RT Registration Control 1.20

R&T Software

Features of RT Registration Control

The RTRegCtl is a library unit (DCU for Delphi, otherwise DLL) containing functions to protect shareware and not only shareware programs against unauthorized use which lets authorize them with a unique registration key. It has the possibility to save the encrypted information about users somewhere in the system and a count and/or date lock in order to make an evaluation copy. Thanks to its registration possibility a shareware program can be marketed fully electronically through computer networks like CompuServe, Internet and various BBSs, because the same shareware version can be turned into a registered version. There is no need for updates to be sent, too. Any update of a given product found on any network can be registered with the same registration key. Another highlight is a support for leased/rented software through linking the registration key with a future date, i.e. the next rate payment time.

From version 1.20 on youll be able to use this component to protect your own Delphi VCL components. Please read licence conditions before you do it.

How it works?

1. After installing the shareware product the user will be asked for a name and if he/she wants to register this product.

2. If the users answers with yes he will be asked to mail the name of the program, his name and any information that is needed to the author. If the registered software should be prohibited to be moved to another workstation after registering a unique PC workstation key can be generated (see RTGetPCKey) , which also should be mailed to the author.

3. The user mails the information asked for to the shareware author.

4. In the meanwhile the evaluation process can be started for a specified number of calls/days.

5. RTGENKEY utility (will be delivered upon registering RT Registration Control) generates a registration key dependent on a user name, product code, PC key generated on the users workstation (if applicable) and a date till the product can be used (if applicable).

6. The generated registration key is mailed to the user. A special key file for a file based registration process can also be generated.

7. The user enters obtained registration key into the dialog supplied by the author or copies the key file into the program directory.

8. The software is registered and all are happy!

An sample registering scheme can be viewed in the demo programs.

Benefits:

⇒ Delivering one copy of the program, which can be remotely registered through a generated registration key. The software itself along with any updates can also be put in any computer network, where potential customer can find it, so that there would be no need for delivering anything else but the registration key upon receiving money. The registration key can be generated uniquely per user identification key, target product key, a workstation identification code and/or next payment day for a leased/rented software.

⇒ file based registration, so that the shareware author doesnt need to supply any dialogs, where the registration information should have been entered.

⇒ Ability to set an evaluation period for a certain number of calls/days without writing a special code or delivering a special version of a product.

⇒ Ability to save any user information, such as user name, company name or serial number.
⇒ Ability to generate a date dependent registration key for leased/rented software, which can be at any time remotely prolonged.
⇒ Delphi direct linkable DCU file along with a VCL Component to set all the parameters visually
⇒ A DLL to be used with any language able to call it, such as C/C++, PowerBuilder, SQLWindows or Visual Basic.
⇒ Sample projects in Delphi, C and VB.

The shareware version is fully functional but shows a message box every time a piece of information is about to be saved.

There are three version of the RT Registration Control:

1. Delphi DCU version.

   includes free Delphi VCL component in source code and examples using function call and VCL approach. Allows for better integration of the code, because directly links into the EXE file, no DLLs necessary.

2. Language independent DLL version

   includes Visual-C++ and VB examples.

3. Both the above versions together

   for Delphi users using also other programming languages.

General Considerations

Software lock versus hardware lock
Generally I think it is impossible to generate really unique software keys. Somehow you have to build and store a number or string using the information stored on the PC. In a large company where several PCs could have been bought at the same time from the same vendor chances are that they are really same, so the generated PC key will not be unique. In this case buying software once lets use it on all same PCs. The other problem is that the place where we store our registration information can be found by some smart users and used to copy the software for friends.

The hardware dongle would always be unique and cannot be put in two places at the same time.

The other way round if a customer decides to change her/his PC and our software was able to generate a very unique key, then he will not be able to do that without receiving another registration key! But how would you know if he really changes hardware or perhaps want to make another copy for a friend. Again a hardlock works perfect in this case. He takes it, plugs into her/his new PC and its ready.

So actually you cannot beat a good hardware lock with software. There is only one positive side - price.

So its up to you, but be warned!

Dialog based registration versus file based registration
Basically the shareware software can be registered in one of two following procedures. After the customer sends necessary information to the shareware author, a register key will be generated by a RTGENKEY utility. This key can be mailed or faxed to the customer. Now he has to have a place to enter a obtained registration key, so the shareware author must supply a dialog in his product to accomplish this task. It is perhaps not difficult but it must be done.

The other possibility is to generate a key file, which, after being copied into the program directory or any other place where the shareware program can look into, can be used to register the software automatically. This file will be erased after registering. But it must be delivered by e-mail or disk, it cannot be faxed!

Data Security
As you may think, the registration data has to be saved somewhere. Surely you will ask how secure is this place. First, there is no 100% security. If there is a place on your PC, where the data can be saved, there will be someone that will find it. It is just a question of how easy it is. It if requires significant amount of time to look after the evaluation or registration data, then it will be probably seldom done. If you use one of the RTSet...File functions to put the data into some INI or DLL file, name it something like KRNL286.EXE and give a date of 8.10.92, then chances are, that nobody looks into it. And the file age will be preserved by RtReg writing functions.

Secondary Copy
Through the use of RTSafe... functions it is possible to set a secondary saving place, so that if the primary data will be found and destroyed or manipulated, there is another copy ready for use. Both copies must return the same results if the registration or evaluation are to be allowed.

