

DEVELOP SOFTWARE WITH POWERBUILDER

Train with ravi

BY RAVI KUMAR PARAMKUSAM

Contents

Forword	iii
Preface	v
Acknowledgements	vii
Chapter 1	
Introduction	1
Development of Master windows	3
Chapter 2	65
Chapter 3	91
Chapter 4	96
Chapter 5	97
Chapter 6	98
Epilogue	100
About	102

Foreword

This book on software development is meant for beginners, who want to learn software coding on PowerBuilder. There are techniques to develop screens, with the detailed use of the screens and their place in the software. If anyone is interested to have the source code they may contact on these email ids:

ravi_kp_com@yahoo.com or
ravikiransofttech@gmail.com

My Books website: <https://ravikiran.azurewebsites.net/>

All my books links are available in the website.

Also free pdf copies available.

You are welcome to contact me for any kind of support.

Write to me @ ravi_kp_com@yahoo.com for a quicker reply.

Ravi Kumar Paramkusam

Software Professional and Writer

Date 10th June 2023

Preface

In my 30+ years of experience I have Developed many software for many companies in different technologies and different domains. I have worked extensively with PowerBuilder and Visual Studio developing Applications and web applications. This book is about how to start with PowerBuilder. It is assumed that you have some basic knowledge of using PowerBuilder.

Ravi Kumar Paramkusam

Date 24th May 2023

Acknowledgements

This is to acknowledge the developers who have worked with me all these years.

This book is dedicated to all the companies and developers, I worked for and developed software.

Chapter 1

Introduction

PowerBuilder is a GUI-based software development tool that follows all OOPS concepts.

Inheritance is the main concept in developing in 'PB' in short it is called, which reduces the development time.

PB is an evolving software. From v5 in 1998 now after releasing v12.5, they started naming versions by the year. The latest version is 2022 release 2Beta.

Some say "Thousands of PowerBuilder applications still run today. Even though the platform is supposedly supported, its development has been slow in the past decade. It has changed hands multiple times which has caused the technology to become less desirable and lacking basic functionality from more modern platforms."

But it is not true that it is lacking basic functionality, it is a robust program and 'datawindows' is the best functionality that exists on any platform. Datawindows allows to retrieve the data from tables and allows it to update the tables with just one command. It can recognize whether the data is newly updated, modified, or deleted and act accordingly.

PowerBuilder is compatible with all the modern databases. It comes preloaded with most of the database drivers. Others can be downloaded from their respective websites.

It has the latest functionality to update with JSON, XML, or any other type of data like Excel, CSV or any other table-based data.

New releases of the PB mostly focus on .net compatibility. From Version 11 conversion to .net started. The code can be converted to Asp.Net and ported to IIS and run as a web-based application.

The latest version contains Powerscript migrator, which converts PowerBuilder code to C# code.

Powerserver helps to distribute the PowerBuilder application through the web to multiple locations.

But first I will show how to create master windows for the best performance. This is the first step in developing applications at rapid speed.

There are two main master windows one with a single datawindow for master data entry and double datawindows for master details type data entry. Then there are two types of report masters. The first type does not use any retrieval arguments and is mainly useful for displaying master data and simple reports. The second report master window uses retrieval arguments like dates and other data. Based on the data selected the report is retrieved and there are buttons for zooming the data and send to a printer.

Development of Master windows



This is the single datawindow master window. The white space in the middle of the window is the one where we will place datawindow. I will show how the datawindow is made later in the chapter.

Now let us see what the page consists of. The ribbon containing the commands Save, Modify, View, Delete and Cancel is called a user object. A user object can be created and this object is created by placing buttons and writing code for the buttons.

Global variables

```
string coname,addok,am,fayear,module,machine, loc,  
username
```

```
string voutype,ind_pol, corp_pol, pre_auth, claim_reg,  
companyname,company, branch, ls_string, gs_fayear,  
gs_custcd, claim_proc, claim_set, mis_rep ,ttype
```

```
date std_date,today
```

```
datetime lst_date,fafdrdt,fatodt,sysdatetime
```

```
int level
```

```
long ctr,ref_no
```

```
char login,item_flag
```

```
int ts
```

```
DataWindow gdw1,gdw2
```

```
string gs_which_window
```

```
char o_flag,sales,stores,purchase,finance,setup,other
```

```
datawindow dw_x
```

```
datawindowchild dwc
string pay,fin,pname,g_logincd,g_login,stype
string chk,subaccd
real amt
long vouno
string ptype,ins_co,ins_type,contype
```



Instance variables fr user object

```
datawindow dw_1
boolean commit_flag,save_flag
function: uof_reg returns none, input datawindow dw_1a
dw_1=dw_1a
```

Code for save button:

```
dw_1.setredraw(false)
integer rtn
if delete_flag = true then
if messagebox("CONFIRM DELETE", "Are you sure you want to
delete this record?",Question!,YesNo!,2) = 1 then
    rtn = dw_1.deleterow(0)
    if rtn = 1 then
        parent.triggerevent("delete_event")
        rtn = dw_1.update()
        if rtn = 1 then
            commit using sqlca;
w_mastermenu.sle_1.text = "Record has been deleted."
        delete_flag = false
        pb_cancel.triggerevent("clicked")
```

```

        else
messagebox("Deletion Error","Related Data Exists
        pb_cancel.triggerevent("clicked")
end if
end if
        end if
else
        parent.triggerevent("save_before_event")
        if update_flag = true then
            rtn = dw_1.update()
            if rtn = 1 then
                parent.triggerevent("save_event")
                if commit_flag = true then
                    commit using sqlca;
w_mastermenu.sle_1.text = "Records have been saved."
                    update_flag = false

                pb_cancel.triggerevent("clicked")
                    else
                        rollback using sqlca;
                    end if
                else
messagebox("Save Error","Save has been cancelled")
                    rollback using sqlca;
                end if
            else
                rollback using sqlca;
            end if
        end if
end if

```

```
dw_1.setredraw(true)
```

code for modify button:

```
dw_1.setredraw(false)
delete_flag = false
parent.triggerevent("modify_before_event")
if ts=1 then
    return 1
end if
dw_1.reset()
dw_1.insertrow(0)
dw_1.setfocus()
parent.triggerevent("modify_event")
dw_1.setredraw(true)
```

code for view button:

```
dw_1.setredraw(false)
delete_flag = false
parent.triggerevent("view_before_event")
if ts=1 then
    return 1
end if
dw_1.reset()
dw_1.insertrow(0)
dw_1.setfocus()
parent.triggerevent("view_event")
dw_1.setredraw(true)
```

code for delete button:

```
dw_1.setredraw(false)
```

```
parent.triggerevent("delete_before_event")
```

```
if ts=1 then
```

```
    return 1
```

```
end if
```

```
dw_1.reset()
```

```
dw_1.insertrow(0)
```

```
dw_1.setfocus()
```

```
dw_1.setredraw(true)
```

Code for cancel button:

```
parent.triggerevent("cancel_event")
```

Code for exit button:

```
if dw_1.modifiedcount() + dw_1.deletedcount() > 0 then
```

```
    if messagebox("UNSAVED CHANGES!", "Do you wish to  
exit without saving ?", Question!, YesNo!, 2) = 1 then
```

```
        close(parent)
```

```
    end if
```

```
else
```

```
    close(parent)
```

```
end if
```

Next code for the master window:

Code for open event of the window:

```
dw_1.settransobject(sqlca)
```

```
w_mastermenu.pb_1.enabled = false
```

```
uo_2.uf_register(dw_1)
```

```
setbuttons(true)
```

```
uo_2.pb_new.triggerevent(clicked!)
```

```
i_height = this.height
i_width = this.width
sle_1.taborder = 0
dw_1.modify("datawindow.color="+string(rgb(216,228,248))
+""")
rr_1.fillcolor=rgb(216,228,248)
return 0
```

Code for close event:

```
w_mastermenu.pb_1.enabled = true
// the main window will open
commit using sqlca;
// all the pending data updates will be committed
```

clicked event of the window – useful for web application

```
#IF Defined PBDOTNET THEN
    dw_1.JavaScriptFile = "MyScriptFile.js"
    dw_1.OnClientClicked = "MyDWClickedEventHandler"
#END IF
****
```

Instance variables

```
integer i_height, dw_height, dw_width
integer i_width, error_flag
string i_dml_flag, colname
long l_rows
int finyear
```

Function setbuttons: Boolean a_type as input returns none

```
//this function sets the buttons as per the user type and
module can be reconfigured
```



```
//string add,del ,mod
//Instant variables datawindow dw_1, boolean update_flag,
//delete_flag, newflag,commit_flag
add='N'
del='N'
mod='N'
addok='N'
if level=0 or level>3 then
    addok='Y'
    add='Y'
    mod='Y'
    del='Y'
end if
if add = 'Y' then
    addok='Y'
else
    addok='N'
end if
if add="Y" then
    uo_2.pb_new.enabled=true
else
    uo_2.pb_new.enabled=false
end if
if mod = 'Y' then
    uo_2.pb_modify.enabled=true
else
    uo_2.pb_modify.enabled=false
end if
if del = 'Y' then
```

```

        uo_2.pb_delete.enabled=true
else
        uo_2.pb_delete.enabled=false
end if
if add = 'N' and del = 'N' and mod = 'N' then
        uo_2.pb_save.enabled=false
else
        uo_2.pb_save.enabled=true
end if
uf_enable_on_prev input string ucode
string new,mod,del,view
declare priviledges cursor for
        select privnew,privdel,privmod,privview
        from users
        where ucode = :ucode
        using sqlca;
open priviledges;
fetch priviledges into :new,;:del,;:mod,;:view;
if sqlca.sqlcode <> 0 then
        messagebox("SQL ERROR",sqlca.sqlerrtext)
        close priviledges;
        return
end if
close priviledges;
if new = 'Y' then
        pb_new.enabled = true
        pb_save.enabled = true
end if

```

```
if del = 'Y' then
    pb_delete.enabled = true
    pb_save.enabled = true
end if
if mod = 'Y' then
    pb_modify.enabled = true
    pb_save.enabled = true
end if
if view = 'Y' then
    pb_view.enabled = true
end if
uf_register input by ref datawindow a_dw_1
dw_1=a_dw_1
return
uf_setbuttons input Boolean atype
if atype then
    pb_new.visible=false
    pb_save.visible=false
    pb_delete.visible=false
    pb_modify.visible=false
    pb_view.visible=false
    pb_cancel.visible = false
else
    pb_new.visible=true
    pb_save.visible=true
    pb_delete.visible=true
    pb_modify.visible=true
    pb_view.visible= true
```

```
        pb_cancel.visible =true
end if
```

Datawindow events:

Dberror event: this will raise the error in case of any database error:

```
#IF Defined PBNATIVE THEN
return gf_dberror(sqldbcode,sqlerrtext,row,dw_1)
#END IF
```

Itemchanged event:

//Most useful code for the datawindow

```
#IF DEFINED PBWEBFORM THEN
    dw_1.JavaScriptFile = "MyScriptFile.js"
    dw_1.OnClientItemChanged = "MyItemChanged"
#END IF
#IF DEFINED PBWEBFORM THEN
    dw_1.JavaScriptFile = "MyScriptFile.js"
    dw_1.OnClientItemChanged = "MyItemChanged"
#END IF
```

```
#IF not Defined PBWEBFORM THEN
```

```
if trim(data) = "" then
    any nullval
    setnull(nullval)
    dw_1.setitem(row,dw_1.getcolumn(),nullval)
    return 1
```

```

end if
string descr
choose case dw_1.describe(dwo.name+".ColType")
    case "number","decimal(2)","decimal(0)"
        if isnumber(data) then
            if real(data) < 0 then
                gf_message("positive","ERR")
                return 2
            end if
        else
            gf_message("number","ERR")
            return 2
        end if
    end choose
integer count
string colval,filter
choose case i_dml_flag
    case 'N'
        if dw_1.getcolumn()= 1 AND
Dw_1.describe(dwo.name+".dddw.required") = 'yes' then
            if gf_fkeyvalidation(dwo.name,data,dw_1) = 1 then
                gf_messask("Code Already Exists",1)
                return 1
            end if
        END IF
    case 'M','V','D'
        if dw_1.describe(dwo.name+".dddw.required") = 'yes'
then
            if gf_fkeyvalidation(dwo.name,data,dw_1) = 0 then

