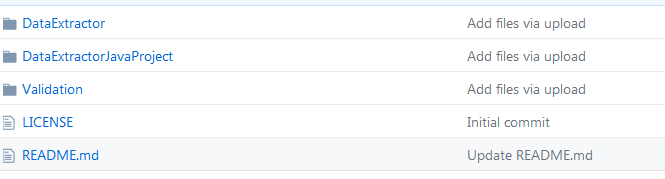
**User guide for data extraction tool**

The github repository at <https://github.com/MRCG-djeffries/DataExtractor> contains 3 folders:



The **DataExtractor** folder contains all the software and test files required to run the application. DataExtractorJavaProject contains the source code for the java application and Validation contains an example validation process.

**The DataExtractor folder contains the files:**

DataExtractor.accdb - MS Access application

OC\_ACCESS\_EXTRACTOR\_V1.0.jar - java XML to Access parser

test\_extract.xml - example OC XML download (contains dummy data only)

juno\_extract.xml - download of JUNO study demo from OC v3.12.2

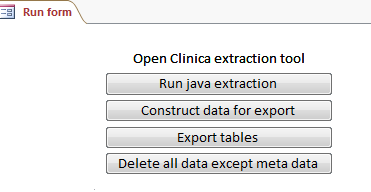
\SqliteFiles\sqlite3.dll - SQLite application (only required for export to SQLite)

\SqliteFiles\SQLite3\_StdCall.dll - SQLite wrapper for VBA (only required for export to SQLite)

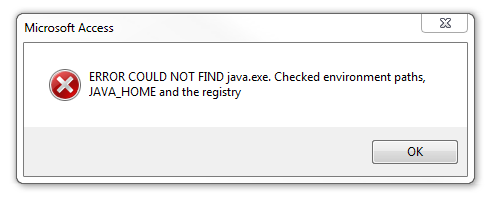
Download **DataExtractor.accdb** and **OC\_ACCESS\_EXTRACTOR\_V1.0.jar** to the **SAME** target folder. If the SQLite export facility is required download the files **sqlite3.dll** and **SQLite3\_StdCall.dll** directly (i.e. no subfolders) to the target folder established previously. (Note if the SQLite export facility is not required then the s\*.dll files in the above list are not required). The two test xml files test\_extract.xml and juno\_extract.xml can be downloaded to any location.

Double click on the DataExtractor.accdb icon, to open the application. The application has been tested with Access applications from 2007 to 2016.

The application will open to the main form: (See Appendix I for security warnings)

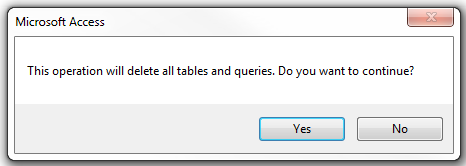


To import an OC XML extraction click on the ‘Run java extraction’ button. The application checks in standard locations and searches the registry for a JRE installation. If no installation is found the following error message appears:

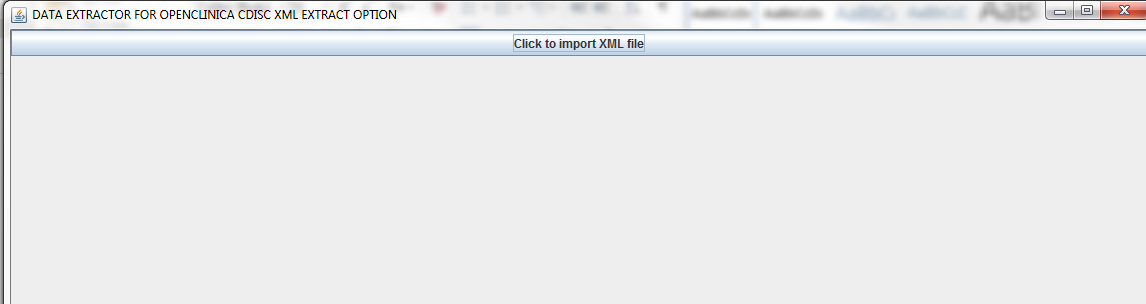


**The application requires a JRE installation – if necessary install the latest JRE from** <http://docs.oracle.com/javase/7/docs/webnotes/install/windows/jre-installation-windows.html> (accessed 1 June 2017).

