



**Updated Archive Data Import Utility
Released with V4.18.16 and V5.18.16**

30th September 2021

Contents

Introduction	3
Backward compatibility.....	3
Added features	3
Cautionary notes and special characters	9

Introduction

As part of the release of versions 4.18.16 and 5.18.16, the standard utility **ArchiveCSV.r** (supplied in the \sosobject\PM folder) has been updated to allow additional archive-level data to be updated in bulk from a CSV file.

Backward compatibility

The updated version of this utility retains backward compatibility with previous versions: it can still handle an input CSV file in the format it was originally designed to load:

- No column headers
- Column 1 = matter code (must be full exact matter code relating to a valid archive)
- Column 2 = fee earner code
- Column 3 = archive location
- Column 4 = destruction date (in DD/MM/YYYY format)

The original behaviour of these updates is retained: any column value which is left blank in the CSV file will leave existing archive data in the relevant field untouched. It has never been possible (and it is still not possible) to clear the values in these fields using this utility.

Added features

The original “no header” capability of the utility (i.e. the CSV file contains only data rows and no header row) has been extended to allow additional archive fields to be updated:

- Column 5 = value to append to archive notes (if specified)
- Column 6-10 = text values for archive UDF text fields 1-5
- Column 11-15 = date values for archive UDF date fields 1-5 (in DD/MM/YYYY format)
- Column 16-20 = numeric (decimal) values for archive UDF value fields 1-5

If there is a value in column 5 then this is appended to the archive notes field, unless the archive notes field already has this text at the end of the notes. If the new notes are being appended via the CSV file and archive notes already exist, then a carriage return is inserted before the new notes are appended to keep them on their own line, for example:



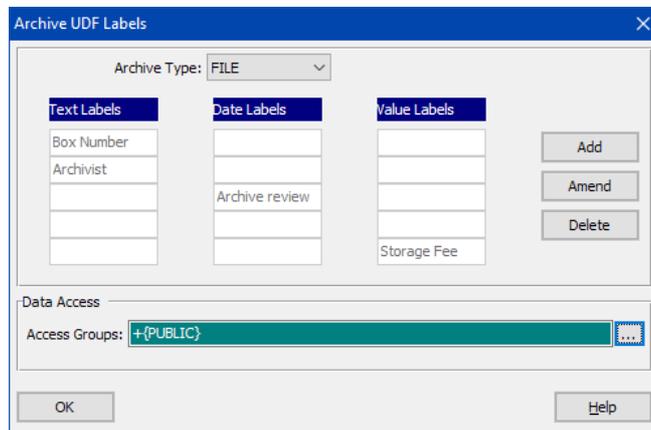
The behaviour of columns 6-20 is as follows:

- You can only specify a value if the archive UDF text/date/value field has been defined. For example trying to update date field 4 with a date when UDF date field 4 has not been defined will generate an error.
- A blank value will not update the UDF field at all
- A single question mark character (either quoted or not quoted, e.g. ? or “?”) will clear the field of its current contents. For numeric value UDF fields the CSV file could also contain a value of 0 to clear the field.
- All other values will overwrite the existing field value

In addition, this utility can now operate in a different “mode” where the first line of the file is a header, with each column value indicating the archive field name to update. To switch the utility into this mode:

- Column 1 of line 1 cannot be a valid matter code (because it is a header row) but the actual column header does not need to be anything specific. Column 1 of the input file is always assumed to be the matter code
- For column 2 and above in line 1, the column header can be one of the following to indicate which archive data field the column relates to in terms of making updates to data. These column header values are not case sensitive.
 - o **FE**
 - o **Location**
 - o **DestructionDate**
 - o **Notes**
 - o ... or the screen label for any defined UDF field

For example, with these UDF fields defined:



The input CSV file can be constructed as follows:

```
Matter,Box Number,FE,Archive Review,Notes,Storage Fee,Archivist,Location,DestructionDate
0000101,BOX123,MP,10/10/2022,Review required,499.99,G Smith,Bath Basement,10/03/2050
```

Below is this same example file shown in tabular format:

Matter	Box Number	FE	Archive Review	Notes	Storage Fee	Archivist	Location	DestructionDate
0000101	BOX123	MP	10/10/2022	Review required	499.99	G Smith	Bath Basement	10/03/2050

The loaded data would then be visible in the archive details and notes screens:

Archive Enquiry 1

Register: 123 Client: Lewis, Bryony
 Branch: 001 Description: archive 0000101

Details | Notes | Ledger | Time | Undertakings | Legal Aid | Relationships | Billing History

Type: FILE Date Opened: 27/05/2021 MT Type: CGEN
 Location: Bath Basement Date Closed: 27/05/2021 Fee Earner: MP
 Reference: Date Archived: 27/05/2021 Legal Aid:
 Matter: 0000101 Destruction Date: 10/03/2050 Charge Band:

