Using the JSON Parser (Version 3.3)

Increasingly, data can be downloaded from online sources as **JSON** files.
The file format is very versatile and efficient allowing rapid data transfer.

However, the data then needs to be processed (parsed) before it can be used in **Access**.
Unfortunately, **Access** does not provide any easy method of importing **JSON** files.

By contrast, JSON data can be imported and parsed using **Excel Power Query** add-in (**2010/2013**) or the built-in **Get & Transform** feature in **Excel 2016**

The Access **JSON Parser** has been created to simplify the reading and parsing of **JSON** files into **Access** so the data can then be imported into normalised **Access** tables.

1. Start form

This is shown when the program first loads.

Please note the copyright information on this form.

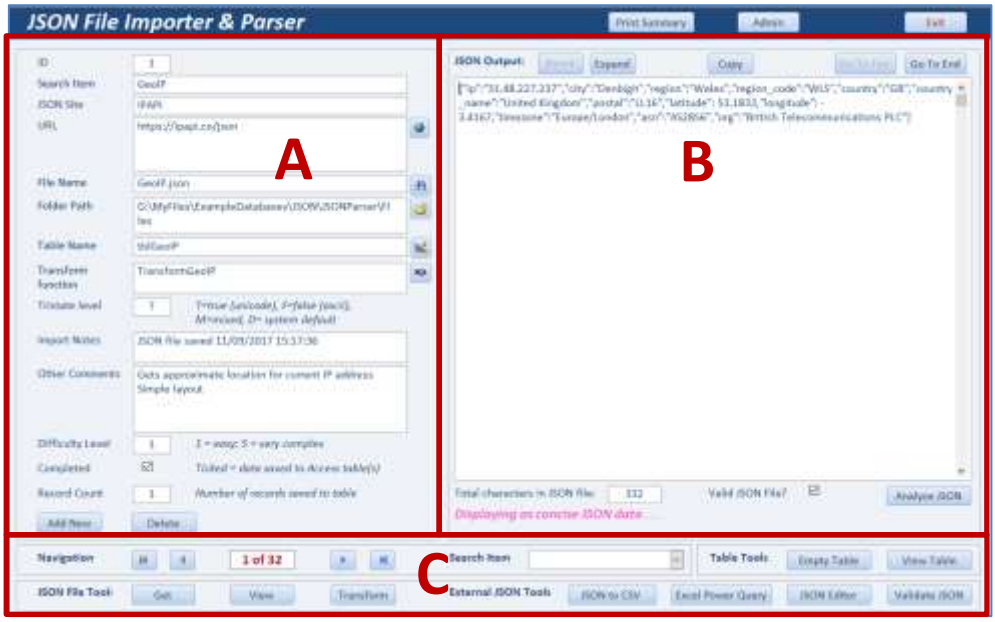
The program makes use of JSON parser code available from https://github.com/VBA-tools/VBA-JSON
Alternative JSON parsers are available from other sources.

If preferred, you can bypass the **startup form** and load the program at the **main form** instead

NOTE: When the program loads it checks its path and updates other folder paths accordingly.
For example, the folders used for **backups and default JSON files** etc

2. *Main form*

This is used for almost all features of the **JSON parser program**



The form controls are located in 3 main sections:

A) Details of JSON file – data source / destination table / transform function
B) JSON file viewer (shrink / expand) plus tools to analyse / fix JSON files
C) Navigation and processing tools

A total of 32 JSON files have been supplied from the very simple to highly complex.
As a guide, files have been assigned a difficulty level on a 1-5 scale: 1 (easy) to 5 (very complex)

i) Processing has been completed for almost all files as examples
ii) Processing of 5 files (24-26,28,31) have been left for end users to work on if they wish to do so

Although part of the parsing process is common to all JSON files, other sections are individualised to match the structure of the source file and the destination table

You can also import your own JSON files into the program. However, it is STRONGLY recommended that you work through the supplied examples before doing so