If a JRE installation is found the following warning appears:



On proceeding a Java file browser requesting an input file of an OC XML download will appear:

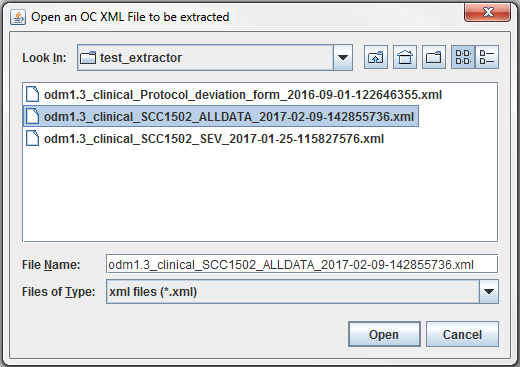


Click the import XML file button and select a suitable file via the resulting browser: (Three test OC XML downloads are included, ~DataExtractor\**test\_extract.xml,** ~DataExtractor\**juno\_extract.xml** and ~Validation\ **adverse\_events\_download.xml**)**.**

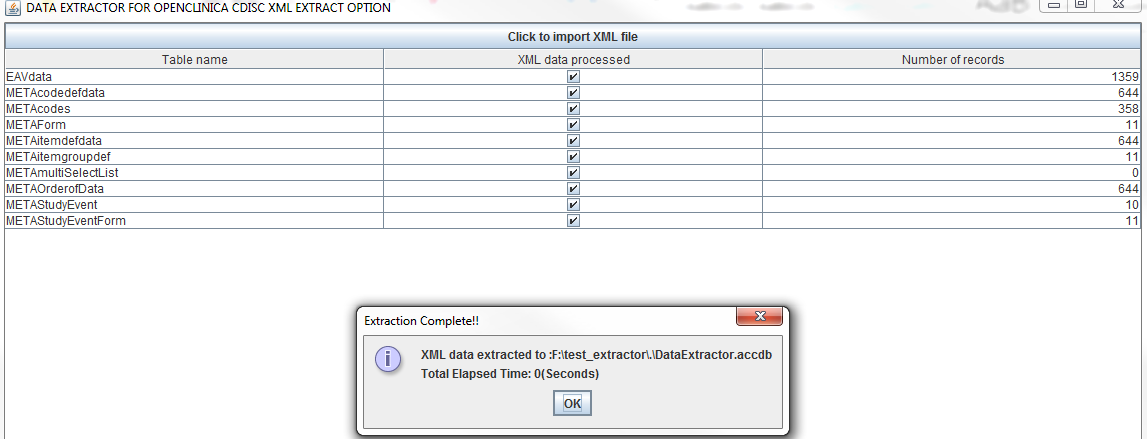
test\_extract.xml consists of a repeating item group within a repeating study event, typical of the hierarchical structure of cluster randomised clinical trials undertaken within MRC The Gambia. OC version 3.4.

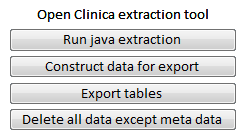
juno\_extract.xml is the extraction of all CRFs for the OC demo diabetes study. This extraction covers all major types, including partial dates. OC version 3.12.2

adeverse\_events\_download.xml (in the Validation folder) consist of a set of CRFs used in an actual study and can be used for both test the extraction and demonstrating validation. Although they are actual forms they are populated with dummy data. OC version 3.4.

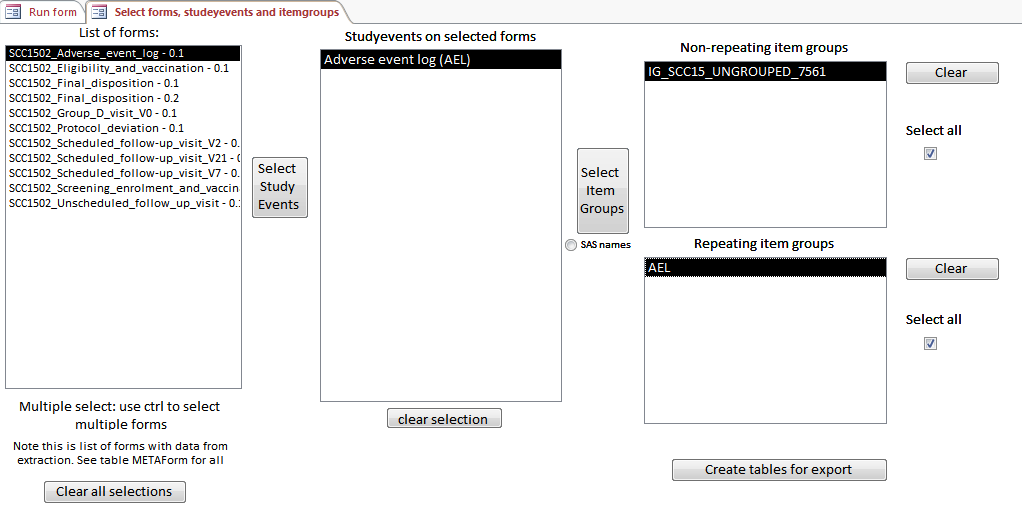


After clicking open the Java application parses the XML data to the XML application. On completion a summary of the records transferred appears and to complete the transfer and re-open the Access application click the OK button below.

