

WhereScape Source Enablement Pack - File Parser

This is a guide for installing Source Enablement Packs for WhereScape RED 8.6.6.1 or higher

Prerequisites

- Python 3.8
 - Download python installer from <https://www.python.org/downloads/>
 - Select "Add Python 3.8 to PATH" from installation Window
- PIP Manager
 - From Command Prompt (Run As Administrator) run below command

PIP Manager Install

```
python -m pip install --upgrade pip
```

- Python Packages
 - From Command Prompt (Run As Administrator) run below command -

Install Python Package

```
pip install pandas fastavro openpyxl xlswriter xlrd pyarrow fastparquet pyorc avro  
avro_python3 jsonpath_ng openpyxl Pillow pyarrow xmltodict lxml  
pip install --upgrade pandas
```

```
Amazon S3  
pip install boto3
```

```
Azure DataLake Storage Gen2  
python -m pip install azure-storage==0.36.0  
python -m pip install azure-storage-file-datalake
```

```
Google Cloud  
python -m pip install --upgrade gcloud  
python -m pip install google_api_python_client google_auth_oauthlib protobuf google-cloud-  
core google-cloud-datastore google-cloud-storage
```

NOTE: Above mentioned python packages can be installed by running install_WslPython_Modules.bat(refer to section *Enablement Pack Setup Scripts*.)

Enablement Pack Setup Scripts

The Enablement Pack Install process is entirely driven by scripts. The below table outlines these scripts, their purpose and if "Run as Administrator" is required.

#	Enablement Pack Setup Scripts	Script Purpose	Run as Admin	Intended Application
1	Setup_Enablement_Pack.ps1	Installs or updates source enablement pack in existing RED Metadata Repository for target database Installs Python scripts and UI Config Files for browsing files from Windows, Amazon S3, Azure Datalake Storage gen2, Google Cloud Storage	Yes	New and Existing installations
2	install_WslPython_Modules.bat	Installs or updates WslPython Modules and required Python libraries on this machine Installs required python packages for Amazon S3, Azure Datalake Storage gen2, Google Cloud Storage mentioned in <i>Prerequisites</i> section	Yes	New and Existing installations

Powershell script above provides some help at the command line, this can be output by passing the "-help" parameter to the script.

Note that on some systems executing Windows Powershell scripts is disabled by default, see troubleshooting for workarounds

Source Enablement Pack Installation

Installation Script to existing target database repository

Run Windows Powershell as Administrator

Install Source Connectivity Packs

```
<Script1 Location > Powershell -ExecutionPolicy Bypass -File .\Setup_Enablement_Pack.ps1
```

Important Upgrade Notes

This enablement pack will overwrite any existing Source Enablement Pack UI Configs:

Connection UI Config	Load UI Config
Amazon S3	Load From Amazon S3
Azure Data Lake Storage Gen2	Load From Azure Data Lake Storage Gen2
Google Cloud	Load From Google Cloud

To ensure existing Source Enablement Pack connections and associated Load Tables continue to browse and load:

Go into UI Configuration Maintenance in RED prior to installing this Enablement Pack and rename the affected UI Configurations. While the updated Load Template will work with previous Source Enablement Pack's we recommend moving these previous versions of Load Tables to newly created Parser based connections following this install. The earlier versions of the Source Enablement Pack will be deprecated following this release.

File Parser Connection Setup

Post install checks:

1. File Parser Browse Script - In RED ensure the File Parser Browse Script was installed, under the Host Scripts object tree node check for the object named: 'Browse_File_Parser'
2. UI Configurations - In RED check the Menu: 'Tools->UI Configurations->Maintain UI Configurations' for the appropriate UI Configurations*.

**Note: UI Configurations generally come in sets of 2 or 3 for a particular source type, a minimum set will have both a Connection UI Config and a Load UI Config, optionally a Column UI config may also be included.*

