

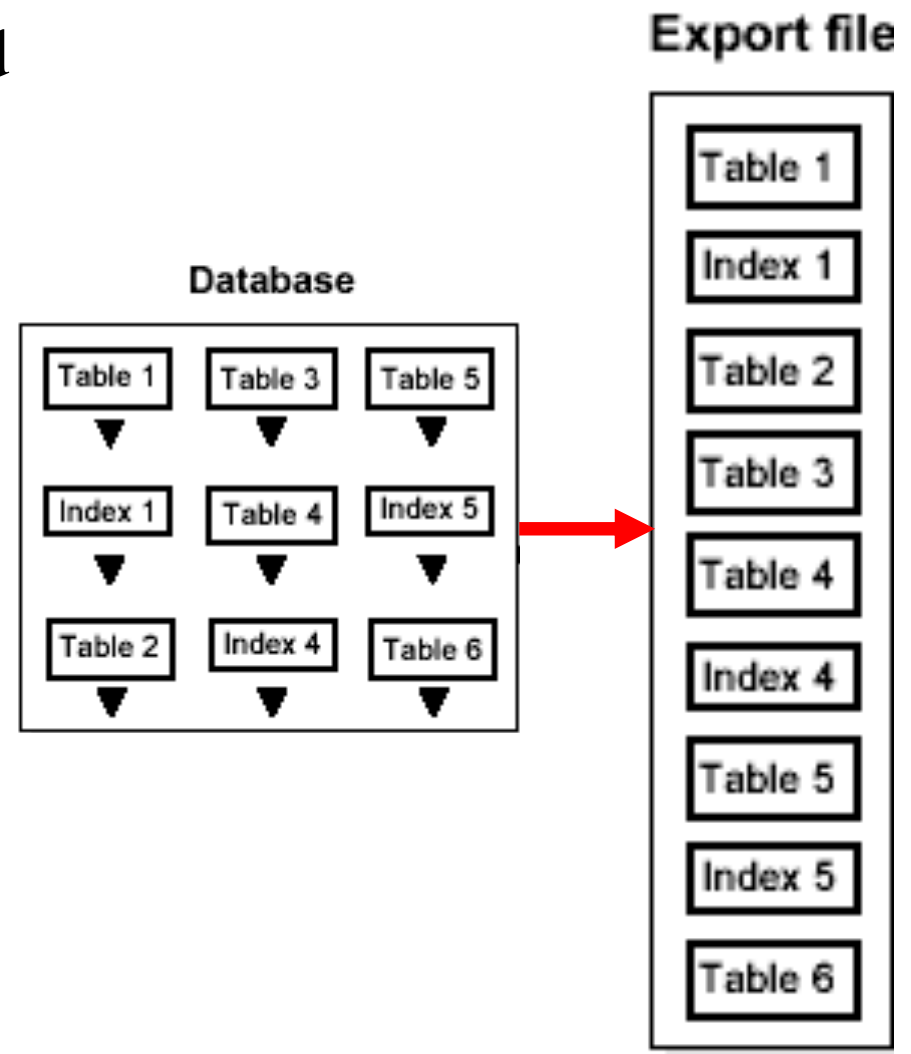
Export Utility

Export Utility

- The Export utility provides a simple way for you to transfer data objects between Oracle databases, even if they reside on platforms with different hardware and software configurations.
- When you run Export against an Oracle database, objects (such as tables) are extracted, followed by their related objects (such as indexes, comments, and grants) and the extracted data is written to an export file.
- An Export file is an Oracle binary-format dump file that is typically located on disk or tape.

Export Utility

- The dump files can be transferred using FTP or physically transported to a different site.
- The files can then be used with the Import utility to transfer data between databases that are on systems not connected through a network.
- The files can also be used as backups in addition to normal backup procedures.
- Export dump files can only be read by the Oracle Import utility. (*The version of the Import utility cannot be earlier than the version of the Export utility used to create the dump file.*)



Invoking Export

You can invoke Export and specify parameters by using any of the following methods:

- Command-line entries
- Interactive Export prompts
- Parameter files

Command-Line Entries

- You can specify all valid parameters and their values from the command line using the following syntax:

```
exp username/password PARAMETER=value
```

or

```
exp username/password  
PARAMETER=(value1,value2,...,valuen)
```

- The number of parameters cannot exceed the maximum length of a command line on the system.

Interactive Export Prompts

- If you prefer to let Export prompt you for the value of each parameter, you can use the following syntax to start Export in interactive mode:

```
exp username/password
```

- Export will display commonly used parameters with a request for you to enter a value.
- This method exists for backward compatibility and is not recommended because it provides less functionality than the other methods.