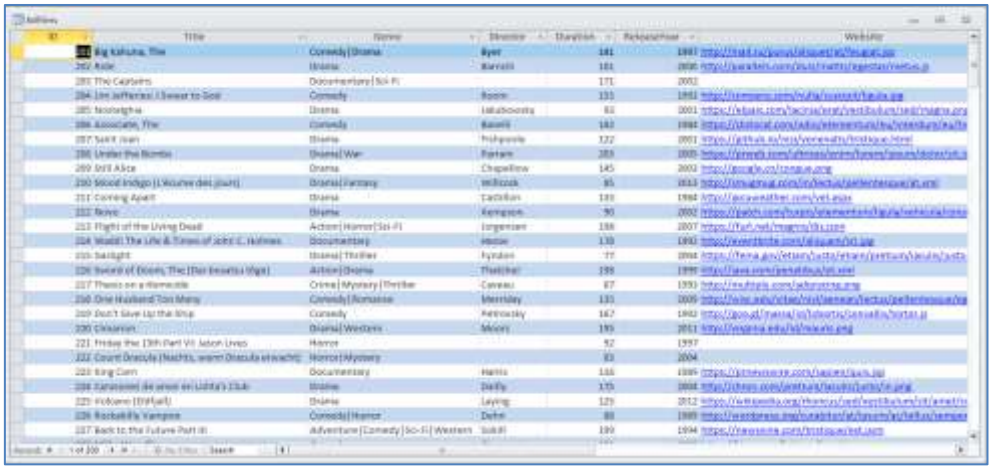
a) *Parse & import an existing file using the transform function*

Select a file using the navigation buttons or the search combo box
The **JSON** file will be 'read' into the window on the right of the form

Click the **Transform** button. The data is parsed and imported into one or more normalised tables.
The process is extremely fast – usually less than a second even for files of several hundred records

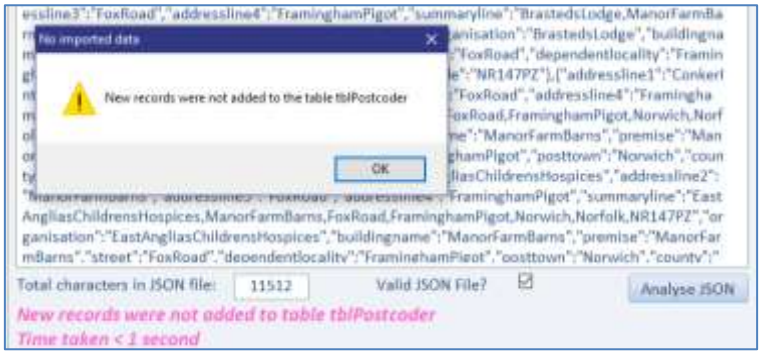
When completed you will see a message similar to this:

Click **Yes** to view the table of imported data:

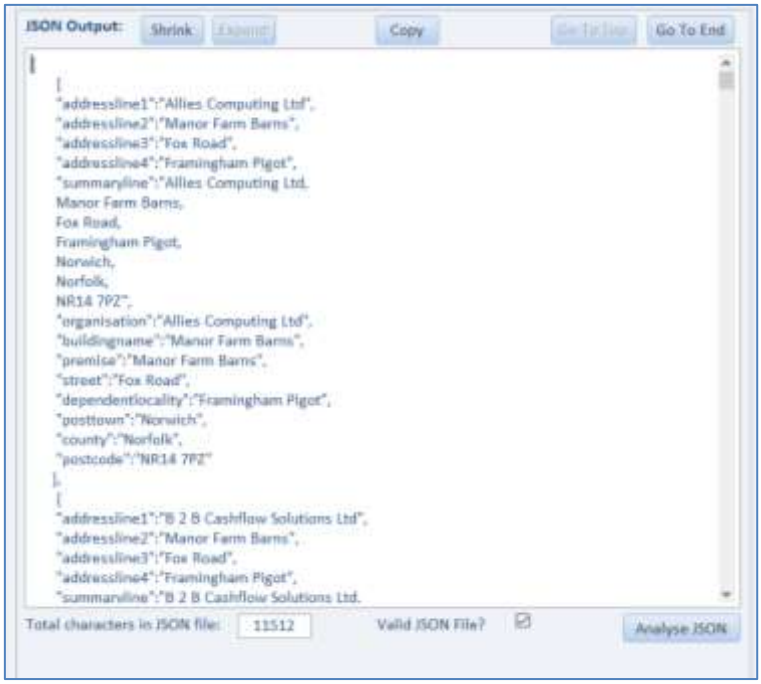


NOTE: before importing new data, existing data is deleted to prevent duplicate records