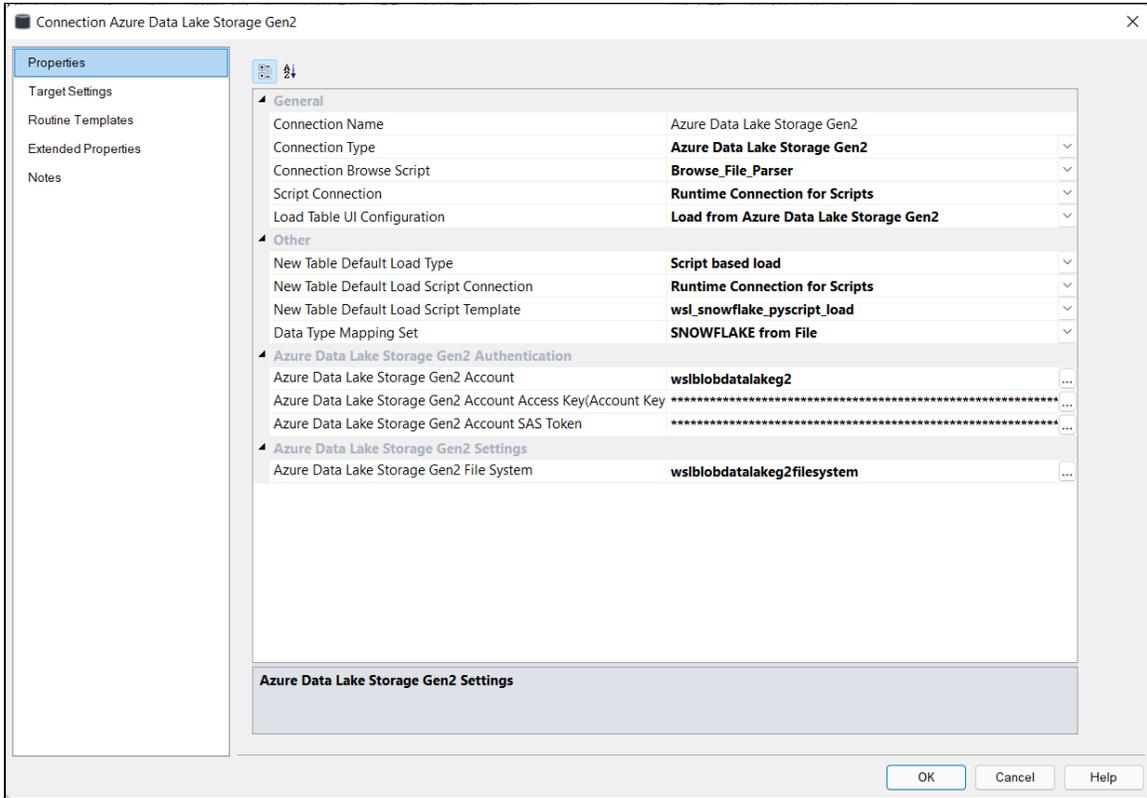
Amazon S3 Connection Setup

The screenshot shows the 'Connection Amazon S3' configuration window. The left sidebar contains 'Properties', 'Target Settings', 'Routine Templates', 'Extended Properties', and 'Notes'. The main area is divided into sections:

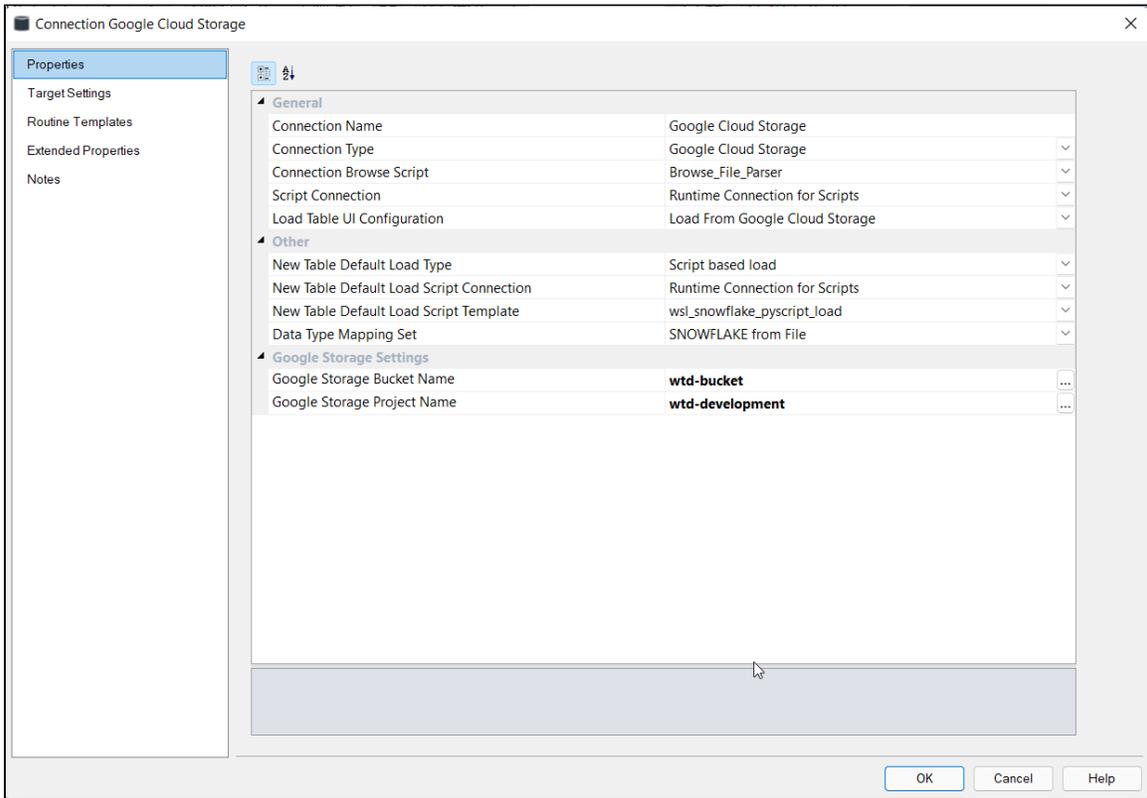
- General:** Connection Name (Amazon S3), Connection Type (Amazon S3), Connection Browse Script (Browse_File_Parser), Script Connection (Runtime Connection for Scripts), Load Table UI Configuration (Load from Amazon S3).
- Other:** New Table Default Load Type (Script based load), New Table Default Load Script Connection (Runtime Connection for Scripts), New Table Default Load Script Template (wsl_snowflake_pyscript_load), Data Type Mapping Set (SNOWFLAKE from File).
- S3 Settings:** S3 Bucket Name (iderabucket), S3 Region (us-east-1).
- S3 Authentication:** Access Key (AKIA533YNQGESIVPXJUD), Secret Key (masked with asterisks).

Buttons for 'OK', 'Cancel', and 'Help' are at the bottom right.