```

```

        gf_messask("Code Does Not Exist",1)
        return 1
    end if
END IF
end choose
choose case i_dml_flag
    case "M","D","V"
        colname = dwo.name
        colval = data
        filt = colname + " = '" + colval + "'"
        dw_1.setfilter(filt)
        dw_1.retrieve(data)
        if dw_1.rowcount() = 0 then
            gf_message("notfound","ERR")
            dw_1.insertrow(0)
            dw_1.scrolltorow(1)
            return 11
        else
            if i_dml_flag = "D" then
w_mastermenu.sle_1.text = "Please save to confirm delete"

                dw_1.object.datawindow.readonly = true
                else
                dw_1.settaborder(1,0)
                end if
                if i_dml_flag = "V" then

                dw_1.object.datawindow.readonly = true
                end if

```

```
        i_dml_flag = i_dml_flag+"G"  
    end if
```

```
end choose
```

```
#END IF
```

Code for itemerror event

```
#IF Defined PBNATIVE THEN
```

```
    RETURN 1
```

```
#END IF
```

Code for ue_dwnprocessenter

```
#IF Defined PBNATIVE THEN
```

```
send(handle(this),256,9,long(0,0))
```

```
return 1
```

```
#END IF
```

Code for user object in the window: used as uo_2

Code for new_event

```
if i_dml_flag = "S" or i_dml_flag = "DG" then
```

```
    datawindowchild dddw
```

```
    if dw_1.getcolumn() <> 1 then
```

```
        dw_1.setcolumn(1)
```

```
    end if
```

```
    dw_1.getchild(dw_1.getcolumnname(),dddw)
```

```
    dddw.settransobject(sqlca)
```

```
    dddw.retrieve()
```

```
end if
```

```
i_dml_flag = "N"
```

cancel_event

```
if i_dml_flag = "N" or i_dml_flag = "S" then
```

```

        dw_1.reset()
    end if
    uo_2.pb_new.triggerevent("clicked")

    pb_view.enabled = true
delete_before_event
    if dw_1.modifiedcount() + dw_1.deletedcount() > 0 then
        if messagebox("UNSAVED CHANGES!", " Changes have
        not been saved ~n Do you wish to
        continue?", Question!, YesNo!, 2) = 2 then
            messagebox("Information", "Please click the
            'Save' Button to ~n Save the changes")
            dw_1.setcolumn(1)
            dw_1.setfocus()
            ts=1
            sle_1.text=" New"
            return 1
        end if
    end if

    dw_1.modify(dw_1.getitemstring(dw_1.getrow(),1)+".dddw.r
    equired = no")
    choose case i_dml_flag
        case "DG"
            dw_1.object.datawindow.readonly = false
            dw_1.settaborder(1,10)
        case "MG"
            dw_1.settaborder(1,10)
    end choose

```



```
delete_flag = true
sle_1.text = "Delete"
i_dml_flag = "D"
```

```
pb_modify.enabled = false
pb_view.enabled = false
pb_delete.enabled = false
```

modify_before_event

```
if dw_1.modifiedcount() + dw_1.deletedcount() > 0 then
    if messagebox("UNSAVED CHANGES!"," Changes have
not been saved ~n Do you wish to
continue?","Question!,YesNo!,2) = 2 then
        messagebox("Information","Please click the
'Save' Button to ~n Save the changes")
        dw_1.setcolumn(1)
        dw_1.setfocus()
        ts=1
        sle_1.text=" New"
        return 1
    end if
end if
```

```
sle_1.text = "Modify"
choose case i_dml_flag
    case "DG"
        dw_1.object.datawindow.readonly = false
        dw_1.settaborder(1,10)
    case "MG"
        dw_1.settaborder(1,10)
```

```

end choose
if dw_1.rowcount() = 0 then
    dw_1.insertrow(0)
    dw_1.scrolltorow(dw_1.rowcount())
    dw_1.setfocus()
    dw_1.setcolumn(1)
end if
modify_event
i_dml_flag = "M"

pb_modify.enabled = false
pb_delete.enabled = false
pb_view.enabled = false
code for new_before_event
choose case i_dml_flag
    case "V","MG"
        if
dw_1.getitemstatus(dw_1.getrow(),0,primary!) <> new! then
            dw_1.reset()
        end if
        dw_1.settaborder(1,10)
    case "DG","VG"
        dw_1.object.datawindow.readonly = false
        dw_1.reset()
    case "N"
        if
dw_1.getitemstatus(dw_1.getrow(),0,primary!) = new! then
            newflag = true
            return

```

```

        else
            newflag = false
        end if
    case "M","D"
        dw_1.reset()
    end choose
    if dw_1.getcolumn() <> 1 then
        dw_1.setcolumn(1)
    end if
    setbuttons(true)
    sle_1.text = " New"
save_before_event
    if i_dml_flag ="N" and dw_1.getitemstatus(
    1,0,primary!)=New! then
        return 0
    end if
    if gf_chkreqcol(dw_1) = 1 then
        update_flag = true
        i_dml_flag = "S"
        dw_1.setitem(1,"usrcd",g_login)
        dw_1.setitem(1,"sysdate",sysdatetime)
    else
        update_flag = false
        return
    end if
save_event
    i_dml_flag = "S"
    commit_flag = true
    pb_delete.enabled = false

```

```

pb_view.enabled = false
pb_modify.enabled = false
view_before_event
if dw_1.modifiedcount() + dw_1.deletedcount() > 0 then
if messagebox("UNSAVED CHANGES!"," Changes have not
been saved ~n Do you wish to continue?",
Question!,YesNo!,2) = 2 then
    messagebox("Information","Please click the 'Save'
Button to ~n Save the changes")
        dw_1.setcolumn(1)
        dw_1.setfocus()
        ts=1
        sle_1.text=" New"
        return 1
    end if
end if

choose case i_dml_flag
    case "MG"
        dw_1.settaborder(1,10)
    case "DG","VG"
        dw_1.object.datawindow.readonly = false
        dw_1.settaborder(1,10)
end choose
sle_1.text = " View"
i_dml_flag = "V"
view_event
pb_modify.enabled = false
pb_delete.enabled = false

```

```
pb_save.enabled = false
pb_view.enabled = false
```

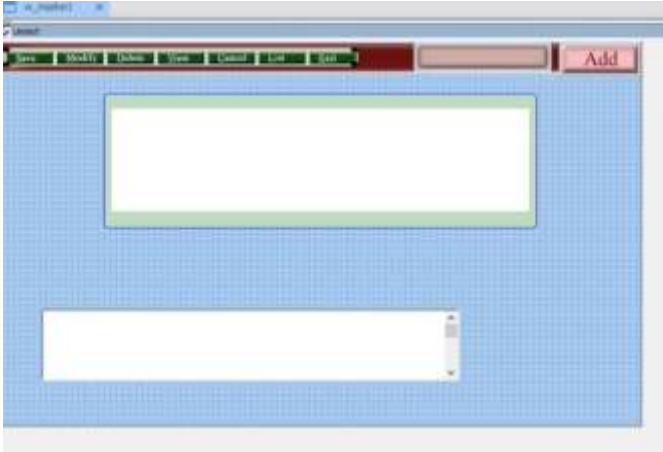
```
****
```

Some of the global variables might be needed.

```
string coname,addok, am,fayear,module, machine,loc,
username,ghpl_id
date std_date,today
datetime lst_date,fafrrdt,fatodt,sysdatetime
int level
long ctr,ref_no
char login
string voutype, ind_pol, corp_pol, pre_auth, claim_reg,
claim_proc, claim_set, mis_rep ,ttype
string companyname, company,branch, ls_string, gs_fayear,
gs_custcd, item_flag
int ts
DataWindow gdw1,gdw2
string gs_which_window
char o_flag,sales,stores,purchase,finance,setup,other
//New for Rk Apex
datawindow dw_x
datawindowchild dwc
string pay,fin,pname,g_logincd,g_login,stype
string chk,subaccd
real amt
long vouno
string ptype,ins_co,ins_type,contype
```

```
*****
```

Second single master window with list option



Extra code for List

This list will only work if a datawindow is placed on dw_2

```
if trim(dw_2.dataobject)<>"" then
```

```
    dw_2.visible=true
```

```
    dw_2.retrieve()
```

```
end if
```

quit event:

```
“ “
```



Event coding same as the user event

List clicked event

```
parent.triggerevent("list")
```

quit clicked event

```
if dw_1.modifiedcount() + dw_1.deletedcount() > 0 then
```

```

        if messagebox("Unsaved changes","Changes have not
        been saved ~n Do you wish to continue ?
        ",Question!,YesNo!,1) = 2 then
            return
        end if
    end if
end if
close (parent)

```

User object for double datawindow



The coding for double datawindow is slightly different

Coding for save button clicked event

//please note the use of dw_2 for second datawindow

```

int rtn
parent.triggerevent("savebefore")
if dw_2.modifiedcount()+dw_2.deletedcount() > 0 then
    if save_flag = true then
        rtn=dw_1.update()
        if rtn <> 1 then
            gf_messask("Save Cancelled-Rolling
            Back"+ sqlca.sqlerrtext,1)
            ROLLBACK USING SQLCA;
            return 1
        else
            rtn=dw_2.update()
            if rtn <> 1 then
                gf_messask("Save Cancelled-Rolling
                Back"+ sqlca.sqlerrtext,1)
                ROLLBACK USING SQLCA;
            end if
        end if
    end if
end if

```

```

        return 1
    else

        parent.triggerevent("saveafter")
        if commit_flag = true then
            COMMIT USING SQLCA;

            gf_mess("Additions / Changes have been saved")
            dw_2.reset()
            dw_1.reset()

            pb_cancel.triggerevent("clicked")
            return 0
        else

            gf_messask("Cancelled-Rolling Back"+
            sqlca.sqlerrtext,1)
            rollback using sqlca;
            return 1
        end if
    end if
end if
else
    gf_messask("Not Saving",1)
    return 1
end if
else
    dw_2.reset()
    dw_1.reset()

```



```

        pb_cancel.triggerevent("clicked")
    end if
modify clicked event
if dw_2.modifiedcount() + dw_2.deletedcount() > 0 then
    if messagebox("Unsaved changes", "Changes have not
been saved ~n Do you wish to continue ?
", Question!, YesNo!, 1) = 2 then
        return
    end if
end if

    dw_1.reset()
    dw_2.reset()
    dw_2.insertrow(0)
    dw_1.insertrow(0)
dw_1.setcolumn(1)
dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
parent.triggerevent("modify")
dw_1.setfocus()
delete clicked event
if dw_2.modifiedcount() + dw_2.deletedcount() > 0 then
if messagebox("Unsaved changes", "Changes have not been
saved ~n Do you wish to continue ? ", Question!, YesNo!, 1) = 2
then
        return
    end if
end if

    dw_1.reset()
    dw_2.reset()

