4

# Oracle Data Pump Legacy Mode

With Oracle Data Pump legacy mode, you can use original Export and Import parameters on the Oracle Data Pump Export and Data Pump Import command lines.

## 4.1 Oracle Data Pump Legacy Mode Use Cases

Oracle Data Pump enters legacy mode when it encounters legacy export or import parameters, so that you can continue using existing scripts.

If you previously used original Export (`exp`) and Import (`imp`), then you probably have scripts that you have been using for many years. Oracle Data Pump provides a legacy mode, which allows you to continue to use your existing scripts with Oracle Data Pump.

Oracle Data Pump enters legacy mode when it determines that a parameter unique to original Export or Import is present, either on the command line, or in a script. As Oracle Data Pump processes the parameter, the analogous Oracle Data Pump Export or Oracle Data Pump Import parameter is displayed. Oracle strongly recommends that you view the new syntax, and make script changes as time permits.

> **Note:**
>
> The Oracle Data Pump Export and Import utilities created and read dump files and log files in Oracle Data Pump format only. They never create or read dump files compatible with original Export or Import. If you have a dump file created with original Export, then you must use original Import (`imp`) to import the data into the database. The original Export utility (`exp`) can no longer be used.

## 4.2 Parameter Mappings

You can use original Oracle Export and Import parameters when they map to Oracle Data Pump Export and Import parameters that supply similar functionality.

- Using Original Export Parameters with Oracle Data Pump
  Oracle Data Pump Export accepts original Export parameters when they map to a corresponding Oracle Data Pump parameter.
- Using Original Import Parameters with Oracle Data Pump
  Oracle Data Pump Import accepts original Import parameters when they map to a corresponding Oracle Data Pump parameter.

## 4.2.1 Using Original Export Parameters with Oracle Data Pump

Oracle Data Pump Export accepts original Export parameters when they map to a corresponding Oracle Data Pump parameter.

**Oracle Data Pump Interpretation of Original Export Parameters**

> **Note:**
>
> Original Export was desupported for general use as of Oracle Database 11g.

To see how Oracle Data Pump Export interprets original Export parameters, refer to the table for comparisons. Parameters that have the same name and functionality in both original Export and Oracle Data Pump Export are not included in this table.

**Table 4-1 How Oracle Data Pump Export Handles Original Export Parameters**

| Original Export Parameter | Action Taken by Data Pump Export Parameter |
|---|---|
| `BUFFER` | This parameter is ignored. |
| `COMPRESS` | This parameter is ignored. In original Export, the `COMPRESS` parameter affected how the initial extent was managed. Setting `COMPRESS=n` caused original Export to use current storage parameters for the initial and next extent.<br>The Oracle Data Pump Export `COMPRESSION` parameter is used to specify how data is compressed in the dump file, and is not related to the original Export `COMPRESS` parameter. |
| `CONSISTENT` | Oracle Data Pump Export determines the current time, and uses `FLASHBACK_TIME`. |
| `CONSTRAINTS` | If original Export used `CONSTRAINTS=n`, then Oracle Data Pump Export uses `EXCLUDE=CONSTRAINTS`.<br>The default behavior is to include constraints as part of the export. |
| `DIRECT` | This parameter is ignored. Oracle Data Pump Export automatically chooses the best export method. |