Azure Data Lake Storage Gen2 Connection Setup

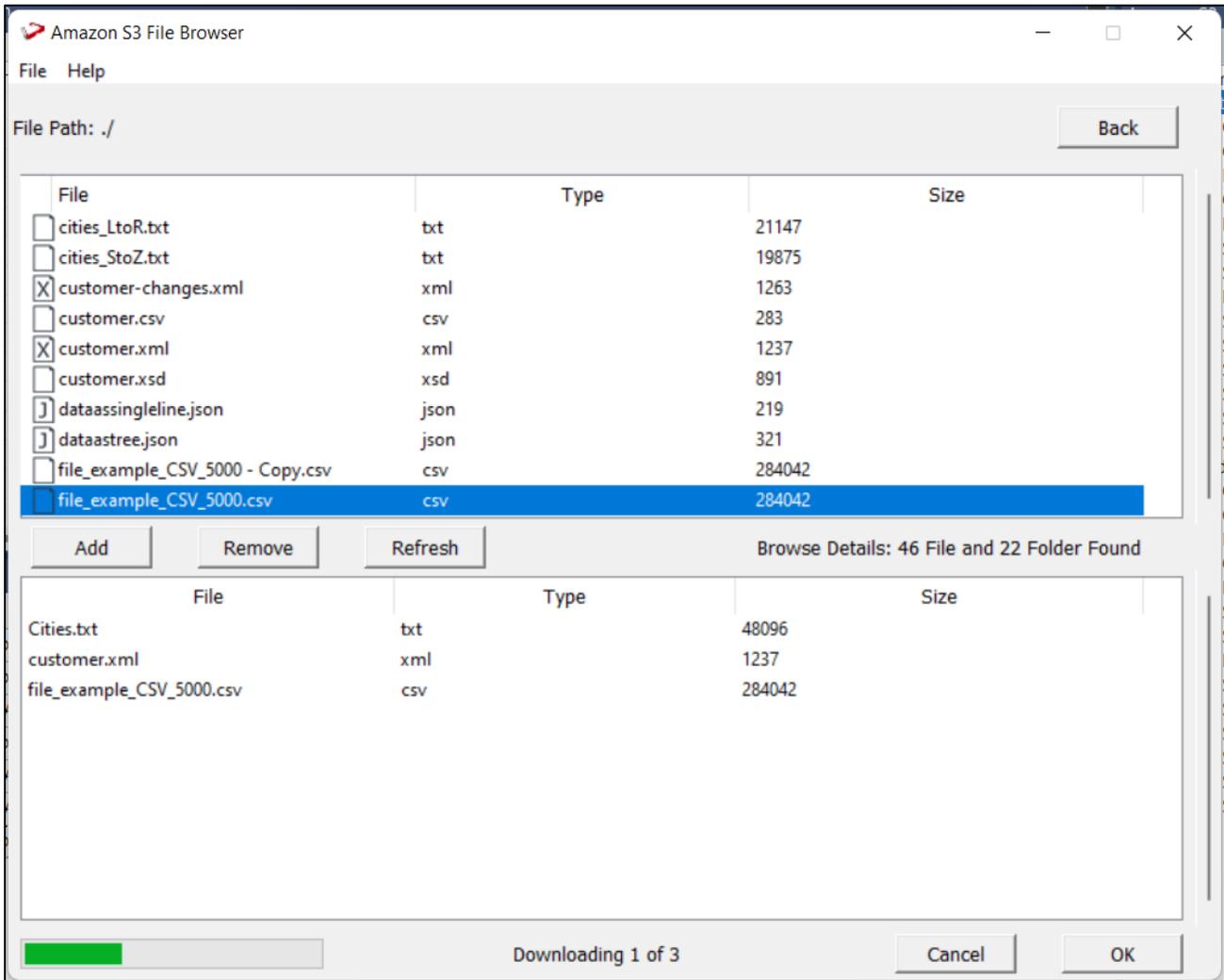


Google Cloud Connection Setup



NOTE: For google cloud, Install and configure Google Cloud SDK

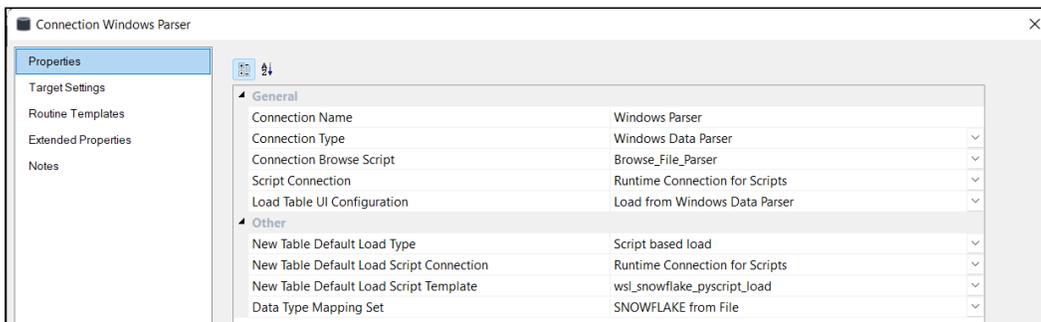
Cloud Browser



1. Select and click Add to copy files to staging area.
2. Click Back to navigate to previous directory
3. Click OK to download files for parsing.

