

# OBLOADER/OBDUMPER 4.0.0 官方主页

OceanBase 自研的客户端导入导出工具已正式开放下载：[OBLOADER/OBDUMPER 4.0.0](#)

通告：OBLOADER/OBDUMPER 1.x/2.x 已不再维护，请用户参考 [《升级指南》](#) 酌情升级。

## 什么是 OBLOADER?

OBLOADER 是一款使用 Java 语言开发的客户端工具，目前该工具仅适用于 OceanBase 数据库。用户可以将存储介质中的数据库对象的定义文件和表数据文件导入到 OceanBase 数据库中。通常我们推荐 OBLOADER 与 OBDUMPER 搭配使用。如果用户希望借助于 OBLOADER 完成数据迁移工作，它也兼容 mysqldump, Mydumper 等客户端工具导出的 CSV 格式的文件。

OBLOADER 专门优化了数据的导入性能，内置多种数据预处理函数，自动容错保证数据导入的稳定性，以及提供较为丰富的监控信息，以便于用户实时观测到数据文件导入的性能和进度。

## OBLOADER 主要具备以下功能特性：

- 支持从本地磁盘，Apache Hadoop, 阿里云OSS 或者 AWS S3 导入数据库对象定义和表数据。
- 支持导入 mysqldump 导出的 SQL-Format 格式的文件。
- 支持导入标准的 CSV, Insert SQL, ORC, Parquet 等格式的数据文件。
- 支持配置数据预处理的控制规则以及文件与表之间的字段映射关系。
- 支持导入限速、防导爆、断点恢复和错误自动重试。
- 支持指定自定义的日志目录，导入时可以保存坏数据和冲突数据。
- 支持导入工具自动对大文件进行切分且不占用额外的存储空间，无需人工切分文件。
- 支持对命令行中指定的敏感参数进行加密。包括：数据库的账号密码，云存储的账号密钥。

## 什么是 OBDUMPER?

OBDUMPER 是一款使用 Java 语言开发的客户端工具，目前该工具仅适用于 OceanBase 数据库。用户可以使用该工具将 OceanBase 数据库中定义的对象和表数据以指定的文件格式导出到存储介质中。

如果用户希望借助于 OBDUMPER 进行逻辑备份，可以直接将该工具集成到数据库运维系统中。（注：不支持增量备份）与 mysqldump 等客户端导出工具相比，OBDUMPER 具备以下显著的优势：

- 快速的数据导出能力，设计了多种数据查询策略，大幅提升导出的性能。
- 丰富的数据交换能力，支持将表中数据以多种格式导出到多种存储介质。
- 强大的数据处理能力，导出前对数据进行压缩，加密，脱敏，预处理等。

## OBDUMPER 主要具备以下功能特性：

- 支持导出数据库对象定义和表数据到本地磁盘，阿里云 OSS 和 AWS S3。
- 支持将表中的数据按照 CSV, Insert SQL, ORC, Parquet 等格式导出到文件中。
- 支持指定分区名，仅导出指定的表分区内的数据。
- 支持指定全局的过滤条件，仅导出满足条件的数据。
- 支持配置数据预处理规则，导出前对数据进行转换、脱敏等预处理。
- 支持指定 SCN 或者 TIMESTAMP，仅导出有效事务点或者时间点的历史快照数据。

- 支持从 OceanBase 的备副本中导出数据。（注：区别于备集群）
- 支持指定自定义的查询语句，仅导出该查询语句的结果集。
- 支持通过最新的快照版本以不锁表的方式导出全局一致的数据。
- 支持对命令行中指定的敏感参数进行加密。包括：数据库的账号密码，云存储的账号密钥。

## 快速上手

### 步骤一：下载软件

提示：导入导出 4.2.1 及以后的版本，不再区分商业版和社区版。用户可以从社区下载中心获取软件包。

下载最新版本的导入导出软件包并解压：

```
$ unzip ob-loader-dumper-4.2.1-RELEASE.zip
$ cd ob-loader-dumper-4.2.1-RELEASE
```

### 步骤二：配置运行环境

提示：您本地的环境必须安装 Java 8+，并配置 JAVA\_HOME。强烈建议 1.8.0\_3xx 以上版本。

Linux操作系统，请您编辑 {ob-loader-dumper}/bin/obloader 和 {ob-loader-dumper}/bin/obdumper 运行脚本。该步骤主要是让用户学会调整 JVM 参数。JVM 内存太小可能会影响导入导出的性能，甚至影响导入导出功能的稳定性。例如：Full GC 或者 GC Crash。强烈建议 JVM 内存（默认：-Xms4G -Xmx4G）调整至机器可用内存的 60%。擅长 Java 性能调优的用户可以按需调整 JAVA\_OPTS 选项中的 JVM 参数。

```
JAVA_OPTS="$JAVA_OPTS -server -Xms4G -Xmx4G -XX:MetaspaceSize=128M -
XX:MaxMetaspaceSize=128M -Xss352K"
JAVA_OPTS="$JAVA_OPTS -XX:+UnlockExperimentalVMOptions -XX:+UseG1GC -
Xnoclassgc -XX:+DisableExplicitGC -XX:MaxGCPauseMillis=200 -
XX:+HeapDumpOnOutOfMemoryError"
```