Parameter Files

- You can specify all valid parameters and their values in a parameter file.
- Storing the parameters in a file allows them to be easily modified or reused. The command-line option
 - `PARFILE= filename` tells Export to read the parameters from the specified file. For example:

```
exp PARFILE=filename
```

```
exp username/password PARFILE=filename
```

- The syntax for parameter file specifications is one of the following:

```
PARAMETER=value
```

```
PARAMETER=(value)
```

```
PARAMETER=(value1, value2, ...)
```

- The following example shows a partial parameter file listing:

```
GRANTS=y
```

```
INDEXES=y
```

- The position of the `PARFILE` parameter and other parameters on the command line determines which parameters take precedence. For example, assume the parameter file `params.dat` contains the parameter `INDEXES=y` and Export is invoked as:

```
exp username/password PARFILE=params.dat INDEXES=n
```

- In this case, because `INDEXES=n` occurs *after* `PARFILE=params.dat`, `INDEXES=n` overrides the value of the `INDEXES` parameter in the parameter file.

Export Modes

The Export utility provides four modes of export:

- Full
- User (Owner)
- Table
- Tablespace

Objects Exported and Imported in Each Mode

	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Analyze cluster	No	Yes	Yes	No
Analyze tables/statistics	Yes	Yes	Yes	Yes
Application contexts	No	No	Yes	No
Auditing information	Yes	Yes	Yes	No
B-tree, bitmap, domain functional indexes	Yes ¹	Yes ¹	Yes	Yes
Cluster definitions	No	Yes	Yes	Yes
Column and table comments	Yes	Yes	Yes	Yes

Objects Exported and Imported in Each Mode

	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Database links	No	Yes	Yes	No
Default roles	No	No	Yes	No
Dimensions	No	Yes	Yes	No
Directory aliases	No	No	Yes	No
External tables (without data)	Yes	Yes	Yes	No
Foreign function libraries	No	Yes	Yes	No
Indexes owned by users other than table owner	Yes (Privileged users only)	Yes	Yes	Yes
Index types	No	Yes	Yes	No
Java resources and classes	No	Yes	Yes	No

Objects Exported and Imported in Each Mode

	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Job queues	No	Yes	Yes	No
Nested table data	Yes	Yes	Yes	Yes
Object grants	Yes (Only for tables and indexes)	Yes	Yes	Yes
Object type definitions used by table	Yes	Yes	Yes	Yes
Object types	No	Yes	Yes	No
Operators	No	Yes	Yes	No
Password history	No	No	Yes	No
Postinstance actions and objects	No	No	Yes	No
Postschema procedural actions and objects	No	Yes	Yes	No
Posttable actions	Yes	Yes	Yes	Yes

Objects Exported and Imported in Each Mode

	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Posttable procedural actions and objects	Yes	Yes	Yes	Yes
Preschema procedural objects and actions	No	Yes	Yes	No
Pretable actions	Yes	Yes	Yes	Yes
Pretable procedural actions	Yes	Yes	Yes	Yes
Private synonyms	No	Yes	Yes	No
Procedural objects	No	Yes	Yes	No
Profiles	No	No	Yes	No
Public synonyms	No	No	Yes	No
Referential integrity constraints	Yes	Yes	Yes	No
Refresh groups	No	Yes	Yes	No
Resource costs	No	No	Yes	No

Objects Exported and Imported in Each Mode

	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Role grants	No	No	Yes	No
Roles	No	No	Yes	No
Rollback segment definitions	No	No	Yes	No
Security policies for table	Yes	Yes	Yes	Yes
Sequence numbers	No	Yes	Yes	No
Snapshot logs	No	Yes	Yes	No
Snapshots and materialized views	No	Yes	Yes	No
System privilege grants	No	No	Yes	No
Table constraints (primary, unique, check)	Yes	Yes	Yes	Yes

