User guide

TopoL for Windows

Application Forms

TopoL Software s.r.o.

Content

APPLICATION FORMS 5

FORM TYPES 5
Form with text..... 6
Form with raster..... 6
Form with video 6
Information form..... 7
Sound accompaniment..... 7
FILE OF APPLICATION FORMS DESCRIPTION 7
Header application forms description 7
Section of descriptions of forms 8
Items for description of forms are like this 9
FILE WITH INFORMATION FORM DESCRIPTION 10
Information form heather..... 11
Declaration part of information form..... 12
Main body of information form description 12
EXAMPLE OF APPLICATION FORMS DEFINITION 13

Application forms

For objects of type point, polyline and area the application forms can be created. In the case of selecting these graphical objects (command User Info) the set of forms is displayed. User can select offered forms.

Application forms enable to display various information connected with single objects. Forms can include database, text, raster, video or user defined information. Form can also include button for sound. Contents of the application form can be defined for each type and group separately. Database, text, raster, information form an sound can be defined for all objects of certain type and group directly by its full path or indirectly by the field of DB which defines the name of the file

Contents of the application form is defined by text file which is unique for the type and the group of the object. This file includes the description of single forms. Each description directly or indirectly includes corresponding file. The description of information form is also text file which is defined by special language related to Pascal.

Form types

A set of application forms can content one or more forms of several different types. At present there are implemented four form types: form with a text, form with a raster, form with a video and an information form. All the forms besides the form with a video can content a button specified as a *Sound* for an invocation of sound accompaniment. The program makes automatic padding of it in case that there is Item for sound mentioned within the framework of a form description. To evoke a sound accompaniment you have to press a button by mouse or a corresponding hot key. Each form can be also completed by buttons for printing of a form content and its copying to clipboard.

Form with text

Form with text displays a text document which is saved in file defined directly or indirectly in a description of form. A file with clean text without any special format characters (ASCII text) is supposed. If the text contents characters with diacritics, coding of diacritics must be up to coding page set for block to which objects a form is joined, is necessary for right displaying of the text. The end of single lines of a document is specified by character CR (code 13) which can be followed by character LF (code 10), which are usually generated by pressing of a button.

Form with raster

Form with a raster displays a raster picture which is saved in the file defined directly or indirectly in a form description. Raster can be in formats RAS, TIFF or BMP and has to be placed in co-ordinates despite ignoring of its placing during displaying. Implicitly a size of a picture specified by dimension of the picture in pixels is used for drawing of raster. During definition of form extension or reduction of picture size may be set..

Form with video

Form with video allows viewing of a videoclip, which is saved in a file defined directly or indirectly in description of form. To a form a button marked *Video* allowing playing over set clip is completed. Program TopoL uses standard instrument of Windows - Media Control Interface system to treat a videoclip. Accompaniment sound can be also a part of file with videoclip. System Video for Windows must be installed to treat files in AVI at concrete computer. For treating of another forms (QuickTime, MPEG, etc.) is necessary to have installed corresponding driver - MCI driver (for ex. QuickTime for Windows) at computer.

Information form

Information form is defined by file with its description. It creates text file which defines content of information form by language ToPas which has syntax similar to language Pascal.

Sound accompaniment

To treat file with sound program TopoL uses standard instruments of Windows - Media Control Interface system. It means that playing over sounds wave (files with extension WAV) and files MIDI (files with extension MID and RMI) is possible in standard way. For processing of sounds in another forms it is necessary to have corresponding driver of media - MCI driver installed at computer..

File of application forms description

The description of application forms is made in text file, which contents optional heading and sections with description of single forms. The file can have any number of empty lines or lines with comments (they start with ;). Lines with comments are introduced by character semi-colon (;). Using of VELKYCH A MALYCH PISMEN within the framework of application forms description is not important. The name of this file is defined by the type of object and its group. It has got always the extension NDI and is saved in working directory, which is defined by starting of TopoL. In case that file isn't found in working directory, it is also seeking in directory of working block.

```
Example:
for areas of group 9 file has name P0009 NDI
for polylines of group 562 file has name
L0562 NDI
```

Header application forms description

Heather of application forms description is optional and is always mentioned in the beginning of application forms description file. Heather can content items which defines caption of window with displayed forms and implicit place and size of application forms window.

