TValidator Component for Borland Delphi

TValidator is a non visual component for Delphi. It simplifies a process of implementation of validation for shareware programs. Really a ***must*** component for shareware developers.

How to Use TValidator

Sample application vt.dpr is included in this package.

Published Properties

TValidator has two published properties: **RandSeed** and **IniFileName**.
RandSeed is the key property. Based on this LongInt number plus user name TValidator calculates 12 digit random number, which is a validation code. RandSeed has to be unique for your each application (well, nothing horrible happens if it is the same for all your applications, but it is not for your - shareware developers - benefit to have the same validation number for all your shareware programs).
IniFileName is a name of .INI file. TValidator uses this file to store the name of user and the registration number. Each time the program starts up, you can check these values and compare with calculated values (by calling IsValidated method of TValidator), check out if the program has been validated and perform appropriate actions, if necessary. If you *do not* initialize IniFileName property either in the Object Inspector or at runtime, default will be a file in the startup directory of application (please note, not in windows directory) with the same name as your application .EXE file name and extension .INI. If you *do* initialize IniFileName *without extension* .INI will be added. If you omit the *path* during initialization, .INI file will be placed in windows directory.

Published Events

There are two published events. OnVldOk and OnVldFailed.
OnVldOk determines what happens if after the validation process appears that Registration number entered by user is correct. You can use this event to praise user or thank him for registration or whatever you want.
OnVldFailed is opposite to OnVldOk method i.e. this event happens when user entered the *invalid* registration number. You can use this events to punish the user the way you want.
There is no default handling for this two events, so if you leave them not assigned, simply nothing happens.

Methods

There are three significant public methods: IsValidated, Validate, and Devalidate.
IsValidated is a function which returns boolean value. It is *true* if the user did validate the program and is *false* if he (or she) did not. Best place to run this method is OnCreate event of your main form.
Validate is a procedure which displays dialog box and asks the user to enter his/her name and validation number, latter supposed to be provided by you, shareware developer. If after clicking OK button in the above mentioned dialog everything is fine (i.e. if result of

calculation of the registration number, based on the name entered by user coincides with one provided by you), .INI file gets updated and next time call to IsValidated method will return *true*.
Devalidate removes [Validation] section from .INI file and devalidates the program.

Package

This package includes following files:

valid.dcu	delphi unit containing *TValidator* component
valid.dcr	dcr resource file with bitmap which will represent *TValidator* on your component palette
validdlg.dcu	code for the *TValidator* dialog box
validdlg.dfm	form resource file for the *TValidator* dialog box
valmisc.dcu	supplementary unit
gen.dpr	project file for the *Registration Number Generator*
genmain.pas	main file for the *Registration Number Generator*
genmain.dfm	form resourse file for the *Registration Number Generator*
vt.dpr	project file for the *Validator Test*
vtmain.pas	main file for the *Validator Test*
vtmain.dfm	form resource file for the *Validator Test*
readme.wri	this file
regform.wri	registration form
valid.int	interface part of valid.pas unit
valmisc.int	interface part of valmisc.pas unit
file_id.diz	brief description of this package

Please redistribute this package exactly as you received it, without adding or removing anything.

registered users will receive the complete source code (i.e. all above files, just valid.int will be replaced by valid.pas and valmisc.int - be valmisc.pas).

Installation

Unzip valid.zip file in some temporary directory and copy following files:

valid.dcu
valid.dcr
validdlg.dcu
validdlg.dfm
valmisc.dcu

to delphi\lib directory, and following files:

gen.dpr
genmain.pas
genmain.dfm
vt.dpr

vtmain.pas
vtmain.dfm
readme.wri
regform.wri
valid.int
valmisc.int
file_id.diz

to the directory of your choice.

Then run delphi and install valid.dcu into your library (don't try to install validdlg and valmisc). Validator will appear on the samples page of component palette. After this you can load and compile gen.dpr and vt.dpr. Actually, gen.exe does not use TValidator component, it just uses routines from valmisc unit, but valmisc should already be in \lib directory to compile gen.exe.

Registration

TValidator is shareware. It's registration is US$10.95. If you are Compuserve user, soon you will be able to GO SWREG and select registration number. I'll send you original source code via compuserve mail. Other users should complete the registration form (see regform.wri) and mail it along with payment to:

Artchil Gogava
200 Balliol St. #1405
Toronto ON M4S 1C6
Canada

If you wish to obtain complete uuencoded source code via Email, include your Email address, otherwise add US$2.00 if you live in USA or Canada or US$4.00 if you live in the other region of World and you will receive source code on the disk.

Please make checks payable to Artchil Gogava.

compuserve: 75231,330
internet: 75231.330@compuserve.com