Objects Exported and Imported in Each Mode

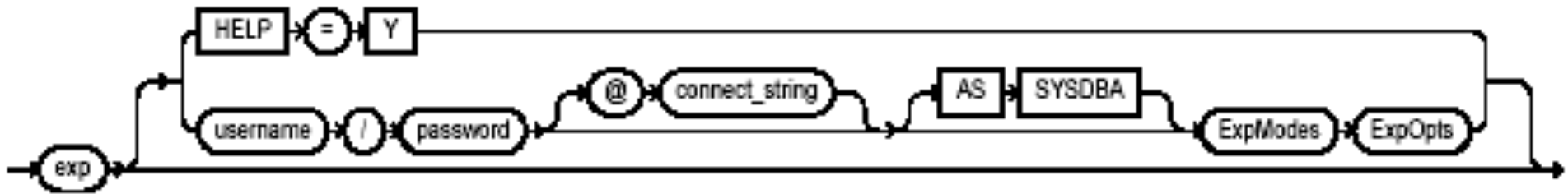
	Table Mode	User Mode	Full Database Mode	Tablespace Mode
Table data	Yes	Yes	Yes	No
Table definitions	Yes	Yes	Yes	Yes
Tablespace definitions	No	No	Yes	No
Tablespace quotas	No	No	Yes	No
Triggers	Yes	Yes ²	Yes	Yes
Triggers owned by other users	Yes (Privileged users only)	No	No	No
User definitions	No	No	Yes	No
User proxies	No	No	Yes	No
User views	No	Yes	Yes	No
User-stored procedures, packages, and functions	No	Yes	Yes	No

Table-Level and Partition-Level Export

- You can export tables, partitions, and subpartitions in the following ways:
 - **Table-level Export:** exports all data from the specified tables
 - **Partition-level Export:** exports only data from the specified source partitions or subpartitions
