

北京海量数据技术股份有限公司

**Vastbase G100 V2.2**

**(Build 10)**

**工具参考**

【版权声明】

©2007-2022 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。

【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](#_Toc16963)

[2. 工具介绍 5](#_Toc26473)

[2.1. createdb 5](#_Toc21075)

[2.2. createuser 7](#_Toc7221)

[2.3. dropedb 10](#_Toc4510)

[2.4. dropuser 11](#_Toc8036)

[2.5. ECPG 13](#_Toc13953)

[2.6. gs\_encrypt 16](#_Toc20618)

[2.7. pg\_xlogdump 18](#_Toc16962)

[2.8. vacuumdb 21](#_Toc21456)

[2.9. vb\_basebackup 24](#_Toc14456)

[2.10. vb\_bulkload 28](#_Toc5689)

[2.11. vb\_config 30](#_Toc4794)

[2.12. vb\_controldata 32](#_Toc28955)

[2.13. vb\_ctl 33](#_Toc23225)

[2.14. vb\_dump 40](#_Toc20046)

[2.15. vb\_dumpall 56](#_Toc26159)

[2.16. vb\_guc 62](#_Toc31453)

[2.17. vb\_initdb 72](#_Toc19277)

[2.18. vb\_isready 78](#_Toc1090)

[2.19. vb\_licensetool 79](#_Toc6538)

[2.20. vb\_probackup 80](#_Toc7276)

[2.21. vb\_restore 96](#_Toc518)

[2.22. VPatch 102](#_Toc9002)

[2.23. vsql 113](#_Toc13273)

# 工具一览表

Vastbase提供客户端和服务器端应用程序（工具），可以帮助用户更好地维护Vastbase，提供的所有工具如下表所示。工具位于安装数据库服务器的$GAUSSHOME/bin路径下。

| **工具名称** | **简介** |
| --- | --- |
| createdb | 创建一个新的数据库，通常，执行这个命令的数据库用户将成为新数据库的所有者。 不过，如果拥有合适的权限，那么也可以通过 -o指定其他用户。createdb 是一个 SQL 命令 CREATE DATABASE的封装。 |
| createuser | createuser创建一个新的用户，只有超级用户和具有CREATE ROLE权限的用户可以创建新的用户。 因此createuser必须由超级用户或者是具有 CREATE ROLE权限的用户执行。createuser是对 SQL命令CREATE ROLE的封装。 |
| dropdb | dropdb 删除一个现有 数据库。 执行此命令的用户必须是数据库超级用户或者是数据库的所有者。dropdb是对 SQL命令DROP DATABASE的封装。 |
| dropuser | dropuser删除现有用户。 只有超级用户和拥有CREATE ROLE权限的用户可以删除用户。 （要删除一个超级用户，你必须首先是超级用户。）dropuser是对 SQL命令DROP ROLE的封装。 |
| EPCG | ECPG工具用于处理C中的嵌入式SQL语句，将包含嵌入式SQL语句的C程序转换成可以被C/C++编译器编译执行的C/C++程序。 |
| gs\_encrypt | gs\_encrypt是一款用于为输入的明文字符串进行加密操作。 |
| pg\_xlogdump | 该工具用于对wal日志的解析。 |
| vacuumdb | vacuumdb是一个用于整理数据库的工具。vacuumdb还将会生成用于查询优化器的内部统计数据。vacuumdb是 SQL 命令VACUUM的封装。 |
| vb\_basebackup | 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\_bulkload | vb\_bulkload就是批量加载工具，可以加载表数据，批量加载可以提升加载的效率和性能。在使用vb\_bulkload工具导入出错之后，本次导入的数据都要进行回滚，所以需要将本次导入数据清理，防止这些数据被访问，造成查询、执行计划不准确等问题。 1、使用此功能需安装pg\_bulkload插件。  2、实现vb\_bulkload导入时，将在导入报错时清理本次导入数据，需要指定-o “ERROR\_RECOVERY=YES”,该参数默认值为NO。 |
| vb\_config | 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。  可使用vb\_config --help命令查看具体参数。 |
| vb\_controldata | vb\_controldata工具用来显示数据库目录下control文件的信息。 |
| vb\_ctl | 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vastbase执行。 1、启动、停止、重启vastbase节点。 2、在不停止数据库的情况下，重新加载配置文件（postgresql.conf，pg\_hba.conf）。 3、主备切换，主备状态查询、重建和重建状态查询。 |
| 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。 |
| 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。 |
| 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。 |
| vb\_initdb | 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。 |
| vb\_isready | 检查服务器的连接状态。 |
| vb\_licensetool | 该工具用于查看license的相关信息。 |
| vb\_probackup | 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。 1、可用于备份单机数据库，也可对主机或者主节点数据库备机进行备份，为物理备份 2、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 3、支持增量备份、定期备份和远程备份 4、可设置备份留存策略 |
| 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。 |
| VPatch | VPatch是用于Vastbase版本控制的工具可以实现在单机环境下的升级和回退操作。 |
| 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，比如连接数据库、执行SQL语句、执行元命令。vsql还提供了若干高级特性，便于用户使用，比如命令自动补齐，设置变量等等。 |
| vb\_resetxlog | vb\_resetxlog是一个重新设置数据库预写日志以及其它控制内容。 注意：通过vb\_resetxlog重新设置之前，需要先停止数据库。 |
| vb\_receivexlog | pg\_receivexlog用于从一个运行的集群中流事务日志。该事务日志流使用流复制协议，并写入到文件的本地目录中。 这个目录可以被用作使用时间点恢复的归档目录. |
| vb\_recvlogical | vb\_recvlogical工具通过连接指定的节点，创建或删除逻辑复制槽，以及持续、实时的从该节点获取逻辑解码中间结果，输出到文件或标准输出。 |

# 工具介绍

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。

## createdb

**功能描述**

创建一个新的Vastbase数据库。通常，执行这个命令的数据库用户将成为新数据库的所有者。 不过，如果拥有合适的权限，那么也可以通过 -O指定其他用户。

createdb 是一个 SQL 命令 CREATE DATABASE的封装。 因此，用哪种方法创建的数据库都一样。

**语法格式**

createdb [connection-option...] [option...] [dbname [description]]

详细选项可使用如下命令查看：

createdb --help

返回结果为：

createdb creates a Vastbase database.  
Usage:  
 createdb [OPTION]... [DBNAME] [DESCRIPTION]  
Options:  
 -D, --tablespace=TABLESPACE default tablespace for the database  
 -e, --echo show the commands being sent to the server  
 -E, --encoding=ENCODING encoding for the database  
 -l, --locale=LOCALE locale settings for the database  
 --lc-collate=LOCALE LC\_COLLATE setting for the database  
 --lc-ctype=LOCALE LC\_CTYPE setting for the database  
 -O, --owner=OWNER database user to own the new database  
 -T, --template=TEMPLATE template database to copy  
 -V, --version output version information, then exit  
 -?, --help show this help, then exit  
  
Connection options:  
 -h, --host=HOSTNAME database server host or socket directory  
 -p, --port=PORT database server port  
 -U, --username=USERNAME user name to connect as  
 -w, --no-password never prompt for password  
 -W, --password force password prompt

--maintenance-db=DBNAME alternate maintenance database  
By default, a database with the same name as the current user is created.

**示例**

用缺省数据库服务器创建一个demo数据库：

[vastbase@localhost root]$ createdb demo

返回结果为：

NOTICE: Last successful login info  
login time:2022-08-16 11:57:26  
application name:vb\_isready  
Ip address:[local]  
method:Trust  
  
Failed 0 times since last successful login

## createuser

**功能描述**

createuser创建一个新的 用户（更准确地说是一个角色）。 只有超级用户和具有CREATE ROLE权限的用户可以创建新的用户。 因此createuser必须由超级用户或者是具有 CREATEROLE权限的用户执行。

如果你想创建一个新的超级用户，你必须以超级用户身份连接， 而不仅仅是一个有CREATE ROLE权限的用户。 成为超级用户就意味着将在数据库里绕开所有访问权限检查，因此，不要轻易授予超级用户权限。

createuser是对 SQL命令CREATE ROLE的封装。 因此，用哪种方法创建的用户都一样。

**语法格式**

createuser [connection-option...] [option...] [username]

详细选项可使用如下命令查看：

createuser --help

返回结果为：

createuser creates a new Vastbase role.  
Usage:  
 createuser [OPTION]... [ROLENAME]  
Options:  
 -c, --connection-limit=N connection limit for role (default: no limit)  
 -d, --createdb role can create new databases  
 -D, --no-createdb role cannot create databases (default)  
 -e, --echo show the commands being sent to the server  
 -E, --encrypted encrypt stored password  
 -i, --inherit role inherits privileges of roles it is a  
 member of (default)  
 -I, --no-inherit role does not inherit privileges  
 -l, --login role can login (default)  
 -L, --no-login role cannot login  
 -N, --unencrypted do not encrypt stored password  
 -P, --pwprompt assign a password to new role  
 -r, --createrole role can create new roles  
 -R, --no-createrole role cannot create roles (default)  
 -s, --superuser role will be system admin  
 -S, --no-superuser role will not be system admin (default)  
 -V, --version output version information, then exit  
 --interactive prompt for missing role name and attributes rather  
 than using defaults  
 --replication role can initiate replication  
 --no-replication role cannot initiate replication  
 --auditadmin role can administer audit data  
 --no-auditadmin role cannot administer audit data  
 -?, --help show this help, then exit  
  
Connection options:  
 -h, --host=HOSTNAME database server host or socket directory  
 -p, --port=PORT database server port  
 -U, --username=USERNAME user name to connect as (not the one to create)  
 -w, --no-password never prompt for password  
 -W, --password force password prompt

**示例**

**示例1：**在端口5432的服务端里创建joe用户， 明确指定属性，请看下列命令：

createuser -p 5432 -S -D -R -e jotr

返回结果为：

NOTICE: Last successful login info  
login time:2022-08-16 12:11:02  
application name:createuser  
Ip address:[local]  
method:Trust  
  
Last failed login info  
login time:2022-08-16 12:13:48  
application name:createuser  
Ip address:::1  
method:SHA256  
  
Failed 3 times since last successful login  
CREATE ROLE jotr NOSYSADMIN NOCREATEDB NOCREATEROLE INHERIT LOGIN;

**示例2：**创建一个用户joe，并立即指定密码(输入两次密码Huawei@123)：

[vastbase@localhost root]$ createuser -P -s -e joe  
Enter new system admin password:  
Enter it again:

返回结果为:

NOTICE: Last successful login info  
login time:2022-08-17 13:46:16  
application name:createuser  
Ip address:[local]  
method:Trust  
Last failed login info  
login time:2022-08-16 12:13:48  
application name:createuser  
Ip address:::1  
method:SHA256  
Failed 0 times since last successful login  
CREATE ROLE joe PASSWORD 'Huawei@123' SYSADMIN CREATEDB CREATEROLE INHERIT LOGIN;

## dropedb

**功能描述**

删除一个Vastbase数据库。执行此命令的用户必须是数据库超级用户或者是数据库的所有者。

dropdb是对 SQL命令DROP DATABASE的封装。 因此，用哪种方法删除数据库都一样。

**语法格式**

dropdb [connection-option...] [option...] dbname

详细选项可使用如下命令查看：

dropdb --help

返回结果为：

dropdb removes a Vastbase database.  
Usage:  
 dropdb [OPTION]... DBNAME  
  
Options:  
 -e, --echo show the commands being sent to the server  
 -i, --interactive prompt before deleting anything  
 -V, --version output version information, then exit  
 --if-exists don't report error if database doesn't exist  
 -?, --help show this help, then exit  
  
Connection options:  
 -h, --host=HOSTNAME database server host or socket directory  
 -p, --port=PORT database server port  
 -U, --username=USERNAME user name to connect as  
 -w, --no-password never prompt for password  
 -W, --password force password prompt  
 --maintenance-db=DBNAME alternate maintenance database

**示例**

用缺省数据库服务器创建一个demo数据库：

[vastbase@localhost root]$ dropdb demo

返回结果为：

NOTICE: Last successful login info  
login time:2022-08-16 12:04:12  
application name:createdb  
Ip address:[local]  
method:Trust  
  
Failed 0 times since last successful login

## dropuser

**功能描述**

dropuser删除现有的用户。 只有超级用户和拥有CREATEROLE权限的用户可以删除用户。 （要删除一个超级用户，你必须首先是超级用户。）

dropuser是对 SQL命令DROP ROLE的封装。 因此，用哪种方法删除用户都一样。

**语法格式**

dropuser [connection-option...] [option...] [username]

详细选项可使用如下命令查看：

dropuser --help

返回结果为：

dropuser removes a Vastbase role.  
Usage:  
 dropuser [OPTION]... [ROLENAME]  
Options:  
 -e, --echo show the commands being sent to the server  
 -i, --interactive prompt before deleting anything, and prompt for  
 role name if not specified  
 -V, --version output version information, then exit  
 --if-exists don't report error if user doesn't exist  
 -?, --help show this help, then exit  
  
Connection options:  
 -h, --host=HOSTNAME database server host or socket directory  
 -p, --port=PORT database server port  
 -U, --username=USERNAME user name to connect as (not the one to drop)  
 -w, --no-password never prompt for password

**示例**

删除一个已经存在的用户joe：

[vastbase@localhost root]$ dropuser joe

返回结果如下则表示用户已经删除：

NOTICE: Last successful login info  
login time:2022-08-17 13:49:08  
application name:createuser  
Ip address:[local]  
method:Trust  
  
