

# 达梦技术手册

## DM8\_dexp 和 dimp 使用手册

Service manual of DM8\_dexp&dimp



# 前言

## 概述

本文档主要介绍如何使用 DM 的命令行工具 dexp（逻辑导出）和 dimp（逻辑导入），它们所提供的功能、以及详细的参数介绍。

## 读者对象





本文档主要适用于 DM 数据库的：

- 开发工程师
- 测试工程师
- 技术支持工程师
- 数据库管理员

## 通用约定

在本文档中可能出现下列标志，它们所代表的含义如下：

表 0.1 标志含义

标志	说明
 警告：	表示可能导致系统损坏、数据丢失或不可预知的结果。
 注意：	表示可能导致性能降低、服务不可用。
 小窍门：	可以帮助您解决某个问题或节省您的时间。
 说明：	表示正文的附加信息，是对正文的强调和补充。

在本文档中可能出现下列格式，它们所代表的含义如下：

表 0.2 格式含义

格式	说明
宋体	表示正文。
Courier new	表示代码或者屏幕显示内容。
粗体	表示命令行中的关键字（命令中保持不变、必须照输的部分）或者正文中强调的内容。标题、警告、注意、小窍门、说明等内容均采用粗体。
<>	语法符号中，表示一个语法对象。
::=	语法符号中，表示定义符，用来定义一个语法对象。定义符左边为语法对象，右边为相应的语法描述。
	语法符号中，表示或者符，限定的语法选项在实际语句中只能出现一个。
{ }	语法符号中，大括号内的语法选项在实际的语句中可以出现 0...N 次 (N 为大于 0 的自然数)，但是大括号本身不能出现在语句中。
[ ]	语法符号中，中括号内的语法选项在实际的语句中可以出现 0...1 次，但是中括号本身不能出现在语句中。
关键字	关键字在 DM_SQL 语言中具有特殊意义，在 SQL 语法描述中，关键字以大写形式出现。但在实际书写 SQL 语句时，关键字既可以大写也可以小写。

## 访问相关文档

如果您安装了 DM 数据库，可在安装目录的“\doc”子目录中找到 DM 数据库的各种手册与技术丛书。

您也可以通过访问我们的网站 [www.dameng.com](http://www.dameng.com) 阅读或下载 DM 的各种相关文档。

## 联系我们

如果您有任何疑问或是想了解达梦数据库的最新动态消息，请联系我们：

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

技术服务电话：400-991-6599

技术服务邮箱：[dmtech@dameng.com](mailto:dmtech@dameng.com)

# 目录

<b>1 功能简介 .....</b>	<b>1</b>
<b>2 dexp 逻辑导出 .....</b>	<b>2</b>
2.1 使用 dexp 工具.....	2
2.2 dexp 参数一览表 .....	3
2.3 dexp 参数详解 .....	5
2.3.1 特殊参数.....	5
2.3.2 普通参数.....	12
<b>3 dimp 逻辑导入 .....</b>	<b>22</b>
3.1 使用 dimp 工具.....	22
3.2 dimp 参数一览表 .....	22
3.3 dimp 参数详解 .....	25
3.3.1 特殊参数.....	25
3.3.2 普通参数.....	29
<b>4 dexp 和 dimp 应用实例.....</b>	<b>40</b>
4.1 一个完整示例 .....	40
4.2 使用关联参数示例 .....	40

# 1 功能简介

DM 数据库的备份还原包括两种类型：物理备份还原和逻辑备份还原。物理备份还原是对数据库的操作系统物理文件（如数据文件、控制文件和日志文件等）的备份还原。例如使用 RMAN 工具进行的备份还原。物理备份还原请参考《DM8 备份与还原》。逻辑备份还原是对数据库逻辑组件（如表、视图和存储过程等数据库对象）的备份还原。例如使用 dexp 和 dimp 进行的备份还原。逻辑备份还原正是本书所要介绍的内容。

逻辑导出（dexp）和逻辑导入（dimp）是 DM 数据库的两个命令行工具，分别用来实现对 DM 数据库的逻辑备份和逻辑还原。逻辑备份和逻辑还原都是在联机方式下完成，联机方式是指数据库服务器正常运行过程中进行的备份和还原。dexp 和 dimp 是 DM 数据库自带的工具，只要安装了 DM 数据库，就可以在安装目录/dmdbms/bin 中找到。

逻辑导出和逻辑导入数据库对象分为四种级别：数据库级、用户级、模式级和表级。四种级别独立互斥，不能同时存在。四种级别所提供的功能：

- 数据库级（FULL）：导出或导入整个数据库中的所有对象。
- 用户级（OWNER）：导出或导入一个或多个用户所拥有的所有对象。
- 模式级（SCHEMAS）：导出或导入一个或多个模式下的所有对象。
- 表级（TABLES）：导出或导入一个或多个指定的表或表分区。

## 2 dexp 逻辑导出

dexp 工具可以对本地或者远程数据库进行数据库级、用户级、模式级和表级的逻辑备份。备份的内容非常灵活，可以选择是否备份索引、数据行和权限，是否忽略各种约束（外键约束、非空约束、唯一约束等），在备份前还可以选择生成日志文件，记录备份的过程以供查看。

### 2.1 使用 dexp 工具

dexp 工具需要从命令行启动。在 cmd 命令行工具中找到 dexp 所在安装目录 /dmdbms/bin，输入 dexp 和参数后回车。参数在下一节详细介绍。

语法如下：

---

```
dexp PARAMETER=<value> { PARAMETER=<value> }
```

---

PARAMETER: dexp 参数。多个参数之间排列顺序无影响，参数之间使用空格间隔。

<value>: 参数取值。

例如，将用户名和密码均为 SYSDBA，IP 地址为 192.168.0.248，端口号为 8888 的数据库采用 FULL 方式完全导出。/user/data 为数字证书路径。导出文件名为 db\_str.dmp，导出的日志文件名为 db\_str.log，导出文件的路径为 /mnt/dexp/data。

```
./dexp USERID=SYSDBA/SYSDBA@192.168.0.248:8888#/user/data FILE=db_str.dmp  
DIRECTORY=/mnt/dexp/data LOG=db_str.log FULL=Y
```

## 2.2 dexp 参数一览表

本节提供 dexp 的参数一览表，供用户快速参考。

表 2.1 dexp 参数一览表

参数	含义	备注
USERID	用户名/口令@主库名:端口号#证书路径。例如: SYSDBA/SYSDBA *MPP_TYPE@server:5236#ssl_path@ssl_pwd	必选。其中主库名、端口号和证书路径为可选项
FILE	明确指定导出文件名称	可选。如果缺省该参数，则导出文件名为 dexp.dmp
DIRECTORY	导出文件所在目录	可选
FULL	导出整个数据库 (N)	可选，四者中选其一。 缺省为 SCHEMAS
OWNER	用户名列表，导出一个或多个用户所拥有的所有对象	
SCHEMAS	模式列表，导出一个或多个模式下的所有对象	
TABLES	表名列表，导出一个或多个指定的表或表分区	
FUZZY_MATCH	TABLES 选项是否支持模糊匹配 (N)	可选
QUERY	用于指定对导出表的数据进行过滤的条件。	可选
PARALLEL	用于指定导出的过程中所使用的线程数目	可选
TABLE_PARALLEL	用于指定导出每张表所使用的线程数, 在 MPP 模式下会转换成单线程	可选
TABLE_POOL	用于设置导出过程中存储表的缓冲区个数	可选
EXCLUDE	1. 导出内容中忽略指定的对象。对象有 CONSTRAINTS、INDEXES、ROWS、TRIGGERS 和 GRANTS。 比如: EXCLUDE= (CONSTRAINTS, INDEXES) 2. 忽略指定的表, 使用 TABLES:INFO 格式, 如果使用表级导出方式导出, 则使用 TABLES:INFO 格式的 EXCLUDE 无效。例如: EXCLUDE= TABLES:table1,table2	可选