- In all modes, partitioned data is exported in a format such that partitions or subpartitions can be imported selectively.
- **Table-Level Export**
 - In table-level Export, you can export an entire table (partitioned or nonpartitioned) along with its indexes and other table-dependent objects.
 - If the table is partitioned, all of its partitions and subpartitions are also exported.
 - You can perform a table-level export in **any Export mode**.
- **Partition-Level Export**
 - In partition-level Export, you can export one or more specified partitions or subpartitions of a table.
 - You can only perform a partition-level export in **Table mode**.

Export Parameters

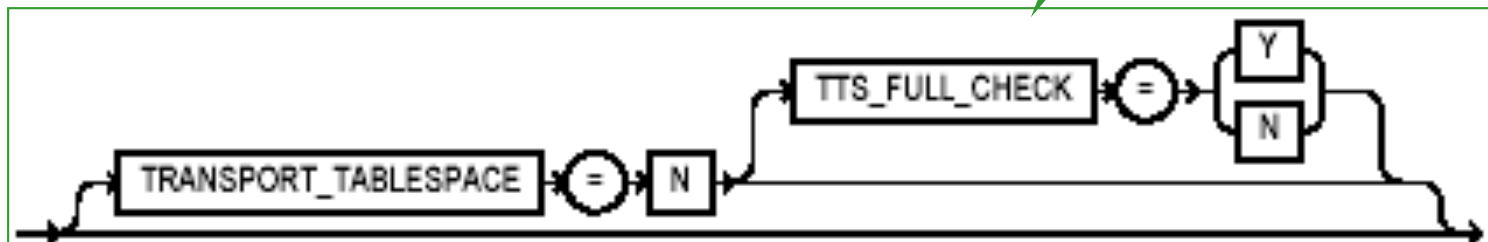
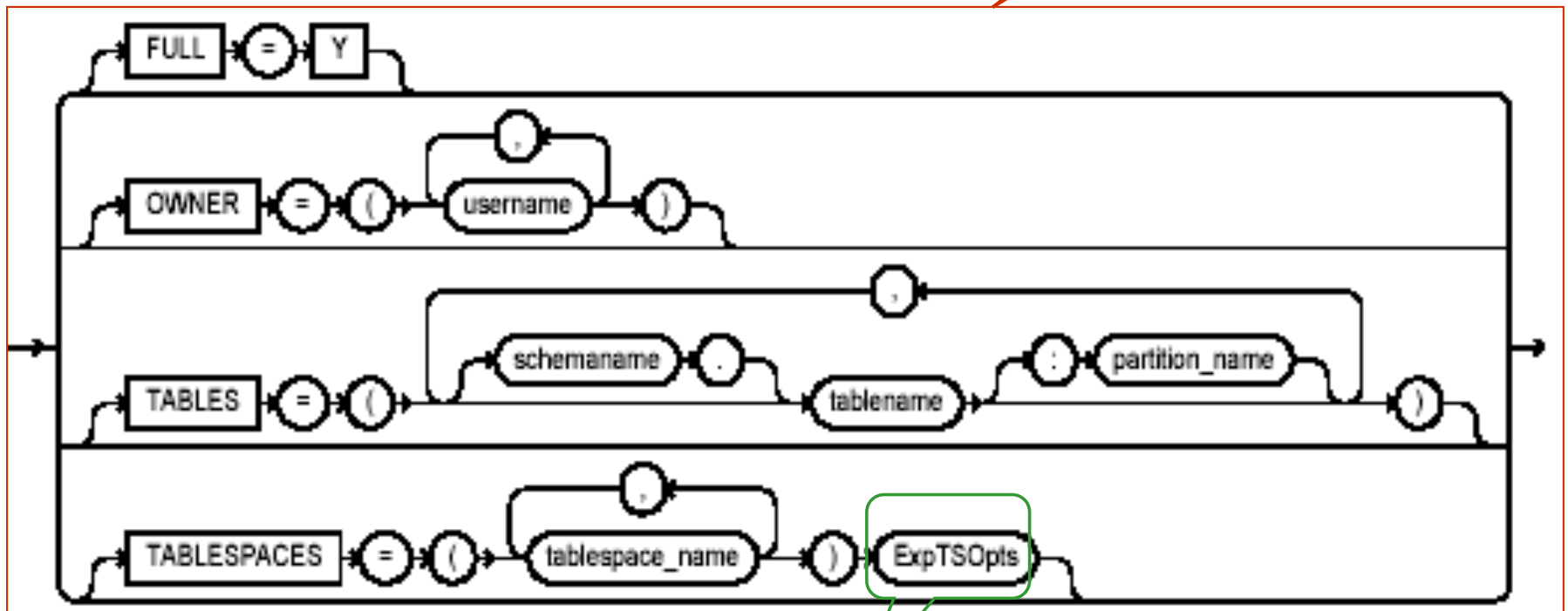
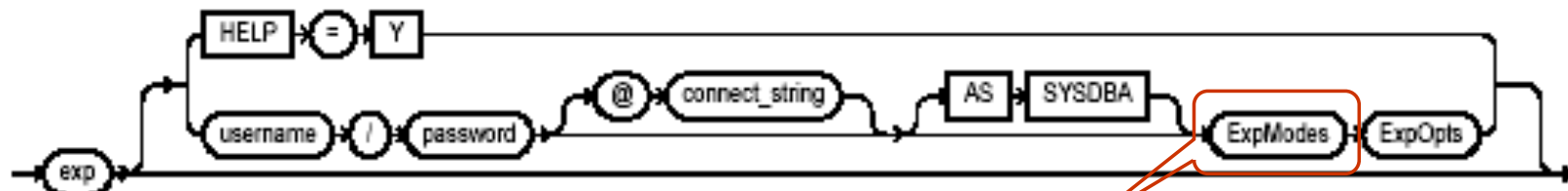
Export Start



