

# 虚谷数据库

## 控制台工具 V1.4.5

### 用户指南

文档版本 01

发布日期 2024-07-30



版权所有 © 2024 成都虚谷伟业科技有限公司。

## 声明

未经本公司正式书面许可，任何企业和个人不得擅自摘抄、复制、使用本文档中的部分或全部内容，且不得以任何形式进行传播。否则，本公司将保留追究其法律责任的权利。

用户承诺在使用本文档时遵守所有适用的法律法规，并保证不以任何方式从事非法活动。不得利用本文档内容进行任何侵犯他人权益的行为。

## 商标声明



为成都虚谷伟业科技有限公司的注册商标。

本文档提及的其他商标或注册商标均非本公司所有。

## 注意事项

您购买的产品或服务应受本公司商业合同和条款的约束，本文档中描述的部分产品或服务可能不在您的购买或使用范围之内。由于产品版本升级或其他原因，本文档内容将不定期进行更新。

除非合同另有约定，本文档仅作为使用指导，所有内容均不构成任何声明或保证。

## 成都虚谷伟业科技有限公司

地址：四川省成都市锦江区锦盛路 138 号佳霖科创大厦 5 楼 3-14 号

邮编：610023

网址：[www.xugudb.com](http://www.xugudb.com)

# 前言

## 概述



本文档对控制台工具的功能和安装使用进行了详细的介绍。

## 读者对象

- 数据库管理员
- 数据库用户

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 注意	用于传递设备或环境安全警示信息，若不避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。
 说明	对正文中重点信息的补充说明。“说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

## 修改记录

文档版本	发布日期	修改说明
01	2024-07-30	第一次发布

# 目录

1	工具概述 .....	1
2	启动方式 .....	2
2.1	Windows .....	2
2.2	Linux .....	3
3	功能介绍 .....	5
3.1	切换用户 .....	5
3.2	替换命令文本 .....	5
3.3	退出控制台工具 .....	6
3.4	主动重连数据库 .....	6
3.5	设置控制台属性 .....	6
3.5.1	PAGESIZE .....	6
3.5.2	B_PAUSE .....	7
3.5.3	HEADING .....	7
3.5.4	SHOW_LINE_NUMBER .....	8
3.5.5	SHOW_TYPE .....	8
3.5.6	SPUT_HEAD .....	8
3.5.7	SHOW_TIME .....	9
3.5.8	SHOW_USE_TIME .....	9
3.5.9	DATE_FORMAT .....	10
3.5.10	NULL .....	11
3.5.11	ERRQUIT .....	11
3.6	重置控制台属性 .....	12
3.6.1	PAGESIZE .....	12
3.6.2	HEADING .....	12
3.6.3	SHOW_LINE_NUMBER .....	12
3.6.4	SHOW_TYPE .....	12
3.6.5	SPUT_HEAD .....	12

3.6.6 SHOW_TIME .....	13
3.6.7 SHOW_USE_TIME .....	13
3.7 清理屏幕 .....	13
3.8 清理输入缓存 .....	13
3.9 重新建立连接 .....	14
3.10 数据导入与导出 .....	14
3.10.1 数据导入 .....	14
3.10.2 数据导出 .....	17
3.11 EDIT 编译命令 .....	17
3.12 执行脚本 .....	18
3.13 查询表信息 .....	18

# 1 工具概述

虚谷数据库控制台工具是由虚谷数据库提供的连接数据库的客户端工具，目前有 Windows 版和 Linux 版两种。控制台工具比较常用，功能丰富，主要包括以下几点功能：

- 数据库的维护，如关闭数据库。
- 执行 SQL 语句和 PL/SQL 语句。
- 执行 SQL 脚本。
- 数据的导入和导出。
- 在脚本中使用。

## 说明

控制台的语句不区分大小写。

# 2 启动方式

## 2.1 Windows

在虚谷数据库产品包的虚谷数据库-客户端文件中解压“XuguConsole-win-x64”压缩包，然后找到 Windows 版控制台执行文件，默认文件名为“XuguConsole.exe”。

支持以下两种方式启动：

- 直接启动
- 参数启动

### 直接启动

双击“XuguConsole.exe”运行，按照程序提示输入相应的信息，如果连接建立失败，则重新输入。

### 参数启动

在 CMD 中进入控制台所在的路径，输入如下命令。

```
C:\console>XuguConsole.exe USESSL IP PORT DB_NAME USER_NAME  
USER_PASS
```

