首先，我们导入导出数据，肯定是要通过**[Oracle](http://lib.csdn.net/base/oracle" \t "_blank" \o "Oracle知识库)**自带的可运行程序来完成数据的导入导出工作，imp.exe 和exp.exe这两个可运行文件都放在**[oracle](http://lib.csdn.net/base/oracle" \t "_blank" \o "Oracle知识库)**安装目录下的BIN目录下。

## 一、以cmd命令操作符导入导出

先以cmd命令操作符来演示，比如我的BIN目录是在：D:\app\Administrator\product\11.1.0\db\_1\BIN  这里面。

**exp**导出数据以dmp格式的文件保存，命令符如下：

   D:\app\Administrator\product\11.1.0\db\_1\BIN>exp scott/tiger@orcl file=D:\database.dmp;

   exp --> 表示 执行BIN目录下的exp.exe可运行程序

   scott/tiger  --> 分别表示导出数据的用户名和密码

   orcl --> 表示 oracle[**数据库**](http://lib.csdn.net/base/mysql)的实例名称

   file  --> 用于去指定我们导出的数据的存放位置。

执行完成之后就可以完成数据的导出需求。

**imp**导入数据到oracle数据库中，命令符如下：

   D:\app\Administrator\product\11.1.0\db\_1\BIN>imp scott/tiger@orcl file=D:\database.dmp;

语法和exp差不多，只不过最前面关键字不一样而已。。

但是在实际中会出现一些错误，比如

 (1) 只有 DBA 才能导入由其他 DBA 导出的文件



这是由于指定的登录用户没有导入导出数据的权限，

解决方案：grant dba to scott;

 (2) 必须指定 FULL=Y 或提供 PROMUSER/TOUSER 或 TABLES 参数



这是由于oracle不知道你到底要导入多少数据，比如：全部还是具体到某几张表，所以我们需要明确指定导入的数据表

解决方案： 在file 前面加上 full=y  全库导出，或者指定表名tables=(tableName,tableName1) 即可，比如：

    D:\app\Administrator\product\11.1.0\db\_1\BIN>imp scott/tiger@orcl full=y file=D:\database.dmp;

或者

    D:\app\Administrator\product\11.1.0\db\_1\BIN>imp scott/tiger@orcl file=D:\database.dmp tables=(tb1,tb2);

(3)在导入数据的过程中可能会遇到 数据表已经存在的现象，可以同ignore=y  忽略掉，比如：

    D:\app\Administrator\product\11.1.0\db\_1\BIN>imp scott/tiger@orcl  ignore=y  file=D:\database.dmp;

但是如果表结构发生变化，添加了唯一索引等等可能出现的问题，这一方法就不是很好了，可以考虑删除该用户之后，重建该用户信息，这样子就可以完整的导入数据了。

## 二、用PL/SQL Develpoer工具完成导入导出

如果觉得导入导出数据库 每次都要输入那么长一段命令不太好，可以用PL/SQL Develpoer自带的Tools实现，其原理一样，也是执行imp.exe 和exp.exe ，但是PL/SQL Develpoer已经帮我们封装好了，要用直接指定数据库地址就好了。

具体实现方式如下：

用账号在PL/SQL Develpoer中连接oracle数据库，然后 选择Tools --> Import tables ，看到显示了三个选项卡，分别是Oracle Import ，Sql Insert，PL/SQL Developer。

同样的，在Tools --> Export tables，也有和上面选项卡对应的导出功能，它们分别是Oracle Export ，Sql Insert，PL/SQL Developer。

需要说明的是：

   第一种导入导出为.dmp的文件格式，.dmp文件是二进制的，可以跨平台，还能包含权限，效率也很不错，用得最广 。  
   第二种导入导出为.sql文件的，可用文本编辑器查看，通用性比较好，但效率不如第一种，适合小数据量导入导出。尤其注意的是表中不能有大字段（blob,clob,long），如果 有，会提示不能导出(提示如下：table contains one or more LONG columns cannot export in sql format,user Pl/sql developer format instead)，可以用第一种和第三种方式导出。  
   第三种导入导出为.pde格式的，.pde为PL/SQL Develpoer自有的文件格式，只能用PL/SQL Develpoer自己导入导出；不能用编辑器查看。

### 1、Oracle Import/Export

为了层次清晰一点，先看Oracle Import 和Oracle Export的导入导出功能。

打开Oracle Import导入窗口之后，效果图如下所示：

左边的多选框是控制我们导入数据时候 要导入的东东，比如数据表，索引，序列等等等之类的，勾上了才会去导入

右边的Buffer size缓存大小，默认是30kb

From user 表示来自于哪个用户导出的数据库

To user 表示将导入的数据库指定给哪一个用户，这个必须指定

Import Exexutable 表示的是我们导入数据库指定的导入工具，也就是指定Imp.exe 的位置， 我的是：d:\app\Administrator\product\11.1.0\db\_1\bin\imp.exe

然后下面的Import file 指定的是我们导入的数据库的路径

点击Import 按钮，会弹出cmd命令窗口显示我们导入的数据表明细，

效果图如下：

同样的，在导出的时候，在Oracle Export导出窗口中需要指定Output file 即导出文件的物理路径，如下图所示：

需要注意的是，上面列出的表全是当前用户可操作的表，如果要多选的话按住Ctrl + 左键就好了。

选择后，点击Export 会看到导出的效果图，如下：

并且PL/SQL Develpoer非常友好，如果发现已经有导出的文件，会有提示是否覆盖，效果图如下：

### 2、SQL Inserts

接下来我们来看下SQL Inserts 的导入导出的实现方式。

在它的导出界面 几乎和Oracle Export 界面一样，效果图如下：

基本上上面的多选框 使用默认的就好，我们要选择就是Output file 的导出物理路径就好，并且上面已经说了 只能导出.sql文件格式的。

选择完之后，点击Export ，到已经导出的sql文件中看到，上面是一些创表创约束，触发器等等的命令，下面千篇一律的是Insert into 的插入语句。

如下图：

所以这方式导出的数据库 导入的时候也很方便的，执行这里面所有的sql语句就可以了。

我们打开SQL Inserts 导入的界面看看，如下图：

界面很简单，直接指定sql文件的路径就好了，然后点击Import ，效果图如下：

### 3、PL/SQL Developer

最后我们看看来PL/SQL Develpoer 自带的导入导出功能。

在我们的PL/SQL Develpoer 导出选项卡中，可以看到几乎和上面的实现方式没有什么太大的差别，唯一不同的是必须指定文件格式为.pde格式的。界面效果如下图：

点击Export，界面上没什么变化，可以根据下方的Exporting Done 知道 导出成功了。

然后在我们的PL/SQL Develpoer 导入选项卡中，可以看到如下操作界面：

细心的话会发现，选择pde格式的文件后，右边空白的地方会对应的加载该文件中存在的数据表，并且显示出来。

点击Import ，后需要等一会，才会导入成功，我发现用PL/SQL Develpoer 导入导出的时候效率最慢。