Windows Parser Connection Setup

1. Login to RED
2. Check **Host Script** - Browse_File_Parser.py in objects list.
3. Check UI Configurations in Menu, Tools UI Configurations Maintain UI Configurations
4. Create new connection in RED
5. Select properties as shown in below screenshot



Browse Parser

Choose parser as per file type

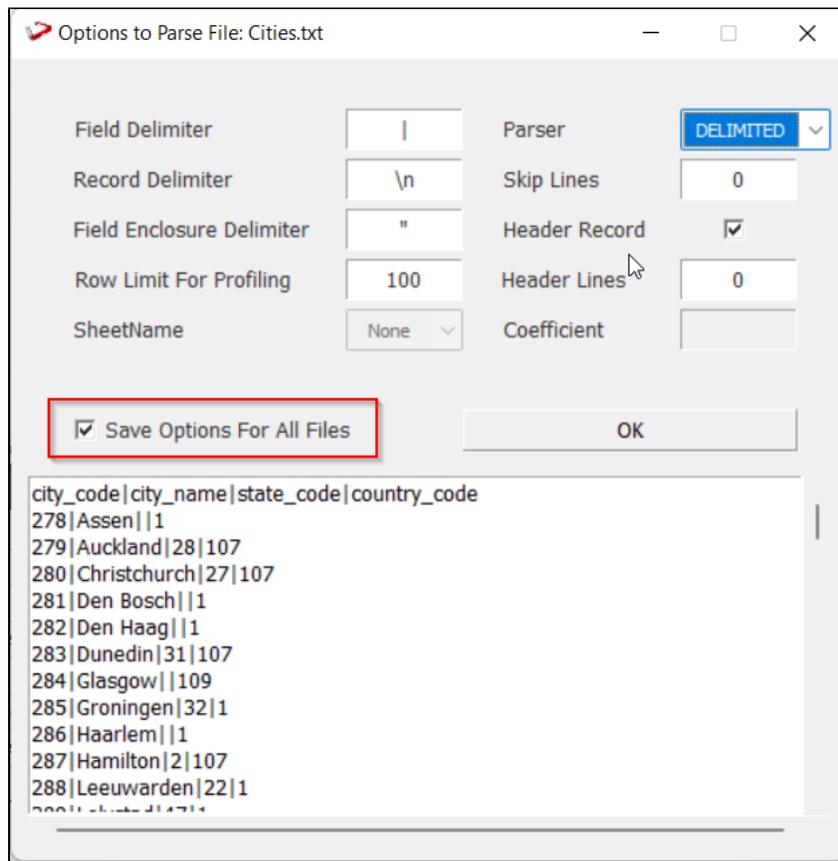
Options to Parse File: Cities.txt

Field Delimiter		Parser	DELIMITED
Record Delimiter	\n	Skip Lines	0
Field Enclosure Delimiter	"	Header Record	<input type="checkbox"/>
Row Limit For Profiling	100	Header Lines	0
SheetName	None	Coefficient	

Save Options For All Files OK

```
city_code|city_name|state_code|country_code
278|Assen||1
279|Auckland|28|107
280|Christchurch|27|107
281|Den Bosch||1
282|Den Haag||1
283|Dunedin|31|107
284|Glasgow||109
285|Groningen|32|1
286|Haarlem||1
287|Hamilton|2|107
288|Leeuwarden|22|1
```

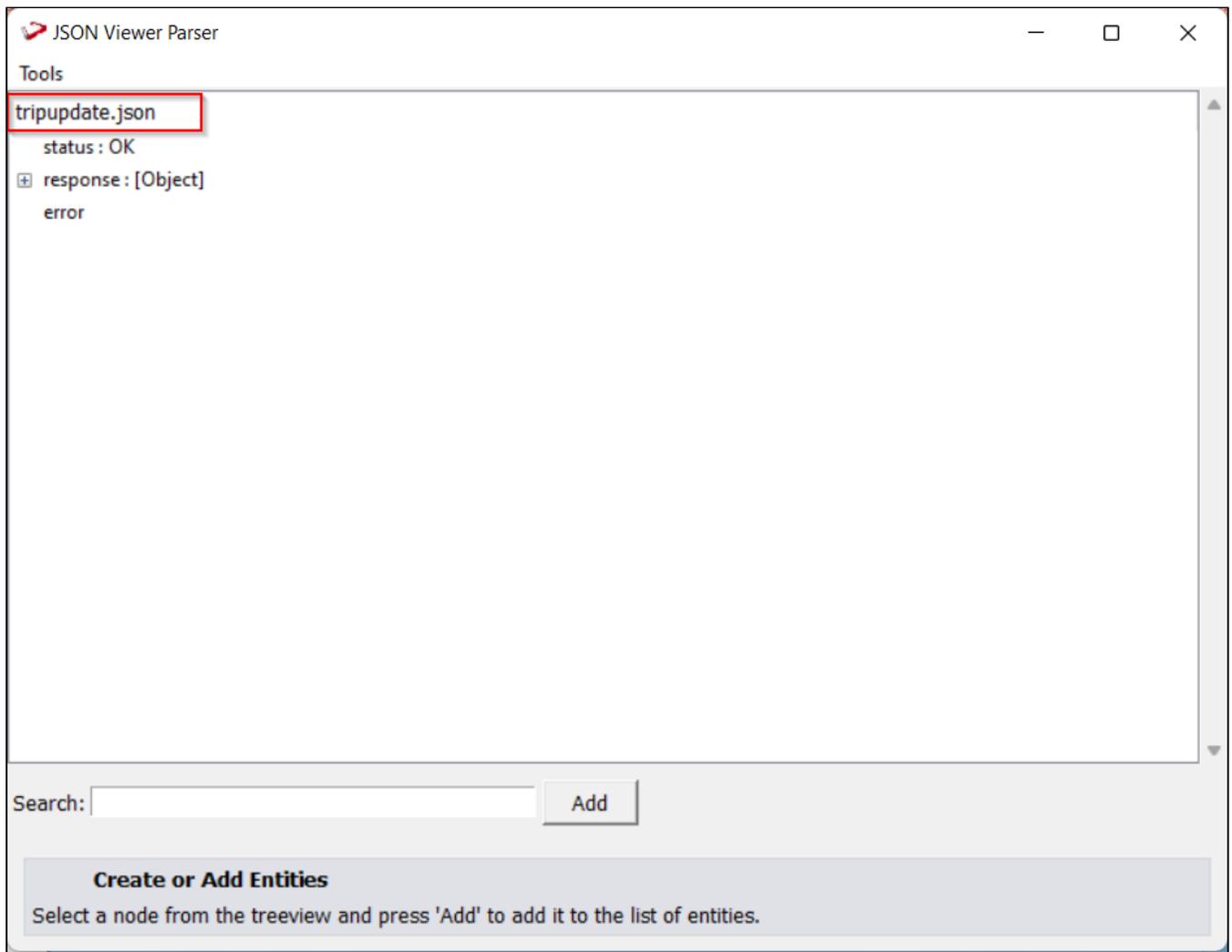
If the files are of same type and parsing options are same, check highlighted box to save same options.



