Database Access Control Manager

Support - EMail 72361.2107@compuserve.com or the 3rdParty section of the CIS Delphi forum subject DACM.

Welcome to Version 2 of the Database Access Control Manager (DACM).

What is DACM?

DACM is a flexible, easy to use, Delphi package that provides a comprehensive and professional security system handling all aspects of security for your Delphi database applications. Although simple in design DACM can be used to create complex security strategies.

Thank you for trying the Database Access Manager. A lot of time and effort has gone into this product and we hope you will find it a useful tool to add to your application but if you use it don't forget to register it!

Reference

Security Model Overview

see also

The Database Access Control Manager (DACM) is a powerful, integrated database access control system providing application wide security.This system has been designed to be both simple to administer and flexible in construction to meet the many different requirements that may exist in your application's environment whilst providing watertight protection of your application and its data sources.

Access control within DACM consists of three principal elements, User Accounts (Users), User Groups (Groups) and Securitised Objects (Objects). By the careful interaction of these elements system administrators can create a variety of security strategies and protect a wide range of activities and data sources.

User accounts are created in the system and are allocated a Personal Identity Number (PIN).Users are then assigned to Groups and finally Groups are granted access to particular Objects.

example.
An account is created with a UserName of Richard and a PIN of STAR1, he is assigned to the Management Group and this group is granted access to the Auto Sales Folder Object.

All Security administration except for changing a Users PIN is handled by the members of the administrators group. User PIN's are usually set by the User however there is a design time option to allow PIN allocation to be administered centrally by member's of the Administrator's group.

Security Overview

Users

A user account is identified to the system by a UserName and Personal Identity Number(PIN). Without an account a user cannot log on to the system.

Although a UserName is not unique the combination of a UserName and a PIN is unique, this means that although there can be only one Richard/STAR1, it is possible for there to be a Richard/RICH, this feature is particularly useful where you may wish to have different user profiles for carrying out different activities i.e. when administering the system Richard may log on as Richard/ADMIN and this profile has Administrator group access rights, whilst normally Richard logs on as Richard/STAR1 which has Sales group access rights.

Unless the option has been set at design time for central PIN administration a user can change his/her own PIN whenever they LogOn, it is the Users responsibility to change there own PIN. After the initial creation of a User account an administrator will not know (and cannot find out) a particular users PIN nor can they amend it nor set it to a null value. If a User forgets his own PIN the account must be deleted and recreated and Group memberships reset.

Users Interface

Groups

Groups are a collection of user accounts. DACM allows you to create Groups to simplify the administration of control of access to data and program functions.
You may create as many groups as you wish, you then can grant these groups access to particualr objects.
Each group may have rights to access a different set of Objects.These rights and the membership of a group can be changed at any time by an administrator.

Groups Interface

Objects

Objects are anything within the application that you wish to restrict access to, it could be a Form or Folder or it could be a Report.

Within your application when a secured object is accessed the permissions will be checked for the current user. The User may belong to anumber of different groups of which more than one may have access permissions for the object. If so the users permissions returned will be the sum of all these groups.

For example:

User RICH/STAR1 access's the management folder, this object has given ReadOnly and, Insert permissions to the Local Managers group and ReadOnly, Modify and Delete permissions to mebers of the Head Office Managers group. RICH/STAR1 is a member of both groups so his permissions are ReadOnly,Insert,Modify and Delete.

The creation of Securitised Objects is handled totally by the system, the administrator need only be concerned with granting a Group access to a particular Object.

Access to an Object may be granted or revoked at any time by the administrator. The changes take place immediately i.e. when changed groups are instantly either granted or denied access to an object even if they are currently logged on to the system.

Objects Interface

Users Interface

The Users interface is on the Users page of the Security Maintenance dialog box, this is displayed automatically when a member of the Administrators group logs on or by calling Maintenance method of the global security object.

Name

This is a combo box containing the UserName for each User account. All subsequent actions are carried out on the User account that you have currently selected in this combo box.

When you select a user account the Member Of listbox is updated to show all groups that this account is a member of. The Available listbox list's all groups that this account is currently not a member of.

The Assign buttons are used to change the groups that the currently selected Users belongs to.You can also double click on an item to transfer it from one listbox to the other.

When you are finished assigning groups for this User press the save button to update the Users security record in the DAC Table.

Save

Saves any changes to the group assignments you may have made for the currently selected User account.

Cancel

Undoes any unsaved changes by reloading the Group memberships as last saved to disk.

Add

The Add button displays the New User dialog box, allowing you to create new user accounts. You must enter the UserName (up to 30 characters) and PIN (up to 20 characters) to complete the New User dialog.

Delete

Deletes the User account that is currently selected in the User combobox.

Finished

Closes the dialog box and returns to your application.