Last failed login info  
login time:2022-08-16 12:13:48  
application name:createuser  
Ip address:::1  
method:SHA256  
  
Failed 0 times since last successful login

## ECPG

**功能描述**

ECPG工具用于处理C中的嵌入式SQL语句，将包含嵌入式SQL语句的C程序转换成可以被C/C++编译器编译执行的C/C++程序。支持Vastbase SQL语句和ECPG SQL两种嵌入式SQL语句：

* 除部分SQL命令不支持外其余Vastbase SQL命令全部支持。以下为不支持的SQL命令：
* 匿名块相关，包括开启匿名块和创建包含匿名块的函数。

[DECLARE [declare\_statements]]   
 BEGIN  
 execution\_statements   
 END;  
 /

* 不支持创建package。

create package XXX

注：除以上SQL语句之外Vastbase支持的所有SQL语句都可以作为嵌入式SQL语句使用。

* ECPG SQL命令：详细的ECPG SQL语法使用及功能参考《PG11.2手册》（<http://www.postgres.cn/docs/11/>）。

ECPG SQL语法列表如下：

* ALLOCATE DESCRIPTOR ：分配一个SQL描述符区域
* CONNECT ：建立一个数据库连接
* DEALLOCATE DESCRIPTOR：释放一个SQL描述符区域
* DECLARE ：定义一个游标
* DESCRIBE ：得到有关一个预备语句或结果集的信息
* DISCONNECT ： 终止一个数据库连接
* EXECUTE IMMEDIATE ： 动态地准备和执行一个语句
* GET DESCRIPTOR ： 从一个SQL描述符区域得到信息
* OPEN ：打开一个动态游标
* PREPARE ： 准备一个语句用于执行
* SET AUTOCOMMIT ： 设置当前会话的自动提交行为
* SET CONNECTION ： 选择一个数据库连接
* SET DESCRIPTOR ： 在一个SQL描述符区域中设置信息
* TYPE： 定义一种新数据类型
* VAR ： 定义一个变量
* WHENEVER ：指定一个要在一个SQL语句导致发生一个特定类别的情况时要采取的动作
* 注：在Vastbase 中，ECPG SQL中的CALL和 IDENTIFIED改名为ECPGCALL和 ECPGIDENTIFIED。

**语法格式**

在C语言中使用嵌入式SQL语句的语法如下：

EXEC SQL stmt;

**参数说明**

stmt：一条SQL语句。

**使用流程**

1、安装EPCG工具及其依赖库。

2、编写一个C程序，后缀为.pgc，在文件中使用嵌入式SQL语句，例如在prog1.pgc文件中使用SQL命令。

EXEC SQL CREATE TABLE t1(id int, info text); --定义表t1

3、使用ECPG工具将pgc文件，例如将prog1.pgc文件编译为prog1.c。

ecpg prog1.pgc

4、编译可执行程序。

cc -I/usr/local/pgsql/include -c prog1.c  
cc -o prog1 prog1.o -L/usr/local/pgsql/lib -lecpg

5、运行可执行程序。

./prog1

**示例**

在C程序中通过嵌入式SQL实现数据库连接、创建表、插入、修改、查询、连接断开。

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
EXEC SQL INCLUDE ../regression;  
int main() {  
 EXEC SQL BEGIN DECLARE SECTION;  
 int i1[3], i2[3], i3[3], i4;  
 EXEC SQL END DECLARE SECTION;  
  
 ECPGdebug(1, stderr);  
 EXEC SQL CONNECT TO REGRESSDB1;  
  
 EXEC SQL WHENEVER SQLWARNING SQLPRINT;  
 EXEC SQL WHENEVER SQLERROR SQLPRINT;  
  
 EXEC SQL CREATE TABLE insupd\_test(a int, b int);  
  
 EXEC SQL INSERT INTO insupd\_test (a,b) values (1, 1);  
 EXEC SQL INSERT INTO insupd\_test (a,b) values (2, 2);  
 EXEC SQL INSERT INTO insupd\_test (a,b) values (3, 3) returning a into :i4;  
  
 EXEC SQL UPDATE insupd\_test set a=a+1 returning a into :i3;  
 EXEC SQL UPDATE insupd\_test set (a,b)=(5,5) where a = 4;  
 EXEC SQL UPDATE insupd\_test set a=4 where a=3;;  
  
 EXEC SQL SELECT a,b into :i1,:i2 from insupd\_test order by a;  
  
 printf("changes\n%d %d %d %d\n", i3[0], i3[1], i3[2], i4);  
 printf("test\na b\n%d %d\n%d %d\n%d %d\n", i1[0], i2[0], i1[1], i2[1], i1[2], i2[2]);  
 EXEC SQL DISCONNECT ALL;  
 return 0;  
}

## gs\_encrypt

**背景信息**

Vastbase提供了gs\_encrypt工具为输入的明文字符串进行加密操作。

**语法**

gs\_encrypt [OPTION]... PLAINTEXT

image1

* PLAINTEXT”前不需要加短选项或长选项。
* “PLAINTEXT”是需要加密的明文字符串。

**参数说明**

* -?, --help
* 显示帮助信息。
* -V, --version
* 显示版本号信息。
* -k PASSWORD，--key=PASSWORD
* 用户输入的口令，用于派生出加密操作时使用的密钥。口令长度有效范围为8~16个字符，并且至少包含大写字母、小写字母、数字和特殊字符中的三种字符。
* -v VectorValue，--vector=VectorValue
* 用户输入的盐值，计算口令单向哈希时用于对口令进行加扰。字符串长度必须等于16。
* -f FilePrefix,--file-prefix=FilePrefix
* 加密操作时使用的cipher和rand文件前缀字符串。cipher和rand文件必须存放在$GAUSSHOME/bin目录下。
* -B Value，--key-base64=Value
* 加密时使用的明文密钥，以base64编码。
* -D Value， --vector-base64=Value
* 加密时使用的随机数，以base64编码。base64编码前的字符串长度必须为16。

image1

出于安全考虑，建议用户在使用gs\_encrypt工具前设置HISTCONTROL=ignorespace，强制不记录敏感历史命令，然后在执行gs\_encrypt命令行前面加空格。

**示例**

**示例1：**

通过输入key和vector明文值来对明文字符串进行加密操作。由于每次加密时使用的IV值是随机生成的，因此使用同样的key和vector以及明文字符串加密后对应的密文串不相同。单次支持加密的明文字符串最大长度为335字节。

gs\_encrypt -k password@123 -v 1234567890123456 test\_encrypt

命令回显：

G0R7Y2oROfuiEjWtUBx9+eHP3I21A1wMOJ2onVQIiAsxMjM0NTY3ODkwMTIzNDU2

**示例2：**

已经存在cipher和rand文件，通过文件前缀来对明文字符串进行加密操作。

1、生成cipher和rand文件。

vb\_guc generate -S Mppdb@123 -D $GAUSSHOME/bin -o test

命令回显：

vb\_guc encrypt -S \*\*\*

2、执行加密操作。

gs\_encrypt -f test test\_encrypt

命令回显：

JlKZgPIuDq5+PjJfb+vvXiH7Qt0SygmbGdszQAAASnkm69Ve0W/qgaSUUw9nbI7Z

## pg\_xlogdump

**功能描述**

该工具用于解析XLOG日志，包括日志类型、对应的事务号、修改的文件等等。。

**语法格式**

pg\_xlogdump [option...] [startseg [endseg] ]

**参数说明**

* **startseg**

开始从指定WAL段文件读取。这隐式的确定了搜索文件和使用的时间线。

* **endseg**

指定到哪个WAL段文件停止读取。

* **-b, --bkp-details**

输出关于备份块的详细信息。

* **-e, --end=RECPTR**

指定停止读取的日志位置，代替读取的日志流的结尾。

* **-n, --limit=N**

指定显示记录的条目数。

* **-p, --path=PATH**

指定搜索WAL段文件的目录。默认从当前目录的 pg\_xlog子目录去搜索它们。

* **-r, --rmgr=RMGR**

过滤由指定资源管理器产生的记录。如果 list 一个指定一个名字， 打印一列合法的资源管理器的名字的记录，然后退出。

* **-s, --start=RECPTR**

开始读取日志的位置。默认是从最早的文件搜索到的第一个有效的日志记录开始。

* **-t, --timeline=TLI**

从哪个时间线读取日志记录。如果startseg的值被指定默认就使用startseg; 否则默认值是1。

* **-V, --version**

打印出vb\_xlogdump版本并且退出。

* **-w, --write-FPW**

设置整页写出到磁盘。

* **-x, --xid=XID**

仅显示指定事务ID的记录。

* **-z, --stats**

显示统计信息。

* **-v, --verbose**

显示详细的信息。

* **-?, --help**

显示vb\_xlogdump 命令参数帮助，然后退出。

**示例**

1、查询pg\_xlog目录下面的wal日志。

[vastbase@vdb2 pg\_xlog]$ ll

返回结果为：

total 229376  
-rw------- 1 vastbase vastbase 16777216 Oct 13 10:14 000000010000000000000001  
-rw------- 1 vastbase vastbase 16777216 Oct 13 10:15 000000010000000000000002  
-rw------- 1 vastbase vastbase 16777216 Oct 14 21:08 000000010000000000000003  
-rw------- 1 vastbase vastbase 16777216 Oct 16 06:58 000000010000000000000004  
-rw------- 1 vastbase vastbase 16777216 Oct 17 08:53 000000010000000000000005  
-rw------- 1 vastbase vastbase 16777216 Oct 18 12:32 000000010000000000000006  
-rw------- 1 vastbase vastbase 16777216 Oct 19 21:17 000000010000000000000007  
-rw------- 1 vastbase vastbase 16777216 Oct 20 23:38 000000010000000000000008  
-rw------- 1 vastbase vastbase 16777216 Oct 22 01:55 000000010000000000000009  
-rw------- 1 vastbase vastbase 16777216 Oct 23 10:59 00000001000000000000000A  
-rw------- 1 vastbase vastbase 16777216 Oct 24 13:01 00000001000000000000000B  
-rw------- 1 vastbase vastbase 16777216 Oct 25 15:17 00000001000000000000000C  
-rw------- 1 vastbase vastbase 16777216 Oct 27 00:24 00000001000000000000000D  
-rw------- 1 vastbase vastbase 16777216 Oct 27 16:25 00000001000000000000000E  
drwx------ 2 vastbase vastbase       6 Oct 13 10:14 archive\_status

2、展现从 000000010000000000000001到 000000010000000000000003之间的 WAL 日志，-n 表示显示的行数。

[vastbase@vdb2 pg\_xlog]$pg\_xlogdump -b -n2 000000010000000000000001 000000010000000000000003

返回结果为：

REDO @ 0/1000028; LSN 0/10000A8: prev 0/0; xid 0; term 0; len 88; total 122; crc 907798150; desc: XLOG - checkpoint: redo 0/1000028; len 0; next\_csn 0; recent\_global\_xmin 0; tli 1; fpw false; xid 4; oid 10000; multi 2; offset 0; oldest xid 3 in DB 1; oldest running xid 0; oldest xid with epoch having undo 0; shutdown at Thu Oct 13 10:14:51 2022; remove\_seg 0/0  
REDO @ 0/10000A8; LSN 0/10000D0: prev 0/1000028; xid 1; term 1; len 4; total 38; crc 2563411093; desc: XLOG - nextOid: 18192

3、展现 WAL 中所有的 resource manager 名称。

pg\_xlogdump --rmgr=list

4、仅展现 WAL 中资源名称Database 的日志。

pg\_xlogdump -r Database -b -n2 000000010000000000000001 000000010000000000000003

返回结果为：

REDO @ 0/32E82E0; LSN 0/32E8318: prev 0/32E8240; xid 12209; term 1; len 16; total 50; crc 4052575843; desc: Database - create db: copy dir 1/1663 to 16561/1663  
​  
REDO @ 0/32F2E10; LSN 0/32F2E48: prev 0/32F2D70; xid 12580; term 1; len 16; total 50; crc 2420045689; desc: Database - create db: copy dir 1/1663 to 16566/1663

5、查看当前LSN。

select pg\_current\_xlog\_location();

返回结果为：

pg\_current\_xlog\_location  
--------------------------  
 0/EB34AD0  
(1 row)

6、查看LSN位于哪个WAL文件。

select pg\_xlogfile\_name('0/EB34AD0');

返回结果为：

    pg\_xlogfile\_name  
--------------------------  
 00000001000000000000000E  
(1 row)

