

# **Telem SCADA System**

**Schema design and  
association with database**

**Software manual**



## Contents

1. Introduction	3
1.1. Telem Schema System	3
1.2. Telem Schema design environment.	4
1.3. Publishing and viewing schemas	5
2. Macromedia Flash settings	6
2.1. Menus and panels of Macromedia Flash.	6
2.2. Flash document settings.	7
2.3. Layers in schema.fla.	8
3. Telem schema library symbols.	9
3.1. Picking up library symbols	10
4. Drawing schema.	11
4.1. Viewing designed schema.	12
4.2. Grouping symbols into swf-layers	13
4.3. Creating feeder symbols	14
4.3.1. Adding symbols inside of feeder to swf-layers	15
4.4. Using symbols for loading schemas and panels	16
4.4.1. Load schema symbol	16
4.4.2. Load additional schema symbol	16
4.4.3. Symbols for loading additional panels	17
4.4.4. Modules library	18
4.4.5. Measurements collection	18
5. Configuring Schema-control panel.	19
5.1. Setting up schemas list	19
5.2. Setting up submenu items.	21
5.3. Editing file control.cfg	22
6. Test operations.	23
6.1. Test.swf description	23
6.2. Testing all symbols.	24
6.3. Generating reports .	25
7. Symbols association with database objects	26
7.1. Schema.log -method	27
7.2. Schema.AS and schema.DBC -methods	28
7.2.1. Schema.AS -method	29
7.2.2. Schema.DBC -method	31
7.3. DBCgen -method	33
8. Scripts and variables for schema's root	34
9. Using SchemaHelp.html.	36
9.1. Color settings	36
9.2. Main sheet of SchemaHelp.html	38
9.3. General - SchemaHelp.html	39
9.4. Steps of schema design - SchemaHelp.html	40
9.4. Drawing with LineColor - SchemaHelp.html	41

## 1. Introduction

**Telem SCADA schemas are developed using Macromedia Flash.**

**Flash** is an authoring tool that allows you to create anything from a simple drawing to a complex interactive web application, such as an online store. You can make your Flash applications media rich by adding pictures, sound, and video. Flash includes many features that make it powerful but easy to use, such as drag-and-drop user interface components, built-in behaviors that add ActionScript to your document, and special effects that you can add to objects.

When you author in Flash you work in a Flash document, a file that, when saved, has the file extension .fla. When you are ready to deploy your Flash content, you publish it, creating a file with the extension .swf. Flash Player runs the SWF file.

**Macromedia Flash Player**, which runs the applications (SWF files) that you create, installs by default along with Flash. Flash Player ensures that all SWF content is viewable and available consistently and across the broadest range of platforms, browsers, and devices.

The Macromedia Flash Player is distributed with products from every major partner, including Microsoft, Apple, Netscape, AOL, and Opera. Flash Player is distributed freely to anyone who wants to use it. You can get the latest version of Flash Player at the Macromedia Flash Player Download Center.

### 1.1. Telem Schema System

**Telem Schema System** is the collection of **SWF files** (and text-files for configuration) placed into predetermined tree structure. Telem SCADA user's main program uses this schema system.

Telem Schema System enables to

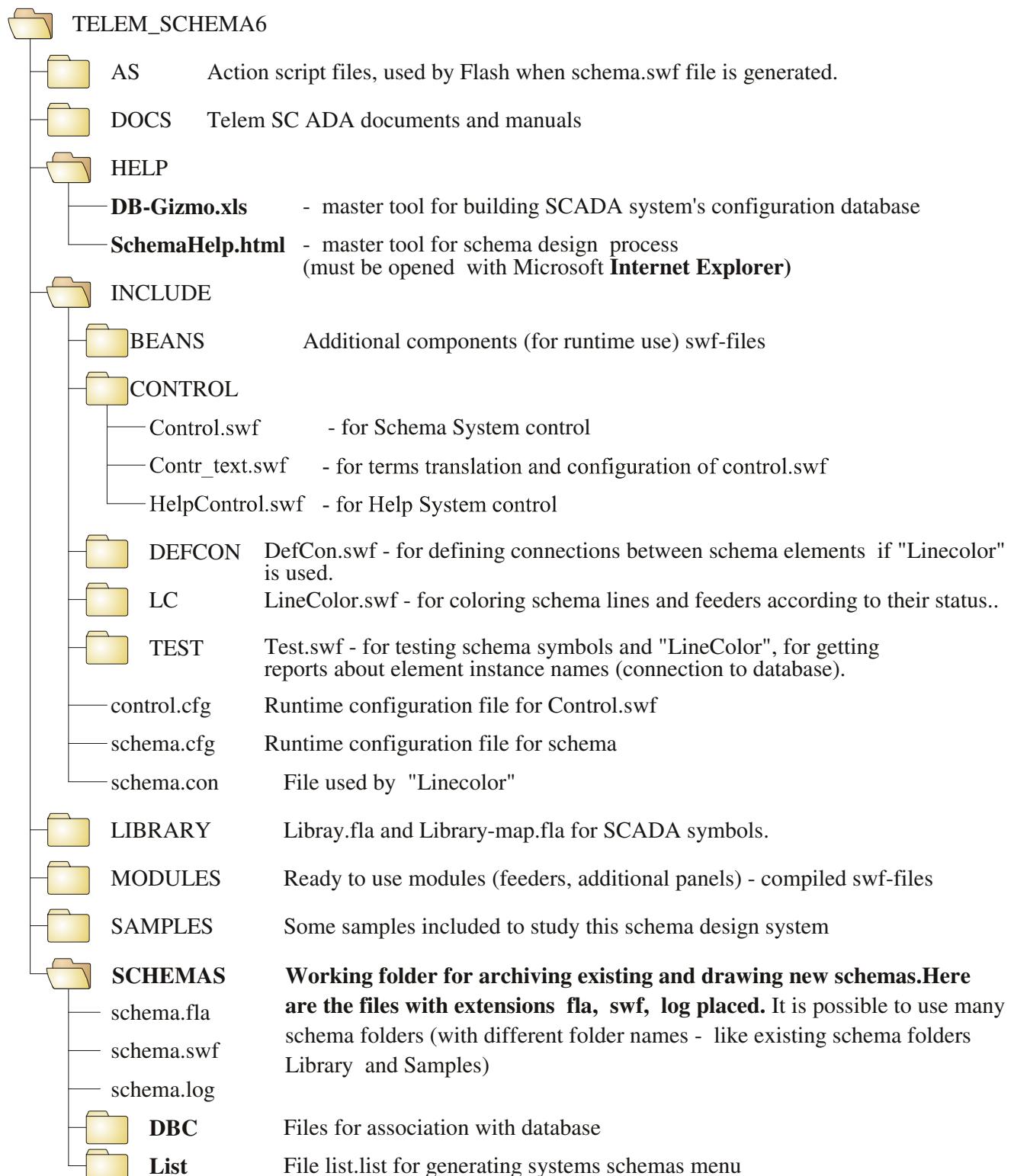
- load different schemas, schema components and documents
- configure loaded schema (zoom in, drag, zoom out, show and hide schema layers etc.)
- go back and forward thru displayed schemas
- save configuration or schema symbols status information into file
- test schema symbols and other components
- create and test line coloring feature for schemas
- communicate with host program (if Flash Player is placed into host program)

Telem SCADA system users interface (SCADA system client) uses Telem Schema System for displaying SCADA schemas. SCADA systems configuration locates where Telem Schema System files tree is placed and which file must be opened if user enter the system. Usually Telem Schema System files are placed on SCADA system server PC or on clients workstation.

**Telem Schema System files** must be created in **Telem Schema design environment**, where all schemas main features could be tested independently from Telem SCADA systems other applications.

## 1.2. Telem Schema design environment.

**Telem schema design environment** is the collection of different kind of files and scripts placed into predetermined tree structure. This structure includes additionally to Telem Schema system (SWF files) all FLA, AS, LOG, DBC, CFG, CON, TXT, LIST, HTML, XLS etc. files. All these files are necessary for designing and testing schemas and it's components. Telem schema design environment is usually placed into folder TELEM\_SCHEMA (or TELEM\_SCHEMA6 - version 6 - this name could be chosen freely).



**All schemas should be designed in this environment, which enables:**

- Draw schema, using Telem Library symbols.
- Give names to analog and digital symbols for association with database.
- Generate all schema's files , including line coloring feature.
- Test schema.
- Copy schema's files (Telem Schema system) into SCADA system server.

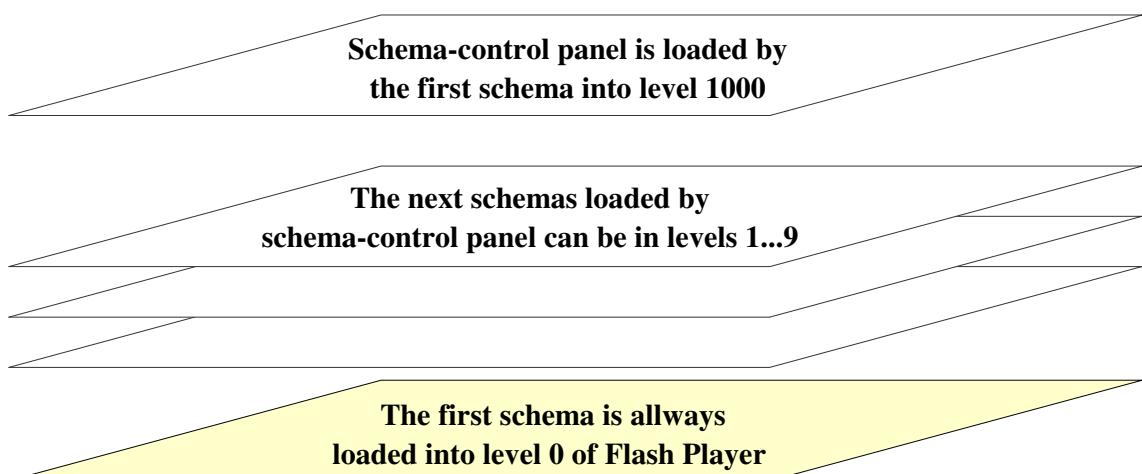
### 1.3. Publishing and viewing schemas

When you create and save Macromedia Flash documents within the Flash authoring environment, the documents are in FLA file format. To display a document in Macromedia Flash Player, you must publish or export the document as a SWF file.

Flash Player has a stacking order of levels starting with level 0. These levels are like layers of acetate; they are transparent except for the objects on each level. The initial SWF file loaded into an instance of the Flash Player is automatically loaded into level 0. The SWF file in level 0 sets the frame rate, background color, and frame size for all subsequently loaded SWF files. SWF files are then stacked in higher-numbered levels above the SWF file in level 0. If you load a new SWF file into level 0, every level in Flash Player is unloaded, and level 0 is replaced with the new file.

When the SWF file is created from FLA file, the SWF file is loaded always into level 0 of Flash Player. During this compilation process Macromedia Flash includes actionscript files from folder AS. The script included to created file schema. SWF opens configuration file schema.cfg from folder Include and loads upon the schema (to upper level) **schema-controlpanel** (file control.swf from folder Include/Control). Schema.swf and control.swf also include scripts for communication with host program (if the Flash Player is placed into Telem-SCADA client program). Next schemas, loaded using control.swf or using "LoadSchema" symbol on any schema, can be on levels 1,2,3 and so on. The visible schema is determined by state of control.swf.

Following diagram illustrates Telem Schema system in Flash Player:



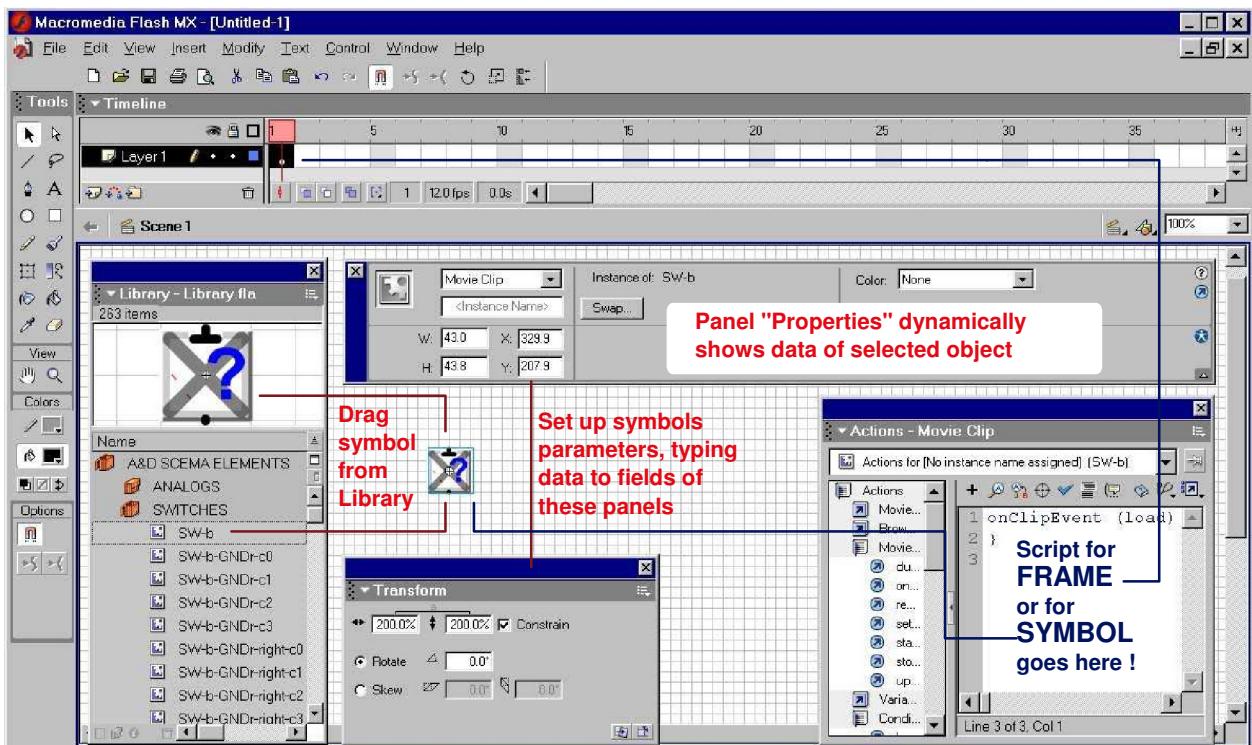
Don't confuse Flash Player levels and Flash FLA file layers !

## 2. Macromedia Flash settings

### 2.1. Menus and panels of Macromedia Flash.

"Macromedia Flash MX is the fastest way to create rich Internet content and applications. Powerful video, multimedia, and application development features allow the creation of rich user interfaces and enterprise application front-ends."

Macromedia Flash environment makes the design convenient and quick. If Telem library is used for picking the schema elements then user must know only small part from all Macromedia Flash features. Only few menus and panels are used.



It's recommended to use keyboard shortcuts for frequently used operations:

- |                                     |                   |
|-------------------------------------|-------------------|
| - Window / Properties               | Ctrl+F3           |
| - Window / Transform                | Ctrl+T            |
| - Window / Actions                  | F9                |
| - Window / Library                  | Ctrl+L (or F11)   |
| - Modify / Arrange / Bring to Front | Ctrl+Shift+Up     |
| - Modify / Arrange / Send to Back   | Ctrl+Shift+Down   |
| - Modify / Group                    | Ctrl+G            |
| - Modify / Break Apart              | Ctrl+B            |
| - Insert / Convert to Symbol        | F8                |
| <b>- Control / Test Movie</b>       | <b>Ctrl+Enter</b> |

Pay attention to menus:

- View / Grid / Show Grid and View / Grid/snap to Grid
- Window / Save Panel Layout - save your Flash panels layout, comfortable for schema design
- Window / Panel Sets / Telem - open your Flash panels layout
- Edit / Preferences - remove feature "contact-sensitive Selection and Lasso tools"

## 2.2. Flash document settings.

For drawing a new schema it's recommended to take an existing schema with appropriate library, Blank.fla or Library.fla and save it as New schema.fla. In this case all schema's common settings will have the normal values and you can start with drawing.