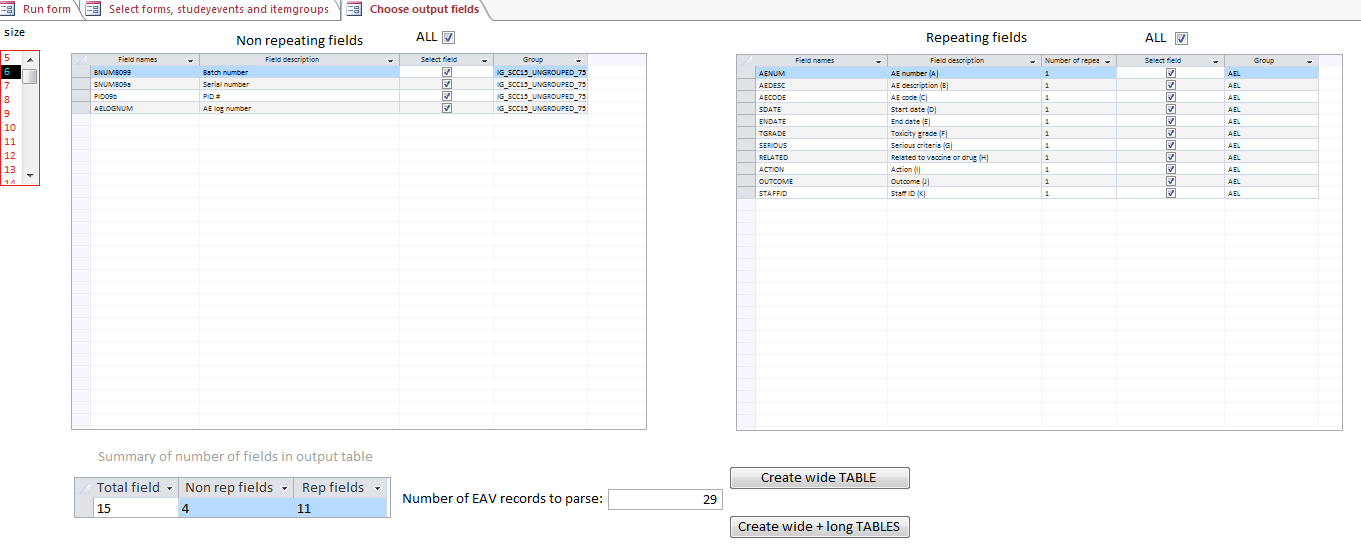




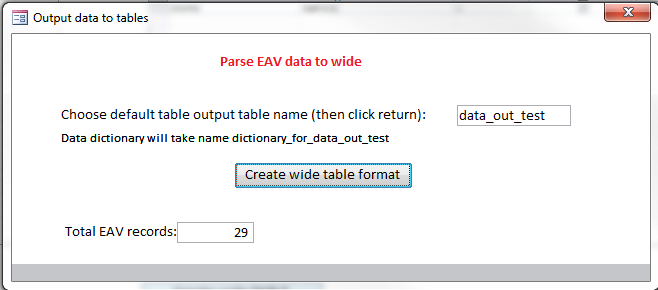
Click on the ‘Construct data for export’ button to build the required dataset. The main window with the OC objects will then appear, simply select the desired objects to create a table of data.



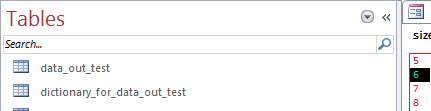
After selecting required objects, click the ‘Create tables for export’ button and then the individual field names appear on the subsequent form.



As described in the paper, either wide tables of repeating data can be created or one-to-many long tables. Click the ‘create wide TABLE’ button and the user is prompted to enter a table name, **click return** and the following menu appears.

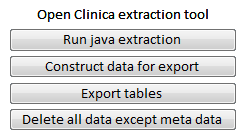


Finally click the ‘Create wide table format’ button to create the table data\_out\_test. This can be verified by opening the Access table objects tab and data\_out\_test table and its accompanying data dictionary can be seen.

