandir
打印手册页的位置。
- ❖ --sharedir

打印体系无关的支持文件的位置。

❖ --sysconfdir

打印系统范围的配置文件位置。

❖ --pgxs

打印扩展 makefiles 文件的位置。

❖ --configure

打印配置和编译数据库时给 configure 脚本传递的选项，用于复制相同配置，或者找出二进制包是用哪些选项编译的。不过请注意二进制包通常包含供应商提供的补丁。

❖ --cc

打印用于编译数据库的 CC 变量的数值。这个选项显示使用的 C 编译器。

❖ --cppflags

打印用于编译数据库的 CPPFLAGS 变量的数值。这个选项显示在预处理的时候所需要的 C 编译器开关(通常是-I 开关)。

❖ --cflags

打印编译数据库时使用的 CFLAGS 变量的数值。这个选项显示 C 编译器开关。

❖ --cflags_sl

打印编译数据库时使用的 CFLAGS_SL 变量的数值。这个选项显示了编译共享库的额外的 C 编译器开关。

❖ --ldflags

打印编译数据库时使用的 LDFLAGS 变量的数值。它显示了链接器的开关。

❖ --ldflags_ex

输出用于创建数据库的 LDFLAGS_EX 变量的值。这显示用于构建可执行文件的链接器开关。

❖ --ldflags_sl

打印编译数据库时使用的 LDFLAGS_SL 变量的数值。它显示了用于编译共享库的链接器的开关。

❖ --libs

打印编译数据库时使用的 LIBS 变量的数值。其内容通常包含连接到数据库的外部库的-l 开关。

❖ --version

打印数据库的版本。

❖ -?, --help

显示关于 vb_config 命令行参数的帮助文件，并且退出。

4.13. vb_controldata

功能介绍

vb_controldata 工具用来显示数据库目录下 control 文件的信息。

前提条件

确保数据目录正确，以及相应数据目录下的 pg_control 文件存在。

语法

- ❖ 查看 control 文件信息

```
vb_controldata [DATADIR]
```

- ❖ 显示版本号信息

```
vb_controldata -V | --version
```

- ❖ 显示帮助信息

```
vb_controldata  
-? | --help
```

参数说明

- ❖ DATADIR

查看 control 文件信息时所用的参数，DATADIR 为实例对应的数据库目录。

- ❖ -V, --version

显示版本信息。

- ❖ -?, --help

打印帮助信息。

4.14. vb_resetxlog

功能介绍

vb_resetxlog 是一个重新设置数据库事务文件的工具。

前提条件

保证数据库目录存在且路径正确。

语法

```
vb_resetxlog [OPTION]... DATADIR
```

参数说明

OPTION 取值如下所示：

- ❖ DATADIR
需要修改的数据库目录，确保路径正确。
- ❖ -e XIDPOCH
设置下一个事务 id。
- ❖ -f
强制更新。
- ❖ -l xlogfile
为新的事务日志指定最小的 WAL 起始位置。
- ❖ -m XID
设置下一个 multitransaction ID。
- ❖ -n
不更新，仅显示配置文件的值。
- ❖ -o OID
设置下一个 OID。
- ❖ -O OFFSET
设置下一个 multitransaction 的偏移量。
- ❖ -V, --version
显示版本信息。
- ❖ -x XID
设置下一个事务 ID。
- ❖ -?, --help
打印帮助信息。

