

## Component TFilteredEdit

### Files

Filter.pas	Basic functionality of component
FilterNode.pas	Auxiliary unit
wizardForm.pas, wizardForm.dfm	Component editor
FilteredEdit.pas	Component itself
FilterRTL.dpk	Component project
FilterREG.dpk	Editor project

### The purpose of component

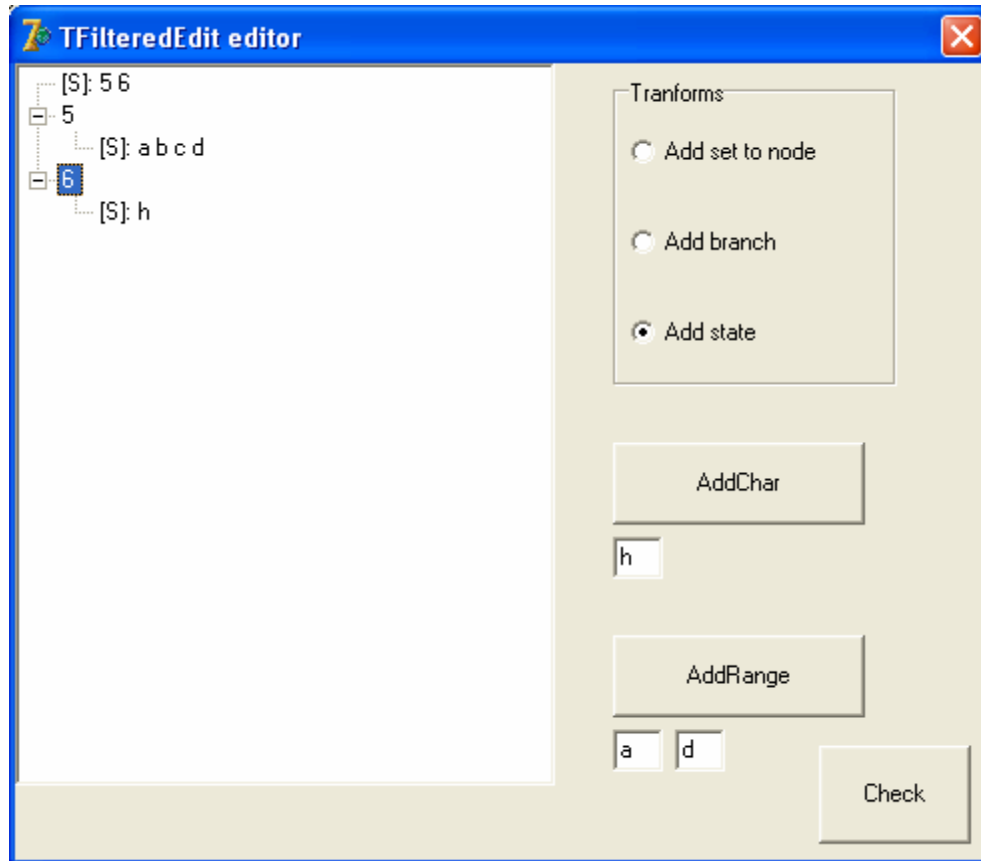
Component is intended for protected input of strings of any type and gives full control under input. **There is no any wildcards you just work in the spirit of Delphi!**

Component represents state-machine. The state diagram type is tree. In each state only certain set of defined characters are admitted for input. Depending on typed character current state is changed with other (with its own set) etc.

### State tree of component

Consists two kinds of vertices – state-vertices and branch-vertices. State-vertex consists a set of characters admissible for input in current state. Branch-vertex describes jump to the next state-vertex.

In the picture 1 the component's editor shown.



Picture 1- Component's editor

Any state-vertex has label of the next format:

[S]: char\_1 char\_2 .. char\_n,

where any char\_i,  $i=1..n$  is admissible for input in respective state.

Vertically below state-vertex its branch - vertices are arranged.

For example, in initial state admissible set is [5, 6]. Typing 5 results in jump to state with set [a,b,c,d], typing 6 passes control to state with set [h].

### State-verteces

State-vertex is terminal if and only if it has no branches. If this case achieved and any character from the vertex's set is typed, input is terminated (you can begin input pressing Backspace).

If state-vertex has at least on branch then the following required:

- 1) Union of all branches of the vertex must be equal to set of the vertex
- 2) Intersection of all branches of the vertex must be equal to empty set

Requirement 2) is accomplished automatically thanks to editor's control. User must look for requirement 1 (after editing the component's tree use button Check).

## Branch-verteces

Any branch-vertex can be both terminal and nonterminal. Any branch -vertex can have single spring – state-vertex which accepts control from the branch.

*Notice: in tree branch-vertex can have many springs, but only the first is considered to be the state-vertex. All others are branches of the state-vertex*

State-vertices cannot have springs.

## Editing component

Is realized by buttons AddChar/AddRange and radiogroup Transforms. The transform is applied to the currently chosen vertex if it's allowably. After editing it's strongly recommended to check the tree by Check button.

## Bugs

Component doesn't work in design-time. Test it only after compilation.

*Notice: although it's implemented in design/runtime package, few owned elements (TEdit) doesn't work properly in design-time.*

## Example

```
if FilteredEdit1.Terminated then  
    ShowMessage(FilteredEdit1.Text);
```