```

```
        dw_1.insertrow(0)
        dw_2.insertrow(0)
dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
dw_1.setcolumn(1)
```

```
parent.triggerevent("deleted")
dw_1.setfocus()
```

view clicked event

```
if dw_2.modifiedcount() + dw_2.deletedcount() > 0 then
if messagebox("Unsaved changes", "Changes have not been
saved ~n Do you wish to continue ? ", Question!,YesNo!,1) = 2
then
```

```
        return
```

```
    end if
```

```
end if
```

```
        dw_2.reset()
        dw_1.reset()
        dw_2.insertrow(0)
        dw_1.insertrow(0)
```

```
dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
dw_1.setcolumn(1)
```

```
parent.triggerevent("view")
dw_1.setfocus()
```

cancel clicked event

```
if dw_2.modifiedcount() + dw_2.deletedcount() > 0 then
```

```
        if messagebox("Unsaved changes","Changes have not
been saved ~n Do you wish to continue ?
",Question!,YesNo!,1) = 2 then
```

```
            return
```

```
        end if
```

```
end if
```

```
        dw_1.reset()
```

```
        dw_1.insertrow(0)
```

```
        dw_2.insertrow(0)
```

```
        dw_2.reset()
```

```
dw_1.settaborder(1,10)
```

```
dw_1.settaborder(2,20)
```

```
dw_1.setcolumn(1)
```

```
parent.triggerevent("cancel")
```

```
dw_1.setfocus()
```

```
quit clicked event
```

```
parent.triggerevent("quit")
```

```
user object events:
```

```
uo_master2 is the name of the userobject
```

```
uo_master2 cancel event
```

```
dw_1.setfocus()
```

```
uo_master2 deleted event
```

```
dw_1.setfocus()
```

```
uo_master2 modify event
```

```
dw_1.setfocus()
```

```
uo_master2 quit event
```

```
if dw_2.modifiedcount() + dw_2.deletedcount() > 0 then
```

```
if messagebox("Unsaved changes","Changes have not been saved ~n Do you wish to continue ? ", Question!,YesNo!,1) = 2 then
```

```
    return
```

```
end if
```

```
end if
```

```
//gf_mess(coname)
```

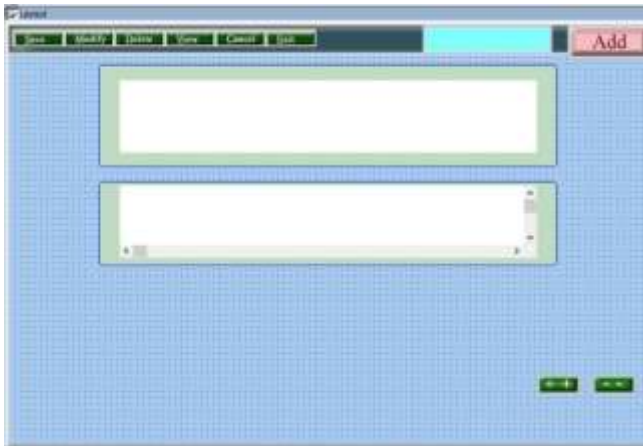
```
close (parent)
```

uo_master2 view event

```
dw_1.setFocus()
```

```
//all uo_master events can be programmed through windows
```

window with 2 datawindows Master details window



Cancel event on userobject

```
if dw_1.rowcount() = 0 then
```

```
    dw_1.insertrow(0)
```

```
end if
```

```
if dw_2.rowcount()=0 then
```

```

        dw_2.insertrow(0)
    end if
    dw_1.setcolumn(1)
    dw_1.setfocus()
    uo_1.pb_save.enabled=true
    dw_2.Object.DataWindow.ReadOnly="No"
    dw_1.Object.DataWindow.ReadOnly="No"
    datawindowchild dwchld
    dw_1.getchild(dw_1.getcolumnname(),dwchld)
    dwchld.settransobject(sqlca)
    IF fyok='Y' then
        dwchld.setfilter("fincode = "+string(finyear))
    end if
    dwchld.retrieve()
    dw_1.setcolumn(2)
    dw_1.settaborder(1,0)
    dw_1.settaborder(2,10)
    dw_1.setfocus()
    if addok='Y' then
        flag = 'A'
        mle_1.picturename='bgadd.jpg'
    else
        uo_1.pb_view.triggerevent(clicked!)
    end if
Deleted event on userobject
    if dw_1.rowcount() = 0 then
        dw_1.insertrow(0)
    end if

```

```

if dw_2.rowcount()=0 then
    dw_2.insertrow(0)
end if
dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
dw_1.setcolumn(1)
dw_1.setfocus()
flag = 'D'
dw_2.Object.DataWindow.ReadOnly="No"
dw_1.Object.DataWindow.ReadOnly="No"
mle_1.picturename='bgdelete.jpg'
modify event on userobject
dw_2.Object.DataWindow.ReadOnly="No"
dw_1.Object.DataWindow.ReadOnly="No"
if dw_1.rowcount() = 0 then
    dw_1.insertrow(0)
end if
if dw_2.rowcount()=0 then
    dw_2.insertrow(0)
end if
dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
dw_1.setcolumn(1)
dw_1.setfocus()
flag = 'M'
uo_1.pb_save.enabled=true
mle_1.picturename='bgmodify.jpg'
save_after event on userobject

```

```

commit_flag = true
save_before event on userobject
int cntr
if flag='DG' then
if gf_messask("Select Option~rDo you Really want to
Delete",2)=1 then
            dw_1.deleterow(1)
            dw_2.rowsmove(1,dw_2.rowcount(),primary!,
dw_2,1,delete!)
            flag="DY"
            if dw_2.rowcount()>0 then
gf_messask("Not All Data Deleted",1)
            end if
        else
            flag="DN"
        end if
    else
//          Delete Last Row if New
        if
dw_2.getitemstatus(dw_2.rowcount(),0,primary!) = New!
then
dw_2.deleterow(dw_2.rowcount())
end if
        if
dw_2.getitemstatus(dw_2.rowcount(),0,primary!) = New!
then
            dw_2.deleterow(dw_2.rowcount())
        end if
//          Check for Other Blank Rows

```

```

        cntr=dw_2.rowcount()
do while cntr>0
if dw_2.getitemstatus(cntr,0,primary!) = New! then
            dw_2.deleterow(cntr)
        end if
                cntr--
loop
if gf_chkreqcol(dw_1) = 1 then
        commit_flag= true
        save_flag = true
else
commit_flag= false
save_flag = false
gf_messask("Save Cancelled-Req.Cols",1)
dw_1.setfocus()
return 1
end if
if gf_chkreqcol(dw_2) = 1 then
        commit_flag= true
        save_flag = true
else
        commit_flag= false
        save_flag = false
gf_messask("Save Cancelled-Req.Cols",1)
        dw_2.setfocus()
        return 1
end if
end if

```



```

if left(flag,1)<>'D' then
dw_1.setitem(dw_1.getrow(),"usrcd",g_logincd)
    if fyok='Y' then

        dw_1.setitem(dw_1.getrow(),"fincode",finyear)
        end if
        for ctr=1 to dw_2.rowcount()
            dw_2.setitem( ctr,"usrcd",g_logincd)
            if fyok='Y' then
                dw_2.setitem(ctr,"fincode",finyear)
            end if
        next
    end if
save_flag= true
view event on userobject
if dw_1.rowcount() = 0 then
    dw_1.insertrow(0)
end if
if dw_2.rowcount()=0 then
    dw_2.insertrow(0)
end if

dw_1.settaborder(1,10)
dw_1.settaborder(2,20)
dw_1.setcolumn(1)
dw_1.setfocus()
flag = 'V'
dw_2.Object.DataWindow.ReadOnly="No"

```

```
dw_1.Object.DataWindow.ReadOnly="No"
```

```
mle_1.picturename='bgmodify.jpg'
```

Itemchanged event on datawindow 1

```
#IF DEFINED PBWEBFORM THEN
```

```
dw_1.JavaScriptFile = "MyScriptFile.js"
```

```
dw_1.OnClientItemChanged = "MyItemChanged"
```

```
#END IF
```

```
#IF DEFINED PBWEBFORM THEN
```

```
dw_1.JavaScriptFile = "MyScriptFile.js"
```

```
dw_1.OnClientItemChanged = "MyItemChanged"
```

```
#END IF
```

```
#IF Defined PBNATIVE THEN
```

```
er_flag=gf_chkval(
```

```
dw_1.describe(dwo.name+".ColType"),data)
```

```
if er_flag <> 0 then
```

```
    return er_flag
```

```
end if
```

```
IF dw_1.getcolumn()<> 1 then
```

```
if dw_1.describe(dwo.name+".dddw.required") = 'yes' then
```

```
if gf_fkeyvalidation(dwo.name,data,dw_1)=0 then
```

```
    er_flag=8
```

```
    return 8
```

```
end if
```

```
end if
```

```
end if
```

```
if dw_1.describe(dwo.name+".EDIT.required") = 'yes' then
```

```
    IF trim(data)=" then
```

```
        er_flag=9
```

```

        return 9
    end if
end if

string colname

colname = dwo.name
choose case flag
    case "M","V","D"
        srno=data
        if trim(srno)= "" then
            return 1
        end if
        dw_1.SetFilter("")
        if fyok='Y' then
            if voutype='ST' then

                dw_1.retrieve(srno,finyear,stype)
            else
                dw_1.retrieve(srno,finyear)
            end if
        else
            dw_1.retrieve(srno)
        end if
        if dw_1.rowcount() > 0 then
            dw_1.settaborder(1,0)
            flag=flag+"G"
            dw_2.setfilter("")
            if fyok='Y' then

```

```

        if voutype="ST" then

dw_2.retrieve(srno,finyear,stype)
            else

dw_2.retrieve(srno,finyear)
            end if
        else
            dw_2.retrieve(srno)
        end if
        if dw_2.rowcount() =0 then
            dw_2.insertrow(0)
        else
            dw_1.settaborder(1,10)
        end if
        dw_1.settaborder(1,0)
        dw_1.setcolumn(2)
        if flag='DG' then

dw_2.Object.DataWindow.ReadOnly="Yes"

dw_1.Object.DataWindow.ReadOnly="Yes"
            gf_mess("Please Save to Delete")
            uo_1.pb_save.setfocus()
            elseif flag="VG" then

dw_2.Object.DataWindow.ReadOnly="Yes"

dw_1.Object.DataWindow.ReadOnly="Yes"

```

```

        gf_mess("Please Cancel to Cancel")
        uo_1.pb_save.enabled=false
        uo_1.pb_cancel.setFocus()
    else
        dw_1.setColumn(2)
    end if
ELSE
    dw_1.insertrow(0)
    RETURN 1
end if
end choose
#end if

```

Itemerror event on dw_1

Input: long row, dwobject dwo, string data returns long

#IF Defined PBNATIVE THEN

```

choose case er_flag
    case 1
        gf_messask("Wrong Data Entry Re-Check",1)
    case 2
        gf_messask("Number Should be positive",1)
    case 3
        gf_messask("Number Should be Entered",1)
    case 4
        gf_messask("Not a Date - Re-Enter",1)
    case 5
        gf_messask("Wrong Date -Reenter",1)
    case 6
        gf_messask("Wrong Date-Date should be between
"+string(date(fafprd))+ ' '+string(date(fatodt)),1)

```

```

    case 7
    gf_messask("Code Already Entered",1)
    case 8
    gf_messask("No. Does not Exist",1)
    case 9
    gf_messask("Column cannot be blank",1)

end choose
er_flag=0
RETURN 1
#end if

Dberror on dw_1
#IF Defined PBNATIVE THEN
return gf_dberror(sqldbcode,sqlerrtext,row,dw_1)
#end if

Enter event on dw_1
#IF Defined PBNATIVE THEN
send(handle(this),256,9,long(0,0))
return 1
#end if

Itemchanged event on dw_2
#IF DEFINED PBWEBFORM THEN
    dw_2.JavaScriptFile = "MyScriptFile.js"
    dw_2.OnClientItemChanged = "MyItemChanged"
#END IF
#IF DEFINED PBWEBFORM THEN
    dw_2.JavaScriptFile = "MyScriptFile.js"
    dw_2.OnClientItemChanged = "MyItemChanged"

