Command Reference: ReadTableFromJSON()

Read a table from a JSON file

Version 11.07.05, 2015-12-06

The `ReadTableFromJSON()` command reads a table from a JavaScript Object Notation (JSON) file. See the following for a description of JSON:

http://www.json.org/

An example of JSON returned from a web service is as follows. This example illustrates water quality monitoring locations from Colorado's Data Sharing Network web services retrieved with the `WebGet()` command and URI: http://awqmsws.goldsystems.com/api/MonitoringLocationsVer1?StateCode=CO.


```
[
    {   "CountyName":"Adams",
        "Huc8":"","Huc12":"",
        "MonitoringLocationIdentifier":"NFE",
        "Latitude":"39.812806",
        "Longitude":"-104.954333",
        "MonitoringLocationName":"Metro Wastewater Reclamation District North Final Effluent - CO-0026638",
        "OrganizationIdentifier":"MWRD_WQX",
        "ResultSummaries":[],
        "StateCode":"CO",
        "MonitoringLocationType":"Facility Municipal Sewage (POTW)",
        "WaterbodyName":"","WatershedManagementUnit":""
    },
    {   "CountyName":"Adams",
        "Huc8":"","Huc12":"",
        "MonitoringLocationIdentifier":"SFE",
        "Latitude":"39.812772",
        "Longitude":"-104.95444",
        "MonitoringLocationName":"Metro Wastewater Reclamation District South Final Effluent - CO-0026638",
        "OrganizationIdentifier":"MWRD_WQX",
        "ResultSummaries":[],
        "StateCode":"CO",
        "MonitoringLocationType":
        "Facility Municipal Sewage (POTW)",
        "WaterbodyName":"","WatershedManagementUnit":""
    }
]
```


The following example shows water quality data retrieve using the following URI: http://awqmsws.goldsystems.com/api/ResultsVer1?MonitoringLocationIdentifiersCsv=UPSTREAM&OrganizationIdentifiersCsv=AURORA_WQX


```
[
  { "ActivityIdentifier":"UPSTREAM10/7/199817-OCT-980",
    "ActivityType":"Sample-Routine",
    "AssemblageSampled":"",
    "BottomDepthHeight":"",
    "BottomDepthHeightUnit":"",
    "CountyName":"Adams",
    "DepthAltitudeReferencePoint":"",
    "DepthHeight":"",
    "DepthHeightUnit":"",
    "MediaName":"Water",
    "MediaSubdivisionName":"",
    "MonitoringLocationIdentifier":"UPSTREAM",
    "OrganizationIdentifier":"AURORA_WQX",
    "Projects":
        [
          { "ProjectIdentifier":"SP CURE"}
        ],
```

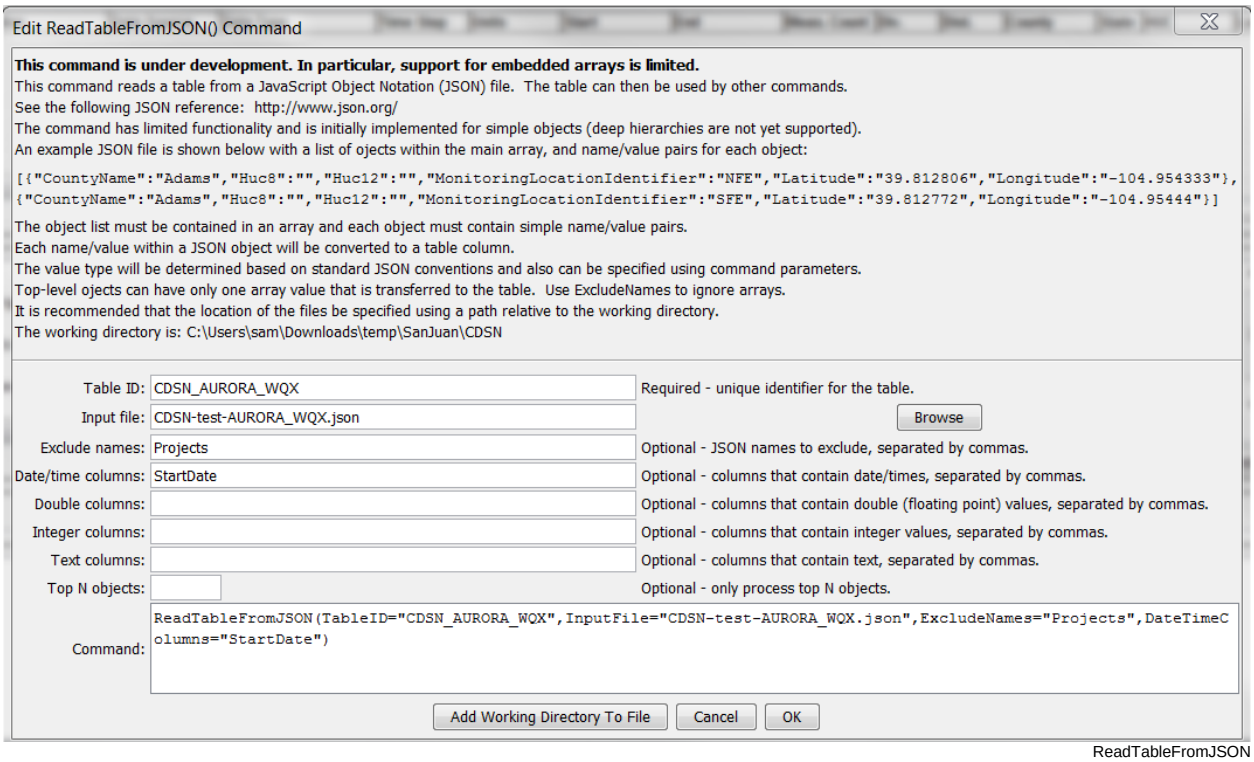
```
   "Results":
        [
          { "AnalyticalMethodContext":"APHA",
            "AnalyticalMethodIdentifier":"4500-NH3(D)",
            "BiologicalIntent":"",
            "CharacteristicName":"Ammonia",
            "Comment":"",
            "DataLoggerLine":"",
            "DepthHeight":"",
            "DepthHeightUnit":"",
            "DetectionCondition":"Not Detected",
            "MethodSpeciation":"as NH3",
            "ParticleSizeBasis":"",
            "QualifierCode":"",
            "ResultUnit":"mg/l",
            "ResultValue":"<0.05",
            "SampleFraction":"Total",
            "StatisticalBaseCode":"",
            "Status":"Final",
            "TaxonomicName":"",
            "TemperatureBasis":"",
            "TimeBasis":"",
            "ValueType":"Actual",
            "WeightBasis":""
          },
          {
            "AnalyticalMethodContext":"APHA",
            "AnalyticalMethodIdentifier":"9222D",
            "BiologicalIntent":"",
            "CharacteristicName":"Fecal Coliform",
            "Comment":"",
            "DataLoggerLine":"",
            "DepthHeight":"",
            "DepthHeightUnit":"",
            "DetectionCondition":"",
            "MethodSpeciation":"",
            "ParticleSizeBasis":"",
            "QualifierCode":"",
            "ResultUnit":"#/100ml",
            "ResultValue":"300",
            "SampleFraction":"",
            "StatisticalBaseCode":"",
            "Status":"Final",
            "TaxonomicName":"",
            "TemperatureBasis":"",
            "TimeBasis":"",
            "ValueType":"Actual",
            "WeightBasis”:””
       ],
    "SampleCollectionEquipmentComment":"",
    "SampleCollectionEquipmentName":"Miscellaneous (Other)",
    "SampleCollectionMethodIdentifier":"GRAB",
    StartDate":"1998-10-07",
    "StateCode":"CO",
    "StartTime":"",
    "StartTimeZone":"",
    "TopDepthHeight":"",
    "TopDepthHeightUnit":""
  }
]
```