Definition of window caption has figure:

Header = text

Parameter text is any string of characters, which will be written to window caption.

Definition of window implicit area and size has a figure:

X = horizontal_position
Y = vertical_position
W = width
H = height

Parameters **width**, **height**, **horizontal_position** and **vertical_position** are specified as integral numbers which define position and size of whole window in pixels. It isn't necessary to define all explicit parameters.

Section of descriptions of forms

All descriptions are defined by its key word which defines type of the form. It can be optionaly completed by direct or indirect specification of file with sound accompaniment and also by competent form caption and its other parameters determination. In case that form description contents both direct and indirect file specification program will choose direct specification.Form description is finished by presence of key word at the beginning of next form description or in the end of file. The beginning of form description has a figure:

[key word]

Parameter key word is put into parenthesis and can come into value:

TXT for forms with text,

RAS for forms with raster,

AVI for forms with video,

DIN for information form.

Items for description of forms are like this

Descr = text

Parameter **text** can be any text which will be the caption of the form

Tab = text

Parameter **text** can be any text which will be the description on marker. Length of used string is limited at 16 characters. If this value isn't defined, parameter Descr. value will be used for writing at marker. If this isn't also defined, number of form will be used.

File = file_name

Parameter **file_name** defines directly the name of desired file. It can be full or relative path defined to working directory but always with extension

Item_for_file = field_name

Parameter **field_name** defines the name of internal DB field, where the name of file is to be found. Filed name is specified by full path to file or in relative way owing to working directory which is defined by starting of TopoL. Filed name must have competent extension. By this way form content is defined indirectly and can be defined for each object of competent group.

SoundFile = file name

Parameter file name defines directly name of file with sound accompaniment for just describing form. File name is specified by full path to file or in relative way according to working directory which is defined by starting of TopoL. File name must have competent extension.

Item_for_sound = field name

Parameter field name defines the name of internal DB field, in which the name of the file with sound accompaniment for just describing form is saved. File name saved in this field of DB is specified by full path to file or in relative way according to working directory which is defined by starting of TopoL. File name must content competent extension. By this way a sound accompaniment is defined indirectly and can be defined for each object of competent group.

`Clipboard = agreement`

Parameter **agreement** is created by random string of characters. If the first character is A, Y or T, form is completed by button *Clipboard*. By means of this button a form content can be copied into clipboard. By this way a simply connected document or results of calculations from information form can be used in another program (for ex. in text editor).

`Print = agreement`

Parameter agreement is created by random string of characters. If the first character is A,Y or T, form is completed by button *Print*. By means of this button a form content can be printed by printer.

`Zoom = size`

This item is important only for forms with raster picture. Implicitly for drawing of raster a size of picture defined by dimension of a picture in pixels is used. Parameter size is put as an integer and allows to set extension (reduction) of raster picture in percents according to original size. Resulting size of picture is then rounded to integral pixels. Value 100 then define original size of picture. Value 50 then means reduction of picture to a half, while 200 defines extension to double multiple.

File with information form description

Information form description is formed by text file which defines information form content by language ToPas which has syntax derivated from language Pascal. Description of language in a reference guide Language ToPas contents

overviw of single language marks. In any case it cann't be taken as a programming textbook.

Self information form description consists of form heather, of a declaration part and of main body form description.

Form heather is obligate and specify type and group of object for which form is defined. Optional declaration part allows to define constants, variables and symbolic names for used DB fields and blocks of graphic objects. Main body of form description consists of command sequence which are made during form elaboration before its displaying. Specifying of enquiries (values), which will be displayed in form as a result, is possible by commands for output. A language ToPas limited variant is also used to define expressions used within the framework of program TopoL.

Information form heather

Information form heather contents symbolic name of form, type and group of object for which form is defined.

