

***Mendip Data Systems***

***JSON Analyse  
and Transform  
for Access  
(JATFA)***

***User Guide***

***Version 4.20***

# JSON Analyse & Transform for Access (Version 4.20)

Increasingly, data is being made available from online sources as **JSON** files. The file format is very versatile and efficient allowing rapid data transfer. See [Appendix A](#) for more information about **JSON** file structure.

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**.

**JSON Analyse & Transform for Access (JATFA)** has been created to simplify the processing of **JSON** files directly into **Access** so the data can be saved in normalised **Access** tables.

## 1. Installation

The **JSON Analyse & Transform for Access** application has been designed to run in **Access 2010** or later. It is compatible with both 32-bit and 64-bit **Access**.

The application has **not** been fully tested in **Access 2007**.

The application will be installed to the folder **C:\Programs\MendipDataSystems\JATFA**. This location will be made a trusted location. Shortcuts will be created on the desktop and start menu.

If you want to remove the application at a later date, do so using **Add/Remove Programs**.

During installation, you will be asked to provide user information.

This will be stored in the registry key together with information about the application:  
**HKEY\_CURRENT\_USER\Software\Mendip Data Systems\ JATFA**

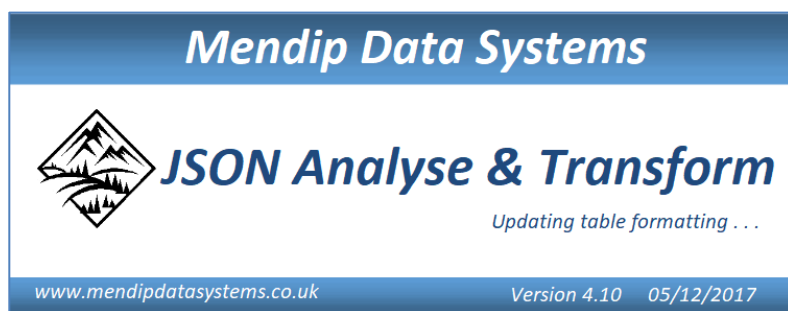
### NOTE:

At a minimum, you must enter the user name. Other items are optional. If you bypass this information during installation, it can be added from the application itself.

## 2. Initial Setup

Each time the application runs, a splash form appears briefly.

Various checks are run whilst the form is visible. It then closes automatically.



When the application is run for the first time, a user information screen will be shown:

**New User Information** Close

**Congratulations**  
You have successfully installed JSON Parser version 4.05

Please check / complete all items below  
Details entered during program installation are shown where available.

Registered User Name

Company

Address

Telephone

EMail

Licence Key

Support Expiry Date

Information entered during installation will be retrieved from the registry. If you left any of these fields blank, please enter the information now.

The **licence key and support expiry date** are entered automatically  
Support is available for 90 days after installation

When the user information form is closed, this message appears:

**Email settings not done**

**?** Email has not yet been setup for this application.  
Do you want to enter email settings now?

If you click Yes, the **Email Settings** form opens:

**EMail Settings** Close Clear EMail Settings

Use this form to test settings used to send emails from this program

Use Outlook for Email  Clear EMail Settings

**Settings:** *NOTE: The items below are NOT required if Outlook is used to send email from the program*

Send Mail Using Method  1 = local; 2 = network

Port used to send email  Usually 25

Email server  e.g. "smtp.gmail.com"

SMTP Authenticate  Usually 1

UserName  Default user email for email from this program

Password  Password for default user email

Timeout (seconds)  e.g. 60

Use SSL  True / False

**Test email:**  
Fill in the details below then click the 'Send Email' button to try & send a test email message with an attached file.  
Click the 'Save Settings' button if the email is sent successfully.  
Click the "Help" button for more information if an error message is shown.  
Click the "Clear" button to remove test email data from this section.

Recipient email address

Attachment filename (optional)  Browse  
This must be the full file path of the attachment

**Default email addresses:** *The following are used by various features in the program to send emails automatically*

Program Support EMail

Registered User EMail

**Email** is used to send feedback to customer support  
Email messages can be sent direct from the application or using **Outlook**.

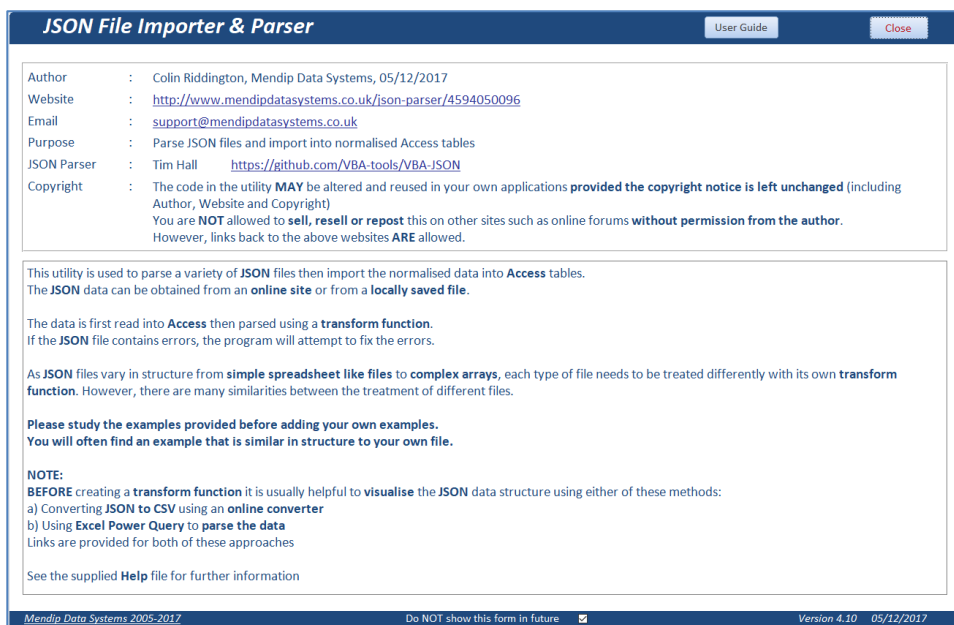
Complete this form so that email can be sent direct from the application  
Send a test email to check this is working correctly.  
If errors occur, click the **Help** button for more information

If the email is sent successfully, click **Save Settings**.  
NOTE: The settings are **NOT** saved automatically

Close the email settings form.  
The default **Start form** now opens

### 3. Start form

This is the default form shown when the application first loads.  
It contains a summary of the program together with copyright information

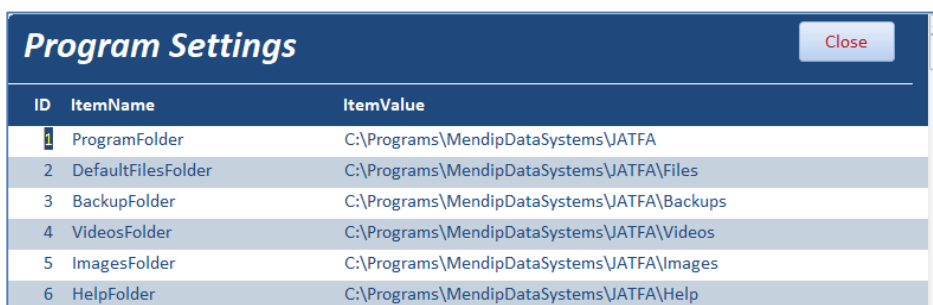


Please note the copyright information on this form.

The application uses 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 application at the **main form** instead

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



## 4. Main form

This is used for many features of the **JSON Analyse & Transform for Access (JATFA)** application

The screenshot shows the 'JSON File Importer & Parser' application. Section A (left) includes fields for FileID (1), Search Item (GeoIP), Data Source (IPAPI), URL (https://ipapi.co/json), File Name (GeoIP.json), Folder Path (C:\Programs\MendipDataSystems\JSONParser\Files), Table Name(s) (tblGeoIP), Transform function (TransformGeoIP), Tristate level (T), Import Notes (JSON file saved 23/10/2017 19:58:08), and Comments / Hints (Simple array - straightforward to parse). Section B (right) shows the JSON Output window with a sample record: [{"ip": "86.154.3.26", "city": "Bude", "region": "England", "region\_code": "ENG", "country": "GB", "country\_name": "United Kingdom", "postal": "EX23", "latitude": "50.8243", "longitude": "4.5413", "timezone": "Europe/London", "asn": "AS2856", "org": "British Telecommunications PLC"}]. Section C (bottom) includes navigation buttons (1 of 46), search item, table tools (Empty Table, View Table), and JSON file tools (Get, View, Transform, External JSON Tools, JSON to CSV, Excel Power Query, JSON Editor, Validate JSON).

The form controls can be considered in 3 main sections:

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

Over 65 JSON files have been supplied from the very simple to highly complex.

As a rough guide, files have been assigned a difficulty level on a 1-5 scale: 1 (easy) to 5 (very complex)

Processing has been completed for all files as examples

More files are also available for which processing has not been done

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

You can also import your own **JSON** files into the application. However, it is **STRONGLY** recommended that you work through a variety of 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	Duration	ReleaseYear	Website
201	Big Kahuna, The	Comedy Drama	Byer	141	1997	<a href="http://mail.ru/purus/aliquet/at/teugiat.jsp">http://mail.ru/purus/aliquet/at/teugiat.jsp</a>
202	Ride	Drama	Barnish	101	2000	<a href="http://parallels.com/duis/mattis/egestas/metus.is">http://parallels.com/duis/mattis/egestas/metus.is</a>
203	The Captains	Documentary Sci-Fi		171	2002	
204	Jim Jefferies: I Swear to God	Comedy		153	1993	<a href="http://comsenz.com/nulla/suscipit/ligula.jpg">http://comsenz.com/nulla/suscipit/ligula.jpg</a>
205	Nostalgia	Drama	Jakubowsky	93	2001	<a href="https://elpais.com/lacmia/erat/vestibulum/sed/magna.png">https://elpais.com/lacmia/erat/vestibulum/sed/magna.png</a>
206	Associate, The	Comedy	Baselli	182	1984	<a href="https://cbslocal.com/odio/elementum/eu/interdum/eu/tr">https://cbslocal.com/odio/elementum/eu/interdum/eu/tr</a>
207	Saint Joan	Drama	Fishpoole	122	2001	<a href="https://github.io/nisi/venenatis/tristique.html">https://github.io/nisi/venenatis/tristique.html</a>
208	Under the Bombs	Drama War	Parram	203	2005	<a href="https://prweb.com/ultrices/enim/lorem/ipsum/dolor/sit.js">https://prweb.com/ultrices/enim/lorem/ipsum/dolor/sit.js</a>
209	Still Alice	Drama	Chapellow	145	2003	<a href="http://google.cn/congue.png">http://google.cn/congue.png</a>
210	Mood Indigo (L'écume des jours)	Drama Fantasy	Willcock	85	2013	<a href="http://smugug.com/in/lectus/pellentesque/at.xml">http://smugug.com/in/lectus/pellentesque/at.xml</a>
211	Coming Apart	Drama	Castillon	133	1984	<a href="http://accuweather.com/vel.aspx">http://accuweather.com/vel.aspx</a>
212	Novo	Drama	Kempson	90	2002	<a href="https://patch.com/turpis/elementum/ligula/vehicula/cons">https://patch.com/turpis/elementum/ligula/vehicula/cons</a>
213	Flight of the Living Dead	Action Horror Sci-Fi	Jorgensen	198	2007	<a href="https://furl.net/magnis/dis.json">https://furl.net/magnis/dis.json</a>
214	Wadd: The Life & Times of John C. Holmes	Documentary	Hesse	178	1992	<a href="http://eventbrite.com/aliquam/sit.jpg">http://eventbrite.com/aliquam/sit.jpg</a>
215	Gaslight	Drama Thriller	Fyndon	77	2004	<a href="https://hema.gov/etiam/justo/etiam/premium/iaculis/justo">https://hema.gov/etiam/justo/etiam/premium/iaculis/justo</a>
216	Sword of Doom, The (Dai-bosatsu tōge)	Action Drama	Thatcher	198	1999	<a href="http://iava.com/penatibus/et.xml">http://iava.com/penatibus/et.xml</a>
217	Thesis on a Homicide	Crime Mystery Thriller	Caveau	87	1993	<a href="http://multiply.com/adipiscing.png">http://multiply.com/adipiscing.png</a>
218	One Husband Too Many	Comedy Romance	Merriday	135	2009	<a href="http://wisc.edu/vitae/nisi/aenean/lectus/pellentesque/eg">http://wisc.edu/vitae/nisi/aenean/lectus/pellentesque/eg</a>
219	Don't Give Up the Ship	Comedy	Petrovsky	167	1992	<a href="http://goo.gl/massa/id/lobortis/convallis/tortor.js">http://goo.gl/massa/id/lobortis/convallis/tortor.js</a>
220	Cimarron	Drama Western	Moors	195	2011	<a href="http://virginia.edu/id/mauris.png">http://virginia.edu/id/mauris.png</a>
221	Friday the 13th Part VI: Jason Lives	Horror		92	1997	
222	Count Dracula (Nachts, wenn Dracula erwacht)	Horror Mystery		83	2004	
223	King Corn	Documentary	Harris	116	1989	<a href="https://prnewswire.com/sapient/quis.jsp">https://prnewswire.com/sapient/quis.jsp</a>
224	Canciones de amor en Lolita's Club	Drama	Dailly	175	2004	<a href="http://chron.com/premium/iaculis/justo/in.png">http://chron.com/premium/iaculis/justo/in.png</a>
225	Volcano (Eldfjall)	Drama	Laying	125	2012	<a href="https://wikipedia.org/rhoncus/sed/vestibulum/sit/amet/c">https://wikipedia.org/rhoncus/sed/vestibulum/sit/amet/c</a>
226	Rockabilly Vampire	Comedy Horror	Dehn	88	1989	<a href="http://wordpress.org/curabitur/at/ipsam/ac/tellus/semper">http://wordpress.org/curabitur/at/ipsam/ac/tellus/semper</a>
227	Back to the Future Part III	Adventure Comedy Sci-Fi Western	Sokill	199	1994	<a href="https://newsvine.com/tristique/est.json">https://newsvine.com/tristique/est.json</a>