7、根据LSN及日志文件可以解析查询日志文件中内容。

例如查询 00000001000000000000000E文件中LSN号为EB34AD0的相关内容命令为：

pg\_xlogdump 00000001000000000000000E | grep EB34AD0

## vacuumdb

**功能描述**

vacuumdb是一个用于整理数据库的工具。vacuumdb还将会生成用于 查询优化器的内部统计数据。

vacuumdb是 SQL 命令VACUUM的封装。 因此，用哪种方法清理数据库都一样。

**语法格式**

vacuumdb [connection-option...] [option...] [ --table | -t table [( column [,...] )] ] ... [dbname]

vacuumdb [connection-option...] [option...] --all | -a

**参数说明**

vacuumdb接受下列命令行参数：

* -a ,--all：清理所有数据库。
* [-d] dbname ,[--dbname=]*dbname*：声明要被清理或分析的数据库名称。
* -e ,--echo：回显vacuumdb生成并发送给服务器的命令。
* -f,--full：执行"完全"清理。
* -F ,--freeze：强制"冻结"元组。
* -q,--quiet：不显示进程信息。
* -t *table* [ (*column* [,...]) ] ,--table=*table* [ (*column* [,...]) ]：只是清理或分析*table*。 字段名称只是在与--analyze或--analyze-only 选项联合使用时才需要声明。可以通过写多个-t选项清理多个表。**提示:** 如果你声明了要清理的字段，你可能不得不在 shell 上转义圆括弧(见下面的例子)。
* -v ,--verbose：在处理过程中打印详细信息。
* -V , --version：打印vacuumdb版本然后退出。
* -z ,--analyze：计算用于优化器的统计数据。
* -Z, --analyze-only：只计算用于优化器的该数据库的统计值（不清理）。
* --analyze-in-stages：只计算优化器使用的统计（没有清理），像--analyze-only。 用不同的配置设置运行几个（当前是3个）阶段的分析，以更快的产生可用的统计。这个选项对于分析刚用存储的转储或通过pg\_upgrade 填充的数据库来说是有用的。这个选项将尽快的尝试创建一些统计， 使得数据库可用，然后在随后的阶段中产生完全统计。
* -? ,--help：显示关于vacuumdb命令行参数的帮助然后退出。

vacuumdb还接受下列命令行参数作为连接参数：

* -h *host* ,--host=*host*：指定运行服务器的主机名。如果数值以斜杠开头，则被用作到 Unix 域套接字的路径。
* -p *port* ,--port=*port*：指定服务器正在侦听的 TCP 端口或本地 Unix 域套接字文件的扩展(描述符)。
* -U *username* ,--username=*username*：连接的用户名。
* -w ,--no-password：从不发出密码提示问题。如果服务器要求密码认证并且密码不可用于其他意思如 .pgpass文件，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vacuumdb在连接到数据库之前提示一个密码。这个选项从来不是至关重要的，因为如果服务器需求密码认证，则vacuumdb 自动提示一个密码。不过，vacuumdb 将在找出服务器想要一个密码上浪费一个连接尝试。在某些情况下，值得输入-W 以避免额外的连接尝试。
* --maintenance-db=*dbname*：指定要连接到的数据库的名字以发现其他应该清理的数据库。

**示例**

整理数据库：

[vastbase@localhost root]$ vacuumdb db\_oracle

为优化器清理和分析一个名为db\_oracle的数据库：

$ vacuumdb --analyze db\_oracle

## vb\_basebackup

**功能描述**

Vastbase部署成功后，在数据库运行的过程中，可能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当前支持热备份模式和压缩格式备份模式。

* vb\_basebackup仅支持主机和备机的全量备份，不支持增量。
* vb\_basebackup当前支持热备份模式和压缩格式备份模式。
* 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。
* vb\_basebackup在备份包含绝对路径的表空间时，如果在同一台机器上进行备份，可以通过tablespace-mapping重定向表空间路径或使用归档模式进行备份。
* 若pg\_xlog目录为软链接，备份时将不会建立软链接，会直接将数据备份到目的路径的pg\_xlog目录下。
* 备份过程中收回用户备份权限，可能导致备份失败或者备份数据不可用。
* 如果因为网络临时故障等原因导致Server端无法应答，vb\_basebackup将在最长等待120秒后退出。

**前提条件**

* 可以正常连接Vastbase数据库。
* 备份过程中用户权限没有被回收。
* pg\_hba.conf中需要配置允许复制链接，且该连接必须由一个系统管理员建立。
* 如果xlog传输模式为stream模式，需要配置max\_wal\_senders的数量，至少有一个可用。
* 如果xlog传输模式为fetch模式，有必要把wal\_keep\_segments参数设置得足够高，这样在备份末尾之前日志不会被移除。
* 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。

**语法格式**

* 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。
* -T，-tablespace-mapping=olddir=newdir：在备份期间将目录olddir中的表空间重定位到newdir中。为使之有效，olddir必须正好匹配表空间所在的路径(但如果备份中没有包含olddir中的表空间也不是错误）。olddir和newdir必须是绝对路径。如果一个路径凑巧包含了一个=符号，可用反斜线对它转义。对于多个表空间可以多次使用这个选项。
* -F，-format=plain|tar：设置输出格式为plain(默认）或者tar。没有设置该参数的情况下，默认-format=plain。plain格式把输出写成平面文件，使用和当前数据目录和表空间相同的布局。当集簇没有额外表空间时，整个数据库将被放在目标目录中。如果集簇包含额外的表空间，主数据目录将被放置在目标目录中，但是所有其他表空间将被放在它们位于服务器上的相同的绝对路径中。tar模式将输出写成目标目录中的 tar 文件。主数据目录将被写入到一个名为base.tar的文件中，并且其他表空间将被以其 OID 命名。生成的tar包，需要用gs\_tar命令解压。
* -X，-xlog-method=fetch|stream：设置xlog传输方式。没有设置该参数的情况下，默认-xlog-method=stream。在备份中包括所需的预写式日志文件(WAL文件）。这包括所有在备份期间产生的预写式日志。fetch方式在备份末尾收集预写式日志文件。因此，有必要把wal\_keep\_segments参数设置得足够高，这样在备份末尾之前日志不会被移除。如果在要传输日志时它已经被轮转，备份将失败并且是不可用的。stream方式在备份被创建时流传送预写式日志。这将开启一个到服务器的第二连接并且在运行备份时并行开始流传输预写式日志。因此，它将使用最多两个由max\_wal\_senders参数配置的连接。只要客户端能保持接收预写式日志，使用这种模式不需要在主控机上保存额外的预写式日志。
* -x，-xlog：使用这个选项等效于和方法fetch一起使用-X。
* -Z ，-compress=level：启用对 tar 文件输出的 gzip 压缩，并且制定压缩级别(0 到 9，0 是不压缩，9 是最佳压缩）。只有使用 tar 格式时压缩才可用，并且会在所有tar文件名后面自动加上后缀.gz。
* -z：启用对 tar 文件输出的 gzip 压缩，使用默认的压缩级别。只有使用 tar 格式时压缩才可用，并且会在所有tar文件名后面自动加上后缀.gz。
* -t，-rw-timeout：设置备份期间checkpoint的时间限制，默认限制时间为120s。当数据库全量checkpoint耗时较长时，可以适当增大rw-timeout限制时间。
* 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/vastbase/data/bak -h 127.0.0.1 -p 5432

**从备份文件恢复数据**

当数据库发生故障时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。

* 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，这时需要修改配置文件的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postgresql.conf中的replconninfo1、replconninfo2等。
* 若配置文件postgresql.conf的参数data\_directory打开且有配置，当使用备份目录启动数据库时候，data\_directory和备份目录不同会导致启动失败。可以修改data\_directory的值为新的数据目录，或者注释掉该参数。

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。

2、将原数据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，需要修改使其链接到正确的文件。

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。

* 暂不支持备份文件增量恢复。
* 恢复后需要检查数据库中的链接文件是否链接到正确的文件。

## vb\_bulkload

**功能描述**

vb\_bulkload就是批量加载工具，可以加载表数据，批量加载可以提升加载的效率和性能。

在使用vb\_bulkload工具导入出错之后，本次导入的数据都要进行回滚，所以需要将本次导入数据清理，防止这些数据被访问，造成查询、执行计划不准确等问题。

**语法格式**

vb\_bulkload -r options

options支持列表如下：

| **Options选项** | **说明** |
| --- | --- |
| -o "DIRS=val"--option="DIRS=val" | val可指定含.loadstatus为后缀的文件目录。 |
| -d dbname--dbname dbname | 要连接的数据库的名称。 |
| -h host--host host | 正在运行服务器的计算机的主机名。 |
| -p port--port port | 服务器侦听连接的TCP端口。 |
| -U username--username username | 连接的用户名。 |
| -W--password | 强制vb\_bulkload在连接到数据库之前提示输入密码。 |
| -e--echo | 显示发送到服务端的命令。 |
| --help | 显示帮助信息。 |
| --version | 显示版本信息。 |

**注意事项**

* 使用此功能需安装pg\_bulkload插件。
* 实现vb\_bulkload导入时，将在导入报错时清理本次导入数据，需要指定-o “ERROR\_RECOVERY=YES”,该参数默认值为NO。
* vb\_bulkload不支持ustore表格式。

**示例：**

1、创建vb\_bulkload插件。

create extension vb\_bulkload;

2、创建目标数据库表。

create table tb\_evt(begintime varchar(32),usernum varchar(32),imei varchar(32),calltype text,netid varchar(32),lai varchar(32),ci varchar(32),imsi varchar(32));  
  
insert into tb\_evt values(20210101133519,17100022821,46003824071220,'JQ',18900005401,960221,94352844,35382407122);

3、退出数据库。

\q

4、执行命令。

vb\_bulkload -o "INFILE=/data/inputdata/file" -O tb\_evt -o "TYPE=CSV" -o "DELIMITER=,"-o "SET\_COMMITTED=YES" -o "FILTER=DATETIME\_FORMAT(1，YYYYMMDDHH24MISS）" -o "ERROR\_RECOVERY=YES" -d vastbase -Ujq229\_7173

5、中途退出ctl+c，查看执行时间。

6、进入数据，查看数据库文件大小。

\dt+

7、查询数据。

select count(\*) from tb\_evt;

8、查看pg\_log日志。

## vb\_config

**功能描述**

vb\_config工具打印当前安装的Vastbase配置参数。 它可以用于那些希望与Vastbase相连接的软件包， 这样可以通过它找到相对应的头文件和库。都一样。

**语法格式**

vb\_config [option...]

详细选项可使用如下命令查看：

vb\_config --help

返回结果为：

vb\_config provides information about the installed version of Vastbase.  
  
Usage:  
 vb\_config [OPTION]...  
  
Options:  
 --bindir show location of user executables  
 --docdir show location of documentation files  
 --htmldir show location of HTML documentation files  
 --includedir show location of C header files of the client  
 interfaces  
 --pkgincludedir show location of other C header files  
 --includedir-server show location of C header files for the server  
 --libdir show location of object code libraries  
 --pkglibdir show location of dynamically loadable modules  
 --localedir show location of locale support files  
 --mandir show location of manual pages  
 --sharedir show location of architecture-independent support files  
 --sysconfdir show location of system-wide configuration files  
 --pgxs show location of extension makefile  
 --configure show options given to "configure" script when  
 Vastbase was built  
 --cc show CC value used when Vastbase was built  
 --cppflags show CPPFLAGS value used when Vastbase was built  
 --cflags show CFLAGS value used when Vastbase was built  
 --cflags\_sl show CFLAGS\_SL value used when Vastbase was built  
 --ldflags show LDFLAGS value used when Vastbase was built  
 --ldflags\_ex show LDFLAGS\_EX value used when Vastbase was built  
 --ldflags\_sl show LDFLAGS\_SL value used when Vastbase was built  
 --libs show LIBS value used when Vastbase was built  
 --version show the Vastbase version  
 -?, --help show this help, then exit  
  
With no arguments, all known items are shown.

**示例**

要重用当前Vastbase安装的配置进行编译，运行下面的命令：

[vastbase@localhost root]$ pg\_config --configure

返回结果为：

'--gcc-version=7.3.0' '--prefix=/home/vastbase/mppdb\_temp\_install' '--3rd=/home/vastbase/binarylibs' 'PYTHON=/home/vastbase/binarylibs/dependency/centos7.6\_x86\_64/python3.8/comm/bin/python3' '--enable-thread-safety' '--with-readline' '--without-zlib' '--with-python' 'CFLAGS=-O2 -g3 ' '--enable-mot' '--with-ossp-uuid' '--enable-nls=zh\_CN' '--with-libxml' '--enable-oracle\_fdw' '--enable-mysql\_fdw' 'CC=g++' '--svn\_input=7387'  
NOTICE: Last successful login info  
login time:2022-08-16 11:57:26  
application name:vb\_isready  
Ip address:[local]  
method:Trust  
  
Failed 0 times since last successful login

## 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vb\_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。

## vb\_ctl

**背景信息**

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vastbase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](#table145081017222)。
* 公共参数，详细请参见[表2](#X2e90ae1b4c0c772adafcd2c4b6d991bd9ee3d46)。
* start和restart模式的参数，详细请参见[表3](#Xf6ea670fc2715927d4f95ec29eb2f1b3b15b553)。
* stop和restart模式的参数，详细请参见[表4](#Xbc45c098f2ea5b4a4674db1cc2705ce4a5539a6)。
* switchover模式的参数，详细请参见[表5](#X03d33d627e02ce57f1f38679f3c3aa7709736ae)。
* build参数，详细请参见[表6](#X9912e4bb24a91a21fbafdbe3db5504f2d6770cf)。
* member参数，详细请参见[表8](#table1055392110383)。
* changerole参数，详细参见[表9](#table591372895218)。
* setrunmode参数，详细参见[表10](#table1451519418810)。

**表 1** 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gs\_om -t refreshconf 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gs\_om -t refreshconf 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stanby 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 2** 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。 | - |
| -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。 |
| -?, --help | 显示关于vb\_ctl命令行参数的帮助信息。 | - |

**表3** 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4** 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： 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5** 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。 默认值：fast |