Function Reference

Registration functions:
RTIsReg
RTDelReg
RTCheckReg
RTRegister
RTRegisterByFile
RTSetPCLock
RTGetPCKey

Count Lock functions:
RTIsCountLock
RTSetCountLock
RTDelCountLock
RTCheckCountLock

Date Lock functions:
RTIsDateLock
RTSetDateLock
RTDelDateLock
RTCheckDateLock

Information functions:
RTSetUserInfo
RTGetUserInfo
RTGetUserKey
RTDelUserInfo

Data Target functions:
RTGetSaveKind
RTSetSaveKind
RTGetSaveFile
RTSetSaveFile
RTGetSafeKind
RTSetSafeKind
RTGetSafeFile
RTSetSafeFile

Miscellanous function:
RTGetVersion
RTGetGenKind
RTSetGenKind
RTSetEncodeKey

RTIsReg

Delphi/Pascal syntax
function RTIsReg(const ProductKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTIsReg(LPCSTR ProductKey);

VB syntax
Declare Function RTIsReg Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer

This function checks if the correct registration information for a given product has already been saved.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
True if a registration information has been saved.

RTDelReg

Delphi/Pascal syntax

```
function RTDelReg(const ProductKey: string): Boolean;
```

C/C++ syntax

```
BOOL far pascal RTDelReg(LPCSTR ProductKey);
```

VB syntax

```
Declare Function RTDelReg Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer
```

This function tries to delete all registration information that has been saved for a given program.

Parameters

ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns

True if the information could be deleted.

RTCheckReg

Delphi/Pascal syntax
function RTCheckReg(const ProductKey: string): longint;

C/C++ syntax
long far pascal RTCheckReg(LPCSTR ProductKey);

VB syntax
Declare Function RTCheckReg Lib "RTREGW.DLL" (ByVal ProductKey$) As Long

The check of the current projects registration will be executed using information such as user key, product key, current date and possibly PC workstation identification code.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
For a date dependent registration values greater or equal null confirm a valid registration, for a standard unlimited registration a null value is a positive one. In all cases the negative return value means lack of a proper registration information.

RTRegister

Delphi/Pascal syntax
function RTRegister(const ProductKey,UserKey,RegKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTRegister(LPCSTR ProductKey,LPCSTR UserKey, LPCSTR RegKey);

VB syntax
Declare Function RTRegister Lib "RTREGW.DLL" (ByVal ProductKey$,ByVal UserKey$, ByVal RegKey$) As Integer

This function tries to register software product with a proper registration key identified by ProductKey for a user identified by UserKey.

Parameters

ProductKey	identification code for a product to be registered
UserKey	identification code for a registered user
RegKey	the generated registration key

Returns
True if a registration succeeded.

RTRegisterByFile

Delphi/Pascal syntax

```
function RTRegisterByFile(const ProductKey, FileName: string): Boolean;
```

C/C++ syntax

```
BOOL far pascal RTRegisterByFile(LPCSTR ProductKey, LPCSTR FileName);
```

VB syntax

```
Declare Function RTRegisterByFile Lib "RTREGW.DLL" (ByVal ProductKey$,ByVal FileName$) As Integer
```

This function tries to register software product with a proper registration key from a key file named by FileName. The key file will be erased after the first try, so make sure that you have a backup copy!

Parameters

ProductKey	identification code for a product to be registered
FileName	the name of a key file with the registration information.

Returns

True if a registration could be accomplished.

RTSetPCLock

Delphi/Pascal syntax
procedure RTSetPCLock(Value: Boolean);

C/C++ syntax
void far pascal RTSetPCLock(BOOL Value);

VB syntax
Declare Sub RTSetPCLock Lib "RTREGW.DLL" (ByVal Value As Integer)

This function toggles the checking of the PC workstation identification code. If set True the PC workstation will be checked and an identification code will be generated. Using of this code lets the RTGENKEY program generate a registration key unique not only to a combination of a user identification code and a product key, but also to a PC workstation where the product should be used.
It is a kind of copy protection. Contrary to hardware dongles however this identification code is not 100% safe. Most often it should do unless the user has a large number of identical PC equipment.

Parameters
A new value of the PC identification check toggle.

Returns
There is no return value.

RTGetPCKey

Delphi/Pascal syntax
function RTGetPCKey: string;

C/C++ syntax
BOOL far pascal RTGetPCKey(LPCSTR PCKey);

VB syntax
Declare Function RTGetPCKey Lib "RTREGW.DLL" (ByVal PCKey$) As Integer

This function checks the hardware equipment installed of a PC and tries to generate a unique identification code, that could be used as kind of a copy protection scheme.

Parameters
DLL version passes an address of a buffer where the PC workstation code should be saved.

Returns
Delphi/Pascal version returns the actual PC workstation identification key as a pascal string, DLL version has no return value.

RTIsCountLock

Delphi/Pascal syntax
function RTIsCountLock(const ProductKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTIsCountLock(LPCSTR ProductKey);

VB syntax
Declare Function RTIsCountLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer

RTIsCountLock check if a Count Lock has been set.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
True if a Count Lock has been detected.

RTSetCountLock

Delphi/Pascal syntax
function RTSetCountLock(const ProductKey: string; Value: longint): Boolean;

C/C++ syntax
BOOL far pascal RTSetCountLock(LPCSTR ProductKey, long Value);

VB syntax
Declare Function RTSetCountLock Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Value As Long) As Integer

Tries to set an evaluation Count Lock for a specified number of calls.

