

Visual Prolog

Lab 2..





Objectives:

- Continue Revision.
- Introduction to a GUI.
- GUI: Responding to Events.
- Starting a Simple GUI Project

Quick Revision

- The Whole program written with small letters **Except Variables** starting in capital letter.
- Each statement end with **.** Full stop except in **Run statements.**
- With each **run():-** statement there is **fail.**
- End your program with
- **_=stdIO::readChar().**
- **In declaration of facts and predicates :**
- **nameFacts: (arguments dataType).**
- **namePrdicates : (arguments dataType).**

Where I Put My Code

```
main.pro
22:1  Insert  Indent  Modified

*****/

implement main
  open core

constants
  className = "main".
  classVersion = "".

clauses
  classInfo(className, classVersion).

  class facts
  class predicates

clauses
  run():-
    console::init(),
    succeed().

  % place your own code here

end implement main

goal
  mainExe::run(main::run).
```

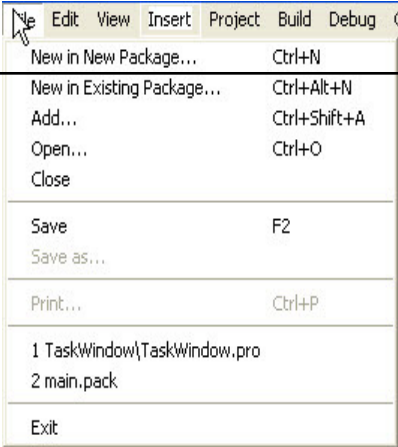
Introduction to a GUI



Ins.Reham AIMukhallafi

Introduction to a GUI

- A **GUI** is simply an acronym for **Graphical User Interface**.
- In the Windows operating system environment the term represents the familiar windows:

menu bar.	toolbar	Buttons
		

GUI In Programming Terms

- **GUI does two things :**
- It uses complex graphical routines to put graphical images on the relevant parts of the computer monitor.
- It also controls the behavior of the mouse and other input devices over these graphical areas.

GUI In Programming Terms :Cont

- Both these detailed programming are done by the operating system which provide **API** (Application Programming Interface) which can be used to setup the **GUI** required for any program.

Difference between Console & GUI

	<u>GUI</u>	<u>Console</u>
Graphical elements	Show	Do not show
Sequence of Activities	flexible	cannot modify
Knowledge & Control	Separate	Not separate