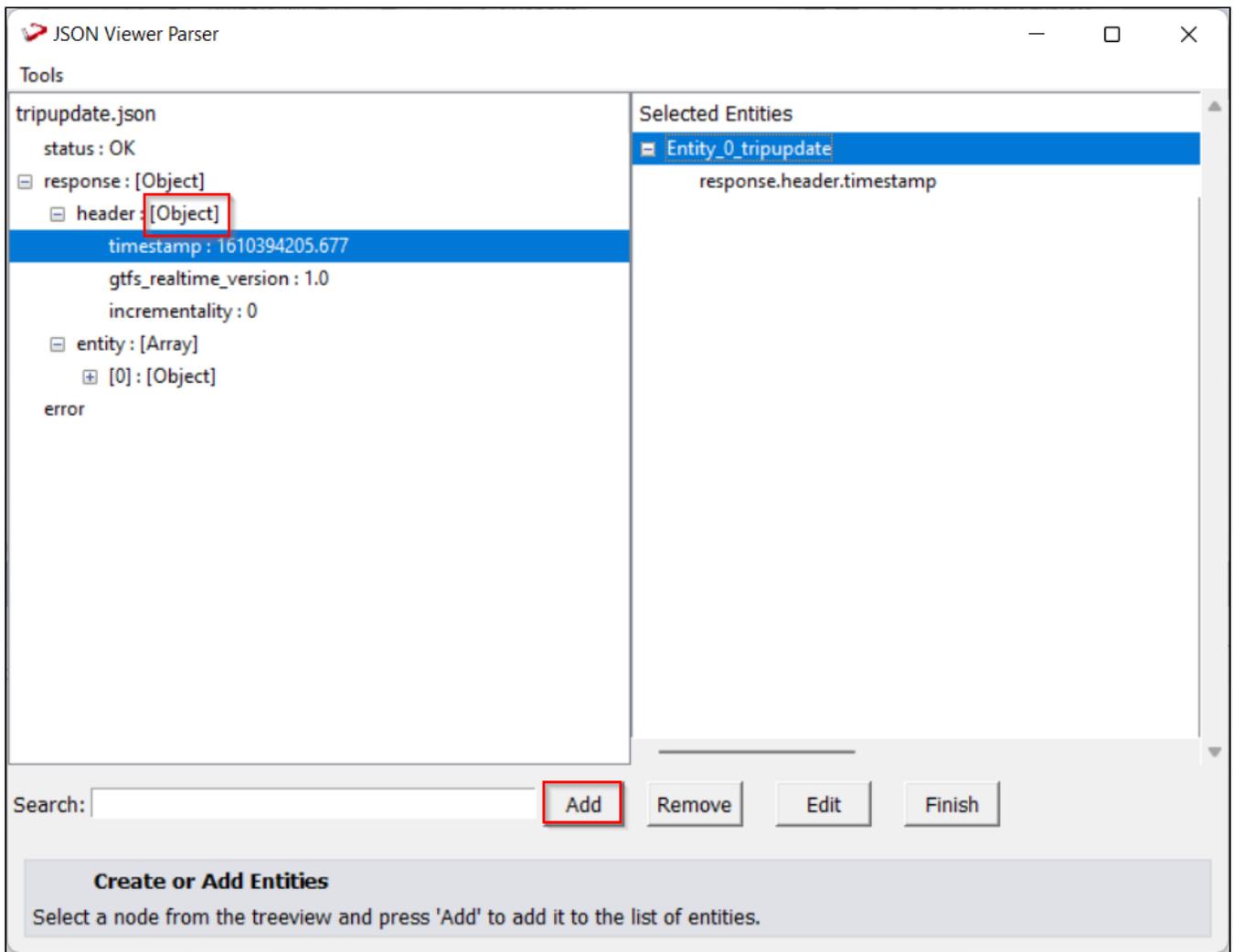
Parser for JSON and XML Files

The JSON parser GUI's main pane, The file name is highlighted, and the JSON tree structure is shown below it.

Hovering the cursor over any widget or element in the GUI will display information about that widget or element in the bottom help box.