4.15. vb_recvlogical

功能介绍

vb_recvlogical 控制逻辑解码复制槽和来自这些复制槽的流数据。

前提条件

保证数据库目录存在且路径正确。

语法

```
vb_recvlogical [OPTION]...
```

参数说明

OPTION 取值如下所示：

- ❖ -f, --file=FILE
将接收到的和解码的事务数据写入到这个文件中。使用-来标准输出。
- ❖ -n, --no-loop
当到服务器的连接断开时，不在一个循环中重试，只是退出。
- ❖ -v, --verbose
启用详细模式。
- ❖ -V, --version
输出版本信息，然后退出
- ❖ -?, --help
显示帮助信息，然后退出
- ❖ -d, --dbname=DBNAME
要连接的数据库名
- ❖ -h, --host=HOSTNAME
声明正在运行服务器的机器的主机名。如果该值以一个斜线开始，那么它用作 Unix 域套接字的路径。
- ❖ -p, --port=PORT
声明 TCP 端口或服务器正在监听链接的本地 Unix 域套接字文件扩展。
- ❖ -U, --username=NAME
要连接的用户名。
- ❖ -w, --no-password
永不发出密码提示。
- ❖ -W, --password
强制 pg_recvlogical 在连接到数据库之前提示密码。
- ❖ -F, --fsync-interval=INTERVAL
频率同步到输出文件（以秒为单位，默认为 10）
- ❖ -o, --option=NAME[=VALUE]
传递选项 name 到输出插件，选项值为 value（如果声明了的话）。
- ❖ -P, --plugin=PLUGIN
当创建一个槽时，使用指定的逻辑解码输出插件。（默认 mppdb_decoding）
- ❖ -s, --status-interval=INTERVAL
发送到服务器的状态包的时间间隔（以秒为单位，默认 10 秒）
- ❖ -S, --slot=SLOT

在--start 模式下，使用现有的名字为 slot_name 的逻辑复制槽。

❖ -l, --startpos=PIR

在--start 模式下，从给定的 LSN 开始复制。

4.16. vb_receivexlog

功能介绍

vb_receivexlog 被用来从一个运行着的 PostgreSQL 集群以流的方式得到事务日志。事务日志会被使用流复制协议以流的方式传送，并且被写入到文件的一个本地目录。这个目录可以被用作归档位置来做一次使用时间点恢复的恢复。

前提条件

保证数据库目录存在且路径正确。

语法

```
vb_receivexlog [OPTION]...
```

参数说明

OPTION 取值如下所示：

❖ -D directory, --directory=directory

要把输出写到这个目录。

❖ -n, --no-loop

不要在连接错误上循环。相反，碰到一个错误时立刻退出。

❖ -v, --verbose

启用详细模式。

❖ -V, --version

输出版本信息，然后退出

❖ -?, --help

显示帮助信息，然后退出

❖ -d, --dbname=DBNAME

要连接的数据库名

❖ -h, --host=HOSTNAME

声明正在运行服务器的机器的主机名。如果该值以一个斜线开始，那么它用作 Unix 域套接字的路径。

❖ -p, --port=PORT

声明 TCP 端口或服务器正在监听链接的本地 Unix 域套接字文件扩展。

- ❖ -U, --username=NAME
要连接的用户名。
- ❖ -w, --no-password
永不发出密码提示。
- ❖ -W, --password
强制 pg_recvlogical 在连接到数据库之前提示密码。
- ❖ -s, --status-interval=INTERVAL
发送到服务器的状态包的时间间隔（以秒为单位，默认 10 秒）

4.17. vb_probackup

背景信息

vb_probackup 是一个用于管理 Vastbase 数据库备份和恢复的工具。它对 Vastbase 实例进行定期备份，以便在数据库出现故障时能够恢复服务器。

可用于备份单机数据库或者集群主节点数据库，为物理备份。

可备份外部目录的内容，如脚本文件、配置文件、日志文件、dump 文件等。

支持增量备份、定期备份和远程备份。

可设置备份的留存策略。

前提条件

- ❖ 可以正常连接 Vastbase 数据库。
- ❖ 在 postgresql.conf 中手动添加参数 “enable_cbm_tracking = on” 。

限制说明

- ❖ 备份必须由运行数据库服务器的用户执行。例如，数据库服务器由用户 postgres 运行，则备份必由用户 postgres 执行。如果是通过 ssh 在远程模式下执行备份，-remote-user 参数的值应为 postgres。
- ❖ 备份和恢复的数据库服务器的主版本号必须相同。
- ❖ 如果要通过 ssh 在远程模式下备份数据库，需要在本地和远程主机安装相同主版本的数据库，并通过 ssh-copy-id remote_user@remote_host 命令设置本地主机备份用户和远程主机数据库用户的无密码 ssh 连接。
- ❖ 远程模式下只能执行 add-instance、backup、restore 子命令。