**表 6** 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。说明：增量重建适用于主备双主等因日志造成的不一致场景。  增量重建不适用于一主一备并且没有开启最大高可用的场景，此种场景下需要使用全量重建或者开启最大高可用后再进行增量重建。  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。  standby\_full：通过指定的备机ip和port全量重建故障备机。使用该参数时需要同时使用-C蚕食指定镜像的ip和port。默认值：auto说明：重建级联备机需要加上-M cascade\_standby参数。 |
| -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秒提示：超时时间需根据业务繁忙程度设置，业务繁忙等待主机的响应时间需加长，建议超时时间=checkpoint的执行时间+1min。 |
| -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，重启默认最大等待时间是60秒。如果在此时间内数据库未能启动成功，就会报不再等待，命令退出。此后可以通过vb\_ctl status 或其他查看数据库状态的命令，来确认数据库是否启动成功。 |  |

**表 7** query参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -L | 查询lsn并展示最大长度。 | 无参数 |

**表 8** member参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| –operation | vastbase数据库实例，DCF模式下，增删节点的操作。 | add： 向数据库实例的DCF节点配置中增加一个节点。remove：从数据库实例DCF节点配置中删除一个节点。 |
| -u, –nodeid | 新增节点的DCF NODE ID。 | 无符号整型 |
| -i, –ip | 新增节点与数据库实例DCF模块通信的IP。 | 字符串。 |
| -e, –port | 新增节点与数据库实例DCF模块通信的port。 | 整型。 |
| [-D, –pgdata=]DATADIR | 指定数据目录的位置。 | DATADIR的取值：用户自定义。 |

**表 9** changerole参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -R, –role | Vastbase数据库实例，DCF模式下节点角色（需要在角色为primary的节点上使用）。 | passive： passive角色。follower：follower角色。 |
| -u, –nodeid | 新增节点的DCF NODE ID。 | 无符号整型。 |
| [-D, –pgdata=]DATADIR | 指定数据目录的位置。 | DATADIR的取值：用户自定义。 |

表 10 setrunmode参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -v, –votenum | 投票个数。 | 无符号整型。 |
| -x, –xmode | 运行模式（在少数派的DN节点上使用）。 | minority：少数派强起。normal：加回。 |
| [-D, –pgdata=]DATADIR | 指定数据目录的位置。 | DATADIR的取值：用户自定义。 |

## 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），回收站对象除外。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vastbase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时生成列不会被转储。

vb\_dump支持导出兼容v1版本数据库的文本格式文件。

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gsql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[导出文件格式](#导出文件格式)。该导出结果必须与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，具体如导出文件格式所示。

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。 |
| 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。 | - |



可以使用vb\_dump程序将文件压缩为目录归档或自定义归档导出文件，减少导出文件的大小。生成目录归档或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。

**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。
* 不支持加密导出存储过程和函数。

**语法格式**

vb\_dump [OPTION]... [DBNAME]



“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，要求当前用户对该目录具有读写权限，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gs\_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gs\_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gsql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。tar归档格式和纯文本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，gs\_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gs\_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，可以先指定任意数据库用于执行创建数据库的命令，数据不会恢复到指定的数据库中，而是恢复到创建的数据库中。）该选项只对文本格式有意义。针对归档格式，可以在调用gs\_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。



* 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\_shl2.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，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\_shl2.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。



* -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\_shl2.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\_shl2.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。



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）。
* -q, --target：指定导出兼容其他版本数据库的文本文件，目前支持v1和v5参数。v1参数用于导出v5数据库的数据为兼容v1的文本文件。v5参数用于导出v5数据库的数据为v5格式的文本文件，减少了导入v5时的可能的报错情况。在使用v1参数时，建议和--exclude-guc="enable\_cluster\_resize"，--exclude-function，--exclude-with等选项共用，否则导入到v1时可能报错。
* --exclude-guc：导出的文本文件中，不包括相关guc参数的set命令，目前只支持enable\_cluster\_resize。
* --exclude-function：不导出函数和存储过程。
* --exclude-with：导出的表定义，末尾不添加WITH(orientation=row，compression=on）这样的描述。
* --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$，并强制使用SQL标准字符串语法对其进行引用。
* --disable-triggers：该参数为扩展预留接口，不建议使用。
* --exclude-table-data=TABLE：指定不转储任何匹配表pattern的表这方面的数据。依照针对-t的相同规则理解该pattern。可多次输入--exclude-table-data来排除匹配任何pattern的表。当用户需要特定表的定义但不需要其中的数据时，这个选项很有帮助。排除数据库中所有表的数据，参见[--schema-only](#schema-only)。
* --inserts：发出INSERT命令(而非COPY命令）转储数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。注意：如果重排列顺序，可能会导致整个恢复失败。列顺序改变时，--column-inserts选项不受影响，虽然会更慢。
* --no-publications：不转储发布。
* --no-security-labels：该参数为扩展预留接口，不建议使用。
* --no-subscriptions：不转储订阅。
* --no-tablespaces：不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。该选项只对文本格式有意义。针对归档格式，可以在调用gs\_restore时指定选项。
* --no-unlogged-table-data：该参数为扩展预留接口，不建议使用。
* --non-lock-table：该参数为扩展预留接口，不建议使用。
* --include-alter-table：转储表删除列。该选项会记录列的删除。
* --quote-all-identifiers：强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。
* --section=SECTION：指定已转储的名称区段(pre-data、data和post-data）。
* --serializable-deferrable：转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致；要实现该操作需要在无异常状况的事务流中等待某个点，因为这样才能保证转储成功，避免引起其他事务出现serialization\_failure要重新再做。但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候，加载一个数据库的拷贝作为报告或其他只读加载共享的转储是有帮助的。没有这个选项，转储会反映一个与任何事务最终提交的序列化执行不一致的状态。如果当vb\_dump启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始时间可能会延迟一段不确定的时间。
* --use-set-session-authorization：输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功，而ALTER OWNER需要的权限则低得多。
* --with-encryption=AES128：指定转储数据需用AES128进行加密。
* --with-key=KEY
* AES128密钥规则如下：
* 密钥长度为8~16个字符。
* [至少包含大写字母(A-Z）、小写字母(a-z）、数字(0-9）、非字母数字字符(限定为~!@#$%^&\*()-\_=+|[{}];:,<.>/?）四类字符中的三类字符。](mailto:至少包含大写字母(A-Z）、小写字母(a-z）、数字(0-9）、非字母数字字符(限定为~!@#$%^&*()-_=+|[{}];:,<.>/?）四类字符中的三类字符。)



* 使用vb\_dump工具进行加密导出时，仅支持plain格式导出。通过-F plain导出的数据，需要通过gsql工具进行导入，且如果以加密方式导入，在通过gsql导入时，需要指定--with-key参数。
* 不支持加密导出存储过程和函数。
* --with-salt=RANDVALUES：vb\_dumpall使用此参数传递随机值。
* --include-extensions：在转储中包含扩展。
* --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。
* --pipeline：使用管道传输密码，禁止在终端使用。
* --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



* -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；“MPPDB\_backup.sql”表示导出的文件；“5432”表示数据库服务器端口；“postgres”表示要访问的数据库名。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 vastbase -W Bigdata@123 -f MPPDB\_backup.sql -p 5432 postgres -F p  
vb\_dump[port='5432'][postgres][2022-08-03 11:45:15]: The total objects number is 440.  
vb\_dump[port='5432'][postgres][2022-08-03 11:45:15]: [100.00%] 440 objects have been dumped.  
vb\_dump[port='5432'][postgres][2022-08-03 11:45:15]: dump database postgres successfully  
vb\_dump[port='5432'][postgres][2022-08-03 11:45:15]: total time: 941 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 vastbase -W Bigdata@123 -f MPPDB\_backup.tar -p 5432 postgres -F t  
vb\_dump[port='5432'][postgres][2022-08-03 11:47:37]: The total objects number is 440.  
vb\_dump[port='5432'][postgres][2022-08-03 11:47:37]: [100.00%] 440 objects have been dumped.  
vb\_dump[port='5432'][postgres][2022-08-03 11:47:37]: dump database postgres successfully  
vb\_dump[port='5432'][postgres][2022-08-03 11:47:37]: total time: 896 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 vastbase -W Bigdata@123 -f MPPDB\_backup.dmp -p 5432 postgres -F c  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:13]: The total objects number is 440.  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:13]: [100.00%] 440 objects have been dumped.  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:13]: dump database postgres successfully  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:13]: total time: 899 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 vastbase -W Bigdata@123 -f MPPDB\_backup -p 5432 postgres -F d  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:51]: The total objects number is 440.  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:51]: [100.00%] 440 objects have been dumped.  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:51]: dump database postgres successfully  
vb\_dump[port='5432'][postgres][2022-08-03 11:48:51]: total time: 893 ms

**示例5：**执行vb\_dump，导出postgres数据库信息，但不导出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 vastbase -W Bigdata@123 -p 5432 postgres --exclude-table-file=MPPDB\_temp.sql -f backup/MPPDB\_backup.sql  
vb\_dump[port='5432'][postgres][2022-08-03 11:50:50]: The total objects number is 1367.  
vb\_dump[port='5432'][postgres][2022-08-03 11:50:50]: [100.00%] 1367 objects have been dumped.  
vb\_dump[port='5432'][postgres][2022-08-03 11:50:50]: dump database postgres successfully  
vb\_dump[port='5432'][postgres][2022-08-03 11:50:50]: 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。

前置步骤：进入数据库创建测试表testtable。

c postgres  
create table testtable(id int);

* 备份仅依赖于testtable的视图。

vb\_dump -s -p 5432 postgres -t PUBLIC.testtable --include-depend-objs --exclude-self -f backup/MPPDB\_backup.sql -F p  
vb\_dump[port='5432'][vastbase][2022-08-03 11:58:23]: The total objects number is 432.  
vb\_dump[port='5432'][vastbase][2022-08-03 11:58:23]: [100.00%] 432 objects have been dumped.  
vb\_dump[port='5432'][vastbase][2022-08-03 11:58:23]: dump database vastbase successfully  
vb\_dump[port='5432'][vastbase][2022-08-03 11:58:23]: total time: 1014 ms

* 修改testtable名称。

vsql -p 5432 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。

vsql -p 5432 postgres -r -f MPPDB\_backup.sql

## 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vastbase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时生成列不会被转储。

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的全部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gsql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。

**语法格式**

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$，并强制使用SQL标准字符串语法对其进行引用。
* --disable-triggers：该参数为扩展预留接口，不建议使用。
* --inserts：发出INSERT命令(而非COPY命令）转储数据。这会导致恢复缓慢。注意：如果重排列顺序，可能会导致恢复整个失败。--column-inserts选项更加安全，虽然可能更慢些。
* --no-security-labels：该参数为扩展预留接口，不建议使用。
* --no-tablespaces：请勿输出创建表空间的命令，也请勿针对对象选择表空间。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。
* --no-publications：不转储发布。
* --no-security-labels：该参数为扩展预留接口，不建议使用。
* --no-subscriptions：不转储订阅。
* --include-alter-table：导出表中已删除的列信息。
* --quote-all-identifiers：强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。
* --dont-overwrite-file：不重写当前文件。
* --use-set-session-authorization：输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功，而ALTER OWNER需要的权限则低得多。
* --with-encryption=AES128：指定转储数据需用AES128进行加密。
* --with-key=KEY
* AES128密钥规则如下：
* 密钥长度为8~16个字符。
* 至少包含大写字母(A-Z），小写字母(a-z），数字(0-9），非字母数字字符(限定为~!@#$%^&\*()-\_=+|[{}];:,<.>/?）四类字符中的三类字符。
* --include-extensions：如果include-extensions参数被设置，将备份所有的CREATE EXTENSION语句。
* --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。
* --pipeline：使用管道传输密码，禁止在终端使用。



* -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：指定服务器所侦听的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，即侦听端口+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。



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 5432  
vb\_dump[port='5432'][dbname='vastbase'][2022-08-03 12:09:30]: The total objects number is 446.  
vb\_dump[port='5432'][dbname='vastbase'][2022-08-03 12:09:30]: [100.00%] 446 objects have been dumped.  
vb\_dump[port='5432'][dbname='vastbase'][2022-08-03 12:09:30]: dump database dbname='vastbase' successfully  
vb\_dump[port='5432'][dbname='vastbase'][2022-08-03 12:09:30]: total time: 1030 ms  
gs\_dumpall[port='5432'][2022-08-03 12:09:30]: dumpall operation successful  
gs\_dumpall[port='5432'][2022-08-03 12:09:30]: total time: 2100 ms

## 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。

**语法**

* 显示帮助信息。

vb\_guc -? | --help

* 显示版本号信息。

vb\_guc -V | --version

* 检查配置文件中参数。

vb\_guc check [-Z NODE-TYPE] [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} {-c "parameter", -c "parameter", ...}

或者

vb\_guc check [-Z NODE-TYPE] [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} {-c

* 修改配置文件中参数。

vb\_guc {set | reload} [-Z NODE-TYPE] [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} [--lcname=LCNAME] [--ignore-node=NODES] {-c "parameter = value" -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。若选择reload会同时发送信号量到pg\_hba.conf，即无需重启即可生效。

vb\_guc {set | reload} -Z NODE-TYPE [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE DATABASE USERNAME IPADDR IPMASK AUTHMEHOD authentication-options"  
   
 vb\_guc {set | reload} -Z NODE-TYPE [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK AUTHMEHOD authentication-options"  
   
 vb\_guc {set | reload} -Z NODE-TYPE [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE DATABASE USERNAME HOSTNAME AUTHMEHOD authentication-options"



