



# OCP 认证培训教材

## ORACLE 备份恢复篇



# Oracle 数据库导入与导出

导出（EXPORT）是用 EXP 将数据库部分或全部对象的结构和数据导出。

导入（IMPORT）是用 IMP 将 OS 文件中的对象结构和数据装载到数据库中的过程。

EXP 和 IMP 用于实现逻辑备份和逻辑恢复，导入和导出的作用如下：

- 1、使用导出和导入可以重新组织表。
- 2、使用导出和导入可以在用户之前移动对象。
- 3、使用导出和导入可以在数据库之间移动对象。
- 4、使用导出和导入可以升级数据库到其它平台。
- 5、使用导出和导入可以升级数据库到更高版本。
- 6、使用导出和导入可以实现逻辑备份和恢复。

## EXP 的参数：

Keyword	Description (Default)	Keyword	Description (Default)
USERID	用户名/口令	FULL	导出整个文件 (N)
BUFFER	数据缓冲区大小	OWNER	所有用户名列表
FILE	输出文件 (EXPDAT.DMP)	TABLES	表名列表
COMPRESS	导入到一个区 (Y)	RECORDLENGTH	IO 记录的长度
GRANTS	导出权限 (Y)	INCTYPE	增量导出类型
INDEXES	导出索引 (Y)	RECORD	跟踪增量导出 (Y)
DIRECT	直接路径 (N)	TRIGGERS	导出触发器 (Y)
LOG	屏幕输出的日志文件	STATISTICS	分析对象 (ESTIMATE)
ROWS	导出数据行 (Y)	PARFILE	参数文件
CONSISTENT	交叉表的一致性 (N)	CONSTRAINTS	导出约束条件 (Y)



OBJECT_CONSISTENT	只在对象导出期间设置为只读的事务处理 (N)
FEEDBACK	每 X 行显示进度 (0)
FILESIZE	每个转储文件的最大大小
FLASHBACK_SCN	用于将会话快照设置回以前状态的 SCN
FLASHBACK_TIME	用于获取最接近指定时间的 SCN 时间
QUERY	用于导出表的子集的 select 子句
RESUMABLE	遇到与空格相关的错误时挂起 (N)
RESUMABLE_NAME	用于标识可恢复语句的文本字符串
RESUMABLE_TIMEOUT	RESUMABLE 的等待时间
TTS_FULL_CHECK	对 TTS 执行完整或部分相关性检查
VOLSIZE	number of bytes to write to each tape volume
TABLESPACES	要导出的表空间列表
TRANSPORT_TABLESPACE	导出可传输的表空间元数据 (N)
TEMPLATE	用于 iAS 模式导出的模板名

## EXP 部分参数解释:

**USERID**: 登录数据库的用户名/口令和连接字符串, 当以特权用户身份导出时, 必须用单引号引住 (userid='sys\as\sysdba') 如: USERID=duanl/duanl, 如果 USERID 是 EXP 后的第一个选项, 可以忽略 USERID, 如: user/password

**BUFFER**: 数据缓冲区的大小, 根据导出的实际设置 (如 LOB 数据类型适当设置较大的 buffer), 如: buffer=81920

**FILE**: 指定导出文件名 (EXPDAT.DMP), 如: file=a.dmp

**COMPRESS**: 指定导出导入管理初始区的方法 (Y), 使得当引入并且重新创建对象时, 对象初始化的大小为已经导出的对象大小, 如: compress=n

**GRANTS**: 指定是否导出对象权限信息 (Y), 如: grants=n

**INDEXES**: 指定是否导出索引 (Y), 如: indexes=n



LOG: 指定导出日志文件的名称, 该日志记录整个备份过程, 默认不生成导出日志文件, 如: log=a.log

FULL: 导出整个数据库 (N), 如: full=y, 如果执行整个数据库的导出, 则连接用户必须有 DBA 权限或具有 exp\_full\_database 权限

