

OSYNC for DB 用户使用手册 v3.8.0.1

目录

OSYNC for DB 用户使用手册 v3.8.0.1	1
1. OSYNC 概述	2
2. 产品新特性	3
3.8.0.1 版本	3
3. 安装 OSYNC	4
1. 解压缩 osyncwin3.8.0.X 压缩文件	4
2. 安装数据库客户端驱动程序	5
4. 使用基础	6
1. 目标端导入准备工作	6
2. 导入步骤	15
3. 数据对比	18
4. 导出到文本文件	19
5. 导出到 Kafka	19
5. 使用限制	22
6. 常见问题	24
1. 导入到 ORACLE	24
2. 导入到 Sybase ASE	25
3. 导入到 Sybase IQ	26
4. 从 Teradata 导出	26
5. 导入到 Teradata	26
7. 附录	27
OSYNC 命令行工具	27

1. OSYNC 概述

OSYNC 是一款高速、易用、且支持异构数据库的全量数据迁移工具，极大地简化了数据库间的数据导出/导入工作。

对于 ORACLE 数据库间数据迁移，OSYNC 采用**独有的**完全直接路径导出、导入技术，实际测试导入速度可达到 20 万行/秒。

对于异构数据库间数据迁移，支持 ORACLE/SQL Server/MySQL/Postgre/Sybase(ASE,ASA,IQ)/DB2/SAP HANA/Teradata/Tibero 数据库互导，充分利用了本数据库的批量导入特性，实际测试导入到 ORACLE/SQL Server/MySQL/Postgre/HANA/Sybase/Tibero 可达到 2 万行/秒。

另外，支持导出到 MongoDB 数据库、Kafka 和文本文件。



功能特点：

➤ 高速导出、导入

- ORACLE 数据库间的导出/导入均采用直接路径方式，以达到最佳性能。依据多个真实客户案例，在 I/O 和网络充足的情况下，导出、导入速度至少能提高 3-5 倍，测试导入速度可达到 20 万行/秒，取决于数据库负载和硬件性能。
- 异构数据库之间的导入操作，针对各数据库分别做了特定优化，实际测试导入到 ORACLE、SQLServer、MySQL、PostgreSQL、Sybase、HANA、Tibero 可达到或超过 2 万行/秒。

➤ 极大减少了对数据库本身的影响

- ORACLE 间的导出/导入，导出时避开了 SQL 接口，这对导出库数据缓存影响几乎为零；导入时采用 NOLOGGING 方式，避免了导入库大量的 UNDO、REDO 生成。如果中途异常中断，也不会产生回滚。
- 其它数据库间的同构、异构导出/导入，也充分利用各数据库的批量导入特性，最大可能地减少性能影响。

➤ **全平台支持**

- ORACLE 数据库支持 10g 及以上各平台、各版本间的直接路径导出/导入，9i 版本仅做有限支持。
- SQL Server/MySQL/PostgreSQL/Sybase ASE/Sybase IQ/DB2/HANA/Teradata/Tibero 支持各主流版本间同构、异构导出/导入。

➤ **安装、配置简单**

- OSYNC 图形界面开箱即用，无需单独安装 ORACLE/MySQL/Postgre 客户端。
- OSYNC 命令行界面可运行于任意一台安装有 ORACLE 客户端软件的 UNIX、Linux 机器，配置简单。

2. 产品新特性

3.8.0.1 版本

1. 支持 Tibero 数据库。
2. ORACLE 数据库间，单表导出，导入支持并行。

3. 安装 OSYNC

OSYNC 系统运行环境如下：

➤ **操作系统**

Windows 7 及以上 64 位版本。

➤ **数据库**

数据库类型	支持版本
ORACLE	9i 及以上版本
Sybase IQ/ASA	16 及以上版本
PostgreSQL	7.1 及以上版本
SQL Server	2005 及以上版本
Sybase ASE	12 及以上版本
MySQL	4.0 及以上版本
DB2	8.0 及以上版本
SAP HANA	所有常见版本
Teradata	所有常见版本
Tibero	所有常见版本

1. 解压缩 osyncwin3.8.0.X 压缩文件

从 3.7 版本起，OSYNCWIN 程序采用绿色安装方式，直接解压缩文件至非中文路径，手工创建 osyncwin.exe 快捷方式至桌面即可。



2. 安装数据库客户端驱动程序

数据库	是否需要安装客户端
ORACLE MySQL PostgreSQL MongoDB	无须安装客户端。
SQLServer	<ol style="list-style-type: none"> 1. 点击控制面板->数据源(ODBC)，进入 ODBC 数据源管理器，点击添加，确认“SQL Server Native Client ##.0”存在。 2. 如果未安装 NativeClient，请进入<OSYNC>安装目录/db_client/下，安装 sqlncli.msi，该程序为 SQL Server 2008 数据库驱动。
Sybase ASE	<ol style="list-style-type: none"> 1. 从完整安装程序或单独的 ASE client 安装程序拷贝到本地，解压缩，再安装 ASE OLE DB Driver, ASE ODBC Driver, Open Client。 2. 安装完后，ODBC 数据库配置中，点击新建按钮，确认 “Adaptive Server Enterprise”存在。
Sybase IQ/ASA	<p>请安装 IQ 或 SQL Anywhere 16 及以上版本客户端，否则无法导入。</p> <ol style="list-style-type: none"> 1. 从完整安装程序或单独的 IQ client 安装程序拷贝到本地，解压缩，再安装 ODBC Driver。 2. 安装完后，ODBC 数据库配置中，点击新建按钮，确认 “Sybase IQ”或“SQL Anywhere ##”存在。
DB2	<ol style="list-style-type: none"> 1. 下载 ibm_data_server_runtime_client_win64_v11.1.exe，双击安装。 2. 安装完后，ODBC 数据库配置中，点击新建按钮，确认 “IBM DB2 ODBC DRIVER”存在。 <p>注意： 如果之前手工添加过 ODBC 驱动(例如 v10.5fp8_ntx64_odbc_cli.zip)，可能会产生冲突，导致无法使用。</p>