If the file cannot be processed, a message like this will be shown instead



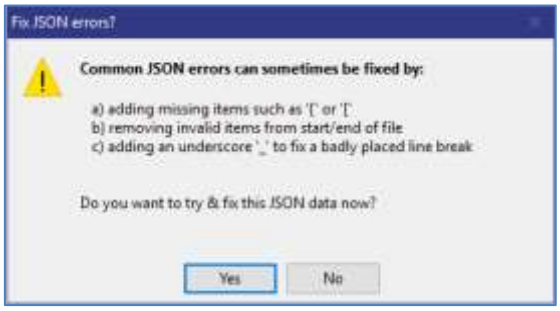
Click the **Expand** button to display the data in a format that is easier to understand



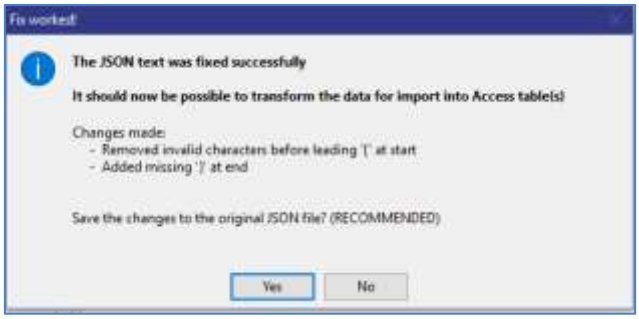
Click the **Shrink** button to return to the default concise format

Occasionally downloaded JSON files may be supplied in an invalid format.
Two examples have been supplied (ID=7 & 31)
Files which fail a validation check are shown in **RED** and the **Transform** button is **disabled**

You can correct many **JSON** errors by clicking the **Fix JSON** button



Click **Yes** to attempt to fix the errors.
If successful, a message like this will be shown and the file text will revert to BLUE



If unsuccessful, the text will remain RED as you will not be able to transform (process) the data. The example Google maps location file (ID 31) cannot be fixed using this method.

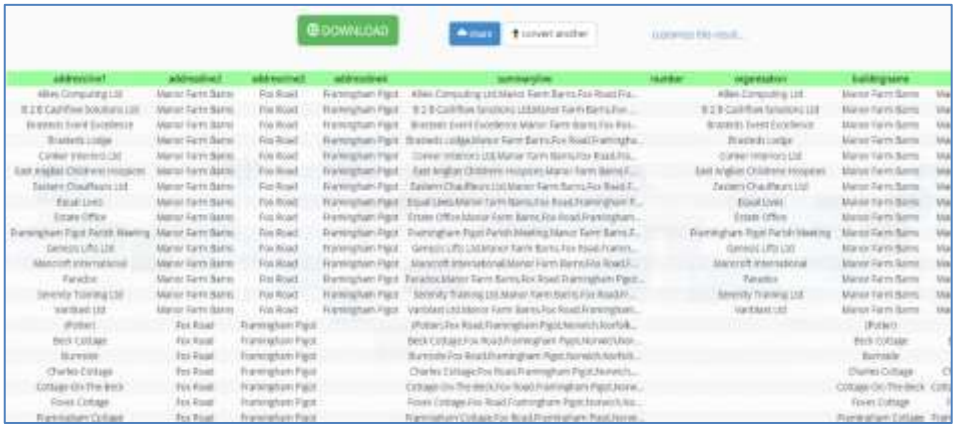
b) External JSON conversion tools

Four other JSON conversion tools are supplied for such cases

i) Online JSON to CSV converter

Click the button – the JSON text is copied to the clipboard
Paste the text into the window that appears