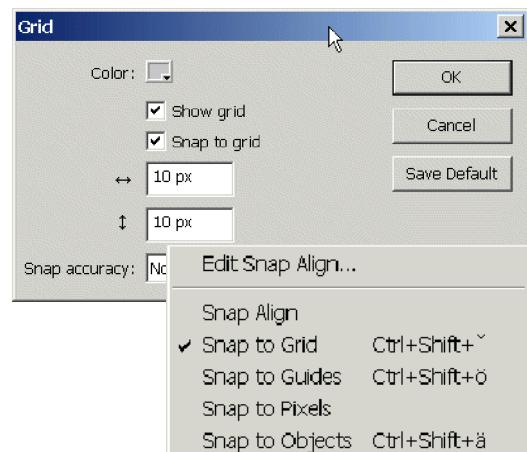
The most important setting usually are:

- Size: 1270 x 890 pixels. All schemas of system must have the same size.
- Background: white.
- Frame rate: 12 fps
- Publish settings: Player 7, ActionScript 1
- Grid spacing: 5 or 10px



From menu "View/Grid/Edit Grid" set grid spacing to 5 or 10px

Using menu "View/Snapping" enable only "Snap to Grid"



## 2.3. Layers in schema.fla.

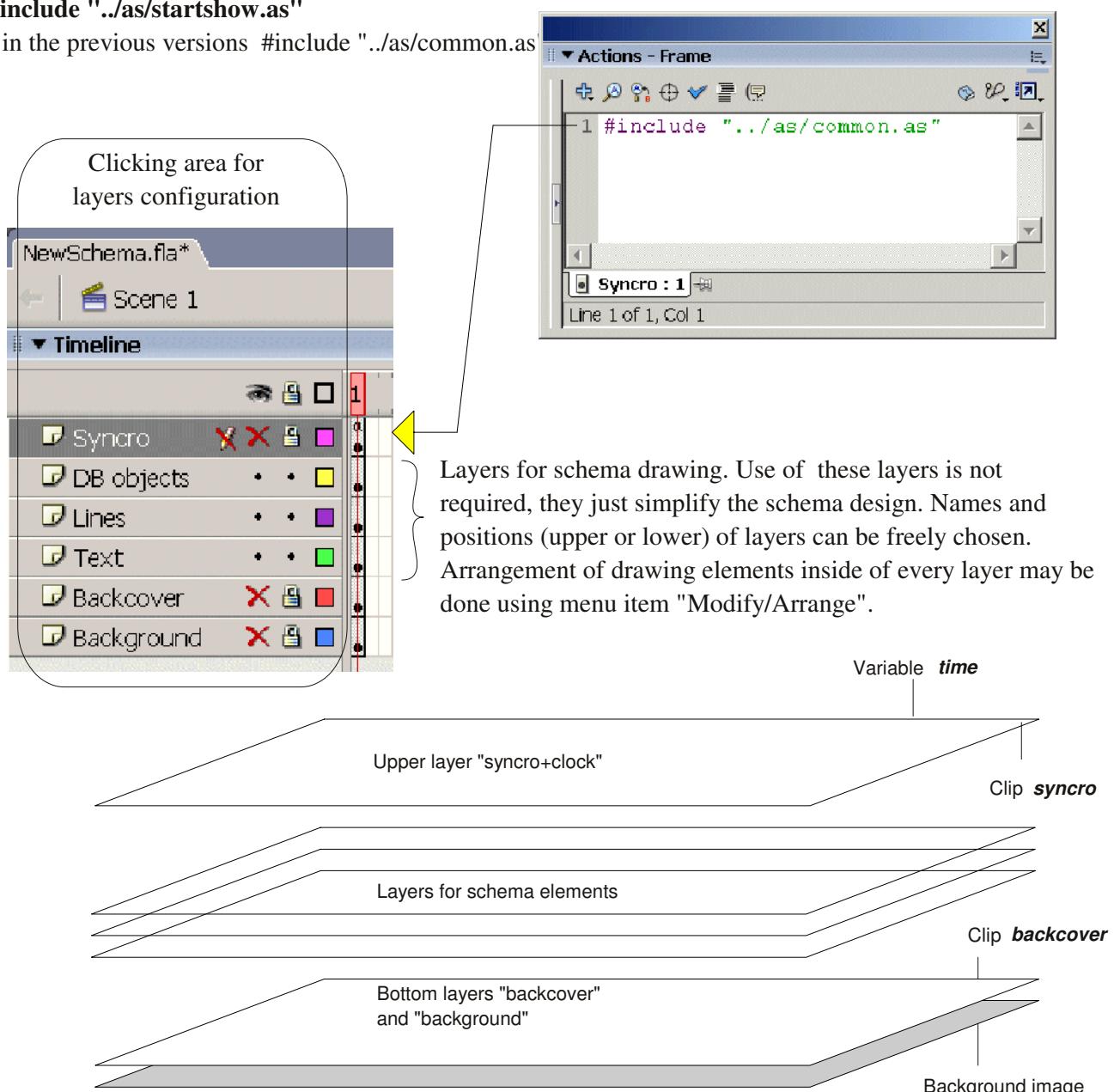
Every flash schema.fla includes layers "syncro", "backcover" and "background". These layers are usually invisible and locked.

- Layer "syncro" has clip (instance name: syncro ) for controlling the loading process of this schema, and dynamic text field for time display (variable: time).
- Layer "backcover" has clip (instance name: backcover) for changing the background brightness of this schema.
- Layer "background" is for adding the background image to the schema. Visibility of this image depends on backcover's transparency property ("alpha"- property). Usually the background image is a dark gray (RGB = 333333) rectangle. **If different background color is used, some schema symbols with "internal cover" maybe displayed not correctly.**

Every flash schema.fla must have action script for frame, placed usually on the upper layer. For example there may be only one line:

```
#include "../as/startshow.as"
```

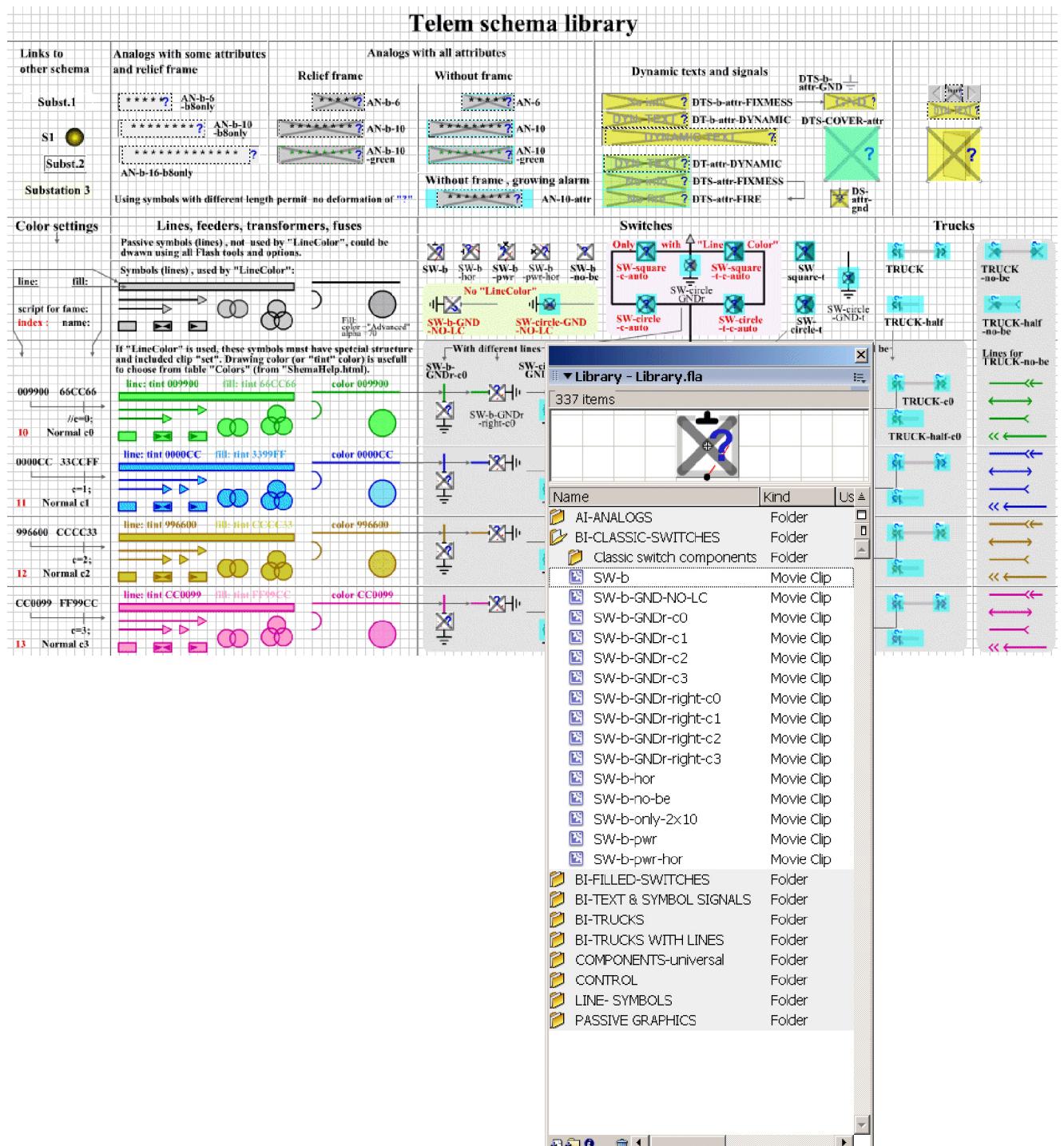
( in the previous versions #include "../as/common.as"



### 3. Telem schema library symbols.

Every flash schema includes the following basic symbols which should be taken from library:

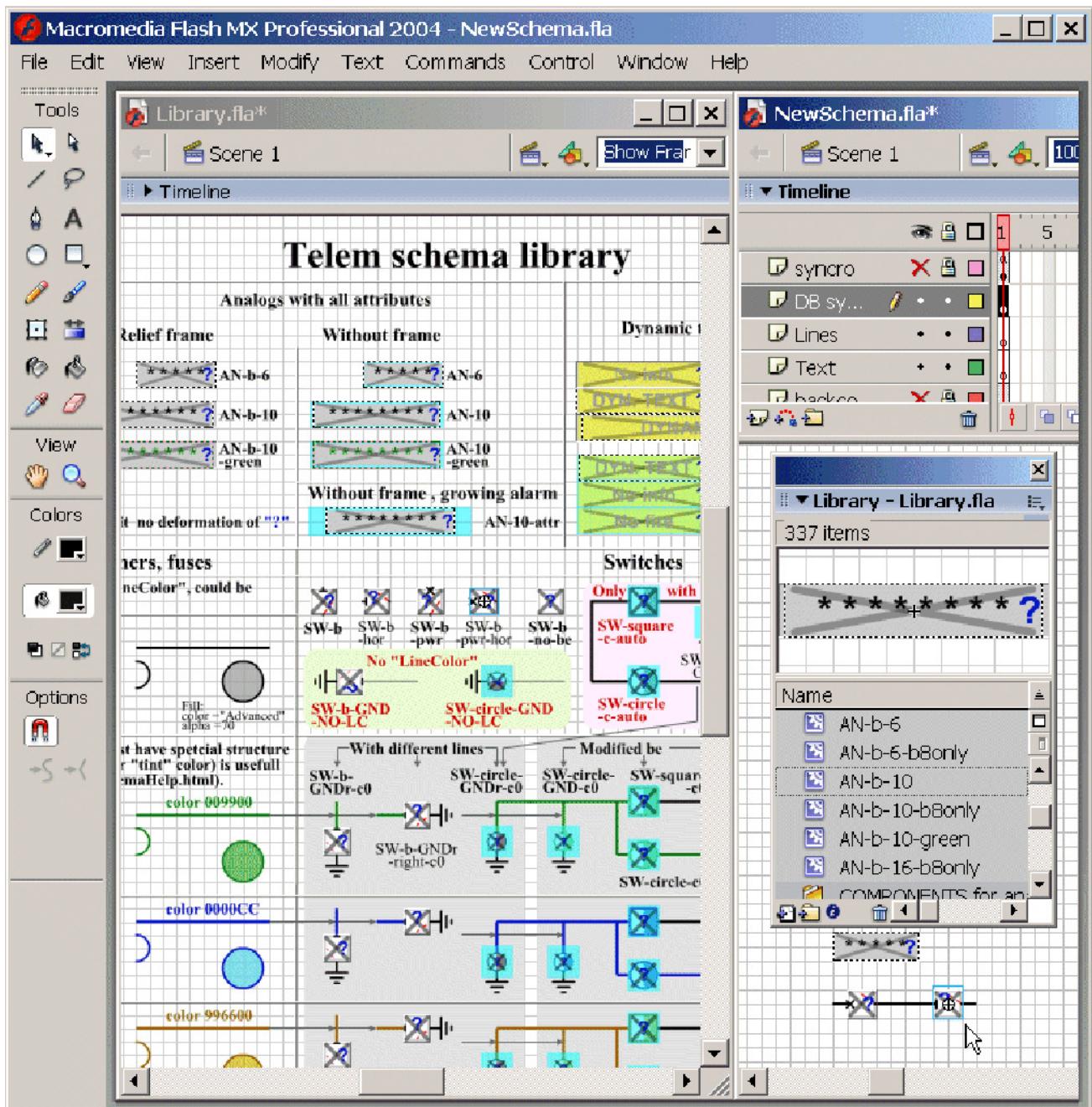
1. Symbols for displaying status of binary input objects .
  2. Symbols for displaying status and value of analog input objects .
  3. Symbols for links to other schemas (for opening other schemas).
  4. Symbols for opening additional schemas.
  5. Symbols for opening additional data panels.
  6. Symbols for displaying lines, feeders, transformers, fuses and so on, if line coloring feature is used.
- Other graphical elements are static and can be drawn without any limits.



### 3.1. Picking up library symbols

The schema symbols can be chosen directly from library.fla or picked up from external Library panel. Opening external Library panel from menu "File / Import / Open External Library" and choose library.fla. For picking the schema elements following ways are recommended:

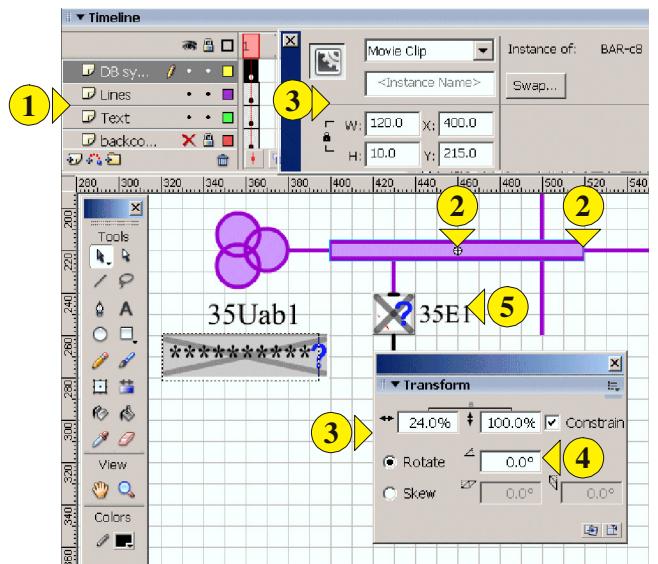
- directly from library.fla
  - drag a symbol from Library.fla to new Schema.fla.
  - copy a symbol from Library.fla and paste to new Schema.fla.
- from external Library panel
  - drag a symbol from External Library panel to new Schema.fla.
  - drag a symbol from External Library panel to new Schema Library panel. In this case the symbol appears to new Schema Library only, to new Schema.fla it must be placed additionally.



## 4. Drawing schema.

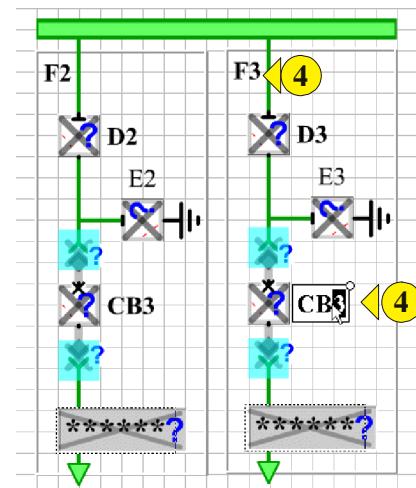
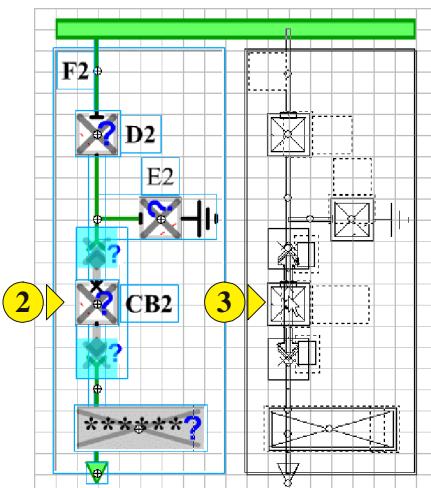
For drawing the schema.fla is recommended the best practice, which guarantees the high accuracy of schema:

1. Use 2...3 layers for schema symbols.  
Arrangement of drawing elements inside of every layer may be changed using menu item "Modify/Arrange" (Ctrl+Shift+Up; Ctrl+Shift+Down).
2. Shift symbol to its place from **center of symmetry**, using "Snap to Grid". If necessary, zoom in this schemas area.
3. Resize symbols using panel "Properties" or "Transform".
4. Rotate symbol using panel "Transform" (type the angle value to corresponding field).
5. Add element name to schema (a text near the symbol, recommended placement -to some lower layer)



### Drawing the repeated parts of schema (multiplay feeders)

1. Configure the feeder adding onto schema:  
- binary and analog symbols  
- lines  
- element names (a text near the symbol)
2. Select all elements of feeder (drag rectangle over these elements with arrow tool, if some elements selection must be changed, press "Shift" and click on required elements).
3. Press Ctrl-key and **drag a copy of feeder** to its new place. For dragging take selected symbols from any center point
4. Edit duplicated feeders and its element names (a text near the symbol).