Groups Interface

The Groups Interface is on the Groups page of the Security Maintenance dialog displayed automatically on LogOn by members of the Administrators group or by calling the Maintenance method of the global security object .

Name

This is a combo box containing the GroupName for each UserGroup. All actions are carried out on the UserGroup that you have currently selected in this combo box.

Add

The Add button displays the New Group dialog box, allowing you to create new user groups. You must enter the GroupName (up to 20 characters) and a four alpha/numeric PIN to complete the New Group dialog.

Delete

Deletes the User Group that is currently selected in the Group combobox.

Finished

Closes the dialog box and returns to your application.

Objects Interface

The Objects interface is on the Objects page of the Security Maintenance dialog box, this can be displayed either automatically when a member of the Administrators group logs on or by calling Maintenance method.

Name

This is a combo box containing the ObjectName for each securitised object currently defined.All actions are carried out on the Object that you have currently selected in this combo box.

When you select an object the Assigned listbox is updated to show all groups that have been granted access to this object.The Available listbox list's all groups that this object has currently not granted access to.

The Assign buttons are used to change the groups that the currently selected Object grants access to.You can also double click on an item to transfer it from one listbox to the other.

When you are finished assigning groups for this Object press the save button to update the Objects security record.

Permissions

The permissions section shows the permission or access level granted to a particular group by the current object.
Selecting a group in the assigned listbox will automatically update this section to the display its current permissions.
You can change these permissions by clicking the relevant checkbox to assign or unassign a permission.
You can Set permissions for more than one group at the same time by selecting multiple groups in the Assigned Listbox then setting the permissions.

Save

Saves any changes to the group assignments you may have made for the currently selected Object.

Cancel

Undoes any unsaved changes by reloading the group assingments as last saved to disk.

Add

This button is disabled on the Objects page, new objects can only be created under program control using the AppendObject method.

Delete

This button is disabled on the Objects page, objects can only be deleted under program control by using the DeleteObject method.

Finished

Closes the dialog box and returns to your application.

Log On

The LogOn dialog box can be accessed either by calling the LogOn method or automatically on the first call to the CheckPermissions method.

Name

This box is used to enter your UserName.

PIN

This box is used to enter your PIN.

OK

This marks as complete your UserName and PIN and verifies your account. If your account is verified then you will be granted access to the application, otherwise the dialog box is redisplayed and you can try again upto the maximum number of retries set by the LogOnTries property.

Cancel

This aborts the LogOn process.

Change PIN

The change PIN dialog box is displayed either by checking the change PIN option on the LogOn dialog box or by calling the ChangePIN method. You can change your PIN to any alpha/numeric combination of upto 20 characters. If the MinPinLength property of the active DACM has been set the PIN must exceed the minimum length specified by that property.

Old PIN

This field must match your existing PIN.

New PIN

This field is used to enter your new PIN.

Verify

This field must match the New PIN.

TSecurity Object

Properties Methods Events

Unit

DACMgr

Description

Within the DACMgr unit, the variable Security is declared as an instance of TSecurity, ready for you to use, you should not delete this object or try to create a new one.

If not automatically logged on by the security system you can call the LogOn method to log a new user on at any time. Call the ChangePIN method to change the current Users PIN. Call the Maintenance method to display the Security Maintenance dialog and allow access to all the maintenance facilities. Call the Verify or VerifyUI methods to have the user reconfirm their PIN.

Test the LoggedOn property to determine if a user is currently logged on. The value of the UserName property is the name of the current user.

Use theAppendUser, AppendGroup, and AppendObject methods to add new security records under program control. Call the DeleteObject method to permanently remove a security record from the DACM database.

Call the CheckPermissions method any time you wish to check whether the user has access rights to a particular object and what level of permissions they have.

TSecurity Properties

>Run-time only

>LoggedOn	>UserName	>UserOID

TSecurity Methods

AppendUser	ChangePIN	Maintenance
AppendGroup	CheckPermissions	Verify
AppendObject	LogOn	VerifyUI

UserOID Property

Applies to

TSecurity

Declaration

property UserOID : longint;

Description

This property contains the OID of the User currently logged on to the security system.

UserName Property

Applies to

TSecurity

Declaration

property UserName : TSecurityName;

Description

This property contains the UserName for the User currently logged on to the security system.

LoggedOn Property

Applies to

TSecurity

Declaration

property LoggedOn : Boolean;

Description

This property indicates wether there is currently a valid user logged on to the security system

LogOn Method

Applies to

TSecurity

Declaration

procedure LogOn;

Description

The LogOn method displays the LogOn dialog allowing a new user to LogOn to the security system. If the UserName and PIN are correctly entered and verified this procedure will set the LoggedOn property to True else it will raise an exception of type ESecurity..