命令说明

- ❖ 打印 vb_probackup 版本。

```
vb_probackup -V|--version
vb_probackup version
```

- ❖ 显示 vb_probackup 命令的摘要信息。如果指定了 vb_probackup 的子命令，则显示可用于此子命令的参数的详细信息。

```
vb_probackup -?|--help
vb_probackup help [command]
```

- ❖ 初始化备份路径 backup_dir 中的备份目录，该目录将存储已备份的内容。如果备份路径 backup_dir 已存在，则 backup_dir 必须为空目录。

```
vb_probackup init -B backup_dir [--help]
```

- ❖ 在备份路径 backup_dir 内初始化一个新的备份实例，并生成 pg_probackup.conf 配置文件，该文件保存了指定数据目录 data_dir 的 vb_probackup 设置。

```
vb_probackup add-instance -B backup_dir -D data_dir --instance=instance_name
[--help] [-E external-directory-path]
```

```
[remote options]
```

- ❖ 在备份路径 backup_dir 内删除指定实例相关的备份内容。

```
vb_probackup del-instance -B backup_dir --instance=instance_name [--help]
```

- ❖ 将指定的连接、压缩、冗余、日志相关设置和外部目录设置添加到 pg_probackup.conf 配置文件中，或修改已设置的值。不推荐手动编辑 pg_probackup.conf 配置文件。

```
vb_probackup set-config -B backup_dir --instance=instance_name
[--help] [--pgdata=pgdata-path]
[--retention-redundancy=redundancy] [--retention-window=window]
[--compress-algorithm=compression_algorithm] [--compress-level=compression_level]
[-d dbname] [-h host] [-p port] [-U username]
[--external-dirs=external_directory_path]
[--restore-command=cmdline]
[remote options] [logging options]
```

- ❖ 将备份相关设置添加到 backup.control 配置文件中，或修改已设置的值。

```
vb_probackup set-backup -B backup_dir --instance instance_name -i backup_id
[--ttl=ttl | --expire-time=time] [--help]
```

- ❖ 显示位于 “backup_dir/backups/instance_name” 目录中的 pg_probackup.conf 配置文件的内容。可以通过指定 --format=json 选项，以 json 格式显示。默认情况下，显示为纯文本格式。

```
vb_probackup show-config -B backup_dir --instance=instance_name [--format=plain|json]
```

- ❖ 显示备份目录的内容。如果指定了 instance_name 和 backup_id，则显示该备份的详细信息。可以通过指定 --format=json 选项，以 json 格式显示。默认情况下，备份目录的内容显示为纯文本格式。

```
vb_probackup show -B backup_dir
[--help] [--instance=instance_name] [-i backup_id] [--format=plain|json]
```

- ❖ 创建指定实例的备份。backup_mode 选项指定要使用的备份模式。

```
vb_probackup backup -B backup_dir -b backup_mode --instance=instance_name
[--help] [-D pgdata-path] [-C] [--stream [-S slot_name] [--temp-slot]]
[--backup-pg-log] [-j num_threads] [--progress]
[--no-validate] [--skip-block-validation]
[-E external_directory_path] [--no-sync] [--note=text]
[connection options] [compression options] [remote options]
[retention options] [pinning options] [logging options]
```

- ❖ 从备份目录 backup_dir 中的备份副本恢复指定实例。如果指定了恢复目标选项，pg_probackup 将查找最近的备份并将其还原到指定的恢复目标。否则，使用最近一次备份。

```
vb_probackup restore -B backup_dir --instance instance_name
[--help] [-D data_dir] [-i backup_id] [-j num_threads] [--progress]
[-T OLDDIR=NEWDIR] [--external-mapping=OLDDIR=NEWDIR] [--skip-external-dirs] [-I incremental_mode]
[-R | --restore-as-replica] [--no-validate] [--skip-block-validation] [--force]
[--restore-command=cmdline]
[recovery options] [logging options] [remote options]
```