	3. 忽略指定的模式, 使用 SCHEMAS:INFO 格式, 如果使用表级, 模式级导出方式导出, 则使用 SCHEMAS:INFO 格式的 EXCLUDE 无效。例如: EXCLUDE=SCHEMAS: SCH1, SCH2	
INCLUDE	导出内容中包含指定的对象 例如: INCLUDE=(CONSTRAINTS, INDEXES) 或者 INCLUDE=TABLES:table1, table2	可选
CONSTRAINTS	导出约束 (Y)	可选。 此处单独设置与和 EXCLUDE/INCLUDE 中批量设置功能一样。 设置一个即可
TABLESPACE	导出的对象定义是否包含表空间 (N)	
GRANTS	导出权限 (Y)	
INDEXES	导出索引 (Y)	
TRIGGERS	导出触发器 (Y)	
ROWS	导出数据行 (Y)	
LOG	明确指定日志文件名称	可选, 如果缺省该参数, 则导出文件名为 dexp.log
NOLOGFILE	不使用日志文件 (N)	可选
NOLOG	屏幕上不显示日志信息 (N)	可选
LOG_WRITE	日志信息实时写入文件 (N)	可选
DUMMY	交互信息处理: 打印 (P), 所有交互都按 YES 处理 (Y), NO (N)。默认为 NO, 不打印交互信息。	可选
PARFILE	参数文件名, 如果 dexp 的参数很多, 可以存成参数文件	可选
FEEDBACK	每 x 行显示进度 (0)	可选
COMPRESS	是否压缩导出数据文件 (N)	可选
ENCRYPT	导出数据是否加密 (N)	可选, 和 ENCRYPT 同时使用
ENCRYPT_PASSWORD	导出数据的加密密钥	
ENCRYPT_NAME	导出数据的加密算法	可选。 和 ENCRYPT 、 ENCRYPT_PASSWORD



		同时使用。缺省为 RC4
FILESIZE	用于指定单个导出文件大小的上限。可以按字节 [B]、K[B]、M[B]、G[B] 的方式指定大小	可选
FILENUM	多文件导出时，一个模板可以生成文件数，范围为 [1, 99]，默认 99	可选
DROP	导出后删除原表，但不级联删除 (N)	可选
DESCRIBE	导出数据文件的描述信息，记录在数据文件中	可选
LOCAL	MPP 环境下使用 MPP_LOCAL 方式登录 (N)	可选
HELP	显示帮助信息	可选



每个参数含义后面的括号内为 (N) 则表示该参数缺省为否，为 (Y) 则表示为是，

**注意：**虽然参数大部分为可选参数，但后台这些参数都会传到服务器，此时参数的值就是缺省值。

## 2.3 dexp 参数详解

本节详细介绍 dexp 的各个参数。

### 2.3.1 特殊参数

特殊参数中介绍了必选参数的和最常用的可选参数。

#### 2.3.1.1 USERID

USERID 用于连接服务器。必选参数。

语法如下：

---

```
USER/PWD*MPP_TYPE@SERVER:PORT#SSLPATH@SSLPWD
```

---

USER/PWD：用户名和密码，用户名必写，密码缺省为 SYSDBA。

MPP\_TYPE：MPP 登录属性，此属性的设置对非 MPP 系统没有影响。此属性的有效值为 GLOBAL 和 LOCAL，默认为 GLOBAL。GLOBAL 表示 MPP 环境下建立的会话为全局会话，对数据库的导入导出操作在所有节点进行；LOCAL 表示 MPP 环境下建立的会话为本地会话，

对数据库的导入导出操作只在本地节点进行。

SERVER:PORT: 服务器 IP 地址和端口号。缺省情况下默认为本地服务器和端口号 LOCALHOST:5236。当服务器为本机时, SERVER:PORT 可直接写 LOCALHOST。例如: 192.168.0.248:8888。

SSLPATH@SSLPWD: 通信加密的 SSL 数字证书路径和密码, 缺省为不使用加密。数字证书路径由用户自己创建, 将相应的证书需放入该文件夹中。其中服务器证书必须与 dmserver 目录同级, 客户端目录可以任意设置。

当连接其他服务器时, SERVER:PORT 需写上 IP 地址和 PORTNUM。



**PWD 密码中带有特殊字符(/, @, . :等), 需要外加三层双引号("''''')**

**小窍门:** 进行转义。

PWD 密码中带有特殊字符的具体转义处理规则:

登录时, 密码中含有 /, @, . :等特殊字符时, 密码需要外加三层双引号("''''')

下面使用一个例子进行解析。例如: 密码为 aaa/aaa, 首先, dexp 要求对含有特殊字符的密码内部进行转义 (外加一层双引号), 写成"aaa/aaa"。其次, 要将转义后的内容作为一个整体传给 dexp (外加一层双引号), 写成"aaa/aaa"。最后, 整个内容需要作为一个整体传给操作系统 (外加一层双引号), 因此写成"aaa/aaa"。

例如, 使用上面的规则举例, 密码为 SYSDBA/, @, . : , 端口号为 5236 的本地服务器, USERID 可写成如下所示:

```
./dexp USERID=SYSDBA/"'"SYSDBA/, @, . : '"'"@LOCALHOST:PORT
```

### 2.3.1.2 FILE

FILE 用于明确指定导出的文件 (可以包含路径), 可以包含多个文件, 用逗号分隔。可选参数, 如果不选用 FILE 参数 (来明确指定导出文件名称), 那么默认导出文件名称为 dexp.dmp。

语法如下:

```
FILE=<文件>
```

例如, 设置 FILE=db\_str.dmp, 导出文件名为 db\_str.dmp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
```

```
DIRECTORY=/mnt/data/dexp
```

例如，FILE 指定的文件也可以包含路径，设置 FILE=/mnt/db\_str.dmp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=/mnt/db_str.dmp LOG=db_str.log FULL=Y
```



说明：

如果 **FILE** 指定的文件包含生成路径，则忽略 **DIRECTORY** 中的路径；

如果 **FILE** 没有指定路径时，使用 **DIRECTORY** 中指定的路径；

如果 **FILE** 和 **DIRECTORY** 都没有指定路径，则使用系统当前路径。

文件也可以在生成过程中自动扩展成多个。首先，文件名需要包含通配符%U，用于作为自动扩充文件的文件名模板。%U 表示为 2 个字符宽度的数字，由系统自动生成，起始为 01。其次，使用 FILESIZE 参数来指定文件的大小。

例如，设置 FILE=db\_str%U.dmp，导出文件名为 db\_str%U.dmp。同时使用%U 对文件进行扩展，使用 FILESIZE=128m 指定文件大小。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str%U.dmp LOG=db_str.log FULL=Y
DIRECTORY=/mnt/data/dexp FILESIZE=128m
```

### 2.3.1.3 LOG

LOG 用于明确指定导出的日志文件名称（可以包含路径），可以包含多个文件，用逗号分隔。可选参数。如果不选用 LOG 参数（来明确指定日志文件名称），那么默认导出日志文件名称为 dexp.log。

语法如下：

```
LOG=<文件名>
```

例如，指定日志文件名为 db\_str.log。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
DIRECTORY=/mnt/data/dexp
```

文件也可以在生成过程中自动扩展成多个。首先，文件名需要包含通配符%U，用于作为自动扩充文件的文件名模板。%U 表示为 2 个字符宽度的数字，由系统自动生成，起始为 01。其次，使用 FILESIZE 参数来指定文件的大小。

例如，指定日志文件名为 db\_str%U.log。同时使用%U 对文件进行扩展，使用 FILESIZE=128m 指定文件大小。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str%U.log FULL=Y
```

---

```
DIRECTORY=/mnt/data/dexpFILESIZE=128m
```

---



如果 LOG 指定的文件包含生成路径，则忽略 DIRECTORY 中的路径；

说明：如果 LOG 没有指定路径时，使用 DIRECTORY 中指定的路径；

如果 LOG 和 DIRECTORY 都没有指定路径，则使用系统当前路径。

LOG 文件记载了导出对象的统计信息。供用户查看导出内容。

LOG 内容具体介绍如下：

1. 如果导出级别为 TABLES，那么日志信息包含一张或多张表的索引、权限、数据行数。
2. 如果导出级别为 SCHEMAS，那么日志信息只包含一个或多个模式中的对象统计信息，

每个模式下的内容具体有：

- 导出 SEQUENCE、VIEW、COMMENT、PROCEDURE、SYNONYM、DBLINK、TRIGGER、PACKAGE、OBJECT、CLASS\_BODY 和 DOMAIN 等对象的个数和名称，以及对象的权限。
- 依次导出表的索引、权限、数据行数。

3. 如果导出级别为 OWNER，那么日志内容包含一个或多个用户所拥有的所有模式中的对象统计信息，每个模式下的内容和导出级别为 SCHEMAS 时一样。

4. 如果导出级别为 FULL，那么日志内容除了包含所有模式中的对象信息，还包括系统包和系统视图。

和 LOG 有关的参数还有 NOLOGFILE、NOLOG 和 LOG\_WRITE。NOLOGFILE=Y 用于设置不使用日志文件。NOLOG=Y 用于设置不在屏幕上显示日志信息。LOG\_WRITE=Y 用于设置日志的实时打印（到日志文件）功能。

#### 2.3.1.4 DIRECTORY

DIRECTORY 用于指定导出文件及日志文件生成的路径。可选参数，缺省为导出到 dexp 所在路径。

语法如下：

---