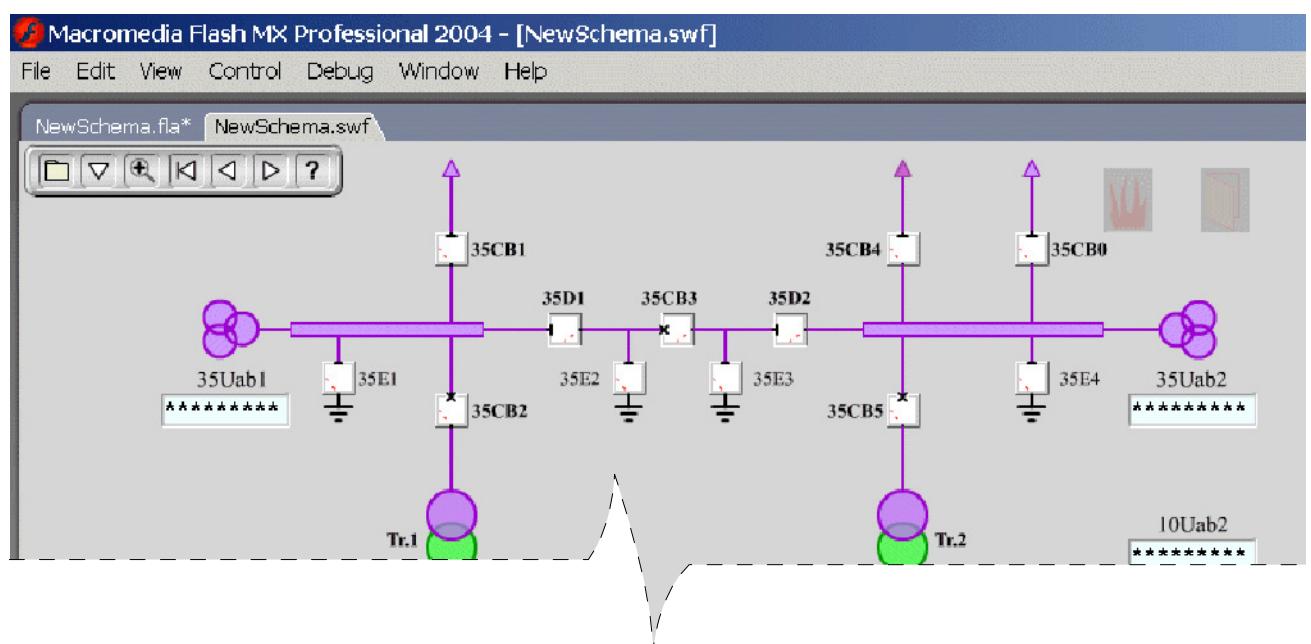
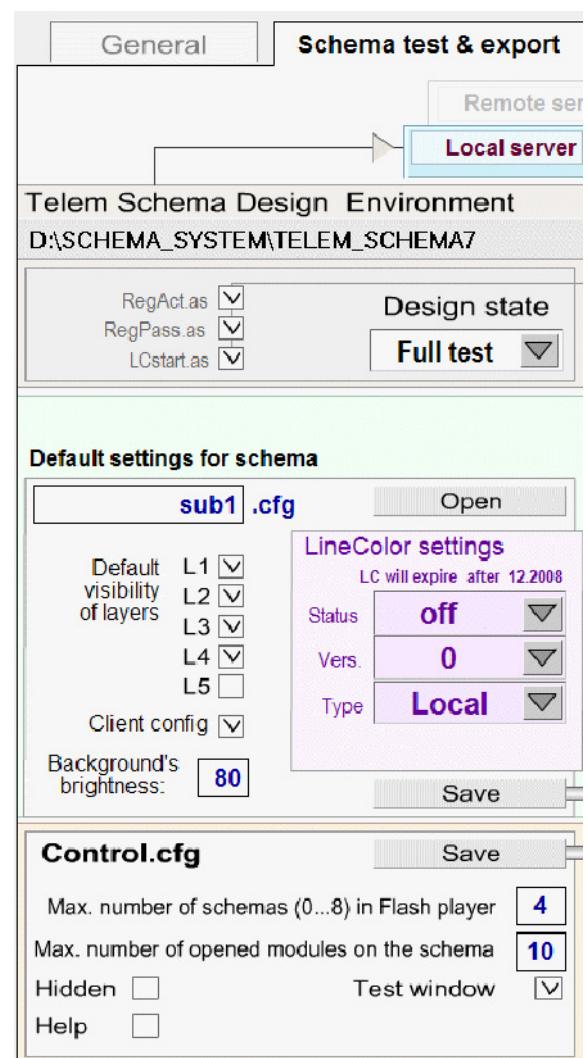


## 4.1. Viewing designed schema.

To display a document in Macromedia Flash Player, the document must be publish or exported as a SWF file. During this process the action script files from folder AS are included and generated SWF file opens configuration file *schema.cfg*. To prepare these files use **SchemaHelp.html** from folder Help. Before openig **SchemaHelp.html** publish schema.swf from Flash menu "File/Publish" (or use keyboard shortcut Shift+F12). In this case SchemaHelp.html script find published SWF file as last created.

1. Set Design state to **Full test**.
2. Set Linecolor status to **off**.
3. Save schema.cfg

Return to Macromedia Flash and for testing the drawing results choose from menu **Control / Test Movie** (or press **Ctrl+Enter**). *Schema.swf* is generated from *schema.fla* and displayed in imported flash player. It is recomended to use this "Test Movie" operation during schema design if some new features are added to schema.



## 4.2. Grouping symbols into swf-layers

Symbols can be grouped into swf-layers for making possible show/hide operations for this group if schema is running in flash player:

1. Select elements for grouping (drag rectangle over elements with arrow tool or if they are placed on the same layer, click on these layer's frame icon).
2. Choose menu "Insert / Convert to Symbol" (or press key "F8") and give the name for this new symbol in library. Behavior must be "Movie clip".
3. The new symbol appears into library of this schema.fla.
4. Give "Instance Name" to this symbol. "Instance Name" for layer is fixed and must be L1, L2, L3 up to L9.

Repeat 1 ... 4 for other swf-layers on this schema. Result of this to schema-control panels submenu appears swf-layers items with deafault names described in chapter "Configuring Control.swf".

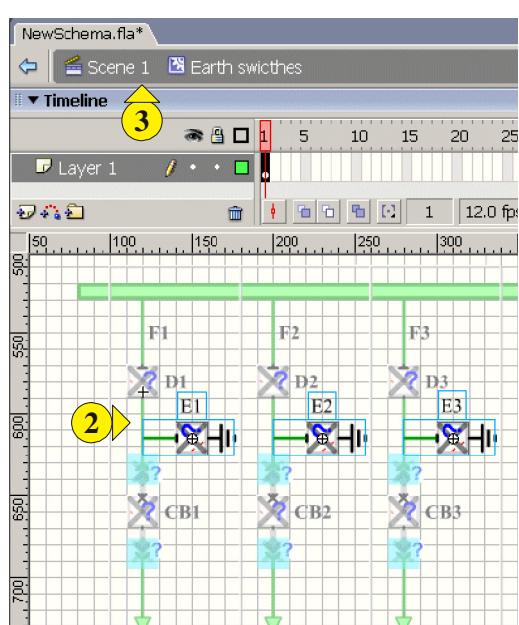
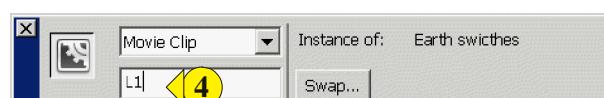
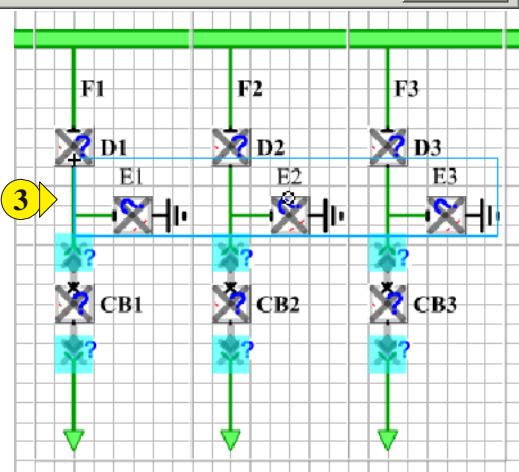
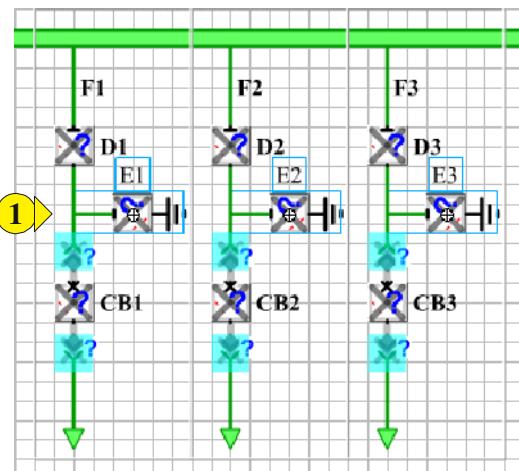
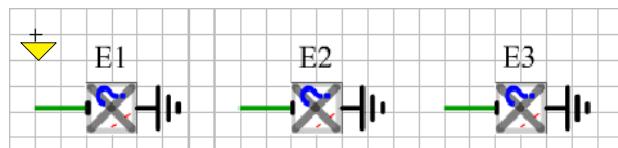
If swf-layer must have unique name (default name is not acceptable) , the required name could be attached using one of following variants:

- put on this symbol script
- ```
onClipEvent (load) {
    name = "required name";
}
```
- or with doubleclick on symbol go inside of it and put on frame script
- ```
name = "required name";
```

### Swf-layers or other Library symbols could be edited in place:

1. With doubleclick on symbol go inside of it. Flash displays rest of schema with lower contrast and indicates currently editable symbols name.
2. Edit swf-layer if needed (shift or rotate symbols, edit names etc.)
3. Exit to schemas root clicking on "Scene 1"

Clicking on the swf-layer symbol with mouse right button opens menu where "Edit" could be chosen. In this case displays for editing only symbols of swf-layer:

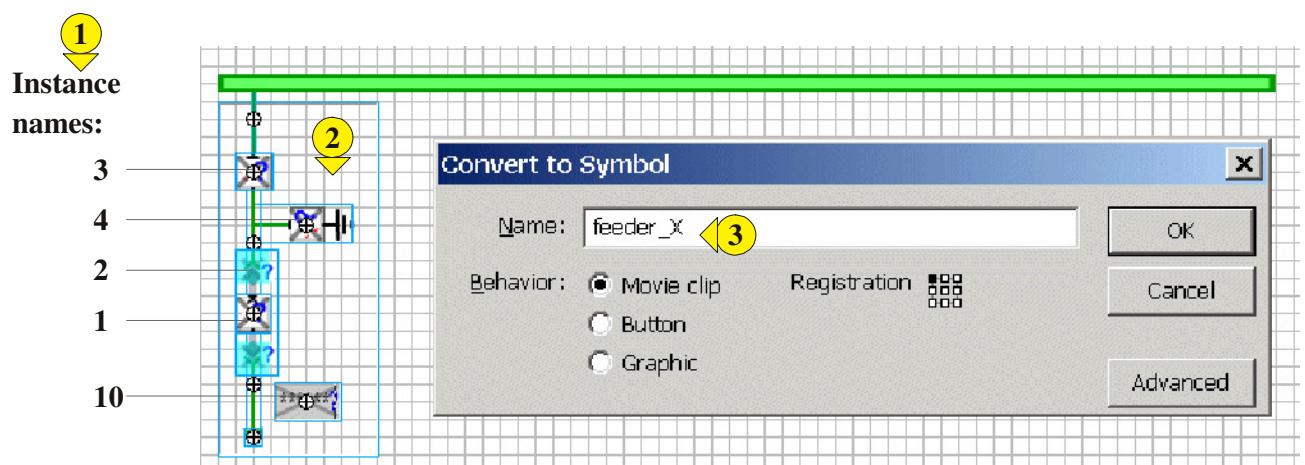


### 4.3. Creating feeder symbols

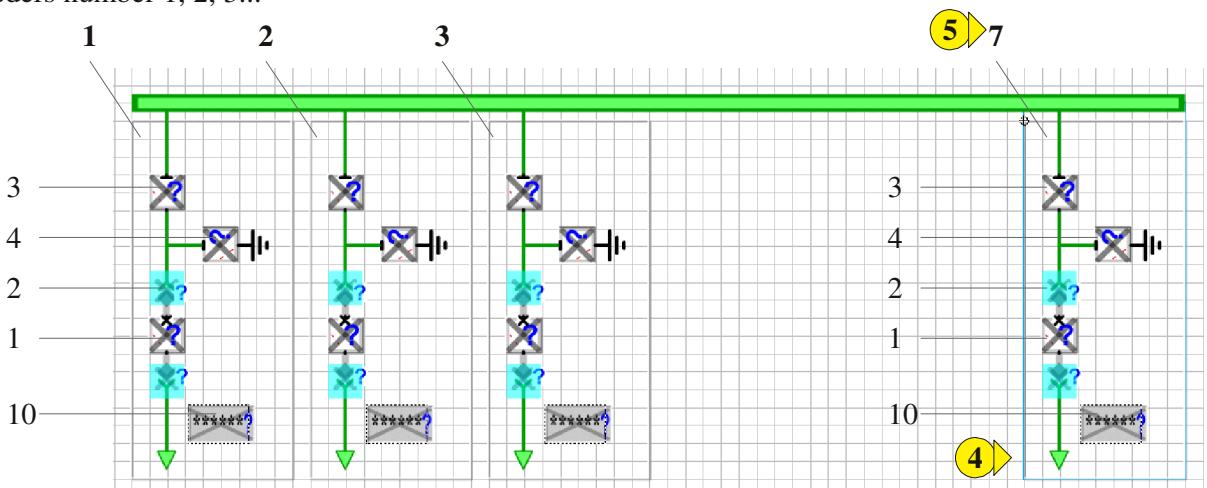
Most of SCADA symbols behavior is "Movie clip". The parent-child relationships of movie clips are hierarchical. To understand this hierarchy, consider the hierarchy on a computer: the hard disk has a root directory (or folder) and subdirectories. The root directory is analogous to the main Timeline of a Flash document: it is the parent of everything else. The subdirectories are analogous to movie clips.

Symbols groups can be converted to new symbol (feeder) to simplify processes of drawing and association with database.

1. Draw feeder and give instance names to symbols. Instance names of symbols could be A1, A2,... for analogue measurements and B1, B2,... for binary signals or numbers 1, 2, 3,... independently of symbols type.
2. Select all feeders elements (analogue measurements, switches, lines, arrows...) for converting.
3. Choose menu "Insert / Convert to Symbol" (or press key "F8") and give the name for this new symbol in library of this schema. Behavior must be "Movie clip".



4. Make required amount of feeder copies.
5. Give "Instance Names" to these copies. "Instance Name" of feeders could be F1, F2, F3... or only the feeders number 1, 2, 3...