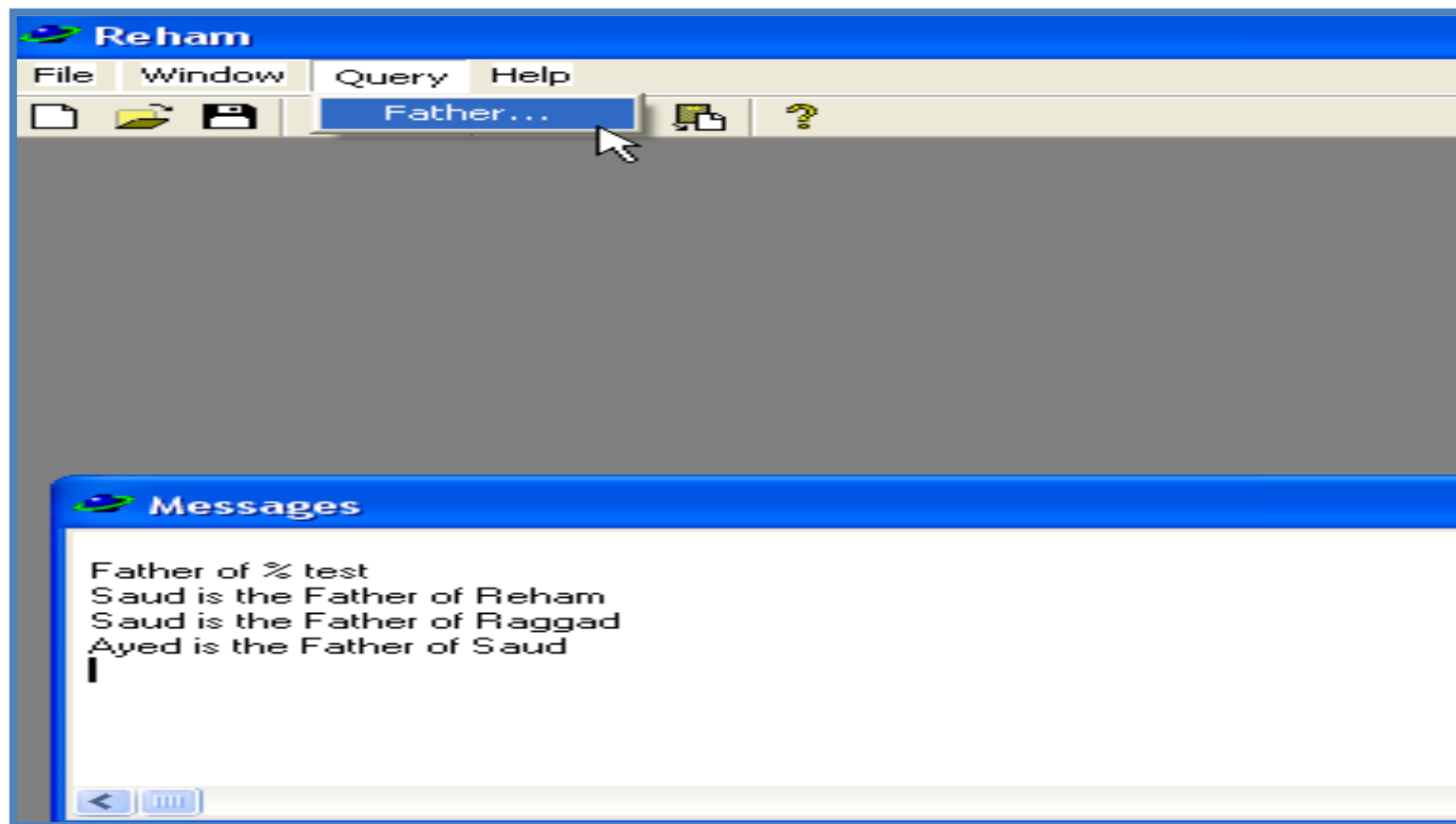
- **But on the other hand**, the programmer needs to know how each of the **GUI** components works

Responding to Events

- In a **GUI** program, all the **GUI** components wait for inputs from the keyboard and/or mouse.
- The information from such input devices is known as an **event**.
- And for each **event** there is a **listener**.

Simple GUI Project

- Just contain the **Main Menu**, **ToolBar** component.