HELP (Default: n)

Displays a description of the Export parameters.

Export Parameters



Export Parameters

FULL (Default: n): Indicates that the Export is a full database mode Export (that is, it exports the entire database). Specify FULL=y to export in full database mode.

OWNER (Default: none) : Indicates that the Export is a user-mode Export and lists the users whose objects will be exported.

TABLES (Default: none) : Specifies that the Export is a table-mode Export and lists the table names and partition and subpartition names to export. You can specify the following when you specify the name of the table:

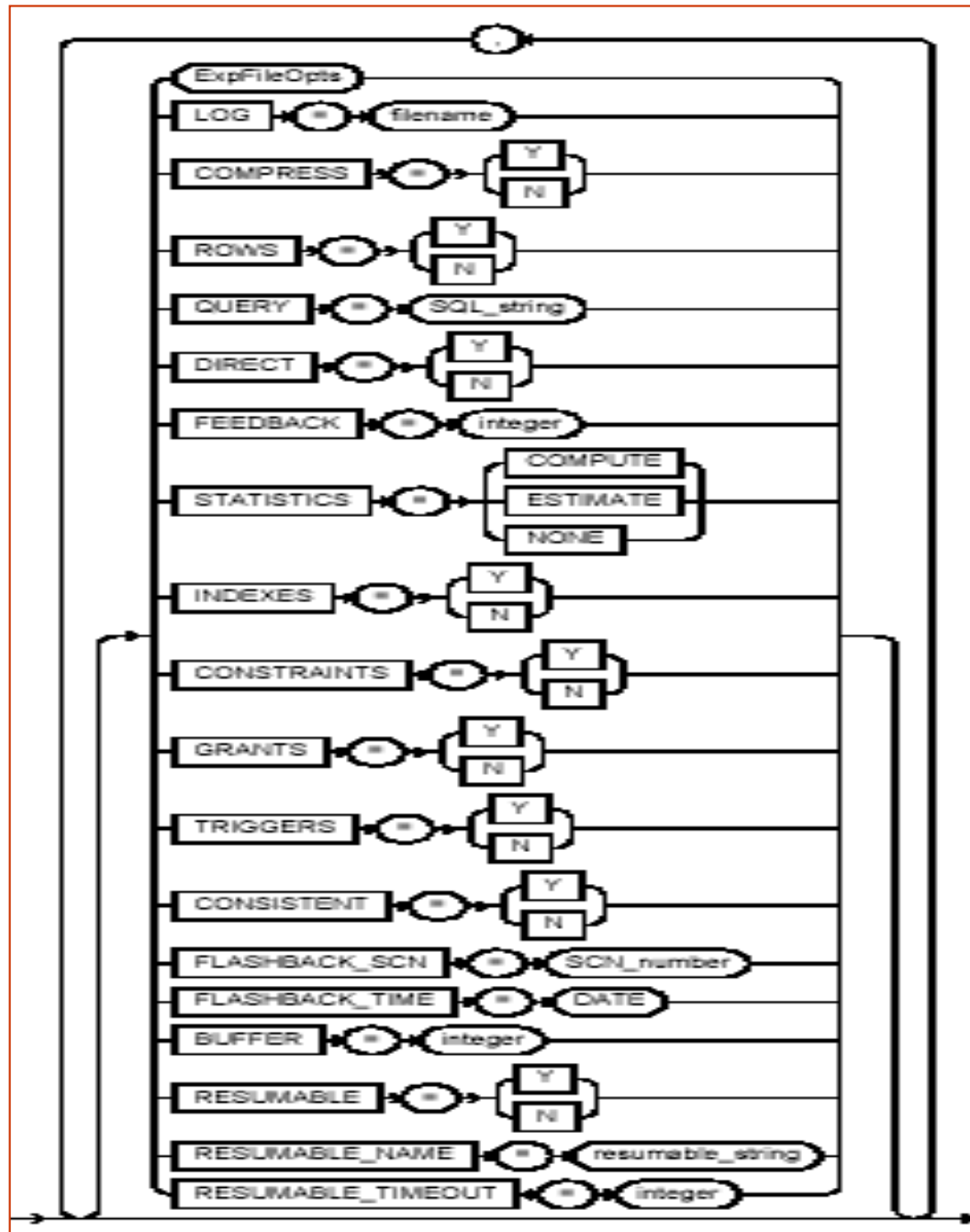
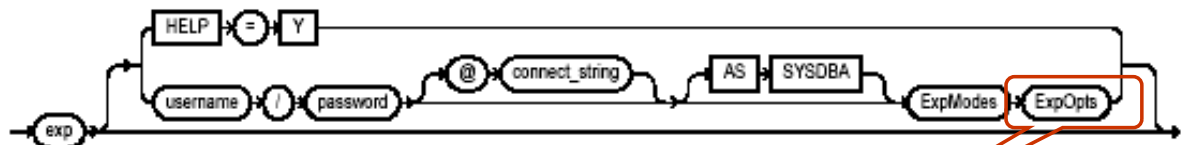
- *schemaname* specifies the name of the user's schema from which to export the table or partition.
- *Tablename* specifies the name of the table or tables to be exported.
- *partition_name* indicates that the export is a partition-level Export. The syntax you use to specify the preceding is in the form:

```
schemaname.tablename:partition_name
```

```
schemaname.tablename:subpartition_name
```

TABLESPACES (Default: none) : The TABLESPACES parameter specifies that all tables in the tablespace be exported to the Export dump file. This includes all tables contained in the list of tablespaces and all tables that have a partition located in the list of tablespaces. Indexes are exported with their tables, regardless of where the index is stored.

Export Parameters



Export Parameters

LOG (Default: none) : Specifies a filename to receive informational and error messages. For example: `exp SYSTEM/password LOG=export.log`

ROWS (Default: y) : Specifies whether or not the rows of table data are exported.

QUERY (Default: none) : This parameter allows you to select a subset of rows from a set of tables when doing a table mode export. The value of the query parameter is a string that contains a WHERE clause for a SQL SELECT statement that will be applied to all tables listed in the TABLE parameter. Example : `exp scott/tiger TABLES=emp QUERY=\"WHERE job=\"SALESMAN\" and sal<1600\"`

DIRECT (Default: n) : Specifies whether you use direct path or conventional path Export.

FEEDBACK (Default: 0 (zero)) : Specifies that Export should display a progress meter in the form of a period for n number of rows exported. For example, if you specify `FEEDBACK=10`, Export displays a period each time 10 rows are exported.