### 4.3.1. Adding symbols inside of feeder to swf-layers

Symbols included into feeders could be grouped to swf-layers for making possible show/hide operations of this group in flash player:

1. Select element inside of feeder for grouping (click on it)
2. Type actionscript for this feeders symbol (example for adding to swf-layer no.1, max layers no is 9)

```
#include "..as/addToL1.as"
```

3. The actionscript on the frame of schema's root must consist the line:

```
#include "..as/Layers.as"
```

4. If swf-layer must have unique name (default name is not acceptable) , the required name can be attached using one of following variants:

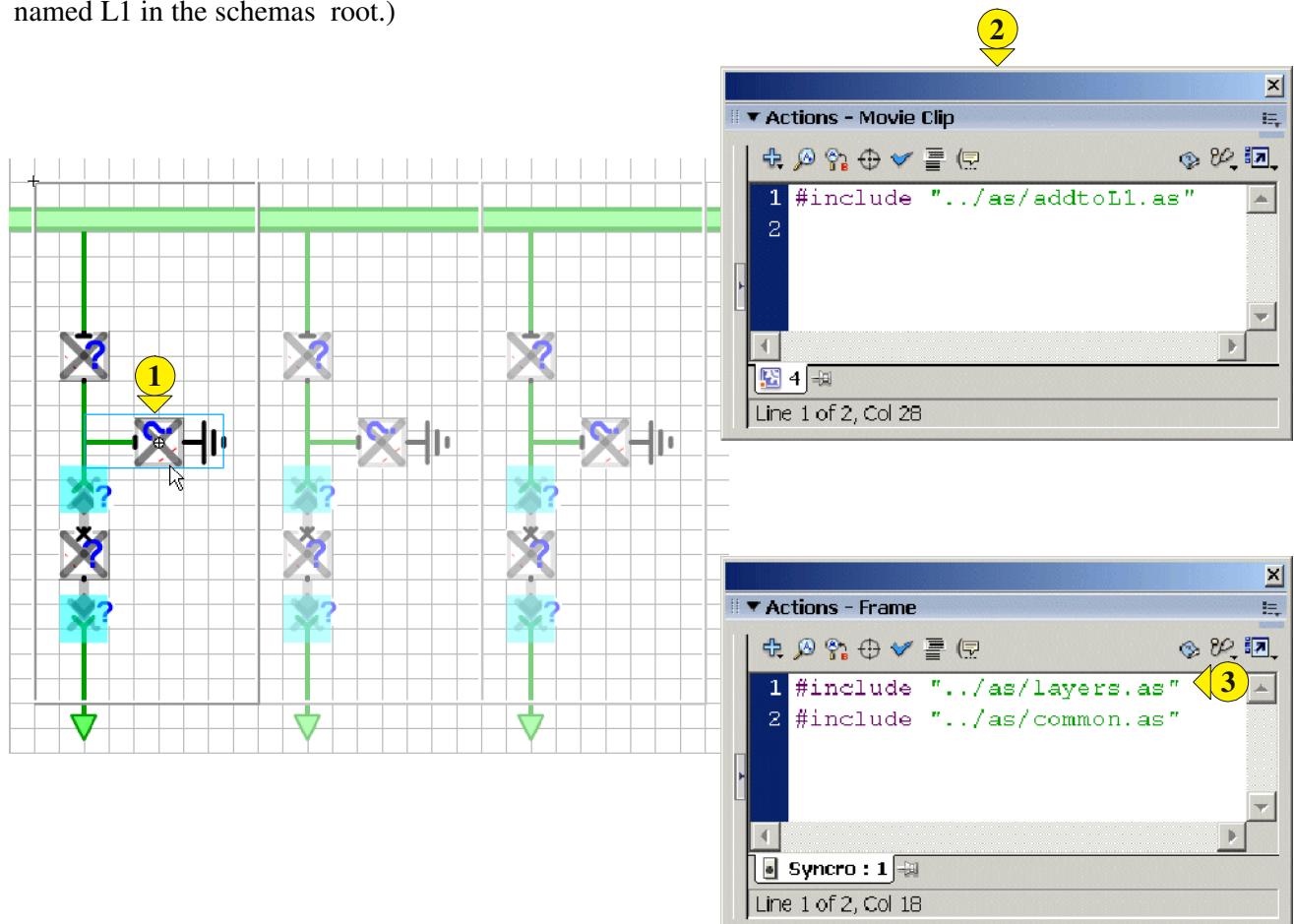
- put on schemas root symbol named L1 symbol's script

```
onClipEvent (load) {
    name = "required name";
}
```

- or with doubleclick on symbol L1, go inside of it and put on frame script

```
name = "required name";
```

(Using older version of TSS the additional condition must be enforced - there must be a swf-layer symbol named L1 in the schemas root.)



## 4.4. Using symbols for loading schemas and panels

There are 3 kinds of symbols for controlling schemas to display:

1. Symbol for opening other schema ("load schema symbol")
2. Symbol for opening additional schema ("load additional schema")
3. Symbol for open and close additional panel

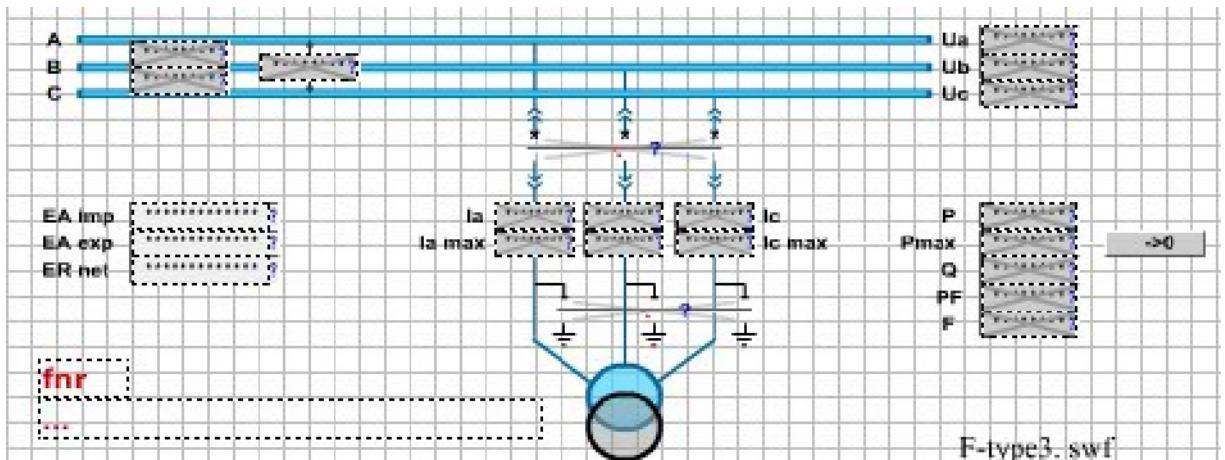
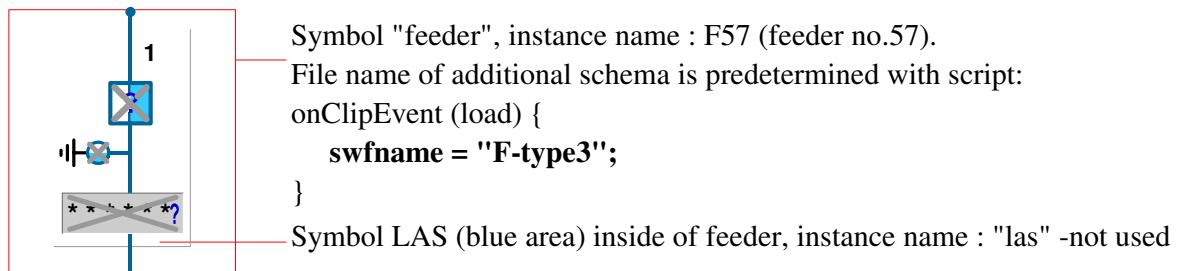
### 4.4.1. Load schema symbol

Telem Schema Library symbols ("Load Schema symbol") for loading new schema into Flash player could be designed as button, frame or any other graphical symbol . The schema-file to display is determined with instance name of "Load Schema symbol". This symbol loads into Flash player new schema or makes visible an existing schema. Clicking on the "Load Schema symbol" is identical with clicking on the line of Schema-control panel schemas list menu. The names of schemas (not file names) displayed on the schema are separately typed Flash texts (S1, Subst.2, Substation 3)



### 4.4.2. Load additional schema symbol

Telem Schema Library symbols ("Load Additional Schema symbol - LAS symbol") for loading additional schema into Flash player could be designed as button, frame or any other graphical symbol, placed inside of feeder . The schema-file to display is determined with additional script for this feeder. The first click on this symbol loads into Flash player new additional schema, next click unloads additional schema. If additional schema is loaded, the additional file for association with database is used. These additional files for association with database are different for every feeder, the additional schema could be the same. Additional schema is the extension of "ordinary" schema's feeder. Schema-control panel schemas list menu do not consists link to additional schemas.



#### 4.4.3. Symbols for loading additional panels

Telem Schema Library consists symbol ("Addmodule symbol") for loading additional panel into currently displayed schema. "Add module symbol" could be designed as button, frame or any other graphical symbol. The swf-file of panel (module) is determined with instance name of "Addmodule symbol". The first click on this symbol loads additional panel into currently displayed schema, next click unloads it. Additional panel is the extension of schema's feeder. Schema-control panel submenu consists items open and close all additional panels together. "Addmodule symbol" is usually placed inside of feeder or inside of feeder's symbol.

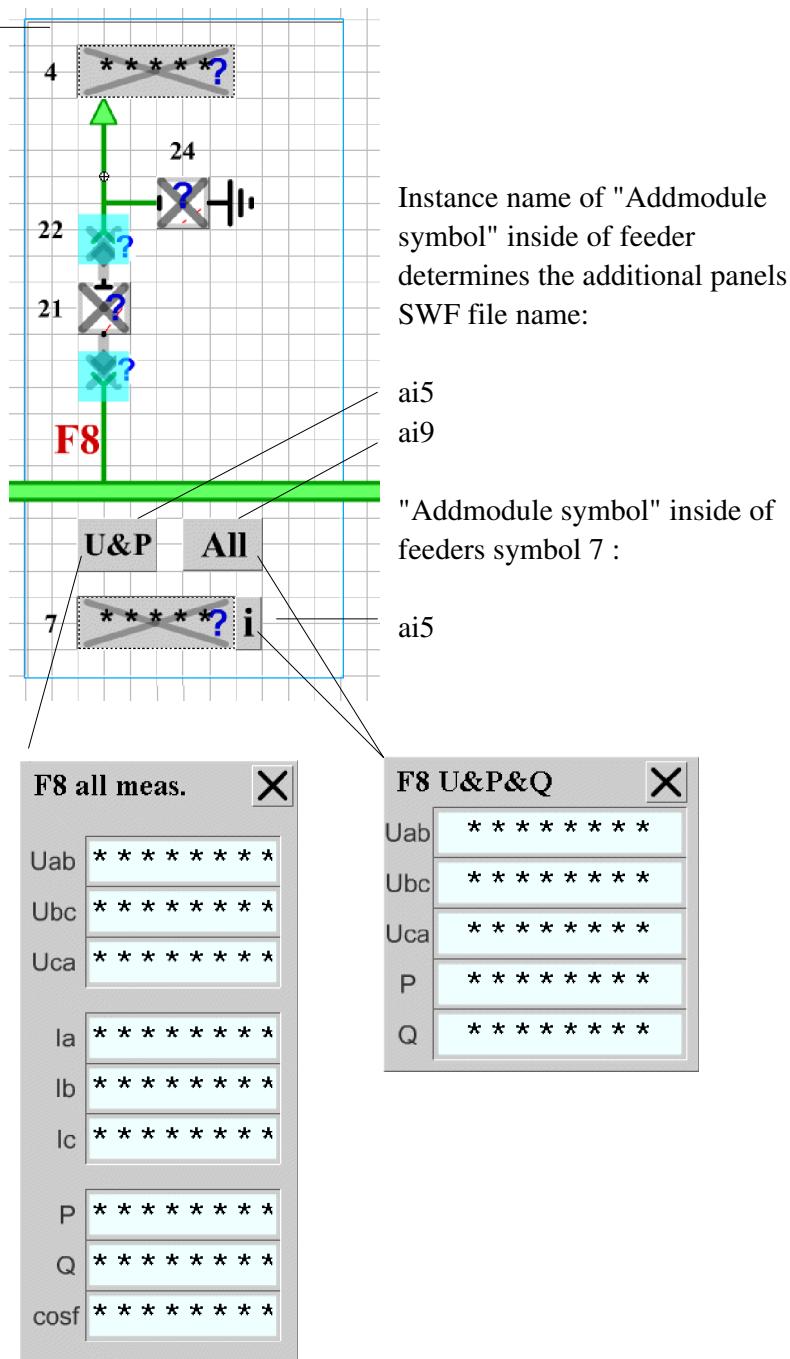
Feeder symbols with buttons for loading additional panels (FLA file example):

Feeder's instance name: **F8**

Feeder's scripts for additional panels :

```
onClipEvent (load) {
    ai5.caption = "F8 U&P&Q";
    ai9.caption = "F8 all meas.";
    set("7.ai5.caption", "F8 U&P&Q");
}
```

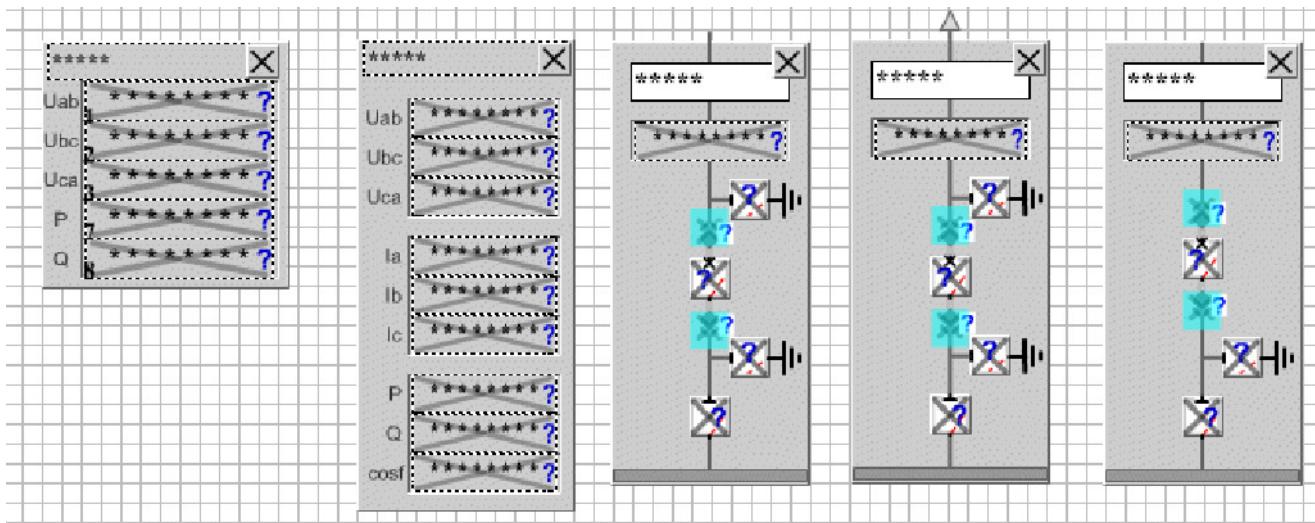
If schema runs in Flash Player, click on button opens panel corresponding to "Addmodule symbol" instance name. Panel ai5.swf could be opened using 2 different ways.



#### 4.4.4. Modules library

Telem Schema Design environment consists folder "Modules" for creating and using SWF files of additional panels. Use file modules.fla to edit existing and creating new panels (SWF files to Telem Schema system).

modules.fla:



Inside of every module on the frame must be the script:

```
_name = "desired modules name";
#include "../as/addModule.as"
```

All symbols inside of module must have script:

```
#include "../as/rename&sendData-for-Module.as" or
#include "../as/sendData-for-Module.as"
```

**For creating module.swf:**

1. Open library (press "Ctrl+L")
2. Find from library desired module
3. Click with right mouse button on the module name and choose "Export Flash Movie".
4. Give the module's name in the library to the exported flash movie and save it into the folder "Moduls"

#### 4.4.5. Measurements collection

In the folder "Modules" of Telem Schema Design environment placed file "duplicates.swf", used for displaying additional panel named Measurements Collection". **This file can't be configured by system's user.**

## 5. Configuring Schema-control panel.

Schema-control panel Control.swf uses following files:

- control\_text.swf for setting up schema-control panel submenu item names and Telem Schema Systems color table, also the time format of the schemas clock (inside of the clip named clock)
- list.list for setting up schemas list of schema-control panel
- control.cfg for additional Telem Schema Systems runtime configuration

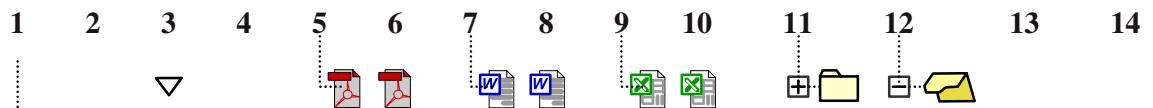
### 5.1. Setting up schemas list

Control.swf uses for generating schemas list file "list.list" placed in folder "list". Every line of this text-file is displayed as line of schemas list with desired icon and color. Line format:

**name | filename | icon no | color no | additional information**

**name** is displayed on the line of schemas list

**filename** - file to open if on this menu item is clicked. Schemas (.swf-files) must be without extension  
**icon no** - number of icon displayed on the line:



**Control.swf generates icon using filename extension, usually icons number is empty**

**color no** - index of text color described in Control\_text.fla array **itc[ ]**. If color no is empty the 0 is used. For currently loaded schema is used colors with higher brightness.

