Table of Contents

1 TUnixCrypt - Overview

Overview

The UnixCrypt component is the ultra fast crypt implementation for Delphi/C++ Builder. It works exactly as **crypt(key, salt)** Unix comand (one-way encryption algorithm), which used to encrypt passwords in .htpasswd files in password protected Web directories.

Description

UnixCrypt is the password encryption component. It is based on the Data Encryption Standard algorythm with variations intended (among other things) to discourage use of hardware implementations of a key search.

Key property is a user's typed password. Salt is a two characters string chosen from the set [a-zA-Z0-9./]. This string is used to perturb the algo-encrypt repeatedly a constant string (usually a string consisting of all zeros).

The returned value (Result property) is the encrypted password, a series of 13 printable ASCII characters (the first two characters represent the salt itsef).

How to use?

Drop component on your form, specify password in Key property, put any two cahracters to Salt property and get encrypted password from Result. You can play with it even at design-time.

Since the UnixCrypt component uses one-way encryption algorithm, there is no way to *decrypt* the keys. For authentication you can only compare two encrypted passwords.

Example

```
procedure TForm1.AuthenticationBtnClick(Sender: TObject);
begin
  // we'd like to take salt from two first characters of username
  UnixCrypt1.Salt := Copy(RealUsername, 1, 2);
  // asking for password
  UnixCrypt1.Key := InputBox('Authentication',
                             'Enter password:', '');
  // comparing two encrypted passwords
  if UnixCrypt.Result <> RealCryptedPassword then
   begin
    ShowMessage('Authentication Failed!');
    Application.Terminate;
   end;
end;
```

Warning

The key space consists of 2**56 equal 7.2e16 possible values. Exhaustive searches of this key space are possible using massively parallel computers. Software (cracks), is available which will search for portions of this key space that is generally used by humans for password. Hence, password selection should, at minimum, avoid common words and names.

The DES algorithm itself has a few quirks which make the use of the *crypt* interface a very poor choice for anything other than password authentication. If you are planning on using *crypt* interface for a cryptography project, don't do it: get a good book on encryption.

2 Installation Instructions

Package without source code

to Delphi 2

1. Unzip files from "Delphi2" directory to your "Delphi 2\Lib" directory.
2. Start Delphi 2 IDE.
3. Select "Component\ Install..." menu item.
4. Press "Add" button and select "UnixCrypt.dcu" file.
5. Rebuild library.

to Delphi 3

1. Unzip files from "Delphi3" directory and copy them to "Delphi 3\Lib".
2. Start Delphi 3 IDE.
3. Open "UnixCryptD3.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 4

1. Unzip files from "Delphi4" directory and copy them to "Delphi 4\Lib".
2. Start Delphi 4 IDE.
3. Open "UnixCryptD4.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 5

1. Unzip files from "Delphi5" directory and copy them to "Delphi 5\Lib".
2. Start Delphi 5 IDE.
3. Open "UnixCryptD5.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 6

1. Unzip files from "Delphi6" directory and copy them to "Delphi 6\Lib".
2. Start Delphi 6 IDE.
3. Open "UnixCryptD6.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 7

1. Unzip files from "Delphi7" directory and copy them to "Delphi 7\Lib".
2. Start Delphi 7 IDE.
3. Open "UnixCryptD7.dpk" file.
4. Install package to the components palette ("Install" button).

to C++ Builder 1

1. Unzip files from "BCB1" directory to your "CBuilder\Lib" directory.
2. Start C++ Builder IDE.
3. Select "Component\ Install..." menu item.
4. Press "Add" button and select "UnixCrypt.dcu" file.
5. Rebuild library.

to C++ Builder 3

1. Unzip files from "BCB3" directory and copy them to "CBuilder3\Lib".
2. Start C++ Builder 3 IDE.
3. Open "UnixCryptCB3.bpk" file.
6. Select "Project \ Make UnixCryptCB3" menu item.

7. Select "Component\ InstallPackages" menu item.
8. Press "Add" button and select "UnixCryptCB3.bpl"file.

to C++ Builder 4
1. Unzip files from "BCB4" directory and copy them to "CBuilder4\Lib".
2. Start C++ Builder 4 IDE.
3. Open "UnixCryptCB4.bpk"file.
4. Install package to the components palette ("Install" button).

to C++ Builder 5
1. Unzip files from "BCB5" directory and copy them to "CBuilder5\Lib".
2. Start C++ Builder 5 IDE.
3. Open "UnixCryptCB5.bpk"file.
4. Install package to the components palette ("Install" button).

to C++ Builder 6
1. Unzip files from "BCB6" directory and copy them to "CBuilder6\Lib".
2. Start C++ Builder 6 IDE.
3. Open "UnixCryptCB6.bpk"file.
4. Install package to the components palette ("Install" button).