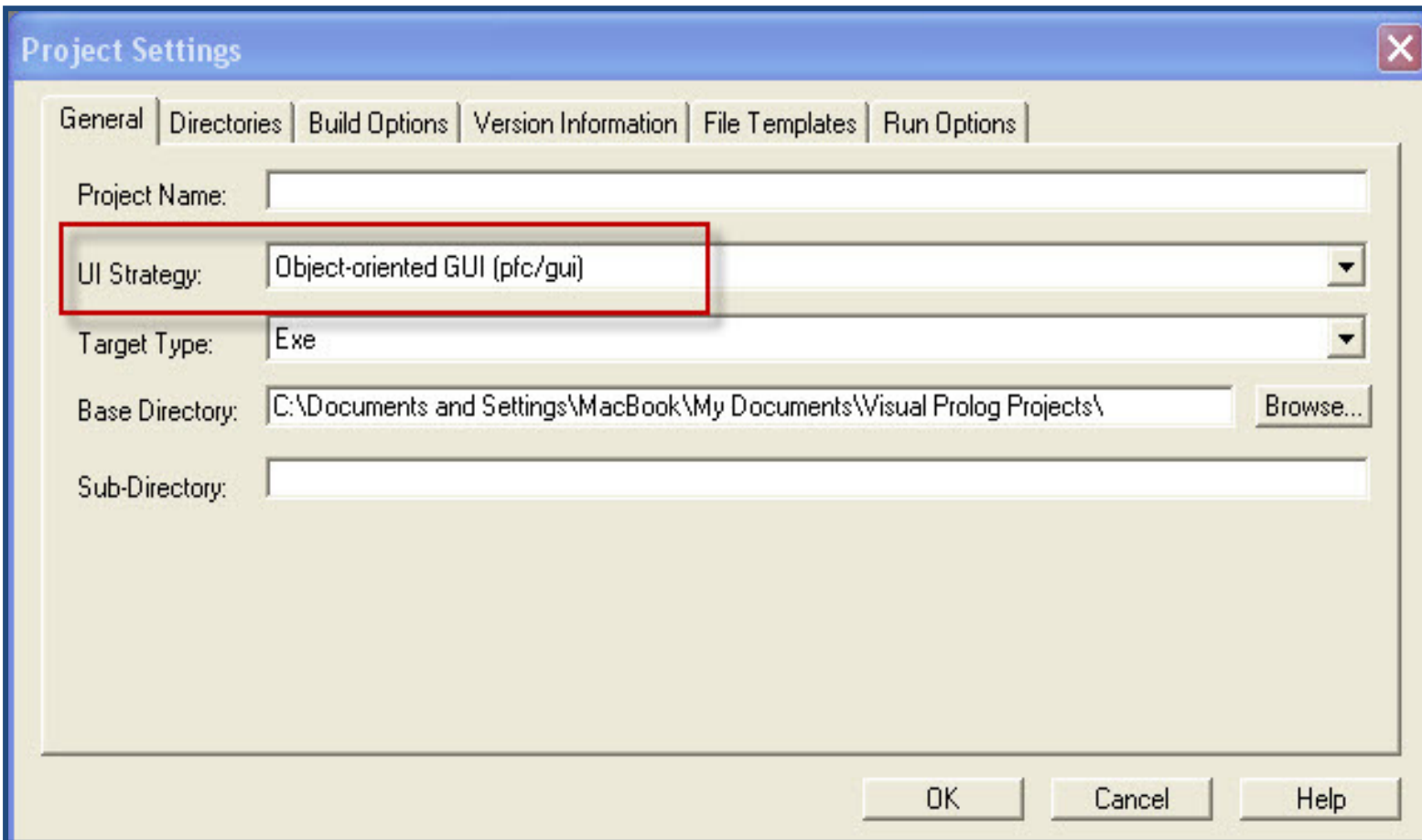



Starting a GUI Project



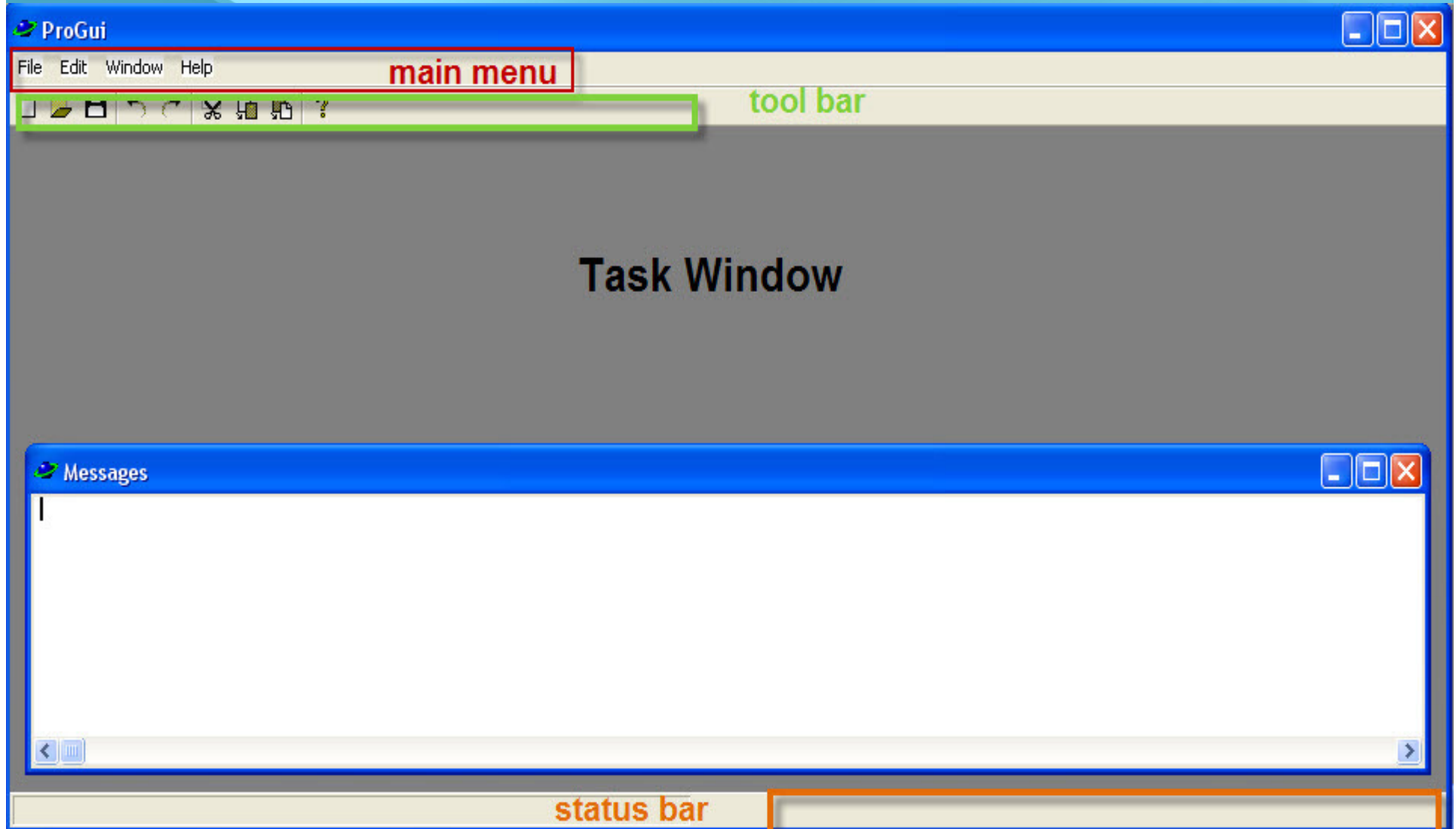
Ins.Reham AlMukhallafi

1 – Project Settings

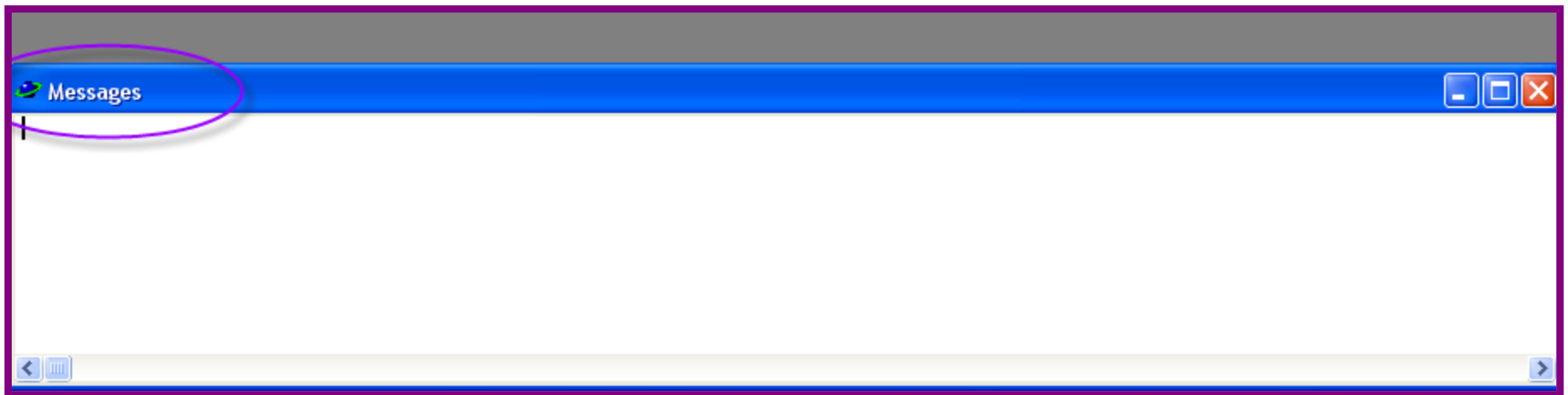