#### Schema is not loaded:

```
itc[0] = 0x0;
itc[1] = 0x990000;
itc[2] = 0x996600;
itc[3] = 0x888800;
itc[4] = 0x339900;
itc[5] = 0x9966;
itc[6] = 0x33FF;
itc[7] = 0x3333CC;
itc[8] = 0x660099;
itc[9] = 0x990066;
```

<b>color 0</b>
<b>color 1</b>
<b>color 2</b>
<b>color 3</b>
<b>color 4</b>
<b>color 5</b>
<b>color 6</b>
<b>color 7</b>
<b>color 8</b>
<b>color 9</b>

#### Schema is loaded:

```
itc[10] = 0x888888;
itc[11] = 0xFF6699;
itc[12] = 0xCC9966;
itc[13] = 0xCCCC00;
itc[14] = 0x66FF00;
itc[15] = 0xFF99;
itc[16] = 0x99CCFF;
itc[17] = 0x9999FF;
itc[18] = 0xCC66FF;
itc[19] = 0xCC66CC;
```

```
itc[20] = 0xdddd; // text color,if filename is empty
itc[22] = 0xcc0000; // Layer name color,if layer has special name
```

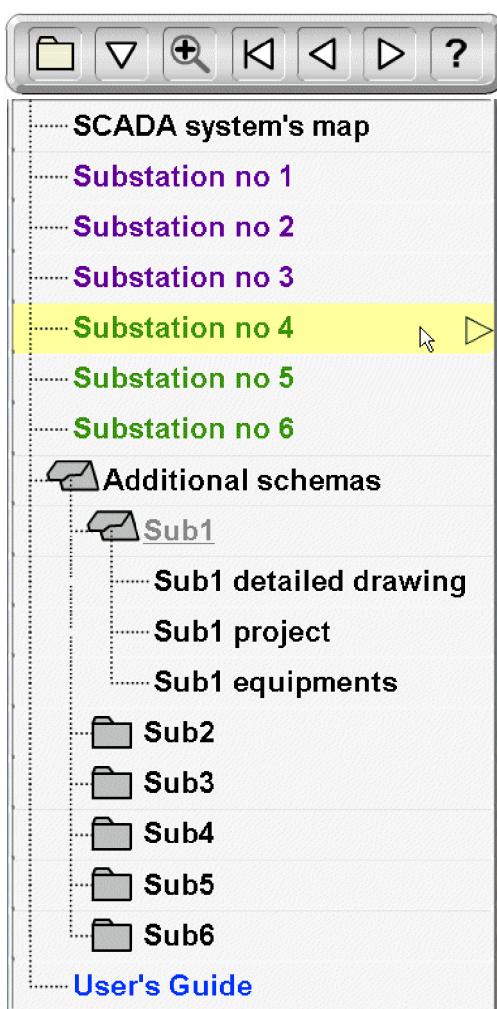
#### additional information

- true -if this item is folder, it is displayed in opened status on load

After editing control\_text.fla (action script for frame1) the control\_text.swf must be published

The file list.list is used for generating schemas list if control.swf is loaded. Every line of file "list.list" is displayed as file or folder of schemas list. These files or folder could be placed into maximum 3 levels of "tree". Clicking on the items left side opens and closes folders, if folders icon exists. Clicking on right side of item loads corresponding file if items filename exists. Also items with folder icon may have filename to open, but it could be only swf-file without extension. Shifting to 2. or 3. level happens if line in file list.list begins with 1 or 2 TAB-symbols. Folders icons are added automatically (if the following line in the list.list file is on the next level then the previous line changes into folder). Comment-lines must be started with // , also empty lines are allowed (max 2 lines sequentially)

Following example illustrates generation of Schema-control panels schemas list from file list.list



```
// file list.list EXAMPLE
// shift lines using TAB key; line format:

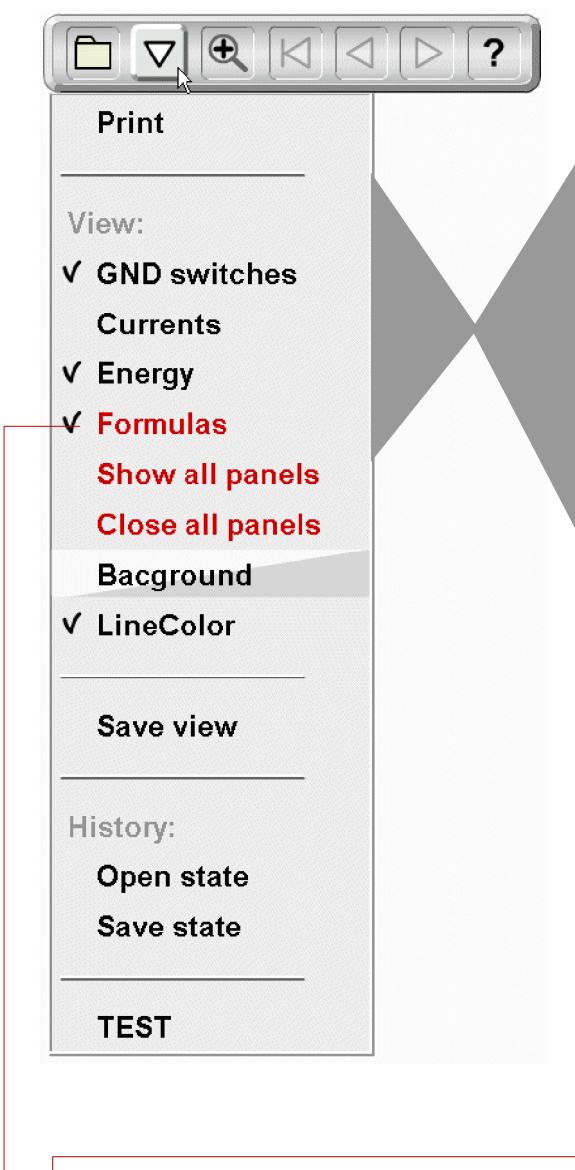
// name
// filename
// icon nol
// color nol
// opened folder(true)or additional info (number>1)

SCADA system's map|sample-map||0|
Substation no 1|sub1||8|
Substation no 2|sub2||8|
Substation no 3|sub3||8|
Substation no 4|sub4||4|
Substation no 5|sub5||4|
Substation no 6|sub6||4|
Additional schemas||0|true
Sub1|sub1||true
    Sub1 detailed drawing|sub1_drawing
    Sub1 project|sub1_project
    Sub1 equipments|sub1
Sub2|sub2
...
Sub3|sub3
...
Sub4|sub4
...
Sub5|sub5
...
Sub6|sub6
    Sub6 detailed drawing|sub6_drawing
    Sub6 project|sub6_project
    Sub6 equipments|sub1
User's Guidel|dispmanual||6|
```

## 5.2. Setting up submenu items.

Control.swf submenu can be configured by Control\_text.swf. For this open Control\_text.fla and edit action script of frame1, then publish Control\_text.swf. Changes take effect if Telem Schema System loads control.swf. The list of submenu items is in array **smi**. In the submenu of Schema-control panel only these layer names are shown which really exist for currently opened schema. If schema has pseudo layer (movie clip) with additional special name (variable: name = "Comments"), this name will also appear in submenu in different color

Example of script on frame of Control\_text.fla (text in **bold** may be edited by user):



```
//____SubMenuItem_____
// Edit only first array element -default layer name
// smi[x]=new Array("itemname", command nr, TYPE )
// TYPE: 1-normal item, 2-backgroundButton,
// 3- gray text, 4- additional line, 5-empty item
//Group1:
smi[10] = new Array("Print", "print", 1);
//Group2:
smi[0] = new Array("View:", "", 3);
smi[1] = new Array("GND switches", 1, 1);
smi[2] = new Array("Currents", 2, 1);
smi[3] = new Array("Energy", 3, 1);
smi[4] = new Array("Appendix4", 4, 1);
smi[5] = new Array("Appendix5", 5, 1);
smi[6] = new Array("Appendix6", 6, 1);
smi[7] = new Array("Appendix7", 7, 1);
smi[8] = new Array("Appendix8", 8, 1);
smi[9] = new Array("Appendix9", 9, 1);
//Group3:
smi[17] = new Array("Show all panels", "showmod", 1);
smi[18] = new Array("Close all panels", "closemod", 1);
smi[21] = new Array("Background", "back", 2);
smi[22] = new Array("LineColor", "lc", 1);
smi[23] = new Array("-", "", 4);
smi[24] = new Array("Save view", "saves", 1);
smi[25] = new Array("-", "hist", 4);
smi[26] = new Array("History:", "hist", 3);
smi[27] = new Array("Open state", "openh", 1);
smi[28] = new Array("Save state", "saveh", 1);
smi[29] = new Array("-", "", 4);
smi[30] = new Array("TEST", "test", 1);
//this name will be added to time in the schemas corner
histname = "History";
```

This item name comes from opened schema:  
swf-layer with instance name L4 has variable **name = "Formulas"**

LineColor status and background brightness are determined by configuration file (Sample-sub.cfg):  
linecolor=off&lcver=1&back=80&layer2=true&layer4=true&layer5=true&clicfg=true&cfg=Sample-sub

### 5.3. Editing file control.cfg

File control.cfg is placed into folder "Include" and could be edited with "SchemaHelp.html" or with any text editor (for example with "Notepad"). File "control.cfg" is in application/x-www-form-urlencoded format. Meaning of parameters values:

Parameter	Value	Comment
test	true ( or false )	In the Telem Design environment test-panel is enabled by default. If Telem Schema System runs in SCADA user's interface the test-panel is disabled by default even <i>test=true</i> . If parameter <i>test</i> not exists, test-panel is not available at all.
maxl	number	Next schemas, loaded using control.swf or using "LoadSchema" symbol on any schema, can be on levels 1,2,3 and so on, up to number, configured in the file contr.cfg as parameter maxl. (maxl=0 - only level0 is used, maxl=4 - additionally to level 0 is used 4 levels). Maximum value for maxl parameter is 8. Only one schema from loaded schemas (0..8) is currently visible. The visible schema is determined by state of control.swf.
maxm	number	The maximum number of additional modules loaded on the every schema level. If you load more modules, the schema system closes the first loaded.
dlcmap	string	Filename of Telem SCADA Systems map, must be loaded as the first schema
hidden	true ( or false )	Hide the schema-control panel if the simple substation system is used
help	false	Removes from schema-control panel the button with "?"
contrcfg	done	contrcfg=done indicates the end of file

Example for Telem Schema Design Environment (the Test panel is enabled):

**test=true&maxl=4&maxm=10&dlcmap=map&contrcfg=done**

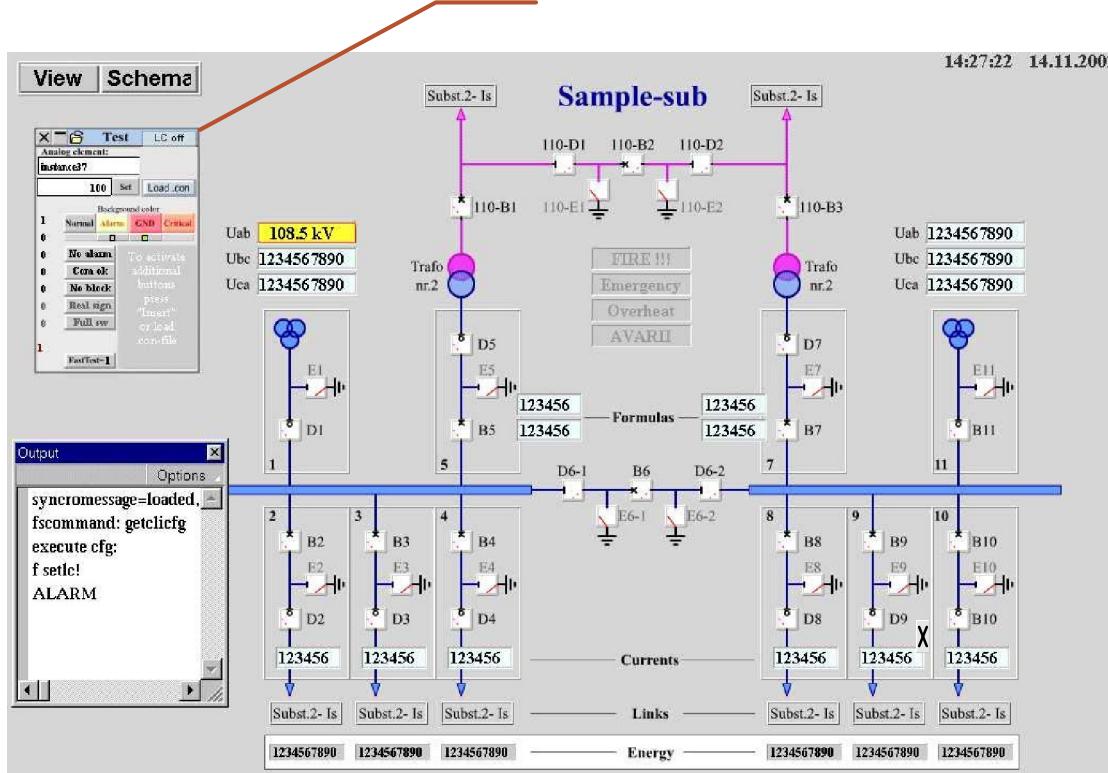
Example for Telem Sschema System, used by SCADA user's interface (SCADA client):

**maxl=4&maxm=10&dlcmap=map&contrcfg=done**

## 6. Test operations.

For testing the drawing results choose from menu **Control / Test Movie** or press (Ctrl+Enter). *Schema.swf* is generated from *schema.fla* and displayed in imported flash player. During this process the action script files from folder AS are included and configuration files *schema.cfg* and *control.cfg* are read. All settings of these files must be done, using "SchemaHelp.html".

If in *control.cfg* the test is enabled then **Test.swf** is loaded into flash player.



### 6.1. Test.swf description

*Test.swf* allows to control position of switches, analog values and all additional attributes of elements.

Open "SchemaHelp.html" or "SchemaTest&Demo.html"

"LineColor" status  
Drag "Test.swf"

**For testing click on any active SCADA schema symbol, test displays and controls following attributes:**

**Digital element:**  
Instance89

**Load .con**

**Open Close both sign.**

**0 no info**

**2**

**0 No alarm**

**0 Com ok**

**0 No block**

**0 Real sign**

**0 Full sw**

**2**

**FastTest=1**

**Symbols type**  
Target on the schema , includes **Instance name**

**Set analog measurements value**

**Set analog's background color** (status of value)

**Control binary signal's position**

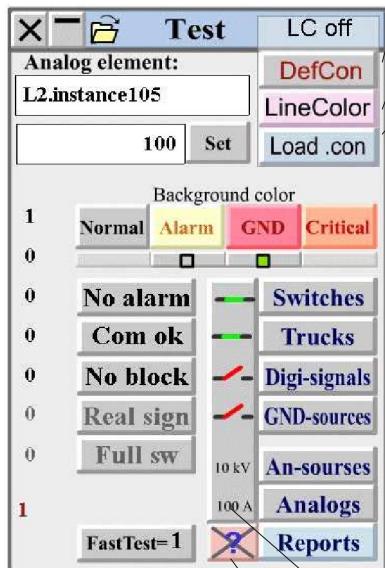
**Turn on/off symbols additional attributes:**

- symbols alarm** (blinking)
- communication status**
- blocking**
- real or pseudo signal

If "FastTest"=1 - every click on any binary element will reverse it's position (open/close), click on analog element will change the status - background color

## 6.2. Testing all symbols.

For testing all symbols with one command the symbols must be registered in this flash test environment. Registration is performed by pressing "Insert" key. If registration results are available, additional buttons on "Test" are visible.



Loads and executes DefCon.swf, which makes the connection analysis of schema elements

Loads and executes Linecolor.swf, which is used for coloring lines on schema

Loads schema.con file (if exists) - schema elements connection analysis is made by "DefCon" and results are saved to file *schema.con*

For testing together all elements of typical group click onto one of them, then:

1. Set the position (or value) and attributes as you like.

2. Pressing on buttons

"Switches",

"Trucks",

"Digi-signals",

"GND-sources",

"Analog sources" or

"Analogs" (all analogs)

causes the change of status and attributes of all elements in this group.

Pressing this button closes all switches and trucks, opens digital signals and GND-sources. Analog sources get value "100 kV", other analogs (not sources) get value "100".

