**Parameters for MASTER menu options (in the form of comma-separated key-value pairs of a dict):**

- **table** tablename where form data will be stored in

e.g. 'table':['item']

- **pk** primary key of the 'tablename'

e.g. 'pk':['itemid']

- **cbo** (optional)

combobox|dropdown with values filled-in from the column of another table

columnname must be same in this master table and the other one

e.g.

'cbo':[''] #no combobox required OR

'cbo':['itemcategory.itemcategory'] OR

'cbo':['itemcategory.itemcategory','itemunit.itemunit']

- **autofill** (optional)

textfield to be filled-in automatically with the values from a column of another table (used to fill combobox)

whenever an item in the combobox|dropdown on the form is selected

e.g.

'autofill':[] #no autofill required OR

'autofill':['itemcategory#itemcategory.itemgstrate']

which will fill 'itemgstrate' textfield on the form with the value of 'itemgstrate' column of the 'itemcategory' table whenever an item in 'itemcategory' combobox|dropdown on the form is selected

**Parameters for TRANSACTION menu options (in the form of comma-separated key-value pairs of a dict):**

- **table** tablename where form data will be stored in

e.g. 'table':['itempurchase']

- **pk** primary key of the 'tablename'

e.g. 'pk':['purchaseid'],

- **dateColumn** date of the transaction; uses a GUI calendar to pick-up the date

e.g. 'dateColumn':'pdate', (do not use [])

- **pickup** crates two types of drop-downs on the form

(1) parent-child comboboxes

(2) stand-alone combobox

Sub-options:

- **masterCBO** creates the parent combobox; leave it blank, if no such parent drop-down is required

e.g.

'masterCBO':['item.itemcategory#item.itemname']

creates 'itemcategory' combobox whose value when selected, fills the child combobox

- **masterLookupItems** creates the child combobox; set it, if stand-alone drop-down is required

e.g.

'masterLookupItems':['item.itemname']

creates a combobox with values filled-in from 'itemname' column of the 'item' table

- **getvalfromothertableonlookupcboevent** (MAY NOT WORK IN THIS VERSION!)

textfields with values filled-in from yet another table when an item in the masterLookupItem combobox is selected

leave it blank!

e.g.

'getvalfromothertableonlookupcboevent' : []

- **masterPrimaryKeys** primary key of the child combo's or stand-alone combo's table

e.g.

'masterPrimaryKeys':['item.itemid']

- **masterAutofillValues** one or more text fields to be filled-in with the values from one or more tables whenever a value is selected in the child or standalone combobox

e.g.

'masterAutofillValues':['item.itemcode', 'item.itemaverageprice',

'item.itemtotalstock', 'item.itemunit', 'item.itemgstrate']

- **condition** condition to be checked in the masterLookupItems (child or stand-alone combobox) table while fetching records (autofill values) from that table whenever an item is selected in the masterLookupItems combobox

e.g.

'condition':['"itempurchase.itemstock > 0"']

condition can be checked for the column values matching the value of a control on the form during run-time

e.g.

'condition' : [' "itemcategory=\'" +itemcategory.get()+ "\'" ']

means "itemcategory" column's value matching with "itemcategory" control on the form.

Complete example of 'pickup' option along with its sub-options together:

'pickup' : [

{

'masterCBO' : ['item.itemcategory#item.itemname'],

'masterLookupItems' : ['item.itemname'],

'getvalfromothertableonlookupcboevent': [],

'masterPrimaryKeys' : ['item.itemid'],

'masterAutofillValues' : ['item.itemcode', 'item.itemaverageprice', 'item.itemtotalstock',

'item.itemunit', 'item.itemgstrate'],

'condition' : [],

},

],

- **transItems** textfields to be filled-in by the user manully; write columnnames without tablenames

e.g.

'transItems':['invoice','itemquantity','itemprice','fright']

- **expressions** textfields to be filled-in with the values of the specified expressions

LHS lvalues are columnnames without tablenames

RHS rvalues are columnnames.get() to pick the values filled-in the form which may be converted to float or int or may be rounded-off

e.g.

'expressions':[

'amount = float(itemquantity.get()) \* float(itemprice.get())',

'gst = round(float(amount.get()) \* float(itemgstrate.get()) / 100.00, 2)',

'netamount = float(amount.get()) + float(gst.get())',

'itemminsaleprice = round(float(grandamount.get()) / float(itemquantity.get()),2)',

'itemstock = float(itemquantity.get())',

],

- **displayonlynoinsert**

textfields with values, which are meant for display only on the form and are not inserted into the transaction table (MAY NOT WORK IN THIS VERSION!). Leave it blank!

e.g. 'displayonlynoinsert':[]

- **invisible** textfields to be filled-in which otherwise are not filled through other options or manually

e.g.

'invisible': [

'suppliercategory = suppliercategory.get()',

'suppliercode = suppliercode.get()',

],

- **prefilled** textfields prefilled (initialized) on form load with hard coded values rather than any value coming from a database table

e.g.

'prefilled':['fright=\"0.00\"'], #to set 0.00 as default inital value in the textfield 'fright'

- **masterUpdates** updates the "table.column" specified on LHS with the value/expression given on RHS

On RHS the expression, if any, contains columnname.get() without tablename which may be converted into float or int or may be rounded-off

e.g.

'masterUpdates' : [

'item.itemtotalstock=float(itemtotalstock.get())+float(itemquantity.get())',

'item.itemtotalstock=float(itemtotalstock.get())-float(itemquantity.get())',

]