Templating Custom Node

Introduction

Custom Nodes are a new feature of Square 9's GlobalCapture and GlobalAction and are available from version 2.3.x. Custom Nodes are seen as a replacement for Call Assembly nodes and extend the functionality of GlobalCapture and GlobalAction but in way that is more familiar to workflow designers than the Call Assembly nodes.

The Templating Custom Node utilises a powerful template scripting language called Scriban to produce text based files from within your workflows.

Pre-Requisites

The following items are required:

- GlobalCapture or GlobalAction Version 2.3 or newer.
- Templating Custom Node - `e8142d89-f608-431c-8205-192053397386.s9n`.
- A license file.

Installation

To install the Templating Custom Node:

1. Log into GlobalCapture with an account that can administer GlobalCapture.
2. Navigate to `Mange >> Nodes`.
3. Click the Menu button in the bottom right corner of the interface and choose Upload S9N File (the middle option).
4. Browse for the Templating Custom Node Package - `e8142d89-f608-431c-8205-192053397386.s9n` and click open to install.

Default Configuration

To configure the default properties for the Templating Custom Node, click on the three-dot menu and select edit and the node's properties will be displayed.

There are two types of default configuration available:

1. If you click on Modal Preview and enter any values, these properties will then be pre-populated when the node is added to a workflow.
2. If you click on Config you are presented with a JSON view of the node's configuration. This contains the properties that can be on the node's config panel as well as properties you only need to set once per install, for example, the path to the license file.
   Any properties configured here will override the properties set on the node's config panel, configuring all nodes on all workflows to use this value. To remove an override, edit the property's value and set it to `null`.
   If you need assistance on how to edit a JSON file please contact support.

Once you have completed configuring any properties, click on save

Licensing

The Templating Custom Node requires a license to function. If it is added to a workflow without a license being present the workflow will be stopped and remain in an error state. Please contact sales to enquire about pricing and obtaining a license.

Once you have obtained a license file, save it somewhere safe, it also needs to be saved in a location that the login user for the GlobalCapture or GlobalAction Engine service(s) has access to. Next you need to configure the node's default config to point to this file by editing the value for the `licenseFilePath` property.

Note: As you can see in the sample value provided, any back slashes need to be escaped for the JSON to be valid, for example: `"C:\\Node License File\\app.license"`

Once you have completed configuring the license file path, click on save.

Workflow Configuration

The Templating Custom Node is used like any of the built-in nodes in the workflow designer, drag it from the node toolbox onto the designer's canvas, configure the properties and link it up as is appropriate for your workflow.

Configure the Templating Custom Node

- ### Header Template File Path
  The path to text file containing a header template. The header template is optional and will be inserted at the start of any new file created.
- ### Body Template File Path
  The path to a text file containing the body template.
- ### Output Directory Path
  The directory to write the output file to.
- ### Output File Name
  The name of the output file.
- ### Append To Existing
  - If selected
    The output file will be created if it does not exist and appended to if it does exist.
  - If not selected
    The output file will be created if it does not exist, and an incremental file name will be assigned if it does exist.

Templates

Scriban templates are extremely flexible and can be a mixture of fixed text and variable data from the workflow's process fields. The documentation for Scriban can be found here.

It is possible to access process field data by surrounding a reference to them in {{ moustache notation }}. When the output text is rendered, the variable reference will be replaced with the value from the field. The fields references can be specified in a long form or short form.

	Long Form	Short Form
Single Value Fields	{{ single_value_fields["FIELD NAME"] }}	{{ s["FIELD NAME"] }}
Multi Value Fields	{{ multi_value_fields["FIELD NAME"] }}	{{ m["FIELD NAME"] }}
Table Fields	{{ table_value_fields["TABLE NAME"] }}	{{ t["TABLE NAME"] }}

Examples

The examples have all been designed to work with the default data set found in our Template Tester utility. The Template tester allows you to test your templates and see what the output looks like without having to put a process through GlobalCapture or GlobalAction. You can copy these examples into the Template Tester to play around and become more familiar.