- ❖ 将指定的增量备份与其父完全备份之间的所有增量备份合并到父完全备份。父完全备份将接收所有合并的数据，而已合并的增量备份将作为冗余被删除。

```
vb_probackup merge -B backup_dir --instance instance_name -i backup_id
[--help] [-j num_threads] [--progress]
[logging options]
```

- ❖ 删除指定 backup_id 的备份，或删除不满足当前保留策略的备份和归档 WAL。

```
vb_probackup delete -B backup_dir --instance instance_name
[--help] [-j num_threads] [--progress]
[--retention-redundancy=redundancy] [--retention-window=window]
```

```
[-i backup-id | --delete-expired | --merge-expired | --status=backup_status]
[--delete-wal] [--dry-run]
[logging_options]
```

参数说明

通用参数

❖ command

vb_probackup 除 version 和 help 以外的子命令：init、add-instance、del-instance、set-config、set-backup、show-config、show、backup、restore、merge、delete。

❖ -?, --help

显示 vb_probackup 命令行参数的帮助信息，然后退出。

子命令中只能使用--help，不能使用-?。

❖ -V, --version

打印 vb_probackup 版本，然后退出。

❖ -B backup_dir, --backup-path=backup_dir

备份的路径。

系统环境变量：\$BACKUP_PATH

❖ -D data_dir, --pgdata=data_dir

数据目录的路径。

系统环境变量：\$PGDATA

❖ --instance=instance_name

实例名。

❖ -i id, --backup-id=id

备份的唯一标识。

❖ -j num_threads, --threads=num_threads

设置备份、还原、合并进程的并行线程数。

❖ --progress

显示进度。

❖ --note=text

给备份添加 note。

backup 参数

❖ `-b mode, --backup-mode=mode`

指定备份模式，支持 FULL 和 PTRACK。

FULL：创建全量备份，全量备份包含所有数据文件。

PTRACK：创建 PTRACK 增量备份。

❖ `-C, --smooth-checkpoint`

将检查点在一段时间内展开。默认情况下，`vb_probackup` 会尝试尽快完成检查点。

❖ `--temp-slot`

在备份的实例中为 WAL 流处理创建一个临时物理复制 slot，它确保在备份过程中，所有所需的 WAL 段仍然是可用的。此参数只能与 `--stream` 参数一起使用。默认的 slot 名为 `pg_probackup_slot`，可通过选项 `--slot/-S` 更改。

❖ `-S slot_name, --slot=slot_name`

指定 WAL 流处理的复制 slot。此参数只能与 `--stream` 参数一起使用。

❖ `--backup-pg-log`

将日志目录包含到备份中。此目录通常包含日志消息。默认情况下不包含日志目录。

❖ `-E external_directory_path, --external-dirs=external_directory_path`

将指定的目录包含到备份中。此选项对于备份位于数据目录外部的脚本、sql 转储和配置文件很有用。如果要备份多个外部目录，请在 Unix 上用冒号分隔它们的路径。

❖ `--skip-block-validation`

关闭块级校验，加快备份速度。

❖ `--no-validate`

在完成备份后跳过自动验证。

❖ `--no-sync`

不将备份文件同步到磁盘。

restore 参数

❖ `-I, --incremental-mode=none|lsn`

若 PGDATA 中可用的有效页没有修改，则重新使用它们。

默认值：none

❖ `-T OLDDIR=NEWDIR, --tablespace-mapping=OLDDIR=NEWDIR`

在恢复时，将表空间从 OLDDIR 重新定位到 NEWDIR 目录。OLDDIR 和 NEWDIR 必须都是绝对路径。如果路径中包含 “=”，则使用反斜杠转义。多个表空间可以多次指定此选项。