The converted data appears in a new window and can be saved to your computer as a CSV file



ii) Excel Power Query

This message appears – click **OK** to open **Excel** and run **Power Query**

Or click **Help** to view a short video demonstrating how this tool is used

Two further online tools are provided

iii) JSON Editor – assists with editing JSON files into a valid format

iv) Validate JSON - check for errors

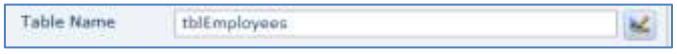
c) Transform other supplied files

Once you have tested several of the prepared **JSON** files and read the code used to process them, you can try to create your own transform functions on some of the other suppled files.

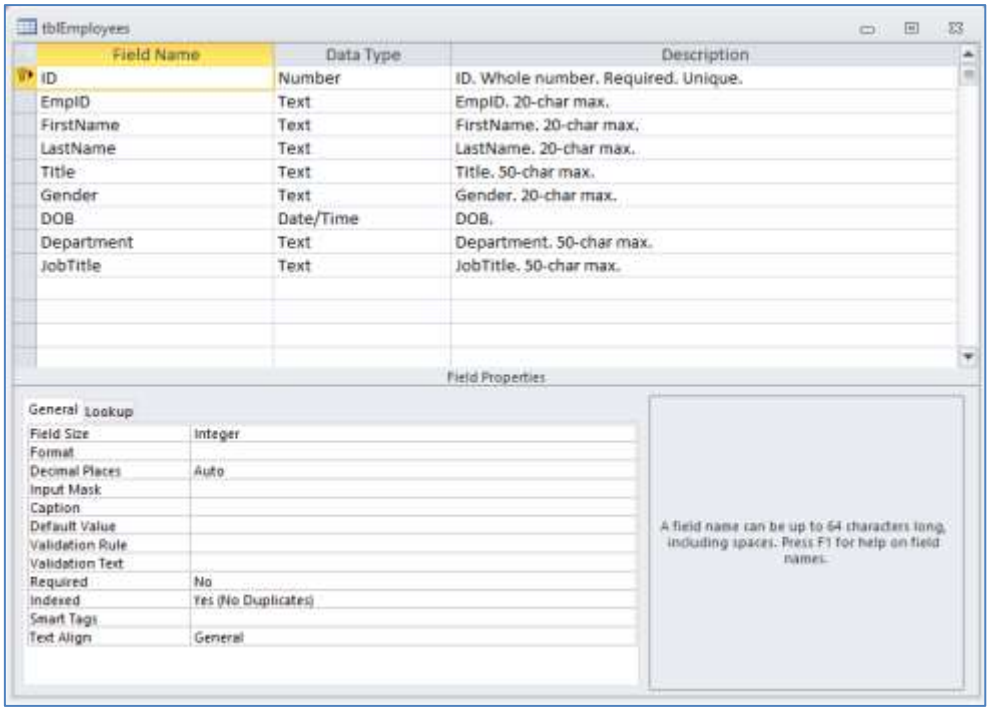
For example, try the **Employees** file (ID = 24)

Your first task is to create the fields for the destination table – **tblEmployees**
This can be done in 2 ways:

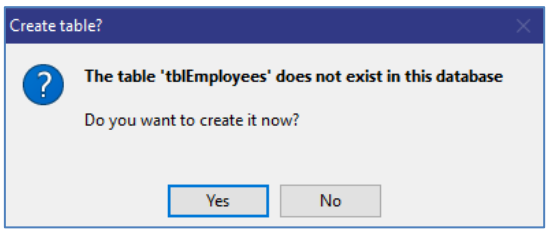
a) Click the small **Design** button to the right of the table name



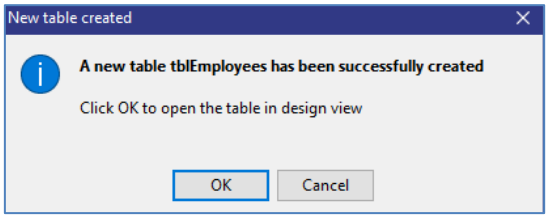
If the table has already been created, it will open in **design view:**