* AUTHMEHOD后的authentication-options为可选参数，AUTHMEHOD支持以下选项：
* trust：不验密，禁止远程主机使用trust方式访问数据库。
* reject：拒绝访问。
* md5：md5认证，默认不支持（MD5加密算法安全性低，存在安全风险，不建议使用）。
* sha256：sha256认证（推荐使用）。
* cert：客户端证书认证。
* gss：kerberos认证，仅用于内部节点间认证。
* sm3：sm3认证（国密SM3）。
* pg\_hba.conf中的认证策略越靠前优先级越高，使用vb\_guc工具配置时会按一定规则排序将新策略插入到原有认证策略中。配置字段比较顺序为：IPADDR/HOSTNAME > HOSTTYPE > DATABASE > USERNAME，即优先比较IPADDR或HOSTNAME，如果无法区分优先级则继续比较HOSTTYPE，以此类推。对于每个配置字段，通常越严格的配置参数优先级越高、排序越靠前，越宽松的配置参数优先级越低、排序越靠后，具体如下：
* IPADDR：当配置为全0时表示不限制IP，会放在指定具体某个IP地址的策略后面。
* DATABASE：当配置为all时表示不限制数据库，会放在指定具体某个数据库的策略后面；当数据库配置为replication时会放在其他策略后面。
* USERNAME：当配置为all时表示不限制用户，会放在指定具体某个用户的策略后面。
* HOSTTYPE：local > hostssl > hostnossl > host。
* 注释已经设置的客户端认证策略。若选择reload会同时发送信号量到pg\_hba.conf，即无需重启即可生效。

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

* vb\_guc encrypt生成加密密码和加密因子文件。

vb\_guc encrypt [-M keymode] -K password [-U username] {-D DATADIR | -R RANDFILEDIR -C CIPHERFILEDIR}



* -K是用户指定的密码，vb\_guc会对该密码进行长度（8<=len<=16）和密码复杂度要求，如果不满足，将会报错。此密码用于保证生成密码文件的安全性和唯一性，用户无需保存或记忆。
* -M是加密类型，当前仅支持server、client和source。默认值为server。
* vb\_guc generate生成其他前缀的加密密码和加密因子文件。

vb\_guc generate [-o prefix] -S cipherkey -D DATADIR



* -o是输出cipher和rand文件前缀名称，默认输出文件名前缀为obsserver。其内容仅支持数字、字母和下划线。
* -S是用户指定的密码，密码需要满足长度要求（8<=len<=16）和复杂度要求，如不满足将会报错。当其值为default时，会随机生成一段字符串作为密码，该密码长度为13。如果不带-S参数则会提示交互式输入密码。为了系统安全，推荐使用交互式输入密码方式。
* 使用vb\_guc encrypt或generate命令生成加密密码和加密因子文件时只是参数不同，本质上是一样的。生成过程中会使用随机数作为加密密钥材料和盐值，因此是每次生成的文件都是不同的。每次生成的加密密码和加密因子文件需要成对使用，不能更换或交替使用，加密和解密时需要使用相同的加密密码和加密因子文件。
* 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/gaussdb/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。

  
需依赖静态文件，编译安装由于没有静态文件，所以无效。

* -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。

  
需依赖静态文件，编译安装由于没有静态文件，所以无效。

* -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。



与“-I” 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。

取值范围：postgresql.conf中的所有参数。



* 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'"。
* 如果需要配置的value内容中包含双引号, 则需要在双引号前加上转义符。例如value为a"b"c, 则命令为-c " parameter = 'a\"b\"c' "
* 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。
* --lcname=LCNAME：要设定的逻辑数据库名称。

取值范围：已经创建的逻辑数据库名称。



* 该参数必需同-Z datanode一起使用。即vb\_guc只允许作用于逻辑数据库的DN实例。
* 逻辑数据库允许操作的参数同完整数据库不同。具体差异可参见$GAUSSHOME/bin/cluster\_guc.conf。
* --ignore-node=NODES：需要忽略的主机名称。



* 该参数必须与set/reload一起使用，且-Z只支持datanode。
* 该参数不支持与-D一起使用。
* 在与reload一起使用时，如果--ignore-node没有指定主节点，则集群中所有节点的参数依然会全部同步修改。
* -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，参数值不允许为空，为空时不会恢复为数据库参数的默认值。
* -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 [authmehod-options]
* HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK [authmehod-options ]
* HOSTTYPE DATABASE USERNAME HOSTNAME [authmehod-options ]
* HOSTTYPE是必选参数，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。
* authmehod-options是可选输入，取值：
* trust
* reject
* md5
* sha256
* sm3
* cert
* gss
* 详细的参数说明请参见“pg\_hba.conf”配置文件中的描述。

  
指定-h的情况下，默认会尝试修改所有节点的pg\_hba文件，但是如果是编译安装，由于没有静态文件，可以成功修改本节点的pg\_hba文件，但是会提示静态文件不存在。

* -?, --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。

  
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。



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。
* -o PREFIX：指定需要输出的OBS类型的cipher和rand文件前缀名称字符串，默认输出文件名前缀为obsserver。
* 取值范围：仅支持数字、字母和下划线。
* -Z NODE-TYPE：指定数据库实例节点类型，参数默认值为datanode。NODE-TYPE用于识别配置文件。
* 取值范围：datanode。

  
数据库实例节点类型有coordinator、datanode、cmserver、cmagent和gtm。vastbase只能取值为datanode，当NODE-TYPE为datanode时，对应数据库的数据目录中的配置文件postgresql.conf。

**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 reload -D $PGDATA -c "max\_connections = 800"

**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 reload -D $PGDATA -c "max\_connections"

**示例3：**设置客户端认证策略。

vb\_guc set -h "host replication testuser 10.252.95.191/32 sha256"

**示例4：**注释清理已经设置的客户端认证策略。

vb\_guc set -h "host replication testuser 10.252.95.191/32"

**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 Gauss@123 -D /data/xzg/wisequery/gaussdb  
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 /gaussdb/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

## vb\_initdb

**功能描述**

vb\_initdb用于初始化数据库，在初始化数据库时，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。



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。

* 创建数据库
* 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提供的最初始的备份数据库，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。

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。

**语法格式**

vb\_initdb [OPTION]... [DATADIR]

**参数说明**

常用参数如下：

* -?，--help

显示vb\_initdb命令的帮助信息，然后退出。

* -V，--version

输出vb\_initdb命令的版本信息，然后退出。

* -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。

取值范围：trust、reject、md5（不安全的算法，为了兼容老版本而存在）、sha256、sm3

默认值：trust（除非用户对本地用户都是信任的，否则不要使用默认值trust）



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。vs\_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）。

取值范围：trust、reject、md5（不安全的算法，为了兼容老版本而存在）、sha256、sm3

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）。

取值范围：trust、reject、md5（不安全的算法，为了兼容老版本而存在）、sha256、sm3、peer（仅用于local模式）

默认值：trust

* [-D, --pgdata=]DATADIR

指定数据目录的位置。

取值范围： DATADIR的取值：用户自定义。不能包括“|”, “;”，“&”，“$”，“<”，“>”，“`”，“\”，“!”这几个字符。

* --nodename=NODENAME

初始化的节点名称。

节点的命名需要遵守如下规范：

* 节点名称必须为小写字母（a-z）、下划线（\_）、特殊符号#、数字（0-9）。
* 节点名称必须以小写字母（a-z）或下划线（\_）开头。
* 节点名称不能为空，且最大的长度为64个字符 。
* -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，则使用系统默认区域的编码格式。系统默认区域和编码格式可以使用locale如下命令查看：locale|grep LC\_CTYPE
* 不能包括“|”, “;”，“&”，“$”，“<”，“>”，“`”，“\”，“!”这几个字符。
* --locale=LOCALE

为新数据库设置缺省的区域（可以用locale -a查看可用的区域），如果不希望指定特定的区域，则可以用C。



* 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。
* 不能包括“|”, “;”，“&”，“$”，“<”，“>”，“`”，“\”，“!”这几个字符。

示例：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、查看系统支持gbk编码的区域。

locale -a|grep gbk  
zh\_CN.gbk  
zh\_SG.gbk

2、初始化数据库时加入--locale=zh\_CN.gbk选项

* --dbcompatibility=DBCOMPATIBILITY

指定兼容的数据库的类型。

取值范围：取值范围：A、B、C、PG。分别表示兼容O、MY、TD和POSTGRES

* --no-locale

和--locale=C等价。

* --pwfile=FILE

vb\_initdb从文件中读取数据库超级用户的密码。该文件的第一行被作为密码使用。



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。不能包括“|”, “;”，“&”，“$”，“<”，“>”，“`”，“\”，“!”这几个字符。

* -T, --text-search-config=CFG

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。

取值范围：english（全文搜索）、simple（普通文本搜索）

默认值：simple

* -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



不能包括“|”, “;”，“&”，“$”，“<”，“>”，“`”，“\”，“!”这几个字符。

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。
* -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。不能包括“|”, “;”，“&”，“$”，“<”，“>”，

“`”，“\”，“!”这几个字符。

* -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。

**使用方法**

执行如下命令初始化数据库。

vb\_initdb -D /home/vastbase/data/vastbase --nodename vastbase -w vbase@123

  
./home/vastbase/data/vastbase为数据库目录，可自定义，需存在且在vastbase用户下，需要0700权限。详细请参见《快速入门->非实例化数据库安装->[初始化数据库运行环境](../%E5%BF%AB%E9%80%9F%E5%85%A5%E9%97%A8/%E9%9D%9E%E5%AE%9E%E4%BE%8B%E5%8C%96%E6%95%B0%E6%8D%AE%E5%BA%93%E5%AE%89%E8%A3%85.md" \l "chushihua)》。

## vb\_isready

**功能描述**

vb\_isready是检查Vastbase数据库服务器连接状态的一个实用工具。退出状态说明连接检查的结果。

可以使用如下命令显示关于vb\_isready命令行参数的帮助：

vb\_isready --help

返回结果为：

vb\_isready issues a connection check to a Vastbase database.  
  
Usage:  
 vb\_isready [OPTION]...  
  
Options:  
 -d, --dbname=DBNAME database name  
 -q, --quiet run quietly  
 -V, --version output version information, then exit  
 -?, --help show this help, then exit  
  
Connection options:  
 -h, --host=HOSTNAME database server host or socket directory  
 -p, --port=PORT database server port  
 -t, --timeout=SECS seconds to wait when attempting connection, 0 disables (default: 3)  
 -U, --username=USERNAME user name to connect as  
 -m, --maintenance use maintenance mode to access database when hot\_standby=off

**退出状态**

如果服务器正常接受连接，那么pg\_isready返回0 到shell；如果服务器拒绝连接（例如在启动期间），那么返回1； 如果连接尝试没有响应，那么返回2；如果没有做任何尝试 （例如由于无效的参数）则返回3。

**示例**

标准用法:

[vastbase@localhost root]$ vb\_isready  
/tmp:5432 - accepting connections  
[vastbase@localhost root]$echo $?  
0

使用连接参数到一个正在启动的Vastabse集群:

[vastbase@localhost root]$ vb\_isready -h localhost -p 5433  
localhost:5433 - rejecting connections  
[vastbase@localhost root]$ echo $?  
1

使用连接参数到一个非响应的集群时：

[vastbase@localhost root]$ vb\_isready -h someremotehost  
someremotehost:5432 - no response  
[vastbase@localhost root]$ echo $?  
2

## vb\_licensetool

**功能描述**

该工具用于查看license的相关信息。

**语法格式**

vb\_licensetool [OPTION]...