NOTE: before importing new data, by default existing data is deleted to prevent duplication of records

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

**No imported data**

New records were not added to the table tblPostcoder

OK

Total characters in JSON file:  Valid JSON File?  Analyse JSON

*New records were not added to table tblPostcoder*

*Time taken < 1 second*

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

**JSON Output:** Shrink **Expand** Copy Edit JSON Go To Top Go To End

```

{
  "addressLine1": "Allies Computing Ltd",
  "addressLine2": "Manor Farm Barns",
  "addressLine3": "Fox Road",
  "addressLine4": "Framingham Pigot",
  "summaryLine": "Allies Computing Ltd, Manor Farm Barns, Fox Road, Framingham Pigot, Norwich, Norfolk, NR14 7PZ",
  "organisation": "Allies Computing Ltd",
  "buildingName": "Manor Farm Barns",
  "premise": "Manor Farm Barns",
  "street": "Fox Road",
  "dependentLocality": "Framingham Pigot",
  "posttown": "Norwich",
  "county": "Norfolk",
  "postcode": "NR14 7PZ"
},
{
  "addressLine1": "B 2 B Cashflow Solutions Ltd",
  "addressLine2": "Manor Farm Barns",
  "addressLine3": "Fox Road",
  "addressLine4": "Framingham Pigot",
  "summaryLine": "B 2 B Cashflow Solutions Ltd, Manor Farm Barns, Fox Road, Framingham Pigot, Norwich, Norfolk, NR14 7PZ",
  "organisation": "B 2 B Cashflow Solutions Ltd",
  "buildingName": "Manor Farm Barns",
  "premise": "Manor Farm Barns",
  "street": "Fox Road",
  "dependentLocality": "Framingham Pigot",
  "posttown": "Norwich",
  "county": "Norfolk",
  "postcode": "NR14 7PZ"
}

```

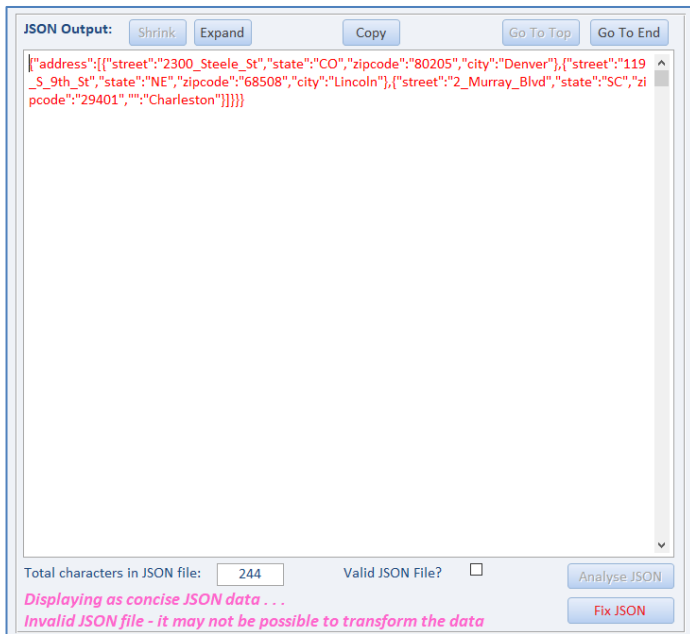
Total characters in JSON file:  Valid JSON File?  Special Case?  Analyse JSON

*Displaying as expanded JSON data . . .*

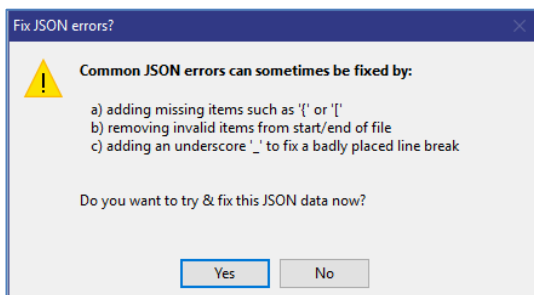
Click the **Shrink** button to return to the default concise format

Click the **Edit JSON** button to view/edit the file in the default application e.g. Notepad

Occasionally, downloaded JSON files may be supplied in an invalid format. Files which fail a validation check are shown in **RED** and the **Transform** button is **disabled**

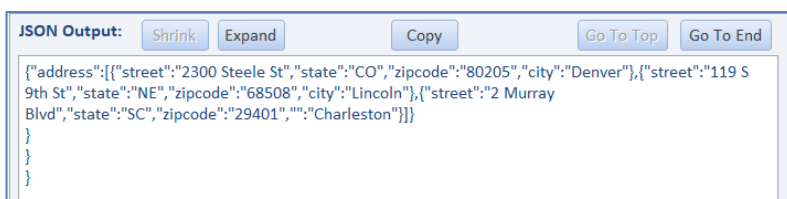
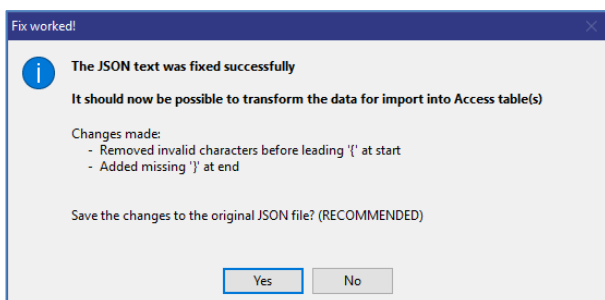


You can correct many **JSON** errors by clicking the **Fix JSON** button



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 original **GoogleMaps** file (in the **More Files** folder) is a file that 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 (or upload it from the file)

The screenshot shows the 'JSON to CSV Converter' website. At the top, it says 'How to convert JSON to CSV' followed by four numbered instructions: 1. Upload your JSON text, file or URL using the online converter below. 2. Press the cog button on the right for advanced settings (optional). 3. Download the resulting CSV file when prompted. 4. Open your CSV file. It is useful to view CSV data in a spreadsheet such as Excel or Open Office. Below the instructions is a large text area with the placeholder text 'Paste your JSON text (or URL) here'. To the right of this area is a gear icon for settings. Below the text area are two buttons: 'Upload (.zip / .json) up to 1 MB' and 'Upgrade to PRO and convert up to 50 MB or log in'.

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

The screenshot shows a window with a table of converted data. At the top, there are buttons for 'DOWNLOAD', 'share', 'convert another', and 'customize this result...'. The table has the following columns: addressline1, addressline2, addressline3, addressline4, summaryline, number, organisation, and buildingname. The data rows contain various company names and addresses, such as 'Allies Computing Ltd', 'B 2 B Cashflow Solutions Ltd', 'Brasteds Event Excellence', etc.

addressline1	addressline2	addressline3	addressline4	summaryline	number	organisation	buildingname
Allies Computing Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Allies Computing Ltd,Manor Farm Barns,Fox Road,Fra...		Allies Computing Ltd	Manor Farm Barns
B 2 B Cashflow Solutions Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	B 2 B Cashflow Solutions Ltd,Manor Farm Barns,Fox ...		B 2 B Cashflow Solutions Ltd	Manor Farm Barns
Brasteds Event Excellence	Manor Farm Barns	Fox Road	Framingham Pigot	Brasteds Event Excellence,Manor Farm Barns,Fox Roa...		Brasteds Event Excellence	Manor Farm Barns
Brasteds Lodge	Manor Farm Barns	Fox Road	Framingham Pigot	Brasteds Lodge,Manor Farm Barns,Fox Road,Framingha...		Brasteds Lodge	Manor Farm Barns
Conker Interiors Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Conker Interiors Ltd,Manor Farm Barns,Fox Road,Fra...		Conker Interiors Ltd	Manor Farm Barns
East Anglias Childrens Hospices	Manor Farm Barns	Fox Road	Framingham Pigot	East Anglias Childrens Hospices,Manor Farm Barns,F...		East Anglias Childrens Hospices	Manor Farm Barns
Eastern Chauffeurs Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Eastern Chauffeurs Ltd,Manor Farm Barns,Fox Road,F...		Eastern Chauffeurs Ltd	Manor Farm Barns
Equal Lives	Manor Farm Barns	Fox Road	Framingham Pigot	Equal Lives,Manor Farm Barns,Fox Road,Framingham P...		Equal Lives	Manor Farm Barns
Estate Office	Manor Farm Barns	Fox Road	Framingham Pigot	Estate Office,Manor Farm Barns,Fox Road,Framingham...		Estate Office	Manor Farm Barns
Framingham Pigot Parish Meeting	Manor Farm Barns	Fox Road	Framingham Pigot	Framingham Pigot Parish Meeting,Manor Farm Barns,F...		Framingham Pigot Parish Meeting	Manor Farm Barns
Genesis Lifts Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Genesis Lifts Ltd,Manor Farm Barns,Fox Road,Framin...		Genesis Lifts Ltd	Manor Farm Barns
Mancroft International	Manor Farm Barns	Fox Road	Framingham Pigot	Mancroft International,Manor Farm Barns,Fox Road,F...		Mancroft International	Manor Farm Barns
Paradox	Manor Farm Barns	Fox Road	Framingham Pigot	Paradox,Manor Farm Barns,Fox Road,Framingham Pigot...		Paradox	Manor Farm Barns
Serenity Training Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Serenity Training Ltd,Manor Farm Barns,Fox Road,Fr...		Serenity Training Ltd	Manor Farm Barns
Variblast Ltd	Manor Farm Barns	Fox Road	Framingham Pigot	Variblast Ltd,Manor Farm Barns,Fox Road,Framingham...		Variblast Ltd	Manor Farm Barns
(Potter)	Fox Road	Framingham Pigot	Framingham Pigot	(Potter),Fox Road,Framingham Pigot,Norwich,Norfolk...		(Potter)	Manor Farm Barns
Beck Cottage	Fox Road	Framingham Pigot	Framingham Pigot	Beck Cottage,Fox Road,Framingham Pigot,Norwich,Nor...		Beck Cottage	Manor Farm Barns
Burnside	Fox Road	Framingham Pigot	Framingham Pigot	Burnside,Fox Road,Framingham Pigot,Norwich,Norfolk...		Burnside	Manor Farm Barns
Charles Cottage	Fox Road	Framingham Pigot	Framingham Pigot	Charles Cottage,Fox Road,Framingham Pigot,Norwich...		Charles Cottage	Manor Farm Barns
Cottage-On-The-Beck	Fox Road	Framingham Pigot	Framingham Pigot	Cottage-On-The-Beck,Fox Road,Framingham Pigot,Norw...		Cottage-On-The-Beck	Manor Farm Barns
Foxes Cottage	Fox Road	Framingham Pigot	Framingham Pigot	Foxes Cottage,Fox Road,Framingham Pigot,Norwich,No...		Foxes Cottage	Manor Farm Barns
Framingham Cottage	Fox Road	Framingham Pigot	Framingham Pigot	Framingham Cottage,Fox Road,Framingham Pigot,Norw...		Framingham Cottage	Manor Farm Barns

### ii) Excel Power Query

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

The screenshot shows a dialog box titled 'How to run Power Query'. It contains the following text: 'When you click **OK**, **Excel** will open with a blank workbook.' Below this, it says 'To open your **JSON** file in **Excel Power Query**:' followed by three numbered steps: 1. Click the **Data** tab, 2. Click **Get Data ...From File ... From JSON**, 3. Select the **JSON** file and click '**Open**'. Below the steps, it says 'The query editor will open with your file loaded ready for parsing'. At the bottom, there is a red text prompt: 'For a short demo video (18 sec), click the **Help** button'. At the very bottom, there are three buttons: 'Help', 'Cancel', and 'OK'.