STATISTICS (Default: ESTIMATE) : Specifies the type of database optimizer statistics to generate when the exported data is imported. Options are ESTIMATE, COMPUTE, and NONE. In some cases, Export will place the precalculated statistics in the export file as well as the ANALYZE statements to regenerate the statistics.

INDEXES (Default: y) : Specifies whether or not the Export utility exports indexes.

CONSTRAINTS (Default: y) : Specifies whether or not the table constraints are exported.

Export Parameters

GRANTS (Default: y) : Specifies whether or not the Export utility exports object grants. The object grants that are exported depend on whether you use full database mode or user mode. In full database mode, all grants on a table are exported. In user mode, only those granted by the owner of the table are exported. System privilege grants are always exported.

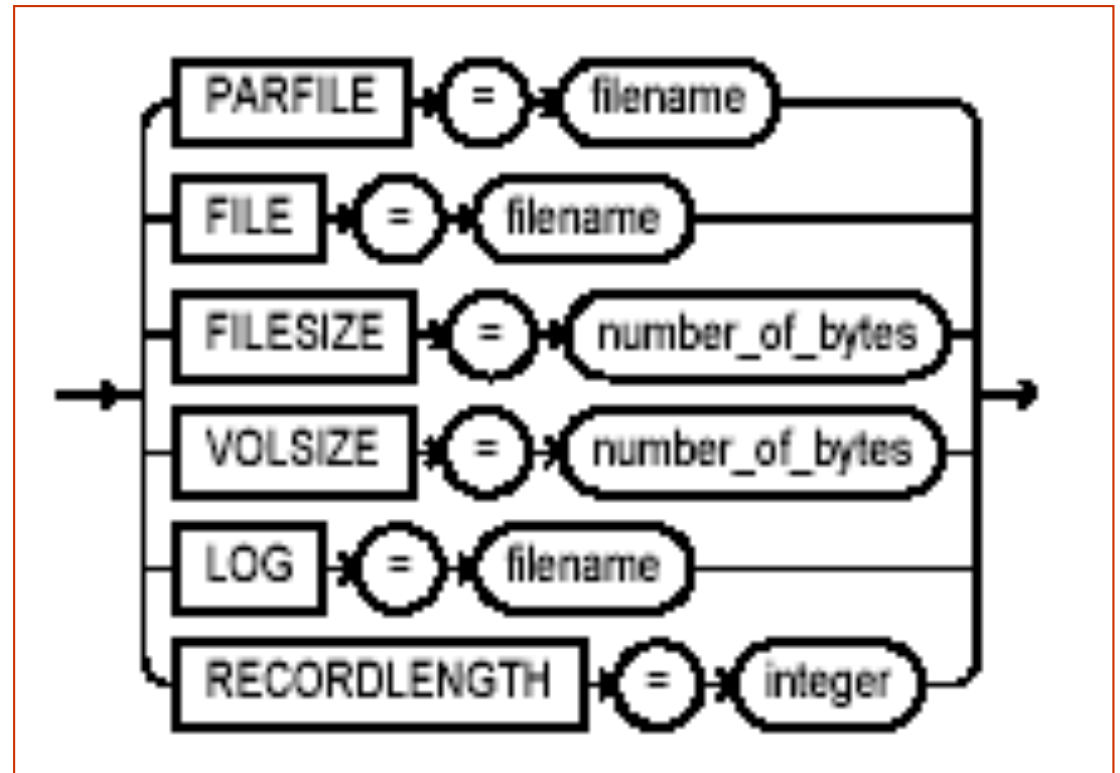
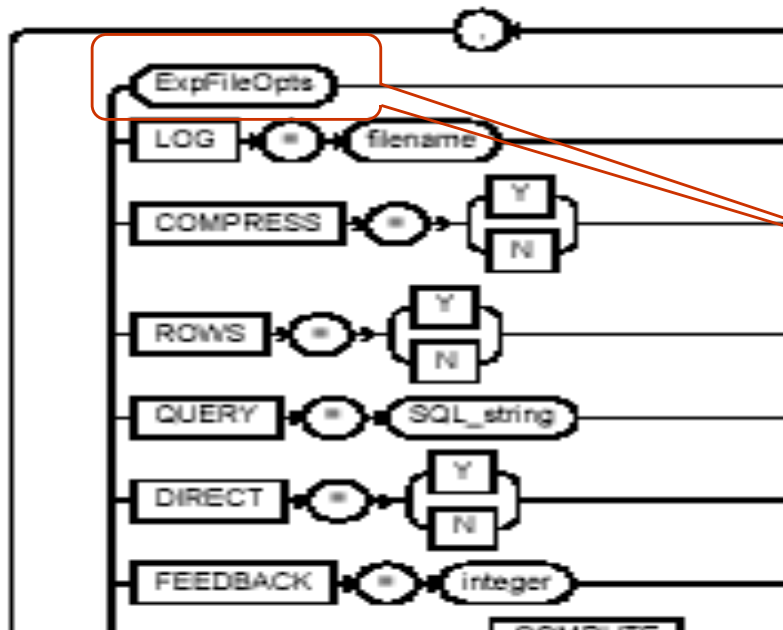
TRIGGERS (Default: y) : Specifies whether or not the Export utility exports triggers.

