TPxValcheckTable and TPxValcheckQuery

These non-visual Delphi components are descendants of TTable/TQuery. They add the ability to read the field valchecks of Paradox Tables. Though you access the valcheck properties as arrays, they are only retrieved on request so not much extra memory is used. If you need to loop through all fields valchecks, do it field by field in the outer loop and property by property in the inner loop to optimize performance (Each time a valcheck is requested, all valchecks for the given field are retrieved. When you then request another valcheck for the same field, it doesn't have to be retrieved again but is read from a one-field valcheck record).

It doesn't make sense to use these components with non-Paradox Tables, they work just like a standard TTable/TQuery then.

If you are interested in Paradox valchecks, you might also want to have a look at my file PICEDITS.ZIP in lib 5 of the Delphi CIS forum. Contains TDBPxPicEdit and TPxPicEdit. Both support pictures, min and max validation, default and required status. The bound control also retrieves the current valchecks of it's DataField from the Paradox Query.

This is a beta, and it's freeware. Sorry, no source at the moment.

Suggestions and bug reports --> Reinhard Kalinke, 100417,3504@compuserve.com

Installation

Unzip the *.DCU and *.DCR files to a directory containing your add-on components. Choose 'Add components', then browse for PXVALTBL.DCU to install TPxValcheckTable and for PXVALQRY.DCU to install TPxValcheckQuery. (Please refer to your Delphi manual on how to add components to the VCL.)

New Properties of TPxValcheckTable and TPxValcheckQuery

(All properties are runtime/readonly. 'Index' refers to the field index as in QueryX.Fields[Index])

property FieldHasValchecks[Index: byte]: boolean
False, if no valcheck has been found for the field. Always False, if the Table/Query is not open or the table is not a Paradox table.

property FieldRequired[Index: byte]: boolean
This value should be no different from the TField's 'Required' property, it's only in here for ease of access.

property FieldPicture[Index: byte]: String
Returns the Paradox picture defined for the given field.

property FieldLookUpType[Index: byte]: LKUPType
Returns the lookup-type. You will have to add either DBITypes to your uses-clause or the following declaration to your unit to evaluate this value. LKUPType is declared as

```
 LKUPType = (                    { Paradox Lookup type }
   lkupNONE,                     { Has no lookup }
   lkupPRIVATE,                  { Just Current Field + Private }
   lkupALLCORRESP,               { All Corresponding + No Help }
   lkupHELP,                     { Just Current Fld + Help and Fill }
   lkupALLCORRESPHELP            { All Corresponging + Help }
```

);

property FieldLookUpQuery[Index: byte]: String
Returns the name of the lookup Query for the given field.

All of the following properties return their values as strings. The date field keyword 'Today' is reported as Min-, Max- resp. DefaultValue '%TODAY%'. The DefaultValue for a ftBoolean field is returned as '0' or '1'.

property FieldMinValue[Index: byte]: String

property FieldMaxValue[Index: byte]: String

property FieldDefaultValue[Index: byte]: String

Everthing else is standard TTable/TQuery behaviour (or should be...).

History:

v0.9 initial release

v0.91 Lookup table info added.

...and here's the inevitable:

DISCLAIMER

The author cannot and does not warrant that any functions contained in the software will meet your requirements, or that its operations will be error free. The entire risk as to the software performance or quality, or both, is solely with the user and not the author. You assume responsibility for the selection of the component to achieve your intended results, and for the installation, use, and results obtained from the software.

The author makes no warranty, either implied or expressed, including without limitation any warranty with respect to this software documented here, its quality, performance, or fitness for a particular purpose. In no event shall the author be liable to you for damages, whether direct or indirect, incidental, special, or consequential arising out the use of or any defect in the software, even if the author has been advised of the possibility of such damages, or for any claim by any other party.

All other warranties of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, are expressly excluded.

Delphi and Paradox are trademarks of Borland International.