参数说明：

- USESSL：是否使用加密，nssl 表示不使用加密。
- IP：数据库连接的 IP 地址。
- PORT：端口号。
- DB\_NAME：目标库名。
- USER\_NAME：目标用户名。
- USER\_PASS：目标用户名的密码。

示例：

```
C:\console>XuguConsole.exe nssl 127.0.0.1 5138 SYSTEM SYSDBA SYSDBA
```

### 📖 说明

控制台登录输入时，隐藏输入的字符。

## 2.2 Linux

将虚谷数据库产品包下虚谷数据库-客户端文件中的“XuguConsole-linux-x64”压缩包上传至 Linux 服务器，解压并找到 Linux 版控制台文件，默认文件名为“xgconsole”，增加可执行权限。

支持以下两种方式启动：

- 直接启动
- 参数启动

### 直接启动

按照程序提示输入相应的信息，如果连接建立失败，则重新输入。

### 参数启动

进入控制台所在的路径，输入如下命令。

```
[root@localhose console]#./xgconsole USESSL IP PORT DB_NAME  
USER_NAME USER_PASS
```

参数说明：

- USESSL：是否使用加密，nssl 表示不使用加密。
- IP：数据库连接的 IP 地址。
- PORT：端口号。
- DB\_NAME：目标库名。
- USER\_NAME：目标用户名。
- USER\_PASS：目标用户名的密码。

示例：

```
[root@localhose console]#./xgconsole nssl 127.0.0.1 5138 SYSTEM  
SYSDBA SYSDBA
```



### 说明

控制台登录输入时，隐藏输入的字符。

# 3 功能介绍

## 3.1 切换用户

### 功能

实现在同一库中不同用户间进行切换。

### 关键字

\$

### 语法格式

```
SQL> $用户名
```

### 示例

```
SQL> $user_test
```

## 3.2 替换命令文本

### 功能

使用新字符串替换输入缓存区中的字符串。

### 关键字

c

### 语法格式

```
SQL> c/被替换的文本/替换文本
```

### 示例

```
SQL> select * from sys_table;  
Error: [E5021 L1 C15] 表或视图SYS_TABLE不存在  
SQL> c/sys_table/sys_tables;
```

### 说明

替换之后会自动执行缓存区命令。

## 3.3 退出控制台工具

### 功能

关闭控制台工具。

### 关键字

EXIT

### 示例

```
SQL> EXIT;
```

## 3.4 主动重连数据库

### 功能

与数据库重新建立连接。

### 关键字

RECONNECT 或 RECONN

### 示例

```
SQL> RECONNECT;
```

## 3.5 设置控制台属性

### 3.5.1 PAGESIZE

### 功能

开启结果集分页功能并设置结果集分页尺寸。

### 语法格式

```
SQL> set PAGESIZE XXX
```

### 示例

```
SQL> set pagesize 10;
pagesize 10

SQL> select ACTIVITY_CODE from txn_proc_result limit 100;

ACTIVITY_CODE |
-----
013022 |
013022 |
```

```
013022 |  
013022 |  
013022 |  
013022 |  
013022 |  
013022 |  
013022 |  
013022 |
```

#### 📖 说明

开启分页功能时可以使用 **Q** 或者 **ESC** 退出结果集展示。

## 3.5.2 B\_PAUSE

### 功能

关闭或开启分页功能（与 PAGESIZE 配合作用）。

### 语法格式

```
SQL> set B\_PAUSE on|off
```

### 示例

```
SQL> set B\_PAUSE on;
```

## 3.5.3 HEADING

### 功能

控制查询结果集是否输出列头信息。

### 语法格式

```
SQL> set HEADING on|off
```

### 示例

```
SQL> select 1 from dual;  
  
EXPR1 |  
-----  
1 |  
  
Total 1 records.  
Use time:1 ms.  
  
SQL> set heading off;  
HEADING OFF  
  
SQL> select 1 from dual;
```

```
1 |  
  
Total 1 records.  
Use time:0 ms.
```

### 3.5.4 SHOW\_LINE\_NUMBER

#### 功能

控制输入多行 SQL 的时候，是否在每行开头展示当前行数。

#### 语法格式

```
SQL> set SHOW\_LINE\_NUMBER on|off
```

#### 示例

```
SQL> set show_line_number on;  
SHOW_LINE_NUMBER ON  
  
SQL> select 1  
2 from dual;
```

### 3.5.5 SHOW\_TYPE

#### 功能

控制结果集列头是否展示列的数据类型和精度（需要先开启列头信息 heading）。

#### 语法格式

```
SQL> set SHOW\_TYPE on|off
```

#### 示例

```
SQL> set show_type on;  
SHOW_TYPE ON  
  
SQL> select 1 from dual;  
  
EXPR1 (TINYINT) |  
-----  
1 |  
  
Total 1 records.  
Use time:0 ms.
```