```
DIRECTORY=<path>
```

---

<path>：导出文件和日志文件生成的路径。

如果 FILE 和 LOG 参数指定的文件包含生成路径，则 FILE 和 LOG 参数中指定的路径将替代 DIRECTORY 所指定的路径；如果 FILE 和 LOG 参数指定的文件未包含路径信息，则文

件将被生成到 DIRECTORY 指定的目录下；如果都没有指定路径，程序将根据当前的运行环境来设置相应的导出路径，一般为当前路径。

例如，没有指定路径，使用当前路径。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
```

例如，指定路径 /mnt/data/dexp。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
DIRECTORY=/mnt/data/dexp
```



**FILE** 和 **LOG** 中指定的路径，优先级高于 **DIRECTORY** 中指定的路径。

说明：当 **FILE** 和 **LOG** 不指定路径时，才使用 **DIRECTORY** 指定的路径。

如果都没有指定路径，则使用系统当前路径。

### 2.3.1.5 四种级别的导出方式

针对数据库对象，有 FULL、OWNER、SCHEMAS、TABLES 四种导出方式可供选择。一次导出只能指定一种方式。可选参数，缺省为 SCHEMAS。

#### 2.3.1.5.1 FULL

FULL 方式导出数据库的所有对象。

语法如下：

---

```
FULL=Y
```

---

例如，设置 FULL=Y，导出数据库的所有对象，导出数据库文件和日志文件放在路径 /mnt/data/dexp 下。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log
FULL=Y DIRECTORY=/mnt/data/dexp
```

#### 2.3.1.5.2 OWNER

OWNER 方式导出一个或多个用户拥有的所有对象。

语法如下：

---

OWNER=<用户名>{,<用户名>}

---

例如，设置 OWNER=USER01，导出用户 USER01 所拥有的对象全部导出。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log OWNER=USER01
DIRECTORY=/mnt/data/dexp
```

### 2.3.1.5.3 SCHEMAS

SCHEMAS 方式的导出一个或多个模式下的所有对象。

语法如下：

---

SCHEMAS=<模式名>{,<模式名>}

---

例如，设置 SCHEMAS=USER01，导出模式 USER01 模式下的所有对象。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log
SCHEMAS=USER01 DIRECTORY=/mnt/data/dexp
```



说明：

一般情况下，OWNER 与 SCHEMAS 导入导出是相同的。

但是用户可以包含多个模式，在这种情况下 SCHEMAS 的导入导出是 OWNER 导入导出的一个子集。

### 2.3.1.5.4 TABLES

TABLES 方式导出和导入一个或多个指定的表或表分区。导出所有数据行、约束、索引等信息。

语法如下：

---

TABLES=<表名>{,<表名>}

---

例如，设置 TABLES=table1,table2，导出 table1,table2 两张表的所有数据和信息。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log TABLES=table1,table2
DIRECTORY=/mnt/data/dexp
```

和 TABLES 导出有关的参数还有 QUERY、EXCLUDE 和 INCLUDE，都是用来设置过滤条

件的。

### 2.3.1.5.5 OWNER、SCHEMAS、TABLES 的书写规定

#### 1. 用户名、模式名和表名的书写规定

- 对于输入的用户名、模式名和表名，工具都会自动转换为大写。

例如：输入的命令 TABLES=t1，那么实际处理的表是 T1。

- 对于小写的用户名、模式名和表名，为了不转换为大写，需要转义。

转义步骤如下：第一步，如果想输入小写的名字不被转成大写，dexp 要求将名称使用双引号扩起来。例如：名称为 tab1，要写成"tab1"。第二步，另外，操作系统要求将传入的名称使用双引号括起来作为一个整体，同时要求内层的双引号还需要被转义。所以最终写成 "\"tab1\""

例如，表名为小写 tabname，语句中应该写为 "\"tabname\""

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log
TABLES="\"tabname\""
```

- 如果含有特殊字符（双引号、单引号、空格、逗号等）的用户名、模式名和表名需要转义。

具体规则如下：

转义步骤如下：第一步，dexp 要求对含有特殊字符的对象名称需要使用双引号括起来，同时特殊字符需要被转义（使用双引号"）。例如：表名为 tab1"，需写成"tab1""；第二步，另外，操作系统要求将传入的名称使用双引号括起来作为一个整体，同时要求内层的双引号和特殊字符还需要被转义。所以最终写成： "\"tab1\"\"\""

例如，表名为小写 tabname"，语句中应该写为 "\"tabname\"\"\""

```
./dexp USERID=SYSDBA/SYSDBA@192.168.0.248:8888 FILE=db_str.dmp
LOG=db_str.log TABLES="\"tabname\"\"\""
```

#### 2. 如何转义双引号

- 1) dexp 和 dimp 要求使用双引号"对双引号内的双引号转义。
- 2) WINDOWS 命令行，使用双引号"或者反斜杠\对双引号内的双引号转义。
- 3) LINUX 命令行，使用反斜杠\对双引号内的双引号转义。

## 2.3.2 普通参数

普通参数中介绍了具有一些专门用途的可选参数。

### 2.3.2.1 FUZZY\_MATCH

FUZZY\_MATCH 用于指定 TABLES 选项是否支持模糊匹配。可选参数，缺省为 N。

语法如下：

---

```
FUZZY_MATCH=N/Y
```

---

N: TABLES 选项不支持模糊匹配，指定的表名与数据库中的表名必须精确匹配；

Y: TABLES 选项支持模糊匹配，指定的表名与数据库中的表名采用 LIKE 模糊查询匹配。

例如，导出 OTHER 模式下以 "R" 开头的表。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log TABLES=OTHER.R%  
FUZZY_MATCH=Y
```

### 2.3.2.2 QUERY

QUERY 用于指定过滤条件来对表数据进行导出。可选参数。

语法如下：

---

```
QUERY="<where_condition>"
```

---

<where\_condition>: 过滤条件。更详细了解请参考《DM8\_SQL 语言使用手册》的 <WHERE 子句>。

例如，导出 OTHER.READER 表中满足 "WHERE AGE=19" 条件的数据。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log TABLES=OTHER.READER  
QUERY="WHERE AGE=19"
```

### 2.3.2.3 PARALLEL

PARALLEL 用于指定导出的过程中所使用的线程数目。可选参数，缺省为单线程。如果 CPU 核心数为 N 的话，那一般来说 PARALLEL 为 N 或者 N+1 最合适。

语法如下：

---

```
PARALLEL=<num>
```

---



<num>: 线程数。整数类型，取值范围：1-100。

例如，设置 PARALLEL=2 时则线程数为 2。

```
./dexp      USERID=SYSDBA/SYSDBA      FILE=db_str.dmp      LOG=db_str.log      FULL=Y
DIRECTORY=/mnt/data/dexp PARALLEL=2
```

#### 2.3.2.4 TABLE\_PARALLEL

TABLE\_PARALLEL 用于指定导出每张表所使用的线程数。可选参数，缺省为单线程。

如果 CPU 核数为 N 的话，那一般来说 TABLE\_PARALLEL 为 N 或者 N+1 最合适。在 MPP 模式下会转换成单线程。

语法如下：

```
TABLE_PARALLEL=<num>
```

<num>: 线程数。整数类型，取值范围 1-50。

例如，设置 TABLE\_PARALLEL=3 时则线程数为 3。

```
./dexp      USERID=SYSDBA/SYSDBA      FILE=db_str.dmp      LOG=db_str.log      FULL=Y
DIRECTORY=/mnt/data/dexpTABLE_PARALLEL=3
```

#### 2.3.2.5 TABLE\_POOL

TABLE\_POOL 用于设置导出过程中存储表数据的缓冲区个数。可选参数。

语法如下：

```
TABLE_POOL=<num>
```

<num>: 存储表数据的缓冲区个数。缺省值为 1。整数类型，取值范围：1-20。



**TABLE\_PARALLEL** 固定的情况下，导出时间随着 **TABLE\_POOL** 个数的增加而减少。当 **TABLE\_POOL** 稍大于 **TABLE\_PARALLEL** 时结果为最优。

例如，设置 TABLE\_POOL=3 时。

```
./dexp      USERID=SYSDBA/SYSDBA      FILE=db_str.dmp      LOG=db_str.log      FULL=Y
DIRECTORY=/mnt/data/dexpTABLE_POOL=3
```

### 2.3.2.6 EXCLUDE

EXCLUDE 用来批量设置导出时忽略的对象种类。可选参数。

语法如下：

---

```
EXCLUDE=(<对象种类名>{,<对象种类名>})
```

或者

```
EXCLUDE=TABLES:<表名>{,<表名>}
```

或者

```
EXCLUDE=SCHEMAS:<模式名>{,<模式名>}
```

---

<对象种类名>：包括 CONSTRAINTS、INDEXES、ROWS、TRIGGERS、GRANTS。



如果导出方式为 **TABLES** 表级或 **SCHEMAS** 模式级导出时，  
说明： **EXCLUDE=SCHEMAS:<模式名>{,<模式名>}**排除模式不起作用。