	<p>需要先卸载之前手工安装的*odbc_cli*.zip ODBC 驱动，再安装上述程序。</p> <pre>cd <旧安装目录>/bin Install: db2oreg1.exe -i Uninstall: db2oreg1.exe -u db2cli install -setup db2cli install -cleanup</pre>
SAP HANA	<ol style="list-style-type: none"> 1. 下载 clients_windows.zip，从目录 HDB_CLIENT_WINDOWS_X86_64 安装 64 位客户端程序。 2. 安装完成后，通过 ODBC 数据源，确认"HDBODBC"存在。
Teradata	<ol style="list-style-type: none"> 1. 下载 TeradataODBC__windows_indep.16.#.zip，安装 64 位客户端程序。 2. 安装完成后，通过 ODBC 数据源，确认"Teradata Database ODBC Driver #.#"存在。
Tibero	<ol style="list-style-type: none"> 1. 下载 Tibero6_client_installer_*.exe，安装 64 位客户端程序。 2. 安装完成后，通过 ODBC 数据源，确认"Tibero * ODBC Driver"存在。

4. 使用基础

1. 目标端导入准备工作

1) 目标端创建数据库和用户

各类数据库软件创建数据库和用户命令不尽相同，请参阅数据库安装文档。下面着重说明 ORACLE 数据库下导出、导入用户需要的最小权限。

ORACLE 创建用户

建议 ORACLE 导出/导入用户赋予 dba 角色，如果由于数据库安全策略，不允许赋予 dba 角色，请手工赋予以下最小权限：

1) 导出用户：

```
create user andy_exp identified by andy_exp;
grant create session,
      alter session,
      select_catalog_role,
```

```
select any table,  
lock any table --used by direct path  
to andy_exp;  
grant execute on sys.dbms_flashback to andy_exp; --This must run by SYS
```

2) 导入用户:

```
create user andy_imp identified by andy_imp;  
grant create session,  
alter session,  
select_catalog_role,  
select any table,  
lock any table, --used by direct path  
unlimited tablespace,  
insert any table,  
update any table,  
delete any table,  
create any table,  
alter any table,  
drop any table,  
create any cluster,  
alter any cluster,  
drop any cluster,  
create any index,  
alter any index,  
drop any index  
to andy_imp;
```

Teradata 创建用户

```
CREATE USER andy [from database_name] as  
PASSWORD = andy  
PERMANENT = 1000000000 BYTES  
TEMPORARY = 100000000 BYTES  
SPOOL = 100000000 BYTES;  
grant select on dbc to andy;
```

2) 查看数据库字符集

导出、导入数据库的字符集必须兼容，否则无法兼容的字符将会变成乱码。常见的数据库中文字符集有 UTF8, GBK。

ORACLE

ORACLE 的数据库字符集，在创建数据库时指定，如 ZHS16GBK, UTF8 等。默认表、列名不区分大、小写(表和列名内部以大写存储)，除非使用""限定。

1. 查看数据库字符集

```
select property_value
  from database_properties
 where property_name = 'NLS_CHARACTERSET';
```

2. 更改数据库字符集

```
startup upgrade;
ALTER DATABASE character set INTERNAL_USE UTF8;
shutdown immediate;
startup;
```

说明：INTERNAL_USE选项为强制更改字符集。

SQLServer

对于 SQLServer 来说，数据库字符集和大小写是否敏感，都由排序规则(Collation)决定。比如 Chinese_PRC_CI_AS 使用 GBK 编码，并且表(列)名不区分大、小写。

1. 查看排序规则

1) 查看整个服务器排序规则

```
select SERVERPROPERTY('Collation');
sp_helpsort
```

2) 查看各数据库的排序规则

```
select database_id, name as db_name, collation_name
  from sys.databases;
```

2. 更改数据库排序规则

```
use master
```


go

ALTER DATABASE 数据库名 COLLATE 排序规则名

示例: ALTER DATABASE testdb COLLATE Chinese_PRC_CI_AS

建议:

1. 创建数据库时, 使用不区分大、小写排序规则。如果必须区分大小写, 强烈建议创建表时表名和表名都为小写。
2. 如果想使用 Unicode 字符集, 建议把 char/varchar/text 改为 nchar/varchar/ntext。

MySQL

MySQL 的数据库字符集, 在创建数据库时指定, 如 **gbk**, **utf8** 等。默认表、列名不区分大、小写(表和列名内部以小写存储), 除非用``限定。

1. 查看服务器支持的字符集列表

```
select * from information_schema.character_sets;  
show character set;
```

2. 查看数据库字符集

```
SHOW CREATE DATABASE <db_name>;
```

3. 更改数据库字符集

```
ALTER DATABASE <db_name> CHARACTER SET utf8;
```

PostgreSQL

PostgreSQL 的数据库字符集, 在创建数据库时指定, 如 **euc_cn**, **utf8** 等。默认表、列名不区分大、小写(表和列名内部以小写存储), 除非用""限定。

1. 查看数据库字符集

```
show server_encoding;
```

2. 更改数据库字符集

无。

Sybase ASE

对于 ASE 来说, 数据库字符集由排序规则(Collation)决定。

1. 查看排序规则

1) 查看整个服务器排序规则

sp_helpsort

2. 更改数据库排序规则

```
select id,name from syscharsets;
```

```
sp_configure "default character set id",190 //UTF8
```

Sybase IQ

查看当前数据库字符集: `SELECT DB_PROPERTY('CharSet'), db_property('collation');`

常见的中文字符集有 936zho(gbk), utf8

DB2

列出所有数据库: `db2 list db directory`

查看字符集: `db2 get db cfg for <db_name>`

MongoDB

MongoDB 内部以 UTF8 存储。

SAP HANA

HANA 内部以 UTF8 存储。

Teradata

服务器字符集可以以 LATIN 或 UNICODE 存储。

查看字符集:

1. 查看用户默认字符集

```
SELECT UserName, PermSpace, TempSpace, DefaultCharType  
FROM DBC.users;
```

2. 查看表 varchar 列字符集:

```
select ColumnName, CharType from dbc.columns  
where tablename = 't_varchar';
```

说明:

CharType 值	字符集	说明
1	Latin	Ascii 字符集, 默认
2	Unicode	16 位 Unicode 字符集
3	KANJISJIS	日文字符集
4	Graphic	DB2 兼容字符集

更改用户默认字符集为 **UNICODE**:

```
SQL> modify user ANDY as DEFAULT CHARACTER SET UNICODE;
```

3) 目标端创建表结构

目标端需要先建立表结构, 可以手工创建 SQL, 或借助 OSYNCW 工具的"生成 DDL"按钮生成基本的表、索引、约束, 保存 SQL 文件后适当更改内容。

关于表、列名大小写

目标端创建表时, 需要注意表(或列)名大、小写敏感的情况。

表名、列名大小写规范:

1. 创建表时表名和列名统一以小写格式。
2. 表名或列名请勿添加""等限定符, 以免大小写敏感。

数据库类型	表名是否大小写敏感	列名是否大小写敏感	表(列)名限定符
ORACLE PostgreSQL DB2 SAP HANA	否	否	"" , 会导致大、小写敏感
MySQL	Windows: 否 Linux: 是 由变量 lower-case-table-names 控制, show variables like '%lower_case_table_names%';	否	``
SQL Server Sybase ASE	取决于排序规则 Chinese_PRC_CI_AS: CI 为忽略大小写 Chinese_PRC_CS_AS: CS 为大小写敏感	取决于排序规则	[]

Teradata	否	否	""，限定符可以使关键字作为列名。
Sybase IQ/ASA	否	否	""或[]，限定符可以使关键字作为列名。

各数据库列类型映射

ORACLE	POSTGRESQL	SQL SERVER	MYSQL
NUMBER(5,0)	SMALLINT	SMALLINT	SMALLINT
NUMBER(10,0)	INTEGER	INT/INTEGER	INT
NUMBER(24,0)	BIGINT	BIGINT	BIGINT
NUMBER(10,4)	DECIMAL(P,S)	SMALLMONEY	DECIMAL(P,S)
NUMBER(19,4)	DECIMAL(P,S)	MONEY	DECIMAL(P,S)
NUMBER	DECIMAL(P,S)	DECIMAL(P,S)	DECIMAL(P,S)
BINARY_FLOAT	BINARY_FLOAT	REAL/FLOAT	REAL
BINARY_DOUBLE	DECIMAL(P,S)	DOUBLE	DOUBLE(P,S)
TIMESTAMP	TIMESTAMP(P) WITHOUT TIME ZONE	DATETIME	TIMESTAMP
TIMESTAMP WITH TIME ZONE	TIMESTAMP(P) WITH TIME ZONE	DATETIME	TIMESTAMP
DATE	DATE	DATETIME	DATETIME
CHAR(N)	CHAR(N)	CHAR(N)	CHAR(N)
VARCHAR(N)	VARCHAR(N)	VARCHAR(N)	VARCHAR(N)
CLOB	TEXT/ VARCHAR(N)	VARCHAR(MAX) TEXT	LONGTEXT
BLOB	BYTEA(N)	VARBINARY(MAX) IMAGE	LOB
RAW	BYTEA(N)	VARBINARY(N)	VARBINARY(N) BLOB

ORACLE	HANA	SYBASE/IQ	TERADATA
NUMBER(5,0)	SMALLINT	SMALLINT	SMALLINT
NUMBER(10,0)	INTEGER	INTEGER	INTEGER

NUMBER(24,0)	BIGINT	BIGINT	BIGINT
NUMBER(10,4)	DECIMAL(10,4)	SMALLMONEY	DECIMAL(10,4)
NUMBER(19,4)	DECIMAL(19,4)	MONEY	DECIMAL(19,4)
NUMBER	DECIMAL(P,S)	DECIMAL(P,S)	DECIMAL(P,S)
BINARY_FLOAT	REAL	FLOAT	FLOAT
BINARY_DOUBLE	DOUBLE	DOUBLE	DOUBLE
TIMESTAMP	TIMESTAMP	DATETIME	TIMESTAMP
TIMESTAMP WITH TIME ZONE	TIMESTAMP	DATETIME	TIMESTAMP
DATE	SECONDDATE	DATETIME	TIMESTAMP
CHAR(N)	CHAR(N)	CHAR(N)	CHAR(N)
VARCHAR(N)	VARCHAR(N)	VARCHAR(N)	VARCHAR(N)
CLOB	CLOB	TEXT	CLOB
BLOB	BLOB	IMAGE	BLOB
RAW	VARBINARY(N)	VARBINARY(N)	VARBYTE(N)

ORACLE

expdp+impdp 创建表结构

```
SQL> create or replace directory OSYNC_DIR as '/home/andy';
```

1)源库导出定义

```
--导出用户下所有对象，不包括表数据
```

```
$expdp andy/andy directory=OSYNC_DIR SCHEMAS=ANDY,ANDY2  
content=METADATA_ONLY dumpfile=expdp_andy.dmp
```

```
--导出用户下的表和索引，不包括表数据
```

```
$expdp andy/andy directory=OSYNC_DIR SCHEMAS=ANDY,ANDY2  
content=METADATA_ONLY INCLUDE=table,cluster dumpfile=expdp_andy.dmp
```

```
--导出单张表
```

```
$expdp andy/andy directory=OSYNC_DIR TABLES=t_test,t_iot,t_cluster  
content=METADATA_ONLY dumpfile=expdp_andy.dmp
```

2)目标库导入定义

```
--导入用户对象
```

```
$impdp andy/andy directory=OSYNC_DIR full=y dumpfile=expdp_andy.dmp
```

导入可选项:

```
table_exists_action=replace
```

```
REMAP_SCHEMA=ANDY:ANDY5 REMAP_TABLESPACE=USERS2:USERS
```

注意:

1. Nested、temporary 等表不会被 osync 复制。
2. ORACLE 的 LONG 类型仅做有限支持。

4) 删除目标端表上索引(建议)

建议删除目标端索引，以提高导入性能。

5) 目标端禁用表上外键、触发器

1. ORACLE

生成禁用触发器脚本:

```
select 'alter trigger '||x.owner||'.'||x.trigger_name||' disable;'
from dba_triggers x
where status = 'ENABLED'
and table_owner in ('ANDY');
```

生成禁用外键约束脚本

```
select 'alter table '||owner||'.'||table_name||' disable constraint
' ||constraint_name||';'
from dba_constraints t
where constraint_type = 'R'
and status = 'ENABLED'
and owner = 'ANDY';
```

6) 目标端禁用作业

目标端建议禁用所有业务相关 JOB，以免表被意外更改。

1. ORACLE

1. Scheduler job

查看所有 scheduler job:

```
select owner, job_name, job_action, enabled, state
  from dba_scheduler_jobs
where enabled='TRUE';
```

禁用脚本:

```
select 'execute dbms_scheduler.disable('||''''||owner||'.'||job_name||''''||');'
  from dba_scheduler_jobs
where enabled='TRUE';
```

2. 普通 Job

```
select job, log_user, schema_user, what, broken, last_date, next_date
  from dba_jobs
where broken = 'N';
```

禁用脚本 (SYS 执行):

```
select 'execute dbms_ijob.broken('||job||', true);'
  from dba_jobs
where broken = 'N';
```

禁用完需要 commit;