**参数说明**

* **--dump＝＜path＞**

显示正式license的详细信息。

* **--view-temporary**

显示自动生成临时license的详细的信息。

* **--help**

显示 vb\_licensetool命令参数帮助，然后退出。

**示例**

vb\_licensetool --view-temporary

返回临时license的信息：

license info: Customer:'temporary license', Begins On:'2022-10-13 10:15:07', Expires On:'2023-01-11 10:15:07', MAC:''

## vb\_probackup

**功能描述**

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，也可对主机或者主节点数据库备机进行备份，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。
* 若要使用PTRACK增量备份，需在postgresql.conf中手动添加参数“enable\_cbm\_tracking = on”。
* 为了防止xlog在传输结束前被清理，请适当调高postgresql.conf文件中wal\_keep\_segements的值。

**注意事项**

* 备份必须由运行数据库服务器的用户执行。
* 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。
* 使用restore子命令前，应先停止Vastbase进程。
* 当存在用户自定义表空间时，备份的时候要加上 --external-dirs 参数，否则，该表空间不会被备份。
* 当备份的规模比较大时，为了防止备份过程中timeout发生，请适当调整postgresql.conf文件的参数 session\_timeout、wal\_sender\_timeout。并且在备份的命令行参数中适当调整参数--rw-timeout的值。
* 恢复时，使用-T选项把备份中的外部目录重定向到新目录时，请同时指定参数--external-mapping。
* 当使用远程备份时，请确保远程机器和备份机器的时钟同步，以防止使用--recovery-target-time恢复的场合,启动gaussdb时有可能会失败。
* 当远程备份有效时(remote-proto=ssh)，请确保-h和--remote-host指定的是同一台机器。当远程备份无效时，如果指定了-h选项，请确保-h指定的是本机地址或本机主机名。
* 当前暂不支持备份逻辑复制槽。

**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]

* init：初始化备份路径backup-path中的备份目录，该目录将存储已备份的内容。backup-path必须为空目录。

vb\_probackup init -B backup-path [--help]

* add\_instance：在备份路径backup-path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pgdata-path的vb\_probackup设置。

vb\_probackup add-instance -B backup-path -D pgdata-path --instance=instance\_name  
[-E external-directories-paths]  
[remote\_options]  
[--help]

* del-instance：在备份路径backup-path内删除指定实例相关的备份内容。

vb\_probackup del-instance -B backup-path --instance=instance\_name  
[--help]

* set-config：将指定的连接、压缩、日志等相关设置添加到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-path --instance=instance\_name  
[-D pgdata-path] [-E external-directories-paths] [--archive-timeout=timeout]  
[--retention-redundancy=retention-redundancy] [--retention-window=retention-window] [--wal-depth=wal-depth]  
[--compress-algorithm=compress-algorithm] [--compress-level=compress-level]  
[-d dbname] [-h hostname] [-p port] [-U username]  
[logging\_options] [remote\_options]  
[--help]

* set-backup：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-path --instance=instance\_name -i backup-id  
[--note=text] [pinning\_options]  
[--help]

* show-config：显示位于备份目录中的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-path --instance=instance\_name  
[--format=plain|json]  
[--help]

* show：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-path  
[--instance=instance\_name [-i backup-id]] [--archive] [--format=plain|json]  
[--help]

* backup：创建指定实例的备份。

vb\_probackup backup -B backup-path --instance=instance\_name -b backup-mode  
[-D pgdata-path] [-C] [-S slot-name] [--temp-slot] [--backup-pg-log] [-j threads\_num] [--progress]  
[--no-validate] [--skip-block-validation] [-E external-directories-paths] [--no-sync] [--note=text]  
[--archive-timeout=timeout] [-t rwtimeout]  
[logging\_options] [retention\_options] [compression\_options]  
[connection\_options] [remote\_options] [pinning\_options]  
[--help]

* restore：从备份目录backup-path中的备份副本恢复指定实例。如果指定了恢复目标选项，vb\_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-path --instance=instance\_name  
[-D pgdata-path] [-i backup\_id] [-j threads\_num] [--progress] [--force] [--no-sync] [--no-validate] [--skip-block-validation]  
[--external-mapping=OLDDIR=NEWDIR] [-T OLDDIR=NEWDIR] [--skip-external-dirs] [-I incremental\_mode]  
[recovery\_options] [remote\_options] [logging\_options]  
[--help]

* merge：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-path --instance=instance\_name -i backup\_id  
[-j threads\_num] [--progress] [logging\_options]  
[--help]

* delete：删除指定备份，或删除不满足当前保留策略的备份。

vb\_probackup delete -B backup-path --instance=instance\_name  
[-i backup-id | --delete-expired | --merge-expired | --status=backup\_status]  
[--delete-wal] [-j threads\_num] [--progress]  
[--retention-redundancy=retention-redundancy] [--retention-window=retention-window]  
[--wal-depth=wal-depth] [--dry-run]  
[logging\_options]  
[--help]

* validate：验证恢复数据库所需的所有文件是否存在且未损坏。如果未指定instance\_name，vb\_probackup将验证备份目录中的所有可用备份。如果指定instance\_name而不指定任何附加选项，vb\_probackup将验证此备份实例的所有可用备份。如果指定了instance\_name并且指定backup-id或恢复目标相关选项，vb\_probackup将检查是否可以使用这些选项恢复数据库。

vb\_probackup validate -B backup-path  
[--instance=instance\_name] [-i backup-id]  
[-j threads\_num] [--progress] [--skip-block-validation]  
[--recovery-target-time=time | --recovery-target-xid=xid | --recovery-target-lsn=lsn | --recovery-target-name=target-name]  
[--recovery-target-inclusive=boolean]  
[logging\_options]  
[--help]

* archive-push：将WAL文件复制到备份目录的相应子目录中。

vb\_probackup archive-push -B backup\_dir --instance=instance\_name   
--wal-file-name=wal-file-name  
[-j num-threads][--batch-size=batch\_size]  
[--archive-timeout=timeout]  
[--no-ready-rename][--no-sync][--overwrite][--compress]  
[--compress-algorithm=compression-algorithm] [--compress-level=compression\_level]  
[--remote-proto][--remote-host]  
[--remote-port][--remote-path][--remote-user]  
[--ssh-options]  
[--help]

**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、validate。
* -?, --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-path, --backup-path=backup-path：备份的路径。
* 系统环境变量：$BACKUP\_PATH
* -D pgdata-path, --pgdata=pgdata-path：数据目录的路径。
* 系统环境变量：$PGDATA
* -instance=instance\_name：实例名。
* -i backup-id, --backup-id=backup-id：备份的唯一标识。
* --format=format：指定显示备份信息的格式，支持plain和json格式。
* 默认值：plain
* --status=backup\_status：删除指定状态的所有备份，包含以下状态：
* OK：备份已完成且有效。
* DONE：备份已完成但未经过验证。
* RUNNING：备份正在进行中。
* MERGING：备份正在合并中。
* DELETING：备份正在删除中。
* CORRUPT：部分备份文件已损坏。
* ERROR：由于意外错误，备份失败。
* ORPHAN：由于其父备份之一已损坏或丢失，备份无效。
* -j threads\_num, --threads=threads\_num：设置备份、还原、合并进程的并行线程数。
* --archive：显示WAL归档信息。
* --progress：显示进度。
* --note=text：给备份添加note。

**备份相关参数**

* -b backup-mode, --backup-mode=backup-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。
* -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。
* --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。
* 默认的slot名为pg\_probackup\_slot，可通过选项--slot/-S更改。
* --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。默认情况下包含日志目录，但不包含日志文件。如果修改了默认的日志路径，需要备份日志文件时可使用-E参数进行备份，使用方法见下文。
* -E external-directories-paths, --external-dirs=external-directories-paths：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。例如：-E /tmp/dir1:/tmp/dir2
* --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。
* --note：给备份添加note。
* --archive-timeout=timeout：以秒为单位设置流式处理的超时时间。
* 默认值：300
* -t rwtimeout：以秒为单位的连接的超时时间。
* 默认值: 120

**恢复相关参数**

* -I, --incremental-mode=none|checksum|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
* --external-mapping=OLDDIR=NEWDIR：在恢复时，将包含在备份中的外部目录从*OLDDIR*重新定位到*NEWDIR*目录。*OLDDIR*和*NEWDIR*都必须是绝对路径。如果路径中包含“=”，则使用反斜杠转义。此选项可为多个目录多次指定。
* -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。当-T参数值为外部表空间目录时，必须搭配--external-mapping参数使用。
* --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。
* --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。
* --force-overwrite ：指定非空目录进行恢复。

**恢复目标相关参数**

* --recovery-target-lsn=lsn：指定要恢复到的lsn，当前只能指定备份的stop lsn。
* --recovery-target-name=target-name：指定要将数据恢复到的已命名的保存点，保存点可以通过查看备份中recovery-name字段得到。
* --recovery-target-time=time：指定要恢复到的时间，当前只能指定备份中的recovery-time。
* --recovery-target-xid=xid：指定要恢复到的事务ID，当前只能指定备份中的recovery-xid。
* --recovery-target-inclusive=overwriteboolean：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=latest：恢复未归档xlog。当数据库进程异常杀掉后导致xlog未归档时，指定latest自动恢复未归档的xlog。



当前不支持配置连续的WAL归档的PITR，因而使用这些参数会有一定限制，具体如下描述。 如果需要使用持续归档的WAL日志进行PITR恢复，请按照下面描述的步骤：

1、将物理备份的文件替换目标数据库目录。

2、删除数据库目录下pg\_xlog中的所有文件。

3、将归档的WAL日志文件复制到pg\_xlog文件中(此步骤可以省略，通过配置recovery.conf恢复命令文件中的restore\_command项替代）。

4、在数据库目录下创建恢复命令文件recovery.conf，指定数据库恢复的程度。

5、启动数据库。

6、连接数据库，查看是否恢复到希望预期的状态。若已经恢复到预期状态，通过pg\_xlog\_replay\_resume()指令使主节点对外提供服务。

**留存相关参数**

* --retention-redundancy=retention-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=retention-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。



可以和backup和delete命令一起使用这些参数。

**固定备份相关参数**

* --ttl=interval：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'



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。

**日志相关参数**

* --log-level-console=log-level-console：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-file：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。
* 默认值：off
* --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：$BACKUP\_PATH/log
* --log-rotation-size=log-rotation-size：指定单个日志文件的最大大小。如果达到此值，则启动vb\_probackup命令后，日志文件将循环，但help和version命令除外。0表示禁用基于文件大小的循环。支持的单位：KB、MB、GB、TB(默认为KB）。
* 默认值：0
* --log-rotation-age=log-rotation-age：单个日志文件的最大生命周期。如果达到此值，则启动vb\_probackup命令后，日志文件将循环，但help和version命令$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



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

**连接相关参数**

* -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：$PGDATABASE
* -h hostname, --pghost=hostname：指定运行服务器的系统的主机名。如果该值以斜杠开头，则被用作到Unix域套接字的路径。
* 系统环境变量：$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, -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，则会提示用户输入密码。



可以和backup命令一起使用这些参数。

**压缩相关参数**

* --compress-algorithm=compress-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。
* 当--compress-algorithm=zlib时，--compress-level参数取值不能为0。
* 线程备份不支持pglz压缩，即当--compress-algorithm=pglz时，不能使用-j参数，否则会报错。

默认值：none

* --compress-level=compress-level：指定压缩级别。该参数必须搭配--compress-algorithm一起使用，否则会报错。
* 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。

取值范围：0~9

默认值：1

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



可以和backup命令一起使用这些参数。

**远程模式相关参数**

* --remote-proto=protocol：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。
* 默认值：当前路径
* --remote-libpath=libpath：指定vb\_probackup在远程系统安装的lib库目录。
* --ssh-options=ssh\_options：指定SSH命令行参数的字符串。例如：--ssh-options='-c cipher\_spec -F configfile'

**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

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 -D pgdata-path -i backup\_id

image2

* 如果因为网络临时故障等原因导致server端无应答,vb\_probackup将在等待archive->timeout(默认300秒）后退出。
* 如果备机lsn与主机有差别时，数据库会不停地刷以下log信息，此时应重新build备机。

LOG: walsender thread shut down  
LOG: walsender thread started  
LOG: received wal replication command: IDENTIFY\_VERSION  
LOG: received wal replication command: IDENTIFY\_MODE  
LOG: received wal replication command: IDENTIFY\_SYSTEM  
LOG: received wal replication command: IDENTIFY\_CONSISTENCE 0/D0002D8  
LOG: remote request lsn/crc: [xxxxx] local max lsn/crc: [xxxxx]

## 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vastabse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。导入时生成列会自动更新，并像普通列一样保存。
* 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



* 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。若恢复表数据时，数据量很大，会分批恢复，因此会多次出现“表数据已完成导入”的日志。

**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。



* -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。
* -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gs\_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。

**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 ：导入数据库之前会先使用CREATE DATABASE创建数据库。(指定该选项后，-d指定的数据库仅用以执行CREATE DATABASE命令，所有数据依然会导入到创建的数据库中。）
* -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，也可使用分号(;）在行的开始注出。
* -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 -t 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test1

vb\_restore -h host\_name -p port\_number -d postgres -n PUBLIC -t table1 -n test1 -t table1 backup/MPPDB\_backup.tar



* -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-publications ：不导入发布。
* --no-security-labels ：该参数为扩展预留接口，不建议使用。
* --no-subscriptions ：不导入订阅。
* --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。

image2

* --pipeline ：使用管道传输密码，禁止在终端使用。
* 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。



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。

## VPatch

操作前请做好原始产品和原始用户数据的备份！

**功能描述**

VPatch是用于Vastbase版本控制的工具可以实现在单机环境下的升级和回退操作，具体功能如下：

* 升级环境检查，补丁冲突检查等。
* PSU、PSR补丁的升级、回退及升级或回退失败时的还原。
* 升级过程中记录相关日志，步骤文件。
* 升级过程中管理备份文件。

**前提条件**

需在python3.7环境下运行，即执行python3 --version，返回版本为3.7.\* 。

**语法格式**

* 配置文件如下(ini格式)：

[vpatch]  
userName=用户名user1  
userGroup=用户组group1  
appPath=产品软件目录  
dataPath=集簇的绝对路径  
port=端口  
tmpPath=升级过程临时文件目录

* 执行升级。

vpatch --upgrade

* 执行回退，即回退掉升级记录中对应操作记录号(patch\_record\_id)所升级的补丁，操作记录号被记录在升级历史信息中。

vpatch --rollback=patch\_record\_id

* 执行连续回退。倒叙遍历升级历史信息，逐个回退升级补丁，直到回退了指定操作记录号的补丁为止。注意此处指定的升级操作记录号(patch\_record\_id)也在回退范围内。

vpatch --rollbackTo=patch\_record\_id

* 查看vpatch版本信息。

vpatch -V | --version

* 查看vpatch帮助信息。

vpatch -? | --help

* -I|--information：显示vpatch同目录下升级补丁的信息，在rollback时显示回退的升级补丁的信息，在rollback to时显示最后一个被回退的升级补丁的信息。

vpatch -I | --information

* -U|--upgradeInformation：显示升级历史信息，若补丁已被回退，则相关信息不在此处显示。此信息在用户需要指定回退时尤为重要。需要回退时可以参考升级历史信息中提供的升级操作号(patch\_record\_id)，及升级补丁的相关信息。

vpatch -U [-s] | --upgradeInformation [--simple]