- 
- **When creates Project the initial set of modules:**
 - the main menu
 - one top toolbar
 - one bottom status bar
 - the main **Task Window** of the program.

GUI Project Components

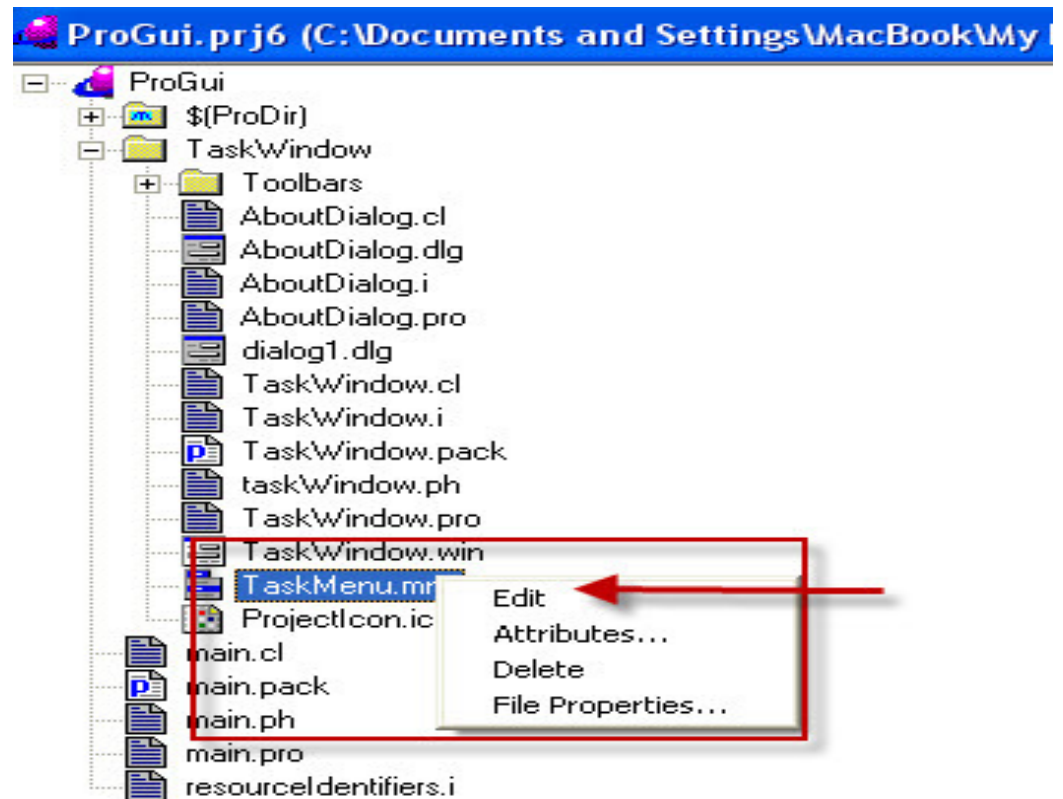


the application Visual Prolog gives another window titled **Messages**. This window is used internally to act as the *console*.