Or click **Help** to view a short video (18 seconds) demonstrating how this tool is used

NOTE:

For **Access 2010/2013**, you must first install the **Excel Power Query** add-in

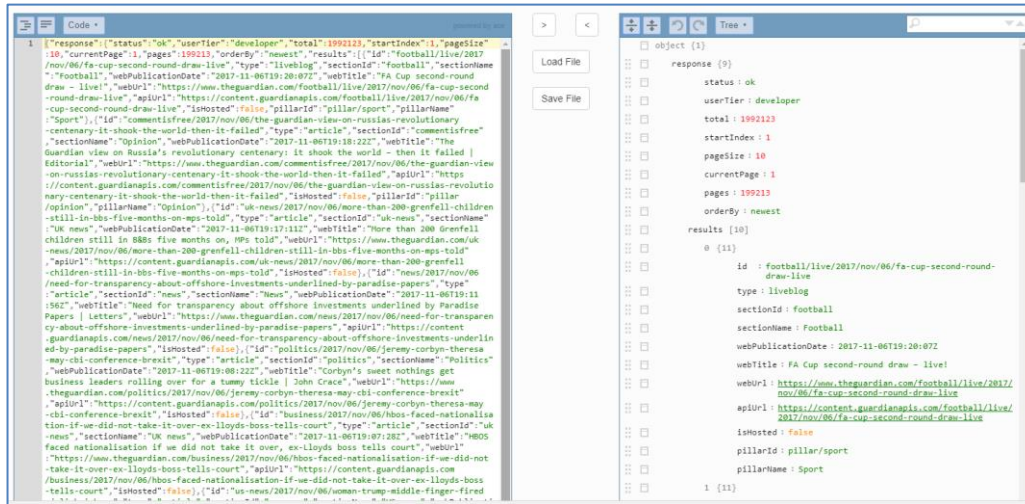


### iii) JSON Editor

This is useful to help edit JSON files into a valid format

Upload your file or paste into the left window.

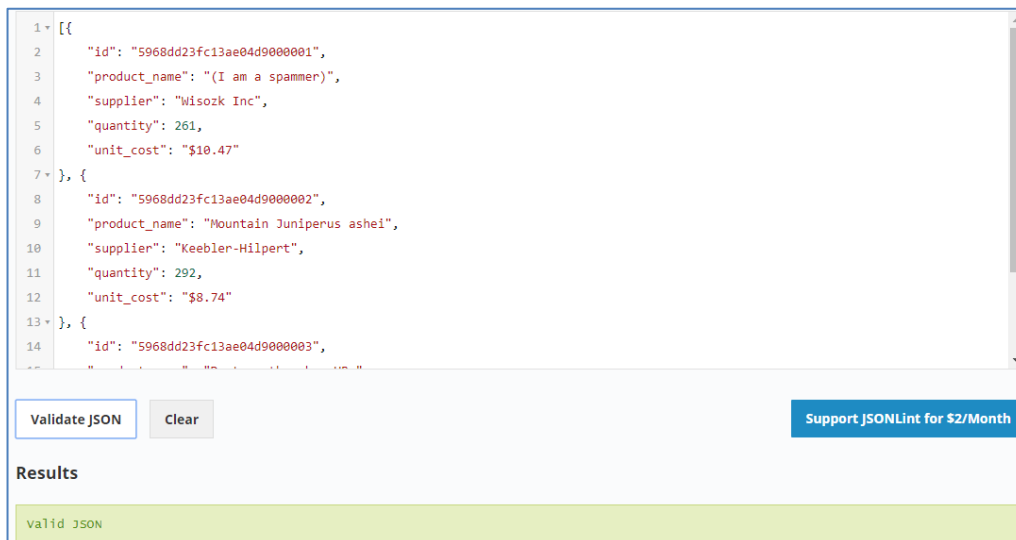
The right window allows you to view each section of the file in a tree-view structure



### iv) Validate JSON

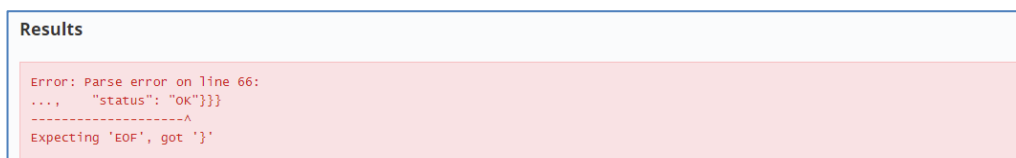
Use to check for errors in the JSON file

Upload your file or paste into the window then click **Validate JSON**



If the file is valid, the above message will be shown in **GREEN**

If there are errors, these will be shown in the Results section in **RED**



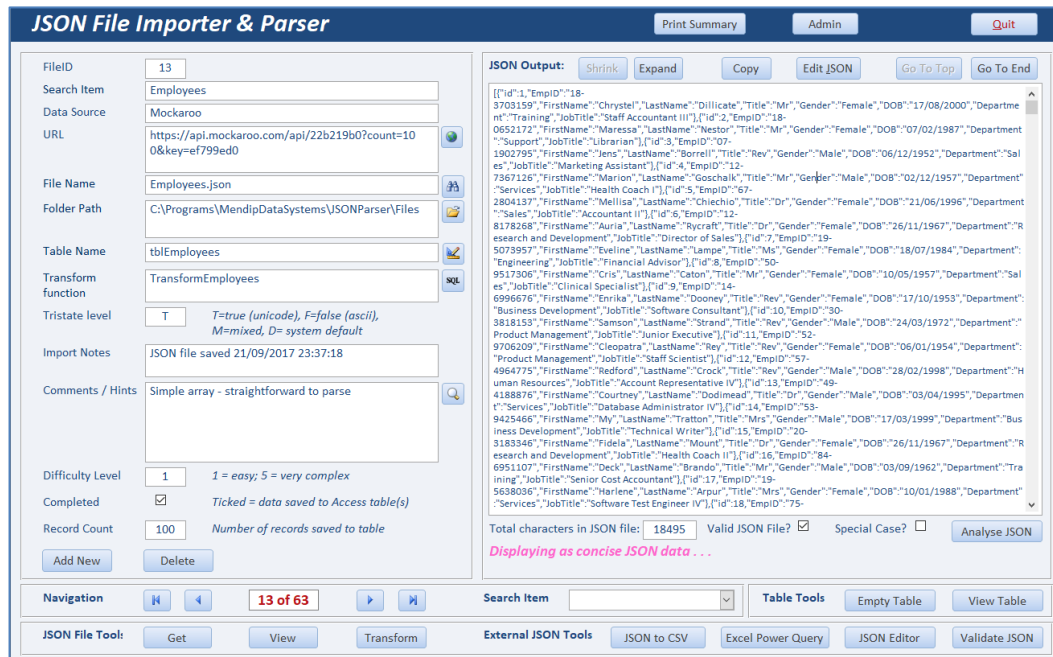
Use this tool to work through and correct the errors.

Once you have a valid file, you can copy the corrected version & overwrite the original file

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.

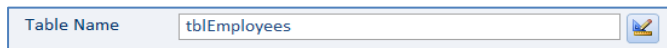
As an example, the **Employees** file (ID = 13) is used below



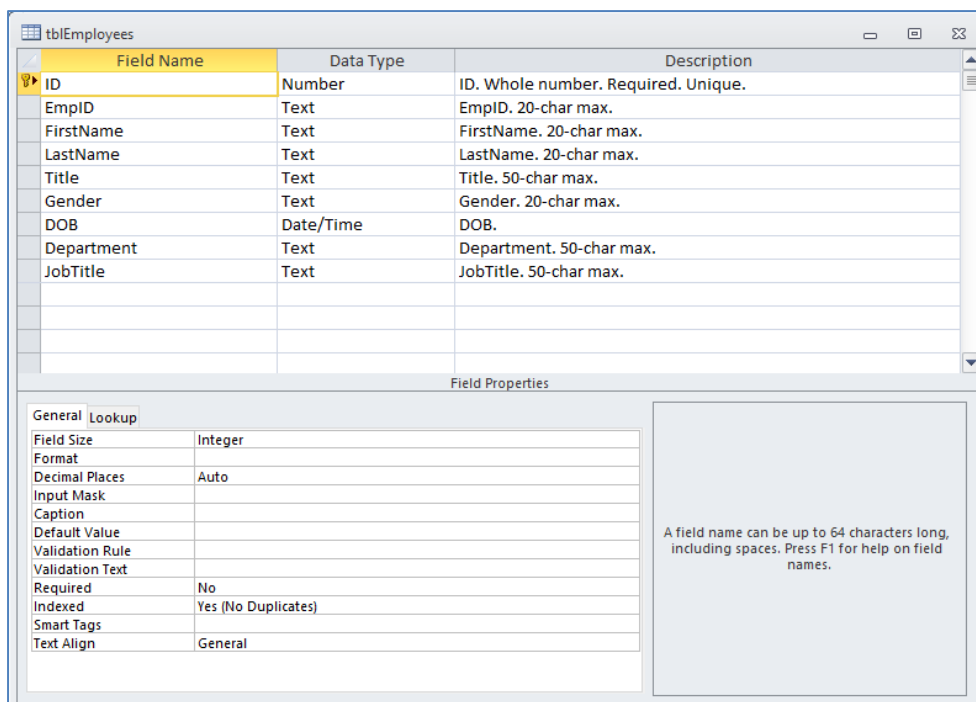
Your first task is to create the fields for the destination table – **tblEmployees**

This can be done in 2 ways:

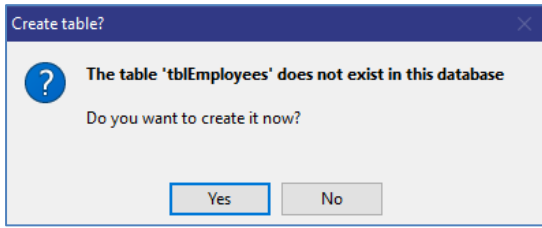
- i) Click the small **Design** button to the right of the table name



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



Otherwise a message like this will be shown:

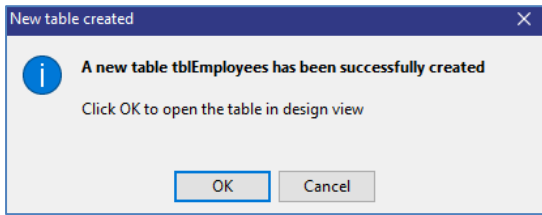


Click **Yes** to create the table.