完整显示升级历史信息。

vpatch -U [-s] | --upgradeInformation --complete

* 为vpatch工具设定参数(root用户执行)。

1、命令行直接指定参数。

vpatch --setparam -u 用户名 -g 用户组 -a 产品目录 -d 集簇目录 -p 端口号 -t 临时目录

2、给定ini配置文件。

vpatch --setparam -f ini配置文件绝对路径

* 对升级或回退补丁做基本检查(root用户执行)。

vpatch -C | --check [ --rollbackId=patch\_record\_id | --rollbacktoId=patch\_record\_id]

* 为升级或回退做准备工作(root用户执行)。

vpatch -P | --prepare [ --rollbackId=patch\_record\_id]|--rollbackToId=patch\_record\_id]

* 连续回退。

vpatch --rollbackTo=patch\_record\_id

**注意事项**

* 对于集群升级，需还原为单机模式进行升级，升级后重新部署集群。
* 升级时指定的端口号应避开日常使用的服务端口号，防止升级过程被外部干扰。
* 在升级或回退前应关闭数据库服务。
* 不支持修改用户数据(vb\_initdb初始化后并未包含的由用户操作产生的表) 升级的场景。
* 不支持修改过数据字典的场景进行rollback 和rollback to回退操作。
* 仅支持未修改系统表且允许回退的补丁进行rollback 和rollback to回退操作。
* 不支持多实例场景，多实例即为单一产品对应多个数据库集簇的场景。
* 升级过程中，check和prepare阶段会向数据库初始化用户的环境配置文件中写入环境变量，所以切换用户时若使用了不同的终端，需要执行以下命令避免出现环境变量缺失的问题。

source ~/.bashrc

* 仅支持Vastbase G100 V2.2(build 10)及以上版本的升级。

**参数说明**

**表1**补丁信息

| **参数** | **说明** |
| --- | --- |
| vbVpatchInfo | 供vpatch使用的信息列表。 |
| patchNum | 补丁编号。 |
| patchFixedBugList | 修补bug列表。 |
| patchType | 补丁类型。 |
| patchReleaseDate | 补丁发布日期。 |
| compatibleProductVersionNum | 保留参数。 |
| patchConflictList | 保留参数。 |
| patchIncludeList | 保留参数。 |
| trunkKernelVersion | 保留参数。 |
| isUpdateCatalog | 是否修改系统表。 |
| isUpdateOnlyFunc | 修改系统表，但仅限于函数。 |
| allowRollback | 补丁是否允许回退，true可以回退，false不可回退。 |
| base\_product\_version | 适配的基础版本。 |
| baseSvn | 原始版本的SVN提交号。 |
| newSvn | 新版本的SVN提交号。 |
| isFallbackPatch | 用于升级还是回退的补丁，true升级，false回退。 |
| osInfo | 适用的操作系统信息。 |
| cpuInfo | 适用的cpu信息。 |
| deleteFile | 此次升级删除的文件，包括数据库和om工具。 |

**表2**历史信息参数说明

|  |  |
| --- | --- |
| **参数** | **说明** |
| auditId | 审计号。 |
| patch\_record\_id | 操作序列号。 |
| patchNum | 补丁编号。 |
| date | 升级操作的日期和时间。 |
| opType | 操作类型（One off patch，升级PSU,升级PSR）。 |
| commitId | svn提交号。 |
| fixedBugs | 修复的bug列表。 |
| isUpdateCatalog | 是否修改系统表。 |
| isUpdateOnlyFunc | 是否修改了函数。 |
| isRollback | 是否已被回退。 |
| upgradeBackupPath | 升级备份的目录。 |
| patchConflictList | 保留参数。 |
| patchInlcudeList | 保留参数。 |
| vpatchVersion | vpatch版本。 |
| vpatchTmpDir | vpatch临时文件目录。 |
| vpatchLogDir | vpatch日志目录 |
| newSVN | 新版本的SVN提交号。 |
| baseSVN | 原始版本的SVN提交号。 |
| deleteFile | 此次升级删除的文件，包括数据库和om工具。 |
| isFallbackPatch | 用于升级还是回退的补丁，true升级，false回退。 |
| allowRollback | 补丁是否允许回退，true可以回退，false不可回退。 |
| base\_product\_version | 适配的基础版本。 |
| osInfo | 适用的操作系统信息。 |
| cpuInfo | 适用的cpu信息。 |
| ConsumedOpSeqNum | 数组的大小。 |

**示例**

**示例1**：升级操作

1、解压升级包并进入目录。

cd /home/数据库用户/解压的升级包/VPatch/script

2、指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

配置文件参考如下：

[vpatch]

userName=jq\_vpatch\_two

userGroup=jq\_vpatch\_two

appPath=/home/jq\_vpatch\_two/local/vastbase

dataPath=/home/jq\_vpatch\_two/data/vastbase

Port=8560

tmpPath=/home/jq\_vpatch\_two/local/omTmp/tmp

3、root执行命令进行检查。

./vpatch -C

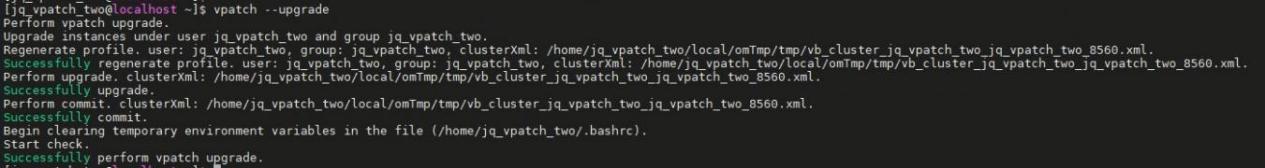
4、root执行命令进行用户权限检查。

./vpatch -P

5、切换数据库初始化用户进行升级。

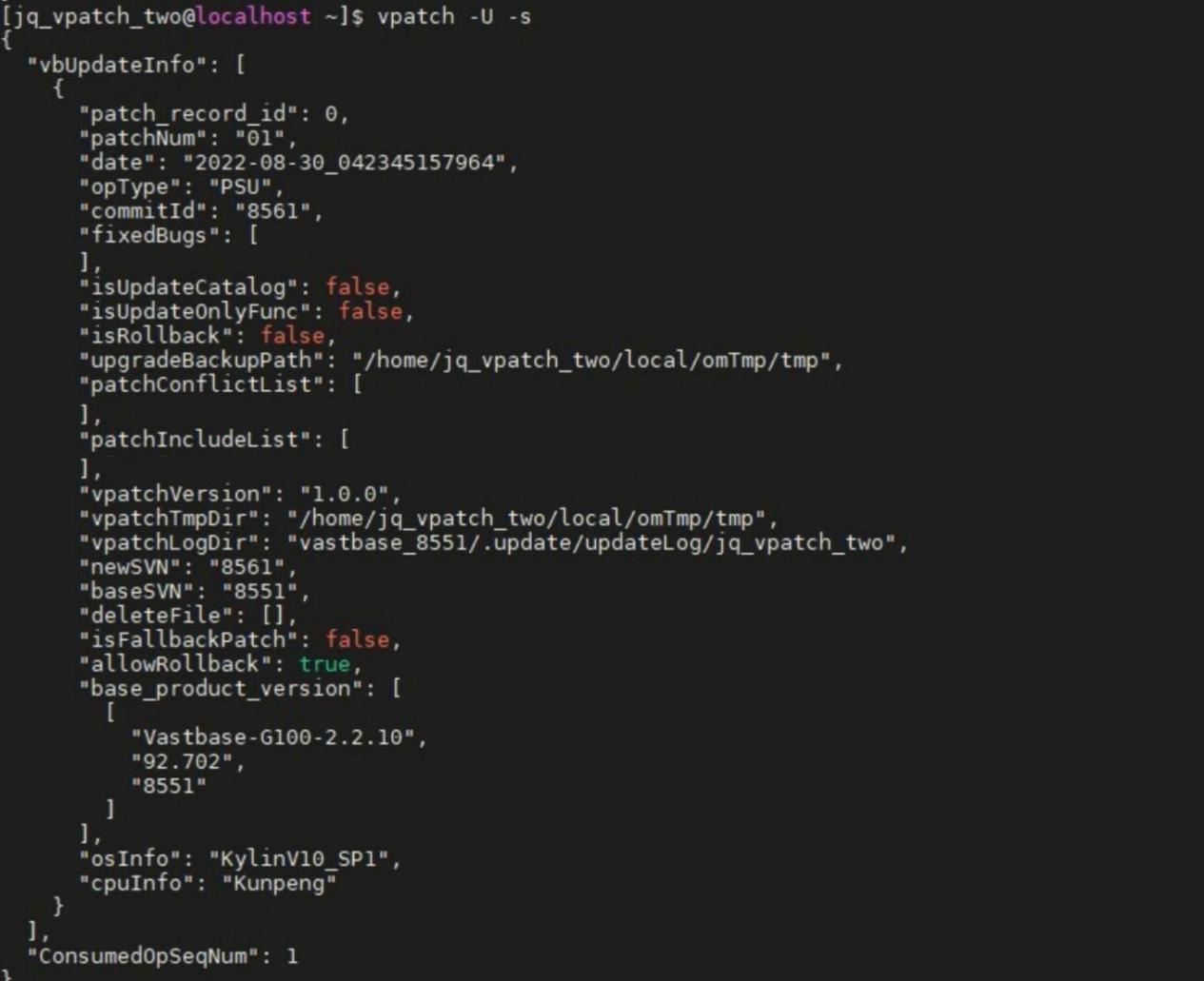
vpatch --upgrade

结果返回如下：

6、升级完成后信息检查。

* 简略显示升级历史信息，信息包括操作序列号、补丁编号、升级操作的日期和时间、操作类型、修复的bug列表、升级是否修改数据字典、是否已被回退、升级备份的目录。参数说明参考表2历史信息。

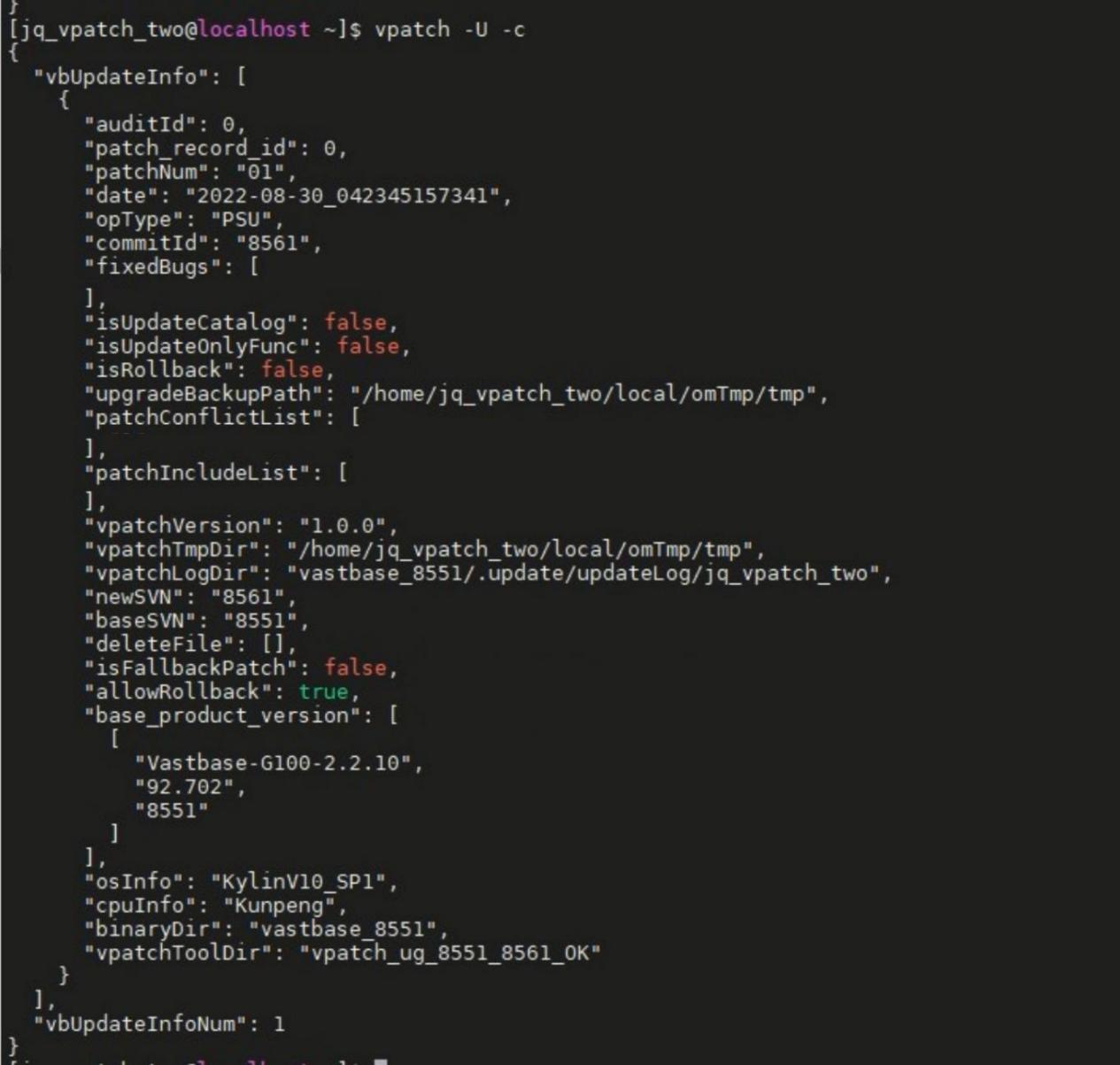
vpatch -U 或 vpatch -U -s

 结果返回如下：

* 完整信息升级记录信息，信息包括操作序列号、补丁编号、升级操作的日期和时间、操作类型、修复的bug列表、升级是否修改数据字典、是否已被回退、升级备份的目录。参数说明参考表2历史信息。