Heather has this figure:.

```
infoform form_name typobj object_type class
group_number ;
```

In heather three key words **infoform**, **typobj** and **class** can be found. It introduces single specific information form parameters.

Parameter **form_name** is the identifier defining the alias of the form. It imports only except self form processing as its identification.

Parameter **object_type** defines the type of object. which can be one of following B,P,L.

Parameter group number defines a object number, for which form is defined. Number is displayed in information about ac-

tive block or can be determined from primary database for example.

```
Example:
 infoform TOWNS typobj P class 9;
```

Declaration part of information form

Optional declaration part allows within the framework of form define constants, variables and symbolic names for used DB fields and graphic objects blocks . Single sections can be alternated and repeated. It is always introduced by key word. Declaration part finishes in the beginning of main form body.

Detailed enquiries vide paragraphs Constants declaration, Variables declaration and Symbolic DB fields names declaration and Symbolic blocks of graphic objects names declaration in reference guide Language ToPas.

Main body of information form description

Main body of form description consists of language ToPas commands sequence which are made during form processing before its displaying. By commands for output it is possible to specify enquiries (values) within the framework of form description which will be really displayed in form as a result.

At present most of standard commands of language Pascal are implemented within the framework of ToPas language. Simple command is value assignment and evocation of procedure for output command. Structured commands are compound command (begin..end construction), deciding command (if construction) and cycle commands (while, repeat and for construction).

Main body of form description is figured as compound command after which a stop symbol (.) must follow. Between identifiers begin and end random sequence of commands is mentioned. Single commands are separated by semi-colon (;).

More detailed enquiries vide paragraphs Assignment command, Procedures for output, Compound command, Deciding

command and Cycle commands in Language ToPas reference guide.

Example of application forms definition

For group 9 areas we first define application forms description file which name will be P0009.NDI and will be situated in working directory.

P0009.NDI	**Comment**
`Header = Apl. forms`	File heather
	Specify window caption
`;Information forms`	
`[DIN]`	Beginning of information form description
`Descr = Commercial data`	Form head.
`Tab = Business`	Description at form marker
`File = towns.din`	File name with information form definition. File is situated in working directory.
`Clipboard = Yes`	Form will be completed by button clipboard.
`; Form with raster`	
`[RAS]`	Beginning of form with raster description
`Descr = Photo of town`	Form head.
`Tab = Foto`	Description at marker of form
`Item_for_file = PHOTO`	In PHOTO field in internal database there is mentioned a name of displayed raster picture.

`Zoom = 50`	Picture will be reduced to a half (to 50 %).
`Item_for_sound = SONNET`	There is mentioned a name of sound file in field SONNET in internal database
`; text form`	
`[TXT]`	Beginning of form description with text document
`Descr = History of town`	Form head
`Tab = History`	Description at form marker.
`Print = Yes`	Form will be completed by button Print
`Item_for_file = HISTORY`	In HISTORY field in internal database there is mentioned name of displayed text document..

For group 9 areas competent information form description is defined in TOWNS.DIN file situated in working directory.

Example:
file TOWNS.DIN

```
infoform TOWNS typobj P class 9;
var
    TurnOver: list of LongInt;
    Month   : list of string;
    I,Suma: double;
begin
  writeln;
  writeln('Basic bussines information');
  writeln('--------------------------');
  writeln;
  writeln('Town name: ',@NAME)
  writeln;
  writeln('Land Size: ',%AREA/100,' km3');
  writeln;
```

```
  writeln('Zip Code: ',@PSC);
  writeln;
  TurnOver := @1.TurnOver;
  Month    := @1.Month;
  if Num(Obraty) < 1
  then writeln('no bussines in this town');
  else begin
    writeln('MONTH': 20,
            'TURN OVER IN MONTH': 20);
    writeln('-----': 20,'-------------': 20);
    writeln;
    Suma := 0;
    for I:=1 to Num(TurnOver) do
    begin
      writeln(Month[I]: 20, TurnOver[I]: 20);
      Suma := Suma + TurnOver[I];
    end;
    writeln;
    writeln('Total TurnOver  ', Suma);
  end;
end.
```