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:

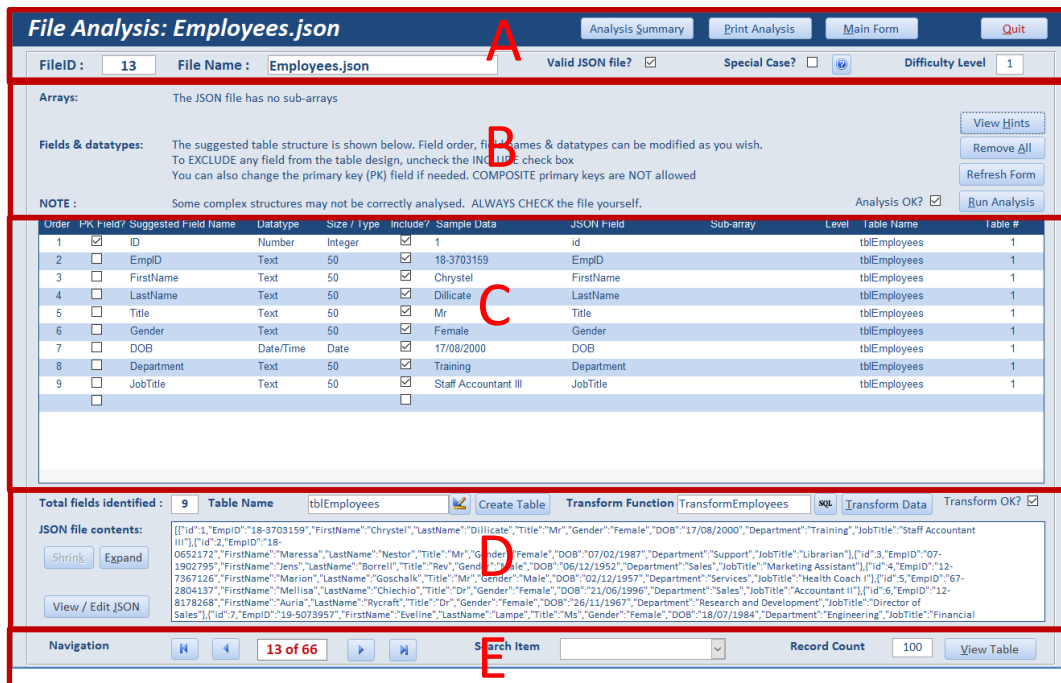


Click **OK** to view the table design.

Edit the fields as required

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

## 5. File Analysis form



This is a very powerful form and can be divided into 5 main sections:

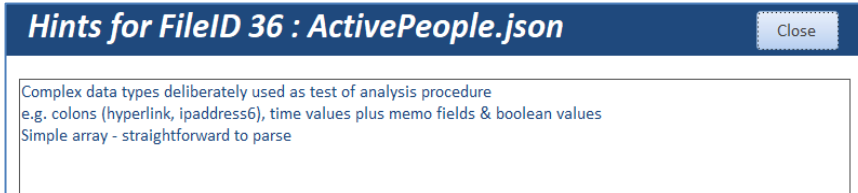
- A) Details of JSON file and reporting controls
- B) Summary of analysis procedure and analysis tools
- C) Analysis details
- D) Table creation and transform tools
- E) Navigation controls

**a) Analyse existing files**

When the form is first opened, file analysis will be done automatically.  
This will only take a couple of seconds unless the file is very large or complex.

Click **Run Analysis** to repeat the process e.g. if the file has been edited  
Click **Remove All** if you want to clear the existing analysis for any reason  
Click **Refresh Form** to update the display if you change the order of the fields listed

Click **View Hints** to view a summary of information about the file structure including any issues arising from the analysis or transform routines. Details can be edited here



The detail section shows the suggested field names, order, datatypes, sample data, sub-array and table info for the destination table(s).

Order	PK Field?	Suggested Field Name	Datatype	Size / Type	Include?	Sample Data	JSON Field	Sub-array	Level	Table Name	Table #
1	<input checked="" type="checkbox"/>	ID	Number	Integer	<input checked="" type="checkbox"/>	1	id			tblEmployees	1
2	<input type="checkbox"/>	EmplID	Text	50	<input checked="" type="checkbox"/>	18-3703159	EmplID			tblEmployees	1
3	<input type="checkbox"/>	FirstName	Text	50	<input checked="" type="checkbox"/>	Chrystal	FirstName			tblEmployees	1
4	<input type="checkbox"/>	LastName	Text	50	<input checked="" type="checkbox"/>	Dillicate	LastName			tblEmployees	1
5	<input type="checkbox"/>	Title	Text	50	<input checked="" type="checkbox"/>	Mr	Title			tblEmployees	1
6	<input type="checkbox"/>	Gender	Text	50	<input checked="" type="checkbox"/>	Female	Gender			tblEmployees	1
7	<input type="checkbox"/>	DOB	Date/Time	Date	<input checked="" type="checkbox"/>	17/08/2000	DOB			tblEmployees	1
8	<input type="checkbox"/>	Department	Text	50	<input checked="" type="checkbox"/>	Training	Department			tblEmployees	1
9	<input type="checkbox"/>	JobTitle	Text	50	<input checked="" type="checkbox"/>	Staff Accountant III	JobTitle			tblEmployees	1
	<input type="checkbox"/>				<input type="checkbox"/>						

The analyser will normally do this perfectly, but you should **ALWAYS CHECK** the details carefully.

Creating tables and transforming the data into normalised tables will be done based on the results of the file analysis. It is therefore **VERY IMPORTANT** to ensure this is correct.

NOTE: Field names will automatically be:

- capitalised and any spaces, underscores or special characters will be removed
- prefixed with the table or subarray name where the default name uses a reserved word in Access such as name/value/date e.g. in the **Statements** file, **Date** becomes **StatementDate**
- prefixed with the subarray name where duplicate fields exist in main array & subarrays

Edit the field names / datatypes / order if required:

- Modify the **primary key** field if the analyser didn't get this correct.  
The field order will automatically be updated to place the new primary key first in the list  
**NOTE: composite primary key fields are NOT allowed**
- Check the field size for text fields (1->255) and the Number field type (integer/long/double)  
If you get an error when transforming data, you may need to increase field size
- 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**  
In such cases, you may need to treat the file as a '**special case**' – see below.

NOTE: Double click on sample data if it is too large to fit in the space available. A 'zoom' window will open showing the full sample data text. The sample data can be edited

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

NOTE: In certain cases, it may be necessary to treat the file as a **special case**. This will allow you to overwrite the default analysis procedures for the specified file. Examples of special cases include: IdealPostcodes (9), GetAddress (10), Colours (16). Click the small button with a question mark to see details of special case code

```
Case "IdealPostcodes.json"
'v4.03 - added sample data for additional fields
CurrentDb.Execute "INSERT INTO tblJSONStructure ( FileID, FieldOrder, PKField, FieldName, Datatype, FieldSizeType, SampleData," & _
" JSONField, Include, SubArrayName, TableName )" & _
" SELECT " & incFileID & " AS FileID, " & (N + 1) & " AS FieldOrder, False AS PKField, 'AddressLine' AS FieldName, 'Memo' AS Datatype," & _
" ' AS FieldSizeType, '12 Second Avenue,,Cliftonville,Margate,Kent' AS SampleData, " & strJSONField & " AS JSONField, True AS Include, " AS Su
" " & strTableName & " AS TableName;"

CurrentDb.Execute "INSERT INTO tblJSONStructure ( FileID, FieldOrder, PKField, FieldName, Datatype, FieldSizeType, SampleData," & _
" JSONField, Include, SubArrayName, TableName )" & _
" SELECT " & incFileID & " AS FileID, " & (N + 2) & " AS FieldOrder, False AS PKField, 'AddressBlock' AS FieldName, 'Memo' AS Datatype," & _
" ' AS FieldSizeType, '12 Second Avenue' AS SampleData, " & strJSONField & " AS JSONField, True AS Include, " AS SubArrayName," & _
" " & strTableName & " AS TableName;"

'v4.03 - exclude unwanted fields
CurrentDb.Execute "UPDATE qryJSONStructure SET qryJSONStructure.Include = False" & _
" WHERE ((qryJSONStructure.SearchItem)='IdealPostcodes') & _
" AND ((qryJSONStructure.FieldOrder) In (3,4,7,9,13,14,17,18,19,20,21,22,23,24,27,28,30,31,32,34,35));"

binSCase = True
```

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:

Order	PK Field?	Suggested Field Name	Datatype	Size / Type	Include?	Sample Data	JSON Field	Sub-array	Array Level	Table Name	Table #
1	<input checked="" type="checkbox"/>	ID	Number	Integer	<input checked="" type="checkbox"/>	1	id			tblCars	1
2	<input type="checkbox"/>	Manufacturer	Text	50	<input checked="" type="checkbox"/>	Oldsmobile	manufacturer			tblCars	1
3	<input type="checkbox"/>	Year	Number	Integer	<input checked="" type="checkbox"/>	2004	year			tblCars	1
4	<input type="checkbox"/>	Model	Text	50	<input checked="" type="checkbox"/>	Suburban	model	model	1	tblCarsModel	2
5	<input type="checkbox"/>	Colour	Text	50	<input checked="" type="checkbox"/>	Indigo	colour	colour	1	tblCarsColour	3
6	<input type="checkbox"/>	Price	Currency		<input checked="" type="checkbox"/>	£19892.77	price	price	1	tblCarsPrice	4

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:

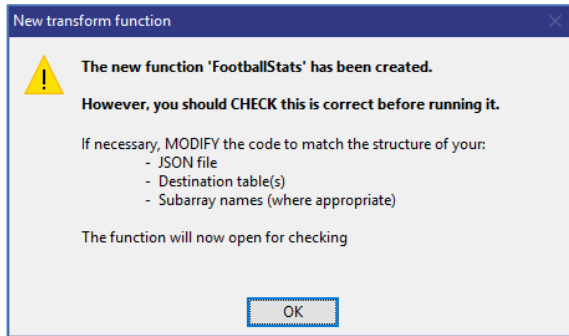
The **transform function** is then created using 2 template files.

The **transform function** is written specifically for the file being tested based on the above file analysis (from version 3.8 onwards).

This should take less than a second to complete.

The new function then opens automatically.

A message like this will be shown indicating that it should be **CHECKED** before use



```
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, ""), 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
```

Carefully **CHECK** the recordset section(s) used to import the parsed data. Whilst this **should** be correct, you should modify it if there are any issues

```
'write to table(s)
Set db = CurrentDb
Set JSON = modJsonConverter.ParseJson(strJSON)

'Build recordset for table(s) matching json fields to destination table fields
'IMPORTANT - the fields are case sensitive - JSON fields are usually lower case

'code for table 1 - tblCricketStats
Set rst = db.OpenRecordset("tblCricketStats", dbOpenDynaset, dbSeeChanges)
With rst
For Each result In JSON("result")
.AddNew

!uniqueID = result("unique_id")
!title = result("title")
!description = result("description")
.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:

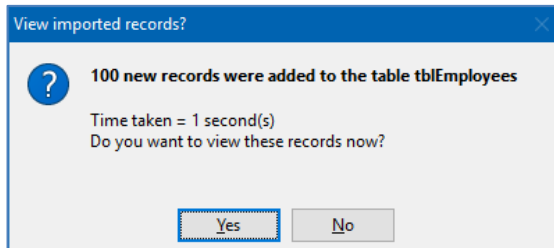
- table field e.g. ContactName
- JSON field e.g. contactName
- Subarray names (where applicable)

When you are satisfied, return to the main form and click the **Transform** button.

The data will then be added to the table(s) created for this file.

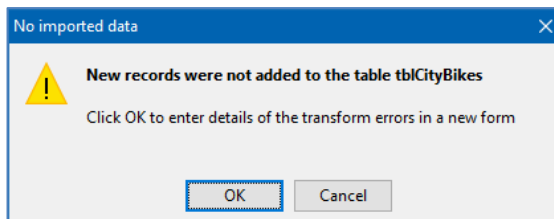
By default, existing data will be overwritten though the function can be modified if appropriate.

If successful, a message similar to this will be shown:



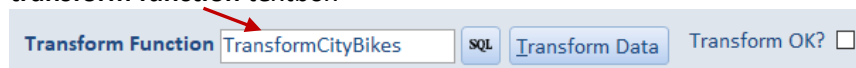
Click the **Yes** button to view the records added.

However if a problem occurs, you may see a message similar to this:



Continue to edit the **Transform function** as necessary until you get this working

If you want to **delete** the **transform function** so you can start again, **double click** on the **transform function** textbox



### NOTE:

The process should be **fully automated** for most **JSON** files with no user intervention required.

However, files with **3 or more levels of nested subarrays** cannot be done completely automatically

### b) 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.

ID: 35

\* Search Item: Postcoder

JSON Site: postcoder.com

URL: http://ws.postcoder.com/pcw/PCW45-12345-12345/address/uk/NR147PZ?format=json&lines=4

\* File Name: Postcoder.json

\* Folder Path: G:\MyFiles\ExampleDatabases\JSON\JSONParser\Files

\* Table Name: tblPostcoder

\* Transform function: TransformPostcoder

\* **REQUIRED** items

Save Cancel

Click the **Save** button

**NOTE:**

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.

**JSON File Importer & Parser** [Print Summary] [Admin] [Exit]

ID: 35

Search Item: Postcoder

JSON Site: postcoder.com

URL: http://ws.postcoder.com/pcw/PCW45-12345-12345/address/uk/NR147PZ?format=json&lines=4

File Name: Postcoder.json

Folder Path: G:\MyFiles\ExampleDatabases\JSON\JSONParser\Files

Table Name: tblPostcoder

Transform function: TransformPostcoder

Tristate level: T (T=true (unicode), F=false (ascii), M=mixed, D= system default)

Import Notes: JSON file saved 12/09/2017 01:08:14

Other Comments:

Difficulty Level: 1 (1 = easy; 5 = very complex)

Completed:  (Ticked = data saved to Access table(s))

Record Count: 0 (Number of records saved to table)

Add New Delete

**JSON Output:** [Shrink] [Expand] [Copy] [Go To Top] [Go To End]