例如，设置 EXCLUDE=(CONSTRAINTS)，将当前用户下的 table1、table2 的对象信息导出时不导出约束。

```
./dexp          SYSDBA/SYSDBA          FILE=/mnt/data/db_str.dmp          LOG=db_str.log
TABLES=table1,table2  EXCLUDE=/(CONSTRAINTS/)
```

### 2.3.2.7 INCLUDE

INCLUDE 用来批量设置导出时包含的对象种类。可选参数。

语法如下：

---

```
INCLUDE=(<对象种类名>{,<对象种类名>})
```

或者

```
INCLUDE=TABLES:<表名>{,<表名>}
```

---

<对象种类名>：CONSTRAINTS、INDEXES、ROWS、TRIGGERS 或 GRANTS。

例如，设置 INCLUDE=(CONSTRAINTS,INDEXES)，指明将当前用户下的 table1 的对象约束和索引信息导出。

```
./dexp  USERID=SYSDBA/SYSDBA  FILE=db_str.dmp  LOG=db_str.log  TABLES=table1
DIRECTORY=/mnt/data/dexp  INCLUDE=/(CONSTRAINTS,INDEXES/)
```

### 2.3.2.8 CONSTRAINTS、TABLESPACE、GRANTS、INDEXES、TRIGGERS、ROWS

这些参数用来指定某类对象是否被导出。类别分别为约束、表空间、权限、索引、触发器和数据。

语法如下：

---

CONSTRAINTS=N/Y

TABLESPACE= Y/N

GRANTS=N/Y

INDEXES=N/Y

TRIGGERS=N/Y

ROWS=N/Y

---

CONSTRAINTS=N：设置不导出约束。可选参数。缺省该参数为 Y，导出约束。

TABLESPACE=Y：设置导出的对象定义中包含表空间。可选参数。缺省该参数为 N，不包含表空间。

GRANTS=N：设置不导出权限。可选参数。缺省该参数为 Y，导出权限。

INDEXES=N：设置不导出索引。可选参数。缺省该参数为 Y，导出索引。

TRIGGERS=N：设置不导出触发器。可选参数。缺省该参数为 Y，导出触发器。

ROWS=N：设置不导出数据。可选参数。缺省该参数为 Y，导出数据。

例如，设置 ROWS=N 不导出表中的数据。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp NOLOGFILE=Y FULL=Y
```

```
DIRECTORY=/mnt/data/dexp ROWS=N
```



说明：

此处，对象（CONSTRAINTS、GRANTS、INDEXES、TRIGGERS、ROWS）分别单独设置，和 EXCLUDE/INCLUDE 中批量设置功能一样。设置一个即可。

如果单独设置和批量设置同时出现时，那么以最后出现的那个为准。

### 2.3.2.9 NOLOGFILE

NOLOGFILE 用于设置是否使用日志文件。可选参数。

语法如下：

---

NOLOGFILE=Y/N

---

Y：不使用。

N 或缺省参数：使用。



一旦设置 **NOLOGFILE=Y** 不使用日志文件。此时即使存在 **LOG** 参数，**LOG**

**说明：** 参数也为无效。

例如，设置 NOLOGFILE=Y，不使用日志文件。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp NOLOGFILE=Y FULL=Y
DIRECTORY=/mnt/data/dexp
```

### 2.3.2.10 NOLOG

NOLOG 设置屏幕上是否显示日志信息。可选参数。

语法如下：

---

NOLOG=Y/N

---

Y：不显示。

N 或缺省参数：显示。

例如，设置 NOLOG=Y 屏幕不显示日志信息。

```
./dexp USERID=SYSDBA/SYSDBA FILE=db_str.dmp NOLOG=Y FULL=Y
DIRECTORY=/mnt/data/dexp
```

### 2.3.2.11 LOG\_WRITE

LOG\_WRITE 用于日志信息实时写入日志文件。可选参数。

语法如下：

---

LOG\_WRITE=Y/N

---

Y：一边将日志内容打印到屏幕上，一边将日志内容写入日志文件。

N 或缺省参数：日志内容先在屏幕上全部打印完毕再写入日志文件。

### 2.3.2.12 DUMMY

DUMMY 用于设置交互信息处理。可选参数，缺省使用 P 打印交互信息。

语法如下：

---

```
DUMMY=P/Y/N
```

---

P：打印交互信息，默认方式。

Y：不打印交互信息，所有交互都按 YES 处理。

N：不打印交互信息，所有交互都按 NO 处理。

### 2.3.2.13 PARFILE

PARFILE 用于将常用的参数设置保存到文件中，然后使用参数文件 PARFILE 进行导出、导入操作。可选参数。

语法如下：

---

```
PARFILE=<path>
```

---

<path>：PARFILE 文件的绝对路径。

例如，设置 PARFILE=/mnt/data/dexp/para.txt。

```
./dexp USERID=SYSDBA/SYSDBA PARFILE=/mnt/data/dexp/para.txt
```

其中，参数文件/mnt/data/dexp/para.txt 的内容如下：

```
FILE=db_str.dmp  
LOG=db_str.log  
TABLES=table1  
DIRECTORY=/mnt/data/dexp
```

### 2.3.2.14 FEEDBACK

FEEDBACK 用来指明在导出数据过程中每间隔多少行打印一次进度信息。可选参数。

当参数缺省时，默认为 0，只打印导出表的总行数。

语法如下：

---

```
FEEDBACK=<num>
```

---

<num>：表间隔的行数。正整数。

例如，将整个数据库导出到文件 dexp.dmp 中，并且在导出的过程中每隔 100 行打印一次进度信息。

```
./dexp SYSDBA/SYSDBA FULL=Y FILE=dexp.dmp DIRECTORY=/mnt/data/dexp FEEDBACK=100
```

### 2.3.2.15 COMPRESS

COMPRESS 指定是否压缩导出文件。可选参数。缺省为不压缩。

语法如下：

---

```
COMPRESS=Y/N
```

---

Y：压缩。

N 或缺省参数：不压缩

### 2.3.2.16 ENCRYPT、ENCRYPT\_PASSWORD、ENCRYPT\_NAME

这三个参数用来设置导出文件加密。可选参数。

语法如下：

---

```
ENCRYPT=Y
```

---

```
ENCRYPT_PASSWORD=<加密密钥>
```

```
ENCRYPT_NAME=<加密算法>
```

---

ENCRYPT=Y：设置导出文件加密。如果设置 ENCRYPT=Y，那么默认 COMPRESS=Y，先压缩后加密。

ENCRYPT\_PASSWORD=<加密密钥>：设置加密密钥。

ENCRYPT\_NAME=<加密算法>：设置加密算法。缺省为 RC4。



**加密算法：**

**说明：**

DES\_ECB 、DES\_CBC 、DES\_CFB、DES\_OFB、DESEDE\_ECB、  
DESEDE\_CBC 、DESEDE\_CFB、DESEDE\_OFB 、AES128\_ECB 、  
AES128\_CBC 、AES128\_CFB 、AES128\_OFB 、AES192\_ECB 、  
AES192\_CBC 、AES192\_CFB 、AES192\_OFB 、AES256\_ECB 、  
AES256\_CBC 、AES256\_CFB 、AES256\_OFB 、RC4

例如，设置导出文件加密，密钥 abcdefg，加密算法为 DES\_CBC。

```
./dexp SYSDBA/SYSDBA DIRECTORY=/mnt/data/dexp FILE=dk15.dmp LOG=db_str15.log
ENCRYPT=Y ENCRYPT_PASSWORD=abcdefg ENCRYPT_NAME=DES_CBC FULL=Y
```



**ENCRYPT、ENCRYPT\_PASSWORD、ENCRYPT\_NAME** 经常一起使用。

**注意：**如果只设置 **ENCRYPT\_PASSWORD** 一个，那么默认 **ENCRYPT=Y**，**ENCRYPT\_NAME** 缺省值 **RC4**。

### 2.3.2.17 FILESIZE

**FILESIZE** 用于指定单个导出文件（**FILE**）大小或日志文件（**LOG**）大小的上限。可以按字节[B]、K[B]、M[B]、G[B]的方式指定大小。可选参数。

使用 **FILESIZE** 参数时，**FILE**（或 **LOG**）必须使用%U 对名称进行自动扩展，否则报错。如果不指定 **FILE**（或 **LOG**），那么即使设置了 **FILESIZE** 也不起作用。**FILE** 和 **LOG** 同时使用%U 时，**FILESIZE** 参数只需指定一次。

语法如下：

```
FILESIZE=<num>
```

**<num>**：单个导出文件上限。整数类型，取值范围：128M-系统最大值。

以下示例表示以 **FULL** 方式导出实例，每个导出文件大小为 128m，文件名以 data01.dmp 自动扩展，存储位置为/mnt/data/dexp 目录。

```
./dexp USERID=SYSDBA/SYSDBA DIRECTORY=/mnt/data/dexp FILE=data%u.dmp
LOG=data%u.log FILESIZE=128m FULL=Y
```