If the UserName or PIN are incorrect then the dialog is redisplayed and the user can retry up to the number of times specified in the LogOnTries property of the active DACM, if this value is exceeded then this procedure will raise an exception and set the LoggedOn property to False.

If the LogOnPrompt property of the active DACM is set to False then the LogOn dialog is not displayed and instead a LogOn Event is raised, upon returing Security will use the UserName and UserPIN parameters of that event to perform a silent LogOn to the security system.

VerifyUI Method

Applies to

TSecurity

Declaration

function VerifyUI : Boolean;

Description

The VerifyUI method displays a MessageBox to allow the User to re-enter there PIN. This is then checked against the PIN for the currently Logged on User.

If the PIN does not match the function returns False.

This method is most useful where you wish to control access to a particularly important object and you want to reconfirm that a valid user is attempting to acces the object or where you wish to have the User enter a password to approve each transaction in a batch.

Verify Method

Applies to

TSecurity

Declaration

function Verify (APIN:string) : Boolean;

Description

The Verify method is similar to the VerifyUI method except it does not display a MessageBox for the User to re-enter their PIN, instead the PIN must be supplied as a parameter to the function.

If the PIN doesnt match the PIN for the currently logged on User the function returns False.

This method is most useful where you wish to control access to a particularly important object and you want to reconfirm that a valid user is attempting to acces the object or where you wish to have the User enter a password to approve each transaction in a batch.

ChangePIN Method

Applies to

TSecurity

Declaration

procedure ChangePIN;

Description

This method displays the ChangePIN dialog, to allow the current user to change their PIN .

Maintenance Method

Applies to

TSecurity

Declaration

procedure Maintenance;

Description

This method displays the Security Maintenance dialog which allows access to all the maintenance features.

See also:

Users Interface

Groups Interface

Objects Interface

CheckPermissions Method

Applies to

TSecurity

Declaration

function CheckPermissions (SID:TSecuritySID) TPermissions;

Description

Call this method to check the rights the current user has to access an object.

SID is the identifier previously set by your application to identify the object you wish to check i.e. the report, form or menu.

If the user has access rights the function returns those permissions as a set, which can be manipulated using Delphi's Set functions and procedures. If the user has been granted access but has no assigned rights or the user has not been granted access the function returns an empty set (see Delphi Help on Sets).

By default these Permissions are assumed to be:-
0 = Read Only
1 = Insert Data
2 = Modify Data
3 = Delete Data
4 = Approve Data
5 = Run
6 = NotAssigned
7 = NotAssigned

These can be changed at design time to anything you want.

It should be noted that a user may belong to a number of groups that have access to a particular object each with its own particular permissions. The permissions returned by this function will be the sum of these various permissions.
i.e. if Sales has Modify rights and Circulation has Insert rights then the permissions returned will be both insert and modify.

AppendUser Method

Applies to

TSecurity

Declaration

procedure AppendUser (UserName:TSecurityName;PIN :TSecurityPIN; var OID :longint);

Description

This method allows you to add a new user under program control, if unsuccessful it will raise an exception.

On returning the OID parameter contains the objects OID, this can be ignored and is only used internally for updating dialogs when using the User interface.

AppendGroup Method

Applies to

TSecurity

Declaration

procedure AppendGroup(Name: TSecurityName; var OID:longint);

Description

This method allows a new Group to be created under program control, it will raise an exception if unsuccessful.

The Name is a string of up to 30 characters that is used as a label in the user interface to refer to the group.

On returning the OID parameter contains the groups OID, this can be ignored and is only of use when this method is called as part of the user interface to update certain dialogs.

AppendObject Method

Applies to

TSecurity

Declaration

procedure AppendObject(Name: TSecurityName;SID:TSecuritySID;var OID longint);

Description

This function allows a new Securitised Object record to be created, the procedure will raise an exception if unsuccessful.

The SID is a string of up to 50 characters set by you that uniquely identifies this object. The Name is a label to be displayed when referring to this object via the user interface.

On returning the OID parameter contains the objects OID, this can be ignored and is only relevant internally to DACManager when accessing this method via the user interface.

The object to be secured can be anything a form, report, or even a menu selection.
The first thing you most likely will secure will be Forms, whilst the SID can be anything you wish an obvious candidate for selection would be the form name, whilst the form caption would be a likely candidate for the Name.
For other objects you will need to make up the SID, remember that the SID will only be used by the program, it will never be seen by the user so there is no need to make it a long descriptive string, the user will only see the Name (via the Maintenance interface) so this must be descriptive enough to enable the user to identify the object it refers to.

Access permissions for a secured object are set by the user via the Maintenance interface, whilst checking whether a user can access an object is handled by your program calling the CheckPermissions method before opening or executing the object.

