

Using TMS Parameter Controls

<http://www.tmssoftware.com/site/paramlabel.asp>

Introduction

The parameter identifier is specified through the controls href tag in the anchor formatting. This takes the format of : `param value` Multiple parameter identifiers can be used in the control but must be unique through the control.

The controls provide an easy interface to get and set parameter values with the `Parameter[identifier:string]:string` property.

Additional interfaces are available for the listbox based and treeview based controls, being :

`ItemParameter[idx:Integer; identifier:string]:string` ; This gets or sets the parameter value in a listbox item

`NodeParameter[Node:TTreeNode; identifier]:string` ; This gets or sets the parameter value in a treeview node

Example :

Select your tool : `Delphi` and OS : `Windows`

The parameter values can here be obtained through :

```
var
  os, tool:string;

  tool := control.Parameter['tool'];
  os := control.Parameter['os'];
```

The parameter values can be set through :

```
control.Parameter['tool'] := 'Delphi';
control.Parameter['os'] := 'Windows';
```

Several ways exist to let the user change the parameter value, for which 3 events are used : `OnParamClick`, `OnParamList`, `OnParamPopup` and further inplace editors can be specified.

By specifying in code

The anchor for identifying the parameter can have 2 additional attributes, being `CLASS` and `PROPS`. The optional `CLASS` attribute can specify which inplace editor has to be used for the parameter. This can currently be: `LIST`, `MENU`, `EDIT`, `SPIN`, `DATE`

- `LIST` : a listbox is used, the `PROPS` attribute specifies the listbox elements
- `MENU` : a popup menu is used, the `PROPS` attribute specifies the popup menu items
- `EDIT` : a simple inplace editor is used
- `SPIN` : an inplace spin editor is used
- `DATE` : an inplace date picker is used

Example :

Select your car here : `BMW`

or

Select the period from 1/1/2001 to 1/12/2001

In a ParamTreeView, the dates could be retrieved using:

```
MyStartDate := StrToDate (ParamTreeView.NodeParameter [TheNode, 'startdate']);
MyEndDate := StrToDate (ParamTreeView.NodeParameter [TheNode, 'enddate']);
```

By using events:

By using events, inplace listbox editors or popup menu contents can be specified.

OnParamClick :

This event returns the href and value of the parameter clicked. The value can be changed in code.

Example 1 : (this toggles the value between Delphi & C++Builder)

```
if (href = 'tool') and (value = 'Delphi') then
  value := 'C++Builder'
else
  value := 'Delphi';
```

Example 2 : (this asks the user for the value)

```
if (href = 'os') then
  InputQuery ('Operating system', 'Name', value);
```

OnParamList:

This event returns the href and value of the parameter clicked and allows setting of the values that will appear in the dropdown listbox. If the doPopup parameter is returned true, the dropdown list will appear. The values of the list can be set through the stringlist Values parameter.

Example :

```
doPopup := (href = 'tool');
values.Add ('Delphi');
values.Add ('C++Builder');
values.Add ('JBuilder');
```

OnParamPopup:

This event is similar to the OnParamList except that it controls the popup menu for parameter selection. This event returns the href and value of the parameter clicked and allows setting of the values that will appear in the popup menu. If the doPopup parameter is returned true, the popup menu will appear. The values of the popup menu can be set through the stringlist Values parameter.

Example :

```
doPopup := (href = 'tool') or (href = 'os');
if (href = 'tool') then
  begin
```

```
values.Add('Delphi');
values.Add('C++Builder');
values.Add('JBuilder');
end;

if (href = 'os') then
begin
values.Add('Windows');
values.Add('Linux');
values.Add('Mac OS/X');
end;
```