

Using the JSON Parser (Version 3.1)

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.



JSON File Importer & Parser

Author: Colin Rickington, Mendip Data Systems, 11/09/2017
Website: <http://www.mendipdatasystems.co.uk>
Email: info@mendipdatasystems.co.uk
Purpose: Parse JSON files and import into normalised Access tables
JSON Parser: Tim Hall <https://github.com/VBA-tools/VBA-JSON>
Copyright: The code in this utility **MAY** be altered and reused in your own applications **provided the copyright notice is left unchanged** (including Author, Website and Copyright).
You are **NOT** allowed to **sell, resell or repost** this on other sites such as online forums **without permission** from the author.
However, links back to the above websites **ARE** allowed.

This utility is used to parse a variety of **JSON** files then import the normalised data into **Access** tables.
The **JSON** data can be obtained from an **online site** or from a **locally saved file**.

The data is first read into **Access** then parsed using a **transform function**.
If the **JSON** file contains errors, the program will attempt to fix the errors.

As **JSON** files vary in structure from **simple spreadsheet like files** to **complex arrays**, each type of file needs to be treated differently with its own **transform function**. However, there are many similarities between the treatment of different files.

Study the examples provided before adding your own examples.
You will often find an example that is similar in structure to your own file.

NOTE:
BEFORE creating a transform function it is usually helpful to **visualise** the **JSON** data structure using either of these methods:
a) Converting **JSON** to **CSV** using an **online converter**.
b) Using **Excel Power Query** to **parse the data**.
Links are provided for both of these approaches.

See the supplied **Help** file for further information.

Mendip Data Systems 2017-2017 Do NOT show this form in future Version 2.5 11/09/2017

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



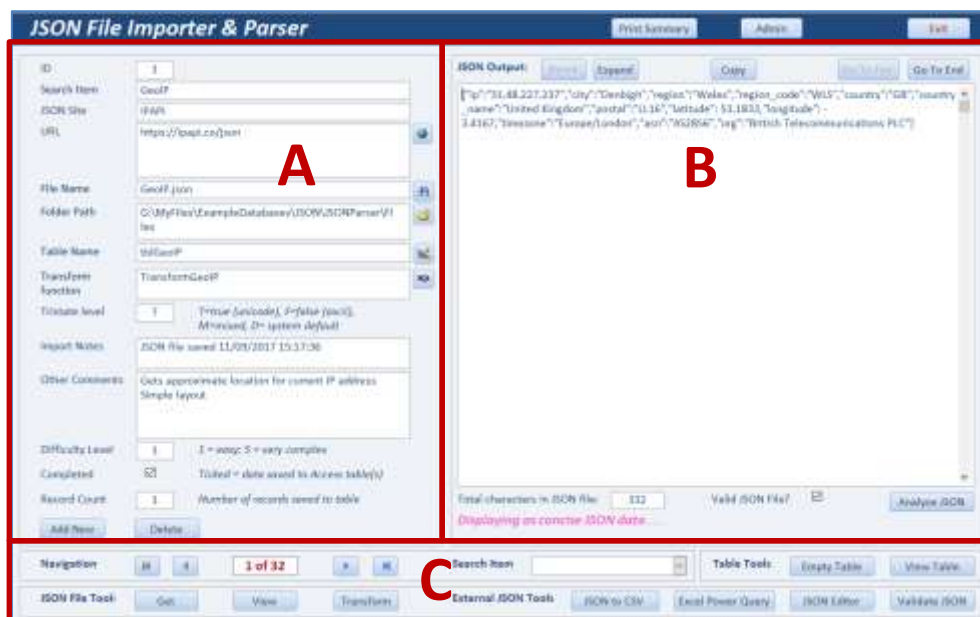
Program Settings

ID	ItemName	ItemValue
1	ProgramFolder	C:\MyFiles\ExampleDatabase\JSON\JSONParser
2	DefaultJSONFolder	C:\MyFiles\ExampleDatabase\JSON\JSONParser\Files
3	BackupFolder	C:\MyFiles\ExampleDatabase\JSON\JSONParser\Backups
4	VBAProjectFolder	C:\MyFiles\ExampleDatabase\JSON\JSONParser\VBAProject
5	ImagesFolder	C:\MyFiles\ExampleDatabase\JSON\JSONParser\Images
6	StartFormLoading	Yes

CARE: Incorrect settings may prevent this program working correctly.

