

Jedox Database Files

Jedox databases consist of 4 different file types: comma-separated values (**.csv**), binary (**.bin**), log (**.log**), and archive (**.archive**) files. The **.bin** and **.csv** files contain the data last saved. Changes from the current session—that is, since the OLAP Server was last started—are registered in the log files, also called “journals”. When stopping Jedox OLAP Server, the database state is transferred from memory to the **.bin** and **.csv** files (a so-called “commit”). After that, the contents from the **.log** files are copied into the **.archive** files. In the case of an unexpected process kill or crash, the server will recover data from the **.log** files during startup. Please note that this recovery will usually take more time than committing the same data on shutdown would have taken.

If your database takes up a lot of hard-disk space on the server, you can delete the **.archive** files from the database directory after stopping the Jedox server. After restarting the Jedox server, the needed **.archive** files will be created anew.

Before deleting these files from your database directory, we recommend that you always complete a backup.

Comma-separated values (.csv) files

The structure of Jedox databases is defined in a comma-separated values (.csv) file **database-*.csv**. In this file, cubes, dimensions, elements, and their properties are set. The filename includes a UNIX timestamp, i.e. **database-1557741467210061.csv**.

Note: if the palo.ini key dimension-file-format binary has been set, then database-*.csv will contain only the dimensions, cubes, and their layout, but not all elements in all dimensions. See the section [Binary files](#) below for more information.

Each cube that exists in database-*.csv (entries in section [CUBES]) has in turn its own CSV file, in which structure and cube values are specified further. These files contain the currently saved Jedox values.

Each Jedox cube also has a unique CSV file that contains the description of the rules in the cube (if any).

Below is an excerpt from the Jedox Demo database with the following entries in in the file database.csv in section [CUBES]:

```
0;"#_GROUP_CUBE_DATA";0;1;1;0;0;
1;"#_Months";3;2;3;0;0;
2;"#_GROUP_DIMENSION_DATA_Months";0;2;1;0;0;
3;"#_Years";5;4;3;0;0;
4;"#_GROUP_DIMENSION_DATA_Years";0;4;1;0;0;
5;"#_Regions";7;6;3;0;0;
6;"#_GROUP_DIMENSION_DATA_Regions";0;6;1;0;0;
7;"#_Datatypes";9;8;3;0;0;
8;"#_GROUP_DIMENSION_DATA_Datatypes";0;8;1;0;0;
9;"#_Measures";11;10;3;0;0;
10;"#_GROUP_DIMENSION_DATA_Measures";0;10;1;0;0;
11;"#_Products";13;12;3;0;0;
12;"#_GROUP_DIMENSION_DATA_Products";0;12;1;0;0;
```

13;"Sales";12;6;2;4;8;10;2;1;1;

14;"#_#_CUBE_";14;1;3;0;0;

There is another csv file for each entry. The individual cubes are described briefly below:

Cube	Description
0;"#_GROUP_CUBE_DATA";0;1;1;0;0;	Contains the dimension cubes and groups, and the rights of the groups to the cubes.
1;"#_Months";3;2;3;0;0; 3;"#_Years";5;4;3;0;0; 5;"#_Regions";7;6;3;0;0; 7;"#_Datatypes";9;8;3;0;0; 9;"#_Measures";11;10;3;0;0; 11;"#_Products";13;12;3;0;0;	These six cubes contain the attribute cube for each dimension.
2;"#_GROUP_DIMENSION_DATA_Months";0;2;1;0;0; 4;"#_GROUP_DIMENSION_DATA_Years";0;4;1;0;0; 6;"#_GROUP_DIMENSION_DATA_Regions";0;6;1;0;0; 8;"#_GROUP_DIMENSION_DATA_Datatypes";0;8;1;0;0; 10;"#_GROUP_DIMENSION_DATA_Measures";0;10;1;0;0; 12;"#_GROUP_DIMENSION_DATA_Products";0;12;1;0;0;	These cubes contain the user group and the dimensions. The rights for each element for each group are registered here.
13;"Sales";12;6;2;4;8;10;2;1;1;	Contains the data for the cube "Sales".
14;"#_#_CUBE_";14;1;3;0;0;	This cube contains the data for the attribute cube for each cube dimension.

Binary (.bin) files

Each cube has a .bin file that contains the same data as the corresponding .csv file, but as binary data rather than text. This file appears as **database_CUBE_*.bin**. The .bin files are usually smaller in size, so loading and saving is much faster than for the .csv files. The binary format greatly speeds up the time needed for processing files on the file system, such as when the server is shut down.

Dimension data can also be stored in .bin files. Setting this option improves startup performance. Note that this option is not set by default; you must enable it in palo.ini with the key [dimension-file-format binary](#). Once set, dimension files will be generated as **database_DIM_*.bin**.

To store cube/dimension data *only* in binary format, you can set this option in palo.ini with the key [no-csv-save \(cube data\)](#) and [no-csv-save-dim \(dimension data\)](#). Setting this option improves shutdown performance. If this option is set for cubes, then the file **database_CUBE_*.csv** will be generated; for dimensions, the file **database_DIM_*.csv** will be generated.

Note: unlike .csv files, binary files are not transferable between Linux and Windows systems.

Log files

A log file is created for each cube defined in Jedox. All values entered in the Jedox OLAP Server are protocolled in this file. If the database is

saved, the values stored here are copied to the respective .archive file. If the OLAP Server wasn't shutdown properly, the .log files are used for data recovery as described above. Note that log files contain OLAP Server version information, and the OLAP Server will only process log files of the same version. If version of a log file is significantly lower than the current OLAP Server version, it's possible that the database can't be loaded. To avoid that, it's recommended to either save all databases explicitly with the OLAP Server API command or, alternatively, restart the OLAP Server once before applying any Jedox updates.

Archive files

All changes are written into the .archive files. Both value changes as well as structural changes are logged together with the name of the user. However, if a database becomes corrupted, there is no generic possibility to recover a database exclusively from .archive files. The files can only be used for tracing changes.

Archive files may be deleted regularly (after a backup) to keep the data quantity in the server system low. A restart of JedoxSuiteMolapService will create these files anew.

Saved files

Files with the name saved-xxxxxx (x = numbers) and a file size of 0 KB are internal files that are created and can be ignored once the



databases are saved. Do not delete these files.