**注意：****dexp** 执行过程中可能遇到单个文件过大的问题。解决办法是通过 **FILESIZE** 参数来设置单个导出文件上限，导出文件名称按照 **FILE** 参数中描述的使用%u 来自动扩展。这样在按照 **FILESIZE** 大小导出到第一个文件之后，会自动生成下一个导出文件。

### 2.3.2.18 FILENUM

**FILENUM** 用于指定一个模板导出文件的个数。范围为 1~99，默认 99。可选参数。

FILENUM 需要和 FILESIZE 搭配使用，FILENUM\*FILESIZE 要大于等于导出的文件总大小，否则会报错空间不足。

### 2.3.2.19 DROP

DROP 用于设置导出后是否删除原表。可选参数。

语法如下：

---

DROP=Y/N

---

Y: 导出后删除原表，但不级联删除。

N 或缺省参数: 导出后不删除原表。

### 2.3.2.20 DESCRIBE

DESCRIBE 用于设置导出数据文件的描述信息，记录在数据文件中。可选参数。

语法如下：

---

DESCRIBE='字符串'

---

字符串: 设置导出数据文件的描述信息，记录在数据文件中。缺省时，导出的数据文件中没有描述信息。

### 2.3.2.21 LOCAL

LOCAL 只用于在 MPP 环境下，是否使用 LOCAL 方式登录，缺省为不使用。可选参数。

语法如下：

---

LOCAL=Y/N

---

Y: 表示用户在登录时为 LOCAL 本地登录方式，即只登录指定的节点，操作数据也只是本节点的数据。

N: 表示不使用，为 GLOBAL 全局登录方式，即操作数据是所有节点上相关数据。

### 2.3.2.22 HELP

输入 dexp HELP 即可查看帮助信息。

语法如下：



---

HELP

---

例如，使用 HELP 查看帮助信息。

```
./dexp HELP
```

HELP 信息会逐个显示所有参数的大致信息，供用户快速参考。

## 3 dimp 逻辑导入

dimp 逻辑导入工具利用 dexp 工具生成的备份文件对本地或远程的数据库进行联机逻辑还原。dimp 导入是 dexp 导出的相反过程。还原的方式可以灵活选择，如是否忽略对象存在而导致的创建错误、是否导入约束、是否导入索引、导入时是否需要编译、是否生成日志等。

### 3.1 使用 dimp 工具

dimp 工具需要从命令行启动。在 cmd 命令行工具中找到 dimp 所在安装目录 /dmdbms/bin，输入 dimp 和参数后回车。参数在下一节详细介绍。

语法如下：

```
dimp PARAMETER=value { PARAMETER=value }
```

PARAMETER: dimp 参数。多个参数之间排列顺序无影响，参数之间使用空格间隔。

value: 参数取值。

例如，将逻辑备份采用 FULL 方式完全导入到用户名和密码为 SYSDBA，IP 地址为 192.168.0.248，端口号为 8888 的数据库。导入文件名为 db\_str.dmp，导入的日志文件名为 db\_str.log，路径为 /mnt/data/dexp。

```
./dimp          USERID=SYSDBA/SYSDBA@192.168.0.248:8888          FILE=db_str.dmp
DIRECTORY=/mnt/data/dexp LOG=db_str.log FULL=Y
```

### 3.2 dimp 参数一览表

本节提供 dimp 的参数一览表，供用户快速参考。

表 3.1 dimp 参数一览表

参数	含义	备注
USERID	用户名/口令@主库名:端口号#证书路径例如: SYSDBA/SYSDBA*MPP_TYPE @server:5236#ssl_path@ssl_pwd	必选。其中主库名、端口号和证书路径为可选项
FILE	输入文件，即 dexp 导出的文件	必选
DIRECTORY	导入文件所在目录	可选

FULL	导入整个数据库 (N)	可选，四者中选其一。缺省为 SCHEMAS
OWNER	导入指定的用户名下的模式	
SCHEMAS	导入的模式列表	
TABLES	表名列表，指定导入的 tables 名称。不支持对外部表进行导入	
PARALLEL	用于指定导入的过程中所使用的线程数目	可选
TABLE_PARALLEL	用于指定导入的过程中每个表所使用的子线程数目	可选。在 FAST_LOAD 为 Y 时有效
IGNORE	忽略创建错误 (N)。如果表已经存在则向表中插入数据，否则报错表已经存在。	可选
TABLE_EXISTS_ACTION	需要的导入表在目标库中存在时采取的操作 [SKIP   APPEND   TRUNCATE   REPLACE]	可选
FAST_LOAD	是否使用 dmfldr 进行数据导入 (N)	可选
FLDR_ORDER	使用 dmfldr 是否需要严格按顺序来导数据 (Y)	可选
COMMIT_ROWS	批量提交的行数 (5000)	可选
EXCLUDE	忽略指定的对象 (CONSTRAINTS, INDEXES, ROWS, TRIGGERS, GRANTS)。格式 EXCLUDE=(CONSTRAINTS, INDEXES, ROWS, TRIGGERS, GRANTS)	可选。例如 EXCLUDE=( CONSTRAINT )
GRANTS	导入权限 (Y)	可选
CONSTRAINTS	导入约束 (Y)	可选
INDEXES	导入索引 (Y)	可选
TRIGGERS	导入触发器 (Y)	可选
ROWS	导入数据行 (Y)	可选
LOG	日志文件	可选
NOLOGFILE	不使用日志文件 (N)	可选
NOLOG	屏幕上不显示日志信息 (N)	可选
DUMMY	交互信息处理：打印 (P)。取值 Y/N。	可选

## DM8\_dexp 和 dimp 使用手册

	Y: 打印所有交互信息。N: 不打印交互信息。	
LOG_WRITE	日志信息实时写入文件 (N)	可选
PARFILE	参数文件名, 如果 dimp 的参数很多, 可以存成参数文件	可选
FEEDBACK	显示每 x 行 (0) 的进度	可选
COMPILE	编译过程, 程序包和函数 (Y)	可选
INDEXFILE	将表的索引/约束信息写入指定的文件	可选
INDEXFIRST	导入时先建索引 (N)	可选
REMAP_SCHEMA	SOURCE_SCHEMA: TARGET_SCHEMA 将 SOURCE_SCHEMA 中的数据导入到 TARGET_SCHEMA 中	可选
ENCRYPT_PASSWORD	数据的加密密钥	可选。和 dexp 中的 ENCRYPT_PASSWORD 设置的密钥一样
ENCRYPT_NAME	数据的加密算法的名称	可选。和 dexp 中的 ENCRYPT_NAME 设置的加密算法一样
SHOW/ DESCRIBE	只列出文件内容 (N)	可选
LOCAL	MPP 环境下, 是否使用 LOCAL 方式登录 (N)	可选
TASK_THREAD_NUMBER	设置 dmfldr 处理用户数据的线程数目	可选
BUFFER_NODE_SIZE	设置 dmfldr 读入文件缓冲区大小	可选
TASK_SEND_NODE_NUMBER	用于设置 dmfldr 发送节点个数 [16, 65535]	可选
LOB_NOT_FAST_LOAD	如果一个表含有大字段, 那么不使用 dmfldr, 因为 dmfldr 是一行一行提交的	可选
PRIMARY_CONFLICT	主键冲突的处理方式 [IGNORE OVERWRITE], 默认报错	可选
TABLE_FIRST	是否强制先导入表 (默认 N), Y 表示先导入表, N	可选

	正常导入	
HELP	显示帮助信息	可选

### 3.3 dimp 参数详解

#### 3.3.1 特殊参数

特殊参数中介绍了必选参数的和最常用的可选参数。

##### 3.3.1.1 USERID

USERID 用于连接服务器，必选参数。用法请参考 dexp 中 USERID。

##### 3.3.1.2 FILE

FILE 用于指定明确导入 dexp 导出的文件（可以包含路径）。为必选参数。

语法如下：

---

FILE=<文件>

---

例如，之前导出的文件路径为：FILE=/mnt/data/dexp/db\_str.dmp，那么导入文件可以写成如下所示：

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
DIRECTORY=/mnt/data/dimp
```

若 FILE 没有指定文件路径，则可在 DIRECTORY 中指定。



说明：

如果 **FILE** 指定的文件包含生成路径，则忽略 **DIRECTORY** 中的路径；

如果 **FILE** 没有指定路径时，使用 **DIRECTORY** 中指定的路径；

如果 **FILE** 和 **DIRECTORY** 都没有指定路径，则使用系统当前路径。

##### 3.3.1.3 LOG

LOG 用于明确指定导入的日志文件名称（可以包含路径），可以包含多个文件，用逗号分隔。可选参数。如果不选用 LOG 参数（来明确指定日志文件名称），那么默认导出日志文件名称为 dimp.log。

语法如下：

---

LOG=<文件名>

---

例如，指定日志文件名为 db\_str.log。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log FULL=Y
DIRECTORY=/mnt/data/dimp
```