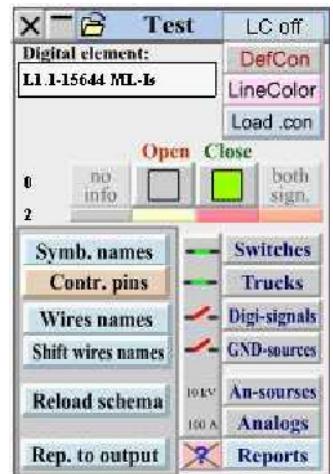
Pressing this button activates all attributes of all symbols.

### 6.3. Generating reports .

Reports to help to control symbol instance names - assiciation with database

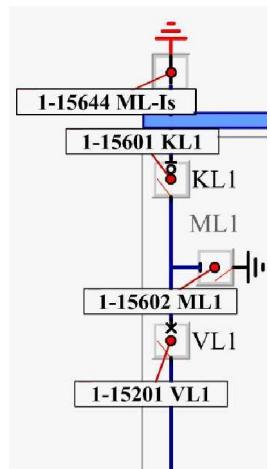
#### Show name labels of symbols:

- Press button "Symb. names"
- Drag one name label to desired position
- Rotate this name label from center,if it is necessary
- Press on group button ("Switches","Trucks",...) to set all labels of this element group to desired position
- Print schema for following check operations, if you like



#### Show wire names:

- Press button "Wire names"
- Drag one name label to desired position
- Rotate this name label, if you like
- Press on group button "Shift wires names" to set all wire labels to desired position
- Print schema for following check operations, if you like

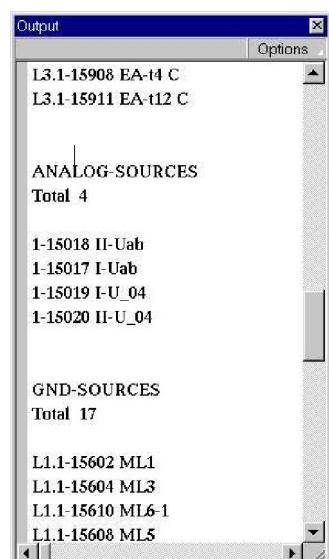


#### Show controller pin labels:

Button "Contr. Pins"- controller no. and pin no. appear on labels from database. This feature is available only if flash player is embedded into special program **DB-gizmo.xls** (not included in this design environment yet).

#### Report to output:

- Press button "Rep. to output"
- Check "Output" window - all registered elements are listed.



## 7. Symbols association with database objects

It is possible to use different methods to create schema symbols association with database table "Objects". Every method has some preferences depending on schema structure and the facts of the case. For each schema in the SCADA system may be used a selection of different methods.

The methods can be devided as static and dynamic methods.

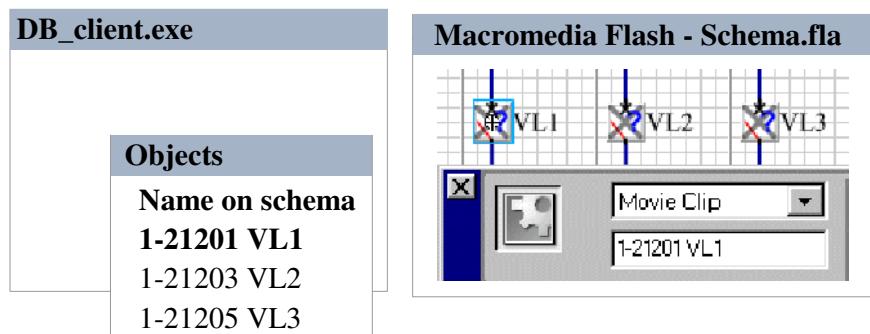
1. Static methods use for creating connection (between objects record in the database and objects symbol on the schema) files schema.log, schema.as or schemadbc. Data from these files are inserted into system server database by confclient.exe if it is used for adding new or updating existing schema.
2. Dynamic methods send information about schema elements to client.exe if schema is loaded. No accociation operations must be done previously.

## 7.1. Schema.log -method

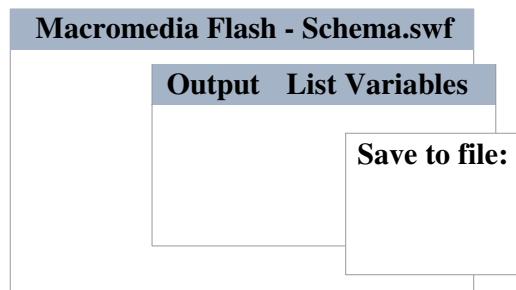
1. Open DB\_client.exe  
and schema.fla

2. Give names to all schema elements:  
  - Copy "Name on schema" from table "objects"
  - Paste this name to Flash symbol <Instance Name>.

Repeat these operations for all symbols



3. Create file schema.log:  
  - In Flash choose from menu "Control/Test Movie"
  - Choose from menu "Debug/ List Variables"
  - Data from "Output"-panel save to file schema.log



4. Copy files schema.swf and schema.log to server's folder Telem\_DBS/Build

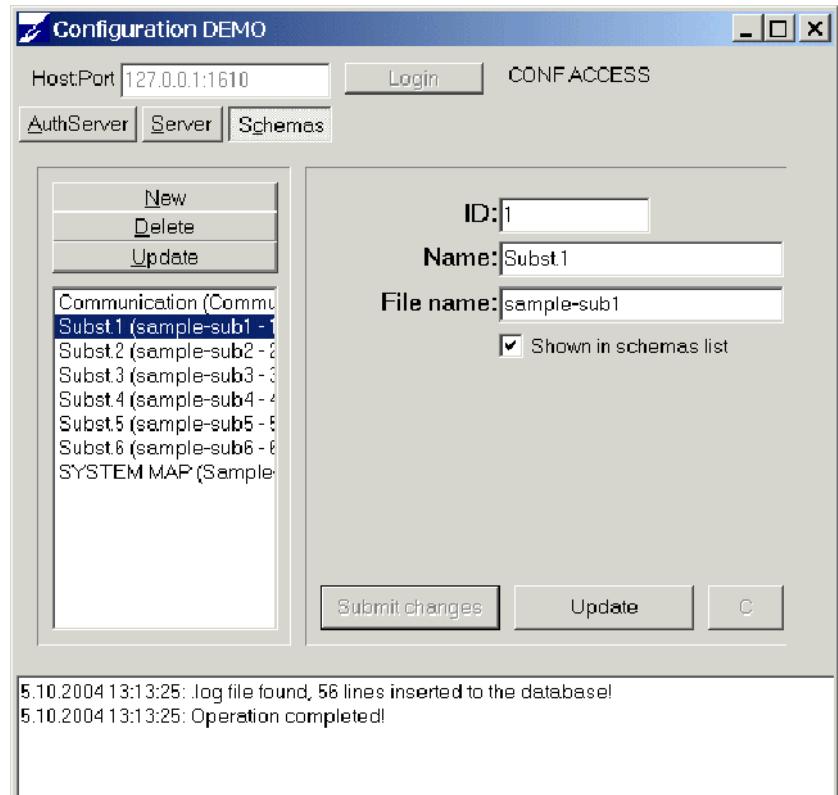
5. Open ConfClient.exe  
  - Press "Schemas"

For new schema:

  - Press "New"
  - Fill fields "Name" and "File name"
  - Press "Submit changes"

For existing schema:

  - Select schema from list
  - Press "Update"
  - Press "Server" and using buttons "Stop" and "Start" restart server.exe



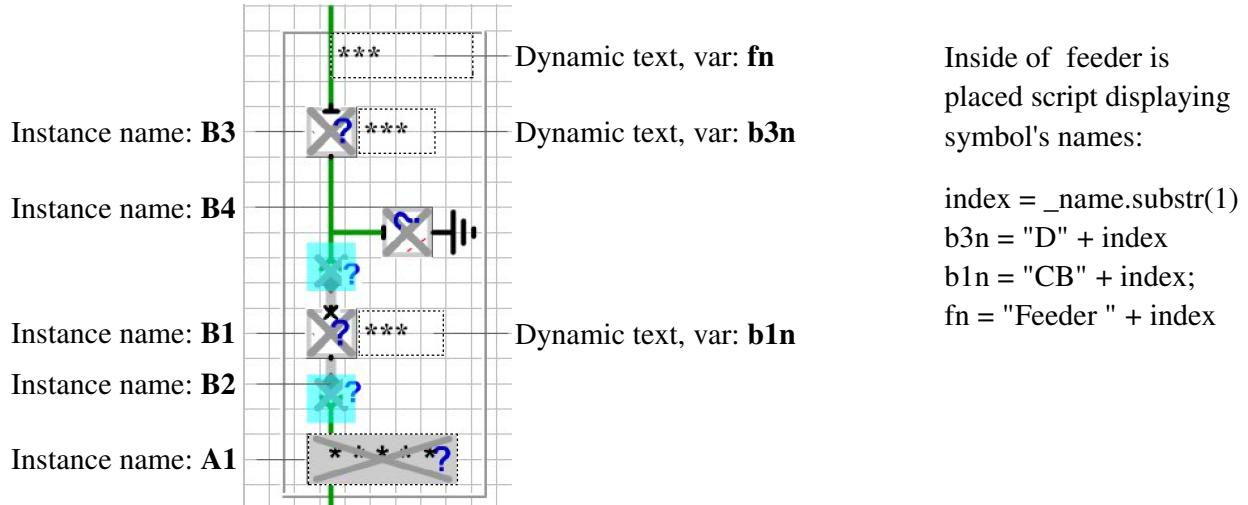
6. Open client.exe and check results

## 7.2. Schema.AS and schema.DBC -methods

To simplify drawing process it could be wise to present some schema parts as special new library elements (type "Movie clip"). For example - most of schema consists from identical feeders

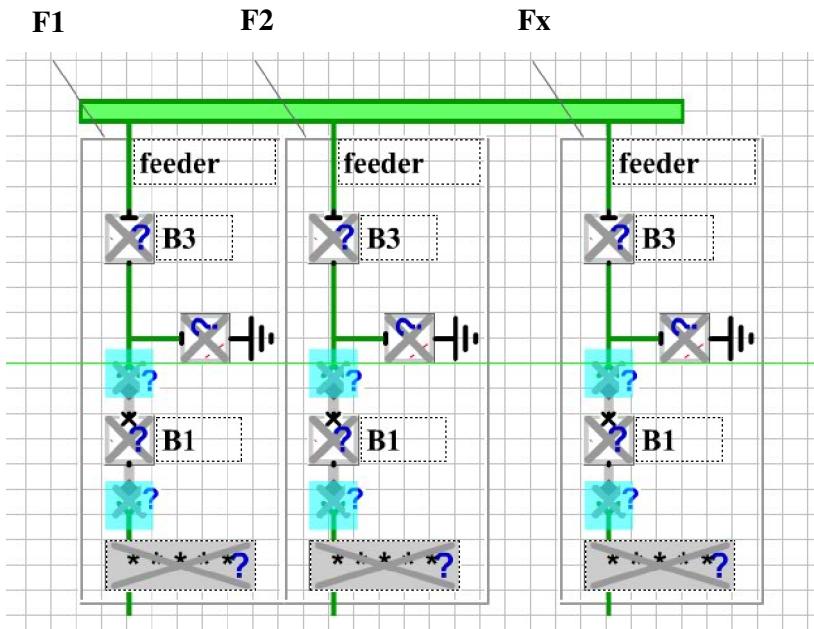
### 1. Create feeder symbol

Symbol "feeder" structure example:



### 2. Draw the schema using this feeder-symbol: drag copies to desired positions, give names to copies.

Instance name:



### 7.2.1. Schema.AS -method

1. Create shema.as file for renaming feeders elements, using **DBgizmo.xls**

- Fill column "DBC for schema":
  - Show start row for feeder (feeder name)
  - Show feeders objects (any character)
- Save shema.as file

Object's ID	NAME	Name on schema	DBC for schema
20200	Feeder 2	1-20200 Feeder 2	
20204	Ia 2	1-20204 Ia 2	F2.
20210	CB 2		
20211	CB 2	1-20211 CB 2	#
20212	Truck 2	1-20212 Truck 2	#
20213	Disconn.2	1-20213 Disconn_2	#
20214	Earth 2	1-20214 Earth 2	#
20300	Feeder 3	1-20300 Feeder 3	
20304	Ia 3	1-20304 Ia 3	F3.
20310	CB 3		
20311	CB 3		
20312	Truc		
20313	Disc		
20314	Earth		
20400	Feed		
20404	Ia 4		
20405	Ener		
20410	CB 4		

More Actions

UNHIDE all columns

HIDE empty columns

Show NAME ON SCHEMA

Generate NAME ON SCHEMA

**Save .DBC file** **Save .AS file**

Gen. pinsproject **Save .SDAT file**

Gen.GLOBAL ID Make SHEET

Clear sheet **Close**

Database connection with schema, using included file in FLASH

Folder: D:\TELEM\_WWW\TELEM\_SCHEMA6\Docs\DBC

File: GlobalDesign.as

DBC column: Choose DBC column!

Use rows: All rows of Excel table

Save Cancel

2. Add to schema.fla "Actions-Frame" - script #include "DBC/schema.as" for renaming symbols of feeder :

File schema.as example:

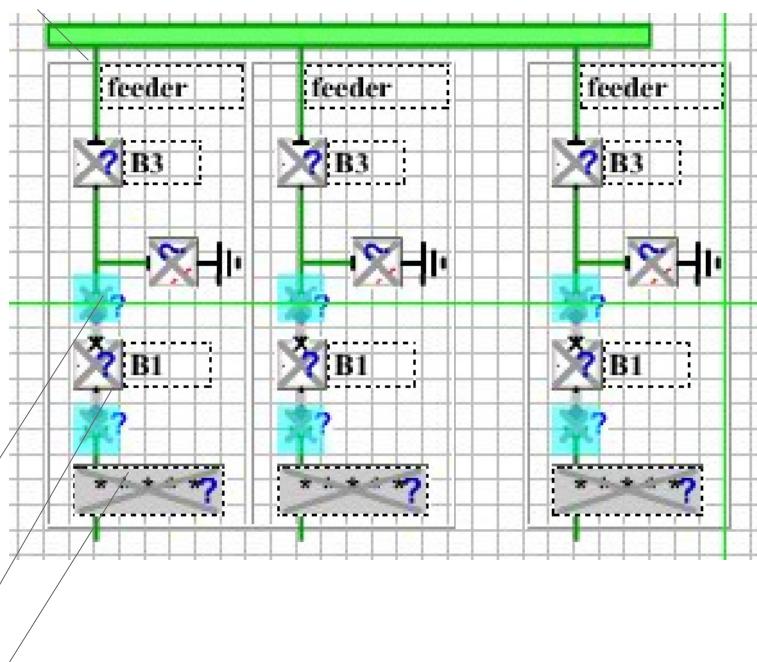
```
Instance name: F1
...
F1.A1._name = "1-20104 Ia 1";
F1.B1._name = "1-20111 CB 1";
F1.B2._name = "1-20112 Truck 1";
F1.B3._name = "1-20113 Disconnector 1";
F1.B4._name = "1-20114 Earth 1";
F2.A1._name = "1-20204 Ia 2";
F2.B1._name = "1-20211 CB 2";
F2.B2._name = "1-20212 Truck 2";
F2.B3._name = "1-20213 Disconnector 2";
F2.B4._name = "1-20214 Earth 2";
...
```

In published schema.swf  
the symbols have now new  
names:

1-20112 Truck 1

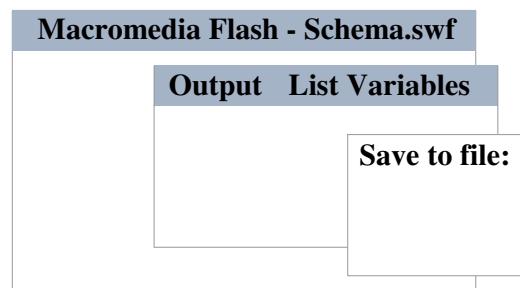
1-20111 CB 1

1-20104 Ia 1



The following steps are identical with log-method.