Parameters

ProductKey	identification code for a product to be registered
Value	a number of calls the product can be called during the evaluation period

Returns
True if a Count Lock could be set.

RTDelCountLock

Delphi/Pascal syntax
function RTDelCountLock(const ProductKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTDelCountLock(LPCSTR ProductKey);

VB syntax
Declare Function RtDelCountLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer

Tries to delete a previously set Count Lock.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
True if the Count Lock could be deleted.

RTCheckCountLock

Delphi/Pascal syntax
function RTCheckCountLock(const ProductKey: string): longint;

C/C++ syntax
long far pascal RTCheckCountLock(LPCSTR ProductKey);

VB syntax
Declare Function RTCheckCountLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Long

Checks if a previously set Count Lock is over.

ATTENTION
Every calls to this function decrements a Count Lock by one.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
Number of calls left before decrementing the Count Lock. Negative values mean end of the evaluation period.

RTIsDateLock

Delphi/Pascal syntax
function RTIsDateLock(const ProductKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTIsDateLock(LPCSTR ProductKey);

VB syntax
Declare Function RTIsDateLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer

RTIsDateLock check if a Date Lock has been set.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
True if a Date Lock has been detected.

RTSetDateLock

Delphi/Pascal syntax

```
function RTSetDateLock(const ProductKey: string; Value: longint): Boolean;
```

C/C++ syntax

```
BOOL far pascal RTSetDateLock(LPCSTR ProductKey, long Value);
```

VB syntax

```
Declare Function RTSetDateLock Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Value As Long) As Integer
```

Tries to set an evaluation Date Lock for a specified number of days.

Parameters

ProductKey	identification code for a product to be registered
Value	a number of days the product can be called during the evaluation period

Returns

True if a Date Lock could be set.

RTDelDateLock

Delphi/Pascal syntax
function RTDelDateLock(const ProductKey: string): Boolean;

C/C++ syntax
BOOL far pascal RTDelDateLock(LPCSTR ProductKey);

VB syntax
Declare Function RtDelDateLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Integer

Tries to delete a previously set Date Lock.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
True if the Date Lock could be deleted.

RTCheckDateLock

Delphi/Pascal syntax
function RTCheckDateLock(const ProductKey: string): longint;

C/C++ syntax
long far pascal RTCheckDateLock(LPCSTR ProductKey);

VB syntax
Declare Function RTCheckDateLock Lib "RTREGW.DLL" (ByVal ProductKey$) As Long

Checks if a previously set Date Lock is over. It compares previously set evaluation end date with a current date.

Parameters
ProductKey is a symbolic or real name of a product (program) that is to be registered. It can be maximum 50 characters long.

Returns
Number of days left. Negative values mean end of the evaluation period.

RTSetUserInfo

Delphi/Pascal syntax
function RTSetUserInfo(const ProductKey,Info,Value: string): Boolean;

C/C++ syntax
BOOL far pascal RTSetUserInfo(LPCSTR ProductKey, LPCSTR Info, LPCSTR Value);

VB syntax
Declare Function RTSetUserInfo Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Info$, ByVal Value$) As Integer

Tries to save any user information under the specified ProductKey and information identification code (variable name).

Parameters

ProductKey	identification code for a product to be registered
Info	identification code for a user information to be saved
Value	address of a memory buffer with the user information

Returns
True if the information could be saved.

RTGetUserInfo

Delphi/Pascal syntax
function RTGetUserInfo(const ProductKey,Info: string; var Value: string): Boolean;

C/C++ syntax
BOOL far pascal RTGetUserInfo(LPCSTR ProductKey, LPCSTR Info, LPCSTR Value);

VB syntax
Declare Function RTGetUserInfo Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Info$, ByVal Value$) As Integer

Retrieves a user information.

Parameters

ProductKey	identification code for a product to be registered
Info	identification code for a user information to be retrieved
Value	address of a memory buffer where the information is to be copied

Returns
True if an information has been found and could be read in.

RTGetUserKey

Delphi/Pascal syntax
function RTGetUserKey(const ProductKey: string; var Value :string): Boolean;

C/C++ syntax
BOOL far pascal RTGetUserKey(LPCSTR ProductKey, LPCSTR Value);

VB syntax
Declare Function RTGetUserKey Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Value$) As Integer

Retrieves a UserKey under which current product has been registered.

Parameters

ProductKey	identification code for a product to be registered
Value	address of a memory buffer where the UserKey (e.g. user name) is to be copied

Returns
True if a UserKey could be retrieved.

RTDelUserInfo

Delphi/Pascal syntax
function RTDelUserInfo(const ProductKey,Info: string): Boolean;

C/C++ syntax
BOOL far pascal RTDelUserInfo(LPCSTR ProductKey, LPCSTR Info);

VB syntax
Declare Function RTDelUserInfo Lib "RTREGW.DLL" (ByVal ProductKey$, ByVal Info$) As Integer

Tries to delete a specified user variable.

Parameters

ProductKey	identification code for a product to be registered
Info	identification code for a user information to be deleted

Returns
True if the variable could be deleted.

RTGetSaveKind

Delphi/Pascal syntax
function RTGetSaveKind: TSaveKind;

C/C++ syntax
int far pascal RTGetSaveKind(void);

VB syntax
Declare Function RTGetSaveKind Lib "RTREGW.DLL" () As Integer

This function returns the saving target.