```

```
#END IF
```

```
#IF Defined PBNATIVE THEN
```

```
er_flag=gf_chkval(  
dw_2.describe(dwo.name+".ColType"),data)
```

```
if er_flag <> 0 then
```

```
    return er_flag
```

```
end if
```

```
if dw_2.describe(dwo.name+".dddw.required") = 'yes' then
```

```
if gf_fkeyvalidation(dwo.name,data,dw_2) = 0 then
```

```
    er_flag=8
```

```
    return 8
```

```
    end if
```

```
end if
```

```
if dw_1.describe(dwo.name+".EDIT.required") = 'yes' then
```

```
    IF trim(data)=" then
```

```
        er_flag=9
```

```
        return 9
```

```
    end if
```

```
end if
```

```
#end if
```

Remaining events are same as dw_1

Code on ++ button for adding rows in dw_2

```
dw_2.insertrow(0)
```

```
dw_2.scrolltorow(dw_2.rowcount())
```

```
dw_2.setcolumn(1)
```

```
dw_2.setfocus()
```

Code on -- button for adding rows in dw_2

```
int delrow
delrow=dw_2.getrow()
if delrow=1 and dw_2.rowcount()=1 then
    dw_2.deleterow(delrow)
    dw_2.insertrow(0)
elseif delrow=1 and dw_2.rowcount()>1 then
    dw_2.scrollnextrow()
    dw_2.deleterow(delrow)
elseif delrow=dw_2.rowcount() then
    dw_2.scrollpriorrow()
    dw_2.deleterow(delrow)
else
    dw_2.scrollnextrow()
    dw_2.deleterow(delrow)
end if
```

```
dw_2.setcolumn(1)
```

```
dw_2.setfocus()
```

opening event for window

```
w_mastermenu.sle_1.enabled=true
```

```
w_mastermenu.sle_1.text=pb_name.text+" Opening Please  
Wait"
```

```
w_mastermenu.pb_1.enabled=false
```

```
this.title=pb_name.text
```

```
uo_1.uf_reg(dw_1,dw_2)
```

```
dw_1.settransobject(sqlca)
```

```
dw_2.settransobject(sqlca)
```

```
dw_1.reset()
```

```
dw_2.reset()
```

```
dw_1.insertrow(0)
```



```
dw_2.insertrow(0)
dw_1.setcolumn(1)
dw_1.setfocus()
flag='A'
wf_enable()
dw_2.setrowfocusindicator(hand!)
this.x=0
this.y=0
this.width=3657
this.height=1992
uo_1.pb_cancel.triggerevent(clicked!)
if left(coname,1)='S' then
    this.icon="backup.ico"
else
    this.icon="tomcat.ico"
end if
dw_1.modify("datawindow.color="+
string(rgb(216,228,248))+""")
r_1.fillcolor=rgb(216,228,248)
dw_2.modify("datawindow.color="+
string(rgb(216,228,248))+""")
r_2.fillcolor=rgb(216,228,248)
```

Report windows



Preview clicked

```
sle_2.text = "Preview"
```

```
long lrows
```

```
lrows = dw_1.retrieve(companyname)
```

```
if lrows = 0 then
```

```
    messagebox("Information","No details exist . . .")
```

```
    return
```

```
end if
```

file clicked event

```
sle_2.text = " To File"
```

```
dw_1.saveas("",Excel5!,true)
```

```
w_mastermenu.sle_1.text = "To Preview the details.."
```

print clicked event

```
sle_2.text = " Print"
```

```
s_prnset |_prnset
```

```
l_prnset.dw_a = dw_1
```

```
l_prnset.default = 1
```

```
openwithparm(w_prnset,l_prnset)
```

zoom clicked

```
openwithparm(w_zoom,dw_1)
```

w_prnset used in report print



Open event

```
s_prnset l_prnset
l_prnset = message.powerobjectparm
if not isnull(l_prnset.default) then
    wf_assigndw(l_prnset.dw_a)
    ddlb_size.selectitem(l_prnset.default)
else
    wf_assigndw(l_prnset.dw_a)
    ddlb_size.selectitem(1)
end if
if LeftA(coname,1)='S' then
    this.icon="backup.ico"
else
    this.icon="tomcat.ico"
end if
```

close event

```
string range,status,str1
if okcancel = "Y" then
    if rb_pages.checked = true then
        range = sle_range.text
        str1 = "datawindow.print.page.range =
"+range+""
        status = dw_a.modify(str1)
        if status <> "" then
```

```

        messagebox("MODIFY", ""+status+"
Unable to modify datawindow properties")
    end if
end if
for ctr=1 to long(copy.text)
    dw_a.print()
next
end if

```

ok clicked event

```

okcancel = "Y"
close(parent)

```

setup clicked event

```

PrintSetup()

```

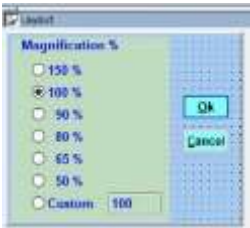
Cancel clicked event

```

okcancel = "N"
close(parent)

```

Zoom window used in printing



Open event in w_zoom

```

dw_x = message.powerobjectparm
zoom = 100
if LeftA(coname,1)='S' then
    this.icon="backup.ico"
else

```

```

        this.icon="tomcat.ico"
    end if
close event in zoom
string status
status = dw_x.modify("datawindow.print.preview.zoom =
"+string(zoom))
status = dw_x.modify("datawindow.zoom = "+string(zoom))
if status <> "" then
        messagebox("MODIFY","Unable to modify
datawindow properties")
end if
ok clicked event
okcancel = "Y"
close(parent)
cancel clicked
okcancel = "N"
close(parent)
selecting zoom option
zoom = 100
sle_1.text='100'
cb_1.triggerevent(clicked!)
same for other options
sle_1 %entry modified event
zoom = integer(sle_1.text)
cb_1.triggerevent(clicked!)
Report window with 2nd datawindow for report options

```



Dw_2 item error

return 1

preview button clicked

dw_2.accepttext()

Remaining same as report 1

Coding for selection on dw_2 written on the report

The following are the some of the global functions used in the masters:

Code for gf_dberror

// input, interger sqldbcode, string sqlerrtext, integer row, datawindow //dw_b, this event is raised automatically.

int li_filename

if sqldbcode <> 0 then

 messagebox("Database Error",sqlerrtext,stopsign!)

 dw_b.scrolltorow(row)

 return 1

 li_filename = fopen("\\ntserver\e\newerror.log", linemode!, Write!, shared!, append!)

 if li_filename > 0 then

 if fwrite(li_filename,"~n~r"+"

 "+string(today())+ " " +string(now())+&

```

"Code: "
+string(sqldbcode) + "Message :" + sqlerrtext) < 0 then
    messagebox("Error Log Failure ",
"Unable to Log this Error ")
    fwrite(li_filename,' ')
    end if
    if fclose(li_filename) < 0 then
        messagebox("Error Log Failure ",
"Unable to close Log File ")
    end if
    else
        messagebox("Error Log Failure ", "Unable to Open Log
Error File ")
    end if
end if
return 0

```

This useful when we have huge data to share with dropdowns in multiple datawindows

dwc_share input by ref datawindow, string colname returns none

```

if w_mastermenu.dw_1.rowcount()=0 then
w_mastermenu.dw_1.retrieve()
dwno.getchild(colname,dwc)
dwc.settransobject(sqlca)
dwc.setfilter("")
dwc.filter()
w_mastermenu.dw_1.sharedata(dwc)

```

dwf_dwcfiltret input datawindow dwno,string colname,string filter returns none

//filter a datawindow dropdown as a childdatawindow

dwno.getchild(colname,dwc)

dwc.settransobject(sqlca)

dwc.setfilter(filter)

dwc.retrieve()

**dwf_dwcfiltset input datawindow dwno,string
colname,string filter returns none**

//sets the filter too

dwno.getchild(colname,dwc)

dwc.settransobject(sqlca)

dwc.setfilter(filter)

dwc.filter()

dwf_rvmb input datawindow dwno

//to check and inform no data retrieved

ctr = dwno.rowcount()

if ctr = 0 then

 messagebox("Information","There is no output for
the criteria you have choosen")

end if

**// Function f_mail_error_to_string (string
a_MailReturnCode, string a_Display) returns string**

// where: a_MailReturnCode MailReturnCode
value

//a_Message Error message to prepend to return string

//a_Display Boolean (TRUE=display error messagebox)

string s

choose case a_MailReturnCode


```
case mailReturnAccessDenied!  
    s = 'Access Denied'  
case mailReturnAttachmentNotFound!  
    s = 'Attachment Not Found'  
case mailReturnAttachmentOpenFailure!  
    s = 'Attachment Open Failure'  
case mailReturnAttachmentWriteFailure!  
    s = 'Attachment Write Failure'  
case mailReturnDiskFull!  
    s = 'Disk Full'  
case mailReturnFailure!  
    s = 'Failure'  
case mailReturnInsufficientMemory!  
    s = 'Insufficient Memory'  
case mailReturnInvalidMessage!  
    s = 'Invalid Message'  
case mailReturnLoginFailure!  
    s = 'Login Failure'  
case mailReturnMessageInUse!  
    s = 'Message In Use'  
case mailReturnNoMessages!  
    s = 'No Messages'  
case mailReturnSuccess!  
    s = 'Success'  
case mailReturnTextTooLarge!  
    s = 'Text Too Large'  
case mailReturnTooManyFiles!  
    s = 'Too Many Files'
```

```

case mailReturnTooManyRecipients!
    s = 'Too Many Recipients'
case mailReturnTooManySessions!
    s = 'Too Many Sessions'
case mailReturnUnknownRecipient!
    s = 'Unknown Recipient'
case mailReturnUserAbort!
    s = 'User Abort'

case else
    s = 'Other'
end choose

if a_Display then MessageBox ( 'Mail Return Code',
a_Message + ' ' + s, Exclamation!)

```

```
return s
```

gf_chkreqcol input datawindow dw_control returns integer

```
integer li_colnbr = 1
```

```
long ll_row = 1
```

```
string ls_colname, ls_textname, dw_name
```

```
IF dw_control.AcceptText() = -1 THEN
```

```
    dw_control.SetFocus()
```

```
    RETURN -2
```

```
END IF
```

```
IF dw_control.FindRequired( Primary!, ll_row, &
```

```
    li_colnbr, ls_colname, false ) < 0 THEN
```

```
    RETURN -2
```

```
END IF
```

```

IF ll_row <> 0 THEN
//      ls_textname = ls_colname + "_t.Text"
//      ls_colname = dw_control.Describe(ls_textname)
if string(dw_control)="dw_1" then
        dw_name="Header"
elseif string(dw_control)="dw_2" then
        dw_name="Details"
else
        dw_name="others"
end if
MessageBox("Required Value Missing", "Please enter Data
for:" +dw_name+",Col:" +ls_colname+ ", Row: "+
String(ll_row) + ". ",StopSign! )
        dw_control.SetColumn(li_colnbr)
        dw_control.ScrollToRow(ll_row)
        dw_control.SetFocus()
        RETURN -1
END IF
RETURN 1

```

Gf_chkval input string coltypes, string datas returns integer

```

integer li_colnbr = 1
long ll_row = 1
string ls_colname, ls_textname,dw_name
IF dw_control.AcceptText() = -1 THEN
        dw_control.SetFocus()
        RETURN -2
END IF
IF dw_control.FindRequired( Primary!, ll_row, &
        li_colnbr, ls_colname, false ) < 0 THEN

```

```

        RETURN -2
    END IF
    IF ll_row <> 0 THEN
        //      ls_textname = ls_colname + "_t.Text"
        //      ls_colname = dw_control.Describe(ls_textname)
        if string(dw_control)="dw_1" then
            dw_name="Header"
        elseif string(dw_control)="dw_2" then
            dw_name="Details"
        else
            dw_name="others"
        end if
        MsgBox("Required Value Missing", "Please enter
Data for:" + dw_name + ",Col:" + ls_colname + ", Row: "+
String(ll_row) + ".",StopSign! )
        dw_control.SetColumn(li_colnbr)
        dw_control.ScrollToRow(ll_row)
        dw_control.SetFocus()
        RETURN -1
    END IF
    RETURN 1
Gf_chkval_null input string coltypes, string datas, returns
integer
integer li_colnbr = 1
long ll_row = 1
string ls_colname, ls_textname,dw_name
IF dw_control.AcceptText() = -1 THEN
    dw_control.SetFocus()
    RETURN -2