CONSISTENT (Default: y) : Specifies whether or not Export uses the SET TRANSACTION READ ONLY statement to ensure that the data seen by Export is consistent to a single point in time and does not change during the execution of the exp command. You should specify CONSISTENT=y when you anticipate that other applications will be updating the target data after an export has started.

BUFFER (Default: operating system-dependent.) : Specifies the size, in bytes, of the buffer used to fetch rows. As a result, this parameter determines the maximum number of rows in an array fetched by Export. Use the following formula to calculate the buffer size:

$buffer_size = rows_in_array * maximum_row_size$ If you specify zero, the Export utility fetches only one row at a time.

Export Parameters



Export Parameters

PARFILE (Default: none) : Specifies a filename for a file that contains a list of Export parameters.

FILE (Default: expdat.dmp) : Specifies the names of the export files. Because Export supports multiple export files you can specify multiple filenames to be used. When Export reaches the value you have specified for the maximum FILESIZE, Export stops writing to the current file, opens another export file with the next name specified by the FILE parameter, and continues until complete or the maximum value of FILESIZE is again reached. If you do not specify sufficient export filenames to complete the export, Export will prompt you to provide additional filenames.

FILESIZE : If you specify a value (byte limit) for the FILESIZE parameter, Export will write only the number of bytes you specify to each dump file.

VOLSIZE : Specifies the maximum number of bytes in an export file on each volume of tape. The VOLSIZE parameter has a maximum value equal to the max value that can be stored in 64 bits.

LOG (Default: none) : Specifies a filename to receive informational and error messages.

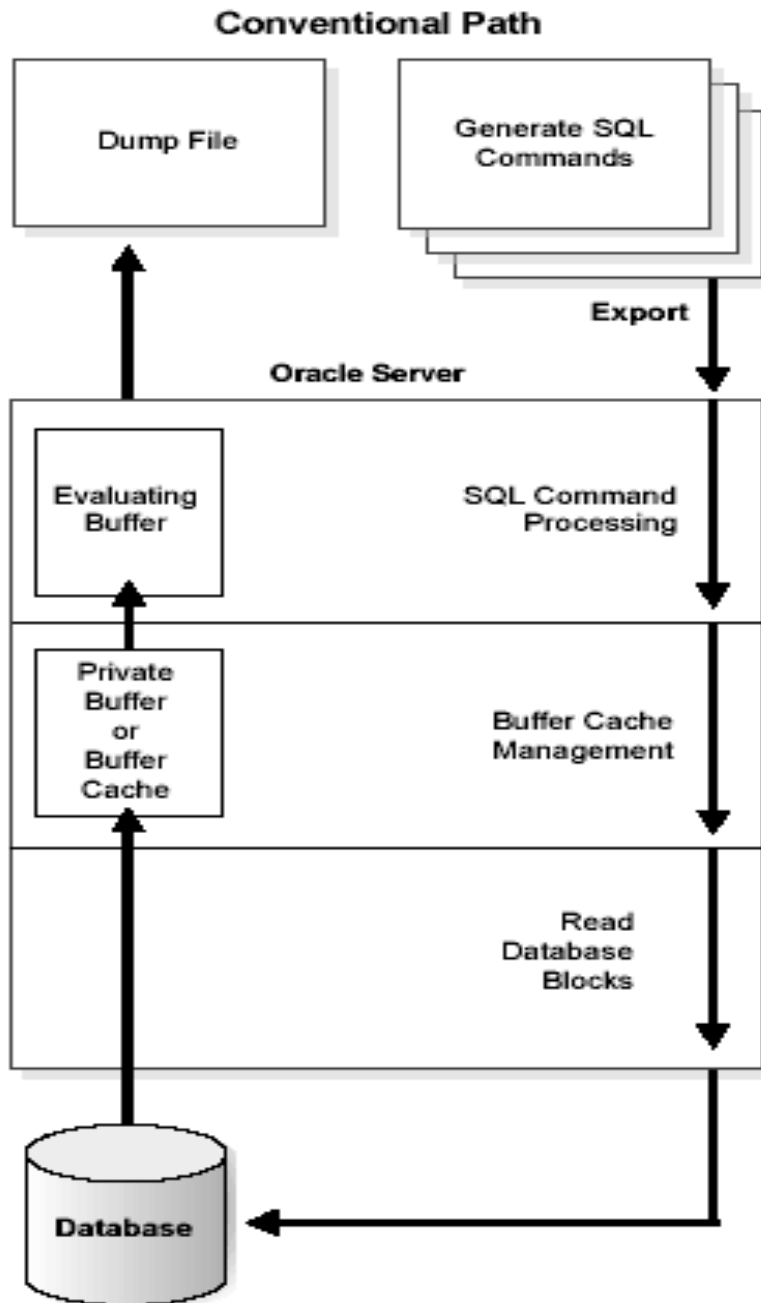
RECORDLENGTH (Default: operating system-dependent) : Specifies the length, in bytes, of the file record. The RECORDLENGTH parameter is necessary when you must transfer the export file to another operating system that uses a different default value.