TDACManager Component

Properties Methods Events

Unit

DACMgr

Description

The DACM component is derived from a TTable and provides live access to the DAC Table through the Borland Database Engine.

Set the DatabaseName property to specify the database to access. Set the TableName property to the table to access.

Set the IsSQLBased property if the Security Table is located in a SQL Server table, this will enable the Security object to correctly handle the LogOn procedure to the server.

Set the Key property to a private word sized numeric key for encrypting data in the Security Table.

Set the LogOnPrompt to indicate whether you wish DACM to display the standard LogOn dialog. If false the Security Object will generate a LogOn Event and allow you to provide the UserName and PIN from another source or dialog.

Set the MinPINLength to other than 0 if you require the User to create or enter a PIN that is at least this length.

The Password property is used to specify the Password used to encrypt the Security Table, if the Security Table is not a Paradox table this property will be ignored, DACM will automatically open the Security Table without any user intervention.

Set the AdminName and AdminPIN properties for the default values for the master Administrator account which will be created the first time you use the application.Note that for increased security this account can be deleted after logging on for the first time.

Set LogOnTries to the number of unsuccessful attempts allowed before a LogOn failure occurs.

Set the PINValidPeriod property to the number of days before a User must change there PIN, by default the value is zero and is ignored by the global Security object.

In addition to these properties, methods, and events, this component also has the properties and methods that apply to all components.

TDACManager Properties

In addition to events inherited from the TTable component, this component introduces the following:

>Run-time only

Active	IsSQLBased	LogOnTries
AdminName	Key	MinPINLength
AdminPINChange	LogOnPrompt	Password
		PINValidPeriod

TDACManager Methods

In addition to methods inherited from the TTable component, this component introduces the following:

DeleteObject

DeleteObject Method

Applies to

TSecurity

Declaration

procedure DeleteObject(OID :longint);

Description

This method allows you to delete any security record, User, Group or, Object under program control. The OID parameter identifies the object to be deleted.Users and Groups are usually deleted under manual control via the Maintenance interface.

An exception will be raised if this procedure fails.

TDAC Manager Events

In addition to events inherited from the TTable component, this component introduces the following:

OnBuildDAO OnSecurityEvent

OnBuildDAO Event

Applies to

TSecurity

Declaration

property OnBuildDAO: TNotifyEvent;

Description

The OnBuildDAO event is raised whenever the DAC Table needs to be recreated.

When first loaded the DACM checks the DAC Table, if the table is empty, either as a result of the original table having been damaged and a new empty table substituted or it is the first time the application has been run, then the event is raised.

The Administrators account and the Administrators Group are recreated (using the property values AdminName and AdminPIN) and then the OnBuildDAO event is raised to allow your application to regenerate the object table.

In the OnBuildDAO event your application should repeatedly call the AppendObject method to recreate the object table.

Example:

```
TDACManger.OnBuidDAO(Sender:TObject);
var
OID:string;
begin

AppendObject('Form 1','FRM1',OID);
AppendObject('Daily Report','RPT1',OID);
AppendObject('End of Month Closing','PRG1,OID);

end;
```

OnSecurityEvent

Unit

DACMgr

Applies to

TSecurity

Declaration

Property OnSecurityEvent:TSecurityNotifyEvent;

Description

The OnSecurityEvent occurs each time the users performs one of the following tasks, LoggingOn, ChangingPIN, accessing the Maintenance dialog, checking Permissions, writing to a security record (either creating a new record or editing an existing record) or changing Permissions for an object.

The most common reason to handle this event is to write a record to a log to enable tracking of actions performed by the user.

TSecurityObject Component

Properties Methods Events

Unit

DACMgr

Description

The TSecurityObject is derived from a TComponent. This component is used to simplify the control of access to an object.

Set the ObjectSID property to the SID of a delphi object you wish to control access to.

A Delphi object's SID is an up to 50 Character string that uniquely identifies the object, most commonly the objects name property is used.

By default this component will select the form it is placed on as the object to protect and use the forms name as the ObjectSID property.

During loading the component will determine the permissions the current user has for the Object identified by the ObjectSID property. It will also generate an OnPermissions event allowing you to carry out other tasks based on the permissions granted, for example you could set the state of navigator buttons on a form based on the User's permissions level for that form (ReadOnly, Insert, Modify etc.).

The Permissions can be checked at anytime by checking the Permissions property.

Permissions can be recalculated by calling the components Refresh method, this will also generate an OnPermissions event.

In addition to these properties, methods, and events, this component also has the properties and methods that apply to all components.

Note: Calling the CheckPermissions method of the global Security object when there is no User logged on will cause the security object to automatically call the LogOn method and an attempt to LogOn a User. If a valid user cannot be logged on CheckPermissions returbs an empty Permisson set.