2. 导入步骤

1) 配置数据库连接

运行 osyncwin, 点击<数据库连接>按钮。



目前支持 ORACLE/SQL Server/MySQL/Postgre/Sybase/HANA 数据库之间的数据互导。

输入连接名，并提供数据库的<主机名/IP, 端口#, 数据库(服务)名, 用户名, 密码>配置信息，点击<测试>按钮可以测试连通性，然后点击保存。

特定数据库说明：

1. HANA

可用以下 SQL 查询 HANA 中的 SQL 端口号和数据库列表

```
select * from m_services;
```

```
select * from m_databases;
```

在 HANA Express 2 中，可用 SYSTEM 用户连接以下端口和数据库：

端口#	数据库
39013	SYSTEMDB
39013	HXE
30015	HXE

2) 配置复制用户或表

源端：选择连接字符串，点击<连接>按钮，选择复制类型，并选择复制用户或复制表。

目标端：选择连接字符串，如果用户名不同，可以映射到不同用户中。

点击<保存>按钮，可以保存当前配置。



*可以使用表过滤（WEHRE）条件，来导出指定条件的表数据。

如果同一张表指定不同的条件， 可以当作手工并行来导出/导入，此时需要把主界面的清空目标表选项去掉。



小技巧:

1. 选择多个复制表时，如果表数太多界面选择不方便，可以保存当前配置，手工更改配置文件。

具体步骤:

1. 打开 osyncwin 的安装目录，进入 conf 文件夹。
2. 编辑配置文件，如 temp5.conf。
3. 在<rep_table>标签中手工添加复制表列表。

```
<rep_table>
```

```
<row> <owner>ANDY</owner><table_name>T_VARIABLE</table_name> </row>
```

```
</rep_table>
```

3) 导入到目标库

说明:

源端导出时, 可以指定 SCN# (仅限 ORACLE) 和并行数。

目标导入时, 可以指定是否使用直接路径加载 (仅限 ORACLE) 和是否清空目标表。

点击<启动>按钮启动导入。



4) 后续工作(ORACLE->ORACLE)

注意: ORACLE->ORACLE 为加快导入速度, 直接路径方式导入前会使目标端索引失效, 导入后请重建目标端索引。

目标端:

```
SQL>alter index XXX rebuild;
```

说明: 分区索引需要对单个分区做 rebuild。

3. 数据对比

选择比对表, 点击启动。

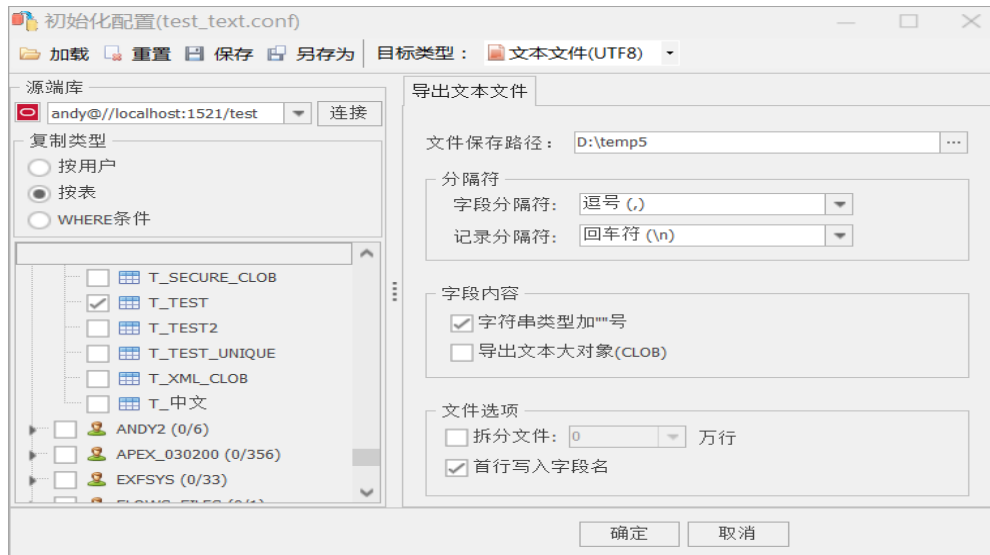
注意: 由于表比对采用 `SELECT * FROM 表名 ORDER BY PK_COL;`