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 done for the first 22 files as examples
- ii) Processing of files 23-32 have been left as practice files for end users

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

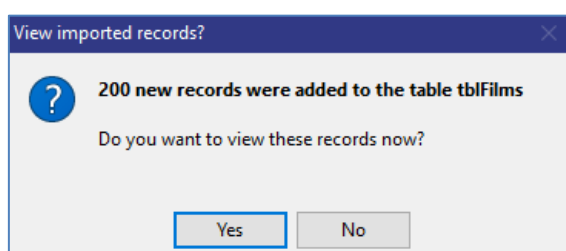
You can also import your own JSON files into the program. However, it is **STRONGLY** recommended that you work through the supplied examples before doing so

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:

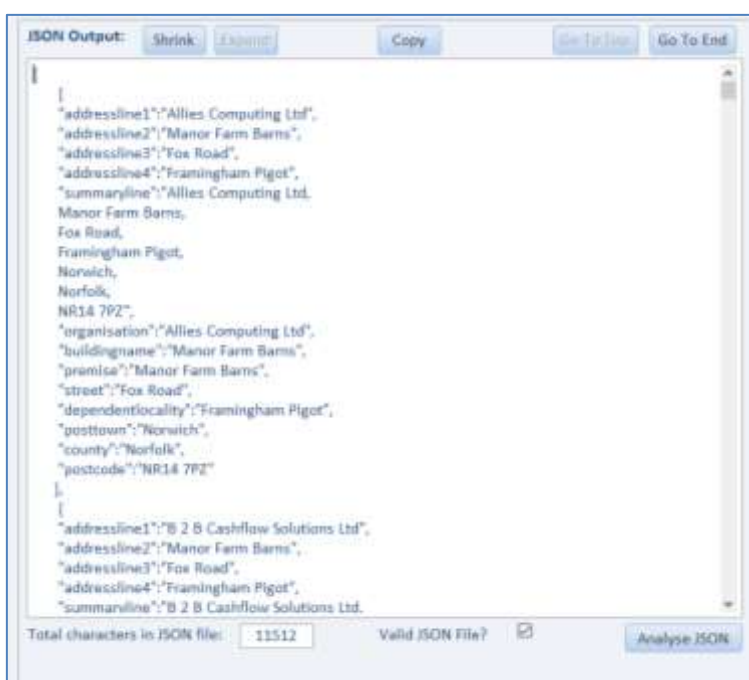
ID	Title	Genre	Director	Year	Rating	Website
101	Big Kahuna, The	Comedy (Drama)	Wesley	1997	1.41	http://www.bigkahuna.com/
102	Alien	Drama	James	1986	1.41	http://www.alien.com/
103	The Captains	Documentary (Sci-Fi)	James	1971	1.71	http://www.captains.com/
104	Line of Honor	Drama	James	1991	1.31	http://www.lineofhonor.com/
105	Nostalgia	Drama	James	2001	1.31	http://www.nostalgia.com/
106	Associate, The	Comedy	James	1998	1.31	http://www.associate.com/
107	Surreal	Drama	James	2001	1.31	http://www.surreal.com/
108	Under the Bombs	Drama (War)	James	2001	1.31	http://www.underthebombs.com/
109	DIT	Drama	James	2001	1.31	http://www.dit.com/
110	Secret Bridge	Drama (Drama)	James	1998	1.31	http://www.secretbridge.com/
111	Cooking	Drama	James	1998	1.31	http://www.cooking.com/
112	Boris	Drama	James	2001	1.31	http://www.boris.com/
113	Flight of the Living Dead	Drama (Horror/Sci-Fi)	James	2007	1.31	http://www.flightofthelivingdead.com/
114	The Life & Times of John L. Holmes	Documentary	James	1991	1.31	http://www.lifeandtimes.com/
115	Night	Drama (Thriller)	James	1991	1.31	http://www.night.com/
116	Secret of Doors	Drama (Drama)	James	1991	1.31	http://www.secretofdoors.com/
117	Photo on a Mosaic	Drama (Mystery/Thriller)	James	1991	1.31	http://www.photoonamosaic.com/
118	One Husband Too Many	Comedy (Romance)	James	1991	1.31	http://www.onehusbandtoomany.com/
119	Don't Give Up the Ship	Comedy	James	1991	1.31	http://www.dontgiveuptheship.com/
120	Chicken	Drama (Western)	James	1991	1.31	http://www.chicken.com/
121	Inside the 13th Part of John Lives	Drama	James	1991	1.31	http://www.insidethe13thpartofjohnlives.com/
122	Great British	Drama (Mystery)	James	1991	1.31	http://www.greatbritish.com/
123	King Cam	Documentary	James	1991	1.31	http://www.kingcam.com/
124	Caravan	Drama	James	1991	1.31	http://www.caravan.com/
125	Indiana	Drama	James	1991	1.31	http://www.indiana.com/
126	Rokkiki	Drama (Horror)	James	1991	1.31	http://www.rokkiki.com/
127	Back to the Future	Adventure (Comedy/Sci-Fi)	James	1991	1.31	http://www.backtothefuture.com/