OWNER: 用于指定用户导出模式, 即导出特定用户的数据库对象, 可以有多个用户名, 使用逗号隔开, 如: owner=username

TABLES: 指定导出的 table 名称, 如果该表属于当前连接用户的, 则直接输入表名, 否则输入模式名. 表名, 当使用多个表时, 用逗号隔开, 如: TABLES=table1, table2 或 tables=username.table

TRIGGERS: 指定是否导出触发器 (Y), 如: triggers=n

STATISTICS: 指定导入导出文件时生成优化统计信息的类型 (ESTIMATE), 设为 estimate 时, 使用 analyze table estimate 语句生成统计信息, 设为 compute 时, 使用 analyze table compute 语句生成统计信息, 设为 none 时, 不会生成统计信息。如: statistice=none

CONSTRAINTS: 指定是否导出约束 (Y), 如: constraints=n

FEEDBACK: 指定导出行数(默认 0)显示进程框, 该参数的值从 0 到任何有效的数字。每导出 n 行, 显示一个圆点 (.), 如: feedback=10

FILESIZE: 指定导出文件的最大尺寸, 可以将数据库对象分布到多个导出文件中, 如: filesize=10240

下列关键字仅用于可传输的表空间

TABLESPACES: 指定表空间导出模式。会导出特定表空间上的所有表, 指定表空间的名称, 该参数可以有多个表空间名用逗号隔开, 如: tablespaces=user02

TRANSPORT\_TABLESPACE : 指定是否导出表空间元数据 (N) 如: TRANSPORT\_TABLESPACE=y



**RECORDLENGTH**: 指定文件记录的长度, 默认值为 BUFFER 选项值, 当需要将导出文件传送到不同 OS 平台时, 可能用到, 该选项的值不能超过 64K。如:

`recordlength=4096`

**ROWS**: 指定是否导出数据行, 选项为 N 时, 只导出表结构, 不导出数据 (Y)

如: `rows=n`

**PARFILE**: 指定导出工具要使用的参数文件名, 命令行选项放到参数文件中。

如: `parfile=parm.txt`

**CONSISTENT**: 指定是否使用 `set transaction read only` 语句确保取得一致时间点数据 (N), 设为 y 时, 所有被导出表会在同一事务内完成导出, 从而确保取得一致时间点的数据。 如: `consistent=y`

**DIRECT**: 指定是否使用直接导出方式, 直接导出速度优于常规导出, 但要求客户端和服务器的字符集必须完本一致。(N) 如: `direct=y`

**QUERY**: 指定 WHERE 条件子句, 从而导出表的部分数据。使用直接导出方式时不能指定该选项。如: `query=' where deptno=10'`

## 一、获取帮助

```
$ exp help=y
```

```
$ imp help=y
```

## 二、三种工作方式

### 1. 交互式方式

```
[oracle@oracle Desktop]$ exp -----按提示输入所需要的参数
```

```
Export: Release 11.2.0.4.0 - Production on Wed Jan 15 06:35:09 2014
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
Username: scott
```

```
Password:
```



Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0  
- 64bit Production

With the Partitioning, OLAP, Data Mining and Real Application Testing  
options

Enter array fetch buffer size: 4096 > ---- 输入数组提取缓冲区大小

Export file: expdat.dmp > abc.dmp ---- 导出文件

(2)U(sers), or (3)T(ables): (2)U > ---- 选择导出方式，如果输入  
的是有导出数据库角色的用户名，会有数据库方式导出，普通用户只能导出当  
前用户的所有数据库对象或导出当前用户的表。默认导出用户的数据库对象。

Export grants (yes/no): yes > ---- 导出权限

Export table data (yes/no): yes > ---- 导出表的数据

Compress extents (yes/no): yes > ---- 指定导出管理初始区的方法，  
会将 initial 设置为表段的当前尺寸

Export done in US7ASCII character set and AL16UTF16 NCHAR character set  
server uses ZHS16GBK character set (possible charset conversion)

- . exporting pre-schema procedural objects and actions
- . exporting foreign function library names for user SCOTT
- . exporting PUBLIC type synonyms
- . exporting private type synonyms
- . exporting object type definitions for user SCOTT

About to export SCOTT's objects ...