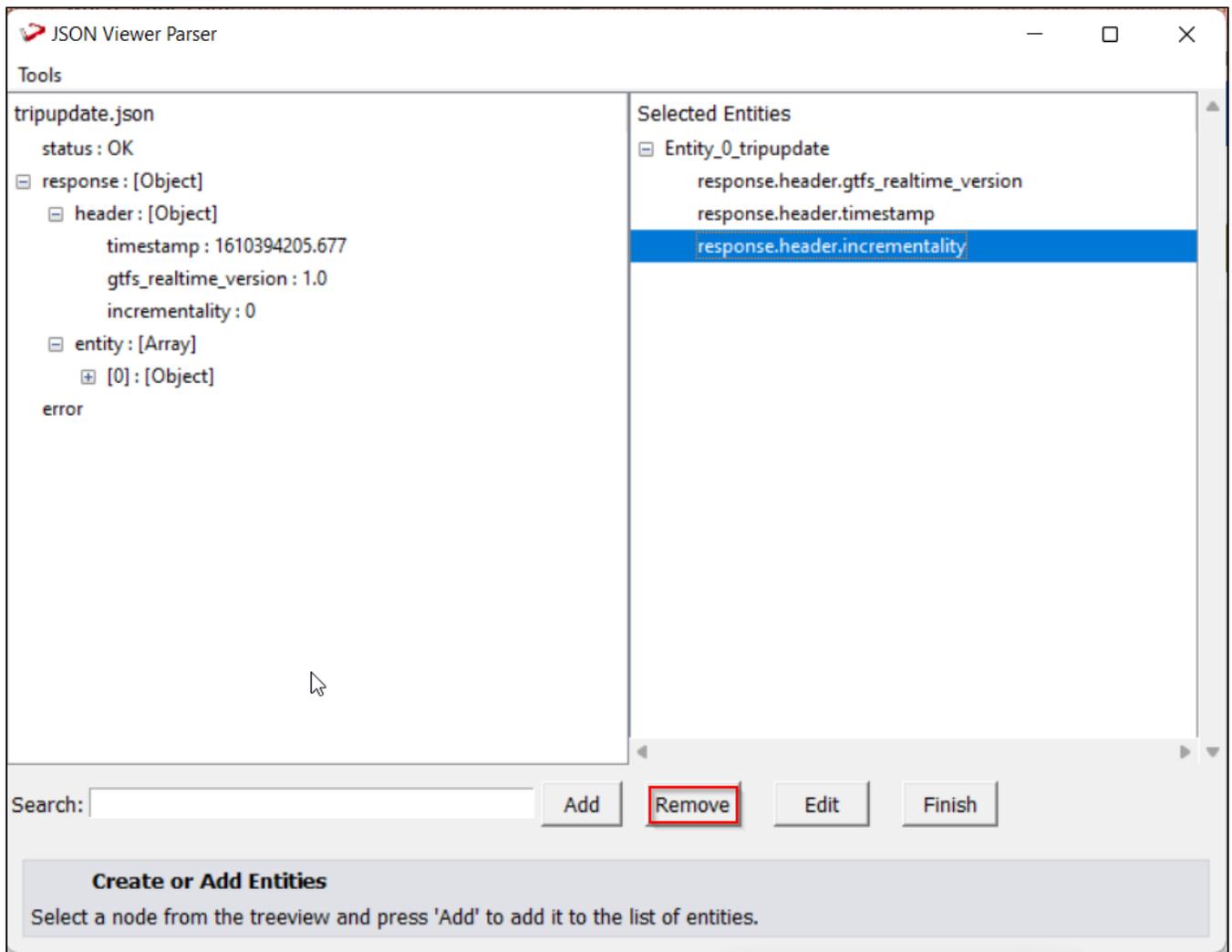


Select any node in the JSON tree and press the "Add" button at the bottom to create a new entity. On the right side of the window, a new pane will appear. The name of the new entity will be "Entity 0" by default. If the selected node is a leaf node (key value pair), this new entity will include only its key; if the selected node is an object or array, this new entity will include all of its children. The data type of the node is highlighted in the figure below.



Select the entity and use the "Remove" button to remove any specific node. To remove the entire entity object, choose the primary node (for example, Entity_0) and press the "Remove" button in the same way.

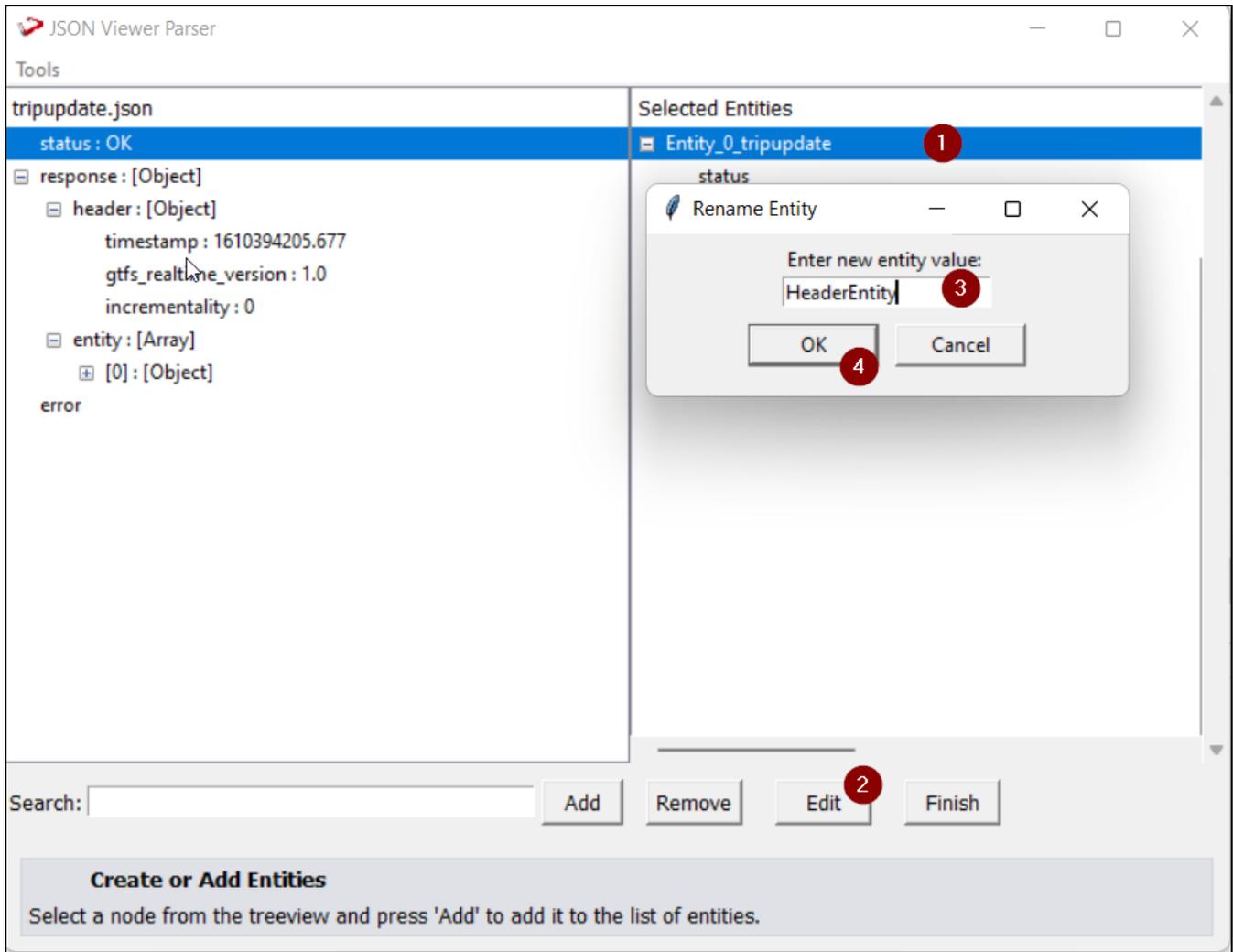
Note: Holding the "Ctrl" key on the keyboard while clicking on different nodes allows the user to select multiple nodes.