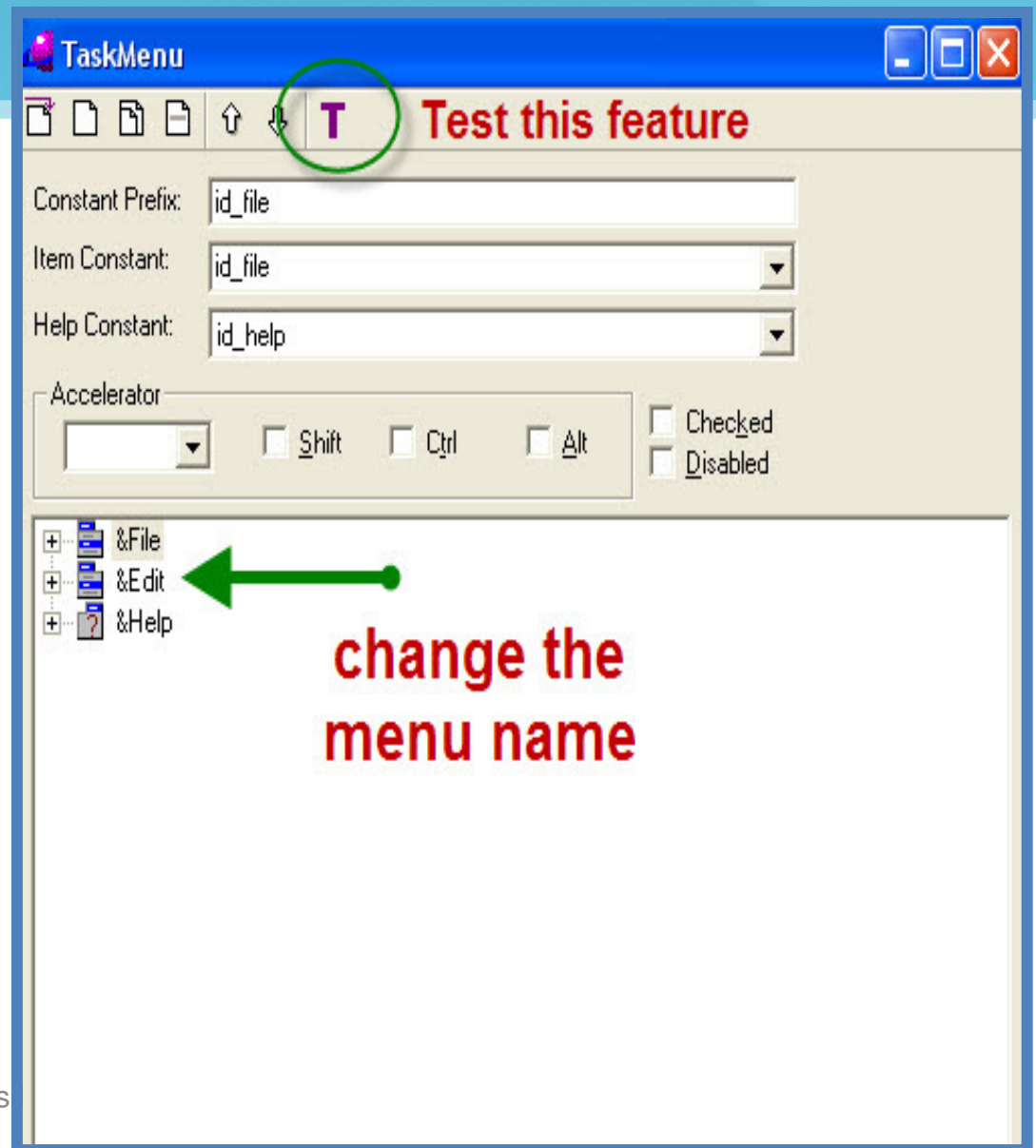
2- Modifying the Main Menu (TaskMenu.mnu)