Source code
1. Uninstall / delete all previous(trial) instances of UnixCrypt.
2. Unzip files from "Sources" directory and copy them to "..\Lib" directory.
3. Run Delphi or ++ Builder IDE.
4. Select "Component\ Install..." menu item.
5. Press "Add" button and select "_UnixCryptReg.pas"file.
6. Rebuild library.

3 Registration Information

UnixCrypt component is SHAREWARE. This means that you can try it out for free, but if you like it and want to use it you have to register it with the author. Before continue read and accept license agreement please.

The only differencebetween the unregisteredand registeredversions is that the registered one has not message box with remind to register and works without Delphi (C++ Builder) running. You can also purchase the source code, if you would like to have it, and be able to compile or modify the UnixCrypt on any version of Turbo/Borland/Virtual/FreePascal, Delphi or C++ Builder.

If you would like to use the UnixCrypt and receive full, unrestricted version, priority support or even source code — you have to purchase proper license.

All prices in US dollars. Registering entitles you to unlimited support via E-Mail, minor version updates indefinitely and major version updates for 6 month from date of purchase.

Registration types:

Full, unrestricted version without source code:
- **https://secure.element5.com/register.html?productid=104898- $9,95**

Full version including 100% Source Code:

- **https://secure.element5.com/register.html?productid=104899 - $17,95**

4 License Agreement

Copyright

The UnixCrypt component (software) is Copyright © 1998-2002, by Utilmind Solutions® (Utilmind). All rights reserved.
The authors - Utilmind Solutions® and Aleksey Kuznetsov (founder of Utilmind), exclusively own all copyrights to the Advanced Application Controls (AppControls) and all other products distributed by Utilmind Solutions®.

Liability disclaimer

THIS SOFTWARE IS DISTRIBUTED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. YOU USE IT AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

Restrictions

You may not attempt to reverse compile, modify, translate or disassemble the software in whole or in part. You may not remove or modify any copyright notice or the method by which it may be invoked.

Operating license

Unregistered version

You may distribute the unregistered version of software freely, provided that all files are included and remain unmodified and that no extra files have been added to the package. You may not ask any money for the distribution. You may use the unregistered version of software free of charge for testing purposes, but if you want to use it for other purposes than testing - you have to register it with the author.

Registered version (single user license)

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use registered version of the software only by a single person, on a single computer at a time. You may physically transfer the software from one computer to another, provided that the software is used only by a single person, on a single computer at a time. In group projects where multiple persons will use the software, you must purchase an individual license for each member of the group or purchase site license. Use over a "local area network" (within the same locale) is permitted provided that the software is used only by a single person, on a single computer at a time. Use over a "wide area network" (outside the same locale) is strictly prohibited under any and all circumstances.

Registered version (site/team license)

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team only in one location (building complex). If you purchase a site license, you may use the program in an unlimited number of your company's computers within this area.

Registered version (Educational site license)
Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your educational organisation (school/college/university etc) in one location (building complex). If you buy a educational site license, you may use the program in an unlimited number of your edicational organisation's computers within this area.

Registered version (World-wide license)
Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team world-wide. If your company has many branches even with thouthands of computers, world wide license covers them all.

Notes (clarification)
"Single-user license" means "single-developer license". "Site license" means that it can be used by any number of software developers within your company.
You can use purchased components in ANY number of your projects and deploy the "end-user" software to ANY number of your users/customers without any additional royalty fees. Howeveryou are not permitted to distribute the component itself (the source code or .dcu files of components).

Back-up and transfer
You may make one copy of the software solely for "back-up" purposes, as prescribed by international copyright laws. You must reproduce and include the copyright notice on the back-up copy.

Terms
This license is effective until terminated. You may terminate it by destroying the program, the documentation and copies thereof. This license will also terminate if you fail to comply with any terms or conditions of this agreement. You agree upon such termination to destroy all copies of the program and of the documentation, or return them to author.

Other rights and restrictions
All other rights and restrictions not specifically granted in this license are reserved by authors.

5 TUnixCrypt - Properties

5.1 Key

Applies to
UnixCrypt component.

Declaration
```
property Key: String;
```

Description
The Key property is a user's typed password. The password can be any length, but if used password with length over 8 characters, only 8 first characters will be used.

See also
Salt and Result properties.

5.2 Salt

Applies to
UnixCrypt component.

Declaration
```
property Salt: String;
```

Description
Salt is a two characters string chosen from the set [a-zA-Z0-9./]. This string is used to perturb the algo-encrypt repeatedly a constant string (usually a string consisting of all zeros).

See also
Key and Result properties.

5.3 Result

Applies to
UnixCrypt component.

Declaration
```
property Result: String; // Read-only!
```

Description
The Ruturn property is the read-only property, which contains the result of encryption. The returned value is the encrypted password, a series of 13 printable ASCII characters (the first two characters represent the Salt itsef).

See also
Key and Salt properties.