- . exporting database links
- . exporting sequence numbers
- . exporting cluster definitions
- . about to export SCOTT's tables via Conventional Path ...



---

. . exporting table	BONUS	0 rows
---------------------	-------	--------

exported

EXP-00091: Exporting questionable statistics.

. . exporting table	DEPT	4 rows
---------------------	------	--------

exported

EXP-00091: Exporting questionable statistics.

EXP-00091: Exporting questionable statistics.

. . exporting table	EMP	14 rows
---------------------	-----	---------

exported

EXP-00091: Exporting questionable statistics.

EXP-00091: Exporting questionable statistics.

. . exporting table	SALGRADE	5 rows
---------------------	----------	--------

exported

EXP-00091: Exporting questionable statistics.

. exporting synonyms

. exporting views

. exporting stored procedures

. exporting operators

. exporting referential integrity constraints

. exporting triggers

. exporting indextypes

. exporting bitmap, functional and extensible indexes

. exporting posttables actions

. exporting materialized views

. exporting snapshot logs

. exporting job queues



- . exporting refresh groups and children
- . exporting dimensions
- . exporting post-schema procedural objects and actions
- . exporting statistics

Export terminated successfully with warnings.

当使用交互方式导出表时，每次只能导出当前用户的一个表，如果要导出多个表，需要多个操作。

## 实例：

- 1) 导出整个数据库
- 2) 导出数据库中某个用户的所有数据库对象
- 3) 导出特定的表

## 2. 命令行方式

在命令行中输入所需的参数即可，如下所示：

```
exp user/pwd@dbname file=/oracle/test.dmp full=y
```

## 实例：

### 1) 导出整个数据库

导出整个数据库，只需要 FULL 和 FILE 二个参数，FULL 表示导出整个文件，包括所有的数据库对象，FILE 表示备份后的数据文件名。

```
exp system/oracle full=y file=full.dmp
```

只导出数据库的结构，不要表的数据，即导出不包含数据的全库备份。

```
exp system/oracle full=y rows=n file=fulla.dmp
```

### 2) 导出数据库中某个用户的所有数据库对象

导出当前数据库中某个用户的所有数据库对象，作为该用户的数据库备份，使用 owner 参数指定。导出该用户所拥有的数据库对象的定义，以及对象所拥有的数据。使用 system 用户登录数据库，该用户具有 DBA 权限，所以它拥有访问 scott 用户的数据库对象权限，可以导出 scott 用户的所有





数据库对象,也可以使用 scott 用户登录数据库,导出它自己拥有的数据库对象。

```
exp system/oracle owner=scott file=userback_scott.dmp
```

或

```
exp scott/scott owner=scott file=back_user_scott.dmp
```

或

```
exp scott/scott file=back_user_scott.dmp
```

### 3) 导出特定的表

导出特定用户的特定表需要 tables 参数,该参数后可以有几个表名,如导出的表不是当前用户的表,需要使用 schema\_name.table\_name 的形式,告诉 exp 这个表属于哪个用户。System 具有对用户 scott 中表的访问权,所以该用户可以导出 scott 用户的表,如果使用 scott 用户登录数据库,在导出自己的表时可以直接写出表名,不用加模式名。

导出一张表

```
exp      userid=hr/hr      tables=jobs      file=hr_tables.dmp
log=/home/oracle/jobs.log
```

导出多张表

```
exp userid=system/oracle tables=scott.dept, tables=scott.emp
file=tables.dmp
```

或

```
exp      userid=system/oracle      tables=scott.dept, scott.emp
file=tables.dmp
```

