



CONTACT SUPPORT:

COMPANY NAME:	Control Concepts Inc.
SUPPORT CONTACT:	Elizabeth Scozzari
EMAIL ADDRESS:	support@controlconcepts.net
PHONE:	(201) 797-7900
ADDRESS:	336 Route 46, Fairfield, NJ 07004

GENERAL INFORMATION

SIMPLWINDOWS NAME:	JSON FileProcessor v1.0.1
CATEGORY:	Utility
VERSION:	1.0.1
SUMMARY:	Module provides a method by which to arbitrarily parse JSON tokens from a file loaded to processor memory and to write stored values to the same file.
GENERAL NOTES:	<p>Each Json file to be parsed must have its own FileProcessor module with a unique Processor ID and a local path to the Json file. Each signal type has a module that may be utilized to parse Json Properties and to assign them to a SimplWindows signal type. Each module can parse up to 10 tokens, but you may have multiple instances of each module type. Parsing modules associated with a FileProcessor module must have matching Processor ID parameters.</p> <p>Data can only be written to a file once it has been 'Stored' by a writer module. 'Stored' data can be retrieved by pulsing 'Read' even if it has not been written yet. This allows for non-persistent data to be stored without writing. This can also be used to minimize write cycles to the onboard memory, prolonging its life. Please note that values 'Stored' but not 'Written' are NOT persistent and will be lost if the program restarts.</p> <p>Please be aware that the reader module takes time to register all requester and writer modules. To this end, read and write requests are buffered until all requester and writer modules have been registered. This usually takes 5-15 seconds.</p> <p>If you do not place a json file on the processor corresponding to the json file path, a blank file will be created on first read or first write.</p> <p>For the demo program to work, the two Json file names must not change (though the path to the filename may change). They will automatically generate files for usage in the demo. Similarly, do not use filenames containing 'myDemo1.json' or 'myDemo2.json', as using those paths will create demo json files for use in the demo program.</p> <p>In addition, the demo program downloads images from the internet for usage in the xpanel. For this reason, for proper functioning of the demo, an internet connection is required. If no internet connection is present, the image locations on the xpanel will be blank.</p> <p>To learn more about what other utility modules are available from Control Concepts visit the CCI Utility Module Store.</p>
CRESTRON HARDWARE REQUIRED:	3-Series Processors only .
SETUP OF CRESTRON HARDWARE:	N/A



VENDOR FIRMWARE:	N/A
VENDOR SETUP:	N/A
CABLE DIAGRAM:	N/A



Control:

<u>Signal/Function Name</u>	<u>D,S,A</u>	<u>Digital, Serial, Analog signal property definition.</u>
Read_File	D	Triggers the file read and parse operation on the rising edge. Placing a '1' on this signal is valid, however not encouraged, as the parsing operation is fairly processor intensive. When combined with other processor intensive on-boot tasks in the program, it may cause sluggish operation until the parsing action has concluded.
Write File	D	Triggers the file write operation on the rising edge. Placing a '1' on this signal is not valid. Values will only be written if they have already been stored by a corresponding 'Writer' module.
Debug	D	Enables debugging trace statements for all associated 'Readers' and 'Writers' when high.



PARAMETERS:

Json File	S	Path to the JSON file that you wish to poll. Remember to use the escape character (\) as necessary in file paths.
Processor ID	A	Setting to indicate the ID for a specific Processor Module. Each 'Reader' and 'Writer' must be paired with an appropriate Processor via matching Processor ID fields.



TESTING:	
OPS USED FOR TESTING:	CP3: 1.601.3935.27221
SIMPL WINDOWS USED FOR TESTING:	4.14.20.00
DEVICE DB USED FOR TESTING:	200.00.015.00
CRES DB USED FOR TESTING:	200.00.004.00
SYMBOL LIBRARY USED FOR TESTING:	1112
REVISION HISTORY:	1.0.0 – Initial Release 1.0.1 – Fix read issue in .clz affecting 4-series processors