❖ `--skip-external-dirs`

跳过备份中包含的使用 `--external-dirs` 选项指定的外部目录。这些目录的内容将不会被恢复。

❖ `--skip-block-validation`

跳过块级校验，以加快验证速度。在恢复之前的自动验证期间，将仅做文件级别的校验。

❖ `--no-validate`

跳过备份验证。

❖ `--restore-command=cmdline`

指定恢复相关的命令。例如：`--restore-command='cp /mnt/server/archivedir/%f "%p" '`

❖ `--force`

允许忽略备份的无效状态。如果出于某种原因需要从损坏的或无效的备份中恢复数据，可以使用此标志。请谨慎使用。

恢复目标参数

说明：

❖ 如果配置了连续的 WAL 归档，则可以和 `restore` 命令一起使用这些参数。

❖ `--recovery-target=immediate|latest`

定义何时停止恢复。

`immediate`：当达到指定备份的一致性状态后，停止恢复；如果省略 `-i/-backup_id` 参数，则恢复到最新的可用的备份之后，停止恢复。

`latest`：持续进行恢复，直到应用了所有存档中的所有可用的 WAL 段。

`--recovery-target` 的默认值取决于要恢复的备份的 WAL 传输方式，`STREAM` 流备份为 `immediate`，归档模式为 `latest`。

❖ `--recovery-target-timeline=timeline`

指定要恢复到的 `timeline`。缺省情况下，使用指定备份的 `timeline`。

❖ `--recovery-target-lsn=lsn`

指定要恢复到的 `lsn`。

❖ `--recovery-target-name=recovery_target_name`

指定要将数据恢复到的已命名的保存点。

❖ `--recovery-target-time=time`

指定要恢复到的时间

❖ `--recovery-target-xid=xid`

指定要恢复到的事务 ID

❖ `--recovery-target-inclusive=boolean`

当该参数指定为 `true` 时，恢复目标将包括指定的内容。

当该参数指定为 `false` 时，恢复目标将不包括指定的内容。

该参数必须和--recovery-target-name、--recovery-target-time、--recovery-target-lsn 或 --recovery-target-xid 一起使用。

❖ --recovery-target-action=pause|promote|shutdown
指定恢复至目标时，服务器应执行的操作。

留存参数

说明：

可以和 backup 和 delete 命令一起使用这些参数。

❖ --retention-redundancy=redundancy

指定在数据目录中留存的完整备份数。必须为正整数。0 表示禁用此设置。

默认值：0

❖ --retention-window=window

指定留存的天数。必须为正整数。0 表示禁用此设置。

默认值：0

❖ --wal-depth=wal_depth

每个时间轴上必须留存的执行 PITR 能力的最新有效备份数。必须为正整数。0 表示禁用此设置。

默认值：0

❖ --delete-wal

从任何现有的备份中删除不需要的 WAL 文件。

❖ --delete-expired

删除不符合 pg_probackup.conf 配置文件中定义的留存策略的备份。

❖ --merge-expired

将满足留存策略要求的最旧的增量备份与其已过期的父备份合并。

❖ --dry-run

显示所有可用备份的当前状态，不删除或合并过期备份。

固定备份参数

说明：

如果要将某些备份从已建立的留存策略中排除，可以和 backup 和 set-backup 命令一起使用这些参数。

❖ --ttl=ttl

指定从恢复时间开始计算，备份要固定的时间量。必须为正整数。0 表示取消备份固定。

支持的单位：ms, s, min, h, d（默认为 s）。

例如：--ttl=30d。

❖ --expire-time=time

指定备份固定失效的时间戳。必须是 ISO-8601 标准的时间戳。

例如：-expire-time='2020-01-01 00:00:00+03'

日志参数

日志级别：verbose、log、info、warning、error 和 off。

❖ --log-level-console=log_level

设置要发送到控制台的日志级别。每个级别都包含其后的所有级别。级别越高，发送的消息越少。指定 off 级别表示禁用控制台日志记录。

默认值：info

❖ --log-level-file=log_level

设置要发送到日志文件的日志级别。每个级别都包含其后的所有级别。级别越高，发送的消息越少。指定 off 级别表示禁用日志文件记录。

❖ --log-filename=log_filename

指定要创建的日志文件的文件名。文件名可以使用 strftime 模式，因此可以使用 %-escapes 指定随时间变化的文件名。

例如，如果指定了 “pg_probackup-%u.log” 模式，则 pg_probackup 为每周的每一天生成单独的日志文件，其中 %u 替换为相应的十进制数字，即 pg_probackup-1.log 表示星期一；pg_probackup-2.log 表示星期二，以此类推。