或

```
exp scott/scott tables=dept,emp file=backup_scott_tables.dmp
```

导出表的部分数据,使用 query 参数

```
exp      userid=hr/hr      file=hr_tables_9000.dmp      tables=jobs
query=\ ' where MAX_SALARY\ =9000\ '
```



**注意：对字符采用\转义。**

导出表结构

```
exp userid=hr/hr tables=jobs file=hr_jobs_str.dmp rows=n
```

不导出与对象相关的权限信息，不导出表的索引，不导出表的约束，不导出触发器

```
exp  userid=hr/hr  tables=jobs  file=hr_job.dmp  triggers=n  
indexes=n grants=n constraints=n
```

#### 4) 导出特定的表空间

exp 对特定的表空间实施导出操作，导出该表空间中所有表的定义和表中的数据。如果不需要导出表中的数据，只需要备份表空间中对象的定义，可以使用 ROWS 参数，设置 ROWS=no 即可。

```
exp                      system/oracle                      tablespaces=users  
file=backup_tablespaces_users.dmp
```

或

```
exp                      system/oracle                      tablespaces=users  
file=backup_tablespaces_users.dmp rows=no
```

或

```
exp  userid='/' as sysdba'  tablespaces=TBS_SINGLE  
file=/tmp/ODU.dmp transport_tableSPACE=y
```

### 3. 参数文件方式，在参数文件中输入所需的参数

```
$ exp parfile=username.par
```

参数文件 username.par 内容

```
userid=system/oracle
```

```
buffer=8192000
```

```
compress=n
```



```
grants=y
```

```
file=/home/oracle/test.dmp
```

```
full=y
```

### 三、三种模式

- 1、表方式，将指定表的数据导出/导入
- 2、用户方式，将指定用户的所有对象及数据导出/导入
- 3、全库方式，将数据库中的所有对象导出/导入

## IMP

### IMP 的参数：

Keyword	Description (Default)	Keyword	Description (Default)
USERID	用户名/口令	FULL	导入整个文件 (N)
BUFFER	数据缓冲区大小	FROMUSER	所有者用户列表
FILE	输入文件 (EXPDAT.DMP)	TOUSER	用户名列表
SHOW	只列出文件内容 (N)	TABLES	表名列表
IGNORE	忽略创建错误 (N)	RECORDLENGTH	IO 记录的长度
GRANTS	导入权限 (Y)	INCTYPE	增量导入类型
INDEXES	导入索引 (Y)	COMMIT	提交数据插入 (N)
ROWS	导入数据行 (Y)	PARFILE	参数文件名
LOG	屏幕输出的日志文件	CONSTRAINTS	导入限制 (Y)
DESTROY	覆盖表空间数据文件 (N)		
INDEXFILE	将表/索引信息写入指定的文件		
SKIP_UNUSABLE_INDEXES	跳过不可用索引的维护 (N)		
FEEDBACK	每 X 行显示进度 (0)		
TOID_NOVALIDATE	跳过指定类型 ID 的验证		



FILESIZE	每个转储文件的最大大小
STATISTICS	始终导入预计算的统计信息 (always)
RESUMABLE	在遇到有关空间的错误时挂起 (N)
RESUMABLE_NAME	用来标识可恢复语句的文本字符串
RESUMABLE_TIMEOUT	RESUMABLE 的等待时间
COMPILE	编译过程，程序包和函数 (Y)
STREAMS_CONFIGURATION	导入流的一般元数据 (Y)
STREAMS_INSTANTIATION	导入流实例化元数据 (N)
DATA_ONLY	只导入数据 (N)
VOLSIZE	number of bytes in file on each volume of a file on tape

下面关键字用于可传输的表空间

TRANSPORT_TABLESPACE	导入可传输的表空间元数据 (N)
TABLESPACES	将要传输到数据库的表空间
DATAFILES	将要传输到数据库的数据文件
TTS_OWNERS	拥有可传输表空间集中数据的用户