异构数据库或同构数据库但字符集不同时, 由于排序规则不同, 可能出现比对不准。



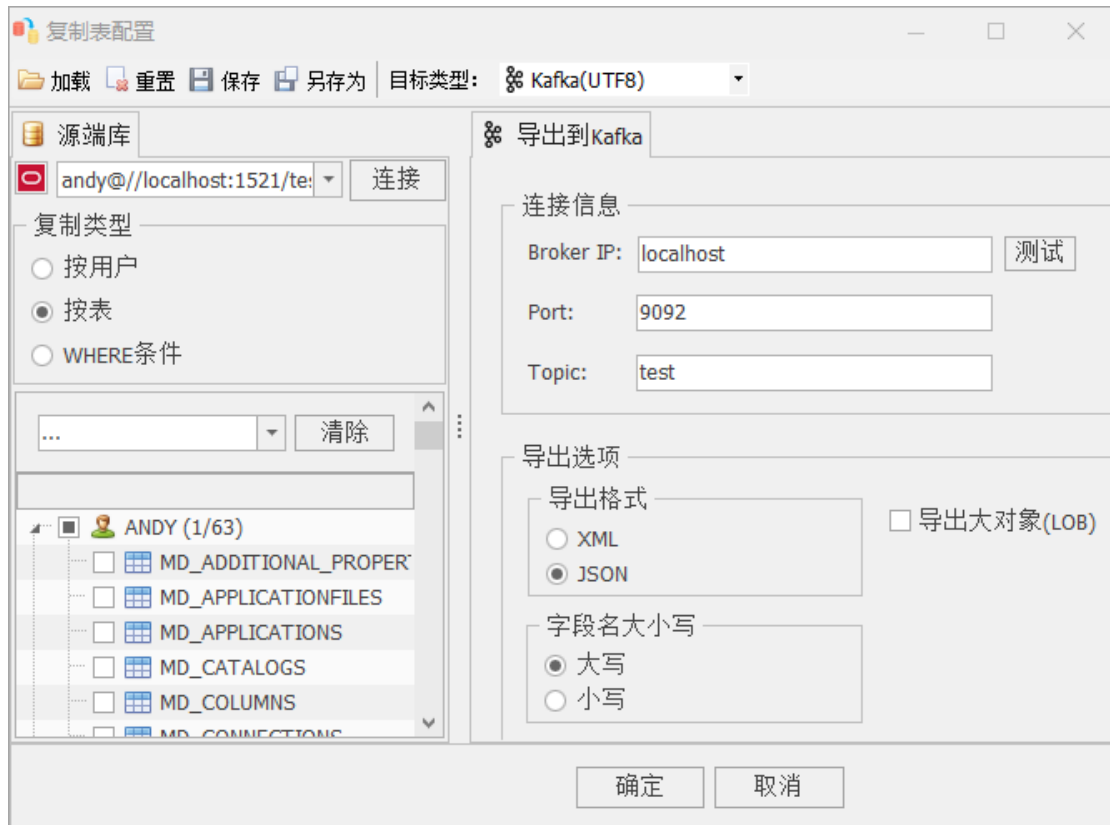
4. 导出到文本文件

OSYNC 支持数据库表的数据导出成文本文件(UTF8 格式)。



5. 导出到 Kafka

OSYNC 支持数据库表的数据导出至 Kafka 消息队列。



导出选项

导出选项	说明
导出格式	支持每行导出成 UTF8 编码的 XML 和 JSON 字符串。
导出大对象	导出 CLOB 和 BLOB，BLOB 会导出成 BASE64 编码后的字符串。
字段名大小写	指定 XML 或 JSON 字符串的列名以大写方式或小写方式。

数据格式映射

列类型	ORACLE 列类型	JSON	XML
字符串	CHAR VARCHAR2 CLOB	"字符串"	字符串
数值	NUMBER BINARY_FLOAT	数值	数值

	BINARY_DOUBLE		
日期	DATE	"yyyy-MM-dd hh:nn:ss"	yyyy-MM-dd hh:nn:ss
	TIMESTAMP	"yyyy-MM-dd hh:nn:ss.zzz"	yyyy-MM-dd hh:nn:ss.zzz
二进制	RAW	"BASE64 编码后的字符串"	BASE64 编码后的字符串
	BLOB		

测试步骤

1. 启动 Kafka

1)启动 zookeeper

```
bin\windows\zookeeper-server-start.bat config\zookeeper.properties
```

2)启动 kafka

```
bin\windows\kafka-server-start.bat .\config\server.properties
```

3)创建、查看 topic

```
bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
```

```
bin\windows\kafka-topics.bat -zookeeper localhost:2181 --list
```

```
bin\windows\kafka-topics.bat --describe --zookeeper localhost:2181 --topic test
```

4)查询 topic 中消息数

```
bin\windows\kafka-consumer-groups.bat --bootstrap-server localhost:9092 --list
```

```
bin\windows\kafka-consumer-groups.bat --bootstrap-server localhost:9092 --describe --group console-consumer-90679
```

5)启动消费者进程

```
.\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
```

2. 测试导出表数据至 kafka。

5. 使用限制

ORACLE

1. 不支持的列类型
 - 1) ORACLE 10G 及以上版本支持直接路径方式，不支持 binary xml 存储格式的 xmltype 类型。
 - 2) 9i 数据库采用支持 bulk 方式，不支持 LONG 类型、UDT、XMLTYPE。
2. 不支持的表类型
不支持聚簇(cluster)表。
3. 导入时，目标表必须禁用触发器、外键约束。如果目标表有全文索引，需要删除掉全文索引后，再导入。
4. 表名限制
 - 1) 暂时不支持中文表名、列名。
 - 2) 大、小写敏感的表名，需要在表名前后加""，如 andy."abc"。

Sybase IQ

1. CLOB 导入暂不支持>32k 的 LOB 值。
错误信息：Function not supported on varchars longer than 32767 Bind host variable。
2. 不支持 BLOB。
3. RAW 类型导出时，末尾会添加多余的 00。

Teradata

1. LOB 导入暂不支持>32k 的 LOB 值。

PostgreSQL

1. 不支持的列类型
导入时不支持 bit, varbit, interval 类型。

Tibero

Tibero 采用直接路径加载(direct path load)方式，有以下限制：

1. LOB 表单行>16M 时，导入会失败。

6. 常见问题

1. 导入到 ORACLE

1. 导出/导入时报 ORA-39773 或 ORA-39774 错误

错误日志:

1)ORA-39773: parse of metadata stream failed SQL:[]

2)ORA-39774: parse of metadata stream failed with the following error:

LPX-00210: expected '<' instead of 'i' SQL:[]

原因:

数据库字符集和当前 NLS_LANG 不兼容, 执行 osync.sh 前需要手工指定 NLS_LANG

例如: 数据库字符集为 WE8MSWIN1252, 导出时可能会遇到上述问题。

解决方法:

手工指定 NLS_LANG

```
export NLS_LANG=.WE8MSWIN1252
```

```
./osync.sh
```

注意, 有的平台不支持设置成 WE8MSWIN1252, 可以设置成 ZHS16GBK。

2. 连接 12.2 数据库时报 ORA-28040 错误

```
D:\app\oracle\product\11.2.0\dbhome_32\bin\sqlplus andy/andy@主机 IP:1521/db12201
```

ERROR:

ORA-28040: No matching authentication protocol

或 ORA-01017: invalid username/password; logon denied

原因: 12c 的客户端连接版本限制, 参见 id:207303.1。

解决方法: 客户端必须使用 11.2.0.3 及以上版本。

3. 表导出过程中报 ORA-03146: invalid buffer length for TTC field

原因: 导出时内部 buffer 超出限制。

解决方法: 减小参数 buffer_size, 可选值: 524288(512k), 1024000(1000k), 16777216(16m)

4. 表导入过程中报 ORA-26086: direct path does not support triggers

原因: 导入表上有触发器。

解决方法: 导入前需要把表上所有的触发器禁用掉, alter trigger XXX disable;

5. OSYNC 启动后 osync.log 报 ORA-00904 错误

[Source DB] Failed to get current scn, exit. ORA-00904::invalid identifier

原因：源端连接用户需要有 dbms_flashback 执行权限

解决方法：grant execute on dbms_flashback to <DB_USER>;

6. 导入时直接报 Unknown error: ORA-65535

原因：可能表名中存在小写列名。

解决办法：目标端创建表、列名时，一律用大写名称。

7. 导入时报 ORA-26085: direct path operation must start its own transaction

原因：目标端表上建有全文索引(INDEXTYPE IS ctxsys.context)

解决办法：删除目标端表上的全文索引，再导入。

drop index owner.index_name;