**Table 4-1 (Cont.) How Oracle Data Pump Export Handles Original Export Parameters**

| Original Export Parameter | Action Taken by Data Pump Export Parameter |
|---|---|
| FEEDBACK | The Oracle Data Pump Export STATUS=30 command is used. Note that this is not a direct mapping because the STATUS command returns the status of the export job, as well as the rows being processed.<br>In original Export, feedback was given after a certain number of rows, as specified with the FEEDBACK command. In Oracle Data Pump Export, the status is given every so many seconds, as specified by STATUS. |
| FILE | Oracle Data Pump Export attempts to determine the path that was specified or defaulted to for the FILE parameter, and also to determine whether a directory object exists to which the schema has read and write access. Original Export and Import and Data Pump Export and Import differ on where dump files and log files can be written to and read from, because the original version is client-based, and Oracle Data Pump is server-based. |
| GRANTS | If original Export used GRANTS=n, then Data Pump Export uses EXCLUDE=GRANT.<br>If original Export used GRANTS=y, then the parameter is ignored and does not need to be remapped because that is the Oracle Data Pump Export default behavior. |
| INDEXES | If original Export used INDEXES=n, then Oracle Data Pump Export uses the EXCLUDE=INDEX parameter.<br>If original Export used INDEXES=y, then the parameter is ignored and does not need to be remapped because that is the Oracle Data Pump Export default behavior. |
| LOG | Oracle Data Pump Export attempts to determine the path that was specified or defaulted to for the LOG parameter, and also to determine whether a directory object exists to which the schema has read and write access.<br>Original Export and Import and Data Pump Export and Import differ on where dump files and log files can be written to and read from, because the original version is client-based, and Oracle Data Pump is server-based.<br>The contents of the log file will be those of an Oracle Data Pump Export operation. |
| OBJECT_CONSISTENT | This parameter is ignored, because Oracle Data Pump Export processing ensures that each object is in a consistent state when being exported. |
| OWNER | The Oracle Data Pump SCHEMAS parameter is used. |

**Table 4-1 (Cont.) How Oracle Data Pump Export Handles Original Export Parameters**

| Original Export Parameter | Action Taken by Data Pump Export Parameter |
|---|---|
| RECORDLENGTH | This parameter is ignored, because Oracle Data Pump Export automatically takes care of buffer sizing. |
| RESUMABLE | This parameter is ignored, because Oracle Data Pump Export automatically provides this functionality to users who have been granted the EXP_FULL_DATABASE role. |
| RESUMABLE_NAME | This parameter is ignored, because Oracle Data Pump Export automatically provides this functionality to users who have been granted the EXP_FULL_DATABASE role. |
| RESUMABLE_TIMEOUT | This parameter is ignored, because Oracle Data Pump Export automatically provides this functionality to users who have been granted the EXP_FULL_DATABASE role. |
| ROWS | If original Export used ROWS=y, then Oracle Data Pump Export uses the CONTENT=ALL parameter.<br>If original Export used ROWS=n, then Oracle Data Pump Export uses the CONTENT=METADATA_ONLY parameter. |
| STATISTICS | This parameter is ignored, because statistics are always saved for tables as part of an Oracle Data Pump export operation. |
| TABLESPACES | If original Export also specified TRANSPORT_TABLESPACE=n, then Oracle Data Pump Export ignores the TABLESPACES parameter.<br>If original Export also specified TRANSPORT_TABLESPACE=y, then Oracle Data Pump Export takes the names listed for the TABLESPACES parameter and uses them on the Oracle Data Pump Export TRANSPORT_TABLESPACES parameter. |
| TRANSPORT_TABLESPACE | If original Export used TRANSPORT_TABLESPACE=n (the default), then Oracle Data Pump Export uses the TABLESPACES parameter.<br>If original Export used TRANSPORT_TABLESPACE=y, then Oracle Data Pump Export uses the TRANSPORT_TABLESPACES parameter, and only the metadata is exported. |
| TRIGGERS | If original Export used TRIGGERS=n, then Oracle Data Pump Export uses the EXCLUDE=TRIGGER parameter.<br>If original Export used TRIGGERS=y, then the parameter is ignored. The parameter does not need to be remapped, because that is the Oracle Data Pump Export default behavior. |

Table 4-1 (Cont.) How Oracle Data Pump Export Handles Original Export Parameters

| Original Export Parameter | Action Taken by Data Pump Export Parameter |
|---|---|
| `TTS_FULL_CHECK` | If original Export used `TTS_FULL_CHECK=y`, then Oracle Data Pump Export uses the `TRANSPORT_FULL_CHECK` parameter.<br>If original Export used `TTS_FULL_CHECK=y`, then the parameter is ignored. The parameter does not need to be remapped, because that is the Oracle Data Pump Export default behavior. |
| `VOLSIZE` | When the original Export `VOLSIZE` parameter is used, it means the location specified for the dump file is a tape device. The Oracle Data Pump Export dump file format does not support tape devices. Therefore, this operation terminates with an error. |

## 4.2.2 Using Original Import Parameters with Oracle Data Pump

Oracle Data Pump Import accepts original Import parameters when they map to a corresponding Oracle Data Pump parameter.

