Table of Contents

© 2017 Industrisoftware AS

1 Introduction to Groupware

Hi,

After struggling with the PCSoft groupware, I finally abandon it, and decided to make my own.

In my current project (for a client) we want to restrict access, but managers should be able to press F11 log in as administrator and disable access to controls, then log inn as user again without restarting the app.
Or a manager needs to press the [Only Admins] button, and he should do that in the same way.

Now I want to share my cGroupware class with anybody that`s want it, as a part of "giving a little back" to this great community.

An small exedemo can be found Here... just unzip in a folder and run the executable file.

Info about the demo, it mainly containing the **Admin** and **User** users (and Admin and User groups), password is blank/Nothing on both.

F11 = Login/Change userid
F12 = Inspect screen (where you set or revoke access)

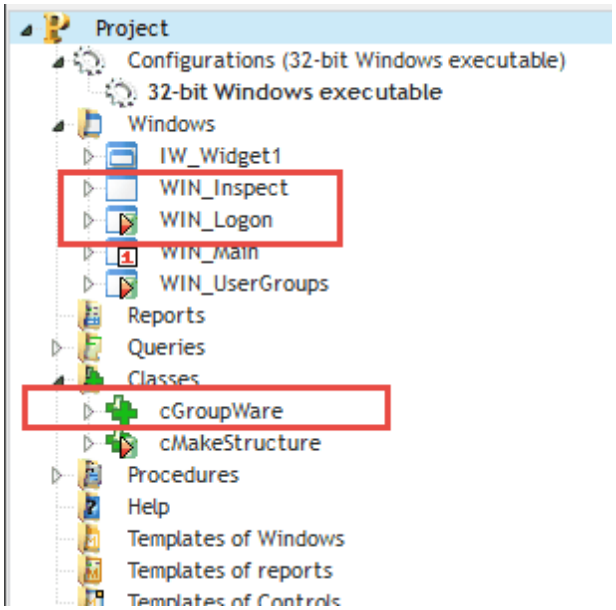
The only buttons that actually do something can be found in the system ribbon.

Might not be suitable for all, but perhaps good starting point if you want to write your own.

© 2017 Industrisoftware AS

Features:

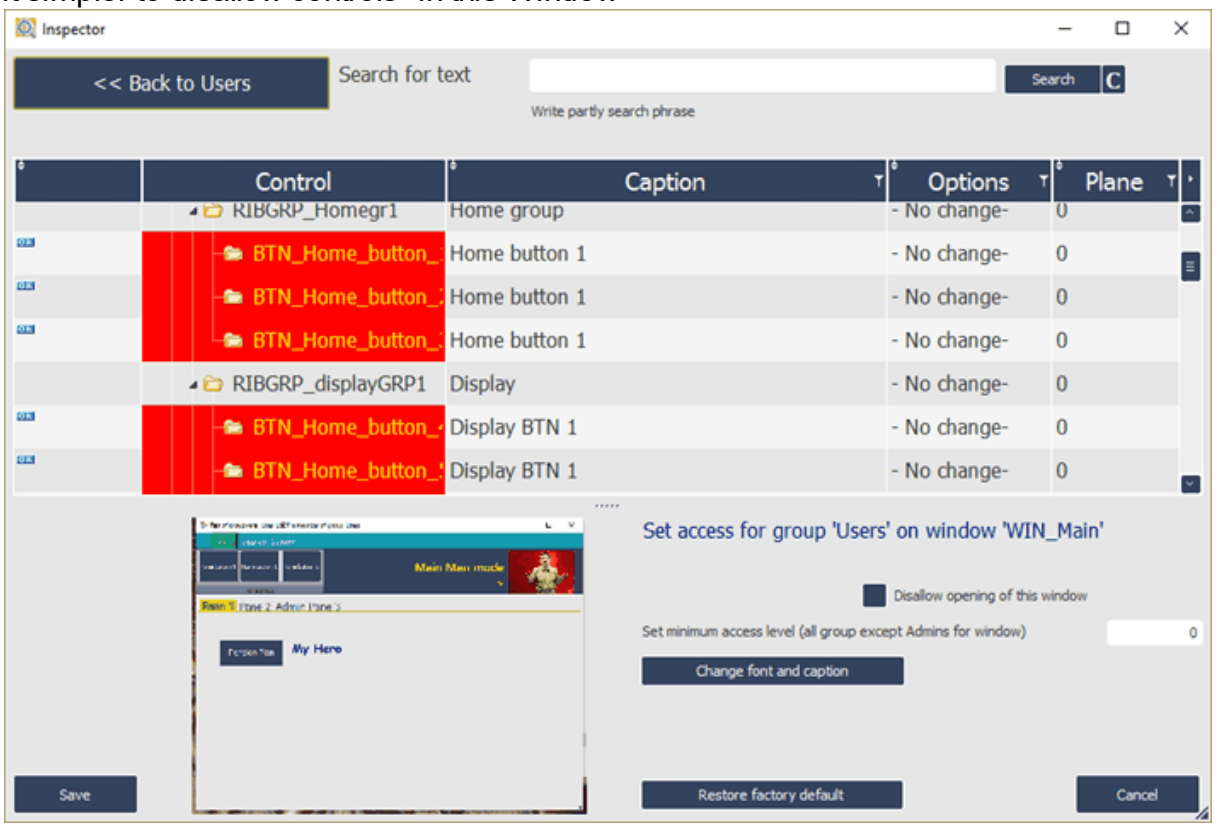
1. One class,One table in the analyze and 2 Windows (Win_Logon,Win_inspect) makes the whole thing work.



2. You can change userID run-time, no need to restart app to change userID and rights

© 2017 Industrisoftware AS

3. The Inspect screen/Window reads only controls for the window your in, not all - making it simpler to disallow controls "in this Window"

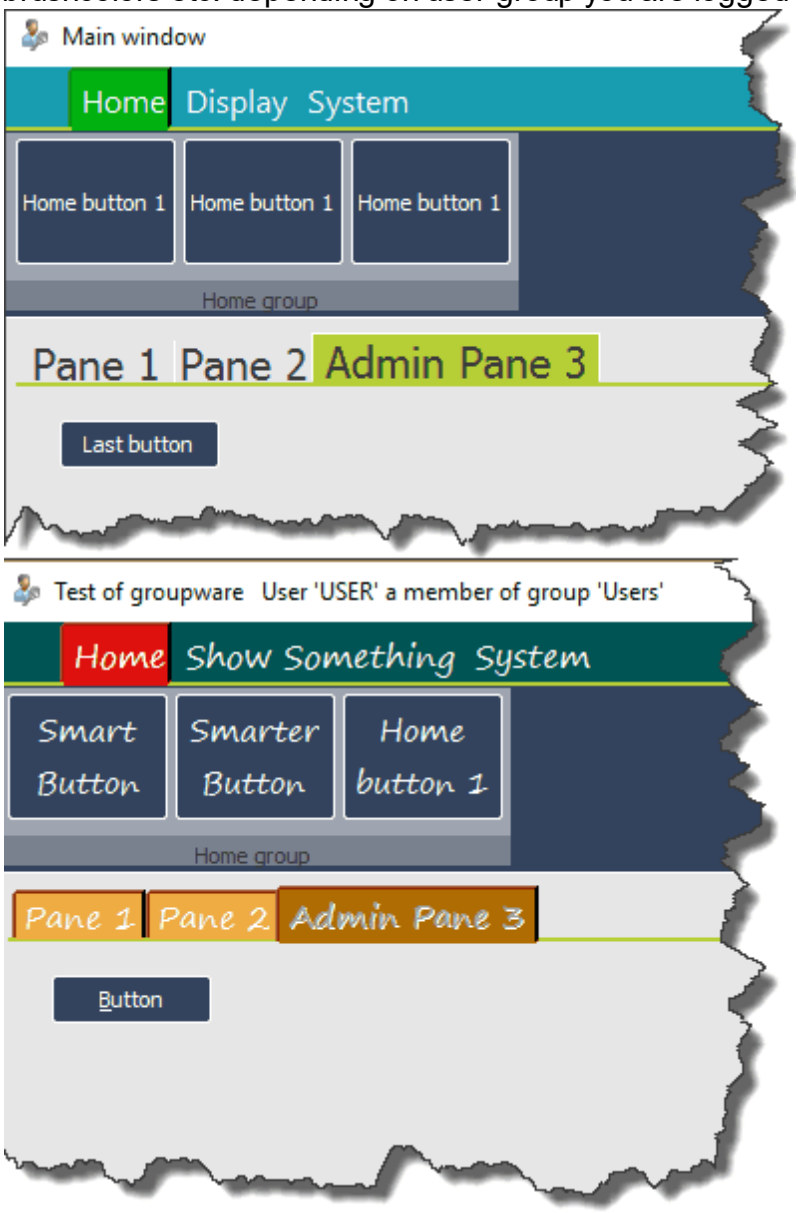


4. Farley easy implementation with your app and if you have existing usergroup/Users tables.
(I will/can write a short PDF Document describing how, if anybody is interested)

5. Written in WinDev 20, I`m on v22 and are using it here, so nothing is problematic about upgrading.
(WinDev 20 is the lowest Window version I have on this PC for the moment)

© 2017 Industrisoftware AS

6. Since WinDev v22 lets you edit a screen I made it possible to change fonts, brushcolors etc. depending on user-group you are logged into.



