



zen_Nucleus

json file format – Programmer’s guide

Release	Version	Date	Rédacteur	Commentaires
v3.0.00		11 avril 2024	OG	First version

This component is intended to create easily list forms, and Menus all attached to a user’s profile.

SUMMARY

Description	2
Lists file format	4
Menus file format	6



Description

json files

All the json files that manage Lists and menus are stored into to places.

Regular files are placed into a folder « zen_json » created near the datafile. This folder contains two folders, « lists_json » and « menus_json ». Each subFolder is exploded into folders with the name as the « profile » for a user.

Then you find files for lists and menus, with the table name for the lists, and just « menus » for a profile.

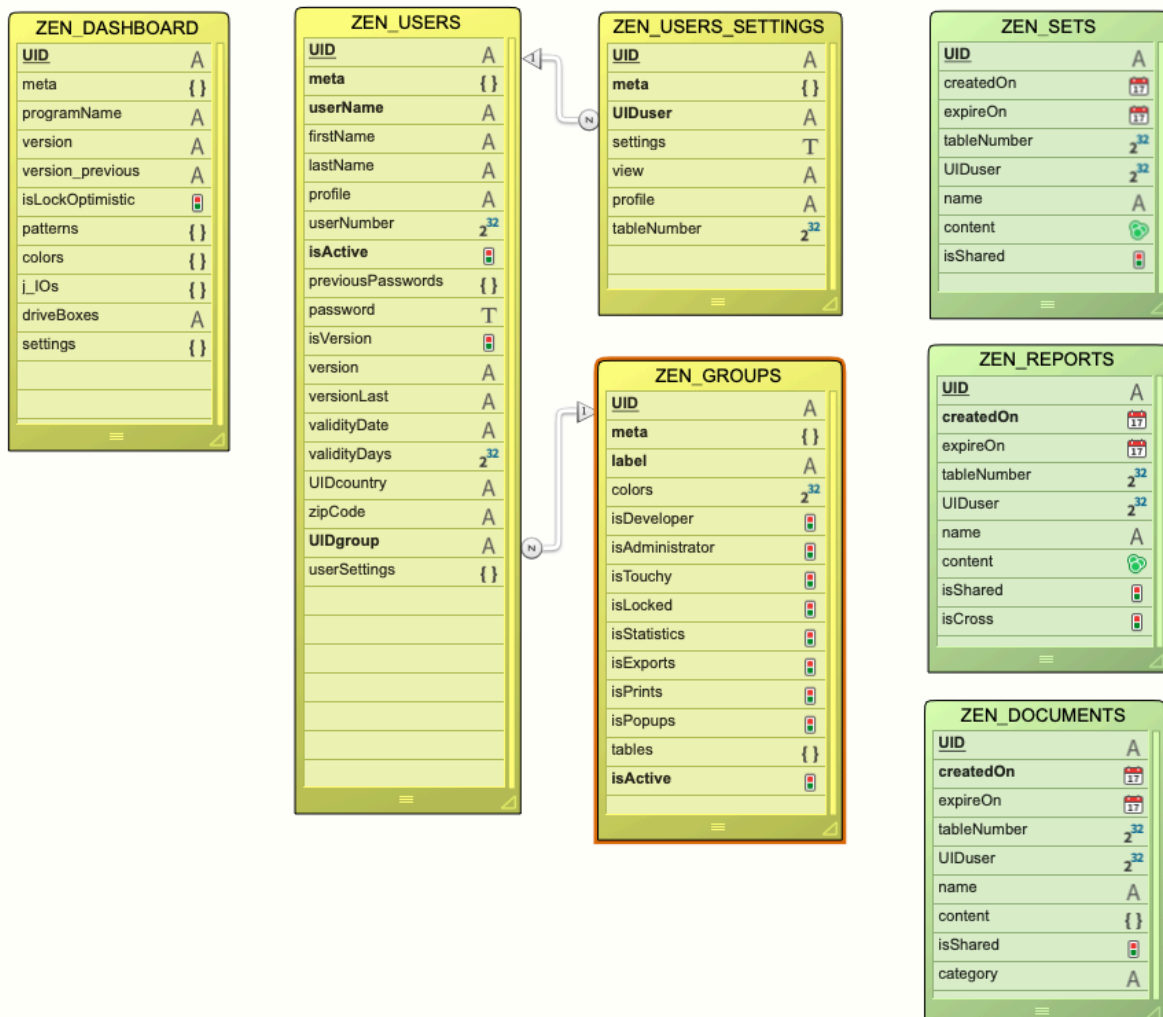
All files incorporate a « l_version » number for automatic upgrade of old version files.