To see how Oracle Data Pump Import interprets original Export parameters, refer to the table for comparisons. Parameters that have the same name and functionality in both original Import and Oracle Data Pump Import are not included in this table.

Table 4-2 How Oracle Data Pump Import Handles Original Import Parameters

| Original Import Parameter | Action Taken by Oracle Data Pump Import Parameter |
|---|---|
| `BUFFER` | This parameter is ignored. |
| `CHARSET` | This parameter was desupported several releases ago, and should no longer be used. Attempting to use this desupported parameter causes the Oracle Data Pump Import operation to stop. |
| `COMMIT` | This parameter is ignored. Oracle Data Pump Import automatically performs a commit after each table is processed. |
| `COMPILE` | This parameter is ignored. Oracle Data Pump Import compiles procedures after they are created. If necessary for dependencies, a recompile can be run. |
| `CONSTRAINTS` | If original Import used `CONSTRAINTS=n`, then Oracle Data Pump Import uses the `EXCLUDE=CONSTRAINT` parameter.<br>If original Import used `CONSTRAINTS=y`, then the parameter is ignored. The parameter does not need to be remapped, because that is the Oracle Data Pump Import default behavior. |
| `DATAFILES` | The Oracle Data Pump Import `TRANSPORT_DATAFILES` parameter is used. |

Table 4-2 (Cont.) How Oracle Data Pump Import Handles Original Import Parameters

| Original Import Parameter | Action Taken by Oracle Data Pump Import Parameter |
|---|---|
| DESTROY | If original Import used DESTROY=y, then Oracle Data Pump Import uses the REUSE_DATAFILES=y parameter.<br>If original Import used DESTROY=n, then the parameter is ignored and does not need to be remapped because that is the Oracle Data Pump Import default behavior. |
| FEEDBACK | The Oracle Data Pump Import STATUS=30 command is used. Note that this is not a direct mapping, because the STATUS command returns the status of the import job, as well as the rows being processed.<br>In original Import, feedback was given after a certain number of rows, as specified with the FEEDBACK command. In Oracle Data Pump Import, the status is given every so many seconds, as specified by STATUS. |
| FILE | Oracle Data Pump Import attempts to determine the path that was specified or defaulted to for the FILE parameter, and also to determine whether a directory object exists to which the schema has read and write access.<br>Original Export and Import and Data Pump Export and Import differ on where dump files and log files can be written to and read from because the original version is client-based and Data Pump is server-based. |
| FILESIZE | This parameter is ignored, because the information is already contained in the Oracle Data Pump dump file set. |
| FROMUSER | The Oracle Data Pump Import SCHEMAS parameter is used. If FROMUSER was used without TOUSER also being used, then import schemas that have the IMP_FULL_DATABASE role cause Oracle Data Pump Import to attempt to create the schema and then import that schema's objects. Import schemas that do not have the IMP_FULL_DATABASE role can only import their own schema from the dump file set. |
| GRANTS | If original Import used GRANTS=n, then Oracle Data Pump Import uses the EXCLUDE=OBJECT_GRANT parameter.<br>If original Import used GRANTS=y, then the parameter is ignored and does not need to be remapped because that is the Oracle Data Pump Import default behavior. |