7. Timed auto-logout or auto-logon as "other" user (perhaps with lower rights)

Some points/limitations of my groupware:

1. Unlike PCSoft`s solution a userid must exist in a group, and the group is actually what you set the rights for.
2. A Userid must be unique. You can not have same userid but in different groups.
3. When pressing F-something (Demo app uses F12) to get the inspect window the

© 2017 Industrisoftware AS

controls are read run-time.
4. It has bugs - Yes I also do bugs, not on purpose, but still.
5. I tried to keep the code clean, but some might not agree, I however personally hate statements like:

```
MySelf..Caption=(MySelf..Pushed?"Trace ON" ELSE "Trace OFF")
```

Not easy to see whats going on here huh?,... having said that, I copied the example from my own code but it`s just one place and in the test-project, not the groupware.

If you do use it, and correct bugs or make smart changes, I would like to get a copy.

Anyhow enough said, here is the link to the Project

© 2017 Industrisoftware AS

1.1 The cGroupware class

I made cGroupware a global class, since I did not see any situation where several cgroupware`s instanses beeing active at the same time inside an application.

You can more or less import the cGroupware class in a existing project, import the 2 windows and the Groupware table and off you go.

If you want to use your own usergroup table and usertable there are some tecniques to do just that and I will conver this a bit later in the document.

© 2017 Industrisoftware AS

1.1.1 Nuts and bolts of cGroupware

When a user with sufficient rights press ALT-F12 (default button for the inspect-window) cGroupware try to read all controls in "this" window run-time.

Now after opening the inspect window, a administrator can change controls from a treestructure of all controls found at the moment.

Supplied are the means to make them invisible or grayed, the third option "no change" leaves the control untouched.

More or less the whole cGroupware is made up array of structures, for some reason I just love them.

In my opinion they are extremly fast, easy to cope with, and so fort.

If you study the code you offen see me looping through the arrays with the code

```
for each ds of ::Arrayname
       //Now DS have the structure and values for each "record"
       //if the array was Changedcontrols the rest migh look like:
       IF ds.sGroupName <> ::ME.sCroupName
           Continue
       END
end
```

Note: the DS name can be anything as it`s defined at that point, but I seem to have been stuck with DS (Data Set), and use the letters "DS" each time I loop through an array.

Another point to make, I sometimes copy from one identical structure to another, in this project a discovered a smarter way than assigning each structure member, and that is some thing like:

```
n is int
For each ds of ::Arrayname
       if ds.field = this or that
              n = add(SecondArray)
              // copies value of the "ds" structure into
              // second arrays newly created "record"
              SecondArray[n] <= ds
       end
end
```

© 2017 Industrisoftware AS

Example from code:

```
i is int = Seek(::TouchedCtrl,asLinear,"sControlName;sPare
IF i = -1
    i  = Add(::TouchedCtrl)
    ::TouchedCtrl[i] <= ds //::ChangedControls[ds.nr]
    ::TouchedCtrl[i].nr = i
END
```

Perhaps you already know all this, but I wanted to share it anyhow.

After a successfull Logon a ::ME structure is set containing information about the user:

```
30  STThisUser is Structure
31      sUsername is string
32      sPassword is ANSI string
33      sGroupname is string
34      nAcceslevel is int
35      sRecID is string
36      sRecIDGroup is string
37  END
```

This cGroupware::ME structure, you can check during program execution to se if a user have the sufficient rights to run part of your code.

A (*stupid*) Example is from the "Calculate Pension" button in the GW20 project:

```
SWITCH Upper(cGroupWare::ME.sGroupname)
    CASE "USERS"
        Info("Your pension is calculated to 0.12")
    CASE "ADMIN"
        Info("Your pension is calculated to 1 000 000")
    OTHER CASE
        Info("Forget it")
END
```

But you got the meaning of it :)

© 2017 Industrisoftware AS

cGroupware hold's all controls that has a different status than their inital value, in a changedcontrols array (of structures), the information beeing stored about a control and only if it`s changed.

```
62  STChangedControls is Structure
63      sControlName is string
64      sParentName is string
65      sGroupName is string
66      nType is int
67      nOption is int
68      bchanged is boolean
69      sRecID is string
70      nr is int
71  END
```

A second array of structures is contains reset information, case a logon with a new user. If this user is allowed to see a hided control that the previous user was not we need to unhide this control again.

So I made a Touchedcontrol array of the same type, just to unhide or make active a control, before loop of setting the control is executed.

Mostly for WinDev 22 where we can invoke the built in "Customize the interface" utility, after testing this I soon wanted to increase size and perhaps change the font of a control, so I built in a way to do just that.

Offcourse you, beeing a WinDev developer and all, can expand this to other properties of a control to.

Some of the methods for changing controlstates is:

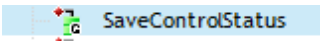
ChangeControlsNow - resetts, then set state,font etc. of changed controls.

GetChangedControls - Returns a <CR> spearated list (a string) of controls changed "This" session, unless you do a cgroupware::GetChangedControlls("*") that will return all changed controls

GetControlType - Returns the type of control as string/Text, input is controltype as

© 2017 Industrisoftware AS

integer. Calling cGroupware::GetControltype(4) returns the string "Button (4)"



- Saves a "changed control" in inspect window" state of a control (Hide, grayed)

The Windows opened of cGroupware is stored inside an ::Wins array (of structures) that is used to check if a user actually can open a window.

```
STWindows is Structure
    sName is string
    sGroupName is string
    nAccesslevel is int
    bCanOpen is boolean
    bchanged is boolean
END
```

Now the obvius part of the bCanOpen value that is True/False, but I also inclueded nAcceslevel the rule for NaccessLeve is: if not **1** (Admin) the user in a group must have equal or higher nAccesslevel to be able to open a window, placing it somwhere between a user with no rights and admin with all rights.

Now this is where your own develoment comes in place if you want a groups of windows suddenly available or unavaliabe for a usergroup(s) you can change the nAccesslevel to a higher value.

The Method cGroupware::AddSkipWin(mywindow..name) signals to cGroupware that everybody has access to this window.

© 2017 Industrisoftware AS

1.1.2 Applying groupware into your project

To implement cGroupware into your application the following criteria must be met:

1. A way to store changes, so we need a table to store this in (Groupware)

Note: For testing purposes you can skip table this and cgroupware is reset each time you start your application.
The groupware table is however supplied with the GW20 project.

2. Some means to change the state of controls, users and groups, supplied for this is the Win_Inspect window that has 6 planes.

3. Possibillity to Log on and change logon, supplied for this is the Win_Logon window.

4. The global cGroupware class offcourse. It`s made global but you need to define it once, to run the constructor code (or move the code into a ::Start method if you prefere)

Now I put most of the code into the global part of the project, but most of it can be anywhyere if you for example wants to show the main window and then the Logon screen.