Archive Enquiry 1

Register: 123 Client: Lewis, Bryony
 Branch: 001 Description: archive 0000101

Details | Notes | Ledger | Time | Undertakings | Legal Aid | Relationships

Existing archive notes
 Review required

Box Number: BOX123
 Archivist: G Smith
 Archive review: 10/10/2022
 Storage Fee: 499.99

If the CSV file was then amended to contain blank values for all but the destruction date as follows:

```
Matter,Box Number,FE,Archive Review,Notes,Storage Fee,Archivist,Location,DestructionDate
0000101,,,,,,,,,10/03/2051
```

Below is this same example file shown in tabular format:

Matter	Box Number	FE	Archive Review	Notes	Storage Fee	Archivist	Location	DestructionDate
0000101								10/03/2051

On loading this file, all existing data would remain unchanged, apart from the destruction date field:

Archive Enquiry 1

Register: 123 Client: Lewis, Bryony
 Branch: 001 Description: archive 0000101

Details | Notes | Ledger | Time | Undertakings | Legal Aid | Relationships | Billing History

Type: FILE Date Opened: 27/05/2021 MT Type: CGEN
 Location: Bath Basement Date Closed: 27/05/2021 Fee Earner: MP
 Reference: Date Archived: 27/05/2021 Legal Aid:
 Matter: 0000101 Destruction Date: 10/03/2051 Charge Band:

Archive Enquiry 1

Register: 123 Client: Lewis, Bryony
 Branch: 001 Description: archive 0000101

Details | Notes | Ledger | Time | Undertakings | Legal Aid | Relationships

Existing archive notes
 Review required

Box Number: BOX123
 Archivist: G Smith
 Archive review: 10/10/2022
 Storage Fee: 499.99

If the CSV file was amended again to have question-mark values in the UDF fields for **Box Number**, **Archive Review**, **Storage Fee** and **Archivist**:

```
Matter,Box Number,FE,Archive Review,Notes,Storage Fee,Archivist,Location,DestructionDate
0000101,?,?,?,?/?
```

Shown in tabular format:

Matter	Box Number	FE	Archive Review	Notes	Storage Fee	Archivist	Location	DestructionDate
0000101	?		?		?	?		

Then the values for these UDF fields would be cleared when the utility is run again:

Box Number:

Archivist:

Archive review:

Storage Fee:

When operating in “CSV header” mode, as long as the first column contains valid matter codes (the header for column 1 is ignored) and there is at least one column header relating to an archive data field that can be updated (**FE, Location, Notes, DestructionDate** or any UDF field label), other columns can exist in the input file but will be ignored by the utility.

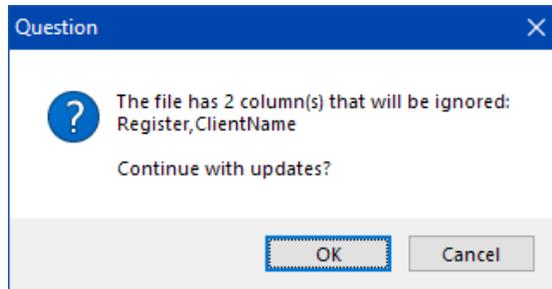
For example if the input file was as followed:

```
Matter,Register,ClientName,Location
0000101,R3931/1,Smith J,Bath Strong Room
```

In tabular format:

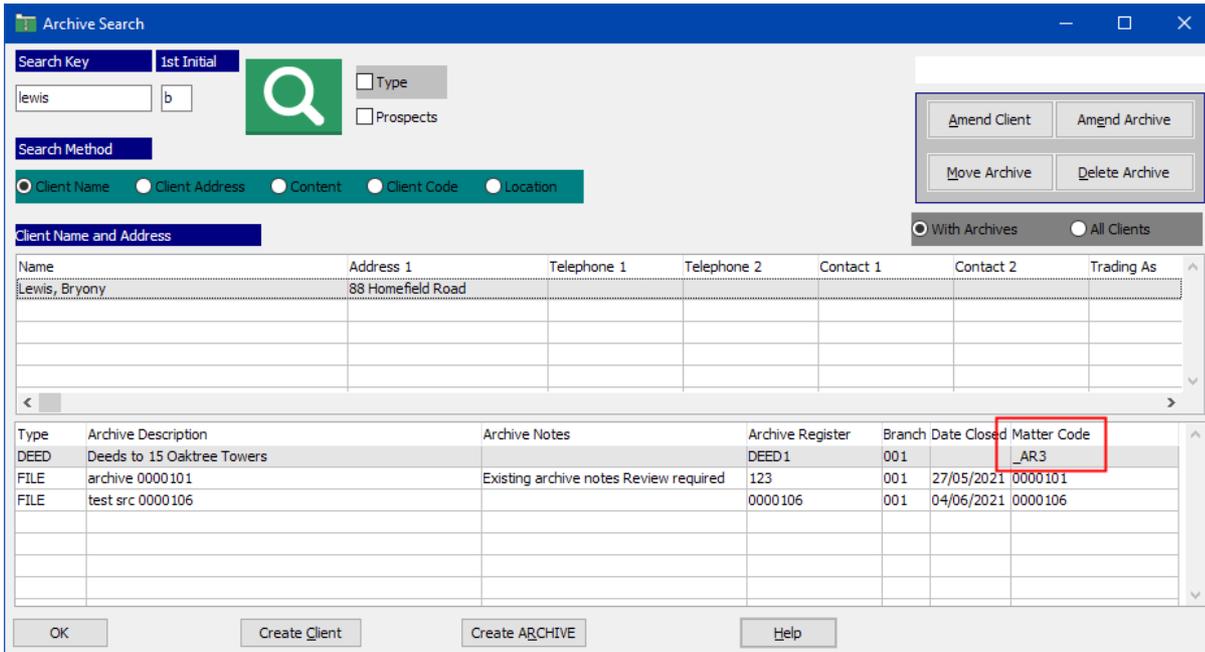
Matter	Register	ClientName	Location
0000101	R3931/1	Smith J	Bath Strong Room

