

DataPump 実行環境の設定

ディレクトリ・オブジェクトの作成

ディレクトリ・オブジェクト一覧の表示

DataPump (データ・ダンプ・ユーティリティ) の実行方法

DataPump Export では、\$ORACLE_HOME/bin の expdp が実行される

DataPump Import では、\$ORACLE_HOME/bin の impdp が実行される

注意事項

1. 操作実行は、サーバーのコンソールキーボードから行う
2. remap_xxx オプションは、Oracle11.1 では使用が確認できなかった
(同一対象へのインポート操作のみしかできなかった)
3. エクスポートしたものの全部を原則インポートするので、エクスポート時に余分なものを対象にしないようにする必要がある
4. テーブルの定義において、VIRTUAL HIDDEN オプション定義が含まれているテーブルは、インポート時のテーブル Create 実行時にエラーが発生するので、使用できない

DataPump 実行環境の設定

(1) ディレクトリ・オブジェクトの作成

バックアップファイル出力先のために、ディレクトリ・オブジェクトを作成する

```
sql> create or replace directory <ディレクトリ・オブジェクト名> as  
        '<ディレクトリ・パス>';
```

ディレクトリ・オブジェクト一覧の表示

```
sql> select directory_name, directory_path from dba_directories ;
```

(2) 実行ユーザーに、ディレクトリ・オブジェクトへの読み込み・書き込み権限を付与する

```
sql> grant read, write on directory <ディレクトリ・オブジェクト名>  
        to <ユーザー名> ;
```

(3) 実行ユーザーに、DataPump の実行に必要な権限／ロールを与える

```
sql> grant connect, resource to <ユーザー名>;
```

```
sql> grant exp_full_database, imp_full_database to <ユーザー名>;
```

DataPunp の起動方法

サーバーのコンソールキーボードのコマンド・プロンプト画面から実行すること

```
$ expdp <ユーザー名>/<パスワード> 【ディレクトリ】 【対象】  
        【ダンプファイル】 【ログ・ファイル】
```

```
$ impdp <ユーザー名>/<パスワード> 【ディレクトリ】 【対象】  
        【ダンプファイル】 【ログ・ファイル】
```

指定項目

項目名	説明	指定例
【ディレクトリ】	出力先／入力先のディレクトリ・オブジェクト名の指定	directory=dump_dir
【対象】	入出力対象の指定 ・表 ・スキーマ ・表領域 ・データベース全体	tables=scott.emp schemas=scott tablespace=users full=y
【ダンプファイル】	ダンプファイル名の指定	dumpfile=exp_file.dmp
	ファイル名にパス指定を含めることは出来ない	
【ログ・ファイル】	ログ出力先のディレクトリ・オブジェクト名とログ・ファイル名	logfile=dplog_dir : emp_file.log
	ディレクトリ・オブジェクト以外のディレクトリを明示してログを出力することは出来ない	

インポート時に、【対象】をエクスポート時と異なる指定で行うことが出来る
たとえば、エクスポート時に表領域を対象にし、インポートはその中の特定の
テーブルだけを指定するようなことが可能

【対象】の複数個指定方法

カンマ (,) で並べる

```
schemas=KOZUE, IKURA, MINKA
```

ダンプファイルのサイズを事前に調査する方法 (データのダンプは作成されない)

estimate_only=y オプションを指定する

```
expdp dpuser/** schema=SCOTT estimate_only=y
```

インポート時に `remap_xxx` オプションを指定することにより、エクスポート時とは異なったテーブル名やエクスポート先表領域のインポートが可能 (Oracle11.1 では、使用できなかった)

remap_xxx オプション

操作内容	指定例
テーブル名を変更する	<code>remap_table=<スキーマ名>. <旧テーブル名></code> <code>: <スキーマ名>. <新テーブル名></code>
インポート先のスキーマを変更する	<code>remap_schema=<旧スキーマ名></code> <code>: <新スキーマ名></code>
インポート先のテーブルスペース先を変更する	<code>remap_tablespace=<旧テーブルスペース名></code> <code>: <新テーブルスペース名></code>

サンプル

インポート時に、スキーマ先とテーブル名を同時に変更する
かつ、既存に存在していたテーブルを上書きする

```
impdp dpuser/*** directory=DPDIR dumpfile=dp_tb1.dmp
      remap_schema=SCOTT:KOZUE
      remap_table=EMP_TEST:EMP
      table_exists_action=replace
```

パラレル実行するためのオプション

`parallel=4`

ダンプオプションには%U を付ける必要あり

クラスター環境の場合は、`cluster=n` オプションの記述も必要あり

```
expdp dpuser/*** tables=SCOTT.big_table directory=DPDIR
      dumpfile=dp_imp%U.dmp parallel=4
```

※ ダンプファイルは `dp_imp01.dmp`、`dp_imp02.dmp`、`dp_imp03.dmp`、`dp_imp04.dmp` の4個が作成される

```
impdp dpuser/*** directory=DPDIR dumpfile=dp_imp%U.dmp
      parallel=4
```

定義情報だけやデータだけをバックアップするオプション

`content=METADATA_ONLY`

`content=DATA_ONLY`

フィルタリングを使った DataPump

ダンプ処理において、不要なオブジェクトを除外する指定
テーブル内のレコードを条件指定して、ダンプする指定

```
expdp dpuser/** directory=DPDIR dumpfile=dp_imp.dmp  
parfile=/oracle/exppara.par ← パラメータファイルを指定
```

↓
/oracle/exppara.par ファイル

```
EXCLUDE=<オブジェクトタイプ>:<オブジェクト名>
```

指定例)

```
EXCLUDE=TABLE:"LIKE 'TEST_%' "
```

```
EXCLUDE=INDEX:"LIKE 'TEST_%' " ← LIKE を使った条件指定
```

```
QUERY= '<スキーマ名>.<表名>:"データ抽出用 WHERE 句" '
```

指定例)

```
QUERY= 'KOZUE.EMP:" WHERE EMPNO < 6 " '
```

EXCLUDE を使って対象を除外することについて

エクスポートをスキーマに対して行い、インポートを特定テーブルだけを
対象にしようとする場合、実際の使用では現実的でないと思えた

オブジェクトの種類が多数（表、権限、プロシージャなど）あり、除外オ
ブジェクトタイプを全部記述するのには無理がある