

# LanguageManual ImportExport

## Import/Export

- [Import/Export](#)
  - [Overview](#)
  - [Export Syntax](#)
  - [Import Syntax](#)
  - [Replication usage](#)
  - [Examples](#)



### Version

The `EXPORT` and `IMPORT` commands were added in Hive 0.8.0 (see [HIVE-1918](#)).

Replication extensions to the `EXPORT` and `IMPORT` commands were added in Hive 1.2.0 (see [HIVE-7973](#) and [Hive Replication Development](#)).

## Overview

The `EXPORT` command exports the data of a table or partition, along with the metadata, into a specified output location. This output location can then be moved over to a different Hadoop or Hive instance and imported from there with the `IMPORT` command.

When exporting a partitioned table, the original data may be located in different HDFS locations. The ability to export/import a subset of the partition is also supported.

Exported metadata is stored in the target directory, and data files are stored in subdirectories.

The `EXPORT` and `IMPORT` commands work independently of the source and target metastore DBMS used; for example, they can be used between Derby and MySQL databases.

`IMPORT` will create target table/partition if it does not exist. All the table properties/parameters will be that of table that was used in `EXPORT` to generate the archive. If target exists, checks are performed that it has appropriate schema, Input/OutputFormat, etc. If target table exists and is not partitioned, it must be empty. If target table exists and is partitioned, partitions being imported must not exist in the table.

## Export Syntax

```
EXPORT TABLE tablename [PARTITION (part_column="value"[, ...])]
  TO 'export_target_path' [ FOR replication('eventid') ]
```

## Import Syntax

```
IMPORT [[EXTERNAL] TABLE new_or_original_tablename [PARTITION (part_column="value"[, ...])]
  FROM 'source_path'
  [LOCATION 'import_target_path']
```

## Replication usage

The `EXPORT` and `IMPORT` commands behave slightly differently when used in the context of replication, and are intended to be used by tools that perform replication between hive warehouses. In most cases, end users will not need to use this additional tag, except when doing a manual bootstrap of a replication-destination warehouse so that an incremental replication tool can take over from that point.

They make use of a special table property called "repl.last.id" in a table or partition (depending on what object is being replicated) object to make sure that a replication export/import will only update objects if the update is newer than the object it affects. On the export end, it tags the replication export dump with an id that is monotonically increasing on the source warehouse (incrementing each time there is a source warehouse metastore modification). In addition, an export that is tagged as being for replication will not result in an error if it attempts to export an object which does not currently exist. (This is because in the general flow of replication, it is quite possible that by the time an event is acted upon for replication by an external tool, it is possible that the object has been removed, and thus, should not halt the replication pipeline.)

On the import side, there is no syntax change, but import is run on an export dump that was generated with the `FOR REPLICATION` tag, it will check the object it is replicating into if it exists. If that object already exists, it checks the `repl.last.id` property of that object to determine if what is being imported is newer than the current state of the object in the destination warehouse. If the update is newer, then it replaces the object with the newer information. If the update is older than the object already in place, the update is ignored, and causes no error.

For those using EXPORT for the first-time manual bootstrapping usecase, users are recommended to use a "FOR replication('bootstrapping') " tag. (Advanced users note : The choice of "bootstrapping" here is arbitrary-ish, and could just as well have been "foo". The real goal is to have a value such that all further incremental replication ids will be greater than this original id. Thus, the integral value of this initial id should be 0, and thus, any string which does not contain numbers is acceptable. Having an initial tag be "123456", however, would be bad, as it could cause further updates which have a repl.last.id < 123456 to not be applied.)

## Examples

Simple export and import:

```
export table department to 'hdfs_exports_location/department';
import from 'hdfs_exports_location/department';
```

Rename table on import:

```
export table department to 'hdfs_exports_location/department';
import table imported_dept from 'hdfs_exports_location/department';
```

Export partition and import:

```
export table employee partition (emp_country="in", emp_state="ka") to 'hdfs_exports_location/employee';
import from 'hdfs_exports_location/employee';
```

Export table and import partition:

```
export table employee to 'hdfs_exports_location/employee';
import table employee partition (emp_country="us", emp_state="tn") from 'hdfs_exports_location/employee';
```

Specify the import location:

```
export table department to 'hdfs_exports_location/department';
import table department from 'hdfs_exports_location/department'
    location 'import_target_location/department';
```

Import as an external table:

```
export table department to 'hdfs_exports_location/department';
import external table department from 'hdfs_exports_location/department';
```