8. 控制台打印 KPEDBG_HDL_POP_FCPTR ORA-24550 错误

错误日志：

OSYNC> KPEDBG_HDL_POP_FCPTR

ORA-24550: signal received: Unhandled exception: Code=ebf00baa Flags=0

原因：11g 及以上版本 oci 多线程访问导致。

解决方法：禁用 ORACLE 客户端 DIAG。

更改\$ORACLE_HOME/network/admin/sqlnet.ora 文件，添加以下项：

DIAG_ADR_ENABLED= OFF

DIAG_SIGHANDLER_ENABLED= FALSE

DIAG_DDE_ENABLED= FALSE

9. 导出表类型：

注意：导出表时请去除临时表、二级表、嵌套表、expdp 导出表、全文检索索引表：

and temporary = 'N'

and secondary = 'N'

and nested = 'NO'

and table_name NOT like 'SYS_EXPORT_TABLE%'

and table_name NOT like 'DR\$%'

2. 导入到 Sybase ASE

1. 导入过程中报错 sybase type 'e' not implemented

原因：有可能 sybase 数据库设备空间满，请查看、添加空间。

解决办法：Sybase Central->数据库->[数据库名]右键->属性，查看<设备>和<使用情况>页，空间不足时增加本数据库空间。

2. 建议数据库级别打开 bulk 选项

方法：

```
use master
go
sp_dboption test, "select into/bulkcopy",true
go
use test
go
checkpoint
go
```

3. 导入到 Sybase IQ

1. 表导入时，报错 **You have run out of space in IQ_SYSTEM_MAIN DBSpace.**

原因：表空间不足，需要加大表空间。

2. 导入结束查看目标数据时，如果导入操作前已打开查询窗口，需要先 commit 再查询输出结果。

4. 从 Teradata 导出

1. 表导出时，报错 **No more spool space in <user name>**

原因：用户的 spool space 空间无法容纳该表，teradata 可能需要把整张表的输出放到 spool space 中。

解决办法：增大用户的 spool space 大小。

```
SELECT UserName, PermSpace, TempSpace, DefaultCharType
FROM DBC.users;
modify user ANDY as DEFAULT TEMPORARY = 1000000000 BYTES;
```

5. 导入到 Teradata

7. 附录

OSYNC 命令行工具

OSYNC 命令行工具支持各 Linux/Unix 平台，以命令行方式进行 ORACLE 数据库间高速导出/导入。

操作系统	操作系统版本	ORACLE 客户端版本
AIX	>=5.2	>=10.2.0.5
HP-UX IA/PA	>=11.23	>=10.2.0.1
Solaris	>=9	>=10.2.0.1
Linux	>=4	>=10.2.0.1

使用注意事项:

1. AIX 平台

OSYNC 使用的 Oracle 客户端库必须 >=10.2.0.5 版本，否则无法使用直接路径接口。如果 OSYNC 所在机器的 ORACLE 版本 <10.2.0.5，可以下载高版本的 instant client，并在 osync.sh 或 osynctl.sh 的启动脚本中指定该 ORACLE_HOME。

2. Unix/Linux 环境设置:

使用前请检查操作系统的 ulimit，确认 data 空间 >256M。

```
$ulimit -a
```

必要时加大 Shell 内存限制:

```
$ulimit -m unlimited
```

```
$ulimit -d unlimited
```

命令列表

类型	命令	说明
通用命令	init sync	从 table.list 读入复制表，并初始化复制状态
	start sync	启动复制，同时启动导出、导入线程。
	stop sync	中断复制操作
	start export	启动导出操作，导出表到本地文件，可供接下来的导入操作使用。
	start import	启动导入操作，把导出操作生成的文件，导入到目标端。
	Status	显示复制状态
	Help	显示命令列表
	quit exit	退出 osync
后台服务控制	start service	启动 osync 后台守护进程

	stop service	关闭 osync 后台守护进程
--	--------------	-----------------

文件说明

1. 安装文件

Osync 命令行安装文件格式为 osync*.tar.gz 文件，解压即可配置使用。

```
$gunzip osync*.tar.gz
```

```
$tar -xvf osync*.tar
```

```
$cd osync
```

2. osync 的文件列表

文件	说明
osync.sh	前台交互方式启动脚本，osync 以前台命令方式运行。 必要时请在脚本中指定以下环境变量： ORACLE_HOME ORACLE_SID NLS_LANG
osynctl.sh	后台守护进程方式运行时，控制脚本。
Osync	主执行文件，链接至 bin_file 下对应平台的二进制文件
Osynctl	控制命令，链接至 bin_file 下对应平台的二进制文件
conf/osync.conf	参数配置文件，配置源和目标端的连接参数等
conf/table.list	表复制清单，指定要复制的表列表
log 目录	复制时的日志文件所在目录
dump 目录	选择落地方式复制时，中间文件所在目录
bin_file 目录	存放各个 Unix/Linux 平台的可执行文件

3. osync.conf 文件说明：

配置项	说明	是否必选项
source	说明： 源端的连接字符串。 连接串格式： >=10g: username/password@ip:port/service_name <=9i: username/password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.1.2)(PORT=1521))(CONNECT_DATA=(SID=sid)))	是
target	说明： 目标端连接字符串	是

<i>flashback_scn</i>	说明：导出时使用的 flashback SCN 号，默认 0 取当前 scn。 设置场景：在和逻辑复制软件配套使用时必须指定参数。 select current_scn from v\$database;	否
<i>parallel</i>	说明：同时导出表数 设置场景：数据库机器性能较好时，可以适当加大并行导出线程数。	是
<i>tcp_port</i>	Osync 以后台服务启动时的监听端口号	是
<i>buffer_size</i>	说明：直接路径导出/导入缓冲区大小 设置场景：某些场景报 ORA-03146: invalid buffer length for TTC field 时，需要减小 <i>buffer_size</i> 。	是
<i>sync_method</i>	说明：指定导出/导入方法	是
	1 DIRPATH_DB: 直接路径不落地方式。 适用版本：10g 及以上版本。 支持列类型：所有 expdp 支持的类型。 适用场景：本地文件系统无充足空间导出，网络带宽充足。机器性能、磁盘、带宽充足时，该选项为首选项。	
	2 DIRPATH_FILE_DB: 直接路径落地方式。 适用版本：10g 及以上版本。 支持列类型：所有 expdp 支持的类型。 适用场景：源端数据库可能会导致 ORA-1555，须尽快地导出，且本地文件系统有充足空间的情况。	
	3 BULK_DB: bulk(array) select+直接路径加载，不落地方式。 适用版本：9i 版本。 支持列类型：本选项只支持基本数据类型和 LOB，不支持 LONG 和其它复合数据类型。另外，LOB 在导入时只能以 bulk(array) 方式加载。 适用场景：从 9i 导出到更高版本时只能用该选项。	

4. table.list 文件

示例：

```
#<source>          <target>
#<owner.table>    <owner.table> <parallel>
andy.t_test2      andy2.t_test2
```