TSecurityObject Properties

>Run Time Only

ObjectSID >Permissons

ObjectSID

Unit

DACMgr

Applies to

TSecurityObject

Declaration

Property ObjectSID:TSecuritySID;

Description

The ObjectSID property contains the SID of the security object that the TSecurityObject component is controlling access to. It must already exist in the DAC Table and uniquely identifies an object or component.

If the ObjectSID cannot be located in the DAC Table the Permissions property will return an empty set.

Permissions

Unit

DACMgr

Applies to

TSecurityObject

Declaration

Property Permissions:TPermissions;

Description

The Permissions property returns the set of permissions the current user has for accessing the object identified by the ObjectSID property of the TSecurityObject component

TSecurityObject Methods

Refresh

Refresh

Applies to

TSecurityObject

Declaration

procedure Refresh;

Description

The Refresh method rechecks the current users permissions fot the object identified by the SID property. Calling the Refresh method will generate an OnPermissions event.

TSecurityObject Events

OnPermissions

OnPermissions

Applies to

TSecurityObject

Declaration

property OnPermissions:TNotifyEvent;

Description

The OnPermissions event occurs when ever the TSecurityObject checks the permissions for the current user.

The event is raised after the permissions property has been set to the current permissions for the current user.

Your application can use this event to set the state of buttons or menus or other objects on the form or carry out other related tasks.

Using TDACManager

Follow the steps below to set up a security system for your application:

1 Place aTDACManager component on the main form of your application.

2 Set the DatabaseName property to the database that contains the DAC Table.

3 Set the TableName property to the name of the DAC Table.

4 Set the AdminName and AdminPIN properties if you wish the master administrator to be something other than Administrator/****.

5 Set the LogOnTries to the number of failed logon attempts you wish the user to be allowed (default is 3).

6 Create an OnBuildDAO event, call AppendObject for each object that you wish to be protected by the DACManger.

7 Add a TSecurityObject to your main form if you wish to automatically LogOn when your application starts, to LogOn at some other time call the LogOn method of the global security object.

8 Add the DACMGR unit to the uses clause of each unit in your application and call the CheckPermissions method at each point in your program where you need to check whether the current user has the right to access an object.

9 Compile and run your application.

Installation

In addition to the installation described below you may make one copy of this software for backup purposes only.

Unzip the file DACM.ZIP using the -D parameter ie PKUNZIP -D DACM.ZIP.

This should create a directory structure as follows;

\DB - Contains a blank paradox DAC Table .
\VCLLIB - DAC Manager v2.0 DCU's, RES and help file.
\VCLSRC - DAC Manager v2.0 Source files.
\DEMO - Demo application and security database VSEC.DB.
README.TXT - Readme file.

Copy the contents of the VCLIB directory to the \DELPHI\LIB directory and install the unit DACMGR.DCU following the instructions given in the Delphi manuals for installing components.

This will install new components DACManager and SecurityObject on the Data Access page of the VCL.

To run the DEMO application;
Create a new BDE Alias "SECURITY" and point this to the \DEMO directory, noted above, containing the Paradox database VSEC.DB.

Start Delphi, open and compile the project PROJECT1.

Run the demo application PROJECT1.EXE. Use the USERNAME "Administrator" and the PIN "MIKE" to LogOn.

Check the demo source code to see more on how DACManager is used to control access to objects.

To see how Access rights change with users try logging on as USERNAME "SUE" and a PIN "1234".

Index

Glossary

Registration and Support

Registration

The demo version of DAC Manager v2.0 has a limited time use, you're application will not load if it contains DACM after the expiry date, you MUST be registered to receive the unrestricted version and to be able to legally distribute applications containing this component.

COMPUSERVE USERS: SWREG Orders Only

You can register via CIS SWREG forum. GO SWREG and follow the instructions for registering shareware.

PRICE - US$45.00
The Registration ID for DACManger v2.0 is **6486**.

To ensure that you get the latest version, CIS will notify us the day of your order and we will Email the latest full registered version of our software to your CIS address, usually within 48hrs.

INTERNET USERS: CREDIT CARD Orders Only

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.

PRICE - US$50.00
The Registration ID for DACManger v2.0 - DACM is **14540**.

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

You can also order OnLine by visiting our Web Site at :
http://ourworld.compuserve.com/homepages/robedgar/
follow the instructions for software registration.

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 **72361.2107@compuserve.com**.

To insure that you get the latest version, PsL will notify us the day of your order and we will Email directly to you the password that is required to unzip the latest full registered version of our software which is available for download from our Web Site's Software Registration page.

Registered users will receive:

1 Full unrestricted version of DAC Manager v2.0 .

2 Full component source code.

3 Revised licence agreement, allowing you to legally distribute applications containing the DACManager component, plus allowing the help file or its contents to be distributed or copied or included in your own application help file.
Help file RTF source documents available upon request.