**Table 4-2 (Cont.) How Oracle Data Pump Import Handles Original Import Parameters**

| Original Import Parameter | Action Taken by Oracle Data Pump Import Parameter |
|---|---|
| IGNORE | If original Import used IGNORE=y, then Oracle Data Pump Import uses the TABLE_EXISTS_ACTION=APPEND parameter. This causes the processing of table data to continue.<br>If original Import used IGNORE=n, then the parameter is ignored and does not need to be remapped, because that is the Oracle Data Pump Import default behavior. |
| INDEXES | If original Import used INDEXES=n, then Oracle Data Pump Import uses the EXCLUDE=INDEX parameter.<br>If original Import used INDEXES=y, then the parameter is ignored and does not need to be remapped, because that is the Oracle Data Pump Import default behavior. |
| INDEXFILE | The Oracle Data Pump Import SQLFILE={directory-object:}filename and INCLUDE=INDEX parameters are used.<br>The same method and attempts made when looking for a directory object described for the FILE parameter also take place for the INDEXFILE parameter.<br>If no directory object was specified on the original Import, then Oracle Data Pump Import uses the directory object specified with the DIRECTORY parameter. |
| LOG | Oracle Data Pump Import attempts to determine the path that was specified or defaulted to for the LOG parameter, and also to determine whether a directory object exists to which the schema has read and write access.<br>The contents of the log file will be those of an Oracle Data Pump Import operation. |
| RECORDLENGTH | This parameter is ignored, because Oracle Data Pump handles issues about record length internally. |
| RESUMABLE | This parameter is ignored, because this functionality is automatically provided for users who have been granted the IMP_FULL_DATABASE role. |
| RESUMABLE_NAME | This parameter is ignored, because this functionality is automatically provided for users who have been granted the IMP_FULL_DATABASE role. |
| RESUMABLE_TIMEOUT | This parameter is ignored, because this functionality is automatically provided for users who have been granted the IMP_FULL_DATABASE role. |

**Table 4-2 (Cont.) How Oracle Data Pump Import Handles Original Import Parameters**

| Original Import Parameter | Action Taken by Oracle Data Pump Import Parameter |
|---|---|
| ROWS=N | If original Import used ROWS=n, then Oracle Data Pump Import uses the CONTENT=METADATA_ONLY parameter.<br>If original Import used ROWS=y, then Oracle Data Pump Import uses the CONTENT=ALL parameter. |
| SHOW | If SHOW=y is specified, then the Oracle Data Pump Import parameter SQLFILE=[directory_object:]file_name is used to write the DDL for the import operation to a file. Only the DDL (not the entire contents of the dump file) is written to the specified file. (Note that the output is not shown on the screen, as it was in original Import.)<br>The file name given is the file name specified on the DUMPFILE parameter (or on the original Import FILE parameter, which is remapped to DUMPFILE). If multiple dump file names are listed, then the first file name in the list is used. The file is placed in the directory object location specified on the DIRECTORY parameter, or the directory object included on the DUMPFILE parameter. (Directory objects specified on the DUMPFILE parameter take precedence.) |
| STATISTICS | This parameter is ignored, because statistics are always saved for tables as part of an Oracle Data Pump Import operation. |
| STREAMS_CONFIGURATION | This parameter is ignored, because Oracle Data Pump Import automatically determines it; it does not need to be specified. |
| STREAMS_INSTANTIATION | This parameter is ignored, because Oracle Data Pump Import automatically determines it; it does not need to be specified |
| TABLESPACES | If original Import also specified TRANSPORT_TABLESPACE=n (the default), then Oracle Data Pump Import ignores the TABLESPACES parameter.<br>If original Import also specified TRANSPORT_TABLESPACE=y, then Oracle Data Pump Import takes the names supplied for this TABLESPACES parameter and applies them to the Oracle Data Pump Import TRANSPORT_TABLESPACES parameter. |
| TOID_NOVALIDATE | This parameter is ignored. OIDs are no longer used for type validation. |