## IMP 部分参数解释：

**USERID**: 登录数据库的用户名/口令和连接字符串。

**BUFFER**: 数据缓冲区的大小

**FILE**: 指定创建的备份文件名

**show**: 说明是否使得导入过程只显示备份文件的内容

**ignore**: 说明是否导入备份文件中的授权

**INDEXES**: 指定是否导入索引

**ROWS**: 指定是否导入数据行

**LOG**: 指定导入日志文件的名称，该日志记录整个备份过程

**FULL**: 导入整个数据库



fromuser:允许将一个备份文件中的对象从一个用户复制到另一个用户, 该参数说明导入数据的源用户名。

touser:允许将一个备份文件中的对象从一个用户复制到另一个用户, 该参数说明导入数据的目的用户名。

**TABLES:** 指定导入的 table 名称, 当使用多个表时, 用逗号隔开

**CONSTRAINTS:** 指定是否导入约束 (Y)

## 实例:

### 1、导入整个数据库

imp 将备份的整个数据库导入当前的数据库中, 在导入过程中包括一系列地创建数据库对象的过程, 如创建表空间、表、表的索引, 序列号以及用户授权等。如果要创建的数据库对象, 但某个表已经存在, 则该创建语句会失败。此时需要使用 ignore 参数, 该参数使得 imp 程序忽略这些错误。使用这种方法恢复整个数据库其实就是重新创建数据库中所有对象的过程, 所以是一种逻辑数据库恢复。

```
exp system/oracle full=y file=full.dmp
```

```
SQL> drop user scott cascade;
```

```
SQL> drop user hr cascade;
```

```
imp system/oracle full=y file=full.dmp ignore=y
```

### 2、导入指定用户的全部数据库对象, 不建用户

```
exp scott/scott file=back_user_scott.dmp
```

```
exp system/oracle owner=scott file=back_user_scott1.dmp
```

```
SQL> drop user scott cascade;
```

```
SQL> create user scott identified by scott DEFAULT TABLESPACE users  
TEMPORARY TABLESPACE temp QUOTA 10M ON users;
```

```
SQL> grant create session to scott;
```

```
imp scott/scott file= back_user_scott.dmp ignore=y
```

或



```
imp system/oracle full=y file=back_user_scott1.dmp ignore=y
```

或

```
imp system/oracle file=back_user_scott1.dmp ignore=y fromuser=scott  
touser=scott
```

### 3、导入特定的表

将特定的表导入指定的用户

```
imp system/oracle tables=emp file=back_user_scott1.dmp fromuser=scott  
touser=system
```

向用户 system 导入了 scott 用户的表 emp，当导入 emp 表时其实就是在新用户 system 的用户表空间中创建一个表并填充数据，最后创建基于该表的触发器。

导入一张表

```
exp          userid=hr/hr          tables=jobs          file=hr_tables.dmp  
log=/home/oracle/jobs.log  
drop table jobs cascade constraints;  
imp          userid=hr/hr          tables=jobs          file=hr_tables.dmp  
log=/home/oracle/jobs.log
```

导入多张表

```
exp userid=system/oracle tables=scott.dept,scott.emp file=tables.dmp  
SQL> drop table emp purge;  
SQL> drop table dept purge;  
imp userid=system/oracle tables=dept,emp file=tables.dmp fromuser=scott  
touser=scott
```

导入表的部分数据

```
exp  userid=scott/scott  file=scott_tables_7902.dmp  tables=emp  
query=\ 'where empno\=7902\  
delete from emp where empno=7902;
```



```
imp userid=scott/scott file=scott_tables_7902.dmp tables=emp ignore=y
```

## 四、传输表空间

表空间传输是实现数据迁移的一种方法，使用传输表空间实现在二个数据库之间复制表空间数据集（表空间逻辑结构和数据）。在下列场合多使用传输表空间：在多个数据库之间复制同一个表空间的多个只读版本，归档历史数据。