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

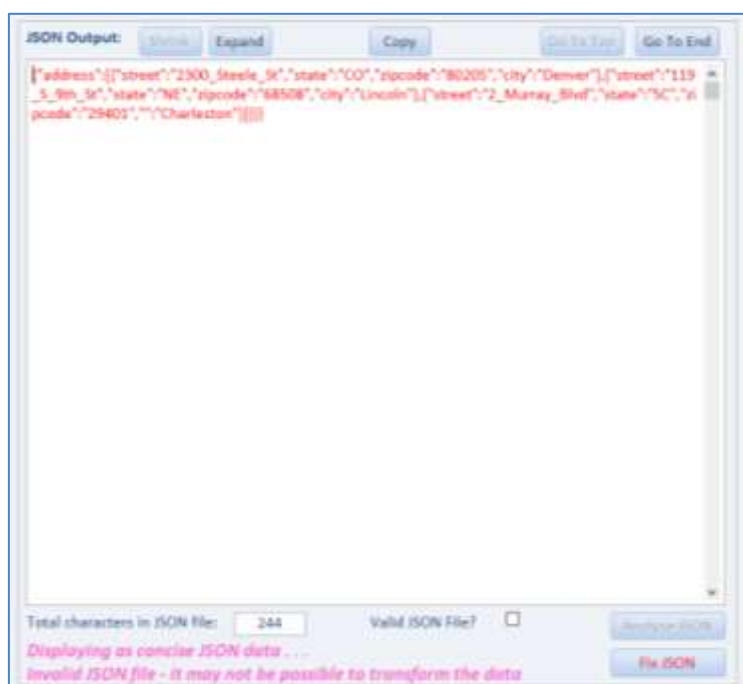


Click the **Expand** button to display the data in a format that is easier to understand

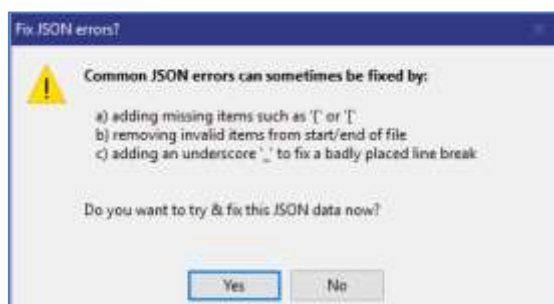


Click the **Shrink** button to return to the default concise format

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**

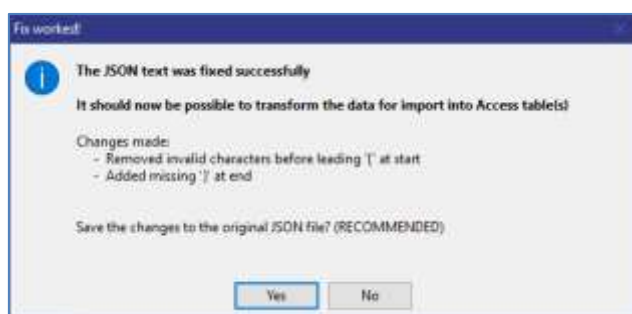


You can correct many **JSON** errors by clicking the **Fix JSON** button



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



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 supplied files.

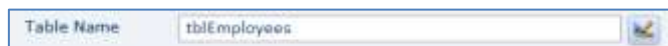
For example, try the **Employees** file (ID = 24)