如果指定了 --log-level-file 参数启用日志文件记录，则该参数有效。

默认值：pg_probackup.log

❖ --error-log-filename=error_log_filename

指定仅用于 error 日志的日志文件名。指定方式与 --log-filename 参数相同。

此参数用于故障排除和监视。

❖ --log-directory=log_directory

指定创建日志文件的目录。必须是绝对路径。此目录会在写入第一条日志时创建。

默认值：\$BACKUP_PATH/log/

❖ --log-rotation-size=log_rotation_size

指定单个日志文件的最大大小。如果达到此值，则启动 vb_probackup 命令后，日志文件将循环，但 help 和 version 命令除外。0 表示禁用基于文件大小的循环。

支持的单位：KB、MB、GB、TB（默认为 KB）。

默认值：0

❖ --log-rotation-age=log_rotation_age

单个日志文件的最大生命周期。如果达到此值，则启动 vb_probackup 命令后，日志文件将循环，但 help 和 version 命令除外。\$BACKUP_PATH/log/log_rotation 目录下保存最后一次创建日志文件的时间。0 表示禁用基于时间的循环。

支持的单位：ms, s, min, h, d（默认为 min）。

默认值：0

连接参数

说明：

可以和 backup 命令一起使用这些参数。

❖ -d dbname, --pgdatabase=dbname

指定要连接的数据库名称。该连接仅用于管理备份进程，因此您可以连接到任何现有的数据库。如果命令行、PGDATABASE 环境变量或 pg_probackup.conf 配置文件中没有指定此参数，则 vb_probackup 会尝试从 PGUSER 环境变量中获取该值。如果未设置 PGUSER 变量，则从当前用户名获取。

系统环境变量：\$PGDATABASE

❖ -h host, --pghost=host

指定运行服务器的系统的主机名。如果该值以斜杠开头，则被用作到 Unix 域套接字的路径。

系统环境变量：\$PGHOST

默认值：local socket

❖ -p port, --pgport=port

指定服务器正在侦听连接的 TCP 端口或本地 Unix 域套接字文件扩展名。

系统环境变量：\$PGPORT

默认值：5432

❖ -U username, --pguser=username

指定所连接主机的用户名。

系统环境变量：\$PGUSER

❖ -w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

❖ -W, --password

强制出现输入密码提示。

压缩参数

说明：

可以和 backup 命令一起使用这些参数。

❖ --compress-algorithm=compression_algorithm

指定用于压缩数据文件的算法。取值包括 zlib、pglz 和 none。如果设置为 zlib 或 pglz，此选项将启用压缩。默认情况下，压缩功能处于关闭状态。

默认值：none

❖ `--compress-level=compression_level`

指定压缩级别。取值范围：0~9

0 表示无压缩。

1 表示压缩比最小，处理速度最快。

9 表示压缩比最大，处理速度最慢。

可与 `--compress-algorithm` 选项一起使用。

默认值：1

❖ `--compress`

以 `--compress-algorithm=zlib` 和 `--compress-level=1` 进行压缩。

远程模式参数

说明：

通过 SSH 远程运行 `vb_probackup` 操作的相关参数。可以和 `add-instance`、`set-config`、`backup`、`restore` 命令一起使用这些参数。

❖ `--remote-proto=proto`

指定用于远程操作的协议。目前只支持 SSH 协议。取值包括：

`ssh`：通过 SSH 启用远程备份模式。这是默认值。

`none`：显式禁用远程模式。

如果指定了 `--remote-host` 参数，可以省略此参数。

❖ `--remote-host=destination`

指定要连接的远程主机的 IP 地址或主机名。

❖ `--remote-port=port`

指定要连接的远程主机的端口号。

默认值：22

❖ `--remote-user=username`

指定 SSH 连接的远程主机用户。如果省略此参数，则使用当前发起 SSH 连接的用户。

默认值：当前用户

❖ `--remote-path=path`

指定 `vb_probackup` 在远程系统的安装目录。

❖ `--ssh-options=ssh_options`

指定 SSH 命令行参数的字符串。

备份流程

1. 初始化备份目录。在指定的目录下创建 `backups/` 和 `wal/` 子目录，分别用于存放备份文件和 WAL 文件。

```
vb_probackup init -B backup_dir
```