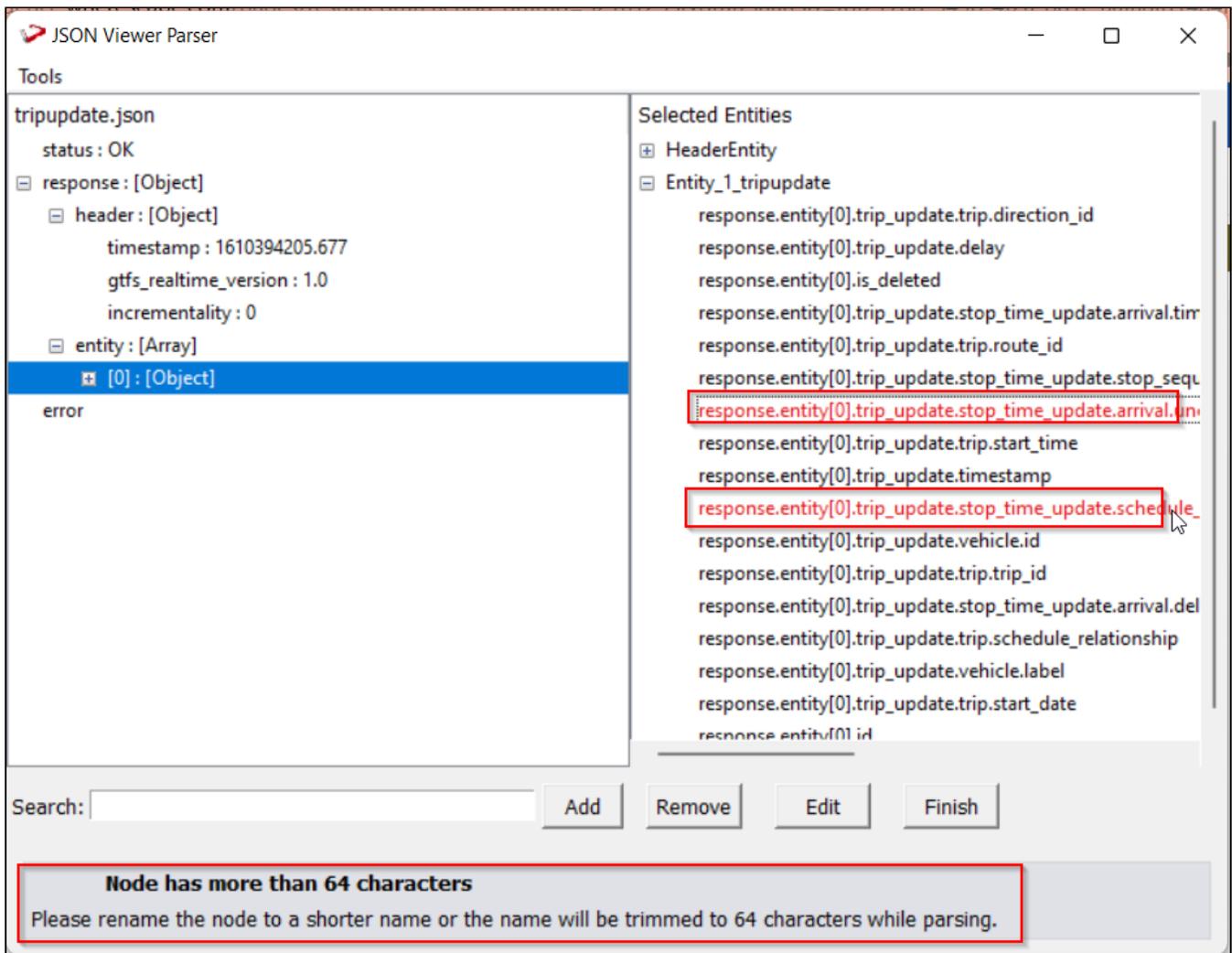
To add a new node from a JSON tree to an entity that has already been created. Select the node in the JSON tree to which the node should be added (Example: Entity_0), select one or more nodes, and click "Add" to add the selected node to the selected entity.