```
[{"addressLine1":"Allies Computing Ltd","addressLine2":"Manor Farm Barns","addressLine3":"Fox Road","addressLine4":"Framingham Pigot","summaryLine":"Allies Computing Ltd,Manor Farm Barns,Fox Road,Framingham Pigot,Norwich,Norfolk,NR14 7PZ","organisation":"Allies Computing Ltd","buildingname":"Manor Farm Barns","premise":"Manor Farm Barns","street":"Fox Road","dependentLocality":"Framingham Pigot","posttown":"Norwich","county":"Norfolk","postcode":"NR14 7PZ"},{"addressLine1":"B 2 B Cashflow Solutions Ltd","addressLine2":"Manor Farm Barns","addressLine3":"Fox Road","addressLine4":"Framingham Pigot","summaryLine":"B 2 B Cashflow Solutions Ltd,Manor Farm Barns,Fox Road,Framingham Pigot,Norwich,Norfolk,NR14 7PZ","organisation":"B 2 B Cashflow Solutions Ltd","buildingname":"Manor Farm Barns","premise":"Manor Farm Barns","street":"Fox Road","dependentLocality":"Framingham Pigot","posttown":"Norwich","county":"Norfolk","postcode":"NR14 7PZ"},{"addressLine1":"Brasteds Event Excellence","addressLine2":"Manor Farm Barns","addressLine3":"Fox Road","addressLine4":"Framingham Pigot","summaryLine":"Brasteds Event Excellence,Manor Farm Barns,Fox Road,Framingham Pigot,Norwich,Norfolk,NR14 7PZ","organisation":"Brasteds Event Excellence","buildingname":"Manor Farm Barns","premise":"Manor Farm Barns","street":"Fox Road","dependentLocality":"Framingham Pigot","posttown":"Norwich","county":"Norfolk","postcode":"NR14 7PZ"},{"addressLine1":"Brasteds Lodge","addressLine2":"Manor Farm Barns","addressLine3":"Fox Road","addressLine4":"Framingham Pigot","summaryLine":"Brasteds Lodge,Manor Farm Barns,Fox Road,Framingham Pigot,Norwich,Norfolk,NR14 7PZ","organisation":"Brasteds Lodge","buildingname":"Manor Farm Barns","premise":"Manor Farm Barns","street":"Fox Road","dependentLocality":"Framingham Pigot","posttown":"Norwich","county":"Norfolk","postcode":"NR14 7PZ"},{"addressLine1":"Conker Interiors Ltd","addressLine2":"Manor Farm Barns","addressLine3":"Fox Road","addressLine4":"Framingham Pigot","summaryLine":"Conker Interiors Ltd,Manor Farm Barns,Fox Road,Framingham Pigot,Norwich,Norfolk,NR14 7PZ","organisation":"Conker Interiors Ltd","buildingname":"Manor Farm Barns","premise":"Manor Farm Barns","street":"Fox Road","dependentLocality":"Framingham Pigot","posttown":"Norwich","county":"Norfolk","postcode":"NR14 7PZ"}]
```

Total characters in JSON file: 11512 Valid JSON File?  [Analyse JSON]

Navigation: 34 of 34 [Search Item] [Table Tools: Empty Table View Table]

JSON File Tool: [Get] [View] [Transform] External JSON Tools: [JSON to CSV] [Excel Power Query] [JSON Editor] [Validate JSON]

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

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



## 6. Reports

Several reports are available from the application:

### a) File Summary

Click the **Print Summary** button on the **Main** form

JSON Files Summary										
ID	Search Item	File Name	URL	Folder Path	Table Name	Transform Function	Level	Valid	Completed	Records
1	GeoIP	GeoIP.json	https://ipapi.co/json	C:\Programs\MendipData\System\JSONParser\Files	tblGeoIP	TransformGeoIP	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
2	Guardian	Guardian.json	https://content.guardianapis.com/search?api-key=0ff966b0-97fa-4f7b-9db9-2fba895e85e1	C:\Programs\MendipData\System\JSONParser\Files	tblGuardian	TransformGuardian	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10
3	Animals	Animals.json	https://api.mockaroo.com/api/88643ad0?count=100&key=ef799ed0	C:\Programs\MendipData\System\JSONParser\Files	tblAnimals	TransformAnimals	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100
4	Countries	Countries.json	https://api.mockaroo.com/api/27159fbd?count=200&key=ef799ed0	C:\Programs\MendipData\System\JSONParser\Files	tblCountries	TransformCountries	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	200
5	Films	Films.json	https://api.mockaroo.com/api/79de0140?count=200&key=ef799ed0	C:\Programs\MendipData\System\JSONParser\Files	tblFilms	TransformFilms	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	200
6	Plants	Plants.json	https://api.mockaroo.com/api/cb94f370?count=100&key=ef799ed0	C:\Programs\MendipData\System\JSONParser\Files	tblPlants	TransformPlants	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100
7	AddressSample	AddressSample.json		C:\Programs\MendipData\System\JSONParser\Files	tblAddressSample	TransformAddressSample	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3
8	Contacts	Contacts.json		C:\Programs\MendipData\System\JSONParser\Files	tblContacts	TransformContacts	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	91
9	IdealPostcodes	IdealPostcodes.json	https://api.ideal-postcodes.co.uk/v1/postcodes?st92!Papi_key=idqd	C:\Programs\MendipData\System\JSONParser\Files	tblIdealPostcodes	TransformIdealPostcodes	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	27
10	GetAddress	GetAddress.json	https://api.getAddress.io/fin/c/ct92!Papi_key=bc1rmRm2kSlyfg-l-qcw9770&format=false&sort=true	C:\Programs\MendipData\System\JSONParser\Files	tblGetAddress	TransformGetAddress	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	27
11	Postcoder	Postcoder.json	http://vs.postcoder.com/pcw/PCW45-12345-12345-1234X/address/uk/NR147PZ?format=json	C:\Programs\MendipData\System\JSONParser\Files	tblPostcoder	TransformPostcoder	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	28

08 November 2017 Page 1 of 6

### b) File Analysis Details

Click the **Print Analysis** button on the **Analysis** form

JSON File Structure Summary - Cars.json										FileID : 34	
<b>File Hints :</b>		File contains 3 separate subarrays - model, colour, price so data needs to be saved into 4 tables									
<b>Arrays :</b>		The JSON file has 3 sub-arrays: model, colour, price It is RECOMMENDED that the data is saved in 4 tables: tblCars, tblCarsModel, tblCarsColour, tblCarsPrice Check the final column and if necessary re-assign fields to the correct table									
<b>Fields &amp; datatypes :</b>		The suggested table structure is shown below. Field names & datatypes can be modified as you wish. To EXCLUDE any field from the table design, uncheck the INCLUDE check box You can also change the primary key (PK) field if needed. COMPOSITE primary keys are NOT allowed									
<b>NOTE :</b>		Some complex structures may not be correctly analysed. ALWAYS CHECK the file yourself.									
Order	PK Field	Field Name	Datatype	Size / Type	Sample Data	Include	JSON Field	SubArray Name	SubArray Level	Table Name	Table Number
1	<input checked="" type="checkbox"/>	ID	Number	Integer	1	<input checked="" type="checkbox"/>	id			tblCars	1
2	<input type="checkbox"/>	Manufacturer	Text	50	Oldsmobile	<input checked="" type="checkbox"/>	manufacturer			tblCars	1
3	<input type="checkbox"/>	Year	Number	Integer	2004	<input checked="" type="checkbox"/>	year			tblCars	1
4	<input type="checkbox"/>	Model	Text	50	Suburban	<input checked="" type="checkbox"/>	model	model	1	tblCarsModel	2
5	<input type="checkbox"/>	Colour	Text	50	Indigo	<input checked="" type="checkbox"/>	colour	colour	1	tblCarsColour	3
6	<input type="checkbox"/>	Price	Currency		£19892.77	<input checked="" type="checkbox"/>	price	price	1	tblCarsPrice	4
Analysis Done : 08/11/2017    Special Case : <input type="checkbox"/> Field Count : 6    Table Count : 4    Tables Created : <input checked="" type="checkbox"/> Transform Done : 04/11/2017    Record Count : 100											

08 November 2017 Page 1 of 1

c) **File Analysis Summary**

Click the **Analysis Summary** button on the **Analysis** form

<b>JSON Analysis &amp; Transform Summary</b>													
ID	Search Item	Special Case	Analysis Summary				Transform Summary						
			Date Completed	OK?	Errors	Time Taken	Table Count	Field Count	Date Completed	OK?	Errors	Time Taken	Record Count
1	GeoIP	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		2	1	13	26/10/2017	<input checked="" type="checkbox"/>		1	1
2	Guardian	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		1	1	11	04/11/2017	<input checked="" type="checkbox"/>		1	10
3	Animals	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		1	1	3	26/10/2017	<input checked="" type="checkbox"/>		2	100
4	Countries	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		2	1	8	26/10/2017	<input checked="" type="checkbox"/>		1	200
5	Films	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	7	26/10/2017	<input checked="" type="checkbox"/>		3	200
6	Plants	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	4	26/10/2017	<input checked="" type="checkbox"/>		2	100
7	AddressSample	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		1	1	5	26/10/2017	<input checked="" type="checkbox"/>		1	3
8	Contacts	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		1	1	7	26/10/2017	<input checked="" type="checkbox"/>		1	91
9	IdealPostcodes	<input checked="" type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	37	04/11/2017	<input checked="" type="checkbox"/>		1	27
10	GetAddress	<input checked="" type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	6	04/11/2017	<input checked="" type="checkbox"/>		1	27
11	Postcoder	<input type="checkbox"/>	06/11/2017	<input checked="" type="checkbox"/>		1	1	14	06/11/2017	<input checked="" type="checkbox"/>		1	28
12	Products	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	5	26/10/2017	<input checked="" type="checkbox"/>		1	3
13	Employees	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		2	1	9	26/10/2017	<input checked="" type="checkbox"/>		6	100
14	Drugs	<input type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	5	26/10/2017	<input checked="" type="checkbox"/>		5	100
15	Statement	<input type="checkbox"/>	08/11/2017	<input checked="" type="checkbox"/>		1	1	5	08/11/2017	<input checked="" type="checkbox"/>		1	60
16	Colours	<input checked="" type="checkbox"/>	05/11/2017	<input checked="" type="checkbox"/>		1	1	3	26/10/2017	<input checked="" type="checkbox"/>		1	148

08 November 2017 Page 1 of 5

d) **File Information**

Click the **JSON File Info** button on the **Help** ribbon at the top of the screen

<b>JSON File Info</b>								
FileID	SearchItem	File Hints	Special Case	Analysis OK	Table Count	Field Count	Transform OK	Record Count
1	GeoIP	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	13	<input checked="" type="checkbox"/>	1
2	Guardian	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	11	<input checked="" type="checkbox"/>	10
3	Animals	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	3	<input checked="" type="checkbox"/>	100
4	Countries	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	8	<input checked="" type="checkbox"/>	200
5	Films	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	7	<input checked="" type="checkbox"/>	200
6	Plants	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	4	<input checked="" type="checkbox"/>	100
7	AddressSample	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input checked="" type="checkbox"/>	3
8	Contacts	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	7	<input checked="" type="checkbox"/>	91
9	IdealPostcodes	This file is well structured and easy to parse as a single table. It has far more fields than are likely to be needed for address purposes.  Two derived fields have been added: address line & address block. For that reason, the file needs to be treated as a 'special case'  Also lots of unnecessary fields have been excluded	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	37	<input checked="" type="checkbox"/>	27
10	GetAddress	This address file from an online provider is poorly structured with no field names The postcode used for the search isn't included in the file output! It is extracted from the web url and added along with a derived address block field.  The file structure actually contains a subarray - addresses. However, it makes more sense to treat it as a single table containing all required fields  For all the above reasons, the file needs to be treated as a special case for the transform function code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	6	<input checked="" type="checkbox"/>	27
11	Postcoder	Well structured address file Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	14	<input checked="" type="checkbox"/>	28