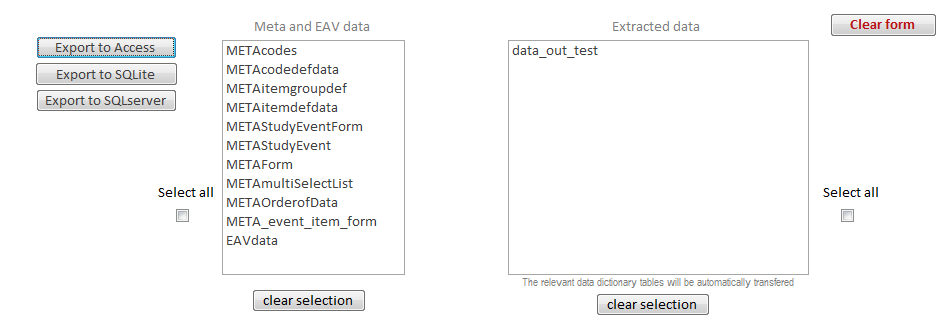


To export the data to the analysis/reporting database, return to the run form tab:





Click on the ‘Export tables’ button to reveal the export menu, where both the data tables and the meta-data is available for export.



The Access and SQLite options warn users if the selected database already exists all tables will be overwritten.

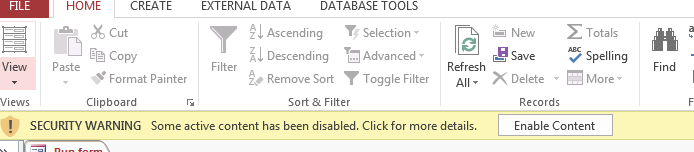
For export to SQL server the database is assumed to exist and for security reasons SQL server database cannot be created by users. Connections to SQL server assumes integrated security via SSPI.

**Note to export to SQLite the two open source dlls, sqlite3 (sqlite application) SQLite3\_StdCall (fast transfer class used in DataExtractor.accdb) contained in the subfolder \sqlite, must be moved to the same folder as the MS Access application DataExtractor.accdb.**

**To export to Access it is recommended that the folder to which the data will be exported is a trusted location. A warning message appears informing users of this.**

**Appendix I Microsoft security warnings**

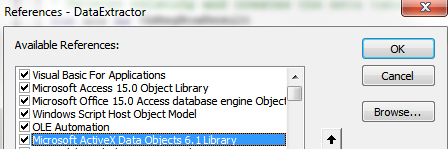
If the application is opened in anew folder for the first time a security warning will appear (exact format will depend on the version of windows), but have the form of below.



To allow the application to run, user must click on the Enable Content button. An alternative is to declare the folder where the application resides a ‘trusted location’, using the Microsoft Trust Center tool.

**Appendix II VBA dependencies (will load automatically)**

The VBA code requires the following libraries:



Version numbers are not critical

**Appendix III Form description**

|  |  |
| --- | --- |
| Construct\_data | Table builder |
| export\_data\_frm | Export data form |
| nonrepeat\_subform | Holds list of non-repeating items on output\_form |
| open\_form | Main user form – no direct access for users to other forms |
| output\_form | Selection of lists for export to wide or long format |
| output\_long\_form | Build relational tables |
| repeat\_subform | Holds list of repeating items on output\_form |
| select\_item\_wide\_form | Build wide single table |
| select\_sqlserver\_form | Selection of SQL server by name for export |

**Appendix IV VBA module description**

|  |  |
| --- | --- |
| export\_code | Helper code for export to Access |
| file browser | File browser – open source |
| helper code | General helper functions |
| mak\_invis | Used to make Access invisible |
| manage\_meta\_tables | Creates and maintains meta tables |
| run\_java\_exe | Runs the jar file, stored in global fname\_extractor |
| Speed\_up | Working files for table builder |
| sql\_server\_connection | SQL server connection |
| sql\_serevr\_transfer | SQL server export |
| sql\_structures | Handles the SQL building for table and data builder |
| SQlite\_Samples | Next four required for fast SQLite transfer – code built in form event |
| SQLIte4Access |  |
| Test\_SQLiteDatabase-Klasse |  |
| SQLiteDatabase (Class) |  |

**Appendix V Testing**

General test procedure:

1. Import OC XML extraction
2. Select all CRF, all study groups and all item groups (repeating and non-repeating)
3. Build wide (reduce fields if greater than 256) and long tables
4. Export to Access, SQL server and SQLite