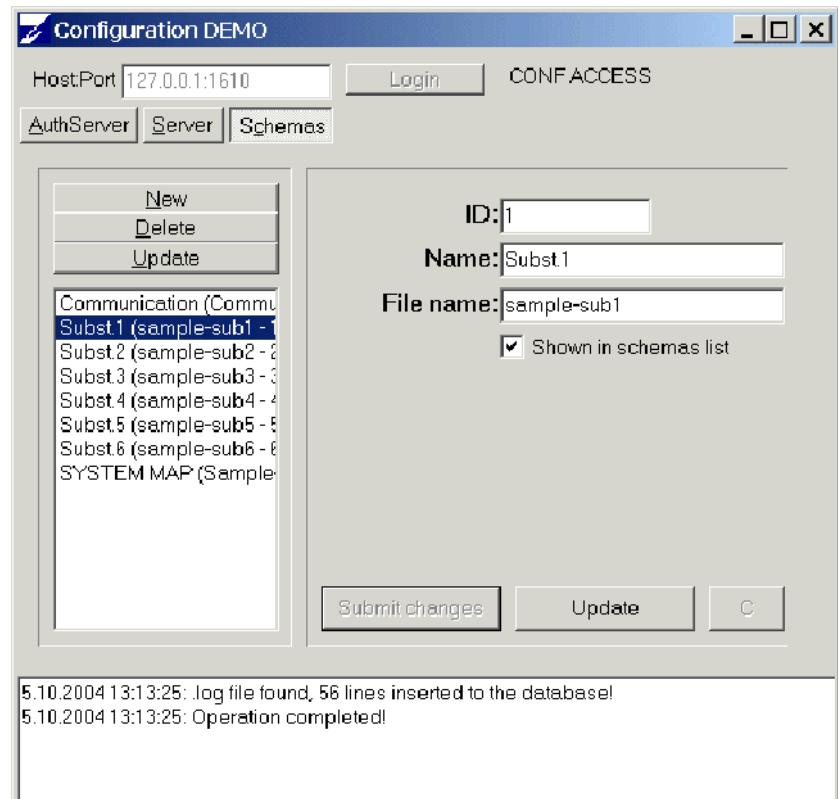
3. Create file schema.log:  
 - In Flash choose from menu "Control/Test Movie"  
 - Choose from menu "Debug/ List Variables"  
 - Data from "Output"-panel save to file schema.log



4. Copy files schema.swf and schema.log to server's folder Telem\_DBS/Build

5. Open ConfClient.exe  
 - Press "Schemas"  
 For new schema:  
 - Press "New"  
 - Fill fields "Name" and "File name"  
 - Press "Submit changes"  
 For existing schema:  
 - Select schema from list  
 - Press "Update"  
 - Press "Server" and using buttons "Stop" and "Start" restart server.exe

6. Open client.exe and check results



## 7.2.2. Schema.DBC -method

1. Create schema.DBCfile for renaming feeders elements, using **DBgizmo.xls**

- Fill column "DBC for schema":

- Show start row for feeder  
(feeder name)
- Show feeders objects  
(any character)

- Save schema.DBCfile

Object's ID	NAME	Name on schema	DBC for schema
20200	Feeder 2	1-20200 Feeder 2	
20204	Ia 2	1-20204 Ia 2	F2.
20210	CB 2		
20211	CB 2	1-20211 CB 2	#
20212	Truck 2	1-20212 Truck 2	#
20213	Disconn.2	1-20213 Disconn_2	#
20214	Earth 2	1-20214 Earth 2	#
20300	Feeder 3	1-20300 Feeder 3	
20304	Ia 3	1-20304 Ia 3	F3.
20310	CB 3		
20311	CB 3		Database connection with schema, using included file in FLASH
20312	Truc		
20313	Disc		
20314	Earth		
20400	Feed		
20404	Ia 4		
20405	Ener		
20410	CB 4		

More Actions

UNHIDE all columns

HIDE empty columns

Show NAME ON SCHEMA

Generate NAME ON SCHEMA

**Save .DBC file** **Save .AS file**

Gen. pinsproject **Save .SDAT file**

Gen.GLOBAL ID **Make SHEET**

Clear sheet **Close**

Folder: D:\TELEM\_WWW\TELEM\_SCHEMA6\Docs\

File: schema.DBC

DBC column: Choose DBC column !

Use rows: All rows of Excel table

Save Cancel

File schema.DBC example:

```

...
F1.A1._name = "1-20104 Ia 1";
F1.B1._name = "1-20111 CB 1";
F1.B2._name = "1-20112 Truck 1";
F1.B3._name = "1-20113 Disconnector 1";
F1.B4._name = "1-20114 Earth 1";
F2.A1._name = "1-20204 Ia 2";
F2.B1._name = "1-20211 CB 2";
F2.B2._name = "1-20212 Truck 2";
F2.B3._name = "1-20213 Disconnector 2";
F2.B4._name = "1-20214 Earth 2";
...

```

**File schema.DBC is identical with file the schema.as, but it is not included to schema.fla**

Adding to schema.fla "Actions-Frame" - script #include "DBC/schema.DBC" for renaming feeders symbols is possible and useful for test actions. In this case script include "DBC/schema.DBC" must be removed before final SWF export for server

The following steps are similar with log-method.

2. Copy files schema.swf and schema.DBC to server's folder Telem\_DBS/Build

3. Open ConfClient.exe

- Press "Schemas"

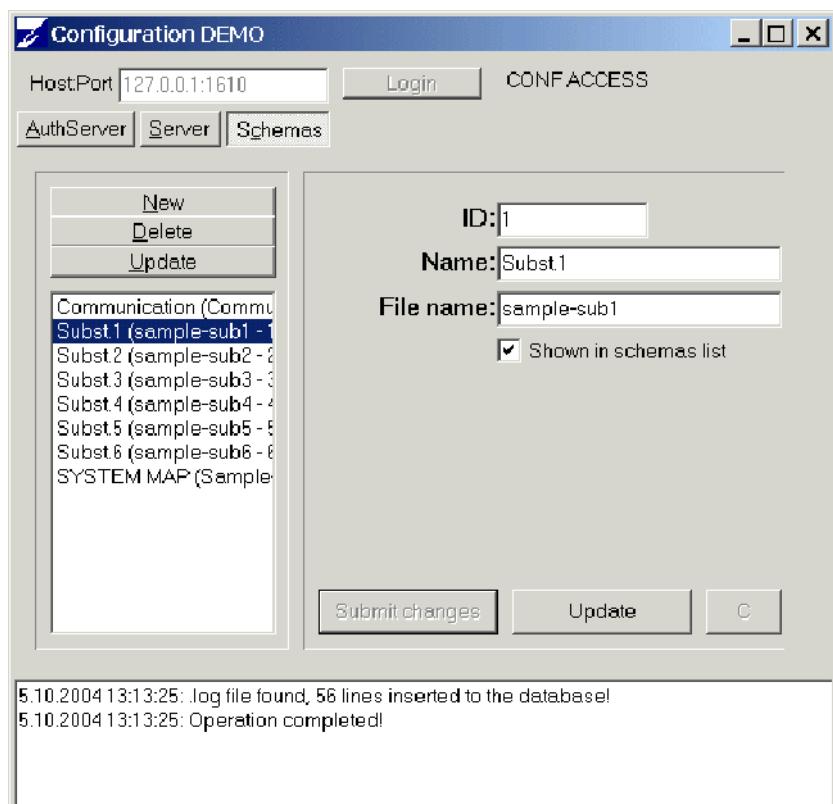
For new schema:

- Press "New"
- Fill fields "Name" and "File name"
- Press "Submit changes"

For existing schema:

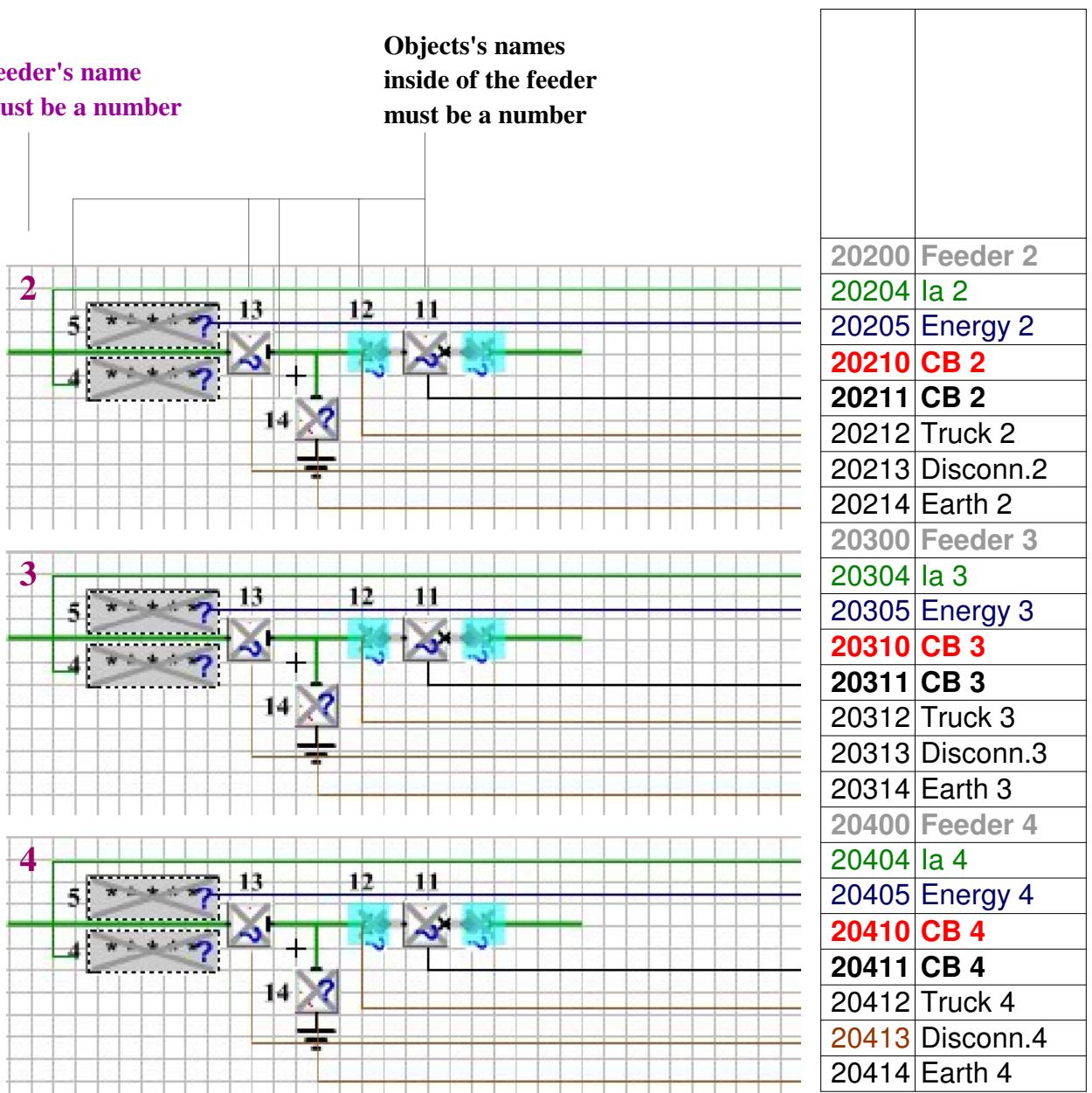
- Select schema from list
- Press "Update"
- Press "Server" and using buttons "Stop" and "Start" restart server.exe

4. Open client.exe and check results



### 7.3. DBCgen -method

This method allows to create connection between database and flash schema without any additional files and actions with confclient.exe. The design of database table "Objects" and flash schema must be coordinated - the feeders structure on the schema must be carried over to database.



1. Add to schema.fla "Actions-Frame" - scripts:

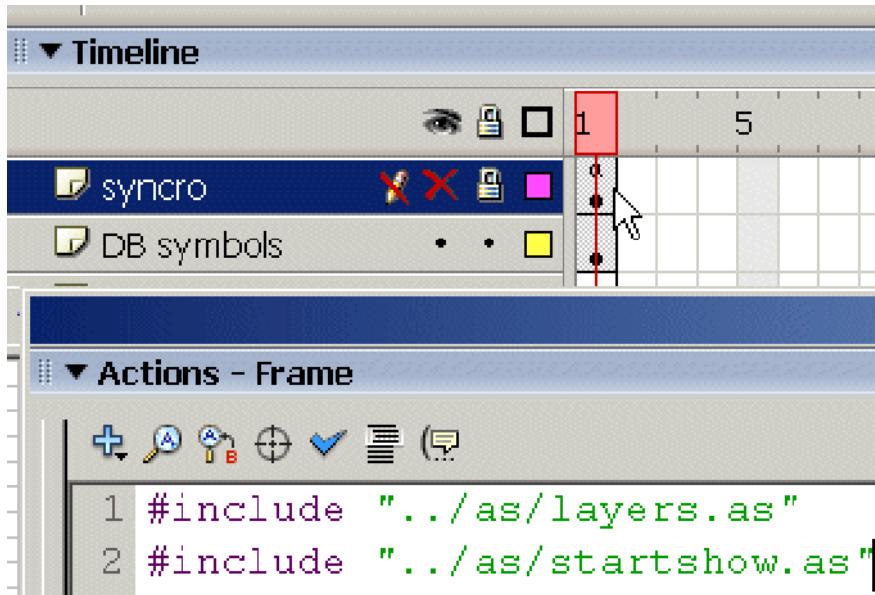
```
conc = 1; //concentrator no.1
baseID = 20000; //ID-numbers of this substations objects starts from 20000
DBCgen_1 = "2-4;4,5,11-14"; // feeders 2,3,4 have objects 4,5,11,12,13,14
#include "../as/startshow.as"
```

Result: all schema's object are connected with database objects using formula:

Object's ID (in database)=

baseID + 100 \* feeder's name + object's name on the schema

## 8. Scripts and variables for schema's root



```

// The variable filename for this schema (without extension) is required for
// schema design environment versions 1...5.
filename = "schema";
// For version 6 or later variable filename is optional

```

### **// Variables for loading and configuration actions**

```

// The optional variable contr_name determines control level movies name. Default value is "control"
// The variable contr_name is activated only if this schema is loaded to level0
contr_name = "control_3x100faf";

```

```

// The optional variable contr_text_name determines text-file name for configuration of
// control level movie (for example: control_3x100faf.swf). Default value is "contr_text"
// The contr_text_name is activated only if contr_name is activated by previous scripts line
contr_text_name = "contr_text_example";

```

```

// The optional variables scfg and cfg allows to load this schema without using file schema.cfg
cfg = "NO"; // Used for non-SCADA schemas
scfg = "lcoff"; // LineColor is off for testing

```

```

// The optional script #include "../as/layers.as" allows to group schemas objects into swf-layers,
// if these objects are already placed into symbols -feeders (type - movie clip). Object added
// to swf-layer must have the script #include "../as/addtoLx.as" where x is the layers number 1...9
#include "../as/layers.as"

```

**// Variables for dynamic connection with database (dynDBC)**

// Concentrator's number for schema's objects. Default value is 1  
**conc = 1;**

// Common variable baseID is used by some dynDBC methods for calculate objects ID  
// The advanced symbols may have and use their own variable baseID for creating dynDBC  
**baseID = 20000;**

// Coefficient for Feeder if dynDBC method calculates objects ID, default value is 100  
**Fcoef = 10;**

// Script for activating DBC generator. Uses variables conc, baseID and Fcoef.  
**DBCgen\_1 = "1-4,6,8-10;1-13,21-26";**  
**DBCgen\_2 = "5,7;1-3,23,24";**

**// Various additional variables**

// Turn off the unit of analog measurements (optional) , default value is true  
**Aunit = false;**

// Send continuously real-time data to this schema, if it is still loaded on its level (optional),  
// default value is false  
**hold = true;**

**// Properties for LineColor (OPTIONAL):**

// For using "Autocolor switches"  
**sw\_color = true;**  
// For using color transform property without "sw\_color = true"  
**c\_transf = true;**  
// Reload this schema always to level0, it is the "LineColor map"  
**lcmap = true;**

**// The conclusional line**

**// The bottom line of schemas root script must be #include "../as/startshow.as" if dynDBC is used**  
**// in other cases it may be #include "../as/common.as" or #include "../as/common7.as"**  
**#include "../as/startshow.as"**

## 9. Using SchemaHelp.html.

SchemaHelp.html is a main instrument for schema design and test process.  
SchemaHelp.htm consists following tabs:

- **General** - general description of steps needed for adding a new schema into SDADA system.
- **Schema test & export** - configuration of schema's design environment, description of configuration files, copying files to server.
- **Steps of schema design** - sequence of actions with schema if "LineColor" is used.  
Additional tab to "Schema test & export" is useful for beginners.
- **Colors** - for configuring colors, used by TelemSchema system.
- **Drawing with LineColor** - rules for placing schema elements if "LineColor" is used..
- **About library** - some remarks about active symbols features.

### 9.1. Color settings

Colors, used on the flash schema can be divided into 2 groups:

1. **Static colors** are displayed as they are drawn. If "LineColor" wires coloring system is used then lines, feeders, transformers and etc. is useful to draw with the color what is used for displaying the normal status of them.
2. **Dynamic colors** are changed at operation time and they represent the status information of the system. All dynamic color tone descriptions are placed into file colors.as (folder ..;/include/control/). Edit this file manually or use "**SchemaHelp.html**" (folder ..;/help/).