Returns
Return value can have one of the following values:

1	default	(skDef or SAVE_DEF)
2	standard INI file	(skIni or SAVE_INI)
3	special DLL	(skDll or SAVE_DLL)

RTSetSaveKind

Delphi/Pascal syntax
function RTSetSaveKind(Value: TSaveKind): Boolean;

C/C++ syntax
BOOL far pascal RTSetSaveKind(int Value);

VB syntax
Declare Function RTSetSaveKind Lib "RTREGW.DLL" (ByVal Value As Integer) As Integer

Sets the saving target. For the list of the allowed values look RTGetSaveKind;

Parameters
New value of the saving target.

Returns
True is succesful.

RTGetSaveFile

Delphi/Pascal syntax
function RTGetSaveFile: string;

C/C++ syntax
void far pascal RTGetSaveFile(LPSTR File);

VB syntax
Declare Sub RTGetSaveFile Lib "RTREGW.DLL" (ByVal File$)

Returns current name of the saving target. Doesnt apply by the default saving option.

Parameters
File/File$ is a buffer to hold the returned file name.

RTSetSaveFile

Delphi/Pascal syntax
function RTSetSaveFile(Name: string): Boolean;

C/C++ syntax
BOOL far pascal RTSetSaveFile(LPCSTR Name);

VB syntax
Declare Function RTSetSaveFile Lib "RTREGW.DLL" (ByVal Name$) As Integer

Allows for the setting of the filename, where all the data should go. Doesnt apply to default saving option. For the INI file or DLL it can take any valid DOS filename. INI file doesnt have to exist before first access, but if the file does exist, its file age will be by RtReg preserved.

Parameters
Name: the name of the file.

Returns
True if succesful.

RTGetSafeKind

Delphi/Pascal syntax
function RTGetSafeKind: TSaveKind;

C/C++ syntax
int far pascal RTGetSafeKind(void);

VB syntax
Declare Function RTGetSafeKind Lib "RTREGW.DLL" () As Integer

This function returns the kind of the security copy of the registration information. It can take one of the following values:

0	no security copy	(skNone or SAVE_NONE)
1	default	(skDef or SAVE_DEF)
2	standard INI file	(skIni or SAVE_INI)
3	special DLL	(skDll or SAVE_DLL)

It must be different from the kind of the primary saving target or, in the case of SAVE_INI/SAVE_DLL at least the filenames must differ.

RTSetSafeKind

Delphi/Pascal syntax
function RTSetSafeKind(Value: TSaveKind): Boolean;

C/C++ syntax
BOOL far pascal RTSetSafeKind(int Value);

VB syntax
Declare Function RTSetSafeKind Lib "RTREGW.DLL" (ByVal Value As Integer) As Integer

Sets the saving target of the security copy. For the list of the allowed values see RTGetSafeKind;

Parameters
Value: a new secondary copy target.

Returns
True if succesful.

RTGetSafeFile

Delphi/Pascal syntax
function RTGetSafeFile: string;

C/C++ syntax
void far pascal RTGetSafeFile(LPSTR File);

VB syntax
Declare Sub RTGetSafeFile Lib "RTREGW.DLL" (ByVal File$)

Returns current name of the security copys saving target. Doesnt apply by the default saving option.

Parameters
File/File$ is a buffer to hold the returned value.

RTSetSafeFile

Delphi/Pascal syntax
function RTSetSafeFile(Name: string): Boolean;

C/C++ syntax
BOOL far pascal RTSetSafeFile(LPCSTR Name);

VB syntax
Declare Function RTSetSafeFile Lib "RTREGW.DLL" (ByVal Name$) As Integer

Allows for the setting of the filename, where the security copy of the data should go.

Parameters

Name:	a new file name of the secondary copy.

Returns
True if succesful.

RTGetVersion

Delphi/Pascal syntax
function RTGetVersion: longint;

C/C++ syntax
long far pascal RTGetVersion();

VB syntax
Declare Function RTGetVersion Lib "RTREGW.DLL" () As Long

Returns
The current RtRegs version number. The integer value is in the high byte, and fractional part in the lower byte of the result.

RTGetGenKind

Delphi/Pascal syntax
function RTGetGenKind: TGenKind;

C/C++ syntax
int far pascal RTGetGenKind(void);

VB syntax
Declare Function RTGetGenKind Lib "RTREGW.DLL" () As Integer

This function returns the current method of generating the registration keys. It can take one of the following values:

0	standard, compatible with previous versions	(gkSingle or GEN_SINGLE)
1	new, more unique	(gkDouble or GEN_DOUBLE)

Attention!
Both the runtime environment and RTGENKEY utility must use the same method. The latter can be set through RTGENKEY.INI:

```
[System]
GenKind=<x>
```

where x=0 stands for gkSingle method, and x=1 equals gkDouble.

RTSetGenKind

Delphi/Pascal syntax
function RTSetGenKind(Value: TGenKind): Boolean;

C/C++ syntax
BOOL far pascal RTSetGenKind(int Value);

VB syntax
Declare Function RTSetGenKind Lib "RTREGW.DLL" (ByVal Value As Integer) As Integer

Sets the method for generating registration keys. For the list of the allowed values see RTGetGenKind;

Parameters
Value: a new generating method.

Returns
True if succesful.

