

Data import and export

From the standard user interface its possible to both export and import CSV data files.

- [Exporting spreadsheet data](#)
- [Importing spreadsheet data](#)
- [Understand how imports work](#)

Exporting spreadsheet data

Data exports can be done in a blink of an eye

1. In list mode of an solution click the XL sheet icon.
2. Choose relevant export options

Note that the export provided respects current filters, sort order and grouping.

The export options are as follows

- All records: Include all records in range
- All fields: Include all exportable fields
- Explode parents: Include fields from parent records (default in list)
- Explode children: Include fields from child records in denormalized format (default in list)
- System export: Use field system names in columns (good for later insert/updates)

Note that the export allow you to build VERY large data sets, so be careful when selecting the "All records" option.

Importing spreadsheet data

In list mode of an solution click the Package import icon.

Columns in the import file will be matched to fields in the active solution. Preferably system field names should be used, as these are guaranteed to be unique. A template for the data can easily be built by selecting the "System export" option during export of existing data.

The result of an import operation is another CSV with response on each line in the file, either

- SUCCES
- FAILED [reason why]

In case of inserts an DataID column is also added to the response.

Tip: In case some lines fail. Just remove the lines with SUCCES, correct the faulty ones and try again after saving changes to the response file.

New data (insert)

New records will be created in accordance with matched fieldnames in the file. Missing values will be set to either the field default value or simply declared empty.

The presence of "DataID" column will make the import fail (countermeasure again human errors during imports).

Changes to existing data (update)

Existing records will be mached using one of the following methods

- Lookup by column named "DataID" in the import file
- Match by database lookup of all fields marked with "Is part of a unique key"

On the matching columns present in the file will be updated. Unmatched values will simply hold on to the value they had in the latest revision.

How to import data

When importing a file you can either update or insert records.

After the import is handled a file containing the results are returned to the user.

General rules using the import

- Validation rules apply by default (**Disable validation**)
- Codeunits are triggered by default (**Disable codeunits**)
- Field system names are used by default (**Allow field display names**)

Field names

Field names should use the system format (uppercase without any special characters), but it is possible to allow display name during imports (**Allow field display names**). An easy way to get the file format right is to export a couple of records and choosing **System format**.

DataID plays a special role in regard to what you want to do

- When creating data the DataID may NOT be included
- When updating data a valid DataID MUST be applied

Field values

Generally values must abide by the same restrictions as during form interaction.

- Validation
- Dependency

In case of lookup or relational fields just insert the value displayed (not the ID). A failure to match a value will result in that record being imported/updated.

It is however possible to skip the handling of fields on certain records by

- Leaving the value empty
- Checking the import option **Ignore empty values**

Troubleshooting

Complete failure to import can be caused by

- Used field name display values
- Forgot DataID column in an update

- Included DataID column in an insert

Errors in single records could be caused by

- Some required fields are missing
- Status field was left empty
- Values in lookup fields are misspelled

Errors during import will be documented in detail in the backend Events.

Understand how imports work

Data inserts and updates are carried out by a system user, that has exactly the same privileges as a normal user. If field permissions are missing field value changes will be ignored, and if restriction is not met the operation will fail as normally. Finally all dynamic effects are the same, including status actions, codeunit execution etc. The later can however be disabled explicitly when uploading data to the server.