12	Products	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input checked="" type="checkbox"/>	3

08 November 2017 Page 1 of 7

## 7. Admin Menu form

This is used to run various admin tasks e.g. backup the database

Button Label	Description
Backup Database	Save a copy of this database to the backups folder
Program Settings	View / edit program settings
EMail Settings	View / edit settings used to send emails from the program
Program Info	View information about the program
Form Resizer	Used to resize forms if these have become over enlarged
User Guide	View a user guide for this program
View Start Form	View the start up form
Empty All JSON Data Tables	Empties all tables based on imported JSON data
Delete All JSON Data Tables	Delete all JSON data tables e.g. prior to version releasse
Delete All Transform Functions	Deletes all transform functions e.g. prior to version release
Delete All File Analysis Data	Deletes all JSON file analysis data
Remove Data / Analysis Tables & Transform Functions	Clears all items prior to version release. Same as running above 3 buttons together
Run Powershell	Runs a file script in Powershell (not available in this version)
View Version Changes Summary	View a summary of changes to the program by version

- **Backup database** – saves a copy of the database to the **Backups** folder with the version number and the backup date/time e.g. JSONParser\_v4.20\_20171206184439.accdb
- **Program Settings** – this contains various values used throughout the application  
NOTE: incorrectly set values may prevent the application working correctly

ID	ItemName	ItemValue
1	ProgramFolder	C:\Programs\MendipDataSystems\JATFA
2	DefaultFilesFolder	C:\Programs\MendipDataSystems\JATFA\Files
3	BackupFolder	C:\Programs\MendipDataSystems\JATFA\Backups
4	VideosFolder	C:\Programs\MendipDataSystems\JATFA\Videos
5	ImagesFolder	C:\Programs\MendipDataSystems\JATFA\Images
6	HelpFolder	C:\Programs\MendipDataSystems\JATFA\Help
7	StartFormStatus	Yes
8	ShowNewJSONWizard	Yes
9	UpdateFlag	No
10	RegisteredUser	Colin Riddington
11	UserCompany	Mendip Data Systems
12	UserEMail	
13	UserAddress	
14	UserPhone	
15	LicenceKey	
16	SupportExpiryDate	
17	WorkstationName	COLIN-PC
18	WindowsVersion	Microsoft Windows 10 Pro 10.0.16299 32-bit
19	AccessVersion	2010 SP2 32-bit; Build 14.0.7162
20	ScreenResolution	1680*1050 Widescreen
21	EMailUseOutlook	

**CARE: Incorrect settings may prevent this program working correctly**

These items can be grouped as follows:

- 1-6 : set automatically relative to the application path – **DO NOT ALTER**
- 7-8 : user preferences
- 9 : used during version updates – **DO NOT ALTER**
- 10-16 : user & licence info entered for this application
- 17-20 : computer info (useful if you need to contact program support)
- 21-29 : settings used to send emails from the application

- **Email Settings**

If not already done, complete this form so that email can be sent direct from the application  
Send a test email to check this is working correctly.

**Email Settings** [Close]

Use this form to test settings used to send emails from this program

Use Outlook for Email: No [Clear EMail Settings]

**Settings:** *NOTE: The items below are NOT required if Outlook is used to send email from the program*

Send Mail Using Method: 2 (1 = local; 2 = network)

Port used to send email: 25 (Usually 25)

Email server: smtp.gmail.com (e.g. "smtp.gmail.com")

SMTP Authenticate: 1 (Usually 1)

UserName: xyz@gmail.com (Default user email for email from this program)

Password: \*\*\*\*\* (Password for default user email)

Timeout (seconds): 60 (e.g. 60)

Use SSL: True (True / False)

**Test email:**  
Fill in the details below then click the 'Send Email' button to try & send a test email message with an attached file.  
Click the 'Save Settings' button if the email is sent successfully.  
Click the "Help" button for more information if an error message is shown.  
Click the "Clear" button to remove test email data from this section.

Recipient email address: [Text Box]

Attachment filename (optional): [Text Box] [Browse]

*This must be the full file path of the attachment*

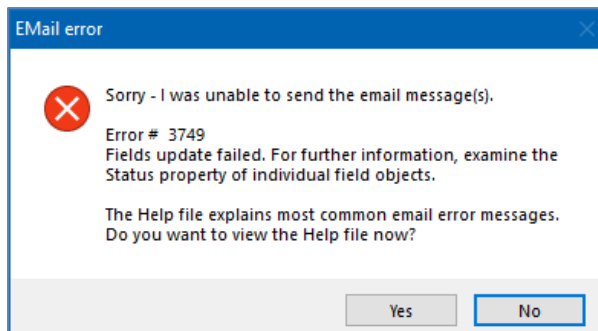
[Send Email] [Clear Test Email Data] [Help] [Save Settings]

**Default email addresses:** *The following are used by various features in the program to send emails automatically*

Program Support EMail: support@mendipdatasystems.co.uk

Registered User EMail: info@mendipdatasystems.co.uk

If errors occur, a message similar to this will be shown:



Click the **Help** button for more information about the error

**NOTE:** Click **Save Settings** when done. Email settings are **NOT** saved automatically

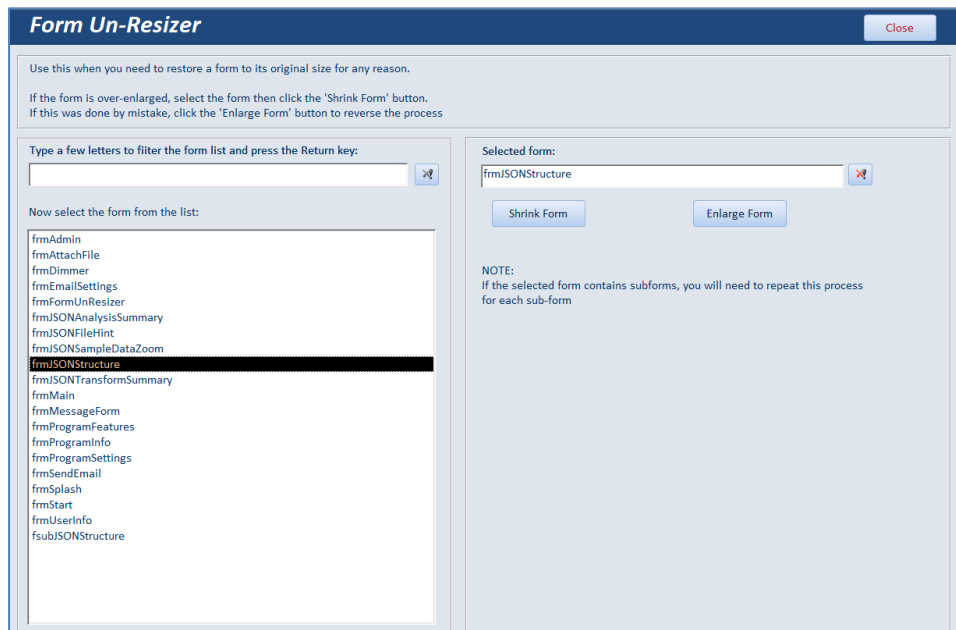
- **Program Info**

Brief summary info for this version

**Program Info** [Close]

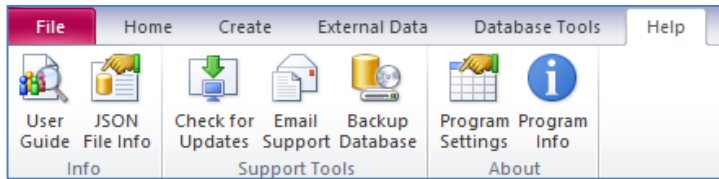
ID	ItemName	ItemValue
1	ProgramName	JSON Parser
2	Version	4.10
3	VersionDate	05/12/2017
4	SupportEmail	support@mendipdatasystems.co.uk
5	ProgramWebsite	http://www.mendipdatasystems.co.uk/json-parser/4594050096
6	Author	Colin Riddington
7	Company	Mendip Data Systems
8	JSONParserAuthor	Tim Hall
9	JSONParserWebsite	https://github.com/VBA-tools/VBA-JSON
10	64-bit compatible	Yes
11	WebVersion	4.05

- **Form Resizer** – sometimes during development work forms can become ‘over-enlarged’. This is unlikely to occur in normal program use  
However if necessary, use this tool to restore any affected forms to their correct size



- **User Guide**  
Opens this document
- **View Start Form**  
See [page 1](#) for more information
- **Empty All JSON Data Tables**  
Use this to delete all records from the JSON data tables.  
The tables themselves are NOT deleted  
A backup is automatically done first
- **Delete All JSON Data Tables**  
Use this to delete all JSON data tables.  
A backup is automatically done first
- **Delete All Transform Functions**  
Use this to delete all transform functions for the data tables  
A backup is automatically done first
- **Delete All File Analysis Data**  
Use this to empty all data in JSON file analysis tables.  
A backup is automatically done first
- **Remove JSON Data / Analysis Tables & Transform Functions**  
Clears all items. Identical to running above 3 items together  
A backup is automatically done first
- **Run Powershell**  
Not available in this version
- **View Version Changes Summary**  
Summary of changes made in each version of the application

## 8. Help ribbon



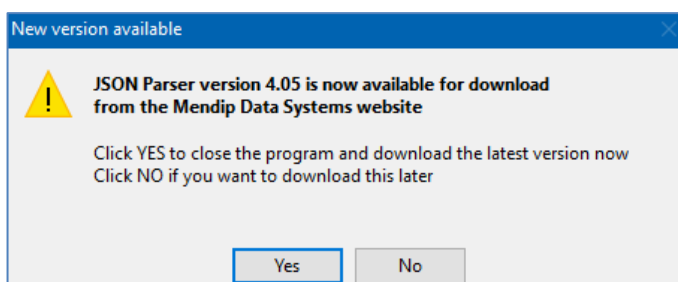
The **Help** ribbon contains 7 items for quick access:

- **User Guide** – opens this document
- **JSON File Info** – summary info report for JSON files used with this application

Field	SearchItem	File Hints	Special Case	Analysis OK	Table Count	Field Count	Transform OK	Record Count
13	Employees	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	9	<input checked="" type="checkbox"/>	100
14	Drugs	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input checked="" type="checkbox"/>	100
15	Statement	Simple array - straightforward to parse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input checked="" type="checkbox"/>	50
16	Colours	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: colour & colour value. For that reason it is treated as a special case	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	3	<input checked="" type="checkbox"/>	148
17	Awkwardstring	This is a simple file deliberately designed to contain a wide variety of 'awkward' characters that could cause issues with parsing code. Although slightly slower than some other files, the file is transformed without difficulty.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	2	<input checked="" type="checkbox"/>	100
18	Currencies	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: currency code & currency. For that reason it is treated as a special case	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	3	<input checked="" type="checkbox"/>	168
19	CurrencyExchange	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: currency & (exchange) rate. The base rate & data download date have been added to each record for clarity. For that reason it is treated as a special case	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	5	<input checked="" type="checkbox"/>	31
20	Geonames	Treated as special case so selected fields are automatically excluded - modify if all fields required. Also, the first field in the file is Lng (Longitude) which could be used as the primary key field. However, it is more logical to use the Geoname ID field as the PK field. For that reason the order has been changed as part of special case code with GeonameID set as 1 & Lng as 2.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	12	<input checked="" type="checkbox"/>	10
21	Books	Simple array - straightforward structure	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	9	<input checked="" type="checkbox"/>	100

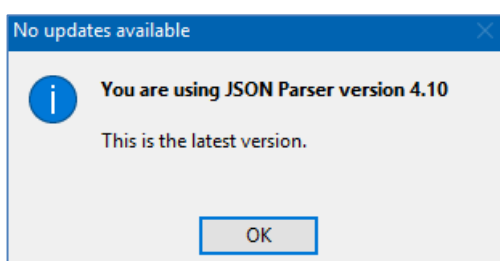
06 December 2017 Page 2 of 8

- **Check for Updates** – checks for a newer version on the **Mendip Data Systems** website. If a newer version is available, you will see a message similar to this:



Click **YES** to download the latest version from the website

If you already have the latest version, a message like this will be shown:



- **Email Support** – send an email to **MDS customer support**:
  - if you have any questions about using specific parts of the application
  - to report any bugs / errors you experience using the application
  - to suggest improvements or additional features for a future version

If you are reporting a bug, please include the workstation info with the email as this may assist in providing a solution

- **Backup database** - identical to the **Admin** menu item
- **Program Settings** - identical to the **Admin** menu item
- **Program Info** - identical to the **Admin** menu item

For more information on using JSON files with this application, see the **Appendices**:

- [Appendix A](#) : JSON File Structure
- [Appendix B](#) : Reading & Editing JSON files in this application
- [Appendix C](#) : Info about supplied JSON files
- [Appendix D](#) : More Files
- Appendix E : Possible Issues

The following are also useful sources of information:

- <https://en.wikipedia.org/wiki/JSON>
- [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- [https://adobe.github.io/Spry/samples/data\\_region/JSONDataSetSample.html](https://adobe.github.io/Spry/samples/data_region/JSONDataSetSample.html)

I hope you find this version of the **JSON Analyse & Transform for Access** application useful. All feedback on its use and suggestions for further development will be very welcome.

Colin Riddington  
Mendip Data Systems

Email: [info@mendipdatasystems.co.uk](mailto:info@mendipdatasystems.co.uk)  
Website: [www.mendipdatasystems.co.uk](http://www.mendipdatasystems.co.uk)

## Appendix A - JSON File Structure

**JSON** (JavaScript Object Notation) is a syntax for storing and exchanging data.

**JSON** files are text with file type '.json' and can be viewed in any text editor e.g. Notepad

### JSON Syntax Rules

- Data is in name/value pairs separated by commas
- Curly braces hold objects, square brackets hold arrays. These are also in pairs
- Files **MUST** always start with a '{' or a '[' and end with a '}' or a ']'
- The number of '{' and '}' must match. Similarly for '[' and ']'

### NOTE:

This application handles files enclosed in square brackets [] slightly differently to those enclosed in braces {}

This is explained in [Appendix B](#)

A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

For example: "name":"John"

**Field names** usually start with a lower case letter

Where combined words are used in field names, capitals or underscores may be used

For example: "firstName":"John" or "first\_name":"John"

**Values** must be one of the following data types:

- String / number / JSON object / array / Boolean / null

**String values** must be written with double quotes: "name":"John"

**Number values** do NOT have quotes: "engineSize": 1495

To improve readability, JSON files may include 'whitespace' and additional square bracketing.

The following examples are structurally identical:

```
{"employee": { "name":"John", "age":30, "city":"New York" } }
```

```
{"employee": [ { "name":"John", "age":30, "city":"New York" } ] }
```

```
{
  "employee":
    { "name":"John",
      "age":30,
      "city":"New York"
    }
}
```

```
{
  "employee": [
    { "name":"John",
      "age":30,
      "city":"New York"
    } ]
}
```

Arrays of values only have square brackets:

```
"manufacturer": [ "Ford", "BMW", "Fiat" ]
```



### ***Nested JSON Objects***

Values in a JSON object can be another JSON object.

```
{
  "name": "John",
  "age": 30,
  "cars": [ {
    "car1": "Ford",
    "car2": "BMW",
    "car3": "Fiat"
  } ]
}
```

### ***Nested Arrays in JSON Objects***

Values in an array can also be another array ('subarray'), or even another JSON object:

```
{
  "name": "John",
  "age": 30,
  "cars": [
    { "name": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },
    { "name": "BMW", "models": [ "320", "X3", "X5" ] },
    { "name": "Fiat", "models": [ "500", "Panda" ] }
  ]
}
```

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

## Appendix B – Reading & Editing JSON files in this application

Often **JSON** files may contain extraneous information that is not required when saving data

This application uses the following rules to determine which portions should be included:

- Where files are enclosed in square brackets, the ENTIRE file is read
- Where files are enclosed in curly braces, the portion to the left of the next square bracket or curly brace is discarded

In this example, the leading section is ignored: `{"id":"0001","type":"donut","name":"Cake","image":`

```
{"id":"0001","type":"donut","name":"Cake","image":["IFile":"images/0001.jpg","lwidth": 200,"lheight": 200],"thumbnail":["TFile":"images/thumbnails/0001.jpg","Twidth": 32,"Theight": 32]}
```

The file detects 6 fields and places the thumbnail data in a subarray.

Order	PK Field?	Suggested Field Name	Datatype	Size / Type	Include?	Sample Data	JSON Field	Sub-array	Array Level	Table Name
1	<input checked="" type="checkbox"/>	IFile	Hyperlink		<input checked="" type="checkbox"/>	images/0001.jpg	IFile			tblImageData
2	<input type="checkbox"/>	lwidth	Number	Integer	<input checked="" type="checkbox"/>	200	lwidth			tblImageData
3	<input type="checkbox"/>	lheight	Number	Integer	<input checked="" type="checkbox"/>	200	lheight			tblImageData
4	<input type="checkbox"/>	TFile	Hyperlink		<input checked="" type="checkbox"/>	images/thumbnails/0001.jpg	TFile	thumbnail	1	tblImageDataThumbnail
5	<input type="checkbox"/>	Twidth	Number	Integer	<input checked="" type="checkbox"/>	32	Twidth	thumbnail	1	tblImageDataThumbnail
6	<input type="checkbox"/>	Theight	Number	Integer	<input checked="" type="checkbox"/>	32	Theight	thumbnail	1	tblImageDataThumbnail

However, if the file is enclosed in square brackets, the entire file is used:

```
[{"id":"0001","type":"donut","name":"Cake","image":["IFile":"images/0001.jpg","lwidth": 200,"lheight": 200],"thumbnail":["TFile":"images/thumbnails/0001.jpg","Twidth": 32,"Theight": 32]}
```

The analyser now finds 9 fields with the 3 fields in the leading section treated as the top level array. The image and thumbnail data form 2 separate subarrays:

Order	PK Field?	Suggested Field Name	Datatype	Size / Type	Include?	Sample Data	JSON Field	Sub-array	Array Level	Table Name
1	<input checked="" type="checkbox"/>	ID	Number	Integer	<input checked="" type="checkbox"/>	0001	id			tblImageData
2	<input type="checkbox"/>	Type	Text	50	<input checked="" type="checkbox"/>	donut	type			tblImageData
3	<input type="checkbox"/>	ImageDataName	Text	50	<input checked="" type="checkbox"/>	Cake	name			tblImageData
4	<input type="checkbox"/>	IFile	Hyperlink		<input checked="" type="checkbox"/>	images/0001.jpg	IFile	image	1	tblImageDataImage
5	<input type="checkbox"/>	lwidth	Number	Integer	<input checked="" type="checkbox"/>	200	lwidth	image	1	tblImageDataImage
6	<input type="checkbox"/>	lheight	Number	Integer	<input checked="" type="checkbox"/>	200	lheight	image	1	tblImageDataImage
7	<input type="checkbox"/>	TFile	Hyperlink		<input checked="" type="checkbox"/>	images/thumbnails/0001.jpg	TFile	thumbnail	1	tblImageDataThumbnail
8	<input type="checkbox"/>	Twidth	Number	Integer	<input checked="" type="checkbox"/>	32	Twidth	thumbnail	1	tblImageDataThumbnail
9	<input type="checkbox"/>	Theight	Number	Integer	<input checked="" type="checkbox"/>	32	Theight	thumbnail	1	tblImageDataThumbnail

When analysing your own files, try editing the file if necessary to help ensure the correct output:

- Add (or remove) square brackets at the start and end of the file
- Always enclose all subarrays with square brackets

JSON files normally use the Unicode character set (tristate = true) though some use ASCII (tristate = false). Both can be detected and read correctly by the parser.

Tristate level  T=true (unicode), F=false (ascii), M=mixed, D= system default

**IMPORTANT:**

**JSON** files can be easily edited using applications such as **Notepad**.

However, by default the file will be saved using the ANSI character set.

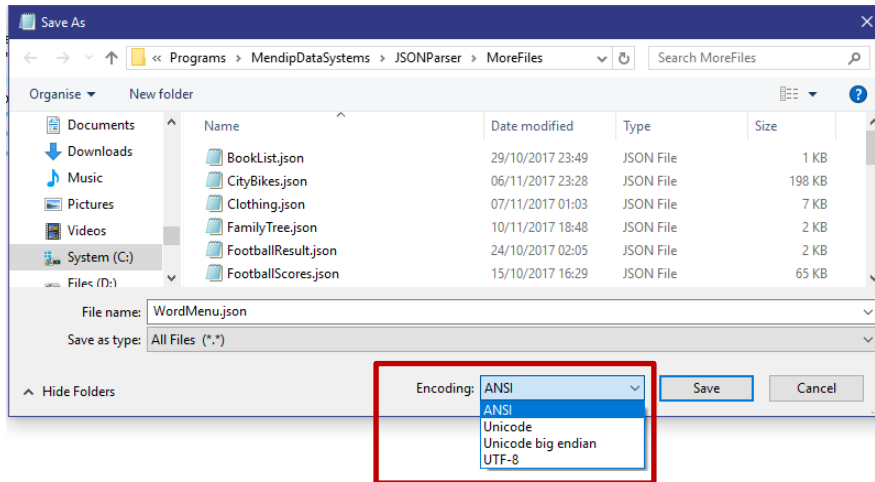
This can cause problems in some cases.

Using incorrect settings may add some unreadable characters at the start of the file and make it invalid.