说明：Windows 操作系统，请您编辑 bin/windows/obloader.bat 和 bin/windows/obdumper.bat 运行脚本。

### 步骤三：准备数据

提示：用户试用导出功能时，您无需准备任何数据，可以直接跳到步骤四。用户试用导入功能时，您可以使用已有的数据文件，也可以使用 tpc-h 工具临时生成一些数据文件。但是，您需要了解文件的内容的格式。如何识别文件中的数据格式？您可以阅读这篇文章《准备好您的数据了吗？》。

### 步骤四：创建数据库

提示：用户试用导出功能时，您需要在创建数据库以后，继续创建表并插入数据。

1. 使用 OCP 管控平台或者 OBD 命令行工具搭建 OceanBase 集群；
2. 创建测试数据库；

### 3. 创建测试表并插入数据；（导入时可选）

## 步骤五：试用导入导出

### 1. 导入数据示例

```
./obloader -h 'IP地址' -P'端口' -u'用户' -t'租户' -c'集群' -p'密码' -D'库名' --table '表名' --csv -f '文件路径' --sys-password '系统租户密码' --external
```

说明：示例仅导入数据不包含模式。想了解导入工具的命令行选项，可以查阅文档中——[【命令行选项】](#) 章节。

### 2. 导出数据示例

```
./obdumper -h 'IP地址' -P'端口' -u'用户' -t'租户' -c'集群' -p'密码' -D'库名' --table '表名' --csv -f '文件路径' --sys-password '系统租户密码' --skip-check-dir
```

说明：示例仅导出数据不包含模式。想了解导出工具的命令行选项，可以查阅文档中——[【命令行选项】](#) 章节。

## 步骤六：恭喜！！

您已成功完成导入导出快速试用！！！ 想要了解更多的信息，我们建议您跟着以下的步骤进行学习：

- 阅读产品简介，了解导入导出的原理与功能，以及它与其他工具的比较。请参阅官方文档 [《OceanBase导数工具》](#)。
- 欢迎加入 [OceanBase 社区](#)，我们一起和 OceanBase 的研发人员在线上讨论导入导出的问题、需求，以及对于未来的规划。

## FAQ

Q1: 启动 obloader/obdumper 时报错: Access denied for user 'root'@'xxx.xxx.xxx.xxx' <br>

A1: --sys-password 选项指定正确的root@sys用户的密码。

Q2: 运行 obloader 过程中报错: Over tenant memory limits <br>

A2: 调大租户内存（或者 指定 --thread 降低并发数）。

Q3: 运行 obloader 过程中报错: No memory or reach tenant memory limit <br>

A3: 调大租户内存（或者 指定 --thread 降低并发数）。

Q4: obdumper 指定 --query-sql '大查询语句' 导出数据过程中报错: Connection reset  
<br>  
A4: obproxy 配置参数 client\_tcp\_user\_timeout, server\_tcp\_user\_timeout 设为0即可。

Q5: 如何在调试模式下运行 obloader/obdumper 以便于排查一些问题? <br>  
A5: 直接运行 bin 目录下的调试脚本。例如: obloader-debug/obdumper-debug

Q6: 运行 obloader 导入数据时, 报错: No tables are exists in the schema: "xxx"  
<br>  
A6: --table 选项指定的是数据库中已经定义的表名。

Q7: 运行 obloader 导入数据时, 报错: The xxx files are not found in the path: "xxx" <br>  
A7: 要求数据文件的名称与表名相同且大小写要一致。mysql默认表名是小写, oracle默认表名是大写。

Q8: 如何使用 obloader 导入与表不同名的数据文件? <br>  
A8: -f 选项可以直接指定一个数据文件。例如: --table 'test' -f '/output/hello.csv'

Q9: 为表配置控制文件, 导入或者导出数据时为何没有生效? <br>  
A9: 要求控制文件的名称与表名相同且大小写要一致。mysql默认表名是小写, oracle默认表名是大写。

Q10: 运行 obdumper 导出数据时, 报错: The target directory: "xxx" is not empty  
<br>  
A10: 为了防止数据覆盖, 导出前 obdumper 会检查输出目录是否为空。--skip-check-dir 选项可以跳过此检查。

Q11: 运行 obdumper 导出数据时, 为什么空表没有产生 (空的) 数据文件? <br>  
A11: 默认空表不会产生对应的空文件。--retain-empty-files 选项可以保留空表所对应的 (空的) 文件。

Q12: 运行 obloader 导入数据时报错: The manifest file: "xxx" is missing <br>  
A12: 元数据文件是 obdumper 导出数据时产生的。--external-data 选项可以跳过元数据文件

的检查。

Q13: 运行 obloader/obdumper 脚本时, 命令行参数没有被正常解析的原因是什么? <br>

A13: 可能是不同的运行平台所加的引号不同或者参数没有加引号, 导致解析错误。(见Q14)

Q14: 运行 obloader/obdumper 脚本时, 什么时候需要对参数加单/双引号? <br>

A14: Windows 平台参数使用双引号, 例如: `--table "\*`"; 类Linux 平台参数使用单引号, 例如: `--table '*'`。