© 2017 Industrisoftware AS

First, every window that is going to be used with cGroupware needs this line of code:

End of initialization of **WIN_Main**

```
1
2
3   ////Making structure vars of table
4   ////Commented 2 idents in, is if you want to make a structure
5   //MakeFL is object cMakeStructure
6   ////IF HExecuteQuery(qry_groups) = True
7   //  MakeFL.MakeStructure(Groups)
8   ////ELSE
9   ////    Info(HErrorInfo())
10  ////END
11  //EndProgram()
12
13  cGroupWare::CanOpen(mywindow..name)
14
```

Since you can logon when your app is already executing you should also have a line detecting a user-change (can also be done in the "USERLOGIN" event)

Focus gain (WM_SETFOCUS) of **WIN_Main**

```
1
2   cGroupWare::ReCheck(MyWindow..Name)
```

Or here

Local Procedure **Handle_userchange** * | If Error: by program | When Exception: by program

```
1  PROCEDURE Handle_userchange()
2
3  WIN_Main..Title = "Test of groupware  User '"+cGroupWare::ME.sUsername+"' a member of group '"+cGroupWare::ME.sGroupname+"'"
4  cGroupWare::ReCheck(MyWindow..Name)
5
```

In the above example the code below needs to be inserted.

```
//Demonstrate callback from groupware
cGroupWare::NotifyME = Handle(MyWindow)
Event(Handle_userchange,"*.*","USERLOGIN")
```

© 2017 Industrisoftware AS

Below are some code examples from the GW20 project and a brief explanation.

```
EXTERN "WinConst.WL"
EXTERN "KeyConst.WL"
```

cGroupware uses them, but I always include these to "inlude lines" in my projects.

Say you do not offer groupware with you app since it`s no need, you can still hide some "***Service only***" options from the normal user and ship the Groupware table along with you app.

But what if the user Deletes the groupware table? - cGroupware will automaticly make ADMIN and USER group if no groups/users not exists with a initial password of a different type than shipped. (if you don`t remove the code).

Example of different password than the "shipped in groupware table" :

```
//Empty / Clean system password (or if groupware DB file is deleted)
cGroupWare::EmptyOrFirstTimePW = "CustomerDontKnowThisPassword"
```

I guess you already use Trace in some form and cgroupware uses it to, I made an internal Trace that can bw switched On/Off so you dont have to look at groupware messages unless working with it.

```
//Debug window / Trace from the cgroupware class
cGroupWare::DebugMsg = True
```

Note: The ::DebugMsg state can off course be saved into a inifile (or what ever)

if you want to log the debugmessages into a text file, you can uncomment as shown below:

```
//You can also log debug messages to file eaven if debugmsg is switched off
cGroupWare::DebugLogFolder = "C:\deleteme"
cGroupWare::DebugToFile = True
```

If you want, you can switch off all the cGroupware code with the statement:

```
//Test without gropware
cGroupWare::Off = True
```

© 2017 Industrisoftware AS

You can off course build your own logon screen or use your existing logon screen, just specify what window you want to be called when logging in with the code:

```
//Optional
cGroupWare::LogonWindow = "WIN_Logon"
```

Make shure to study existing win_logon code, where the most importent step is the call to:

```
IF cGroupWare::CheckUserIDandPassword(EDT_UserID,EDT_Password) = False THEN
END
```

That is the actual logon setting the ::ME structure that will be checked against from methods in the cGroupware.

You also have the possibillity to se Auto-logout time (in minutes), after the time specifyed is run the logon screen Pops-up again:

```
//Auto-Logout
cGroupWare::AutoLogoutTimeOut = 10
```

Now for the "Service-mode" I described earlyer, cGroupware has the possibillity to de-elevate rights in stead of reactivating the logon-screen.
If for example you where connected to a client via Teamviewer, VNC or something and forgot to logon as a user again after being Admin for a while:

```
//If you want to de-elevate rights instead of user re-logging on
cGroupWare::sAutoLogonToUser      = "User"
cGroupWare::sAutoLogonToPw        = ""
```

Now in this "Service mode" we want the "User" to be active (without login) as we start the application and need to specify this at startup.

```
//Auto-logon logon
IF cGroupWare::CheckUserIDandPassword("User","") = False THEN
       cGroupWare::Logon()
END
```

This code will prevent the Login screen (Win_Logon) to appear, and the app is in "User mode" with the supplied rights yu specifyed at design time, and shipped to the client in the Groupware table.

Default in the cGroupware **Alt-F12** is the Inspect window and **Alt-F11** is the runtime logon screen, you can off course change this to what ever keys suits you:

```
//Groupware Optional options
cGroupWare::Alternative_InspectKey       = VK_F12
cGroupWare::Alternative_LogonKey         = VK_F11
```

In the GW20 project I use **F11** for logon and **F12** for inspect screen.

© 2017 Industrisoftware AS

Windows that are to be left untouced can be assigned especially for this with the:

```
cGroupWare::AddSkipWin(MyWindow..Name)
```

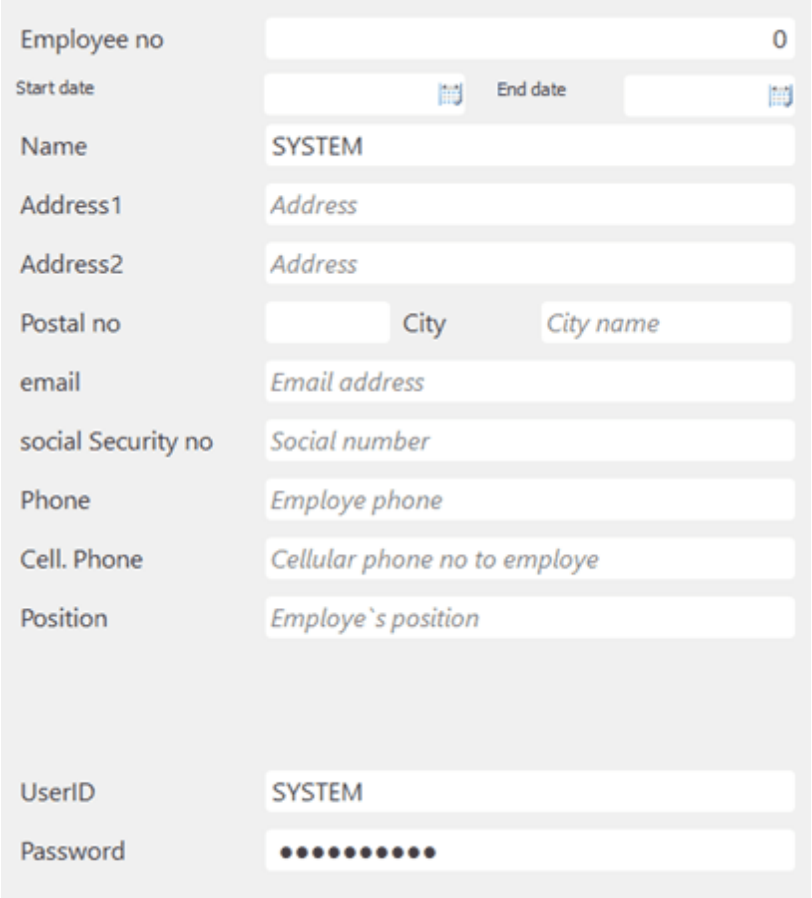
method.

© 2017 Industrisoftware AS

1.1.3 How I quick-implemented gw in a couple of projects

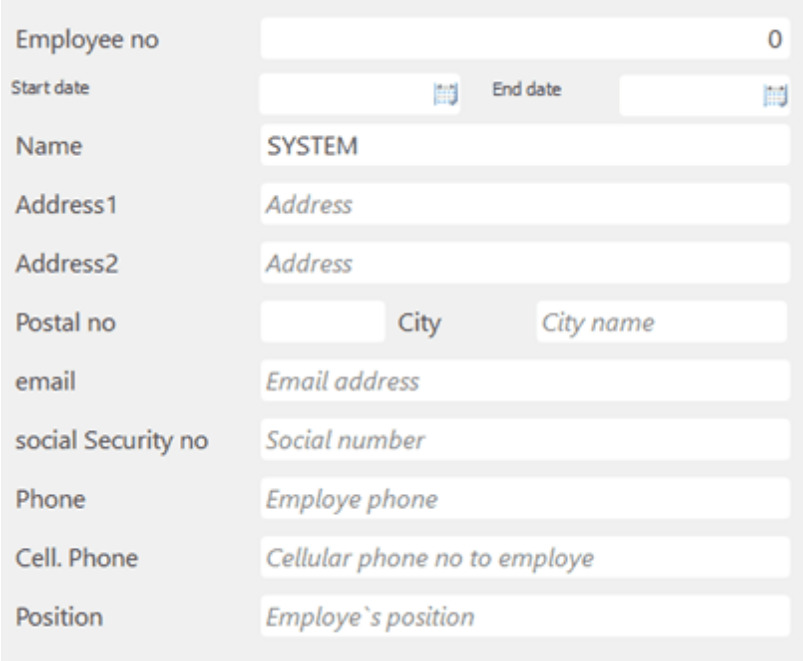
After I finished groupware I had to implement it in a couple of my own projects, and while doing this I discovered a quick and dirty way of implementing it with my projects that I would like to share.