### 3.5.6 SPUT\_HEAD

#### 功能

控制将结果输入到文件时是否输入列名信息。

#### 语法格式

```
SQL> set SPUT\_HEAD on|off
```

### 示例

```
SQL> set sput_head on;  
SPUT_HEAD ON  
  
SQL> select * from t_var; >$ d:/test.csv;  
  
Total:15 record  
Use time:5 ms.
```

## 3.5.7 SHOW\_TIME

### 功能

控制输入命令时是否展示当前时间。

### 语法格式

```
SQL> set SHOW\_TIME on|off
```

### 示例

```
SQL> set SHOW_TIME ON;  
SHOW_TIME ON  
  
09:23:30 SQL>
```

## 3.5.8 SHOW\_USE\_TIME

### 功能

控制是否输出语句执行时间。

### 语法格式

```
SQL> set SHOW\_USE\_TIME on|off
```

### 示例

```
SQL> select 1 from dual;  
  
EXPR1 |  
-----  
1 |  
  
Total 1 records.  
Use time:0 ms.  
  
SQL> set SHOW_USE_TIME Off;  
SHOW_USE_TIME OFF
```

```
SQL> select 1 from dual;

EXPR1 |
-----
1 |

Total 1 records.
```

## 3.5.9 DATE\_FORMAT

### 功能

控制 datetime 或 date 类型数据输出格式。

### 语法格式

```
SQL> set DATE\_FORMAT 任意年月日和中间分隔符的组合
```

### 示例

```
SQL> set DATE_FORMAT yyyy-mm-dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----
2022-10-09 09:30:58.496 AD |
```

```
Total 1 records.
Use time:0 ms.
```

```
SQL> set DATE_FORMAT mm-YYYY-dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----
10-2022-09 09:31:14.952 AD |
```

```
Total 1 records.
Use time:0 ms.
```

```
SQL> set DATE_FORMAT mm%YYYY%dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----
10%2022%09 09:31:30.819 AD |
```

```
Total 1 records.
Use time:0 ms.
```

### 3.5.10 NULL

#### 功能

控制空列值输出内容。

#### 语法格式

```
SQL> set NULL 空值输出内容
```

#### 示例

```
SQL> select null from dual;
```

```
EXPR1 |
```

```
-----  
<NULL> |
```

```
Total 1 records.
```

```
Use time:0 ms.
```

```
SQL> set NULL NULL;
```

```
NULL will be display as NULL
```

```
SQL> select null from dual;
```

```
EXPR1 |
```

```
-----  
NULL |
```

```
Total 1 records.
```

```
Use time:0 ms.
```