5. log 文件列表

配置项	说明
<i>sync.log</i>	主日志文件
<i>sync_table.log</i>	表复制日志，包括表复制过程及错误日志
<i>sync_error.log</i>	表复制错误日志

最佳实践(前台交互方式)

1. 配置 osync.conf

参数名	参数值	说明
Source	andy/andy	设置源端连接字符串，源端连接用户需要有 dba 角色
Target	andy2/andy2	设置目标端连接字符串
flashback_scn	0	选择当前 scn 导出表。 注意： 和逻辑复制软件配合使用时需要手工指定 scn。
parallel	3	设置导出、导入线程数为 3。 注意： 机器性能较好时可以适当加大。
sync_method	1	采用直接路径不落地方式导出、导入。 注意： 请根据实际情况设置， 1 采用不落地方式 2 采用落地方式 3 bulk 不落地方式 具体说明参见参数说明章节。

2. 配置 table.list, 添加复制的表

```
#<source>      <target>
andy.test_blob  andy2.test_blob
```

注意事项:

如果在 windows 下编辑 table.list 文本文件，上传至 unix/linux 时，请使用文本格式上传，以免回车符不一致导致读取失败。

3. 目标端数据库建立表结构，建议先不建立索引和其它对象。

SQL>

1)获取表结构

```
set verify off
set heading off
set feedback off
set term off
set trimout on
set trimspool on
set pagesize 0
set linesize 2000
set timing off
set long 99999
spool ddl.sql
exec
```

```
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'STORAGE', false);
select dbms_metadata.get_ddl('TABLE', table_name, owner) ||';'
from dba_tables
where owner = 'ANDY';
spool off
```

说明：目标端表在导入之前会先 truncate 表数据，请确认 target 连接字符串设置正确。

4. 设置 NLS_LANG 环境变量

```
SQL>SELECT trim(property_value) from database_properties where property_name='NLS_CHARACTERSET';
$export NLS_LANG=zhs16gbk
$export ORACLE_SID=testdb
```

说明：上述 ORACLE 环境变量可加入到 osync.sh 文件中。

特别注意：

- 1) 采用 BULK_DB 导出时，必须设置 NLS_LANG 环境变量，以免导出成乱码。
- 2) 采用 DIRPATH_DB 或 DIRPATH_FILE_DB 导出时，源库为某些特殊字符集如 WE8MSWIN1252 时，需要指定 NLS_LANG，否则导出时可能会出现 metadata 解析错误。
- 3) 另外如果表是 11g 及以上版本，并且 fast add 添加的列，会转换成 BULK 方式，此时必须设置 NLS_LANG。
- 4) 源和目标端字符集不同时，建议 NLS_LANG 设置成目标端的字符集，再执行 osync.sh。

例如：源 ZHT16MSWIN950 => 目标端 ZHS16GBK 时，设置 NLS_LANG 为 ZHS16GBK。

建议先建立两张测试表，确认导入中文无乱码。

```
create table t_varchar(id number, name varchar2(200));
insert into t_varchar values(1, 'gyh');
insert into t_varchar values(2, '耿永辉');
commit;
```

5. 启动 osync，并初始化复制表

```
./osync.sh
```

```
OSYNC> init sync
```

```
DB information:
```

```
-----
|           |Machine   |DB       |Instance |Version   |
-----
|From(Source)|ANDY-PC   |TEST32  |test32   |11.2.0.1.0|
-----
```

```
-----  
Warning, all your old replication status will be reset.  
Are you sure to init/reinit sync? (N) y  
Starting sync init...  
Sync init finish successfully (2 tables, elapsed 0.07 secs).
```

6. 启动复制

OSYNC> start sync

DB information:

```
-----  
|           |Machine  |DB      |Instance |Version  |  
-----  
|From(Source)|ANDY-PC  |TEST32  |test32   |11.2.0.1.0|  
-----  
| To(Target) |shareplex|DB12101 |db12101 |12.1.0.1.0|  
-----
```

```
Are you sure to start sync? (N) y  
Starting sync...
```

7. 显示复制状态

OSYNC> status

Sync Table Statistics (Total: 2, Init: 1, Exported: 0, Imported: 0, Error: 1)

Export Threads (Table exported 1, exporting 1, error 0, remain 0)

Thread# 0: (Table exported 0, error 0, elapsed 0.00 secs, db read/file write: 0.00/0.00 secs)

Current table:[], status:[Unknown] (exported: 0.00 MB 0 rows 0.00 secs)

Thread# 1: (Table exported 0, error 0, elapsed 0.00 secs, db read/file write: 0.00/0.00 secs)

Current table:[ANDY.T_TEST2], status:[Exporting] (exported: 48.00 MB 543586 rows 1.11 secs)

Thread# 2: (Table exported 1, error 0, elapsed 0.04 secs, db read/file write: 0.03/0.00 secs)

Current table:[ANDY.T_TEST], status:[Finish] (exported: 0.00 MB 1 rows 0.04 secs)

Import Threads (Table imported 0, importing 0, error 1, remain 0)

Thread# 0: (Table imported 0, error 1, elapsed 0.00 secs, file read/db write: 0.00/0.00 secs)

Current Table:[ANDY.T_TEST], status:[Error] (imported: 0.00 MB 0 rows 0.00 secs)

Thread# 1: (Table imported 0, error 0, elapsed 0.00 secs, file read/db write: 0.00/0.00 secs)

Current Table:[], status:[Unknown] (imported: 0.00 MB 0 rows 0.00 secs)

Thread# 2: (Table imported 0, error 0, elapsed 0.00 secs, file read/db write: 0.00/0.00 secs)

说明:

1. Sync Table Statistics (Total: 2, Init: 0, Exported: 0, Imported: 0, Error: 2)

Total	总共复制的表数
Init	尚未导出的表数
Exported	导出完成的表数
Imported	导入完成的表数
Error	导出/导入报错的表数

Total = Init + Exported + Imported + Error。

2. Export Threads (Table exported 1, exporting 1, error 0, remain 0)

Exported	导出线程当前已经导出的表数
exporting	正在导出的表数
Error	导出时发生错误的表数
Remain	导出队列中剩余表数

Exporting 和 remain 均为 0 时，表明导出线程已经结束。

3. Import Threads (Table imported 0, importing 0, error 1, remain 0)

imported	导入线程当前已经导和的表数
importing	正在导入的表数
Error	导入时发生错误的表数
Remain	导入队列中剩余表数

Importing 和 remain 为 0，并且导出线程已经结束时，表明导入线程已经结束。

8. 验证导入正确

导入结束后，可以 `select count(*) from 表名`；验证两边导入行数和中文字符，确认导入成功。

9. 导入成功后，目标端请重建或新建索引。

说明：为加快索引创建速度，可以使用 `nologging` 和 `parallel` 选项。

示例：

```
select owner, index_name, status, logging, degree
from dba_indexes
where table_name = 'T_VARCHAR';
```