On running the utility using this file the following message will be displayed:

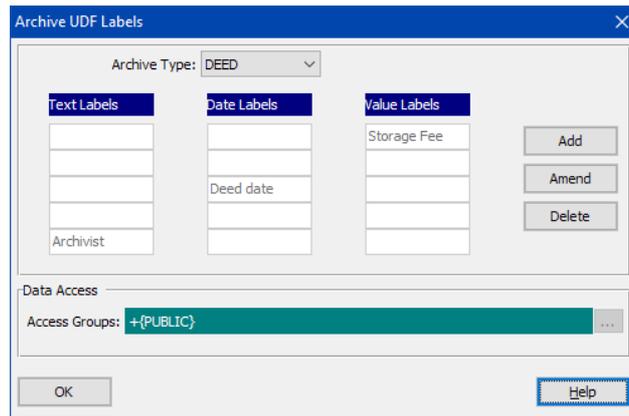


The user has the option at this point to cancel before loading the data if there appears to be a problem with file, or continue if the unrecognized column names are acceptable.

To update data on manual archives (i.e. archives which are not archived matters of type FILE), a valid matter code must be related to the manual archive and the matter code relating to the manual archive should be used in the input file (for example matter codes of the format **_AR<n>**).



Each manual archive type can have its own set of user-defined field definitions, for example:



When using the “header” style of input file, there are no limitations in the file having to contain data for a single archive type (this was also true for the original version of this utility).

For example, a single CSV file can be constructed to update both a FILE and a DEED archive shown in the search screen above, with values for the matching UDF field labels appearing in the same column in the CSV file, even if ultimately these values are stored in different UDF field numbers in the data:

```
Matter,Box Number,FE,Archive Review,Notes,Storage Fee,Archivist,Location,DestructionDate,Deed date
0000101,BOX123,MP,10/10/2022,Review required,499.99,P Jones,Bath Basement,10/03/2050
_AR3,BOX456,,Checked by GKC 23rd Oct 21,220,G Clunes,Tewksbury Strong Room,20/07/2058,20/07/2008
```

Shown in tabular format:

Matter	Box Number	FE	Archive Review	Notes	Storage Fee	Archivist	Location	DestructionDate	Deed date
0000101	BOX123	MP	10/10/2022	Review required	499.99	P Jones	Bath Basement	10/03/2050	
_AR3	BOX456			Checked by GKC 23rd Oct 21	220	G Clunes	Tewksbury Strong Room	20/07/2058	20/07/2008

After loading this file, the DEED and FILE archive data would appear as follows:

Cautionary notes and special characters

All previous limitations with CSV files and compatibility with Microsoft Excel still apply: for example if matter codes contain leading zeros and the CSV file is loaded into Excel, then saved to CSV again from Excel, the leading zeros in the matter code values will be removed by Excel and the file will no longer work. This is because matter codes in column 1 will not match matter codes in the database.

Every line in the CSV file must contain at least 2 column values (matter code plus one other valid field value to update), otherwise the validation phase of the utility will report that there is "Nothing to process" for the lines with fewer than 2 column values and the file will not load.

The new version of the utility will handle special characters in the incoming CSV data better than the previous version, but the following rules still apply to certain special characters: for textual values containing commas (,) the whole text value *must* be surrounded by double quotes. For example if the CSV file was being used to append this text value to archive notes:

Checked by G Smith, T Smith & C Smith

This text would need to be in the CSV file in this format:

"Checked by G Smith, T Smith & C Smith"

So that the file loading process does not interpret the comma character as a field delimiter. If the field value to load contains double-quotes itself, these need to be converted to double-double-quotes in the CSV file, and the whole text field surrounded by double-quotes, for example to load this text value:

Flat 5, "Homeward Bound", 78 High Street, Bath

This would need to be in the CSV file in this format:

"Flat 5, ""Homeward Bound"", 78 High Street, Bath"

All validation is done before the update process is run. Any validation failures found in the input file will result in no data updates at all.