```
["firstName":"John","lastName":"Smith","isAlive":_true,"age":_25,"address":_{"streetAddress":"21_2nd_Street","city":_New_York,"state":"NY","postalCode":"10021-3100"},"phoneNumbers":_["type":"home","number":"212_555-1234"},_{"type":"office","number":"646_555-4567"},_{"type":"mobile","number":"123_456-7890"}_],"children":_[],"spouse":_null]}
```

To fix this, change the default to **Unicode** before saving the file

The character set encoding can be identified if you click **File ... Save As**



Change the setting to **Unicode**.

Once this has been done once for a file, the **Unicode** character set will be retained when the file is saved again after any further editing.

## Appendix C – JSON File Info

A summary of the JSON files supplied with this application is provided over the next few pages. The same information is available as a report from the **JSON File Info** button on the **Help** ribbon

File ID	File	Special Case	File Hint	Table Count	Field Count	Record Count
1	GeolP	No	Simple array - straightforward to parse	1	13	1
2	Guardian	No	Simple array - straightforward to parse	1	11	10
3	Animals	No	Simple array - straightforward to parse	1	3	100
4	Countries	No	Simple array - straightforward to parse	1	8	200
5	Films	No	Simple array - straightforward to parse	1	7	200
6	Plants	No	Simple array - straightforward to parse	1	4	100
7	AddressSample	No	Simple array - straightforward to parse	1	5	3
8	Contacts	No	Simple array - straightforward to parse	1	7	91
9	IdealPostcodes	Yes	This file is well structured and easy to parse as a single table. It has far more fields than are likely to be needed for address purposes Two derived fields have been added: address line & address block. For that reason, the file needs to be treated as a 'special case' Also lots of unnecessary fields have been excluded	1	37	27
10	GetAddress	Yes	This address file from an online provider is poorly structured with no field names The postcode used for the search isn't included in the file output! It is extracted from the web url and added along with a derived address block field. The file structure actually contains a subarray - addresses. However, it makes more sense to treat it as a single table containing all required fields. For all the above reasons, the file is treated as a special case for the transform function code	1	6	27
11	Postcoder	No	Well-structured address file Simple array - straightforward to parse	1	14	28
12	Products	No	Simple array - straightforward to parse	1	5	3
13	Employees	No	Simple array - straightforward to parse	1	9	100
14	Drugs	No	Simple array - straightforward to parse	1	5	100
15	Statement	No	Simple array - straightforward to parse	1	5	60
16	Colours	Yes	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: colour & colour value. For that reason, it is treated as a special case	1	3	148
17	AwkwardString	No	This is a simple file deliberately designed to contain a wide variety of 'awkward' characters that could cause issues with parsing code. Although slightly slower than some other files, the file is transformed without difficulty.	1	2	100
18	Currencies	Yes	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: currency code & currency. For that reason, it is treated as a special case	1	3	168
19	CurrencyExchange	Yes	Although the structure of this file is easy to visualise, by default it would be parsed into a 'spreadsheet-type' format of many fields each with one record. The transform function has been designed to modify the output into a normalised database structure with 2 fields: currency & (exchange) rate. The base rate & data download date have been added to each record for clarity. For that reason, it is treated as a special case	1	5	31
20	Geonames	Yes	Treated as special case so selected fields are automatically excluded - modify if all fields required. Also, the first field in the file is Lng (Longitude) which could be used as the primary key field. However, it is more logical to use the GeonameID field as the PK field. For that reason, the order has been changed as part of special case code with GeonameID set as 1 & Lng as 2	1	12	10
21	Books	No	Simple array - straightforward structure	1	9	100

22	FootballClubs	No	Simple array - straightforward to parse Automated transform function completed 17/10/2017	1	4	20
23	Monarchs	Yes	Simple array - straightforward to parse Years MUST be text field as most values in form: 1132-1147 OPTIONAL: Treat as special case to automate renaming fields	1	5	57
24	States	Yes	Simple array - straightforward to parse OPTIONAL: Treat as special case to automate renaming fields	1	5	56
25	Presidents	Yes	Simple array - straightforward to parse OPTIONAL: Treat as special case to automate renaming fields	1	4	44
26	Images	No	File with 1 subarray - tags so is transformed into 2 tables	2	4	30
27	Restaurants	No	Relatively simple array Small part of a MUCH larger dataset to speed up loading time Issues with field names fixed using code: - Special character \$ removed from field name \$oid	1	9	26
28	Students	No	File with 1 subarray - scores Straightforward to parse	2	4	200
29	TestJLeach	Yes	File supplied by Jack Leach @UtterAccess forum as a challenge Deceptively simple structure but deliberately complex to transform. Various unwanted characters need to be removed to do this successfully. For that reason, it is treated as a special case	1	3	3
30	People	No	File contains 2 subarrays so is transformed into 3 tables	3	13	300
31	MultiSquad	No	File contains 2 subarrays (members, powers) so is transformed into 3 tables Powers table is sub-sub array of members table Age has to be Long integer due to a value = 1,000,000 Detected correctly in analysis	3	9	2
32	ImportedCalls	Yes	File with many fields including some fields that could cause issues All listed fields in calls subarray so treated as 1 table Analysis correctly identifies 5 phone number fields starting with '+' as text (20) fields. Null 'sid' fields excluded to prevent error 3058 Treated as special case to automate this process Automated transform function then works correctly	1	26	7
33	Programs	No	Straightforward file with 1 subarray so data saved into 2 tables Different version values including 9.75; 3.5.6 could individually be read as numbers or dates but overall need to be text Analysis correctly detects this datatype	2	5	200
34	Cars	No	File contains 3 separate subarrays - model, colour, price so data needs to be saved into 4 tables	4	6	100
35	Vehicles	No	File contains 2 nested subarrays - model, colour so data needs to be saved into 3 normalised tables	3	5	100
36	ActivePeople	No	Complex data types deliberately used as test of analysis procedure e.g. colons (hyperlink, ipaddress6), time values plus memo fields & boolean values. Simple array - straightforward to parse	1	20	30
37	Spaceship	No	Relatively simple file with 1 subarray - cargo Cargo table is an array of values so split function used to parse	2	4	10
38	Retailers	No	This is a large file with many fields. File contains 1 subarray so is transformed into 2 tables	2	15	239
39	PhoneRecords	Yes	Shortened version of very large file originally from <a href="https://oresapp.asicanada.net/ores.imis.services/api/member">https://oresapp.asicanada.net/ores.imis.services/api/member</a> The original file is huge - around 470K characters so took almost a minute to load fully. Shortened version used here so it loads faster Added missing [] brackets in Geocode subarrays File contains 2 subarrays so is transformed into 3 tables Duplicate ID field in members subarray is ignored. Special case to automate exclusion of 2 unwanted fields in members subarray	3	24	13
40	Generated	No	Complex array with 2 separate subarrays - tags, friends	3	22	3
41	Weather	No	Simplified JSON file with unnecessary subarrays removed. Straightforward to parse	1	24	1
42	Weather2	No	Simplified version of file from <a href="http://www.wunderground.com">www.wunderground.com</a> Original file had lots of unnecessary subarrays which have been removed. File has huge number of fields that could probably be edited significantly	1	60	1
43	UKElection2015	No	Converted from a CSV file. Very large file for all 650 constituencies so slow to load & analyse Simple array so straightforward to parse	1	28	650
44	UKElection2015Full	Yes	Converted from a CSV file. Subset of 30 constituencies from much larger file of all 650 constituencies. Treated as special case so file can be split into 2 tables: a) constituency info    b) votes for each party	2	17	30

45	Cake	No	File with 2 subarrays - batter, topping which was taken from: <a href="https://adobe.github.io/Spry/samples/data_region/JSONDataSetSample.html">https://adobe.github.io/Spry/samples/data_region/JSONDataSetSample.html</a> The original file used fields id & type in both subarrays. This meant the 2nd subarray was ignored as the parser cannot use the same field name more than once. Edited JSON file to create unique field names & fix the issue	3	8	1
46	CricketStats	No	Simple array - straightforward to parse	1	3	9
47	Inspections	No	Simple array - straightforward to parse	1	10	28
48	MoreCakes	No	File contains 3 top level arrays - batter, topping, filling	4	11	6
49	ForumPosts	No	Original file from stackoverflow forum Transform failed - primary key cannot contain a null value Removed unnecessary owner subarray & it now works Reset PK field as UserID in place of reputation - field order automatically updates	1	14	29
50	RGBColours	No	Original downloaded file could not be analysed as field names were not enclosed in "" e.g. {colour:"red",value:"#f00"} Corrected file is simple array & straightforward to parse	1	3	7
51	Albums	No	File has 1 subarray Track length is a time field - now detected correctly by analyser	2	6	4
52	ProductOffer	No	Some fields missing & hyperlink fields incorrect	4	17	1
53	Shoes	No	Highly complex file with 7 nested subarrays: Level 1 – product, Level 2 - colors, images, relatedItems, referencedItems, sizes, options. Fixed issue with fields in colors subarray Analysis procedure modified to correctly handle identification of fields in level 2 subarrays	8	35	1
54	PostInfo	No	File with 2 nested subarrays: posts & comments	3	6	1
55	ITHelpDesk	No	Simple array - straightforward to parse NOTE: Three other versions of this file with the same data but different structures exist in the More Files folder: - ITHelpDesk2, ITHelpDesk3, ITHelpDesk4 Not all are so easy to parse! It is instructive to try these and compare the analysis results in each case.	1	10	200
56	Bitcoin	No	Simple array - straightforward to parse	1	10	2
57	CarList	No	File with 1 subarray - quality Original file caused issues due to incorrect bracketing - now fixed	2	8	5
58	FootballStats	No	Original file had incorrect bracketing and could not be analysed Modified file is simple array with no issues	1	14	2
59	Bikes	No	Simple array - no issues	1	9	31
60	Athletes	No	Very large file so slow to load / analyse Simple array - straightforward to analyse	1	7	1587
61	Athletics	No	Shortened version of large downloaded file with 1 subarray. Reduced in size to speed up loading time. Original file only had 1 field 'Date' in main table. Added ID field to make this work properly.	2	9	7
62	PostInfoLevel3	Yes	File created with 3 nested levels of subarrays - posts (L1), comments (L2), feedback (L3) This level of complexity can't yet be managed by the analyser: 1. Analysis doesn't detect all fields or level 3 array (responses) correctly. 2 spurious fields detected (items 10/11). Treated as special case to fix analysis errors. 2. However, after correcting analysis, also needed to: a) create final table (responses) manually b) modify code for populating final table	4	9	2
63	Chess	No	Simple array - no issues	1	4	30
64	Badeye	Yes	File supplied by AWF user Badeye but with errors (now fixed) Contains 5 separate level 1 subarrays Treated as a special case to remove 1 spurious field and add another not detected	6	16	4
65	Roads	Yes	File contains 1 unnecessary subarray: 'location'. Treated as special case so it can be 'flattened' into 1 table	1	5	94
66	SanDigital	Yes	Field supplied by Scott Reeves (SanDigital Tech). All required fields are contained in the 'data' subarray so this is treated as a single table 4 fields are not picked up by the analyser as these do not appear in each record. Treated as a special case to include the 'missing' fields	1	19	79

## Appendix D – More Files

Another 25+ files are also supplied in the **More Files** folder for anyone who wishes to try them.

Most of these have been looked at in depth, but not all

Some files have simple structures e.g. BookList / USZipCodes

Others are very complex and / or will need to be treated as special cases e.g. FamilyTree / Clothing

In certain cases, the file may be too complex for the parser to handle e.g. GoogleMapFIXED / Shares

Four of these files were invalid as supplied e.g. GoogleMap (original version), TumblrPhotos

These are included as validation errors can often be fixed using online tools such as **JSON Lint** or **JSON Editor**

File	Comments	Sub Arrays?	Valid?
BookList	Straightforward file	Yes	Yes
CityBikes	Large file with 2 subarrays	Yes	Yes
Clothing	Multiple levels of nested subarrays and many repeated field names	Yes	Yes
ContactInfo	Simple file but with errors	Yes	No
FamilyTree	Badly structured with repeated use of same field names	Yes	Yes
FootballResult	2 subarrays and repeated use of field names	Yes	Yes
FootballScores	Easy to read but very difficult to parse	Yes	Yes
GeoCoords	2 nested levels of subarrays Contains 3 fields called type in main array & 2 different subarrays.	Yes	Yes
Glossary	Deceptively complex file with multiple nested subarrays	Yes	Yes
GoogleMap	Original file as supplied by Google is invalid !!!	Yes	No
GoogleMapFIXED	File fixed using JSONLint but still complex to handle. Multiple subarrays	Yes	Yes
ImageData	2 separate subarrays & repeated field names	Yes	Yes
ITHelpDesk2/3/4	Same data as ITHelpDesk file but in different formats of varying complexity	No	Yes
Playlist	Multiple subarrays	Yes	Yes
ProductGrid	Relatively straightforward file with subarrays	Yes	Yes
ProductList	Relatively straightforward file with subarrays	Yes	Yes
RoutePlan	Another Google file with multiple subarrays and complex to handle	Yes	Yes
Shares	Badly designed file. Difficult to parse	Yes	Yes
TrackInfo	Relatively straightforward file with subarrays	Yes	Yes
TumblrPhotos	Lengthy file with validation errors	Yes	No
USZipCodes	Huge file so will take ages to load but should be straightforward	Yes	Yes
Widget	Relatively simple file but with validation errors	Yes	No
WordMenu	File with 2 nested subarrays	Yes	Yes
YouTube	File with 2 nested subarrays	Yes	Yes

Additional files may be provided from the website at a later date

There are numerous JSON files for a wide variety of topic areas available on the internet.

Typing e.g. 'Shares JSON' into Google will give a long list of possible data sources

These may be real live data or sample files

Unfortunately, many of the JSON files available online are either poorly structured or invalid.

In many cases, validation errors can be fixed by this application or using the **JSON Lint** online utility.

However, if the file structure is poor, the file will be unusable

Many sites will require you to register to obtain an **API key** required to download real data.

**API keys** are usually free but in certain cases you will have to purchase credits to download data.

You can also create your own JSON files using random data generator sites such as **Mockaroo**

## Appendix E - Possible Issues

Like any complex application, issues may occur from time to time.

The following explains some issues you may experience and possible solutions / work rounds:

Issue	Solution / Work round
Subarrays enclosed in curly braces {} are not detected correctly	None at present. The JSON parser engine by Tim Hall can only handle subarrays enclosed in square brackets []
Level 3 arrays are not correctly detected by analyser	The analyser currently handles 3 levels of subarray. A future update may allow deeper array levels  At present, level 3 subarrays must be managed manually by treating as a special case
Subarray table code fails if an autonumber PK field from the main table is used as the foreign key in subsidiary tables	Use a different datatype for the main table PK field
The application doesn't automatically create subdatasheets or relationships between main & subsidiary tables	Planned for a future version For now, this must be done manually if required
I can't delete a table as its marked as in use	Select a different record using the navigation controls then return to the original record before deleting the table.  In rare cases, you may need to close & reopen the app to release the table 'lock'
I get an error 'the database has crashed' after creating several new tables, closing then reopening the database	Repeatedly creating & deleting tables can cause database bloat and instability (especially if the JSON files contain errors)  Decompiling the database followed by recompiling and compacting should fix this issue  Also, regular backups should be made
I would like to view the JSON file in a tree view format	Planned for a future version
I need more guidance to make the best use of this application	A series of You Tube videos are in preparation. Links will be provided from the application website

Please email [support@mendipdatasystems.co.uk](mailto:support@mendipdatasystems.co.uk) to alert us if you experience any other issues.