### 3.5.11 ERRQUIT

#### 功能

SQL 执行失败下，直接退出控制台工具，并将错误信息重定向到 stderr 中。

#### 📖 说明

仅在 Linux 版本控制台中支持。

#### 语法格式

```
SQL> set ERRQUIT on|off
```

#### 示例

```
SQL> set ERRQUIT on;
```



## 3.6 重置控制台属性

### 3.6.1 PAGESIZE

默认值

10

语法格式

```
SQL> reset PAGESIZE
```

### 3.6.2 HEADING

默认值

ON

语法格式

```
SQL> reset HEADING
```

### 3.6.3 SHOW\_LINE\_NUMBER

默认值

OFF

语法格式

```
SQL> reset SHOW_LINE_NUMBER
```

### 3.6.4 SHOW\_TYPE

默认值

OFF

语法格式

```
SQL> reset SHOW_TYPE
```

### 3.6.5 SPUT\_HEAD

默认值

OFF

语法格式

```
SQL> reset SPUT_HEAD
```

## 3.6.6 SHOW\_TIME

### 默认值

OFF

### 语法格式

```
SQL> reset SHOW_TIME
```

## 3.6.7 SHOW\_USE\_TIME

### 默认值

ON

### 语法格式

```
SQL> reset SHOW_USE_TIME
```

## 3.7 清理屏幕

### 功能

清空屏幕输出。

### 关键字

CLS

### 示例

```
SQL> CLS;
```

## 3.8 清理输入缓存

### 功能

清空输入缓存区里面的命令。

### 关键字

CLBUFF

### 示例

```
SQL> CLBUFF;
```

## 3.9 重新建立连接

### 功能

重新输入信息连接新的数据库。

### 关键字

LOGIN

### 示例

```
SQL> LOGIN;
```

## 3.10 数据导入与导出

### 3.10.1 数据导入

#### 3.10.1.1 SQLLDR

### 功能

SQLLDR 是文件导入命令，可以将控制台工具或者其他工具导出的结果集文件导入到数据库的表中，需要注意的是文件的列数要与表的列数一致，并且是顺序对应的关系。

### 示例

先将查询结果集导出成文件，再将文件导入到数据表中。

```
SQL> sqlldr table=sysdba.warehouse datafile=F:\tpcc100w\warehouse.  
txt ft='|' rt=X'0d0a' log=./load.log errlog=./loaderr.log mode=  
replace imp_type=block
```

### 参数解释

- sqlldr: 导入命令标识符。
- table: 目标表名。
- datafile: 数据文件的路径 + 文件名。
- ft: 列分隔符，如 “|”、“,” 或其它。
- rt: 行结尾标识符，可设为 X'0a' 或 X'0d0a'。

### 📖 说明

X'0a' 表示行结尾换行为\n，X'0d0a' 表示行结尾换行为\r\n。

- log: 导入日志。
- errlog: 导入时错误日志。
- mode: 导入方式, 包括 replace 和 append。
- imp\_type: 导入类型, 包括 block 和 insert。
- csv\_deal: 格式化 CSV 文件, 去除单元格开头双引号, 可设为 true 或 false, 默认为 false (若数据文件不为 CSV 格式时, 无需指定该参数)。

### 3.10.1.2 TABLDR

#### 功能

TABLDR 也是文件导入命令, 和 SQLLDR 一样可以将控制台工具或者其他工具导出的结果集文件导入到数据库的表中, 区别在于 TABLDR 导入的效率会比 SQLLDR 高。

#### 示例

先将查询结果集导出成文件, 再将文件导入到数据表中。

```
SQL> tabldr table=NATION datafile=D:/NATION.TXT ft='|' rt=X'0d0a'  
log=./load.log errlog=./loaderr.log mode=APPEND parse_para=8  
ins_para=8
```

#### 参数解释

- tabldr: 导入命令标识符。
- table: 目标表名。
- datafile: 数据文件的路径 + 文件名。
- ft: 列分隔符, 如 “|”、“,” 或其它。
- rt: 行结尾标示符, 可设为 X'0a' 或 X'0d0a'。

#### 说明

X'0a' 表示行结尾换行为\n, X'0d0a' 表示行结尾换行为\r\n。

- log: 导入日志。
- errlog: 导入时错误日志。
- mode: 导入方式, 包括 replace 和 append。

- parse\_para: 服务端解析数据线程数, 从 1 开始。
- ins\_para: 服务端插入数据线程数, 从 1 开始。
- csv\_deal: 格式化 CSV 文件, 去除单元格开头双引号, 可设为 true 或 false, 默认为 false。

#### 📖 说明

若数据文件不为 CSV 格式时, 无需指定该参数。

### 3.10.1.3 COLLDR

#### 功能

COLLDR 不同于 TABLLDR, COLLDR 使用 insert 语句导入数据, 可以实现部分列的导入功能。

#### 示例

先将查询结果集导出成文件, 再将文件导入到数据表中。

```
SQL> collldr table=t_var datafile=d:/test.csv ft=', ' rt=X'0d0a'  
colhead=id,var
```

#### 参数解释

- collldr: 导入命令标识符。
- table: 目标表名。
- datafile: 数据文件的路径 + 文件名。
- ft: 列分隔符, 如 “|”、“,” 或其它。
- rt: 行结尾标示符, 可设为 X'0a' 或 X'0d0a'。

#### 📖 说明

X'0a' 表示行结尾换行为\n, X'0d0a' 表示行结尾换行为\r\n。

- colhead: 导入文件的列名, 列名之间使用逗号分割。

### 📖 说明

不设置 `colhead` 参数表示列名在文件的第一行中存在。

## 3.10.1.4 导入大对象

### 示例

```
SQL> create table test(B blob, C clob);  
SQL> insert into test(B) values (?); <# D:\t1.png;  
SQL> insert into test(C) values (?); <% D:\t3.txt;
```