JSON is a hierarchical representation that can have multiple nesting levels. Arrays are indicated by `[ ]`, objects are indicated by `{ }` and name:value pairs define data for an object. However, a table is a flat data structure, which requires that some JSON data values are repeated in the output rows. The command has the following functionality:

- All JSON names are converted to table columns. Currently processing of names is case-independent and redundant names will overwrite ("NAME1" is equivalent to "name1").
- Types for table columns are determined by examining the JSON structure:
  - Quoted values are interpreted to be strings.
  - Booleans are handled directly.
  - Numbers default to double precision output.
  - Command parameters `DateTimeColumns`, `DoubleColumns`, `IntegerColumns`, and `TextColumns` can be used to override the default data type mapping.
- Currently only one top-level embedded array can be handled (see highlighted `Projects` and `Results` names above), with the 2nd and following arrays being ignored. In the future, full recursion to flatten "deep" objects may be implemented. When the object's array is encountered:
  - separate table rows will be added for each array item
  - columns prior to the array will be filled with the first rows associated with the object to repeat content
  - columns after the array will similarly be filled by repeating content

  In other words, the array determines the number of rows for the object and some column values are repeated to fill out the duplicate rows. This results in a flat table representation of JSON where some column content is repeated. Uniqueness for the row must then be determined by considering array values.
- The `ExcludeNames` parameter is provided to exclude JSON names from processing. Because only one top-level object array is currently supported, this parameter can be used to exclude arrays to ensure that the desired array to process is found. For example, use `ExcludeNames=Projects` to ensure that the `Results` array is processed into output as the only array allowed for top-level objects.

The following dialog is used to edit the command and illustrates the syntax for the command.



ReadTableFromJSON

ReadTableFromJSON() Command Editor

The command syntax is as follows:

```
ReadTableFromJSON(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
`TableID`	Identifier to assign to the table that is read, which allows the table data to be used with other commands. Can be specified using processor `${Property}`.	None – must be specified.
`InputFile`	The name of the file to read, as an absolute path or relative to the command file location. Can be specified using processor `${Property}`.	None – must be specified.
`ExcludeNames`	List of JSON names to exclude from table, separated by commas.	Include all names.
`DateTimeColumns`	List of comma-separated column names for columns that should be treated as containing date/time values.	Automatically determine column types from data.
`DoubleColumns`	List of comma-separated column names for columns that should be treated as containing double-precision (floating point number) values.	Automatically determine column types from data.
`IntegerColumns`	List of comma-separated column names for columns that should be treated as containing integer values.	Automatically determine column types from data.
`TextColumns`	List of comma-separated column names for columns that should be treated as containing text values.	Automatically determine column types from data.
`Top`	Specify the number of data rows to read, useful when prototyping an analysis process.	Process all rows.