vpatch -U -c

结果返回如下：

* 显示升级过程，信息包含停止服务，启动服务操作，执行的每个脚本文件，操作数据库的相关命令，变更的设置参数，查看数据库升级日志。

cat $GAUSSHOME/.update/updatelog/jq\_vpatch\_two/gs\_upgradectlxxx.log

**示例2：**回退操作。

1、停止数据库服务。

2、root用户指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

3、root用户进行检查。

./vpatch -C --rollbackId=0

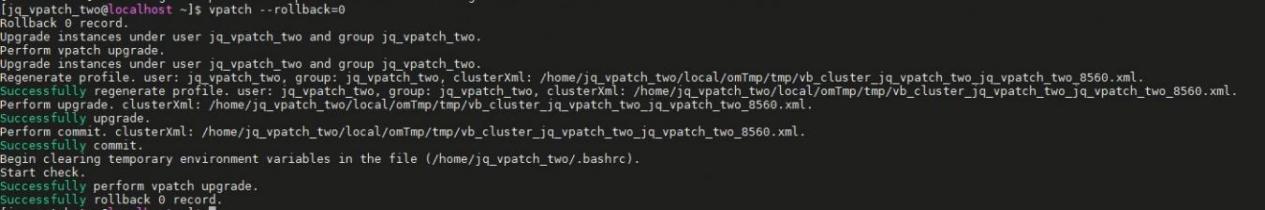
4、root执行命令进行用户权限检查。

./vpatch -P --rollbackId=0

5、切换数据库初始化用户执行回退。

vpatch --rollback=0

结果返回如下：



**示例3：**连续回退操作。

1、停止数据库服务。

2、root用户指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

3、root用户进行检查。

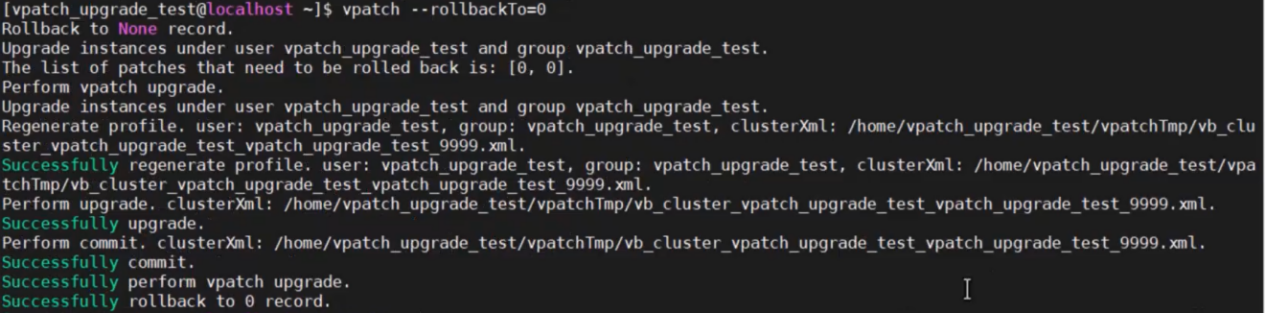
./vpatch -C --rollbackToId=0

4、root执行命令进行用户权限检查。

./vpatch -P --rollbackToId=0

5、切换数据库初始化用户执行连续回退。

vpatch --rollbackTo=0

结果返回如下：

**示例4：**显示vpatch同目录下升级补丁的信息，参数说明参考表1补丁信息。

./vpatch - I

结果返回如下：



**示例5：**集群升级操作

1. 使用root用户登录主机，执行以下命令查看主备状态，/home/vastbase/has/bin/hasctl为hasctl工具目录。

/home/vastbase/has/bin/hasctl -c /home/vastbase/has/etc/patroni\_conf.yml list

1. root用户登录备节点执行以下命令停止has

systemctl stop has

1. root用户登录主节点，执行步骤2命令停止主节点has服务。
2. 复制postgresql.conf做配置文件备份

cp postgresql.conf postgresql.conf.bak

5、注释掉主备节点配置文件postgresql.conf中的replconn参数，并修改port指定升级专用端口号，重启数据库。确认数据库可以正常启动后，停止所有节点的数据库服务进行升级操作。

6、升级参考示例1分别对主备节点进行操作。

7、升级成功后，关闭数据库，恢复postgresql.conf文件主备及端口配置。可将步骤4中备份的postgresql.conf.bak文件恢复为postgresql.conf。

8、以root用户登录主节点，并执行以下命令先启动主节点has：

systemctl start has

9、使用以下命令查看主节点状态，确认主节点状态正常。

/home/vastbase/has/bin/hasctl  -c /home/vastbase/has/etc/patroni\_conf.yml list

10、以root用户登录备节点，并执行以下命令启动备节点has：

systemctl start has

11、所有节点启动后执行以下命令查看集群状态是否正常。集群正常则升级成功

/home/vastbase/has/bin/hasctl  -c /home/vastbase/has/etc/patroni\_conf.yml list

## 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。

**概述**

**基本功能**

* 连接数据库：详情请参考《开发者指南》中数据使用—>连接数据库。
* 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。

**高级特性**

* **设置变量**

vsql提供类似于Linux的shell命令的变量特性，可以使用vsql的原命令\set设置一个变量，格式如下：

\set varname value

要删除一个变量请使用如下方式：

\unset varname



* 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。
* **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。

像元命令的参数一样，vsql变量的一个关键特性是可以把vsql变量替换成正规的SQL语句。此外，vsql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=# select \* from tt;  
 a | b | c | d  
---+---+---+---  
 1 | 2 | 3 | 4  
 5 | 6 | 7 | 8  
(2 rows)  
  
vastbase=# \set foo1 'tt'  
vastbase=# select \* from :foo1;  
 a | b | c | d  
---+---+---+---  
 1 | 2 | 3 | 4  
 5 | 6 | 7 | 8  
(2 rows)

* **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。

* **命令自动补齐**

根据Vastbase语法规则，vsql支持使用Tab键进行命令的自动补齐，当编译时指定了选项–with-readline，且客户端连接时指定“-r”参数，此功能被打开。例如，crea后键入Tab，vsql会将其补齐为create。



* 支持数据库SQL关键字如select、create、table等。
* 支持表名、视图名等自定义标识符的补齐。
* 元命令选项'S'、'+'不支持自动补齐。
* 对表进行补齐时，只有前缀是“pg\_”才会补齐系统表。
* 不支持建表时字段类型的补齐。
* select后不支持任何补齐。
* 不支持常量与宏的自动补齐。
* select \* from a,b… 不支持第二个开始表的自动补齐, insert into t1 (col1, col2, …) 不支持第二个列的自动补齐。
* 不支持create tablespace语句with以及with后参数的自动补齐。
* 创建索引不支持local、global的自动补齐，修改索引不支持rebuild自动补齐。
* set语句仅支持自动补全user和superuser级别的参数。
* 不支持if exists的自动补齐。
* 不支持表名.列名的自动补齐，如alter sequence <name> owned by tableName.colName，owned by
* 不支持自定义操作符自动补齐。使用复制粘贴这种方式输入命令，如果粘贴的命令里面有TAB键有可能会使输入命令的格式错乱，无法正常执行。
* **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。



* 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。

**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）。
* 双引号（“”）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“A weird”“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11 一般的元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table23103470)、[表12 查询缓存区元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table8624338)、[表13 输入/输出元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table62130565)、[表14 显示信息元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table44907299)、[表16 格式化元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table53911699)、[表18 连接元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table37189353)、[表19 操作系统元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table32146234)、[表20 变量元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table50076468)和[表22 大对象元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table55465693)。



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/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 11**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，vsql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。 说明： （1）不支持事务中开启并发执行以及并发中开启事务。 （2）不支持\d这类元命令的并发。 （3）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。 （4）不推荐在并发中使用set语句，否则导致结果与预期不一致。 （5）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。 （6）\parallel执行时最多会启动num个独立的vsql进程连接服务器。 （7）\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。 （8）在\parallel on 之后一条或多条命令，会等到\parallel off执行后才会执行，因而，每个\parallel on需对应一个\parallel off，否则会导致\parallel on之后的命令无法执行。 | num的默认值：1024。 须知： （1）服务器能接受的最大连接数受max\_connection及当前已有连接数限制。 （2）设置num时请考虑服务器当前可接受的实际连接数合理指定。 |
| \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 12**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 13**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 ] ] [parallel integer] | 在任何vsql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。 \COPY 可以指定数据导入时的客户端数量，从而实现数据文件的并行导入，目前并发数范围为[1, 8]。 \COPY并行导入目前存在以下约束：临时表的并行导入不支持、在事务内的并行导入不支持、对二进制文件的并行导入不支持、数据导入支持AES128加密时不支持。在这些情况下，即使指定了parallel参数，仍然会走非并行流程。 |
| \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。 |



[表14 显示信息元命令](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table44907299)中的选项S表示显示系统对象，PATTERN表示显示对象附加的描述信息。用来指定要被显示的对象名称。

**表 14** 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。 vastbase=# \d |
| \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。 vastbase=# \dtable+ a` |
| \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。 vastbase=# \d+ f |
| \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PATTERN的聚集函数。 | 列出所有名称以f开头可用的聚集函数以及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，只显示名称匹配PATTERN的服务器。 | 列出所有的外部服务器。 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，只列出名称匹配PATTERN的语言。 | 列出可用的程序语言。 vastbase=# \dL |
| \dn[S+] [PATTERN] | 列出物化视图。 | 如果指定了PATTERN，只列出名称匹配PATTERN的物化视图。 | 列出物化视图。 vastbase=# \dm |
| \dn[S+] [PATTERN] | 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。 vastbase=# \dn+ d\* |
| \do[S] [PATTERN] | 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。 vastbase=# \do |
| \dO[S+] [PATTERN] | 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。 vastbase=# \dO |
| \dp [PATTERN] | 列出一列可用的表、视图以及相关的权限信息。\dp显示结果如下：rolename=xxxx/yyyy –赋予一个角色的权限=xxxx/yyyy –赋予public的权限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 权限的参数说明](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_t9b9f37d7c1c54a7893307344083e644e)。 | 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。 vastbase=# \dp |
| \drds [PATTERN1 [PATTERN2]] | 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。 vastbase=# \drds \* postgres |
| \dT[S+] [PATTERN] | 列出所有的数据类型。 | 如果指定了PATTERN，只列出名称匹配PATTERN的类型。 | 列出所有的数据类型。 vastbase=# \dT |
| \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。 vastbase=# \du |
| \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。 | 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。 vastbase=# \div |
| \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。 vastbase=# \dx |
| \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。 vastbase=# \l |
| \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。 vastbase=# \sf function\_a vastbase=# \sf “func()name”(argtype1, argtype2) |
| \z [PATTERN] | 列出数据库中所有表、视图和序列以及它们相关的访问特权。 | 如果给出任何pattern ，则被当成一个正则表达式，只显示匹配的表、视图、序列。 | 列出数据库中所有表、视图和序列以及它们相关的访问特权。 vastbase=# \z |

**表 15**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。 |
| A | ALTER：允许用户修改指定对象的属性。 |
| P | DROP：允许用户删除指定的对象。 |
| m | COMMENT：允许用户定义或修改指定对象的注释。 |
| i | INDEX：允许用户在指定表上创建索引。 |
| v | VACUUM：允许用户对指定的表执行ANALYZE和VACUUM操作。 |
| \* | 给前面权限的授权选项。 |

**表 16**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 可调节的打印选项](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_zh-cn_topic_0058968158_table22053343)。 |
| \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 17**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 |
| 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。 |
| 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gsql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。 | - |
| feedback | 切换是否输出结果行数。 | - |

**表 18**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，受限于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 19**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 20** 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21](https://docs.vastdata.com.cn/zh/docs/VastbaseG100Ver2.2.10/doc/%E5%B7%A5%E5%85%B7%E5%8F%82%E8%80%83/vsql.html" \l "zh-cn_topic_0237152146_zh-cn_topic_0059778645_t03d91a7973ee453b918f0b755284f04d)是一个所有特殊对待的变量列表。 |
| \unset NAME | 不设置（或删除）vsql变量名。 |

**表 21** \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：  不支持事务块中的语句错误重试；  不支持通过ODBC、JDBC接口查询的出错重试；  含有unlogged表的sql语句，不支持节点故障后的出错重试；  vsql客户端本身出现的错误，不在重跑考虑范围之内； | retry\_times取值范围为：5-10 |

**表 22**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的注释。 |

**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。如果需要在一个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。



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