Otherwise a message like this will be shown:



Click **Yes** to create the table.
The file will be analysed and suggested field names / datatypes determined.
The new table will then be created based on this analysis.
This process is very quick and will be followed by this message:



Click **OK** to view the table design
Edit the fields as required:

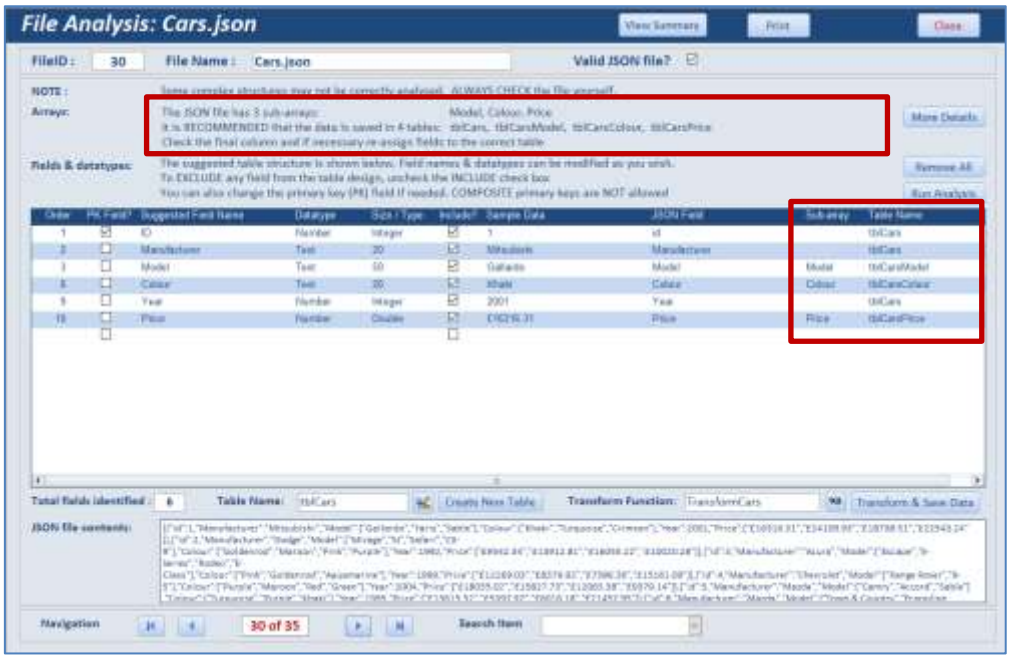
b) If you want more control over the process, click the **Analyse** button
Once again file will be analysed and suggested field names / datatypes determined.
However, this time a file analysis form opens:

This shows details of the suggested field names and datatypes for the destination table.
Normally the analyser will do this perfectly but you should **CHECK** the details carefully
Edit the field names / datatypes if required:

- Check the field size for text fields (1->255) and the Number field type (integer/long/double)
- Untick the **Include** checkbox to exclude any unwanted fields
- Add new fields if necessary – do NOT use spaces or 'special' characters in the field name
- Modify the **primary key** field if the analyser didn't get this correct
  NOTE: composite primary key fields are NOT allowed

Once you are happy with the results, click **Create New Table.**
The new table will then open in **design view** as before

NOTE:
If the JSON file contains subarrays, the analysis screen will suggest ways of splitting the data into 2 or more tables to ensure the tables are fully normalised. For example:



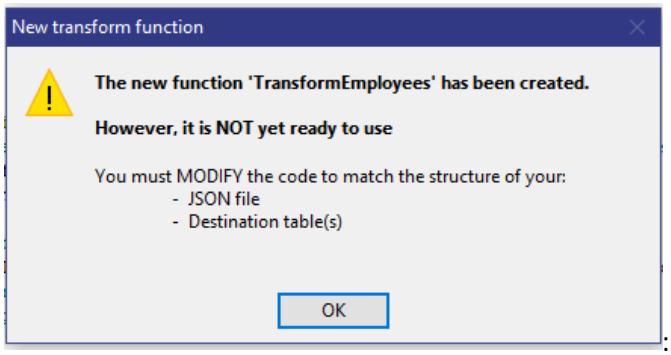
From version 3.3 onwards, all tables are created automatically when you click **Create New Tables**.