In my project I have a emplyee table with a couple of fields named username and password:



Changed to:

© 2017 Industrisoftware AS



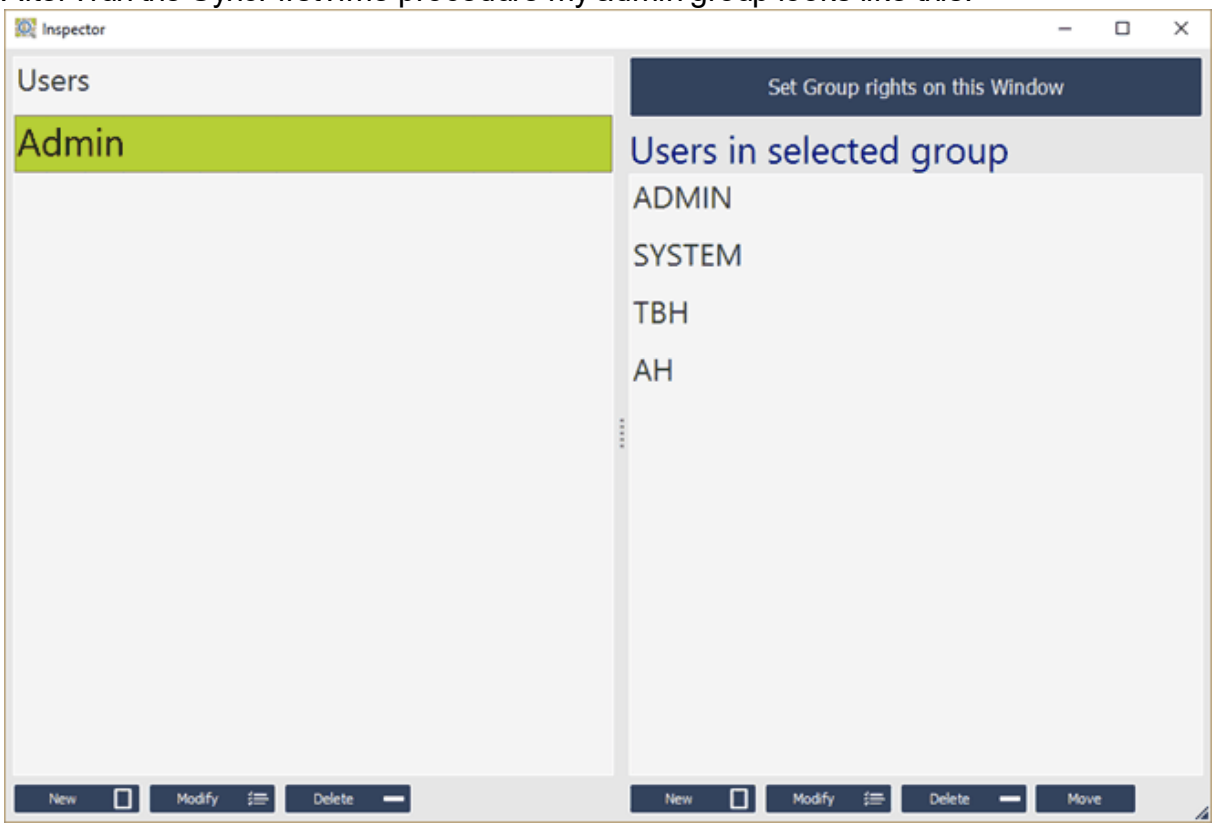
As you see I removed (from screen) the UserID and Password fields.

Then I made a small syncronize one-time routine, putting all users into the Admin group, this sync. routine can offcourse be more advanced, but it`s easy to move multiple selected users to another group in cgroupware so I was Lazy.

```
PROCEDURE SyncFirstTime()

FOR EACH Ansatt //is Norwegian for employee
       IF Ansatt.BrukerID <> "" THEN
              IF cGroupWare::CheckUserExist(Ansatt.BrukerID) = False THEN
                     IF cGroupWare::CheckGroupExist("Admin") = True THEN

cGroupWare::AddUser(Ansatt.BrukerID,Ansatt.Passord,"Admin",Ansatt.PRIMARYKEY)
                     END
              END
       END
END


END:
WriteKey("SYNCFIRSTTIMEDONE",True)
```

© 2017 Industrisoftware AS

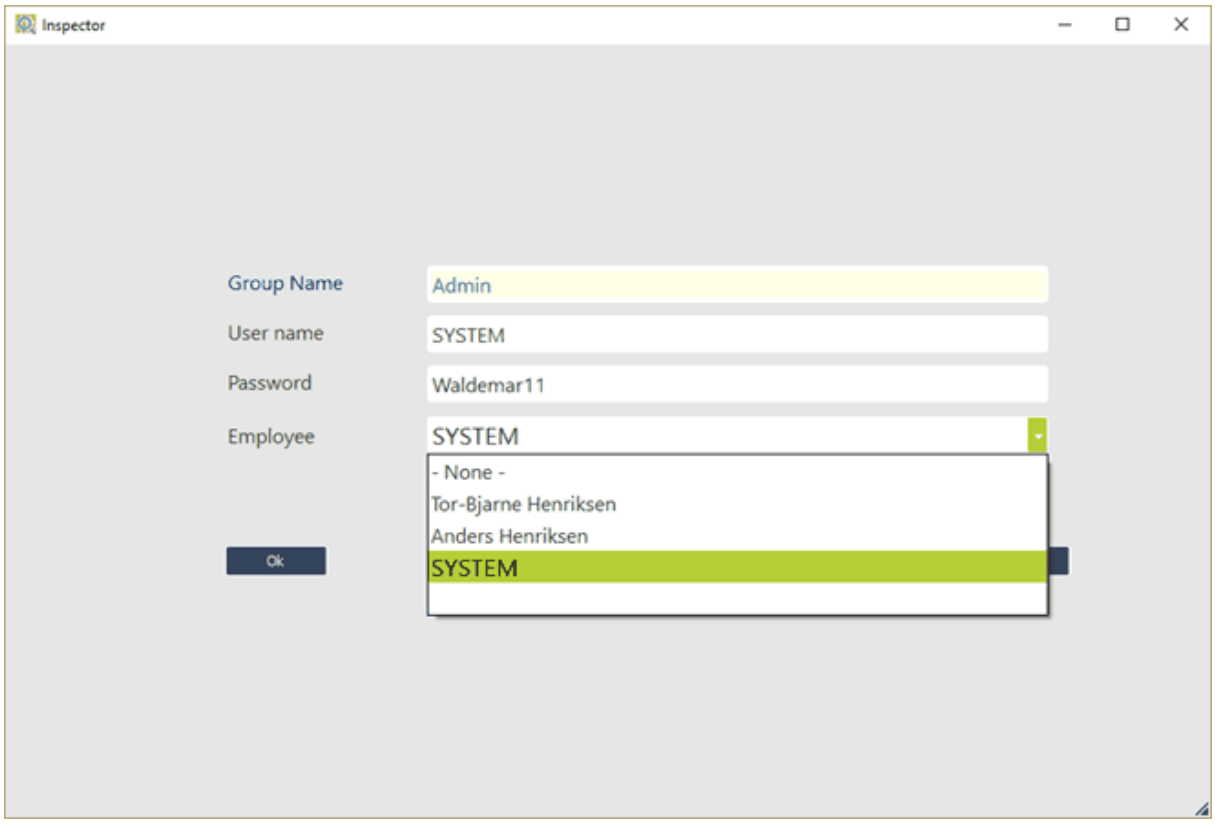
After I run the SyncFirstTime procedure my admin group looks like this:



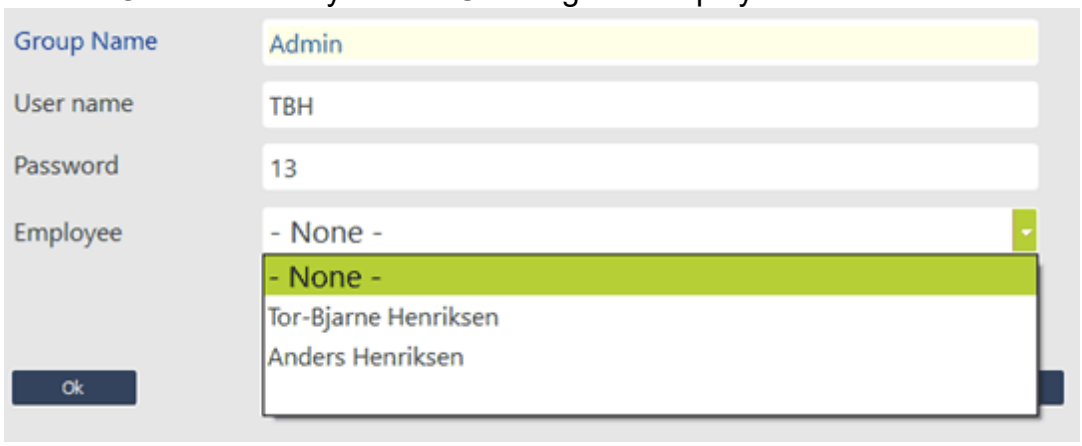
Some new users has appeard (SYSTEM,TBH, and AH)

© 2017 Industrisoftware AS

Then I made a combo in the "Modyfy users" pane where I can link my Employe table with the Groupware:



And the Combo box only shows "Un-assigned" employees:



(Since user "System" is assigned to employe "System" it is not shown in the combo for username "TBH")

© 2017 Industrisoftware AS

The code for building the combo looks like this:

```
PROCEDURE build_combo(sShouldInclude is string="")
s is string = "- None -" + TAB + gLink("") + CR
n is int
bIncluded is boolean
sNvn is string
IF HExecuteQuery(QRY_Get_Ansatt,hDefault) = True THEN
	FOR EACH QRY_Get_Ansatt
		n = Seek(cGroupWare::Users,asLinear,"sRecID",QRY_Get_Ansatt.PRIMARYKEY)
		IF n = -1 THEN
			s = s + QRY_Get_Ansatt.Navn + TAB +
gLink(QRY_Get_Ansatt.PRIMARYKEY) + CR
		END
		IF sShouldInclude = QRY_Get_Ansatt.PRIMARYKEY THEN
			sNvn = QRY_Get_Ansatt.Navn
			bIncluded = True
		END
	END
END
IF bIncluded = True THEN
	s = s + sNvn + TAB + gLink(sShouldInclude) + CR
END
ListDeleteAll(COMBO_Emploee)
ListAdd(COMBO_Emploee,s)
```

Note: I use the Glink of the combo box to hold the RecordID of my employee table

Then I only make some small changes to the BTN_Modify1 code of the groupware:

```
Click BTN_Modify1                    If Error: by program When Exception: by program
1   IF LIST_UsersInAGroup = -1 THEN
2       Error("Please select a user first.")
3   ELSE
4       IF cGroupWare::ME.nAccesslevel = 1 THEN
5           gw.nUserAction  = 2
6           EDT_Group_Name1 = LIST_Usergroups
7           EDT_User_name   = LIST_UsersInAGroup
8           EDT_Password    = cGroupWare::GetPassword(LIST_UsersInAGroup)
9           gsOldUID        = LIST_UsersInAGroup
10          i is int = Seek(cGroupWare::Users,asLinear,"sUserName",EDT_User_name)
11          IF i <> -1 THEN
12              build_combo(cGroupWare::Users[i].sRecID)
13              COMBO_Emploee   = cGroupWare::Users[i].sRecID
14          END
15          MyWindow..Plane = 4
16      ELSE
17          Error("You belong in a User group that are not allowed to change users and groups")
18      END
19  END
20
```

© 2017 Industrisoftware AS

And passing any RecordID of an employe to the groupware:

```
Click BTN_Ok1        If Error: by program When Exception: by program
SWITCH gw.nUserAction
    CASE 1
        n is int = Seek(cGroupWare::Users,asLinear,"sUsername",Upper(EDT_User_name))
        IF n <> -1 THEN
            Error("User names needs to be unique.")
            RETURN
        END
        cGroupWare::AddUser(EDT_User_name,EDT_Password,EDT_Group_Name1,COMBO_Emploee)
    CASE 2
        cGroupWare::ChangeExistingUser(EDT_User_name,EDT_Password,EDT_Group_Name1,COMBO_Emploee,Upper(gsOldUID))
    OTHER CASE
        Error("Unknown?")
END
 MyWindow..Plane = 1
 Refresh()
```

Thats it, now you could write some housekeeping rotines if an employe record get`s deleted (remove it from the groupware), I went for just removing the RecordID of the user but you can remove the user from the groupware if you want:

```
n is int = Seek(cGroupWare::Users,asLinear,"sRecID",Ansatt.PRIMARYKEY)
IF n <> -1 THEN
    cGroupWare::Users[n].sRecID = "" //Clearing any recordid pointing to this record.
END
H.Erase(Ansatt)
TableDisplay(TABLE_Ansatt,taReExecuteQuery)
```

If you wanted to delete the user from the groupware you would have to replace the line :

```
cGroupWare::Users[n].sRecID = "" //Clearing any recordid pointing to this record.
```

with:

```
ArrayDelete(cGroupWare::Users,n)
```

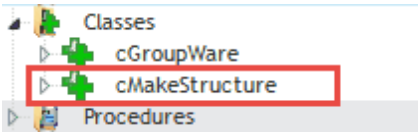
© 2017 Industrisoftware AS

1.1.4 Using your own Group and Usertables method 1

If you have (as me) a project with an easy logon and perhaps a emploee table with a field named Login and password but a lot of other fields, heres how you can implement Groupware using them:

First of all - you could do a search and Replace in the cGroupware class itself, but I would suggest you to rename or create a field in you emploee table named sUsername and sPassword as WinDev will change the exisitng table structure and propose to change in you code as well.

With the GW20 project is a small hlper class called cMakestructure, this is designed to make Structure entries of a table or query.
I supplied 2 tables (Login and groups)



Now if you activate the code as shown below:

```
//Making structure vars of table
//Commented 2 idents in, is if you want to make a structure of a query
MakeFL is object cMakeStructure
//IF HExecuteQuery(qry_groups) = True
       MakeFL.MakeStructure(Groups)
//ELSE
//     Info(HErrorInfo())
//END
EndProgram()
```

it will make something like this in the clipboard buffer for the croups table:

```
GroupsID is numeric
GroupName is string
Description is string
GroupLevel is string
```

© 2017 Industrisoftware AS

Now it`s easy to extend the structure of cGroupware::Groups to include some more fields, remember to delete any duplicates.

```
STUserGroup is Structure
    sGroupName is string
    nAccesslevel is int
    bInspectorwindow is boolean
    bchanged is boolean
    sRecID is string
    //If I wanted to use FiletoArray(::users,groups) i would need more variables matching the file structure of login table
    //The small helperclass cMakestructure is designed for this, se commented code in loading of win_main
    GroupsID is numeric
    GroupName is string
    Description is string
    GroupLevel is string

END
```

Now in the ::LoadArray method I would comment out the loading of ::Groups array and in stead do a `FileToArray(cGroupWare::Groups,Groups)`

Since, in my example I have not changed the Groupname in the table to sGroupname I also must do the step below (when loading)

```
FOR EACH ds OF cGroupWare::Groups
        ds.sGroupName = ds.GroupName
END
```

So I set the empty sGroupName to the value of the field GroupName from the table Group.

Any Changes mode from within the inspect screen to be saved I would use the event routines called from cGroupware, commenting out the portion that saves Groupware::groups

© 2017 Industrisoftware AS

1.1.5 Using your own Group and Usertables method 2

Supplied is code to syncronize 2 existing tables with cGroupware, activate the test by removing the commented code:

```
//Demonstrates one way to sync "real" tables with array
Event(SyncUsers,"*.*","USERSCHANGED")
Event(SyncUsers,"*.*","GROUPSCHANGED")

Event(DeleteGroup,"*.*","GROUPSDELETED")
Event(DeleteUSER,"*.*","USERSDELETED")
```

Also I call the SyncUsers after "End of init win_main".

```
SyncUsers() // Sync "other" tables to groupware
```

I did now hovever make a Guid to the two tables (Login and Grops) but to se already exdisting values synced into cGroupware I suggest you use the WDMap of HFSQL Controlcenter to insert test values.

© 2017 Industrisoftware AS

1.1.6 Known issues

I would like to deliver a "Perfect packaged" product, but my time on cGroupware has ended and I need to focus on the ToDo list of my project.

Within the time avaliable I was not able to resolve the following issues:

1. Menus - I do not use them (agree or not), so there`s no code handling them.

2. When setting the Brushcolor the second time of a control, I try to get the stored value and set the buttons on plane 6 accordingly, but the color seems to be offset, and it have to be reselected.