RTSetEncodeKey

Delphi/Pascal syntax
procedure RTSetEncodeKey(Value: string);

C/C++ syntax
void far pascal RTEncodeKey(LPCSTR Value);

VB syntax
Declare Sub RTSetEncodeKey Lib "RTREGW.DLL" (ByVal Value$)

Sets additional scrambling string. Since it can be set only at runtime and is application-specific, it offers extra security for the encrypted data.

Examples

There are four sample projects in four separate directories. All of them do actually the same, but in the language specific manner.

C\hello.mak	Visual C++/MFC example using DLL version of RT Registration Control
DELPHI\project1.dpr	Delphi project using a DCU linkable module
VB\project1.mak	Visual Basic 3.0 example project also with DLL usage
VCL\project1.dpr	Delphi demo using a VCL approach

Delphi TRtRegControl Component

Properties Methods Events

TRtRegControl encapsulates most if not all of the possibilities of the package as a VCL non-visual component. Its event model accomplishes the most basic registration strategy.

The two key properties are ProductKey, that should uniquely identify the product being developed and PCKey, which decides if a software protection lock should be used. KeyFile property if not empty contains the name of the key file for a file based registration process (this not the place there the fact, that the program has been registered will be saved). The Registered property can used to check if a product has been properly registered, by a leased/rented product RegDaysLeft returns a number of days till the end of the registration.

The SaveFileName,SafeFileName,SaveKind,SafeKind properties allow the choose where the registration data is to be saved. EncodeKey lets additional scramble sctring to be used to encode saved data.

Order of the events:

OnEndCountLock
OnEndDateLock
OnRegister
OnStartCountLock
OnStartDateLock

OnEndCountLock and OnEndDateLock are called after the evaluation period terminates. The OnRegister event will be fired if the product be recognized as not registered. In the same case OnStartCountLock will be raised, if a CountLock hasnt been set up already, the same with OnStartDateLock. Generally only one of the two would be used. The amount of time/calls left can be read with DaysLeft and CallsLeft properties. To register software use Register function method with the registration key delivered from the software author and a UserKey, both supplied by the user in the runtime.

Available since the version 1.20 GenKind property lets you decide about the uniqueness of your generated registration key.

CallsLeft

Declaration

property CallsLeft: longint;

Description

This property encapsulates calls to RTCheckCountLock (on reading) and RTSetCountLock (on writing). It returns a number of calls left within an evaluation period.

DaysLeft

Declaration

property DaysLeft: longint;

Description

This property encapsulates calls to RTCheckDateLock (on reading) and RTSetDateLock (on writing). It returns a number of days left within an evaluation period.

Registered

Declaration

property Registered: Boolean;

Description

True if a product specified by ProductKey has been registered. See also RTCheckReg.

RegDaysLeft

Declaration

property RegDaysLeft: longint;

Description

Returns a number of days left before the next paid registration in case of leased/rented software. See also RTCheckReg.

KeyFile

Declaration

property KeyFile: string;

Description

The name of the key file can be saved, in order to do a file based registration.

PCKey

Declaration

property PCKey: Boolean;

Description

Write only property. If True then the software lock key will be used by the registration. See also RTSetPCLock.

ProductKey

Declaration

property ProductKey: string;

Description

Should uniquely identify a product being registered.

UserInfo

Declaration

property UserInfo[const Info: string]: string;

Description

Lets save and retrieve any information identified by parameter Info, i.e. serial number. Encapsulates RTSetUserInfo and RTGetUserInfo calls.

Version

Declaration

property Version: string;

Description

Runtime and Read only. Returns the current version number of RtReg. See also RTGetVersion.

EncodeKey

Declaration

property EncodeKey: string;

Description

Allows for an extra string to additional scramble the encoded information. See also RTSetEncodeKey.

SaveFileName

See also

Declaration

property SaveFileName: string;

Description

This property contains the filename of the primary information target, if it was set to be saved in INI or DLL file.

SafeFileName

See also

Declaration

property SafeFileName: string;

Description

This property contains the filename of the secondary information target, if it was set to be saved in INI or DLL file.

SaveKind

See also

Declaration

property SaveKind: TSaveKind;

Description

This property defines the place to save the primary information:

SafeKind

See also

Declaration

property SafeKind: TSafeKind;

Description

This property defines the place to save the secondary information:

GenKind

Declaration

property GenKind: TGenKind;

Description

This property defines the method for generating registration keys.

DelCountLock

Declaration

function DelCountLock: Boolean;

Description

Deletes CountLock information thus ending any CountLock based evaluation period for a given product.
See also RTDelCountLock.

DelDateLock

Declaration

function DelDateLock: Boolean;

Description

Deletes DateLock information thus ending any DateLock based evaluation period for a given product. See also RTDelDateLock.

DelRegistration

Declaration

function DelRegistration: Boolean;

Description

Removes Registration information. See also RTDelReg.

DelUserInfo

Declaration

function DelUserInfo(const Info: string): Boolean;

Description

Removes user information identified by the parameter Info. See also RTDelUserInfo.

IsCountLock

Declaration

function IsCountLock: Boolean;

Description

Checks if a CountLock has been installed. See also RTIsCountLock.

IsDateLock

Declaration

function IsDateLock: Boolean;

Description

Checks if a DateLock has been installed. See also RTIsDateLock.

IsRegistration

Declaration

function IsRegistration: Boolean;

Description

Checks if a registration information can be found irrespective of it being right or wrong. See also RTIsReg.