4 Free future minor version upgrades to DAC Manager when available.

Support

Support is currently only available via EMail!!!!

EMail to CIS ID 72361,2107 - Rob Edgar (Internet Users: 72361.2107@compuserve.com)

or send Email to the CIS DELPHI forum. ThirdPartyTools section. with the subject DACM.

License Agreement - Release Version

Important

By using this software you accept the following terms of this License Agreement. If you do not agree with these terms, you should not use the software and promptly return it for a refund.

Ownership

Visual Solutions Ltd. retains the ownership of this copy of the enclosed software package. It is licensed to you for use under the following conditions:

You May

You may transfer this software to another party if the other party agrees to the terms and conditions of the agreement and completes and returns a registration card to Visual Solutions Ltd. The registration card is available by writing to Visual Solutions Ltd. If you transfer the software, you must simultaneously transfer all documentation and related disks.
You may merge this software with your own software or code provided that the primary purpose of your software is not database access control, such software may be distributed without payment of a royalty fee.
You may copy any part of this help file, with the exception of this license agreement, for the sole purpose of informing the users of your software on how to use the security features of the DAC Manger.

You May Not

You may not copy the documentation or software except as described in the installation section of this manual. You may not distribute, rent, sub-license or lease the software or documentation, including translating, decompiling, disassembling, or creating derivative works. You may not reverse-engineer any part of this software, or produce any derivative work. You may not make telecommunication transmittal of this software.

Termination

This license and your right to use this software automatically terminates if you fail to comply with any provision of this license agreement.

Rights

Visual Solutions Ltd. retains all rights not expressly granted. Nothing in this license agreement constitutes a waiver of Visual Solutions LTD's rights under the H.K. copyright laws or any other law.

Limited Warranty

If you discover physical defects in the media, Visual Solutions Ltd. will replace the media or documentation at no charge to you, provided you return the item to be replaced with proof of payment to Visual Solutions Ltd. during the 90-day period after having taken delivery of the software.

License Agreement

Visual Solutions Ltd.excludes any and all implied warranties, including warranties of merchantability and fitness for a particular purpose and limits your remedy to return the software and documentation to Visual Solutions Ltd. for replacement.
Although Visual Solutions Ltd. has tested the software and reviewed the documentation, Visual Solutions Ltd. MAKES NO WARRANTY OF REPRESENTATION, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS SOFTWARE OR DOCUMENTATION, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE AND DOCUMENTATION ARE LICENSED "AS IS" AND YOU, THE LICENSEE, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.

IN NO EVENT WILL Visual Solutions Ltd. BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE

SOFTWARE OR DOCUMENTATION, even if advised of the possibility of such damages. In particular, Visual Solutions Ltd. shall have no liability for any data stored or processed with this software, including the costs of recovering such data.
THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESSED OR IMPLIED. No Visual Solutions Ltd. dealer, agent, or employee is authorized to make any modifications or additions to this warranty.

Information in this document is subject to change without notice and does not represent a commitment on the part of Visual Solutions Ltd. The software described in this document is furnished under this license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the written permission of Visual Solutions Ltd.

Some countries do not allow the exclusion of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from country to country.

Active

Applies to

TDACManager

Declaration

Property Active: Boolean;

Description

Setting the Active property to True causes the DACM to automatically attach to the global Security object and to become the Active DACM during the Loaded procedure. This then becomes the table against which all actions are performed.

If set to false the DACM simply loads and is inactive, this feature can be used by applications that will work on multiple security tables or perform both a local and a remote logon.

A DACM can attach to the global security object at anytime by calling the AttachTable method of the global security object.

Unlike TTable this property has NO DIRECT effect on the state of the underlying Table.

AdminName

Applies to

TDACManager

Declaration

property AdminName : TSecurityName;

Description

Set the AdminName property to specify the master Administrators account name.
This will be used whenever the account needs to be recreated before raising the OnBuildDAO event.

The Administrator's account can subsequently be deleted to provide additional security.

AdminPIN

Applies to

TDACManager

Declaration

property AdminPIN :TSecurityPIN;

Description

Set this property to the master Administrators PIN., a string of up to twenty characters

The Administrators PIN can be changed subsequently to anything the user desires during LogOn or by calling the ChangePIN method but whenever the Account is recreated it will revert to this default value.

License Agreement - Demo Version

Important

By using this software you accept the following terms of this License Agreement. If you do not agree with these terms, you should not use the software and promptly return it for a refund.

Ownership

Visual Solutions Ltd. retains the ownership of this copy of the enclosed software package. It is licensed to you for use under the following conditions:

You May

You may transfer this software to another party if the other party agrees to the terms and conditions of the agreement and completes and returns a registration card to Visual Solutions Ltd. The registration card is available by writing to Visual Solutions Ltd. If you transfer the software, you must simultaneously transfer all documentation and related disks.