如果 LOG 指定的文件包含生成路径，则忽略 DIRECTORY 中的路径；

**说明：** 如果 LOG 没有指定路径时，使用 DIRECTORY 中指定的路径；

如果 LOG 和 DIRECTORY 都没有指定路径，则使用 FILE 指定的路径。

LOG文件记载了导出对象的统计信息。供用户查看导出内容。详细的日志内容和dexp LOG中一样，请参考dexp的LOG。

和 LOG 有关的参数还有 NOLOGFILE、NOLOG 和 LOG\_WRITE。NOLOGFILE=Y 用于设置不使用日志文件。NOLOG=Y 用于设置不在屏幕上显示日志信息。LOG\_WRITE=Y 用于设置日志的实时打印（到日志文件）功能。

### 3.3.1.4 DIRECTORY

DIRECTORY 用于指定导入文件及日志文件生成的路径。可选参数，缺省为 dimp 所在路径。

语法如下：

---

DIRECTORY=<path>

---

<path>：导入文件路径和日志文件生成的路径。

如果 FILE 和 LOG 参数指定的文件包含生成路径，则 FILE 和 LOG 参数中指定的路径将替代 DIRECTORY 所指定的路径；如果 FILE 和 LOG 参数指定的文件未包含路径信息，则导入文件路径和日志生成的文件路径都是 DIRECTORY 指定的路径；如果都没有指定路径，程序将根据当前的运行环境来设置相应的导出路径，一般为当前路径。

例如，没有指定路径，使用当前路径。

```
./dimp USERID=SYSDBA/SYSDBA FILE=db_str.dmp LOG=db_str.log FULL=Y
```

例如，指定 FILE 使用 /mnt/data/dexp 路径，LOG 使用 /mnt/data/dimp 路径。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp db_str.dmp LOG=db_str.log
```

```
FULL=Y DIRECTORY=/mnt/data/dimp
```



**FILE** 和 **LOG** 中指定的路径，优先级高于 **DIRECTORY** 中指定的路径。

**说明：** 当 **FILE** 和 **LOG** 不指定路径时，才使用 **DIRECTORY** 指定的路径。

如果都没有指定路径，则使用系统当前路径。

### 3.3.1.5 四种级别的导入方式

针对数据库对象，有 **FULL**、**OWNER**、**SCHEMAS**、**TABLES** 四种导入方式可供选择。一次导出只能指定一种方式。可选参数，缺省为 **SCHEMAS**。



**小窍门：**

导入方式可以和导出方式不一致。

例如：当导出方式不为 **FULL** 时，导入时也可以指定为 **FULL** 方式，此时的 **FULL** 会将导出文件包含的全部内容导入到指定的实例中。

#### 3.3.1.5.1 FULL

**FULL** 方式导入整个数据库。

语法如下：

```
FULL=Y
```

例如，设置 **FULL=Y**，导入整个数据库，导入的数据库文件在 `/mnt/data/dexp`，即将生成的日志文件放在 `/mnt/data/dimp`。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log FULL=Y
DIRECTORY=/mnt/data/dimp
```

#### 3.3.1.5.2 OWNER

**OWNER** 方式导入一个或多个用户拥有的所有对象。

语法如下：

```
OWNER=<用户名>{,<用户名>}
```

例如，设置 OWNER=USER01，导入用户 USER01 所拥有的对象全部导出。导入的数据库文件在 /mnt/data/dexp，即将生成的日志文件放在 /mnt/data/dimp。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
OWNER=USER01 DIRECTORY=/mnt/data/dimp
```

### 3.3.1.5.3 SCHEMAS

SCHEMAS 方式的导入一个或多个模式下的所有对象。

语法如下：

---

```
SCHEMAS=<模式名>{,<模式名>}
```

---

例如，设置 SCHEMAS=USER01，导入模式 USER01 模式下的所有对象。导入的数据库文件在 /mnt/data/dexp，即将生成的日志文件放在 /mnt/data/dimp。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
SCHEMAS=USER01 DIRECTORY=/mnt/data/dimp
```



说明：

一般情况下，OWNER 与 SCHEMAS 导入导出是相同的。

但是用户可以包含多个模式，在这种情况下 SCHEMAS 的导入导出是 OWNER 导入导出的一个子集。

### 3.3.1.5.4 TABLES

TABLES 方式导入一个或多个指定的表或表分区。导入所有数据行、约束、索引等信息。

语法如下：

---

```
TABLES=<表名>{,<表名>}
```

---

例如，设置 TABLES=table1,table2，导入 table1,table2 两张表的所有数据和信息。导入的数据库文件在 /mnt/data/dexp，即将生成的日志文件放在 /mnt/data/dimp。

```
./dimp  USERID=SYSDBA/SYSDBA  FILE=/mnt/data/dexp/db_str.dmp  LOG=db_str.log
TABLES=table1,table2 DIRECTORY=/mnt/data/dimp
```

和 TABLES 导入有关的参数还有 EXCLUDE，用来指定导入时过滤掉某类对象。



### 3.3.2 普通参数

#### 3.3.2.1 PARALLEL

PARALLEL 用于指定导入的过程中所使用的线程数目。可选参数，缺省为单线程。如果 CPU 核数为 N 的话，那一般来说 PARALLEL 为 N 或者 N+1 最合适。

语法如下：

---

PARALLEL=<num>

---

<num>：线程数。整数类型，取值范围：1-100。

例如，设置 PARALLEL=2 时则线程数为 2。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log FULL=Y  
DIRECTORY=/mnt/data/dimp PARALLEL=2
```

#### 3.3.2.2 TABLE\_PARALLEL

TABLE\_PARALLEL 在 FAST\_LOAD 为 Y 时有效，用于指定导入每张表所使用的线程数。可选参数，缺省为单线程。如果 CPU 核数为 N 的话，那一般来说 TABLE\_PARALLEL 为 N 或者 N+1 最合适。在 MPP 模式下会转换成单线程。

语法如下：

---

TABLE\_PARALLEL=<num>

---

<num>：线程数。整数类型，取值范围 1-16。

例如，设置 TABLE\_PARALLEL=3 时则线程数为 3。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log FULL=Y  
DIRECTORY=/mnt/data/dimp TABLE_PARALLEL=3
```

#### 3.3.2.3 IGNORE

IGNORE 指定忽略创建数据库对象(表、类、java 类、domain、sequence、comment、view、synonym、trigger、package、dblink、user、存储过程/函数、role 对象创建、权限授权语句、索引、约束等)错误。可选参数。缺省为 N，不忽略。

语法如下：

---

IGNORE=Y

---

IGNORE=Y: 忽略导入时所产生的创建数据库对象错误并继续执行。当缺省或 IGNORE=N 时, 不忽略导入时所产生的创建数据库对象错误, 遇到错误即停止执行并报错。

### 3.3.2.4 TABLE\_EXISTS\_ACTION

TABLE\_EXISTS\_ACTION 用于要导入的表已经存在时的处理方式。默认为直接报错。

语法如下:

---

TABLE\_EXISTS\_ACTION=[SKIP | APPEND | TRUNCATE | REPLACE]

---

SKIP: 跳过此表。

APPEND: 直接向现有表中导入数据

TRUNCATE: 先删除现有表中的数据, 再向表中导入数据

REPLACE: 先删除现有表, 再导数据

### 3.3.2.5 FAST\_LOAD

FAST\_LOAD 用于设置是否使用 dmfldr 进行数据导入, 默认为 N。

语法如下:

---

FAST\_LOAD=Y/N

---

Y: 是。

N: 否。



当 FAST\_LOAD=Y 时才可以设置 TABLE\_PARALLEL 参数, 否则  
注意: TABLE\_PARALLEL 参数无效。

### 3.3.2.6 FLDR\_ORDER

FLDR\_ORDER 用于设置使用 dmfldr 时, 是否严格按导出顺序来导入数据。可选参数。

语法如下:

---

FLDR\_ORDER=Y/N

---

Y 或缺省 FLDR\_ORDER 参数: 表示按照导出顺序来导入数据。

N: 表示不按照导出顺序来导入, 数据的创建为随机生成。

### 3.3.2.7 COMMIT\_ROWS

COMMIT\_ROWS 用于设置批量提交的行数，默认为 5000 行。用户可以根据实际情况调整每次提交的行数，以达到性能的最佳点。一般情况下可不设置。可选参数。