**Table 4-2 (Cont.) How Oracle Data Pump Import Handles Original Import Parameters**

| Original Import Parameter | Action Taken by Oracle Data Pump Import Parameter |
|---|---|
| TOUSER | The Oracle Data Pump Import REMAP_SCHEMA parameter is used. There can be more objects imported than with original Import. Also, Oracle Data Pump Import can create the target schema, if it does not already exist.<br>The FROMUSER parameter must also have been specified in original Import. If FROMUSER was not originally specified, then the operation fails. |
| TRANSPORT_TABLESPACE | The TRANSPORT_TABLESPACE parameter is ignored, but if you also specified the DATAFILES parameter, then the import job continues to load the metadata. If the DATAFILES parameter is not specified, then an ORA-39002:invalid operation error message is returned. |
| TTS_OWNERS | This parameter is ignored because this information is automatically stored in the Oracle Data Pump dump file set. |
| VOLSIZE | When the original Import VOLSIZE parameter is used, it means the location specified for the dump file is a tape device. The Oracle Data Pump Import dump file format does not support tape devices. Therefore, this operation terminates with an error. |

## 4.3 Management of File Locations in Oracle Data Pump Legacy Mode

Original Export and Import and Oracle Data Pump Export and Import differ on where dump files and log files can be written to and read from, because the original version is client-based, and Data Pump is server-based.

Original Export and Import used the FILE and LOG parameters to specify dump file and log file names, respectively. These file names always refer to files local to the client system. They can also contain a path specification.

Oracle Data Pump Export and Import used the DUMPFILE and LOGFILE parameters to specify dump file and log file names, respectively. These file names always refer to files local to the server system, and cannot contain any path information. Instead, a directory object is used to indirectly specify path information. The path value defined by the directory object must be accessible to the server. The directory object is specified for an Oracle Data Pump job through the DIRECTORY parameter. It is also possible to prepend a directory object to the file names passed to the DUMPFILE and LOGFILE parameters. For privileged users, Oracle Data Pump supports the use of a default directory object if one is not specified on the command line. This default directory object, DATA_PUMP_DIR, is set up at installation time.

If Oracle Data Pump legacy mode is enabled, and if the original Export FILE=*filespec* parameter and/or LOG=*filespec* parameter are present on the command line, then the following rules of precedence are used to determine file location:

- If the FILE parameter and LOG parameter are both present on the command line, then the rules of precedence are applied separately to each parameter.
- If a mix of original Export/Import and Oracle Data Pump Export/Import parameters are used, then separate rules apply to them.

  For example, suppose you have the following command:

  ```
  expdp system FILE=/user/disk/foo.dmp LOGFILE=foo.log DIRECTORY=dpump_dir
  ```

  In this case, the Oracle Data Pump legacy mode file management rules, as explained in this section, apply to the FILE parameter. The normal (that is, non-legacy mode) Oracle Data Pump file management rules for default locations of Dump, Log, and SQL files locations apply to the LOGFILE parameter.

**Example 4-1 Oracle Data Pump Legacy Mode File Management Rules Applied**

File management proceeds in the following sequence:

1. If you specify a path location as part of the file specification, then Oracle Data Pump attempts to look for a directory object accessible to the schema running the export job whose path location matches the path location of the file specification. If such a directory object cannot be found, then an error is returned. For example, suppose that you defined a server-based directory object named USER_DUMP_FILES with a path value of '/disk1/user1/dumpfiles/', and that read and write access to this directory object has been granted to the hr schema. The following command causes Oracle Data Pump to look for a server-based directory object whose path value contains '/disk1/user1/dumpfiles/' and to which the hr schema has been granted read and write access:

   ```
   expdp hr FILE=/disk1/user1/dumpfiles/hrdata.dmp
   ```

   In this case, Oracle Data Pump uses the directory object USER_DUMP_FILES. The path value, in this example '/disk1/user1/dumpfiles/', must refer to a path on the server system that is accessible to Oracle Database.