My guess is probably since I use a Color variable, perhaps some default settings in Hue or something, or a bug, I do not know.

3. Dynamic tabls, Dashboard controls - the tree list shoud list the dasboard controls but the activated internal windows are below the main parent not the dashboard.

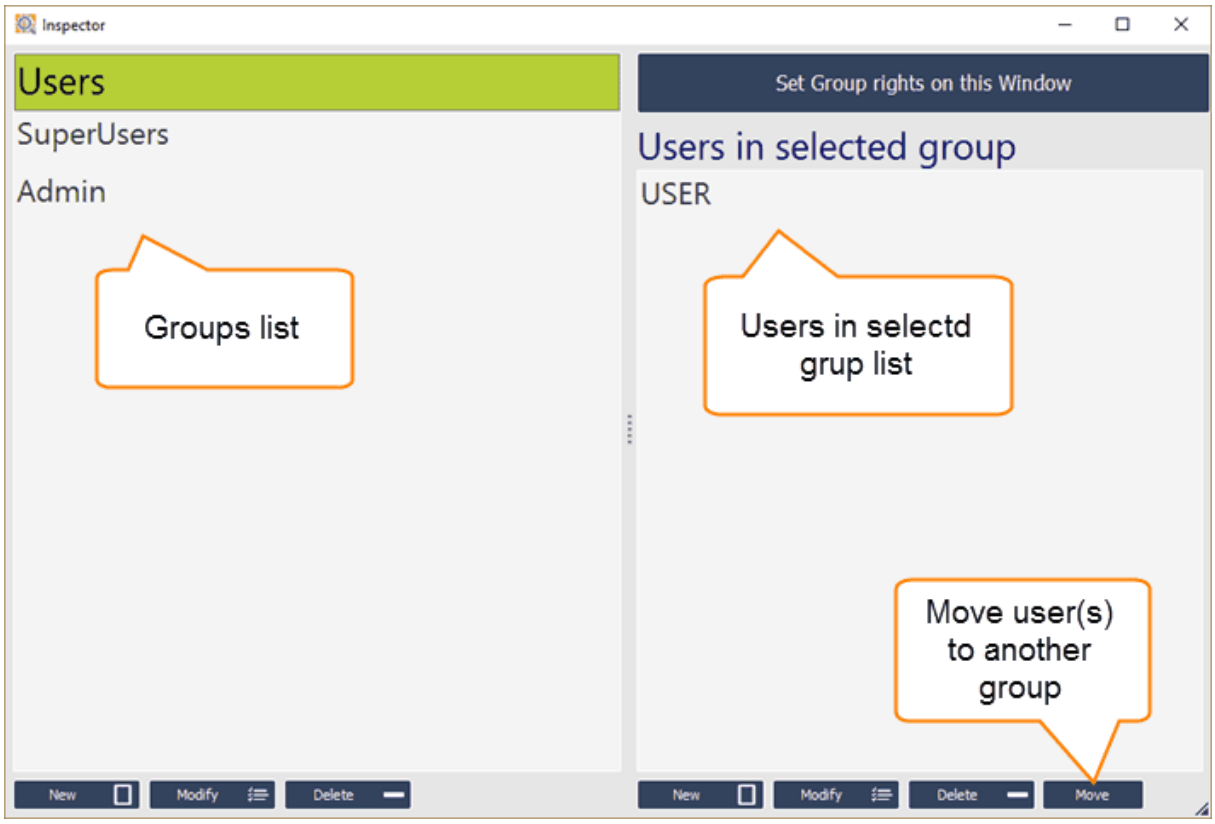
© 2017 Industrisoftware AS

1.2 Win_Inspect

A quick note, you can terminate Win_Inspect with the **ESC** - key.

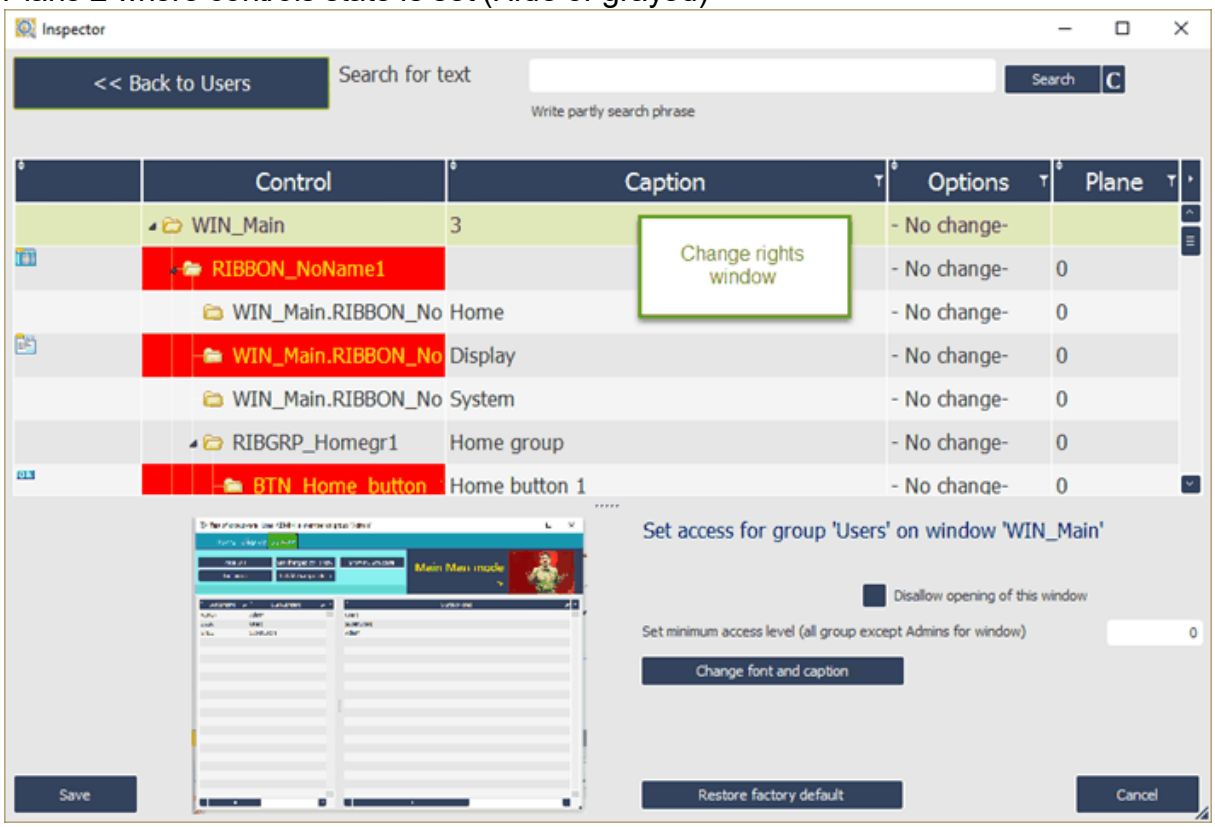
Some pictures with comments on what the diffrent areas is:

Plane 1 main.