The next step is to create a **transform function** which will **parse** the **JSON** file and **save** the data to the table you have created

Click the small **SQL** button to the right of the transform function name

A **draft transform function** is created based on a template file.
The new function opens automatically.

A message like this is shown indicating that further work is required



:

```
Public Function TransformEmployees()

'=============================================================
'Colin Riddington - Mendip Data Systems - 12/09/2017

'TEMPLATE CODE for Transform JSON procedure
'MODIFY the code to match the structure of your JSON file

'When done, add explanatory comments here
'=============================================================

On Error GoTo Err_Handler

'get start time
    Start = Timer

ReadJSON:
    'Read .json file
    'The next line works if Tristate = true (-1). If it is false, replace -1 with 0
    Set JsonTS = fso.OpenTextFile(strFilePath, ForReading, False, -1)
    strJSON = JsonTS.readall

    'Remove unwanted characters - tab & line feed
    strJSON = Replace(Replace(Replace(strJSON, vbTab, ""), vbLf, ""), vbCr, "")
    'Next line - removing space - should usually be omitted
   ' strJSON = Replace(strJSON, " ", "")

    JsonTS.Close

   ' Debug.Print strJSON

ModifyJSON:
    'add this section to enclose the whole JSON string if no overall group is supplied
    'in this example, overall group = "result"
    If InStr(strJSON, "{") = 1 Then
    'add dummy text to make parsing easier
        strJSON = "{""result"":" & strJSON & "}"
       ' Debug.Print strJSON
    End If
```

You will need to modify the recordset section used to import the parsed data

```
    'write to table
    Set db = CurrentDb
    Set rst = db.OpenRecordset("tblEmployees", dbOpenDynaset, dbSeeChanges)

    'Set JSON = JsonConverter.ParseJson(http.responseText)
    Set JSON = modJsonConverter.ParseJson(strJSON)

    'build recordset code
    With rst
        For Each result In JSON("result")
            .AddNew
        '---------------------------------
        'MODIFY THE SECTION BELOW AS NEEDED
            'match json fields to destination table fields
            'IMPORTANT - the fields are case sensitive - JSON fields are usually lower case
          '!ID = result("id") 'autonumber PK field
          ' !CustomerName = result("customername")
          ' !ContactName = result("contactname")
          '  !Address = result("address")
          '  !PostalCode = result("postalcode")
          '  !Country = result("country")
          '---------------------------------
          '  .Update

        Next
        .Close
    End With

    Set rst = Nothing
```

IMPORTANT :
JSON field names are case sensitive and often are lower case

Make sure the details match for the:

- table field e.g. ContactName
- JSON field e.g. contactname

When you are satisfied, return to the main form and click the **Transform** button
Continue to edit the **Transform** function as necessary until you get this working

NOTE:
Creating the transform function will hopefully be fully automated for a later version.
This will be based on the file analysis

d) Add new JSON files

Click the **Add New** button and enter a description in the **Search Item** control.
NOTE: You can use a **wizard** to guide you through the process if preferred

Default values for the file name & folder, table name and transform function are added automatically. You can alter these if necessary.

Enter the name of the **JSON** website and the **URL** if applicable. Click the **Save** button

If using an online **JSON** source, click the **Get** button to save the data to the specified **JSON** file
Otherwise, import the data to the file manually and click the **View** button

The **JSON** data will be read and displayed on the form. This will be very fast (< 1 second).

Analyse the data to get a list of fields then create the **destination table fields** and the **transform function** as previously described.

3. Admin form

This is used to **backup the database** and other admin tasks

I hope you find this **JSON parser** program useful

Please contact me if you have any questions or to report any bugs you find with the program.

Suggestions for improvements or additional features will also be considered for a future version

Colin Riddington
Mendip Data Systems

Email: info@mendipdatasystems.co.uk
Website: www.mendipdatasystems.co.uk