并行方式创建索引：

```
create index andy.idx_t_varchar_id on andy.t_varchar(id) nologging
parallel 5;
```

并行方式重建索引：

```
alter index andy.idx_t_varchar_id rebuild nologging parallel 5;
```

创建/重建结束后，索引置为 logging 和 noparallel 值。

```
alter index andy.idx_t_varchar_id logging noparallel;
```

以下脚本重建用户下所有索引：

```
set serverout on size 1000000
spool rebuild_index.log

declare
v_sql varchar(1024);
parallel number := 5;
begin

for tbl in(select owner, table_name
           from dba_tables
           where owner = 'ANDY2'
           and temporary = 'N'
           and secondary = 'N'
           and nested = 'NO')
loop
dbms_output.put_line('rebuild indexes for table ' ||tbl.owner|| '.' ||tbl.table_name);

for ind in (select owner, index_name, status, partitioned
            from dba_indexes
            where table_owner = tbl.owner
            and table_name = tbl.table_name
            and index_type not in('LOB', 'IOT - TOP'))
loop
--set nologging
v_sql := 'alter index ' || ind.owner || '.' || ind.index_name || ' nologging
parallel 5';
dbms_output.put_line('..' || v_sql);
begin
execute immediate v_sql;
exception
when others then dbms_output.put_line('error: ' || sqlerrm);
end;

if ind.partitioned = 'NO' then --rebuild normal/global index
--rebuild index
v_sql := 'alter index ' || ind.owner || '.' || ind.index_name || ' rebuild
nologging parallel 5';
dbms_output.put_line('..' || v_sql);
```

```

begin
    execute immediate v_sql;
exception
    when others then dbms_output.put_line('error: ' || sqlerrm);
end;

else --rebuild partition index

    for part in (select index_owner, index_name, partition_name, subpartition_count,
tablespace_name
                from dba_ind_partitions
                where index_owner = ind.owner
                  and index_name = ind.index_name)
    loop

        if part.tablespace_name is NOT NULL then
            --rebuild partition
            v_sql := 'alter index ' || part.index_owner || '.' || part.index_name || '
rebuild partition ' || part.partition_name || ' nologging parallel 5';
            dbms_output.put_line('..' || v_sql);
            begin
                execute immediate v_sql;
            exception
                when others then dbms_output.put_line('error: ' || sqlerrm);
            end;
        else
            --rebuild sub-partition
            for subpart in (select index_owner, index_name, partition_name,
subpartition_name
                           from dba_ind_subpartitions
                           where index_owner = part.index_owner
                             and index_name = part.index_name
                             and partition_name = part.partition_name)
            loop
                v_sql := 'alter index ' || subpart.index_owner || '.' ||
subpart.index_name || ' rebuild subpartition ' || subpart.subpartition_name || ' parallel 5';
                dbms_output.put_line('..' || v_sql);
                begin
                    execute immediate v_sql;
                exception
                    when others then dbms_output.put_line('error: ' || sqlerrm);
                end;
            end loop;
        end if;
    end loop;
end if;

```

```

        end loop;
    end if;

    --set logging
    v_sql := 'alter index ' || ind.owner || '.' || ind.index_name || ' logging
noparallel';
    dbms_output.put_line('..' || v_sql);
    begin
        execute immediate v_sql;
    exception
        when others then dbms_output.put_line('error: ' || sqlerrm);
    end;
end loop;

end loop;
end;
/

spool off

```

最佳实践(后台服务方式)

Osync 支持以后台守护进程的方式运行，并通过 `osynctl.sh` 命令交互。
具体方法：

1. 配置 `conf/osync.conf`
2. 配置 `conf/table.list` 文件
3. 运行 `./osynctl.sh`
4. 启动 `osync` 守护进程，默认端口号为 10000。
`OSYNC(10000)> start service`
 说明：
 - 1) `ps -ef|grep osync_` 可以查看守护进程。
 - 2) 如果启动后更改 `osync.conf` 文件内容，需要重启守护进程生效。
5. 初始化配置
`OSYNC(10000)> init sync`
6. 启动复制
`OSYNC(10000)> start sync`
 说明：启动复制后，可以退出该命令程序，等待守护进程完成复制操作。

7. 中断复制
OSYNC(10000)> stop sync
8. 查看复制状态
OSYNC(10000)> status
9. 复制结束后，可以关闭 osync 守护进程
OSYNC(10000)> stop service

单表并行导出

OSYNC 命令行工具支持单表并行导出，步骤如下：

1. 设置 osync.conf 文件中的 parallel 为适当的并行度，比如 3。
2. table.list 文件中设置 1 张需要并行导出的表。
3. OSYNC>init sync

```
OSYNC>start sync
```

```
OSYNC>status
```

查看日志，确认已经正常导出完成。

单表并行导出限制：

*目标库不支持 IOT 表的并行加载。

*目标库如果<11G 版本，LOB 表不支持并行加载。

*DIRPATH_DB 方式只支持分区表并行导出/导入。如果想要非分区表并行导入，请选择 BULK_DB 方式。

其它使用场景

1. 先完全导出，再完全导入，适合>=10g 版本

1) 启动 osync，并初始化复制表

```
$/osync
```

```
OSYNC> init sync
```

2) 启动导出线程，完全导出表到本地

```
OSYNC> start export
```

```
OSYNC> status
```

3) export 结束后，统一导入

```
OSYNC> start import
```

OSYNC> status

配置 Instant client

如果服务器端未安装 ORACLE 软件，可以配置 instant client。

1. 从 oracle 官网下载 instant client 文件，以 instantclient-basic-linux-12.2.0.1.0.zip 为例。
2. 解压文件，并配置符号链接。

```
cd /u02/  
unzip instantclient-basic-linux-12.2.0.1.0.zip  
cd /u02/ instantclient_12_2  
ln -s libclntsh.so.12.1 libclntsh.so
```
3. 更改 osync.sh 或 osynctl.sh 文件
设置 ORACLE_HOME
ORACLE_HOME=/u02/ instantclient_12_2
4. 配置 sqlnet.ora 或 tnsnames.ora 文件(可选)
mkdir -p /u02/instantclient_12_2/network/admin
创建 tnsnames.ora, sqlnet.ora 文件
5. 启动 osync.sh 或 osynctl.sh 文件