Conventional Path Export Versus Direct Path Export

Export provides two methods for exporting table data:

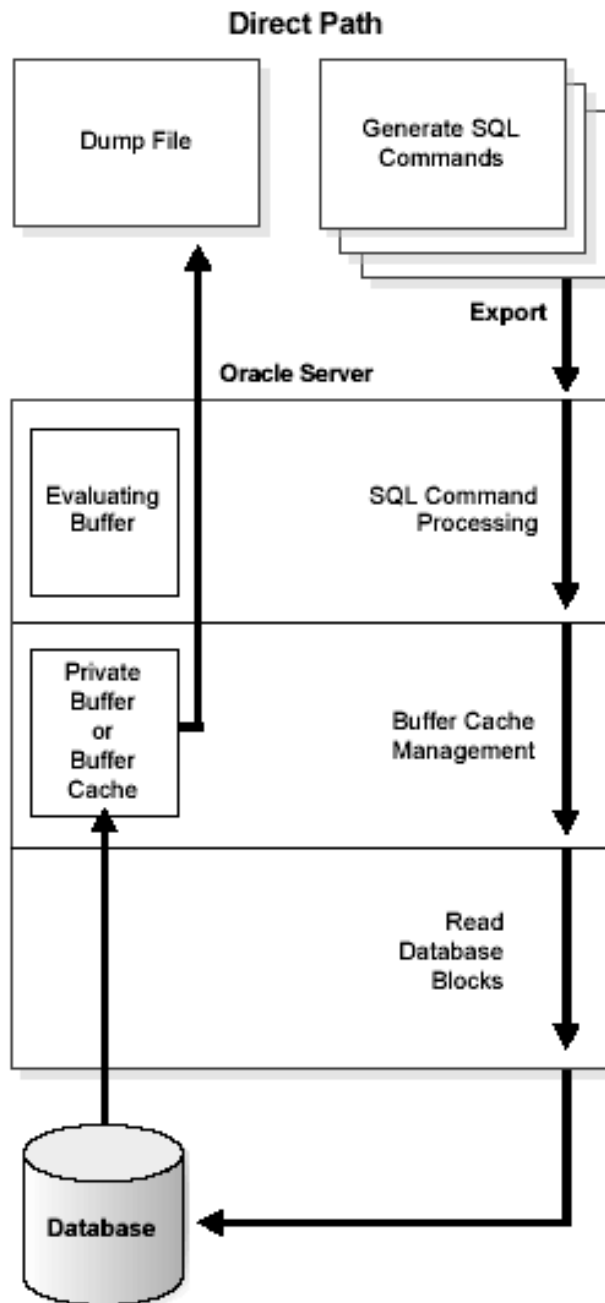
- Conventional path Export
- Direct path Export

Conventional Path Export Versus Direct Path Export



Conventional path Export uses the SQL SELECT statement to extract data from tables. Data is read from disk into a buffer cache, and rows are transferred to the evaluating buffer. The data, after passing expression evaluation, is transferred to the Export client, which then writes the data into the export file.

Conventional Path Export Versus Direct Path Export



Direct path Export is much faster than conventional path Export because data is read from disk into the buffer cache and rows are transferred *directly* to the Export client. The evaluating buffer is bypassed. The data is already in the format that Export expects, thus avoiding unnecessary data conversion. The data is transferred to the Export client, which then writes the data into the export file.