Register

Declaration

function Register(const UserKey,RegKey: string): Boolean;

Description

This method tries to register given software product with a proper registration key RegKey for a user identified by UserKey. See also RTRegister.

RegisterByFile

Declaration

function RegisterByFile: Boolean;

Description

This method tries to register given software product with a key file named in the property KeyFile. See also RTRegisterByFile.

SetCountLock

Declaration

function SetCountLock(const Value: longint): Boolean;

Description

Same as writing CallsLeft property, sets the evaluation period to Value calls. See also RTSetCountLock.

SetDateLock

Declaration

function SetDateLock(const Value: longint): Boolean;

Description

Same as writing DaysLeft property, sets the evaluation period to Value days. See also RTSetDateLock.

OnStartCountLock

Declaration

property OnStartCountLock: TNotifyEvent**;**

Description

This event will be fired if a programm is recognized as not registered and no CountLock has been found.

OnStartDateLock

Declaration

property OnStartDateLock: TNotifyEvent**;**

Description

This event will be fired if a program is recognized as not registered and no DateLock has been found.

OnRegister

Declaration

property OnRegister: TNotifyEvent**;**

Description

This event will be called if a product hasnt been registered, or a registration period has expired.

OnEndCountLock

Declaration

property OnEndCountLock: TNotifyEvent**;**

Description

This event will be fired if an evaluation period base on CountLock has expired.

OnEndDateLock

Declaration

property OnEndDateLock: TNotifyEvent**;**

Description

This event will be fired if an evaluation period base on DateLock has expired.

TSaveKind

Declaration

type TSaveKind = (skNone, skDef, skIni, skDll);

Description

Defines the type of the place, where the information is to be saved.

skNone can be used only for the secondary information target and turns it off
skDef default target
skIni defines an stadard INI file as target.
skDll defines a special kind of DLL as target.

skIni and skDll both require to name a file name.

TGenKind

Declaration

```
type TGenKind = (gkSingle, gkDouble);
```

Description

Defines the precision (uniqueness) of the generated registration keys.

gkSingle	standard method, compatible with previous versions
gkDouble	allows for generation of more unique keys

Attention!
Both the runtime environment and RTGENKEY utility must use the same method. The latter can be set through RTGENKEY.INI:

```
[System]
GenKind=<x>
```

where x=0 stands for gkSingle method, and x=1 equals gkDouble.

DLL

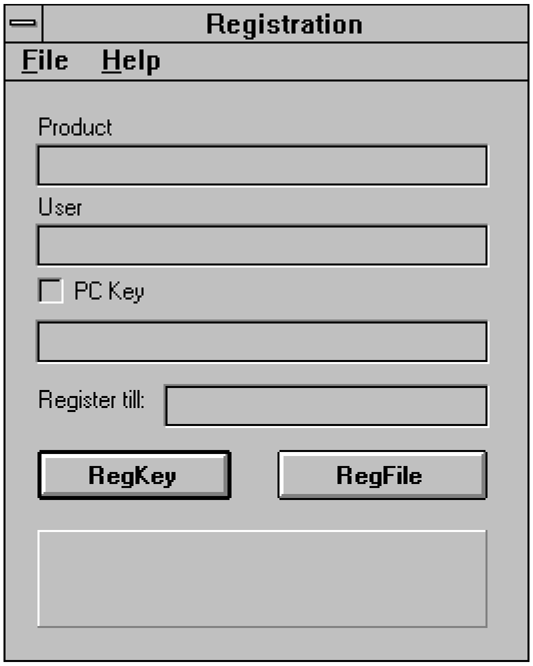
DLLs used to contain registration information should be real Windows excutable modules - DLL or EXE. The only requirement for them to be suitable is a Module Description of RIT0000000.

Registration Key Generation Utitliy

The registered version of RT Registration Control contains the program RTGENKEY.EXE. Its purpose is to generate a unique registration key based on the information provided by the user, as user key, PC key of users PC, end date of the registration and a product identification code provided by the author.

The user key should uniquely identify a user, it can be his first and last name for example. The PC key can be obtained only by the product being registered using the RTGetPCKey function call. Its result must be display to the user, so that he can give this information to the author. The end date of the registration is only for leased or rented software, and specifies the date until the program should work.

For description of the entry fields click on the hotspots on the picture below.



Registration

License fees:

Product	*Price*	*s&h*
Delphi only DCU (including a free VCL component)	$39 (55 DM)	$2 (3 DM)
DLL version	$39 (55 DM)	as above
both together	$59 (85 DM)	as above
Upgrades from 16 bit to 32 bit or the other way round:		
Delphi or DLL version	: $20 (30 DM)	0
Delphi and DLL version	: $30 (45 DM)	0

Only following payment methods will be accepted:

Compuserve: GO SWREG
ID# 8103 16 bit DCU version.
ID# 8104 16 bit DLL version
ID# 8105 16 bit both versions together

ID# 11052 32 bit DCU version.
ID# 11053 32 bit DLL version
ID# 11054 32 bit both versions together

ID# 11055 Upgrade 16 <-> 32 for DCU or DLL ersion
ID# 11056 Upgrade 16 <-> 32 for DCU and DLL version

Checks: Only checks drawn on a US bank valued in US$ or drawn on a German bank and valued in DM will be accepted. The checks should be made payable to **Tomasz Stanczak**. Please let us time after sending check, we cannot send you our product before the money is transferred.