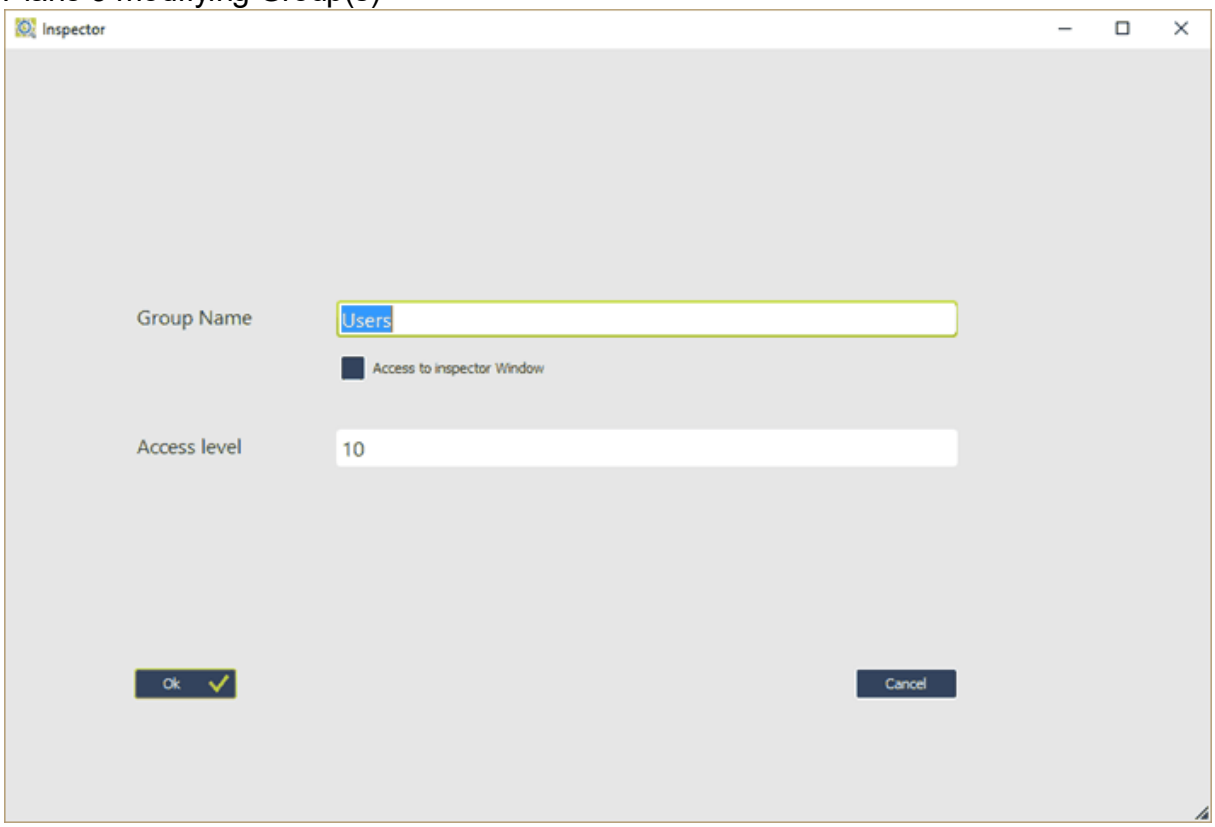
© 2017 Industrisoftware AS

Plane 2 where controls state is set (Hide or grayed)



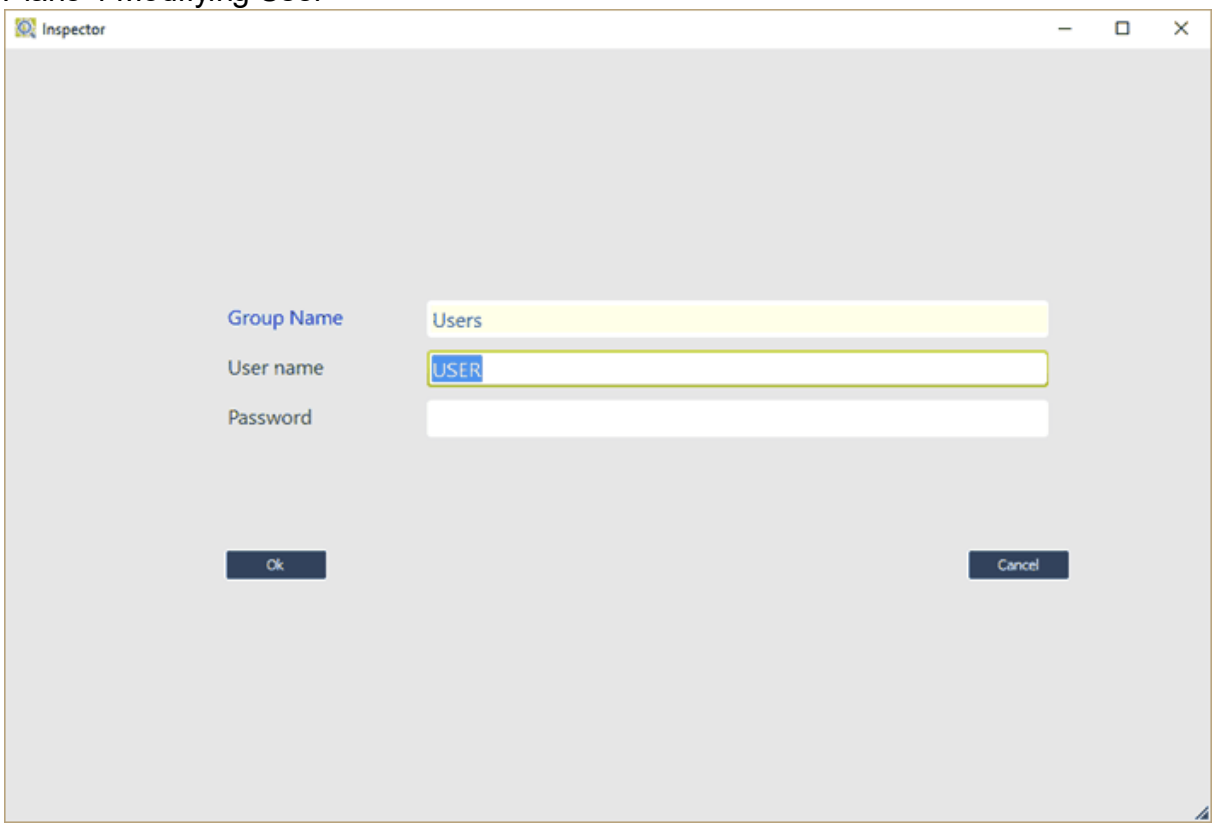
© 2017 Industrisoftware AS

Plane 3 modifying Group(s)