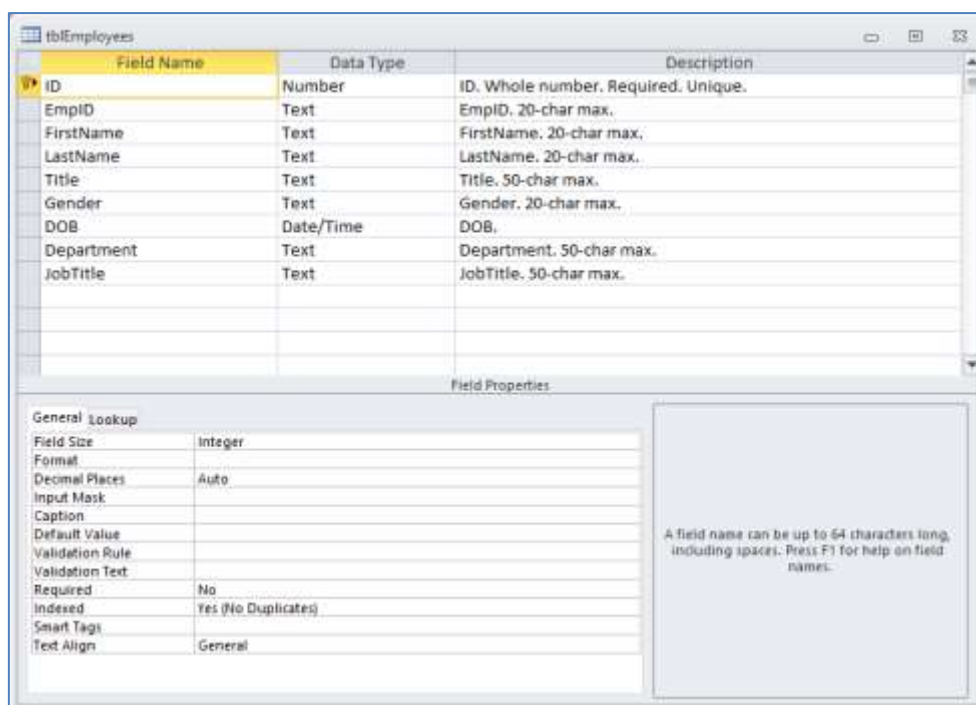
Your first task is to create the fields for the destination table – **tblEmployees**

This can be done in 2 ways:

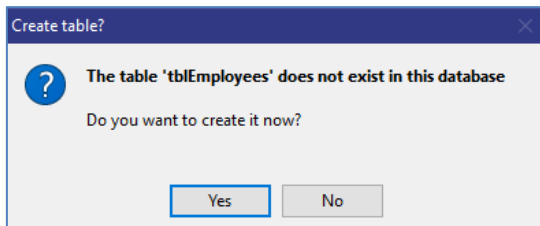
- Click the small **Design** button to the right of the table name



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



Otherwise a message like this will be shown:

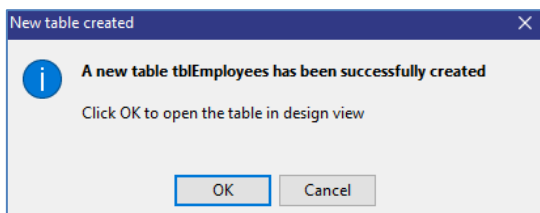


Click **Yes** to create the table.

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

The new table will then open in **design view** as before

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:

This will be addressed in a subsequent version of the program.

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

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 (T). 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, ""), vbCrLf, ""), vbCrLf, "")
'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 test 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

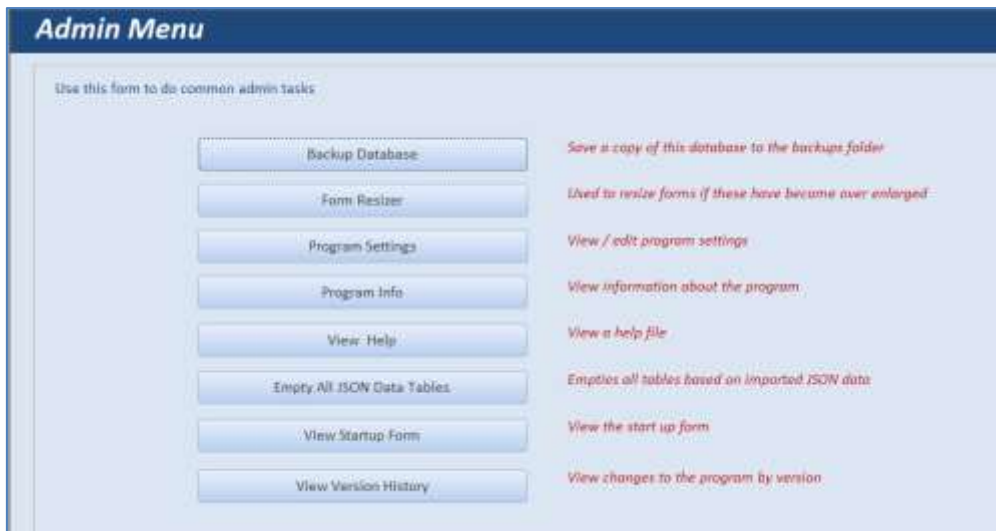


The **JSON** data will be read and displayed on the form. This will be very fast (< 1 second).

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