```

```

END IF
IF dw_control.FindRequired( Primary!, ll_row, &
    li_colnbr, ls_colname, false ) < 0 THEN
    RETURN -2
END IF
IF ll_row <> 0 THEN
//    ls_textname = ls_colname + "_t.Text"
//    ls_colname = dw_control.Describe(ls_textname)
if string(dw_control)="dw_1" then
    dw_name="Header"
elseif string(dw_control)="dw_2" then
    dw_name="Details"
else
    dw_name="others"
end if
MessageBox("Required Value Missing", "Please enter Data
for:" +dw_name+",Col:" +ls_colname+ ", Row: "+
String(ll_row) + ". ",StopSign! )
    dw_control.SetColumn(li_colnbr)
    dw_control.ScrollToRow(ll_row)
    dw_control.SetFocus()
    RETURN -1
END IF
RETURN 1
Gf_deletedata input nobe,returns none
//to save deleted data in a table for audit purpose
long srno
datetime dt
dt=datetime(today())

```

```

SELECT max(deletedata.srno ) INTO :srno FROM
deletedata ;
srno++
INSERT INTO deletedata
      ( srno, ref_no, ghpl_id, ttype,
usracd, sysdate )
VALUES ( :srno, :ref_no, :ghpl_id, :ttype,
:g_logincd, :dt ) ;
if sqlca.sqlcode<>0 then
      gf_messask(sqlca.sqlerrtext,1)
end if
return sqlca.sqlcode
gf_facheckdate input date returns integer
if chkdate < date(fafrdt) or chkdate > date(fatodt) then
//      gf_messask("Wrong Date-Date should be between
"+string(date(fafrdt))+ ' & ' +string(date(fatodt)),1)
      return 6
else
      return 0
end if
      return 0

```

gf_facheckdt input date return integer

```

if chkdate < fafrdt or chkdate > fatodt then
//to check date entered is not out of range
//      gf_messask("Wrong Date-Date should be between
"+string(date(fafrdt))+ ' ' +string(date(fatodt)),1)
      return 6

```

```

else
    return 0
end if
    return 1
gf_fkeyvalidation input string colname, string colval,
datawindow which_dw
which_dw.getchild(colname,dwc)
colval=trim(colval)
ctr= dwc.find(colname+" = '"+colval+"'",1,dwc.rowcount())
if ctr= 0 then
    return 0 // row not found
elseif ctr>0 then
    return 1 // row found
elseif ctr<0 then
ctr= dwc.find("code = '"+colval+"'",1,dwc.rowcount())
if ctr= 0 then
    return 0 // row not found
else
    return 1 // row found
end if
end if
return 0

```

Gf_nextyear input date returns date

```

date todate
todate=date(string(year(frdate)+1)+'/'+
string(month(frdate))+'/'+string(day(frdate)))
todate=relativedate(todate,-1)
return todate

```

gf_mess input string mess return none

//to set message on the messagebar

if w_mastermenu.sle_1.enabled=false then

w_mastermenu.sle_1.enabled=true

w_mastermenu.sle_1.text=mess

gf_message input string cols, string flags returns none

//to set message on the message about the col

if w_mastermenu.sle_1.enabled=false then

w_mastermenu.sle_1.enabled=true

string desc

SELECT descrip

INTO :desc

FROM message

WHERE col = :col ;

if sqlca.sqlcode = 100 then

desc=upper(col)

end if

choose case flags

case "MG"

w_mastermenu.sle_1.text='Please

Change '+desc

case ELSE

w_mastermenu.sle_1.text ='Please

Enter '+desc

end choose

return

gf_messask input string mess, integer ask returns integer

int retval

if trim(mess)="" then

mess="? message"+mess


```

end if
choose case ask
case 1
    messagebox("Message",mess,Exclamation!)
    retval = 0
case 2
    retval =
    messagebox("Message",mess,Question!,YesNo!,1)
case 3
    retval = messagebox("Message",mess,Question!,
YesNoCancel!,2)
case 4
    retval = messagebox("Message",mess,
    Question!,YesNo!,2)
end choose
return retval

```

gf_messask_no input string mess, integer ask returns integer

```

int retval
if trim(mess)="" then
    mess="? message"+mess
end if
choose case ask
case 1
    messagebox("Message",mess,Exclamation!)
    retval = 0
case 2
    retval =
    messagebox("Message",mess,Question!,YesNo!,2)

```

```

        case 3
            retval =
messagebox("Message",mess,Question!,YesNoCancel
!,2)
        case 4
            retval =
messagebox("Message",mess,Question!,YesNo!,2)
        end choose

        return retval

```

gf_modifydata input none return integer

```

long srno
datetime dt
dt=datetime(today())
        SELECT max(srno ) INTO :srno FROM
modifydata ;
srno++
        INSERT INTO modifydata
        ( srno,          ref_no,          ghpl_id,          ttype,
usrcd,          sysdate )
        VALUES ( :srno,          :ref_no,          :ghpl_id,          :ttype,
:g_logincd,          :dt ) ;
        if sqlca.sqlcode<>0 then
            gf_messask(sqlca.sqlerrtext,1)
        end if
return sqlca.sqlcode

```

gf_numbertowords input decimal xcv_net, returns string

```

string cnum,camt,ones,teen,tens
if xcv_net>99999999 then
return ""

```

```

end if
camt=""
ones= 'ONE TWO THREE FOUR FIVE SIX SEVENEIGHT NINE '
teen= 'TEN ELEVEN TWELVE THIRTEEN FOURTEEN FIFTEEN
SIXTEEN SEVENTEEN EIGHTEEN NINETEEN'
tens= 'TWENTY THIRTY FORTY FIFTY SIXTY SEVENTYEIGHTY
NINETY'
cnum=LeftA(string(long(xcv_net)),8)
cnum=space(8 -LenA(cnum))+cnum
IF LeftA(cnum,1)>' ' then
camt= righttrim(MidA(ones,long(LeftA(cnum,1))*5+1,5))+
CRORES '
end if
IF LeftA(cnum,3)>' ' then
if MidA(cnum,2,1)>'1' then
    camt=camt+righttrim(MidA(tens,long(MidA(cnum,2,1))*7 -
13,7))
    IF MidA(cnum,3,1)>' ' then
        camt=camt+'
'+righttrim(MidA(ones,long(MidA(cnum,3,1))*5+1,5))
    end if
    camt = camt+' LAKHS '
elseif MidA(cnum,2,1)= '1' then
    camt=
camt+righttrim(MidA(teen,long(MidA(cnum,3,1))*9+1,9))+
LAKHS '
elseif MidA(cnum,2,2)= '00' then
    camt= camt
elseif MidA(cnum,3,1)> '0' then

```

```

    camt=
    camt+righttrim(MidA(ones,long(MidA(cnum,3,1))*5+1,5))+
    LAKHS '
END if
end if

cnum=MidA(cnum,3,6)

IF LeftA(cnum,3)>' ' then
if MidA(cnum,2,1)>'1' then
    camt=camt+righttrim(MidA(tens,long(MidA(cnum,2,1))*7 -
13,7))
    IF MidA(cnum,3,1)>' ' then
        camt=camt+'
'+righttrim(MidA(ones,long(MidA(cnum,3,1))*5+1,5))
    end if
    camt = camt+' THOUSAND '
elseif MidA(cnum,2,1)= '1' then
    camt=
    camt+righttrim(MidA(teen,long(MidA(cnum,3,1))*9+1,9))+
    THOUSAND '
elseif MidA(cnum,2,2)= '00' then
    camt= camt
elseif MidA(cnum,3,1)> '0' then
    camt=
    camt+righttrim(MidA(ones,long(MidA(cnum,3,1))*5+1,5))+
    THOUSAND '
END if
end if
IF MidA(cnum,4,1)>'0' then

```

```

camt=camt+righttrim(MidA(ones,long(MidA(cnum,4,1))*5+1,
5))+ ' HUNDRED '
end if
if MidA(cnum,5,1)>'1' then
camt = camt+righttrim(MidA(tens,long(MidA(cnum,5,1))*7 -
13,7))
IF RightA(cnum,1)>'0' then
    camt= camt+'
'+righttrim(MidA(ones,long(RightA(cnum,1))*5+1,5))
end if
elseif MidA(cnum,5,1)='1' then
camt=camt+righttrim(MidA(teen,long(RightA(cnum,1))*
9+1,9))
else
camt=camt+righttrim(MidA(ones,long(RightA(cnum,1))*
5+1,5))
END if
//Paise Checking
cnum=RightA(string(round(xcv_net,2)),9)
cnum=space(9 -LenA(cnum))+cnum

if RightA(cnum,2)<>'00' then
camt=camt+ " AND PAISE "
/*check tens and ones for paise
if MidA(cnum,8,1)>'1' then
camt = camt+righttrim(MidA(tens,long(MidA(cnum,8,1))*7 -
13,7))
if RightA(cnum,1)>'0' then
camt= camt+'
'+righttrim(MidA(ones,long(RightA(cnum,1))*5+1,5))

```

```

end if
elseif MidA(cnum,8,1)='1' then
camt=camt+rightright(MidA(teen,long(RightA(cnum,1))*9+1,9)
)
elseif RightA(cnum,2)= '00' then
camt= camt+'ZERO'
else
CAMT=CAMT+rightright(MidA(ONES,long(RightA(CNUM,1))*5
+1,5))
END if
end if
camt="RUPEES "+camt+' ONLY'
RETURN trim(camt)
Gf_pass input string pass returns string
int leng,ascd
string passout
passout=""
leng=Len(pass)
for ctr = 1 to leng
    ascd=Asc(Mid(pass,ctr,1))
    if ascd >127 then
        ascd -=127
    else
        ascd+=127
    end if
    passout=passout+Char(ascd)
next
return passout

```

gf_setfilter input datawindow dwno, string colname, string filter, returns none

```
dwno.getchild(colname,dwc)
```

```
dwc.settransobject(sqlca)
```

```
dwc.setfilter(filter)
```

```
dwc.filter()
```

gf_smsdata input string ph_no, string sms_mess returns integer

//to store the message in smstable to be sent later

```
datetime dt
```

```
long sno
```

```
dt=datetime(today())
```

```
if isnull(ph_no) or len(ph_no)<10 then
```

```
    return -1
```

```
else
```

```
    select max(sno) into :sno from  
    combidata.dbo.sms_data ;
```

```
    sno=sno+1
```

```
    INSERT INTO combidata.dbo.sms_data  
    (sms_mess,ph_no,active,usrcd,sysdate,sno)  
    values(:sms_mess,:ph_no,1,:g_logincd,:dt,:sno);
```

```
    gf_messask(sqlca.sqlerrtext,1)
```

```
    return sqlca.sqlcode
```

```
end if
```

gf_syserror input none, output none

```
messagebox("Error in Apex - Error  
Num:"+string(error.number)," message:"+error.text+' For  
' +error.windowmenu+' In '+error.objectevent)
```

```
int li_filename
```

```
li_filename=fileopen("\\ghplserver\sqlbackup\error.log",line
mode!,write!,shared!,append!)
```

```
if li_filename>0 then
```

```
    filewrite(li_filename,"Error inApplication - Error
Num:"+string(error.number)+" message:"+error.text+"
where:"+error.windowmenu+" Object:"+error.object+"
event:"+error.objectevent+" Line:"+string(error.line)+"
Dt/Tm:"+string(today())+' '+string(now()))
```

```
    filewrite(li_filename," ")
```

```
end if
```

```
fileclose(li_filename)
```

```
gf_verifydt input date xdate, returns integer
```

```
if xdate >std_date or xdate <= today() then
```

```
    return 0
```

```
else
```

```
    return 5
```

```
end if
```

```
return 0
```

```
s_prnset input datawindow dw_a,double default
```

```
//structure for printer set to be sent to printer window
```


Chapter 2

Now you are ready with master widows. To create windows with functionality you have to inherit one of the master windows you have developed.