2. 添加一个新的备份实例。vb_probackup 可以在同一个备份目录下存放多个数据库实例的备份。

```
vb_probackup add-instance -B backup_dir -D data_dir --instance instance_name [remote_options]
```

3. 创建指定实例的备份。在进行增量备份之前，必须至少创建一次全量备份。

```
vb_probackup backup -B backup_dir --instance instance_name -b backup_mode
```

4. 从指定实例的备份中恢复数据。

```
vb_probackup restore -B backup_dir --instance instance_name -i backup_id
```

5. FAQ

[5.1 执行命令报错“Failed to obtain the GPHOME”](#)

5.1. 执行命令报错“Failed to obtain the GPHOME”

问题现象

如果在 root 用户下执行工具命令时，出现如下信息，

```
Failed to obtain the GPHOME.
```

原因分析

可能是环境变量中 GPHOME 变量没有配置正确。需要检查环境变量 GPHOME 是否包含 VastbaseXML 中 VastbaseToolPath 路径。

操作步骤

使用如下命令检查\$GPHOME 路径。

```
echo $GPHOME
```

如果不是安装时默认的路径，请在配置文件中修改 GPHOME 的路径。

```
vim /etc/profile
```

6. gs_collector 工具支持收集的系统和视图列表

表 1-41 gs_collector 工具支持收集的系统和视图列表

CLASS	VIEW NAME
OS	gs_os_run_info
	pg_os_threads
Session /Thread	gs_session_stat
	gs_session_time
	gs_session_cpu_statistics
	gs_session_memory_statistics
	gs_session_memory_context
	gs_session_memory
	gs_session_memory_detail
	pg_stat_activity
WLM	gs_cluster_resource_info
	pg_total_user_resource_info
	pg_total_user_resource_info_oid
	gs_wlm_rebuild_user_resource_pool
	gs_wlm_resource_pool
	gs_wlm_workload_records
	pg_wlm_statistics
	pg_app_workloadgroup_mapping
Query-AP Operator	gs_wlm_operator_info
	gs_wlm_operator_history
	gs_wlm_operator_statistics
Query-AP	gs_wlm_session_info_all
	gs_wlm_session_history
	gs_wlm_session_info
	gs_wlm_session_statistics
	pg_session_iostat
	pg_session_wlmstat

CLASS	VIEW NAME
Cache I/O	pg_statio_all_indexes
	pg_statio_all_sequences
	pg_statio_all_tables
	pg_statio_sys_indexes
	pg_statio_sys_sequences
	pg_statio_sys_tables
	pg_statio_user_indexes
	pg_statio_user_sequences
	pg_statio_user_tables
	gs_stat_db_cu
	gs_stat_session_cu
Memory	gs_total_memory_detail
File	gs_file_stat
	gs_redo_stat
	pg_stat_bad_block
Object	pg_stat_user_tables
	pg_stat_user_indexes
	pg_stat_sys_tables
	pg_stat_sys_indexes
	pg_stat_all_tables
	pg_stat_all_indexes
	pg_stat_database
	pg_stat_database_conflicts
	pg_stat_xact_all_tables
	pg_stat_xact_sys_tables
	pg_stat_xact_user_tables
	pg_stat_xact_user_functions
Lock	pg_locks
Utility	pg_stat_replication
	pg_replication_slots
	pg_stat_bgwriter
Transaction	pg_running_xacts

CLASS	VIEW NAME
	pg_prepared_xacts
Waits	pg_thread_wait_status



电话: 010-82838118

地址: 北京市海淀区学院路 30 号科大天工大厦 B 座 6 层

官网: www.vastdata.com.cn