The screenshot shows the JSON Viewer Parser application window. The title bar reads "JSON Viewer Parser" with standard window controls. The main area is divided into two panes. The left pane, titled "Tools", displays a tree view of a JSON file named "tripupdate.json". The root node is "status : OK", which is highlighted in blue. Below it is a "response : [Object]" node, which is expanded to show "header : [Object]" and "entity : [Array]". The "header" node contains "timestamp : 1610394205.677", "gtfs_realtime_version : 1.0", and "incrementality : 0". The "entity" node contains an array with one element "[0] : [Object]". Below the tree view is an "error" label. The right pane, titled "Selected Entities", shows a list of selected nodes: "Entity_0_tripupdate" (expanded) and "status" (selected and highlighted in blue). Below the list are the paths "response.header.gtfs_realtime_version" and "response.header.timestamp". At the bottom of the window, there is a "Search:" text box followed by "Add", "Remove", "Edit", and "Finish" buttons. Below these buttons is a grey box with the text "Create or Add Entities" and "Select a node from the treeview and press 'Add' to add it to the list of entities."

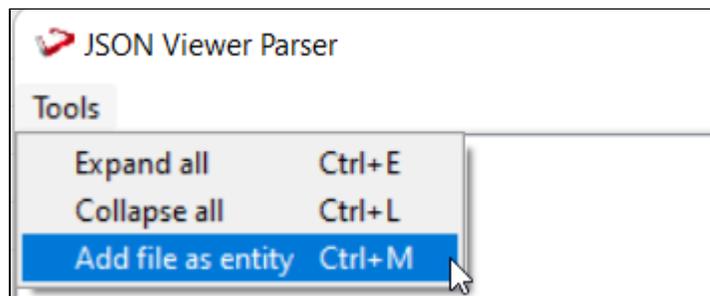
To edit name of Entity, Select the entity and press the "Edit" button to change the name. This will open a window with a text box where you may type in the new name for that object and then click "Ok."



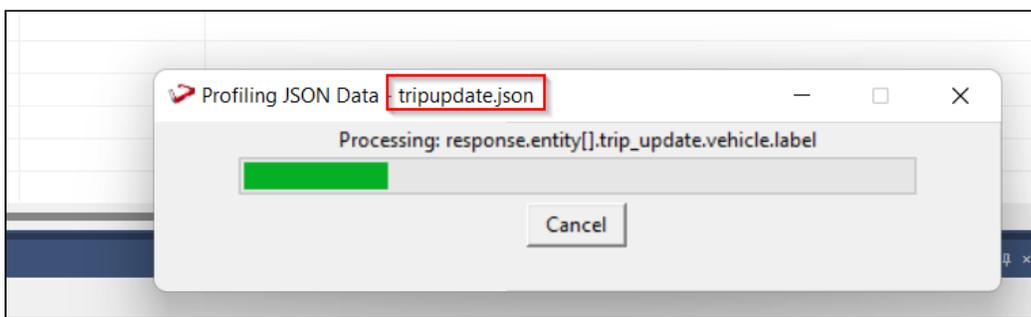
In the selected entities pane, nodes in entities with more than 64 characters are colored "red." In WhereScape RED, the names of these "red" colored nodes are trimmed.



To add complete file for profiling select below option from Tools Menu



After selecting all the entities and files options, progress of the profiling will appear with the progress bar and can be canceled at any point.



The working for XML parser is similar to JSON parser explained above.

Troubleshooting and Tips

Run As Administrator

Press the Windows Key on your keyboard and start typing cmd.exe, when the cmd.exe icon shows up in the search list right click it to bring up the context menu, select "Run As Administrator"

Now you have an admin prompt navigate to the folder where you have unpacked your WhereScape Source Enablement Pack to using the 'cd' command:

```
C:\Windows\system32> cd <full path to the unpacked folder>
```

Run Powershell (.ps1) scripts from the administrator prompt by typing the Powershell run script command, for example:

```
C:\temp\EnablementPack> Powershell -ExecutionPolicy Bypass -File .\install_New_RED_Repository.ps1
```

Notes: In the event you can not bypass the Powershell execution policy due to group policies you can instead try "-ExecutionPolicy RemoteSigned" which should allow unsigned local scripts.

Windows Powershell Script Execution

On some systems Windows Powershell script execution is disabled by default. There are a number of workarounds for this which can be found by searching the term "Powershell Execution Policy".

Here is the most common workaround which WhereScape suggests, which does not permanently change the execution rights:

Start a Windows CMD prompt as Administrator, change directory to your script directory and run the WhereScape Powershell scripts with this command:

- cmd:>Powershell -ExecutionPolicy Bypass -File .\<script_file_name.ps1>

Restarting failed scripts

Some of the setup scripts will track each step and output the step number when there is a failure. To restart from the failed step (or to skip the step) provide the parameter "-startAtStep <step number>" to the script.

Example:

```
Powershell -ExecutionPolicy Bypass -File .\<script_file_name.ps1> -startAtStep 123
```

Tip: to avoid having to provide all the parameters again you can copy the full command line with parameters from the first "INFO" message from the beginning of the console output.

If a valid RED installation can not be found

If you have Red 8.6.6.1 or higher installed but the script install_New_RED_Repository.ps1 fails to find it on your system then you are most likely running PowerShell (x86) version which does not show installed 64 bit apps by default. Please open a 64 bit version of PowerShell instead and re-run the script