Click on the inherit button beside new button and select the single datawindow window. Save the window. Create a data window from the table you want to use as a master data entry. Then another datawindow to list the data of the master.

Place these on the window and adjust the datawinodws.

The window should look something like this.



Itemchanged event for dw_1

```
er_flag = super:: event itemchanged(row,dwo,data)
choose case er_flag
    case 0
    case else
        return 1
end choose
```

No more coding is required to run this window.

So easy to develop the windows for master entry.

Next let us use two datawindow window to develop master, detail window for entering the data.

For this window select the double datawindow from inherit.

Prepared two datawindows one for the master data in free form and the second datawindow with details data in tabular form so data can be entered in rows.

Place the two datawindows on the window so created, it should look something like this.

Code for editchanged event for dw_1



```
dwo.name='ref_no' then
```

```
    if data<='9' then
```

```
        dw_1.getchild("ref_no",dwc)
```

```
        dwc.settransobject(sqlca)
```

```
        dwc.setfilter("policy_no like
```

```
%'"+UPPER(data)+"%")
```

```
        dwc.filter()
```

```
    ELSEIF trim(data)="" then
```

```
        dw_1.getchild("ref_no",dwc)
```

```
        dwc.settransobject(sqlca)
```

```
        dwc.setfilter("")
```

```

        dwc.filter()
    else
        dw_1.getchild("ref_no",dwc)
        dwc.settransobject(sqlca)
        dwc.setfilter("insured_name
like'%" +UPPER(data)+"%'")
        dwc.filter()

    end if
end if
itemchanged event for dw_1
er_flag = super:: event itemchanged(row,dwo,data)
choose case er_flag
    case 0
    case else
        return 1
end choose
string
remark1,remark2,remark3,remark4,cd,co_reg_cd,co_br_code
date dt,dt1,policy_from,policy_to
string ghpl_id1
long sno

if dwo.name="ref_no" then
policy_from=date(dw_1.getitemdatetime
(row,"policy_from"))
policy_to=date(dw_1.getitemdatetime(row,"policy_to"))
        ref_no=real(data)

```

```

ghpl_id=dw_1.getitemstring(row,"ghpl_id")+ "%"
SELECT ins_co_branch.co_cd
  INTO :ins_co
  FROM policy_head,
       ins_co_branch
  WHERE ( policy_head.ins_branch =
ins_co_branch.co_br_code ) and ref_no=:ref_no ;

```

```

SELECT count(*)
  INTO :ctr
  FROM pre_approvals
  WHERE ( pre_approvals.ghpl_id like :ghpl_id ) AND
        ( pre_approvals.adm_date between :policy_from and
:policy_to ) ;

          gf_messask(string(ctr),1)
  if ctr>0 then
      if g_login<>"00" then
          gf_messask("Pre-Authorisation Raised - Can
not Modify",1)

          dw_2.Object.DataWindow.ReadOnly="Yes"

          dw_1.Object.DataWindow.ReadOnly="Yes"
          else
          gf_messask("Pre-Authorisation Raised -
Should not Modify",1)

          end if
  end if

```

```

end if
choose case flag
    case "MG"
        if dwo.name="policy_from" then
            policy_from=date(left(data,10))
            policy_to=date(dw_1.getitemdatetime(row,"policy_to"))
            if daysafter(policy_from,policy_to)>366 then
                gf_messask("Cannot Change Date Range",1)
                return 1
            end if
        elseif dwo.name="policy_to" then
            policy_to=date(left(data,10))
            policy_from=date(dw_1.getitemdatetime(row,"policy_from"))
            if daysafter(policy_from,policy_to)>366 then
                gf_messask("Cannot Change Date Range",1)
                return 1
            end if
        end if
    case "VG"

    CASE "DG"
dt=date(dw_1.getitemdatetime(1,"policy_from"))

dt1=date(dw_1.getitemdatetime(1,"policy_to"))
ghpl_id=dw_1.getitemstring(1,"ghpl_id")+ '%'
SELECT count(claims.cl_no )
INTO :ctr
FROM claims,
    pre_approvals

```

```

WHERE ( pre_approvals.pre_app_no = claims.pre_app_no )
and
      ( ( claims.adm_date between :dt and :dt1 ) AND
        ( pre_approvals.ghpl_id like :ghpl_id ) ) ;
if ctr>0 then
gf_messask("This Policy Contains Claims - Cannot be
deleted",1)
          uo_1.pb_save.enabled=false
end if
case "A"
if dwo.name='policy_no' then
  if LenA(data)<6 then
    gf_messask("Please enter valid Policy No",1)
    return 1
  end if
  SELECT count(*) INTO :ctr
  FROM policy_head WHERE policy_head.policy_no = :data ;
  if ctr>0 then
    gf_messask("This Policy No is already
entered",1)
    return 1
  end if
          SELECT count(*) INTO :ctr
  FROM policy_head_ren WHERE policy_no = :data ;
  if ctr>0 then
    gf_messask("This Policy No is already entered
in Renewal",1)
    return 1
  end if

```

```

        cd=LeftA(data,6)+"%"
SELECT ins_co_branch.co_reg_cd,
       ins_co_branch.co_br_code ,ins_co_branch.co_cd
INTO :co_reg_cd,
     :co_br_code ,:ins_co
FROM ins_co_branch
WHERE ins_co_branch.co_ins_cd like :cd ;
      if sqlca.sqlcode=100 then
        gf_messask("Branch Code Not Found ",1)
        return 1

      end if
      dw_1.setitem(1,"ins_reg",co_reg_cd)
      dw_1.setitem(1,"ins_co",ins_co)

      dw_1.setitem(1,"ins_branch",co_br_code)
      if not isnull(co_br_code) then
        dw_1.setitem(1,"ins_co",ins_co)
        ghpl_id1='GHPL-'+ins_co+'%'
sno=dw_1.getitemnumber(1,"ref_no")
        ghpl_id='GHPL-'+ins_co+'-
'+right('00000'+string(sno),5)
        dw_1.setitem(1,"ghpl_id",ghpl_id)
ghpl_id=dw_1.getitemstring(row,"ghpl_id")
        dw_2.setitem(1,"ghpl_id",ghpl_id+'-1')
        dw_2.setitem(1,"slno",1)
        if dw_2.rowcount()=1 then dw_2.insertrow(0)

      end if

```

```

        if dwo.name='ins_co' then
            ins_co=data
            ghpl_id1='GHPL-'+ins_co+'%'
            SELECT
max(convert(float,substring(policy_det.ghpl_id,10,5)) )
            INTO :sno
            FROM policy_det
            WHERE policy_det.ghpl_id like :ghpl_id1 ;
                if isnull(sno) then sno=0
                    sno=sno+1
                    ghpl_id='GHPL-'+ins_co+'-'
'+RightA('00000'+string(sno),5)
                    dw_1.setitem(1,"ghpl_id",ghpl_id)
                    dw_2.setitem(1,"ghpl_id",ghpl_id+'-1')
                    dw_2.setitem(1,"sno",1)
                if dw_2.rowcount()=1 then dw_2.insertrow(0)
                elseif dwo.name="insured_name" then
                    gf_messask(data,1)
                    dw_2.setitem(1,"name_insured",data)
//      dw_2.setitem(1,"relation","SELF")
            end if
        end if
    end choose
choose case dwo.name
    case "co_cd"
        dw_1.getchild("co_reg_cd",dwc)
        dwc.settransobject(sqlca)
        dwc.setfilter("co_cd='"+data+"'")
        dwc.filter()

```



```

        case "policy_from"
            dt=date(LeftA(data,10))
if dt>today() then
gf_messask("Policy From Date greater than Today",1)
            end if
            dt1=gf_getnextyear(dt)
            dw_1.setitem(row,"policy_to",dt1)
            if dt<date('2004/06/01') then
                gf_messask("Please Check the date",1)
                return 1
            end if
        case "net_premium"
            amt=real(data)
            amt=round(amt*0.1236,0)
            dw_1.setitem(row,"service_tax",amt)
        case "prev_policy_no"
            SELECT count(*)

INTO :ctr
FROM policy_head
WHERE policy_head.policy_no = :data ;
if ctr>0 then
            SELECT policy_head.ref_no
INTO :ctr
FROM policy_head
WHERE policy_head.policy_no = :data ;
            gf_messask("This Prev.Policy is entered in
"+string(ctr)+" Please use Renewal",1)
elseif LeftA(data,4)='2004' or LeftA(data,4)='2005' then
gf_messask("Please enter policy no in format",1)

```

end if

end choose

if dwo.name="insured_name" then

 dw_2.setitem(1,"name_insured",data)

end if

Itemchanged event for dw_2

er_flag = super:: event itemchanged(row,dwo,data)

choose case er_flag

 case 0

 case else

 return 1

end choose

date dt

int age

choose case dwo.name

 case "name_insured"

 if dw_2.rowcount()=row then dw_2.insertrow(0)

 dw_2.setitem(row,"slno",row)

 dw_2.setitem(row,"ghpl_id",trim(dw_1.getitemstring
(1,"ghpl_id"))+'-'+string(row))

 CASE "sum_insured"

 dw_2.setitem(row,"bal_amount",real(data))

 case "cum_prc"

 dw_2.setitem(row,"bonus_amount",dw_2.getitemnu
mber(row,"sum_insured")*real(data)/100)

 case "cum_prc1"

```

        dw_2.setitem(row,"bon_amt1",dw_2.getitemnumber
(row,"sum_ins1")*real(data)/100)
        case "cum_prc2"
            dw_2.setitem(row,"bon_amt2",
dw_2.getitemnumber(row,"sum_ins2")*real(data)/100)
            case "occupation"
                dw_1.getchild('occupation',dwc)
                dwc.settransobject(sqlca)
            case "relation"
            CASE "dob"
                dt=date(left(data,10))
                if dt>date('1900/01/01') then
                    age=year(today())-year(dt)
                    dw_2.setitem(row,"age",age)
                end if
            case "sex"
                if data="M" then
                elseif data='F' then
                else
                    gf_messask("Wrong Sex",1)
                    er_flag=1
                    return 1
                end if
        end choose
    IF ins_co="NIA" or left(ins_co,1)="U" THEN
    else
    choose case dwo.name
        CASE "bonus_amount"

```

```

        amt=real(data)

        dw_2.setitem(row,"cum_prc",amt/dw_2.getitemnum
ber(row, "sum_insured")*100)
        case "bon_amt1"
            amt=real(data)

            dw_2.setitem(row,"cum_prc1",amt/dw_2.getitemnu
mber(row, "sum_ins1")*100)
            case "bon_amt2"
                amt=real(data)

                dw_2.setitem(row,"cum_prc2",amt/dw_2.getitemnu
mber(row, "sum_ins2")*100)

end choose
end if
// most of the above coding is based on requirements for
evaluating the data enterd. Choose case dwo.name reflects
the data to be verified.