语法如下：

---

```
COMMIT_ROWS=<num>
```

---

<num>: 行数。取值范围：1-1000000。

### 3.3.2.8 EXCLUDE

EXCLUDE 用来批量设置导入时忽略的对象种类。可选参数。

语法如下：

---

```
EXCLUDE=(<对象种类名>{,<对象种类名>})
```

---

<对象种类名>: 包括 CONSTRAINTS、INDEXES、ROWS、TRIGGERS、GRANTS。

例如，设置 EXCLUDE=(CONSTRAINTS)，将当前用户下的 table1、table2 的对象信息导入时不导入约束。

```
./dimp      SYSDBA/SYSDBA      FILE=/mnt/data/db_str.dmp      LOG=db_str.log
TABLES=table1,table2  EXCLUDE=/(CONSTRAINTS/) DIRECTORY=/mnt/data/dimp
```

### 3.3.2.9 GRANTS、CONSTRAINTS、INDEXES、TRIGGERS、ROWS

这些参数用来指定某类对象是否被导入。类别分别为约束、权限、索引、触发器和数据。

---

```
GRANTS=N/Y
```

```
CONSTRAINTS=N/Y
```

```
INDEXES=N/Y
```

```
TRIGGERS=N/Y
```

```
ROWS=N/Y
```

---

GRANTS=N: 设置不导入权限。可选参数。缺省该参数为导入权限。

CONSTRAINTS=N: 设置不导入约束。可选参数。缺省该参数为导入约束。

INDEXES=N: 设置不导入索引。可选参数。缺省该参数为导入索引。

TRIGGERS=N: 设置不导入触发器。可选参数。缺省该参数为导入触发器。

ROWS=N: 设置不导入数据。可选参数。缺省该参数为导入数据。

例如, 设置 ROWS=N 不导入表中的数据。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp NOLOGFILE=Y FULL=Y  
DIRECTORY=/mnt/data/dimp ROWS=N
```

### 3.3.2.10 NOLOGFILE

NOLOGFILE 用于设置是否使用日志文件。可选参数。

语法如下:

---

NOLOGFILE=Y/N

---

Y: 不使用日志文件。此时即使存在 LOG 参数, LOG 参数也为无效。

N 或参数缺省: 使用日志文件。

例如, 设置 NOLOGFILE=Y, 不使用日志文件。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp NOLOGFILE=Y FULL=Y
```

### 3.3.2.11 NOLOG

NOLOG 设置屏幕上是否显示日志信息。可选参数。默认为 N, 显示日志信息。

语法如下:

---

NOLOG=Y/N

---

Y: 不显示。

N: 显示。

例如, 设置 NOLOG=Y 屏幕不显示日志信息。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log  
NOLOG=Y FULL=Y DIRECTORY=/mnt/data/dimp
```

### 3.3.2.12 LOG\_WRITE

LOG\_WRITE 用于日志信息实时写入日志文件。可选参数。

语法如下:

---

LOG\_WRITE=Y/N

---

Y: 一边将日志内容打印到屏幕上, 一边将日志内容写入日志文件。

N 或缺省参数：日志内容先在屏幕上全部打印完毕再写入日志文件。

### 3.3.2.13 DUMMY

DUMMY 用于设置交互信息处理。可选参数，缺省使用 P 打印交互信息。

语法如下：

---

```
DUMMY=P/Y/N
```

---

P：打印交互信息，默认方式。

Y：不打印交互信息，所有交互都按 YES 处理。

N：不打印交互信息，所有交互都按 NO 处理。

### 3.3.2.14 PARFILE

PARFILE 用于将常用的参数设置保存到文件中，然后使用参数文件 PARFILE 进行导出、导入操作。可选参数。

语法如下：

---

```
PARFILE=<path>
```

---

<path>：PARFILE 文件的绝对路径。

例如，在 dimp 中使用 PARFILE，设置路径 PARFILE=/mnt/data/dimp/para.txt。

```
./dimp USERID=SYSDBA/SYSDBA PARFILE=/mnt/data/dimp/para.txt
```

其中，参数文件/mnt/data/dimp/para.txt 的内容如下：

```
FILE=/mnt/data/dexp/db_str.dmp  
LOG=db_str.log  
TABLES=table1  
DIRECTORY=/mnt/data/dimp
```

### 3.3.2.15 FEEDBACK

FEEDBACK 用来指明在导入数据的过程中每间隔多少行打印一次进度信息。可选参数。

当参数缺省时，默认值为 0，只打印导入表的总行数。

语法如下：

---

FEEDBACK=<num>

---

<num>: 表间隔的行数。正整数。

例如, 将文件 dexp.dmp 中的对象和数据导入到数据库时, 要求在导出的过程中每隔 100 行打印一次进度信息。

```
./dimp SYSDBA/SYSDBA FULL=Y FILE=/mnt/data/dexp/dexp.dmp LOG=db_str.log
DIRECTORY=/mnt/data/dimp FEEDBACK=100
```

### 3.3.2.16 COMPILE

如果导出时包含过程/函数、视图、物化视图、触发器、类和包, 那么导入时会执行一个编译语句 (alter ...compile)。COMPILE 用于设置是否在导入时, 对过程或函数执行编译。

编译相当于执行重建操作, 用于检查过程或函数是否还有效。编译成功, 说明对象是有效的; 编译失败, 说明对象不存在或发生改变, 导致重建操作失败。编译失败, 导入停止。

不编译就不检查有效性, 直接导入。

语法如下:

---

COMPILE=N/Y

---

Y 或缺省参数: 执行。

N: 不执行。

例如, 导入的过程中含有函数 FUN\_01, 那么导出过程中会自动执行如下语句:

```
alter function "SYSDBA"."FUN_01" compile;
```

如果不想编译, 需要设置 COMPILE=N。

### 3.3.2.17 INDEXFILE

用于将表的索引/约束创建信息写入指定的文件。

语法如下:

---

INDEXFILE=<文件>

---

例如, 设置 INDEXFILE=f:\data\dimp.log。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp LOG=db_str.log FULL=Y
INDEXFILE=/mnt/data/dimp/dimp.log
```

### 3.3.2.18 INDEXFIRST

INDEXFIRST 用于设置导入时是否先建索引。可选参数。

创建索引的过程是先遍历数据再创建索引树。数据量比较大时，应该选择 INDEXFIRST=Y，先创建索引再导入数据，这样就避免创建索引之前遍历大量数据而耗费时间；数据量小时，可任意选择。

语法如下：

---

INDEXFIRST=Y/N

---

Y：是。

N 或参数缺省：否。

### 3.3.2.19 REMAP\_SCHEMA

将源模式中的数据导入到目标模式中。可选参数。

语法如下：

---

<SOURCE\_SCHEMA>: <TARGET\_SCHEMA>

---

<SOURCE\_SCHEMA>：源模式。如果指定的源模式不存在，则导入到对象原来所在的模式。

<TARGET\_SCHEMA>：目标模式。如果目标模式不存在，先创建目标模式，再继续导入。

例如，将 SYSDBA 模式中的数据导入到 PRERSON 模式中。

```
./dimp SYSDBA/SYSDBA FILE=/mnt/data/dexp/imp_exp.dmp DIRECTORY=/mnt/data/dimp
FULL=Y REMAP_SCHEMA=SYSDBA:PERSON
```

### 3.3.2.20 ENCRYPT\_PASSWORD、ENCRYPT\_NAME

如果数据文件是通过加密后得到的，在数据文件导入到数据库时，需要提供加密密钥和加密算法。同时，在 1 个小时内只能尝试 10 次导入解密。

语法如下：

---

ENCRYPT\_PASSWORD=<加密密钥>

ENCRYPT\_NAME=<加密算法>

---

<加密密钥>：与导出文件的加密密钥相同。

<加密算法>：与导出文件的加密算法相同。导出加密算法缺省的情况下，此处加密密码可不写。

例如，将 2.3.3.15 中导出的数据再导入进来。

```
./dimp SYSDBA/SYSDBA FILE=/mnt/data/dexp/dk15.dmp LOG=db_str19.log
FULL=Y DIRECTORY=/mnt/data/dimp ENCRYPT_PASSWORD=abcdefg ENCRYPT_NAME=DES_CBC
```

### 3.3.2.21 SHOW/DESCRIBE

SHOW/DESCRIBE 用于设置是否打印 dexp 导出的数据文件的内容列表。可选参数。

语法如下：

---

SHOW=Y/N 或 DESCRIBE=Y/N

---

Y：是。

N 或缺省参数：否。

例如，想查看 /mnt/data/dexp/ 中 db\_str.dmp 文件的内容列表。

```
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp SHOW=Y
```