Please fill in the Order Form and send with a payment to:

Tomasz Stanczak
Hardenbergstr. 8
31275 Lehrte
Germany

Credit Card Orders: You can order with MC, Visa, Amex, or Discover from Public (software) Library by calling 800-2424-PsL or 713-524-6394 or by FAX to 713-524-6398 or by CIS Email to 71355,470. You can also mail credit card orders to PsL at P.O.Box 35705, Houston, TX 77235-5705.

THE ABOVE NUMBERS ARE FOR CREDIT CARD ORDERS ONLY.
THE AUTHOR OF THIS PROGRAM CANNOT BE REACHED AT THESE NUMBERS.

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, non-credit card orders, etc, must be directed to Tomasz Stanczak:

CIS:	100735,3273
Internet:	100735.3273@compuserve.com
WWW:	http://ourworld.compuserve.com/homepages/tomasz

To order please specify Item# 14430, Rt Registration Control and one of the following options:

Product	*Price*	*s&h*
Delphi VCL 16 bit	$39	$2 (e-mail) $4 (Europe) $6 (overseas)
DLL 16 bit	$39	as above
both Delphi VCL and DLL 16 bit	$59	as above
Upgrade 16 to 32 bit for DCU or DLL ersion	$20	0
Upgrade 16 to 32 bit for DCU and DLL ersion	$30	0

To order please specify Item# 14580, Rt Registration Control and one of the following options:

Product	*Price*	*s&h*
Delphi VCL 32 bit	$39	$2 (e-mail) $4 (Europe) $6 (overseas)
DLL 32 bit	$39	as above
both Delphi VCL and DLL 32 bit	$59	as above
Upgrade 32 to 16 bit for DCU or DLL ersion	$20	0
Upgrade 32 to 16 bit for DCU and DLL ersion	$30	0

To insure that you get the latest version, PsL will notify us the day of your order and we will ship the product directly to you.

Shipping:

Currently only e-mail shipping is supported, so please always state your e-mail address!. In the future post shipping including printed documentation will be possible (no promise!).

DISCLAIMER: There will be no source code selling option, except Delphi VCL, because publishing the encrypting routines defeats the use of the product.

Order Form

Name (block
letters please): ______________________________

Company Name: ______________________________

Street Address: ______________________________

____

City, State,
Zip Code: ______________________________

Country: ______________________________

Phone Number: ______________________________

E-Mail Address: ______________________________

Product Name Quantity Price

____

____

____

____

Shipping______________________________

Total Fee______________________________

License agreement

Program and documentation Copyright 1995 Tomasz Stanczak (R&T). This software is licensed for a single registered developer. Unlimited royalty-free distribution permitted only as a part of an EXE file (DCU version) or as a runtime bound DLL file, provided that internal copyright notice is not altered or removed. If used to protect Delphi VCL components distribution of the following two files is permitted: RTREGB.DCU, RTREGC.DCU.

Registered users may not distribute or duplicate any documentation, source code or DCU files other then described above, may not distribute or sell any components or libraries based on any part of this product, either competing directly with RT Registration Control or offering similiar functionality.

Non-registered users may use the software to test it without any time frame, as long as they use the shareware version. A limited license is granted to copy and distribute shareware version of the software only for the trial use of others provided that all files accompanying it will also be copied and no fee will be requested or accepted.

Warranty

LIMITED WARRANTY

EXCEPT AS PROVIDED ABOVE, AUTHOR DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PRODUCT. SHOULD THE PROGRAM PROVE DEFECTIVE, THE PURCHASER ASSUMES THE RISK OF PAYING THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL AUTHOR BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION AND THE LIKE) ARISING OUT OF THE USE OR THE INABILITY TO USE THIS PRODUCT EVEN IF AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Support

Support is available from the author via Compuserve Mail to 100735,3273 , Internet Mail to 100735.3273@compuserve.com or http://ourworld.compuserve.com/homepages/tomasz. Currently there is no postal, telephone, or fax support. All support is on a "best effort" basis, and no promise is made that all questions can be answered, problems solved, features added nor bugs exterminated.

Revision History

1.20 15.01.96

New Features:

Now it is allowed to protect components. See changed license agreement.

Generation of the registration keys can be made more unique.

Corrected Bugs:

UserKey could be saved only up to 17 character, if skDef saving method were used.

The Till registration option didnt function properly.

1.11 23.12.95

under certain circumstances programs registered with the PCLock option werent recognized as such if a standard saving option was used.

1.10 08.12.95

New Features:

New registration data saving places: INI files, DLL files.

Secondary copy of all the data.

Corrected Bugs:

SetDateLock method did cause a stack overflow.

DateLock was too easy to beat.

RTCheckCountLock used to cause EConvertError exception.

Compatibility with previous versions

Because of the changes in the version 1.1, the KeyFiles created by new RTGENKEY will not be readable by programs linked with the version 1.0.

1.00 15.10.95 intial version.

Other products from R&T Software

RtControls Components Pack

Six Delphi VCL components:

RtCheckList
CompuServe Information Manager, Microsoft Word, MSOffice Setup all have a very interesting control. Every time I see it I wish I could use it. In Word select Extras|Optionen|Kompatibilität (german Version - in English it would probably be Extra|Options|Compatibility) and you will see it. There is a listbox with checkboxes in it. Compared with them the standard multiselect listbox seems boring. The selection is also somehow awkward. If you forget to press Shift or Ctrl key all your selections are gone. So I thought I can make it better. And there is a RtCheckList: a listbox full of checkboxes where your selections stay permanent, and because it uses bitmaps to show them, you can make it look as you want it to be.