You May Not

You may not copy the documentation or software except as described in the installation section of this manual. You may not distribute, rent, sub-license or lease the software or documentation, including translating, decompiling, disassembling, or creating derivative works. You may not reverse-engineer any part of this software, or produce any derivative work. You may not make telecommunication transmittal of this software.

Termination

This license and your right to use this software automatically terminates if you fail to comply with any provision of this license agreement.

Rights

Visual Solutions Ltd. retains all rights not expressly granted. Nothing in this license agreement constitutes a waiver of Visual Solutions Ltd.'s rights under the H.K. copyright laws or any other law.

Limited Warranty

If you discover physical defects in the media, Visual Solutions Ltd. will replace the media or documentation at no charge to you, provided you return the item to be replaced with proof of payment to Visual Solutions Ltd. during the 90-day period after having taken delivery of the software.

License Agreement

Visual Solutions Ltd.excludes any and all implied warranties, including warranties of merchantability and fitness for a particular purpose and limits your remedy to return the software and documentation to Visual Solutions Ltd. for replacement.
Although Visual Solutions Ltd. has tested the software and reviewed the documentation, Visual Solutions Ltd. MAKES NO WARRANTY OF REPRESENTATION, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS SOFTWARE OR DOCUMENTATION, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE AND DOCUMENTATION ARE LICENSED "AS IS" AND YOU, THE LICENSEE, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.

IN NO EVENT WILL Visual Solutions Ltd. BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, even if advised of the possibility of such damages. In particular, Visual Solutions Ltd. shall have no liability for any data stored or processed with this software, including the costs of recovering such data.
THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL

OTHERS, ORAL OR WRITTEN, EXPRESSED OR IMPLIED. No Visual Solutions Ltd. dealer, agent, or employee is authorized to make any modifications or additions to this warranty.

Information in this document is subject to change without notice and does not represent a commitment on the part of Visual Solutions Ltd. The software described in this document is furnished under this license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the written permission of Visual Solutions Ltd.

Some countries do not allow the exclusion of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from country to country.

LoggedOn Property

Applies to

Declaration

property LoggedOn: boolean;

Description

Run time and read only. This property specifies whether the application has a valid logged on user.

LogOnTries

Applies to

TDACManager

Declaration

property LogOnTries: integer;

Description

Set this property to the number of unsuccessful attempts a User may make to LogOn before raising a LogOn exception.

UserName Property

Applies to

TDACManager

Declaration

property UserName :TSecurityName;

Description

Run time and read only. This property specifies the name (up to 30 characters) of the currently logged on user.

Active
<Active>

AdminName
<AdminName>

AdminPIN
<AdminPIN>

AppendObject

<AppendObject Method>

AutoLogOn
<AutoLogOn>

ChangePIN

<ChangePIN Method>

CheckPermissions

<CheckPermissions Method>

DAC Table

The data table containing all security records stored by the DAC Manager for your application.

DACM

The Database Access Control Manager

Groups

A collection of users that have the same profile or privileges granted under the DACM.

LoggedOn

<LoggedOn Property>

LogOn dialog

<Log On>

LogOn method

<LogOn Method>

LogOn

<LogOn Method>

LogOnTries
<LogOnTries>

Maintenance

<Maintenance Method>

Objects

Data, forms, reports or other features that have been secured and are controlled via the Access Control System.

OID

Object Identifier, a unique longint created by DACM and used internally to identify any user or group or object.

OnBuildDAO

<OnBuildDAO Event>

Permissions

A set of values from 0 to 7 indicating the level of access granted to the current user for an object. By default these are 0 = Read Only, 1 = Insert Data, 2 = Modify Data, 3 = Delete Data, 4 = Approve Data, 5 = Run/Execute, 6 and 7 are undefined.

PIN
Personal Identification Number, a 20 alpha/numeric character string.

SID

System Identifier, a unique alpha/numeric string of upto 50 characters identifying a security object to your application.

UserName

The users name, an alpha/numeric string of up to 30 characters.A UserName is not unique, however the combination of the UserName and PIN (also known as the SID) is unique.This can be useful when the user wishes to have different profiles for different activities i.e. one profile for normal work and another for administration.

Users

User accounts identified by a unique combination of a UserName and a PIN

TSecurityNotifyEvent

Unit

DACMgr

Declaration

TSecurityNotifyEvent = procedure(EventType : TSecurityEvents ; Info1 :longint ; Info2 :byte) of object ;

Description

The TSecurityNotifyEvent is the type for all security event notifications.

The most common reason to handle this event would be to enable you to update a log to track actions performed by the user.