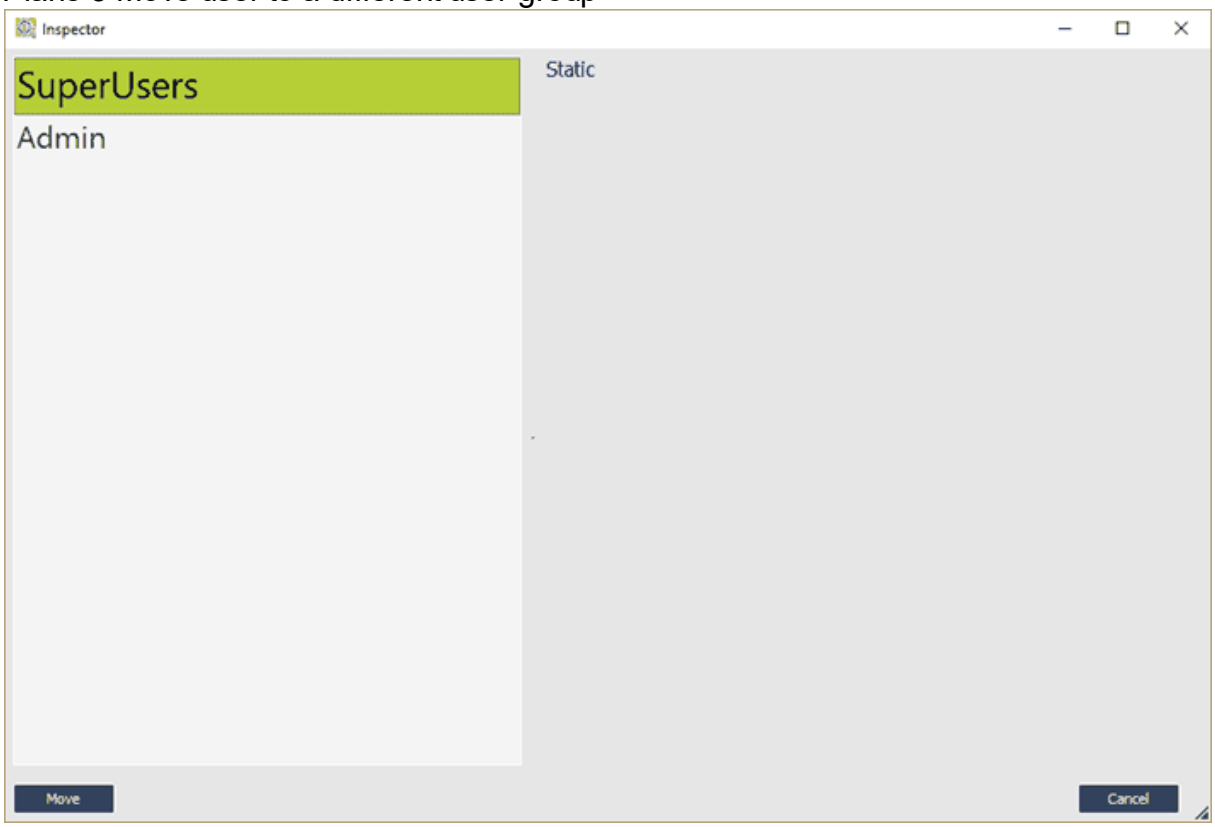
© 2017 Industrisoftware AS

Plane 4 Modifying User



© 2017 Industrisoftware AS

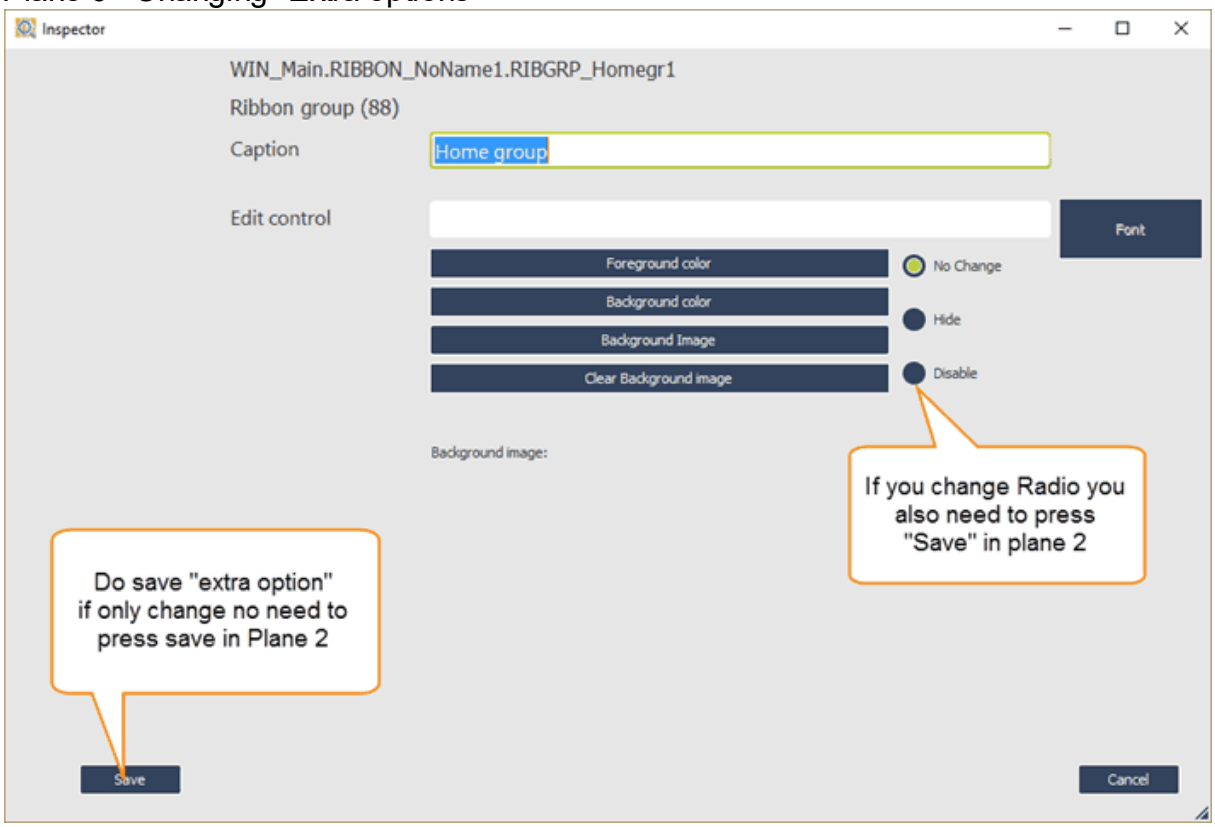
Plane 5 Move user to a different user-group



Note to self, the static used to have a value "moving user"+user+"from "+List_groups"

© 2017 Industrisoftware AS

Plane 6 - Changing "Extra options"



© 2017 Industrisoftware AS

1.3 Win_Logon

As mentioned you can use what ever window you`d like, in my own "***FlyFish project***" - (a Air freight system for fish).

When the user press F11 (change user) I temporarly change the main..plane to one with a "*FlyFish*" Theme

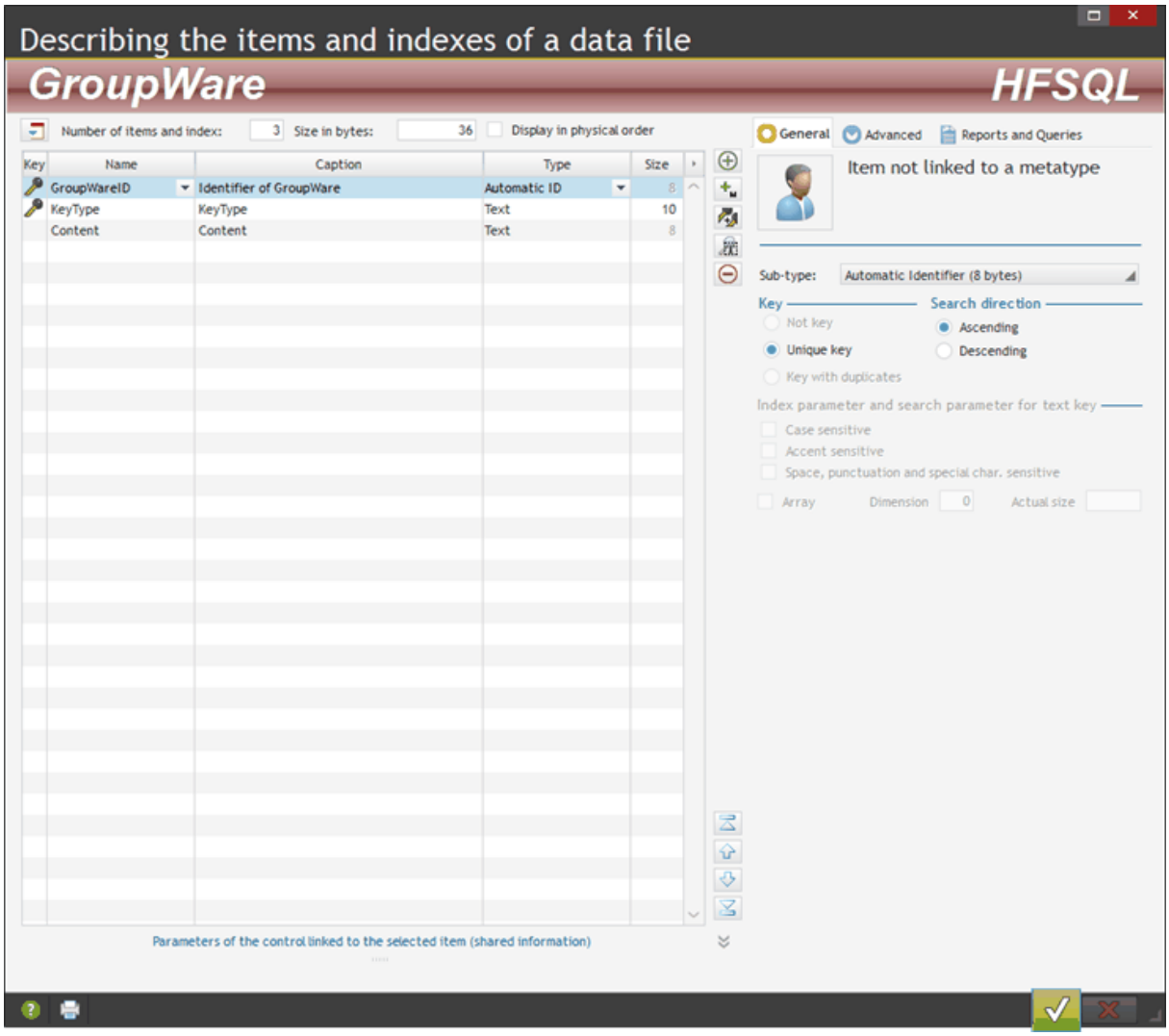
And since it runs on "Terminals" I utilize a modified version of WD_ROLL / Alphanum keyboard.

*Yes, it`s a laidy (and some flying fishes) not sure if it`s **political correct** , and I might have to remove this when deliver to client, I will off course blame the impact of PC-Soft`s brochures over the years.*

© 2017 Industrisoftware AS

1.4 Groupware table

As I save the whole array in a Memo field (*Content*), there`s only 3 fields in the groupware table:



© 2017 Industrisoftware AS

© 2017 Industrisoftware AS