The background color of analog measurement depends on the status of the measured value ("normal", "alarm", "critical" or "near to zero"). These colors RGB-code is placed into array **abc[0...3 ]**. This array (file colors.as) should be edited only with "SchemaHelp.html". Every time when colors are saved, the array **abc[ ]** is saved into two files: "...;/include/control/colors.as" and "...;/as/abccol.as". The array **abc[ ]** is used by schema elements via included action script file "...;/as/abccol.as" when **Schema**.swf is generated from **Schema**.fla. So changes in analog measurement background colors will take effect after publishing new **Schema**.swf.

Colors used by wire coloring system "LineColor" are described in arrays: **coll[ ]** - for simple line figures; **colf[ ]** - for filled areas. The arrays **coll[ ]** and **colf[ ]** are used while file **contr\_text.swf** is generated from **contr\_text.fla**. So changes in "LineColor" system will take effect after publishing new **contr\_text.swf**

Open - read data from file colors.as

**File colors.as**

```
// ANback=NORMAL
abc[0] = 0xEEFFFF;
abc[4] = 0xEEFFFF;
// ANback= ALARM
abc[1] = 0xFFFF66;
abc[5] = 0xFFFF66;
// ANback= CRITICAL
abc[3] = 0xFFBB33;
abc[7] = 0xFFBB33;
// ANback= Near to zero
abc[2] = 0xCCCCCC;
abc[6] = 0xF33333;
```

This part is additionally saved as file abccol.as

**Colors configuration file: ..\Include/Contro/ colors.as**

Analog background colors		U-source		Colors for lines, bars,transformers if LineColor is used		Colors for lines, not connected to sources	
0	AnBack:NORMAL	EEFFFF	EEFFFF	6	ALARM	33	FFFF99
1	AnBack: ALARM	FFFF66	FFFF66	5	Critical	66	FFCC00
3	AnBack: CRITICAL	FFBB33	FFBB33	4	ZERO	FF9AFF	FFFFFF
2	AnBack= Near to zero	CCCCCC	FF3333	3	GND	330000	99A866
				0	SHOTCUT	FFFFFF	FF66FF;

**Normal colors**

10	NORMAL c0	0	CCCCCC	20	Open circ. c0	989999	FFFFFF
11	NORMAL c1	980000	FF6699	21	Open circ. c1	989999	eeFFFF
12	NORMAL c2 main	986600	CC9866	22	Open circ. c2	989999	eeFFFF
13	NORMAL c3	888800	CCCC00	23	Open circ. c3	989999	eeFFFF
14	NORMAL c4 main	339900	66FF00	24	Open circ. c4	989999	FF&FF
15	NORMAL c5	9866	FF99	25	Open circ. c5	989999	FF&FF
16	NORMAL c6 main	33FF	96CCFF	26	Open circ. c6	989999	FFF&ee
17	NORMAL c7	3333CC	9999FF	27	Open circ. c7	989999	FFF&ee
18	NORMAL c8 main	660099	CC66FF	28	Open circ. c8	989999	FFF&FF
19	NORMAL c9	980066	CC86CC	29	Open circ. c9	989999	FF&99FF

**Line-color**

**Fill-color**

**Schema Demo**

Edit RGB-code and comment or choose from palette new line- and fill-colors.

**Less Colors**

**Resume ↵**

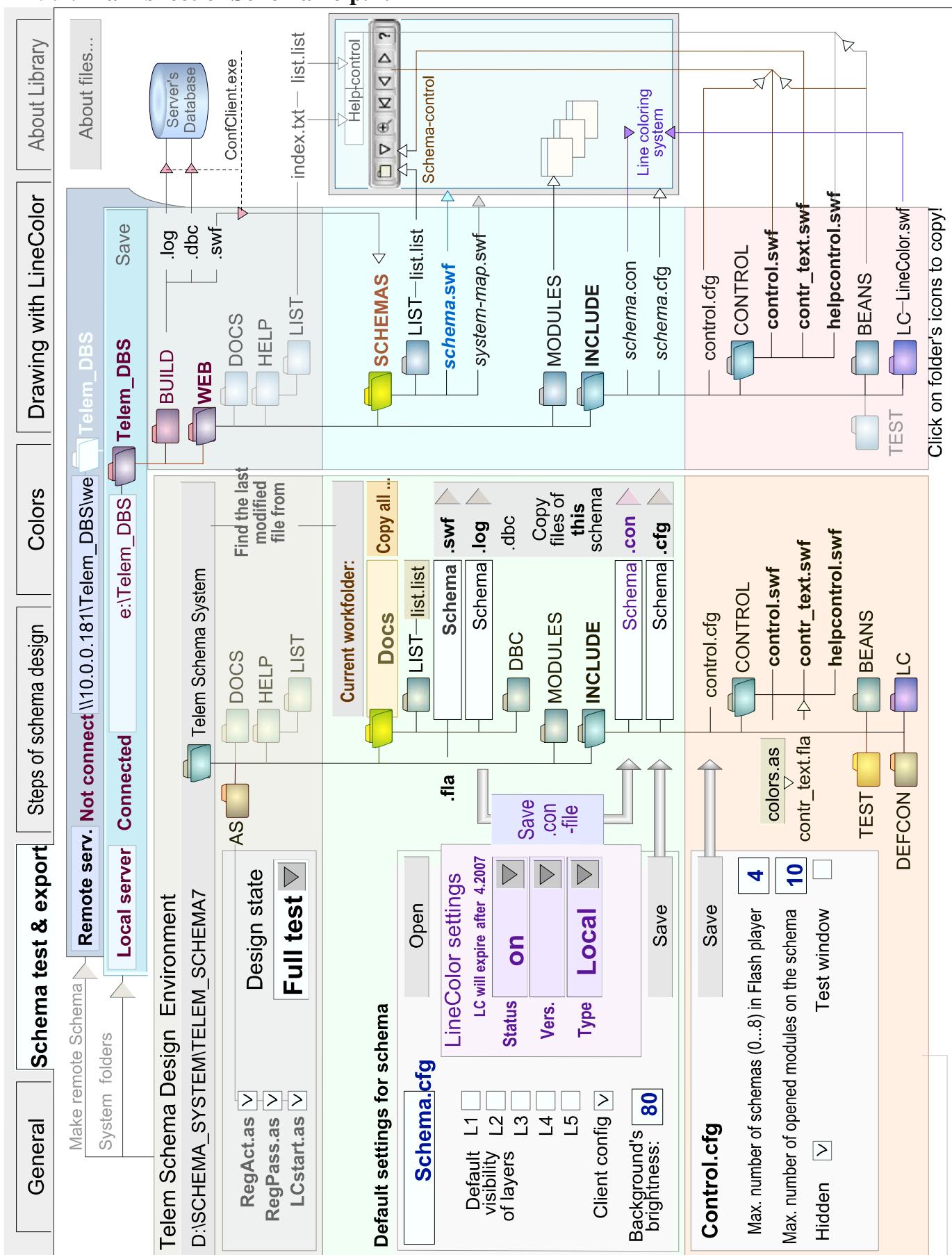
**BACKGROUND**

To add new color description specify line- and fill-colors, and edit comment in last row of this table, then save file. (Both boxes - "normal color" and "... not connected lines" must be filled).

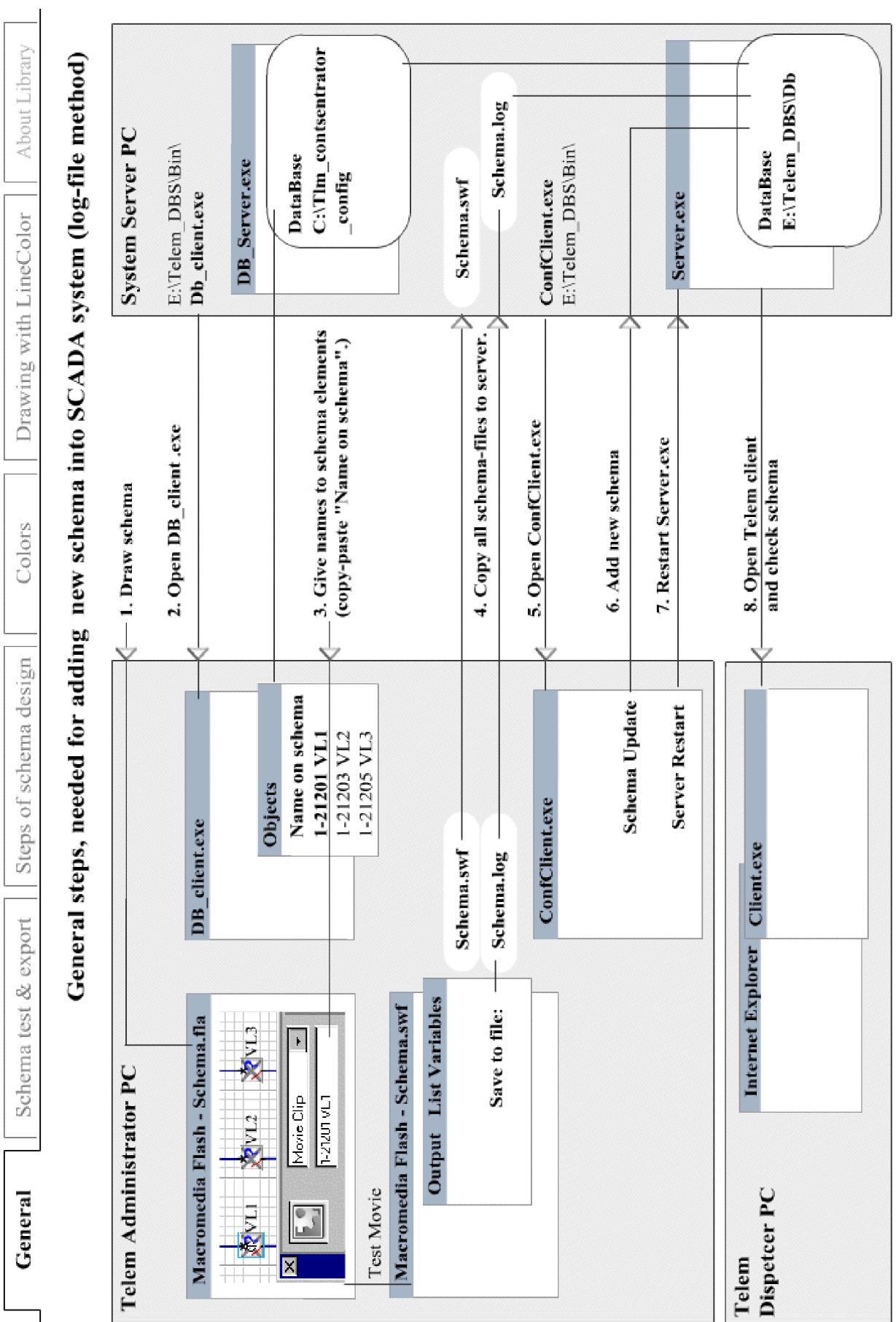
To delete color description clear fill color or comment from last row of this table. Do not delete colors with index 10 or 20

**37**

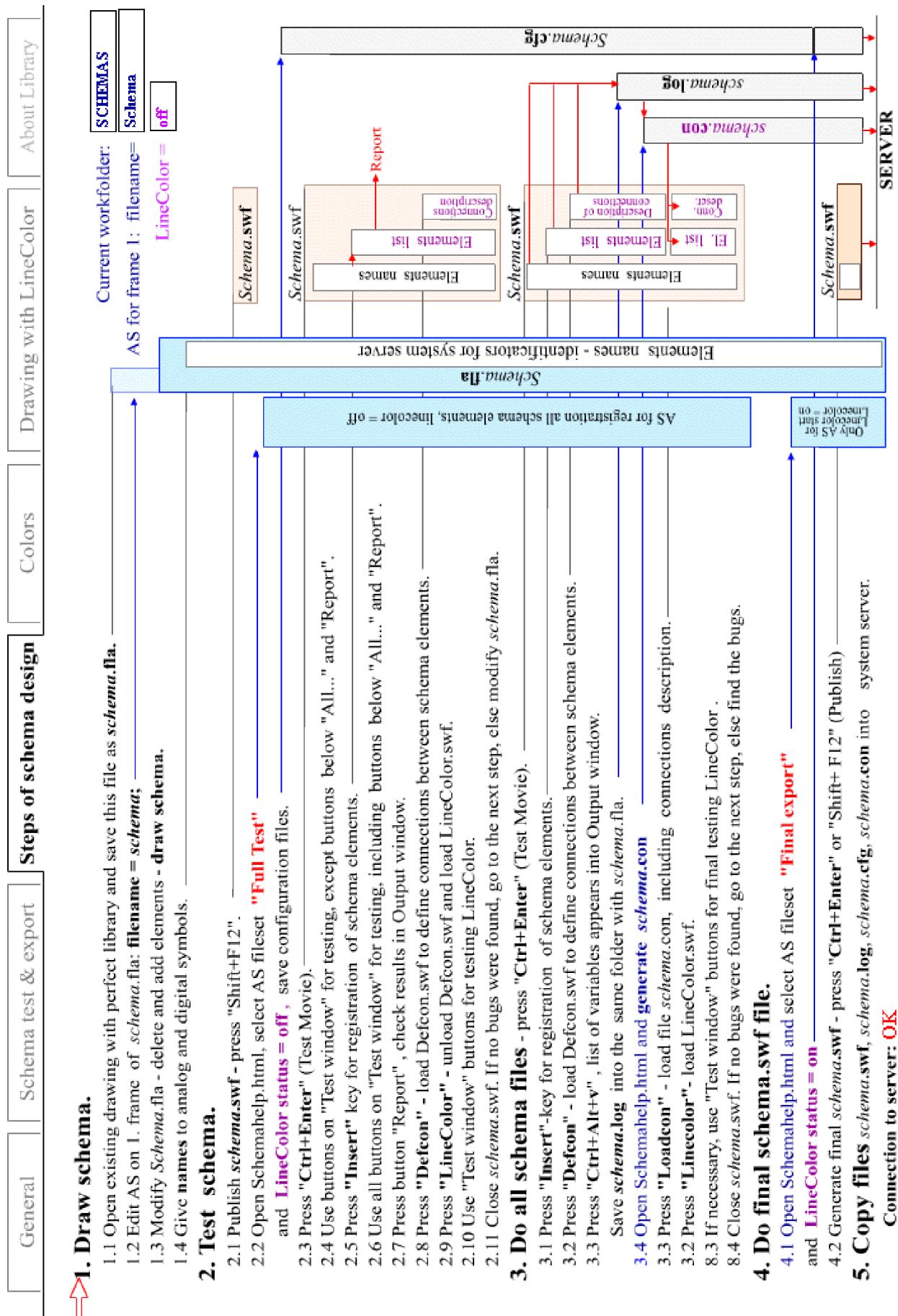
## 9.2. Main sheet of SchemaHelp.html



### 9.3. General - SchemaHelp.html



## 9.4. Steps of schema design - SchemaHelp.html



## 9.4. Drawing with LineColor - SchemaHelp.html

[General](#) [Schema test & export](#) [Steps of schema design](#) [Colors](#) [Drawing with LineColor](#) [About Library](#)

**Types of schema elements of "LineColor".**

**1.Sources.**

1.1. Analog sources.  
Analog elements, include script `#include "../as/RegAcls"`

1.2. Digital GND-sources.  
All GND-switches are defined as GND-sources.

Digital elements (signals), include script `#include "../as/RegDACLs"`

1.3. Sources of normal voltage.  
Lines, include script `#include "../as/RegAcls"`

**2.Switches.**  
All switches and trucks are defined as switches (not GND-switches!).

**3.Lines**

All lines, bars, transformers, fuses and so on are defined as lines .  
To improve performance (diminish connections' file size) and simplify drawing it could be wise to present some schema parts as special new library elements (type=line). Example :  
new element (clip) "BarWithLines", includes clips named "line" and "fill"  
Size of new element  
(connection area)

Clip "RegLineFill",  
recommended name:  
**rlf**

`#include "../as/RegJcl.as"`

**4**

**All graphically connected unnamed lines are automatically considered as physically connected. In case of graphically connected named lines it's possible to turn off the physical connection if their names start with the same capital letter. That is useful if presenting crossing lines.**

**Logical connections between coils of transformer may be:**

1. Connected - unnamed lines (coils).
2. Not connected - coils names start with the same capital letter
3. Asymmetrical transformer connection - coils names must start with the different numbers, color transformation direction : to coil with minor name - number. Status "normal", "alarm" and "critical" will be transformed to secondary coil as status "normal".
4. Symmetrical transformer connection - coils names must start with the same numbers, color transformation logic is the same (like asymmetrical transformer), but occur in both directions.