- To Modify the menu :
- right click on the **TaskMenu.mnu** item as seen below:

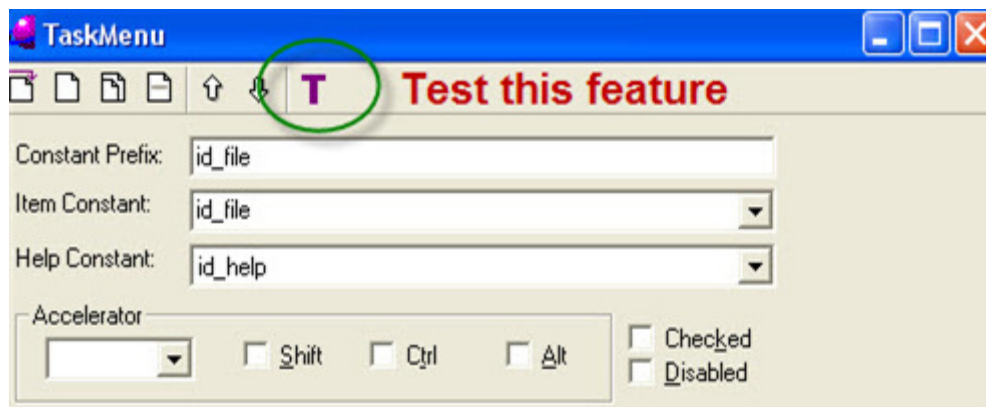


- **Constant Prefix:** used internally in the code to refer to the menu items.

To add **Subitem** ,
click on [New SubItem](#)



- Test the menu Click on T :



2- Modifying the Tool Bar (ProjectToolbar.tb)



- Choose in **constant filed**
The ID of your item
(**Father SubItem**)
- **In status text :**
There is tow parts breaks up with
semicolon:
1- tool-tip on the button
2- appear in the status-line
of the main window.



3- Getting the Main Code into the Program “TaskWindow.pro”

- Open [TaskWindow.pro](#) and put Logical Code .

class facts - familyDB

person : (string Name, string Gender).

parent : (string Person, string Parent).

clauses

person(“John”, “Male”).

Person (“Mike”, “Male”).

Parent(“John”, “Mike”).