Tables to host

Some tables must be created into the host database, even fully managed and embedded inside zen_Nucleus (locals). Those list's json files are into the component Resources/ rather than « zen_json ».

Some tables are mandatory, like

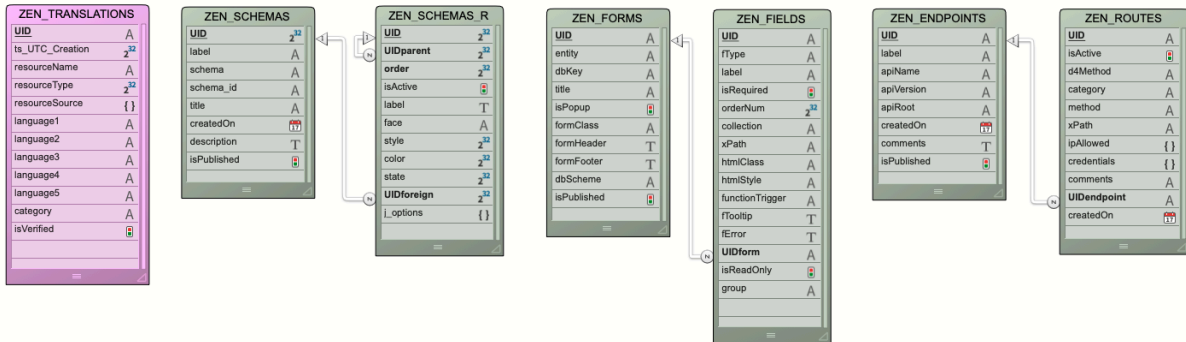
- ZEN_DASHBOARD, ZEN_USERS, ZEN_USERS_SETTINGS, ZEN_GROUPS
- ZEN_SETS ; ZEN_REPORTS ; ZEN_DOCUMENTS





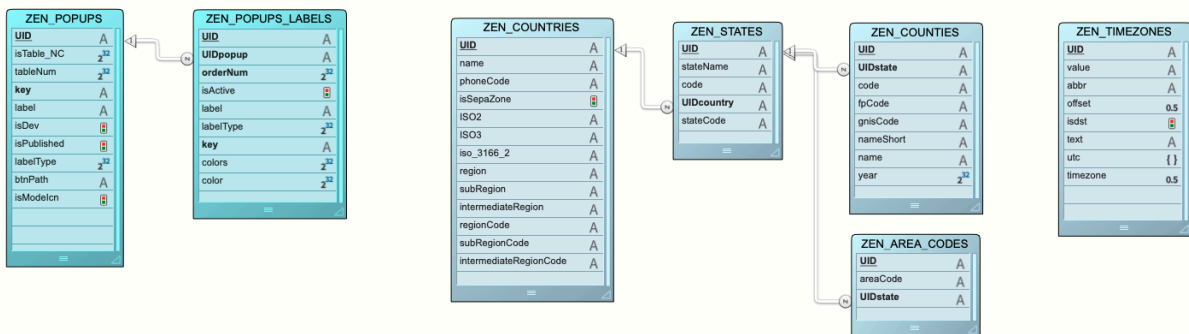
Some tables are optional, for developer like

- ZEN_TRANSLATIONS ; ZEN_SCHEMAS & ZEN_SCHEMAS_R, ZEN_FORMS & ZEN_FIELDS, ZEN_ENDPOINTS & ZEN_ROUTES



or for managing popups or countries like

- ZEN_POPUPS & ZEN_POPUPS_LABELS
- ZEN_COUNTRIES, ZEN_STATES, ZEN_COUNTIES, ZEN_AREA_CODES, ZEN_TIMEZONES





Lists file format

Form parameters

Property	Type	Description
l_version	Integer	Version number, for incremental update
l_vPrevious	Integer	Previous version number before last update
t_table	Text	Table name of this file
l_table	Integer	Table number
t_view	Text	View of the table
is_write	Boolean	Enable the write possibility
is_add	Boolean	Enable the add button
is_edit	Boolean	Enable the edit button
is_delete	Boolean	Enable the delete button
is_transaction	Boolean	Indicates if the records are opened in transaction or not.
is_excel	Boolean	Enable excel export
is_print	Boolean	Enable the print button
is_quickView	Boolean	Enable the quickView button
l_instances	Integer	Maximum instances for this table : not used
l_related	Integer	Table related for a automatic transposition
t_process_title	Text	Title of the process for record - not used
t_class	Text	Class name for the list's form if any, attached to « Form.cl »