## 3.10.2 数据导出

### 3.10.2.1 导出结果集

#### 功能

使用控制台可以将 SELECT 查询的结果导出到文本文件中，导出的数据默认以“|”作为分隔符（csv 文件以“,”作为分隔符）。导出的结果集中不能包含大对象，大对象需要单独导出。

#### 示例

```
SQL> select * from test; >$ D:\test_select.txt;
```

### 3.10.2.2 导出大对象

#### 示例

```
SQL> create table test(B blob, C clob);  
SQL> select B from test; ># D:\t2.png;;  
SQL> select C from test; >% D:\t4.txt;
```

## 3.11 EDIT 编译命令

#### 功能

控制台会将上一次执行过的 SQL 命令（包括 SQL 语句和 PL/SQL 语句，一行或多行）信息保存在缓冲区，输入“EDIT”进行编辑，编辑后通过“/”命令执行修改后的缓冲区中的命令。

#### 示例

```
SQL> EDIT  
SQL> /
```

## 3.12 执行脚本

### 操作步骤

1. 准备脚本。

在脚本中用 EXECBEGIN/EXECEND 包裹 SQL 语句，格式如下：

```
EXECBEGIN
... ..
多条 SQL 命令
... ..
EXECEND
```

2. 执行脚本。

```
SQL> EXECFILE C:\test_execfile.sql

SQL> EXECFILE2 C:\test_execfile.sql
```

### 📖 说明

若使用 EXECFILE2 执行脚本文件，则脚本内的 SQL 语句不需要用 EXECBEGIN/EXECEND 包裹，直接列出即可。

## 3.13 查询表信息

### 功能

查询表的列信息，包括列名、是否为空和列类型。

### 关键字

DESC

### 语法格式

```
DESC 表名
```

### 示例

```
DESC txn_proc_result;
Column_Name                NULL          Data_Type
-----
ACTIVITY_CODE              | NOT NULL    | CHAR(8)
REQ_SYS                    | NOT NULL    | CHAR(4)
REQ_DATE                   | NOT NULL    | CHAR(8)
UPAY_BIZ_SEQ               |              | CHAR(32)
REQ_TRANS_ID               | NOT NULL    | CHAR(32)
SESSION_ID                 |              | CHAR(128)
```

REQ_DATE_TIME	NOT NULL	CHAR (17)
ACTION_CODE	NOT NULL	CHAR (1)
RCV_SYS	NOT NULL	CHAR (4)
RCV_DATE		CHAR (8)
RCV_TRANS_ID		CHAR (32)
RCV_DATE_TIME		CHAR (17)
MER_ID	NOT NULL	CHAR (32)
VAS_TYPE	NOT NULL	CHAR (1)
PAY_TYPE	NOT NULL	CHAR (32)
AMOUNT	NOT NULL	NUMERIC (32,0)
HOME_PROV	NOT NULL	CHAR (3)
CHARGE_NUMBER	NOT NULL	CHAR (32)
TRADE_SEQNO	NOT NULL	CHAR (32)
TRADE_SESSION	NOT NULL	CHAR (32)
OLD_TRADE_SESSION		CHAR (32)
SETTLE_DATE	NOT NULL	CHAR (8)
TRADE_TIME	NOT NULL	CHAR (14)
OLD_SETTLE_DATE		CHAR (8)
OLD_TRADE_TIME		CHAR (14)
SP_CODE	NOT NULL	CHAR (64)
SERVICE_CODE	NOT NULL	CHAR (24)
BAL_PROP	NOT NULL	NUMERIC (32,0)
BILL_FLAG	NOT NULL	CHAR (1)
CONTENT_ID	NOT NULL	CHAR (32)
POINT_COST		NUMERIC (32,0)
VISIT_PROV		CHAR (3)
DEPOSIT_TYPE		CHAR (4)
APPLY_TIME		CHAR (14)
FINISH_TIME		CHAR (14)
READ_CONTENT_TYPE		CHAR (4)
MSISDN_REC		CHAR (11)
DURATION		NUMERIC (32,0)
BYTE_SIZE		NUMERIC (32,0)
MEMBER_TYPE	NOT NULL	CHAR (1)
CP_CODE		CHAR (30)
USER_TYPE		CHAR (1)
CRM_RESULT_CODE		CHAR (32)
RESULT_CODE		CHAR (6)
RESULT_DESC		CHAR (256)
NOTIFY_TYPE		CHAR (2)
CENTREID		CHAR (4)





成都虚谷伟业科技有限公司

联系电话：400-8886236

官方网站：[www.xugudb.com](http://www.xugudb.com)