	Template	Output
Access a single value using long form	`Invoice Number is : {{ single_value_fields["Invoice Number"] }}`	`Invoice number is : I-1001`
Access three single values using short form	`{{ s["Net"] }},{{ s["VAT"] }},{{ s["Total"] }}`	`800,200,1000`

Loop through all single values and print their index, key and value	`{{ for sv in s }}`
	`[{{ for.index }}] {{ sv .key }} : {{ sv .value }}`
	`{{ end }}`
		`[0] Invoice Number : I-1001`
		`[1] Invoice Date : 2022-06-10T12:37:57.6231039Z`
		`[2] Supplier Name : Bob's Bits`
		`[3] Net : 800`
		`[4] VAT : 200`
		`[5] Total : 1000`

Modifying single values, performing some maths, and adding comments	`Invoice Number : {{ s["Invoice Number"] | string.replace( "-", ":") }}  {{`
	`# replace text }}`
	`Invoice Date : {{ date.parse s["Invoice Date"] | date.to_string "%g" }}  {{`
	`# convert date }}`
		`Invoice Number : I:1001`
		`Invoice Date :  10 Jun 2022`
		`Supplier Name : BOB'S BITS`
		`Net : 800`

	Supplier Name : {{ s["Supplier Name"] | string.upcase }} {{ # change case}} Net : {{ s["Net"] }} VAT : {{ s["VAT"] }} Total : {{ s["Total"] }} Checked : {{ (s["Net"] | string.to_float) + (s["VAT"] | string.to_float) == (s["Total"] | string.to_float) }} {{ # convert to number and do some maths then a comparison}}	VAT : 200 Total : 1000 Checked : true
Access the second value of a multi value	{{ m["Delivery Note Numbers"][1] }}	D-2002
Loop through all values in a multi value and remove extra newlines	{{~ for entry in m["Delivery Note Numbers"] ~}} {{ entry }} {{~ end ~}}	D-2001 D-2002 D-2003
Check two multi values are of equal size then join them in a CSV	{{~ if m["Delivery Note Numbers"].size != m["Purchase Order Numbers"].size ~}} Different Sizes {{~ else ~}} {{~ for entry in m["Delivery Note Numbers"] ~}} {{ entry }},{{ m["Purchase Order Numbers"][for.index] }} {{~ end ~}} {{~ end ~}}	D-2001,P-3001 D-2002,P-3002 D-2003,P-3003
Create a CSV from a table field	Description,VAT Code,Quantity,Unit Price,Line Total,Line Net {{~ for row in t["Line Items"] ~}} {{ row["Description"] }},{{ row["VAT Code"] }},{{ row["Quantity"] }},{{ row["Unit Price"] }},{{ row["Line Total"] }}, {{~ end ~}}	Description,VAT Code,Quantity,Unit Price,Line Total,Line Net Brazil Nuts,A,100,1.10,110.00, Hazelnuts,A,200,2.45,490.00, Peanuts,B,300,5.10,1530.00, Pistachio Nuts,C,400,0.90,360.00,

selectec

Create a CSV from a table field and use a function to look up the percentage of VAT from another table field then calculate the NET value	Description,VAT Code,Quantity,Unit Price,Line Total,Line Net,VAT {{~ func getNet(rate, total) ~}} {{~ for row in t["VAT Rates"] ~}} {{~ if row["VAT Code"] == rate ~}} {{~ (total - (((row["VAT Rate"] | string.to_float) / 100) * total)) ~}},{{~ (((row["VAT Rate"] | string.to_float) / 100) * total) ~}} {{~ end ~}} {{~ end ~}} {{~ end ~}} {{~ for row in t["Line Items"] ~}} {{ row["Description"] }},{{ row["VAT Code"] }},{{ row["Quantity"] }},{{ row["Unit Price"] }},{{ row["Line Total"] }},{{ getNet row["VAT Code"] (row["Line Total"] | string.to_float) }} {{~ end ~}}	Description,VAT Code,Quantity,Unit Price,Line Total,Line Net,VAT Brazil Nuts,A,100,1.10,110.00,110,0 Hazelnuts,A,200,2.45,490.00,490,0 Peanuts,B,300,5.10,1530.00,1377,153 Pistachio Nuts,C,400,0.90,360.00,288,72

Contacts

Sales and licensing enquiries to: sales@selectec.com

Support enquiries to: support@selectec.com

Acknowledgements

Selectec Custom Nodes are made possible by open-source software. The following open-source software is distributed and is provided under other licences.

- Custom Workflow Nodes
  https://github.com/Square9Softworks/custom-workflow-nodes
- Nett
  https://github.com/paiden/Nett
- BouncyCastle
  http://www.bouncycastle.org/csharp/
- Newtonsoft.Json
  https://www.newtonsoft.com/json
- Scriban
  https://github.com/scriban/scriban

Thank you to the developers of these softwares.