### 理解 big/little endian

在传输表空间时需要注意“字节存储序列”，因为在不同的平台上是不同的，字节序列决定了数据值在操作系统平台上的存储顺序，有 big endian 和 little endian 之分。

big endian 是大值存储位靠前，而 little endian 是大值存储位靠后。如果 1234 在支持 big endian 的操作系统中存储顺序是 3412,而在支持 little endian 的操作系统中存储顺序是 1234。

在不同平台传输表空间之前，要清楚二个平台的字节序列是否相同，如果不同还需要字节序列转换来转换数据文件的存储序列。

该转换可以在源数据库端，也可以在目标数据库端实现，我们在源数据库端进行转换，首先打开 RMAN，例源端 windows 目标 solaris。

```
RMAN>convert tablespace test2 to platform 'solaris[tm] OE (32-bit)'  
format 'd:/expdp_trans.dmp' ;
```

表空间 test2 的所有数据文件完成了格式转换,使得该数据文件支持 solaris[tm] OE (32-bit) 平台的字节序列存储格式，并将转换后的数据文件保存为 d:/expdp\_trans.dmp 文件。

```
imp sys/oracle file=expdp.dmp datafile=expdp_trans.dmp  
transport_tablespace=y
```

### 传输表空间的限制



并不能随意使用传输表空间，使用传输表空间需要二个数据库平台的一些特性要求，这些限制有：

- 1、源和目标数据库必须使用相同的字符集和国家字符集
- 2、如果目标数据库有一个与传输表空间相同名称的表空间存在，则不能传输该表空间到目标数据库。
- 3、如果一些对象具有更底层的对象，如带有包含对象（分区表），这样的对象不能被传输

### 传输表空间的自包含特性

在一个传输表空间中在不同的数据库对象之间可能会有逻辑或物理的依赖关系，传输表空间中对象的某些依赖对象并不存在于传输表空间中，这样的表空间不能传输，ORACLE 只允许传输自包含的表空间，自包含的意思是指传输表空间中没有任何引用存在于传输表空间之外，以下情况违反了表空间的自包含特性。

- 1、传输表空间中存储了另一个表空间中表的索引，这个索引对象不违反自包含特性，但是，如果传输表空间中存在一个表，而该表的索引存在于另一个表空间中，则该传输表空间就是自包含的。
- 2、分区表部分在传输表空间中，这种情况也是违反表空间的自包含，必须将分区表的所有分区都存储在传输表空间中，或者分区表转换成实体表。
- 3、传输表空间中的表包含 LOG 列，该列指向传输表空间外的 LOG 对象。

ORACLE 允许使用 DBMS\_TTS 包的 TRANSPORT\_SET\_CHECK 过程，验证表空间是否具有自包含特性，执行该过程需要用户具有 EXECUTE\_CATALOG\_ROLE 角色。

### 实现传输表空间的步骤

- 1、对于跨平台的表空间传输需要使用 V\$TRANSPORTABLE\_PLATFORM 查询每个平台支持的 endianness。

```
SELECT * FROM V$TRANSPORTABLE_PLATFORM;
```

说明当前的数据库版本支持的不同操作系统平台，也就是告诉我们可以在这操作系统平台之间进行表空间的迁移。





## 2、 查询当前的传输表空间是否是自包含的

```
EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('TBS_SINGLE', true);
```

```
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

此时发现没有任何违反表空间自包含特性的输出。

## 3、 使用 exp 或 expdp 等工具导出表空间元数据。

元数据是表空间的定义和表的结构，但不包含实际的表数据。使用 exp 工具将表空间的元数据集合导出，些时需要先将表空间 read only

```
alter tablespace tbs_single read only;
```

## 4、 备份该表空间的数据文件，

## 5、 将传输表空间的导出元数据集和数据文件复制到目的数据库

使用 exp 或 imp 工具将传输表空间的元数据导入目的数据库。