

Chapter 4. Window Operations .....	2
4.1 Window Types.....	2
4.2 Window properties .....	7
4.3 Creating New Windows .....	9
4.4Opening Windows.....	12
4.5 Examples.....	14
4.6 Windows related object.....	24
4.7 Resource allocation of windows .....	24

## Chapter 4. Window Operations

Window is one of the most important and unique elements for EB500.



The above example shows creating multiple windows at the same time.

### 4.1 Window Types

Window is the basic element for an EB500 project. Every display is composed by many windows. There are three types of windows: Base, Fast selection and Common. The basic window can also be used as popup windows after changing the size. All windows can be assigned as underlay windows.

Basic Window:

These are the normal windows that would be displayed during HMI runtime operations. When using (change window ) command to order the base window that is switched over , will remove all display on the screen [ Except (Common Window) (Fast Selection Window) ] and then the basic window will show on the screen. As object on base window can be touched off and opened popup windows , the primitive information on base window will be kept at this moment. Being touched off

and opened popup windows will be added on base window , all windows of the same situation can be regarded as the set membership. Switching over from base window N to base window M, all Sub window of the base window N will be closed then show basic window M and its subsidiary window. The basic window must fill-up the size of the screen.

#### Fast Selection Window:

It is activated when the Task Button associated with it is pressed. The window displays on the screen unless it's hided from Task Button. So it can put Task Button which switch over the window or other commonly used object. It is designated as window number 4. When reassigning another window as Fast selection window, the size of that window must be the same as the Fast selection window.

#### Common Window:

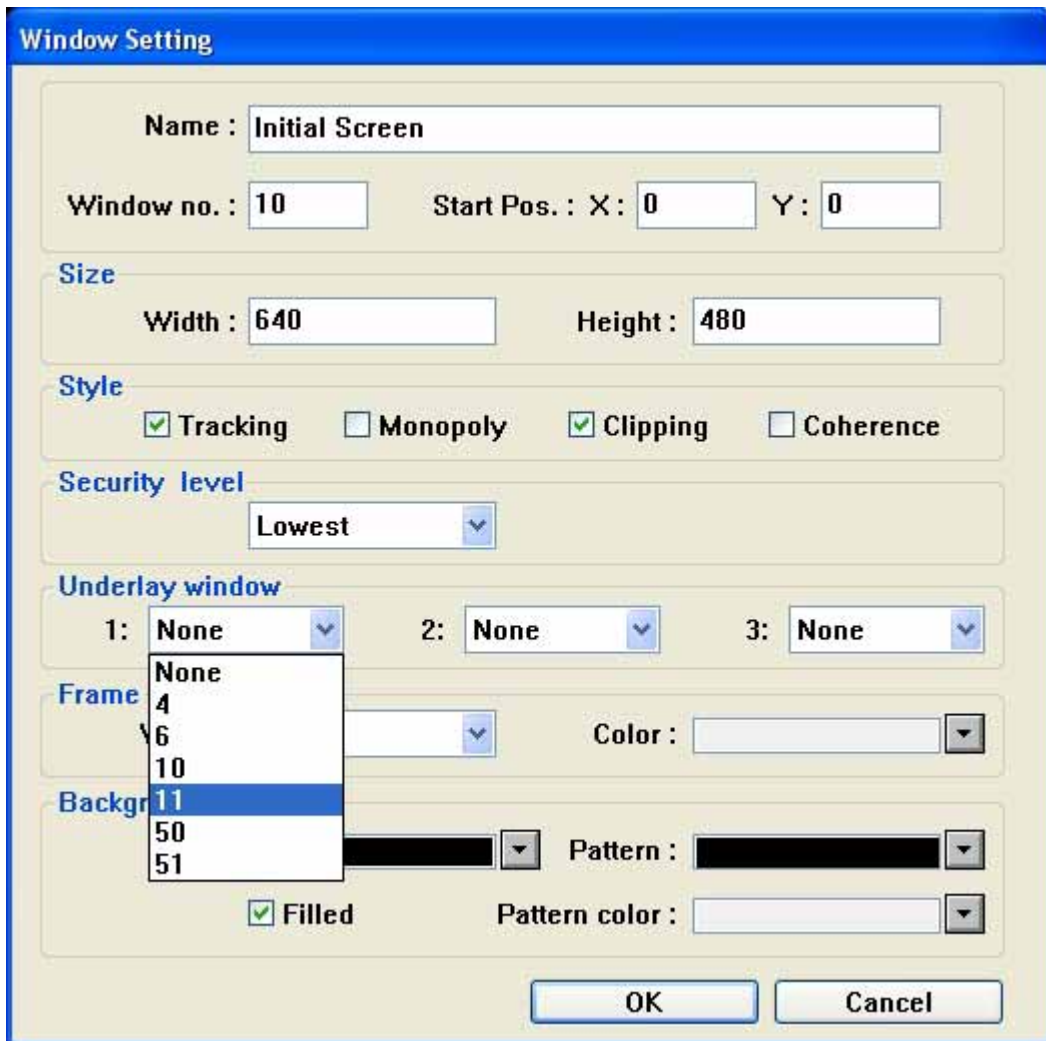
Common window shows on the screen all the time, object showing needing all the time is put in this window,and you can see the state or operation of object at any time. Common window is window 6. The Common window can be changed via Function key but here should be only one active common window.

#### Message broad window:

Each project has only one message broad window which is set from [System Parameter Setting]/[General]. The size of window can be set. If several windows use message broad window, actually they use the same message broad. Please refer to Ch.9 for further detail.