打印结果为：

```
-----导出文件列表信息-----
共包含 1 个文件，文件如下：
./dimp USERID=SYSDBA/SYSDBA FILE=/mnt/data/dexp/db_str.dmp SHOW=Y
-----
- 导出文件为 FULL 方式导出，共包含以下 7 个模式
模式 PERSON 含有 4 个表，分别为：
ADDRESS ADDRESS_TYPE PERSON PERSON_TYPE
模式 PRODUCTION 含有 7 个表，分别为：
PRODUCT_CATEGORY PRODUCT_SUBCATEGORY PRODUCT LOCATION PRODUCT_INVENTORY PRODUCT_
REVIEW PRODUCT_VENDOR
模式 PURCHASING 含有 5 个表，分别为：
VENDOR VENDOR_ADDRESS VENDOR_PERSON PURCHASEORDER_HEADER PURCHASEORDER_DETAIL
模式 RESOURCES 含有 4 个表，分别为：
DEPARTMENT EMPLOYEE EMPLOYEE_ADDRESS EMPLOYEE_DEPARTMENT
模式 SALES 含有 5 个表，分别为：
CUSTOMER CUSTOMER_ADDRESS SALESPERSON SALESORDER_HEADER SALESORDER_DETAIL
模式 SYSDBA 含有 1 个表，分别为：
TAB1
模式 OTHER 含有 10 个表，分别为：
DEPARTMENT EMPLOYEESALARY ACCOUNT ACTIONS READER READERAUDIT DEPTTAB EMPTAB SALGRADE
COMPANYHOLIDAYS
```



### 3.3.2.22 LOCAL

LOCAL 只用于在 MPP 环境下，是否使用 LOCAL 方式登录，缺省为不使用。可选参数。

语法如下：

---

LOCAL=Y/N

---

Y：表示用户在登录时为 LOCAL 本地登录方式，即只登陆指定的节点，操作数据也只是本节点的数据。

N 或缺省参数：表示不使用，为 GLOBAL 全局登录方式，即操作数据是所有节点上相关数据。

### 3.3.2.23 TASK\_THREAD\_NUMBER

TASK\_THREAD\_NUMBER 用于设置 dmfldr 处理用户数据的线程数目。可选参数，缺省值为 CPU 个数。但当 CPU 个数大于 8 时，默认值都被置为 8。

语法如下：

---

TASK\_THREAD\_NUMBER=<num>

---

<num>：线程数。取值范围：1-128。

### 3.3.2.24 BUFFER\_NODE\_SIZE

BUFFER\_NODE\_SIZE 用于设置 dmfldr 指定读取文件缓冲区页大小。可选参数，默认为 10。

值越大，缓冲区的页数越多，每次可以读取的数据就越多，每次发送到服务器的数据也就越多，效率越高。但其大小受dmfldr客户端内存大小限制。

语法如下：

---

BUFFER\_NODE\_SIZE=<size\_num>

---

<size\_num>：缓冲区页数。整数类型，取值范围：1-2048。单位为 M。

### 3.3.2.25 TASK\_SEND\_NODE\_NUMBER

TASK\_SEND\_NODE\_NUMBER 用于设置 dmfldr 发送节点个数。可选参数。

语法如下：

---

TASK\_SEND\_NODE\_NUMBER=<num>

---

<num>：节点个数。整数类型，取值范围：16-65535。在系统内存够的情况下可以适当

的设大数值以提升效率。缺省情况下，由程序自动计算。

#### 3.3.2.26 LOB\_NOT\_FAST\_LOAD

LOB\_NOT\_FAST\_LOAD 用于设置当表中含有大字段时不使用 dmfldr，而使用普通的 INSERT 进行插入，根据 COMMIT\_ROWS 参数决定提交的行数。因为当表中有大字段的时候，dmfldr 是一行一行提交表数据。

语法如下：

---

```
LOB_NOT_FAST_LOAD=Y
```

---

#### 3.3.2.27 PRIMARY\_CONFLICT

PPRIMARY\_CONFLICT 用于设置主键冲突的处理方式。可选参数，不设置此参数时主键冲突报错，设置时不报错，可指定对主键冲突的数据进行不同的处理。

语法如下：

---

```
PRIMARY_CONFLICT=[IGNORE|OVERWRITE]
```

---

IGNORE：不覆盖主键冲突行数据。

OVERWRITE：覆盖主键冲突行数据。

#### 3.3.2.28 TABLE\_FIRST

TABLE\_FIRST 用于控制是否强制先导入表对象。可选参数，默认（N）。

语法如下：

---

```
TABLE_FIRST=[Y|N]
```

---

N：正常导入。顺序为：1.没有依赖的对象（除了表之外的数据库对象）；2.有依赖的 class；3.表；4.依赖对象。

Y：强制先导入表。顺序为：1.表；2.没有依赖的对象（除了表之外的数据库对象）；3.有依赖的 class；4.依赖对象。

#### 3.3.2.29 HELP

输入 dimp HELP 即可查看帮助信息。

语法如下：

---

HELP

---

例如，使用 HELP 查看帮助信息。

```
./dimp HELP
```

HELP 信息会逐个显示所有参数的大致信息，供用户快速参考。

## 4 dexp 和 dimp 应用实例

### 4.1 一个完整示例

下面列出一个完整的导出/导入示例以供参考。

#### 1. 环境准备

导出库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。导出的是 DM 数据库系统安装时自带的名为 BOOKSHOP 的示例库，端口号 5236。

导入库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。准备一个空数据库作为导入库，端口号为 8888。

#### 2. dexp 导出

导出数据库的所有对象 (FULL=Y)，导出文件为 dexp01.dmp，导出日志为 dexp01.log，导出文件和日志文件都存放在 /emc\_2/data/dexp 目录中。

```
./dexp SYSDBA/SYSDBA@192.168.0.248:5236 FILE=dexp01.dmp LOG=dexp01.log  
DIRECTORY=/emc_2/data/dexp FULL=Y
```

#### 3. dimp 导入

导入 SYSDBA、OTHER、PERSON 模式中的数据 (SCHEMAS=SYSDBA,OTHER,PERSON)，导入文件就是上一步导出的文件 dexp01.dmp，导入日志 dimp02.log 放入 /emc\_2/data/dimp 目录中。

```
./dimp SYSDBA/SYSDBA@192.168.0.248:8888 FILE=/emc_2/data/dexp/dexp01.dmp  
LOG=dimp02.log DIRECTORY=/emc_2/data/dimp SCHEMAS=SYSDBA,OTHER,PERSON
```

#### 4. 查看 LOG 日志

如果了解详细的导出、导入内容统计信息，请查看 LOG 日志 dexp01.log 和 dimp02.log。

### 4.2 使用关联参数示例

使用加密相关参数 ENCRYPT、ENCRYPT\_PASSWORD、ENCRYPT\_NAME 进行导出导入。

#### 1. 环境准备

导出库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。导

出库是 DM 数据库系统安装时自带的名为 BOOKSHOP 的示例库，端口号 5236。

导入库：环境为 linux，服务器为 192.168.0.248，用户名和密码均为 SYSDBA。准备一个空数据库作为导入库，端口号为 8889。

## 2. dexp 导出

导出数据库的所有对象(FULL=Y)，导出文件为 dexp03.dmp，导出日志为 dexp03.log，导出文件和日志文件都存放在/emc\_2/data/dexp 目录中。

使用加密参数 ENCRYPT、ENCRYPT\_PASSWORD、ENCRYPT\_NAME，对到出库进行加密。

```
./dexp      SYSDBA/SYSDBA@192.168.0.248:5236      FILE=dexp03.dmp      LOG=dexp03.log
DIRECTORY=/emc_2/data/dexp      FULL=Y ENCRYPT=Y      ENCRYPT_PASSWORD=damengren
ENCRYPT_NAME=DES_CBC
```

## 3. dimp 导入

导入整个数据库 (FULL=Y)，导入文件就是上一步导出的文件 dexp03.dmp，导入日志 dexp04.log 放入/emc\_2/data/dimp 目录中。

在导入的过程中，要提供 ENCRYPT\_PASSWORD、ENCRYPT\_NAME 两个参数，这两个参数值必须和导出时一样。

```
./dimp      SYSDBA/SYSDBA@192.168.0.248:8889      FILE=/emc_2/data/dexp/dexp03.dmp
LOG=dimp04.log      DIRECTORY=/emc_2/data/dimp      FULL=Y      ENCRYPT_PASSWORD=damengren
ENCRYPT_NAME=DES_CBC
```

## 4. 查看 LOG 日志

如果了解详细的导出、导入内容统计信息，请查看 LOG 日志 dexp03.log 和 dimp04.log。

咨询热线：400-991-6599

技术支持：dmtech@dameng.com

官网网址：www.dameng.com



**武汉达梦数据库有限公司**  
**Wuhan Dameng Database Co.,Ltd.**

地址：武汉市东湖新技术开发区高新大道999号未来科技大厦C3栋16—19层

16th-19th Floor, Future Tech Building C3, No.999 Gaoxin Road, Donghu New Tech Development Zone,Wuhan,Hubei Province,China

电话：(+86) 027-87588000 传真：(+86) 027-87588810

---