The Info1 parameter will hold when relevant an OID, Info2 will hold the permissions as indicated in the table below.

Event Type	Info1	Info2
LogOn	Not Used	Not Used
PINChange	Not Used	Not Used
Maintenance	Not Used	Not Used
Check Permissions	OID of object being accessed	Permissions
Writing Record	OID of object being written	Not Used
Writing OList	OID of object being written	Not Used
Delete Object	OID of object deleted	Not Used

The current User performing these actions can be determined by checking the UserName property of the global security object Security.

TSecurityEvents Type

Unit

DACMgr

Declaration

TSecurityEvents=(sLogOn,sPermissions,sPINChange,sMaintenance,sWrite,sOlist,sDelete);

Description

The TSecurityEvents type is used by security event handlers to determine the type of security event when a security event occurs.

TSecurityName Type

Unit

DACMgr

Declaration

TSecurityName=string[30];

Description

The TSecurityName type is a string of up to 30 characters. In combination with the PIN it uniquely identifies the users account.

TSecurityPIN Type

Unit

DACMgr

Declaration

TSecurityPIN=string[20];

Description

The PIN type is a string of upto 20 characters set by the user, in combination with the UserName it uniquely identifies a user account.

TSecuritySID Type

Unit

DACMgr

Declaration

TSecuritySID=string[50];

Description

The SID type is a string of up to 50 characters that uniquely identifies an object.

For User objects this is the PIN and Name, for Group objects it is the Name and for SecuredObject objects this is an identifier set by your application.

TPermission Type

Unit

DACMgr

Declaration

TPermission=0..7;

Description

TPermissions Type

Unit

DACMgr

Declaration

TPermissions= set of TPermission;{set of byte}

Description

TOIDRec Type

Unit

DACMgr

Declaration

```
TOIDRec =     record
                    OID1,OID2,OID3:byte;
                    case integer of
                    0:(OID4:byte);
                    1:(Permissions:TPermissions);
                    end;
              end;
```

Description

ESecurity Type

Unit

DACMgr

Declaration

```
ESecurity=class(Exception);
```

Description

Base class for all security exceptions declared below.

```
EInvalidDACRecord=class(ESecurity);
EInvalidDACTable=class(ESecurity);

EInvalidPIN= class(ESecurity);
ELogOnTriesExceeded=class(ESecurity);
ELogOnAbort=class(EAbort);
EPINChange=class(ESecurity);
```

IsSQLBased Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property IsSQLBased:boolean;

Description

This property indicates whether the underlying DACTable is of a SQL type.

If set to true then when attaching the DACManager a unified LogOn to the SQL Server and DACM will be generated.

This property is only applicable to SQL databases and LogOn will fail if used for non-SQL databases.

LogOnPrompt Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property LogOnPrompt: Boolean;

Description

This property determines wether the LogOn Dailog will be displayed to log a user on to the security system.

If set to false the Dialog is hidden and instead a LogOnEvent occurs.

Password Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property Password:String;

Description

The password property contains the Paradox password used to encrypt and protect the DAC Table.

When the DACManager attaches to the global security object this password will be used to open the Table.

Note: This property is not valid and is ignored if the Table is not of Paradox type.

MinPINLength Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property MinPINLength:integer;

Description

This property determines the minimum length of a PIN that is considered valid.

This can be used to enforce minimum security standards for your application. The longer the PIN the harder it is determine the PIN by chance but it is also harder for users to remember.

Key Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property Key:word;

Description

The Key property contains a number that is used in the encryption of the SID field in the DAC Table and to calculate the value of the CHK field.

This property can be used to ensure that your DAC Table cannot be read by another application developed using DACM.

Note: Unless you ensure that you always use the same value this will make the DAC Tables of different applications you develop incompatible with each other, this may be important if you wish to have multiple applications sharing a DAC Table.

AdminPINChange Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property AdminPINChange:Boolean;

Description

The AdminPINChange property determines wether the Users PIN is changed centrally by a member of the Administrators group (via the Maintenance Dialog) or by the user during the LogOn process.

If set to true the ChangePIN check box on the LogOn Dialog will be hidden. Also in the Maintenance Dialog on the Users page a Button will be displayed to allow an administrator to change the PIN of a selected User.

PINValidPeriod Property

Unit

DACMgr

Applies to

TDACManager

Declaration

Property PINValidPeriof:Integer;

Description

The PINValidPeriod property determines the number of days before the user is forced to change his/her PIN.

During LogOn this property is checked and if non-zero the value of the PXDate field in the Users security record is read. If the date is less than or equal to todays date then the PINChange Dialog is displayed and the User must enter a new PIN.

By default the value is zero and is then ignored by the global Security object.

Note: This property should be set to zero if you have chosen to administer PIN's centrally.