Listbox parameters

Property	Type	Description
t_font	Text	Font name
l_fontsize	Integer	Font size
l_fontcolor	Integer	rgb
l_hgridcolor	Integer	Horizontal grid color (woc, k_transparent = none)
l_vgridcolor	Integer	Vertical grid color (woc, k_transparent = none)
l_foreground	Integer	Foreground color (woc)
l_background	Integer	Background color (woc)
l_alternate	Integer	Background alternate color (woc)
l_with	Integer	Object list's width. Help to calculate remaining
l_height	Integer	Height of the lines
aj_columns[]	Collection	All the column's data. See below



Columns parameters « aj_columns[] »

Property	Type	Description
is_active	Boolean	If False, the column is proposed in the list to be enabled easily
t_label	Text	Column title
l_color	Integer	Color for the cell
t_table	Text	Table name
l_table	Integer	table number
t_field	Text	Field name
l_field	Integer	Field number
l_fieldType	Integer	Field type
t_formula	Text	Formula, like Form.cl.lb_isVerified_img(This)
l_formulaType	Integer	and Value type
l_sort	Integer	0 none ; 1 Ask ; 2 Desc
t_format	Text	Format of the cell
l_width	Integer	Width of the column
l_style	Integer	Style: bit 0 Bold ; bit 1 Italic ; bit 2 Underline
l_align	Integer	Alignment: 0 Default ; 1 Left ; 2 Center ; 3 Right
is_searchbox	Boolean	Included in the text filter if true
l_footer	Integer	Type for auto-calculated footer. 0 None ; 1 Min ; 2 Max ; 3 Sum ; 4 Average ; 5 Count

The files are stored close to the data file in a folder « zen_json/lists_json » followed with as many folders as profiles, and in them a file « TABLE{.view}.json ».



Menus file format

Module Menu datas

Property	Type	Description
<code>l_version</code>	Integer	Version number, for incremental update
<code>l_vPrevious</code>	Integer	Previous version number before last update
<code>t_path_icn</code>	Text	In the form « » Paths to icons in resources to show on menu, separated with space
<code>aj_menus[]</code>	Collection	All the menu's data. See below

Menu parameters « aj_menus[] »

Property	Type	Description
<code>is_line</code>	Boolean	True for a new line break
<code>is_active</code>	Boolean	Not incorporated if false
<code>l_label</code>	Integer	rscType for the label translation
<code>t_label</code>	Text	Menu label, for translation. Use the component translation for ready translated item in it by a prefix of « & »
<code>t_icon</code>	Text	« icn_ » <name> « .png » If name = [<space>:<color>] => color value.
<code>t_table</code>	Text	Table name if any
<code>t_method</code>	Text	Method to call with parameters
<code>t_property</code>	Text	ie : 4D_standard_action_name
<code>t_modifier</code>	Text	Modifier for menu shortcut
<code>t_key</code>	Text	Menu shortcut's key

The files are stored close to the data file in a folder « zen_json/menus_json » followed with as many folders as profiles, and in them a file « menus.json ».

The « t_method » is executed on a menu selection, with the possibility to get some informations like :

```
var $vL_menu; $vL_menu_bar; $vL_menu_item : Integer
var $vT_view; $vT_parameter : Text
$vL_menu:=Menu selected
$vL_menu_bar:=$vL_menu\65536
$vL_menu_item:=$vL_menu%65536
GET MENU ITEM PROPERTY($vL_menu_bar; $vL_menu_item; "t_view"; $vT_view) //
"" if not exists

$vT_parameter:=Get selected menu item parameter
```

\$vT_parameter contains « t_table » and \$vT_view contains « t_property ».