RtCheckGroup
This component is something like Borlands TRadioGroup but it groups checkboxes instead of radiobuttons. Same as TRadioGroup it aligns ist child controls itself, so you dont have to do it. You can set them all at once of every one separately. Because it uses bitmaps to show them, you can make it look as you want it to be.

RtGauge
This little component does what you have certainly seen - it shows a progress bar. There are two other progress bars which have been shipped with Delphi - TBiGauge and TGauge. To tell the truth it was based on TGauge. So if you ask yourself why doing another Gauge component, the answer is easy. The TBiGauge is a VBX, so you have to distribute this along with your application, if you use it. TGauge is a true VCL, but it has at least one subtle error (if you want to know it, take it as your homework, if you then dont find it, send me an e-mail and Ill tell you). But the real reason is, that TRtGauge is smaller AND faster than the other. If the only thing you want to do, is a progress bar then look at it.

RtDbIniFile
Every customer wants something different, even if your code is as general as you can. This truth is nothing new, its just a principle I learned in the last years, it doesnt wonder me, it doesnt make me angry any more. I know I have to build my programs with as many parameters as I can. Well, it works, but if I let my user input all of the parameters needed every time he enters the program hes going to hate me. So I make configuration files. With plain DOS I used to work with DBF files because of my Clipper background. In Windows we have INI files. There is only one problem with INI files - everybody can change it, put garbage in or delete it. If you come with an application into bigger companies they certainly will ask you about security of you data. Well, TRtDbIniFile does all TIniFile can do with INI files with the same methods but it saves all this in database you are connected to, say Oracle, Sybase and the like. So you configuration data can be so secure as you database backend. If I needed to I could have used my old DBF files also.

RtDbCopy
TBatchMove is a very powerful component, but it has several shortcomings. What it cannot handle:
1) if you don't define a problem table and an error occurs, the whole operation will be canceled
2) it cannot change field data types in most cases
3) it cannot influence field data, it's copied as it is

RtDbCopy is a replacement component, which offers:
1) before and after copy event, to do housekeeping actions

2) on every record event, to gauge the whole process or implement function filter
3) on every field event, where you get the data being read and set data being written, so that you can feed three destination fields from one source field or do whatever is appropriate
4) on every error event, where you can ask the user to decide if he wants to continue or break
5) through Mappings definition it lets you change field data types and/or sizes

To achieve the best possible performance low-level BDE calls have been used whenever I could. At the end it comes very close to TBatchMove speed-wise.

RtDbGenID
RtDbGenID is a non-visual component which implements a unique number generator. All values will be saved into a table in database you work with. Every access which generates a new value locks this table and saves the generated value before releasing control, so that other users can safely run it parallel through the network. SQL queries as 'SELECT MAX(...)+1 FROM...' are a bit dangerous, because many users can theoretically execute this query at the same time and get the same results. Databases which support referential integrity can get over it, if the value is supposed for a field defined as unique, but even at the minimum it would rise an exception. dBase knows nothing about real uniqueness and in this case it can cause severe data errors.

GO SWREG:

Component Name	*SWREG ID#*	*Price*	*s&h*
RtControls	7197	$35 (50 DM)	$2.5 (3 DM)

All of the above components are also available separately:

RtCheckList	7191	$15 (20 DM)	$2.5 (3 DM)
RtCheckGroup	7193	$10 (15 DM)	$2.5 (3 DM)
RtGauge	7190	$10 (15 DM)	$2.5 (3 DM)
RtDbIniFile	7194	$10 (15 DM)	$2.5 (3 DM)
RtDbCopy	7405	$20 (30 DM)	$2.5 (3 DM)
RtDbGenID	7406	$10 (15 DM)	$2.5 (3 DM)

PsL Credit Card orders

Component Name	*Item#*	*Price*	*s&h*
RtControls	14429	$35 (50 DM)	$2 (e-mail) $4 (Europe) $6 (overseas)

Properties

CallsLeft
DaysLeft
EncodeKey
GenKind
KeyFile
PCKey
ProductKey
RegDaysLeft
Registered
SafeFileName
SafeKind
SaveFileName
SaveKind
UserInfo
Version

Methods

DelCountLock
DelDateLock
DelRegistration
DelUserInfo
IsCountLock
IsDateLock
IsRegistration
Register
RegisterByFile
SetCountLock
SetDateLock

Events

OnEndCountLock
OnEndDateLock
OnRegister
OnStartCountLock
OnStartDateLock

RTGetSaveFile

RTSetSaveFile

RTGetSaveKind

RTSetSaveKind

RTGetSafeFile

RTSetSafeFile

RTGetSafeKind

RTSetSafeKind

Should uniquely identify a product being registered.

Should uniquely identify a user being registered for the specified product. It can be users first and last name for example.

Check this checkbox if you wish that a PC key be included in generating the registration key. If not checked the following field will be disabled.

If the PCKey has been checked enter the PC key provided by the user.

The date until that the program being registered should work. Appropriate only for leased or rented software.

Generates a registration key.

Generates a registration file key.

Displays the generated registration key.