class predicates

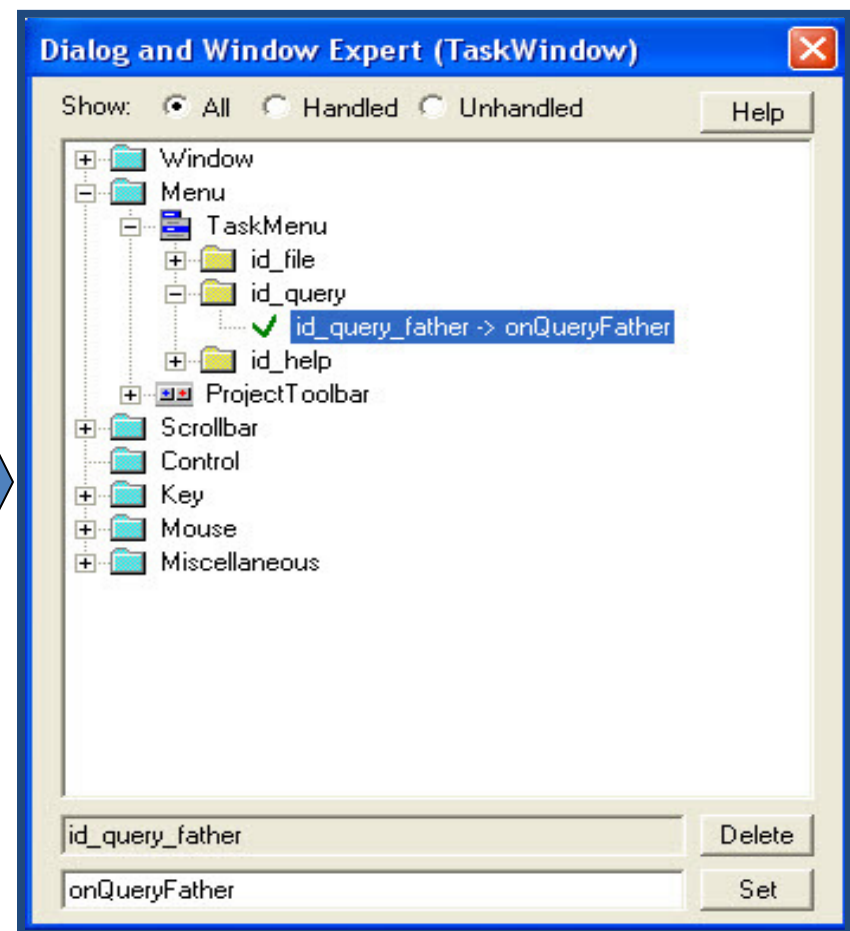
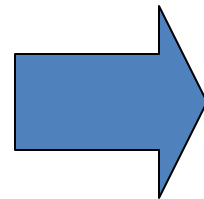
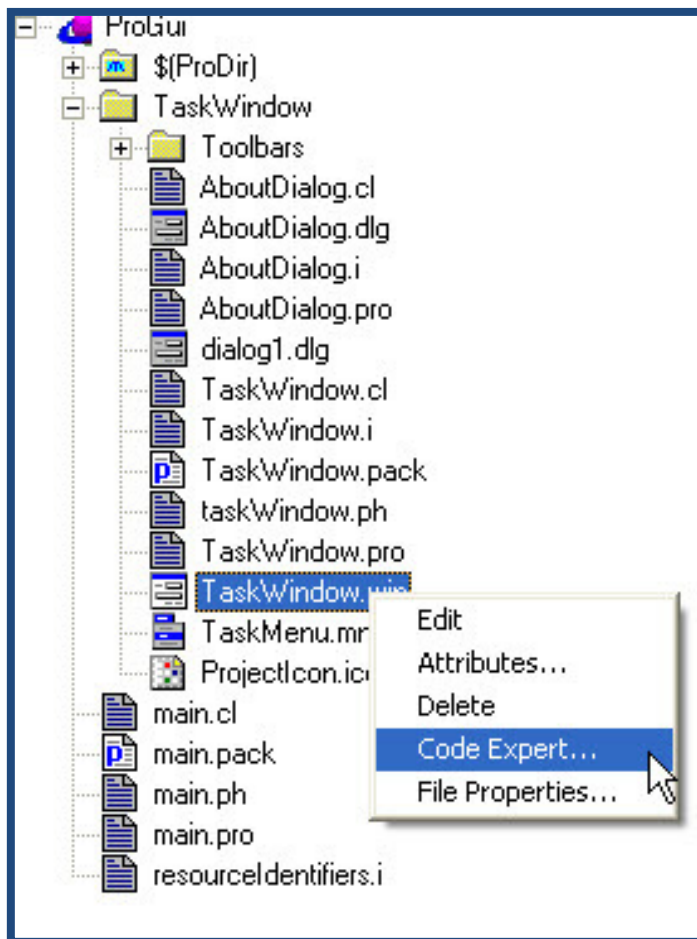
father : (string Person, string Father) nondeterm anyflow.

clauses

father(Person, Father) :-

parent(Person, Father), person(Father, “Male”).

Interactive between Codes & GUI Controls TaskWindow.win



- Double click on *id_query_Father* menu item and write this code :

predicates

onQueryFather : window::menuItemListener.

clauses

```
onQueryFather(_Source, _MenuTag):-  
stdIO::write("\nfather test\n"),  
father(X, Y),  
stdIO::writef("% is the father of %\n", Y, X),  
fail.
```

onQueryFather(_Source, _MenuTag).

- 
- **Any Question ..?**