```

Code for cancel event for the user object buttons

```

dw_1.settaborder("ghpl_id",0)
        SELECT max(policy_head.ref_no )
        INTO :vouno
        FROM policy_head ;
        if isnull(vouno) then vouno=0
        vouno=vouno+1
        dw_1.setitem(1,"ref_no",vouno)
        ghpl_id='GHPL-'+ins_co+'-
'+RightA('00000'+string(vouno),5)

```

```
dw_1.setitem(1,"ghpl_id",ghpl_id)
```

```
if not isnull(policy_no) then  
dw_1.setitem(1,"policy_no",LeftA(policy_no,12))  
if not isnull(agentcd) then dw_1.setitem(1,"agentcd",  
agentcd)  
dw_1.getchild("policy_no",dwc)  
dwc.settransobject(sqlca)  
dwc.setfilter("")  
dwc.filter()
```

Savebefore event, to check the data entered before passing onto save.

```
string  
path,insured_name,add1,add2,add3,city,pincode,bank_name,  
bank_acno,bank_city,ins_branch,izone,ph_no  
string name_ac_holder,bank_ifsc_code,bank_ac_type,email,  
bank_address  
if left(flag,1)<>"D" then  
dwitemstatus st  
string mess  
int ret  
st=dw_1.getitemstatus(1,"ph_no",primary!)  
if st=datamodified! or st=newmodified! then  
mess="Your Mediclaim Policy been registered with us  
and your ref.no. "+dw_1.getitemstring(1,"ghpl_id")  
ph_no=dw_1.getitemstring(1,"ph_no")  
if len(ph_no)=12 then  
ret=gf_smsdata(ph_no,mess)  
end if  
end if
```

```

if left(flag,1)="M" then
    ref_no=dw_1.getitemnumber(1,"ref_no")
    SELECT ins_co_branch.co_cd
INTO :ins_co
FROM policy_head,
    ins_co_branch
WHERE ( policy_head.ins_branch =
ins_co_branch.co_br_code ) and ref_no=:ref_no ;

```

end if

```

    ins_branch=dw_1.getitemstring(1,"ins_branch")
    SELECT ins_co_branch.co_cd
INTO :ins_co
FROM ins_co_branch
WHERE ins_co_branch.co_br_code = :ins_branch ;
else
ref_no=dw_1.getitemnumber(1,"ref_no",delete!,true)
ghpl_id=dw_1.getitemstring(1,"ghpl_id",delete!,true)
ttype='POLICY ENTRY INDIVIDUAL -0'
ctr=gf_deletedata()
end if

```

if flag='A' then

```

    vouno=dw_1.getitemnumber(1,"ref_no")
    ctr=vouno
    SELECT max(policy_head.ref_no )
INTO :vouno FROM policy_head ;
    if isnull(vouno) then vouno=0
    vouno=vouno+1

```

```

dw_1.setitem(1,"ref_no",vouno)
dw_1.setitem(1,"finyear",finyear)

for ctr=1 to dw_2.rowcount()

dw_2.setitem(ts,"ref_no",vouno)
next

// if ctr<>vouno then
ghpl_id='GHPL-'+ins_co+'-
'+RightA('00000'+string(vouno),5)
dw_1.setitem(1,"ghpl_id",ghpl_id)
for ts=1 to dw_2.rowcount()
ghpl_id='GHPL-'+ins_co+'-'+
RightA('00000'+string(vouno),5) + '-' +string(ts)
dw_2.setitem(ts,"ghpl_id",ghpl_id)
dw_2.setitem(ts,"ref_no",vouno)
dw_2.setitem(ts,"finyear",finyear)
next

gf_messask("Press Enter to Update",1)
vouno=dw_1.getitemnumber(1,"ref_no")
insured_name=dw_1.getitemstring(1,"insured_name")
add1=dw_1.getitemstring(1,"add1")
add2=dw_1.getitemstring(1,"add2")
add3=dw_1.getitemstring(1,"add3")
city=dw_1.getitemstring(1,"city")
pincode=dw_1.getitemstring(1,"pincode")
bank_name=dw_1.getitemstring(1,"bank_name")
bank_acno=dw_1.getitemstring(1,"bank_acno")
bank_city=dw_1.getitemstring(1,"bank_city")

```

```

        izezone=dw_1.getitemstring(1,"izezone")
        ph_no=dw_1.getitemstring(1,"ph_no")
name_ac_holder=dw_1.getitemstring(1,"name_ac_holder")
bank_ifsc_code=dw_1.getitemstring(1,"bank_ifsc_code")
bank_ac_type=dw_1.getitemstring(1,"bank_ac_type")
email=dw_1.getitemstring(1,"email")
bank_address=dw_1.getitemstring(1,"bank_address")

```

```

SELECT count(*) INTO :ctr
FROM policy_address WHERE policy_address.ref_no =
:vouno ;
if ctr>0 then
UPDATE policy_address
SET insured_name = :insured_name, add1 = :add1,
add2 = :add2,
add3 = :add3, city = :city, pincode = :pincode,
usrcd = :g_login, sysdate = :sysdatetime, ghpl_id =
:ghpl_id,
bank_name = :bank_name, bank_acno = :bank_acno,
bank_city = :bank_city ,izezone=:izezone, fincode=:finyear,
ph_no=:ph_no, name_ac_holder=:name_ac_holder,
bank_ifsc_code=:bank_ifsc_code,
bank_ac_type=:bank_ac_type,
email=:email,
bank_address=:bank_address
WHERE policy_address.ref_no = :vouno and
isnull(fincode,0)>=:finyear ;
UPDATE policy_address

```



```

        SET   usrcd = :g_login,      sysdate = :sysdatetime,
ghpl_id = :ghpl_id,
        bank_name = :bank_name,      bank_acno =
:bank_acno,      bank_city = :bank_city
,izone=:izone,fincode=:finyear,ph_no=:ph_no
        , name_ac_holder=:name_ac_holder,
bank_ifsc_code=:bank_ifsc_code,
bank_ac_type=:bank_ac_type,
email=:email,
bank_address=:bank_address
        WHERE policy_address.ref_no = :vouno ;

```

else

```

                INSERT INTO policy_address
        ( ref_no,      insured_name,      add1,
add2,      add3,      city,      pincode,
        usrcd,      sysdate,      ghpl_id,
bank_name,      bank_acno,
bank_city,izone,fincode,ph_no,name_ac_holder,
bank_ifsc_code,bank_ac_type,email,bank_address )
        VALUES ( :vouno,      :insured_name,      :add1,
:add2,      :add3,      :city,      :pincode,
        :g_login,      :sysdatetime,      :ghpl_id,
:bank_name,:bank_acno,:bank_city,:izone,0,
:ph_no,:name_ac_holder,
:bank_ifsc_code,:bank_ac_type,:email,:bank_address ) ;
        end if

        for ctr=1 to dw_2.rowcount()

                ghpl_id=dw_2.getitemstring(ctr,"ghpl_id")

```

```
        if ins_co="NIA" then
            if ctr<10 then

                path="\\newserver\d\ghpl_img_nia\"+RightA(trim(
ghpl_id ),7)+".jpg"
                    else

                path="\\newserver\d\ghpl_img_nia\"+RightA(trim(
ghpl_id ),8)+".jpg"
                    end if
            elseif left(ins_co,1)="O" then
                if ctr<10 then

                    path="\\newserver\d\ghpl_img_oih\"+RightA(trim(
ghpl_id ),7)+".jpg"
                        else

                    path="\\newserver\d\ghpl_img_oih\"+RightA(trim(
ghpl_id ),8)+".jpg"
                        end if
                elseif ins_co="UIH" or ins_co="UIA" then
                    if ctr<10 then

                        path="\\newserver\d\ghpl_img_uim\"+RightA(trim(
ghpl_id ),7)+".jpg"
                            else

                        path="\\newserver\d\ghpl_img_uim\"+RightA(trim(
ghpl_id ),8)+".jpg"
                            end if
                    elseif left(ins_co,2)="UI" THEN
```

```

if ctr<10 then

    path="\\newserver\d\ghpl_img_uim\"+RightA(trim(
ghpl_id ),7)+".jpg"
        else

            path="\\newserver\d\ghpl_img_uim\"+RightA(trim(
ghpl_id ),8)+".jpg"
                end if
        elseif left(ins_co,1)="N" THEN
            if ctr<10 then

                path="\\newserver\d\ghpl_img_nat\"+RightA(trim(
ghpl_id ),7)+".jpg"
                    else

                        path="\\newserver\d\ghpl_img_nat\"+RightA(trim(
ghpl_id ),8)+".jpg"
                            end if
                elseif left(ins_co,1)="R" THEN
                    if ctr<10 then

                        path="\\newserver\d\ghpl_img_rgi\"+RightA(trim(
ghpl_id ),7)+".jpg"
                            else

                                path="\\newserver\d\ghpl_img_rgi\"+RightA(trim(
ghpl_id ),8)+".jpg"
                                    end if
                    END IF

```

```
        dw_2.setitem(ctr,"photo_path",path)
next
elseif LeftA(flag,1)='M' then
        vouno=dw_1.getitemnumber(1,"ref_no")

insured_name=dw_1.getitemstring(1,"insured_name"
)
        add1=dw_1.getitemstring(1,"add1")
        add2=dw_1.getitemstring(1,"add2")
        add3=dw_1.getitemstring(1,"add3")
        city=dw_1.getitemstring(1,"city")
        pincode=dw_1.getitemstring(1,"pincode")

bank_name=dw_1.getitemstring(1,"bank_name")

bank_acno=dw_1.getitemstring(1,"bank_acno")
        bank_city=dw_1.getitemstring(1,"bank_city")
        ize=dw_1.getitemstring(1,"ize")
        ph_no=dw_1.getitemstring(1,"ph_no")

name_ac_holder=dw_1.getitemstring(1,"name_ac_holder")
)

bank_ifsc_code=dw_1.getitemstring(1,"bank_ifsc_code")
)

bank_ac_type=dw_1.getitemstring(1,"bank_ac_type")
        email=dw_1.getitemstring(1,"email")

bank_address=dw_1.getitemstring(1,"bank_address")
```

```

UPDATE policy_address
  SET insured_name = :insured_name,  add1 = :add1,
  add2 = :add2,
  add3 = :add3,  city = :city,  pincode = :pincode,
  usrcd = :g_login,  sysdate = :sysdatetime,
  ghpl_id = :ghpl_id,
  bank_name = :bank_name,  bank_acno =
:bank_acno,
  bank_city = :bank_city
,izone=:izone,ph_no=:ph_no
  ,name_ac_holder=:name_ac_holder,
bank_ifsc_code=:bank_ifsc_code,
bank_ac_type=:bank_ac_type,
email=:email,
bank_address=:bank_address

  WHERE policy_address.ref_no = :vouno and
fincode>=:finyear ;

end if
if LeftA(flag,1)='M' then
  for ctr=1 to dw_2.rowcount()
//    dw_2.setitem(ctr,"slno",ctr)
dw_2.setitem(ctr,"ref_no",vouno)

    ghpl_id=dw_2.getitemstring(ctr,"ghpl_id")
      if ins_co="NIA" then
        if ctr<10 then

```

```
path="\\newserver\d\ghpl_img_nia\"+RightA(trim(
ghpl_id ),7)+".jpg"
```

```
else
```

```
path="\\newserver\d\ghpl_img_nia\"+RightA(trim(
ghpl_id ),8)+".jpg"
```

```
end if
```

```
ELSEif ins_co="UIH" or ins_co='UIC' OR ins_co='UIA'
then
```

```
if ctr<10 then
```

```
path="\\newserver\d\ghpl_img_uih\"+RightA(trim(
ghpl_id ),7)+".jpg"
```

```
else
```

```
path="\\newserver\d\ghpl_img_uih\"+RightA(trim(
ghpl_id ),8)+".jpg"
```

```
end if
```

```
elseif left(ins_co,1)="O" then
```

```
if ctr<10 then
```

```
path="\\newserver\d\ghpl_img_oih\"+RightA(trim(
ghpl_id ),7)+".jpg"
```

```
else
```

```
path="\\newserver\d\ghpl_img_oih\"+RightA(trim(
ghpl_id ),8)+".jpg"
```

```
end if
```

```
ELSEif ins_co="UIM" then
```

```
if ctr<10 then
```

```
path="\\newserver\d\ghpl_img_uim\"+RightA(trim( ghpl_id
),7)+".jpg"
```

```
    else
```

```
path="\\newserver\d\ghpl_img_uim\"+RightA(trim( ghpl_id
),8)+".jpg"
```

```
    end if
```

```
elseif left(ins_co,1)="N" THEN
```

```
    if ctr<10 then
```

```
        path="\\newserver\d\ghpl_img_nat\"+RightA(trim(
ghpl_id ),7)+".jpg"
```

```
        else
```

```
            path="\\newserver\d\ghpl_img_nat\"+RightA(trim(
ghpl_id ),8)+".jpg"
```

```
        end if
```

```
end if
```

```
    if
```

```
dw_2.getitemstatus(ctr,0,primary!)=newmodified! then
```

```
        dw_2.setitem(ctr,"photo_path",path)
```

```
    end if
```

```
    string p_path
```

```
    p_path=dw_2.getitemstring(ctr,"photo_path")
```

```
    if isnull(p_path) then p_path=""
```

```
    if p_path="" then
```