   If a path location is specified as part of the file specification, then any directory object provided using the DIRECTORY parameter is ignored. For example, if you issue the following command, then Oracle Data Pump does not use the DPUMP_DIR directory object for the file parameter, but instead looks for a server-based directory object whose path value contains '/disk1/user1/dumpfiles/' and to which the hr schema has been granted read and write access:

   ```
   expdp hr FILE=/disk1/user1/dumpfiles/hrdata.dmp DIRECTORY=dpump_dir
   ```

2. If you have not specified a path location as part of the file specification, then the directory object named by the DIRECTORY parameter is used. For example, if you issue the following command, then Oracle Data Pump applies the path location defined for the DPUMP_DIR directory object to the hrdata.dmp file:

   ```
   expdp hr FILE=hrdata.dmp DIRECTORY=dpump_dir
   ```

3. If you specify no path location as part of the file specification, and no directory object is named by the DIRECTORY parameter, then Oracle Data Pump does the following, in the order shown:

a. Oracle Data Pump looks for the existence of a directory object of the form `DATA_PUMP_DIR_`*`schema_name`*, where *`schema_name`* is the schema that is running the Oracle Data Pump job. For example, if you issued the following command, then it would cause Oracle Data Pump to look for the existence of a server-based directory object named `DATA_PUMP_DIR_HR`:

   ```
   expdp hr FILE=hrdata.dmp
   ```

   The `hr` schema also must have been granted read and write access to this directory object. If such a directory object does not exist, then the process moves to step **b**.

b. Oracle Data Pump looks for the existence of the client-based environment variable `DATA_PUMP_DIR`. For instance, suppose that a server-based directory object named `DUMP_FILES1` has been defined, and the `hr` schema has been granted read and write access to it. Then on the client system, you can set the environment variable `DATA_PUMP_DIR` to point to `DUMP_FILES1` as follows:

   ```
   setenv DATA_PUMP_DIR DUMP_FILES1
   expdp hr FILE=hrdata.dmp
   ```

   Oracle Data Pump then uses the served-based directory object `DUMP_FILES1` for the `hrdata.dmp` file.

   If a client-based environment variable `DATA_PUMP_DIR` does not exist, then the process moves to step **c**.

c. If the schema that is running the Oracle Data Pump job has DBA privileges, then the default Oracle Data Pump directory object, `DATA_PUMP_DIR`, is used. This default directory object is established at installation time. For example, the following command causes Oracle Data Pump to attempt to use the default `DATA_PUMP_DIR` directory object, assuming that system has DBA privileges:

   ```
   expdp system FILE=hrdata.dmp
   ```

**Related Topics**

- Understanding Dump, Log, and SQL File Default Locations

## 4.4 Adjusting Existing Scripts for Oracle Data Pump Log Files and Errors

When you use Oracle Data Pump in legacy mode, you must review and update your existing scripts written for original Export and Import

Oracle Data Pump legacy mode requires that you make adjustments to existing scripts, because of differences in file format and error reporting.

- Log Files
  Oracle Data Pump Export and Import do not generate log files in the same format as those created by original Export and Import.
- Error Cases
  The errors that Oracle Data Pump Export and Import generate can be different from the errors generated by original Export and Import.

- Exit Status
  Oracle Data Pump Export and Import have enhanced exit status values to enable scripts to better determine the success or failure of export and import jobs.

## 4.4.1 Log Files

Oracle Data Pump Export and Import do not generate log files in the same format as those created by original Export and Import.

You must update any scripts you have that parse the output of original Export and Import, so that they handle the log file format used by Oracle Data Pump Export and Import. For example, the message `Successfully Terminated` does not appear in Oracle Data Pump log files.

## 4.4.2 Error Cases

The errors that Oracle Data Pump Export and Import generate can be different from the errors generated by original Export and Import.

For example, suppose that a parameter that is ignored by Oracle Data Pump Export would have generated an out-of-range value in original Export. In that case, an informational message is written to the log file stating that the parameter is being ignored. However, no value checking is performed, so no error message is generated.

## 4.4.3 Exit Status

Oracle Data Pump Export and Import have enhanced exit status values to enable scripts to better determine the success or failure of export and import jobs.

Because Oracle Data Pump Export and Import can have different exit status values, Oracle recommends that you review, and if necessary, update, any scripts that look at the exit status.