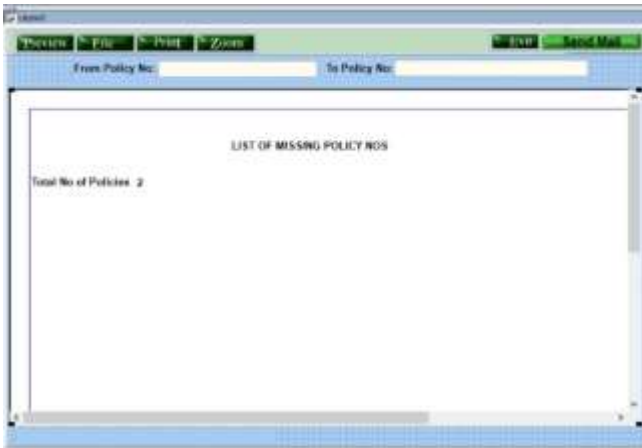
```
        dw_2.setitem(ctr,"photo_path",path)
```

```
    end if
```

```

next
policy_no=dw_1.getitemstring(1,"policy_no")
agentcd=dw_1.getitemstring(1,"agentcd")
end if
Saveafter event will decide whether to send the data for
updatation
string membercode,ph_no,mess
int ret
if ins_co='NIA' then
for ctr=1 to dw_2.rowcount()
    ghpl_id=dw_2.getitemstring(ctr,"ghpl_id")
    membercode=dw_2.getitemstring(ctr,"membercode"
)
    if not isnull(membercode) and trim(membercode)<>"
then
        UPDATE policy_det_ren
        SET membercode = :membercode
        WHERE policy_det_ren.ghpl_id = :ghpl_id ;
end if
next
end if
//now the window is ready for entering data in master, detail
form.
//It is so easy to develop windows and code with
PowerBuilder.
This way entire application can be made.
Now for reports with retrieval arguments

```

Here report data can be retrieved using the small datawindow

Code for itemchanged event for dw_2

```

delete pol_nos;
string pol_no,topol_no,st_no,policy_no,refno
decimal fr_no,to_no,cnt,len,lpos,npos
if dwo.name="prev_policy_no" then
    pol_no=trim(dw_2.getitemstring(1,"policy_no"))
    topol_no=trim(data)
    len=len(pol_no)
    lpos=0
    ctr=1
    for ctr=1 to len
        if mid(pol_no,ctr,1)="/" then
            lpos=ctr
        end if
    next
    st_no=left(pol_no,lpos)
    fr_no=dec(right(pol_no,len-lpos))

```

```

to_no=dec(right(topol_no,len-lpos))

        policy_no=st_no+RightA('00000000'+string(cnt),len-
lpos)

                SELECT min(policy_head.ref_no )
INTO :ctr
FROM policy_head
WHERE policy_head.policy_no = :policy_no ;
        if isnull(ctr) then ctr=0
        if ctr=0 then
                SELECT min(ref_no )
INTO :ctr
FROM policy_head_ren
WHERE policy_no = :policy_no ;
                if isnull(ctr) then ctr=0
                end if
refno=string(ctr)
                INSERT INTO pol_nos
( pol_no ,ref_no) VALUES ( :policy_no ,:refno) ;
        next
        dw_1.retrieve()
end if

```

Chapter 3

Here I will explain some of the commands which will be useful in developing software.

Datawindow commands:

`Dw_1.insertrow(0)` : inserts a row in datawindow `dw_1`

`Dw_1.deletetow(rowno)` marks for deletion the `rowno`

`Dw_1.update()` updates the datawindow to the table

If status of the `dw_1.getitemstatus(rowno,0,primary!)` is `new!` the row will not be updated. If the status is `NewModified` the row will be inserted into the table. If the status is `DataModified` the row data is updated to the table. If `NotModified!` Nothing will happen.

Same way `dw_1.setitemstatus(rowno,colno,primary!,status)` can reset the status of the row or column with status explained above.

There are many properties for datawindow which can be altered programitacally.

`Dw_1.accepttext()` updates the text entered in the datawindow. It will be useful when checking the data entered in the present row. Data as the `enteredvalue` can be checked after passing `accetedtext()`.

`Dw_1.getitem`, `dw_1.setitem` will get the data from a column and row and set the data back to them.

`Getitem` has many forms based on the data type:`getitemstring` will retrieve a string, `getitemnumber` retrieves a number and `getitemdate` or `getitemdatetime` will retrieve the dates or `datetime`.

If you check the `itemchanged` event in the master windows, you will find some of the commands used there for verifying data entered.

Dw_1.getchild command is useful for manipulating the dropdowns in the datawindow and any child windows.

```
Dw_1.setsort("colname asc/desc") //sort is defined
```

```
Dw_1.sort() //actual sort is performed
```

```
Dw_1.setfilter("colname=value") // set to filter out data
```

```
Dw_1.filter() //actula filtering performed
```

```
Dw_1.getsqlselect() //sql query used in the datawindow
```

```
Dw_1.setsqlselect("sql query") // set a new sql select query
```

```
//Check datawindow control in PB help get full list of  
datawindow functions.
```

```
//Same way properties can be managed by code in the runtime  
or program in the columns of the data window.
```

For example you want to manipulate the visibility of the column based on some logic. It can be coded on the datawindow itself or at the runtime in the window script.

```
Dw_1.describe(propertylist)//Use Describe to understand the  
structure of a DataWindow. For example, you can find out  
which bands the DataWindow uses and what the datatypes of  
the columns are. You can also use Describe to find out the  
current value of a property and use that value to make further  
modifications at runtime.
```

```
Ex: string ls_data_type = dw_1.Describe("salary.ColType")
```

It will return the column type of salary.

Datastore: Datastore is a nonvisual datawindow control.

There are many uses for datastore. Holding the data in datastore can be distributed across many datawindows.

Nested objects on datawindow.

Rowscopy and rowsmove. Rows can be copied between datawindows and moved across.

Script for the Browse button

The script for the Browse button creates an HTML string from the data in the DataWindow by assigning the Data.HTMLTable property to a string variable. After constructing the HTML string, the script adds a header to the HTML string. Then the script saves the HTML to a file and runs the Web browser to display the output.

```
String ls_HTML, ls_FileName, ls_BrowserPath
Integer li_FileNumber, li_Bytes,
Integer li_RunResult, li_Result

// Generate the HTML from datawindow.
ls_HTML = dw_1.Object.DataWindow.Data.HTMLTable
IF IsNull(ls_HTML) Or Len(ls_HTML) <= 1 THEN
    MsgBox ("Error", "Error generating HTML!")
    Return
ELSE
    ls_HTML ="<H1>HTML Generated From a DataWindow"&
        + "</H1><P>" + ls_HTML
END IF

//Create the file.
ls_FileName = "custlist.htm"
li_FileNumber = FileOpen(ls_FileName, StreamMode!, &
    Write!, LockReadWrite!, Replace! )

IF (li_FileNumber >= 0) THEN
    li_Bytes = FileWrite(li_FileNumber, ls_HTML)
    FileClose(li_FileNumber)
    IF li_Bytes = Len(ls_HTML) THEN
```

```

// Run Browser with the HTML file.
IF Not FileExists(is_Browser) THEN
    cb_selbrowser.Trigger Event Clicked()
    IF NOT FileExists(is_Browser) THEN
MessageBox("Select Browser", "Could &
not find the browser.")
    RETURN
    END IF
END IF
li_RunResult = Run(is_Browser + " file:///"+&
    ls_FileName)
IF li_RunResult = -1 THEN
MessageBox("Error", "Error running browser!")
    END IF
ELSE
    MessageBox ("Write Error", &
        "File Write Unsuccessful")
    END IF
ELSE
    MessageBox ("File Error", "Could not open file")
END IF

```

Did you know you can run sqlcommands from the windows. You can use paste sql button to generate sql command from the tables availale with the db connection. You can even declare cursors and run them in the windows.

The only problem you may encounter with MS Sql server is that you can't have two sql commands running simultanously. So you can use a datastore to run the second command or you can

create a new transaction object and run them, which can slow the connection a little.

You can access the events and functions from the inherited master with `.notation` parent.

You can even disable the parent code.

Chapter 4

PowerServer is a tool to distribute the PowerBuilder software through the internet to the users. The compiled .pbd will be stored and can be distributed to the users.

Chapter 5

.Net technology and PowerBuilder:

.Net technology was available from version 11 and it becomes more powerful day by day.

In version 11 you can convert the code to c# version and can load in IIS server and run as a web application. There was not much control over the generated software and it was a little slow to run. But now with subsequent versions and at present version 2022Beta2, they have come up with editing the generated code for errors in construction of the code.

It is interesting to check out this feature.

I will bring out the details of using this feature in subsequent books.

Chapter 6

Reports

There are many types of reports available with PowerBuilder.

You can prepare a report as a datawindow and put it on a window for the reports to be activated and print ready.

There are many formats available for reports.

1. Composite datawindows allowed more than one datawindow to be placed on the datawindow reports.
2. the report can be retrieved with the retrieval arguments called on the main datawindow and these or any other column value can be passed on to the datawindows to retrieve the reports.
3. CrossTab Reports are useful when preparing data from multiple values. Like branches and years data.
4. Freeform is used for the master data from a single row. This format is mainly used in data entry operations.
5. Graph is as the name suggests can prepare any type of graph from data supplied.
6. Grid is useful for letting the user have a look at the data in horizontal form and the length of the columns can be manipulated at run time. Useful for data entry or reports where data is presented in multiple rows.
7. Group reports with items grouping.
8. Label You can create Labels and print address and like.
9. N up The N-Up style presents two or more rows of data next to each other. It is similar to the Label style in that you can have information from several rows in the database across the page. However, the information is not meant to be printed on labels. The N-Up presentation style is useful if you have periodic data; you can set it up so that each period repeats in a row.
10. Ole2.0 External links can be attached to the datawindows.

11. Rich text formatting can be done

12. Tabular is similar to grid, but there are no grid lines and columns can't be automatically manipulated at runtime.

13. Treeview

It is interesting to work with many formats of data windows.

Epilogue

There are many commands available in PowerBuilder, which we can checkout when required. PowerBuilder has an excellent help feature. One can learn PB from searching for comands from the help. There is also help available online from Appeon Website. And there are online forums where help is available.

For source code and database schemas contact the writer: on these email ids: ravi_kp_com@yahoo.com

This project is explained in detail in the book:

How to do Software Programming With Examples by Ravi Kumar Paramkusam

You can get the book on the website:

<https://ravikiran.azurewebsites.net/>

All the links are provided in the above website.

This is a small project compared to others. This application was developed using PowerBuilder with sqlserver2000 on a windows server. But maintenance was done for 10 years. As requirements and new insurance companies were added, new type policies came, new processes were introduced by the insurance companies, constant modifications are necessary. Either in the application or in reports.

Web Application link to download developed using visual studio 2005, combidata.bak for sqlserver 2008. These are available in this location in zip format.

<https://www.dropbox.com/s/ent1690a1kfxq1g/Web%20Application.zip?dl=0>

PowerBuilder Application available at this location:

https://www.dropbox.com/s/x17g6in2bgcr987/rk_ghpl_pb11.zip?dl=0

For any other software links to practise contact the writer on above email.

About the Author

Ravi Kumar Paramkusam is a software professional with over 30 years experience. He has worked in many industries on multiple languages and domains.