#### Underlay Window:

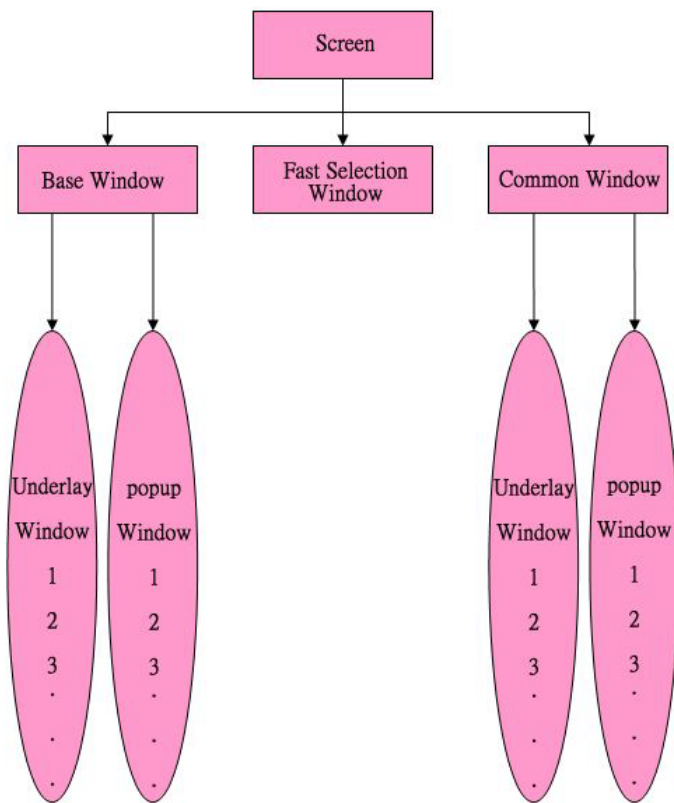
Up to three windows can be specified as underlay windows. The common objects used by many Windows are usually placed on the underlay window, such as background, graph and title. All windows can be assigned as underlay windows.



As showed above, assign the window as an underlay window by selecting the underlay window property. Underlay window is different from base window. The component of the window of ground floor is inserted to the general window while operating, but this window does not show in fact. It is the same as the component of the general window that its component shows and controls etc..

Screen-Window relationship:

a screen includes: common window, base window and fast window; a common window or a base window includes several popup windows.



When carry out (Changing Window ) command , will remove all windows on the screen (including the popup windows) and switch the window going to be displayed. The common window is always active. The popup windows coming from direct or indirect windows don't change the original content and are only attached to the base window. Maximum three underlay windows are available in window setting dialog.

When editing a base window (range of window numbers is from 10 to 1999), display shows all objects on the window. At the same time, if there is an underlay window under the base window, the base window shows all objects defined to the underlay window, such as control objects (set bit, function key, etc). Moreover, if show the object in the common window, can show on the present screen too, all control objects defined to common windows are effective on the active window.

There are 1999 windows available in a project (including base window, common window and fast selection window), but only one fast selection window. Switch the assigned window as a fast selection window by the [Jog FS-window] function key. Please refer to Function Key chapter for further detail.

Each new project has a initial screen, the window number is 10. (or 1, please refer to [System parameters]/ [Editor]). A project often uses several windows. Valid range of the window numbers is from 10 to 1999 (number under 10 is reserved for system, such as window 4 for fast selection window, window 6 for common window).

#### NOTE

1. Maximum 6 popup windows can be open at the same time. (If using print function, maximum 5 windows are displayed. If selecting compressed object function, maximum 4 windows are displayed. The reason is because each of the function occupies one window.
2. One window can be opened once at the same time. Therefore, a window can't be opened by two direct (indirect) windows on a base window.
3. Using Function Key parts with Close Window doesn't close direct or indirect window. This is because the on or off of the direct window is only related to the On or Off status of the bit lamp address, while the on or off of the indirect window is only related to the data of the word lamp. The function key couldn't change the data of the addresses to close these windows.
4. Window 0 to window 9 are used for system, such as window 4 for fast selection window, window 6 for common window. Only from window 10 to window 1999 are available for any usage.
5. Popup window is attached on the active base window. When base window is closed, the popup window is automatically closed. If you switch to the base window you will find the popup window still attaching to the base window is active. (Unless there is order closed)
6. Base screen must be full size.
7. Fast selection window doesn't support popup window, which means that it's not allowed to use popup window on the fast selection window.

8. Being touched off and opened popup windows will be added on base window , all windows of the same situation can be regarded as the set membership. So, object in common window touches off the open window and exists all the time,unless object in common window touches off and closes it.

## 4.2 Window properties

**Window Setting**

Name : Initial Screen

Window no. : 10      Start Pos. : X : 0      Y : 0

**Size**

Width : 640      Height : 480

**Style**

Tracking     Monopoly     Clipping     Coherence

**Security level**

Lowest

**Underlay window**

1: None    2: None    3: None

**Frame**

Width : 4      Color : [White]

**Background**

Color : [Black]    Pattern : [Black]

Filled      Pattern color : [White]

OK      Cancel

[Window Setting ] prove as follows:

Name:Give each window a unique name in order to identify it.

Window No:Valid window numbers are from 10 to 1999 (1 to 10 is reserved for system).

Start Pos:If this window is popup window,the angular vertex will be located in this seat on its left.

Size:Set appropriate width and height .Set window width as full screen size (320x240 or 640x480 depending on model) if it's a base window.

Style:Tracking, Monopoly, Clipping and Coherence.

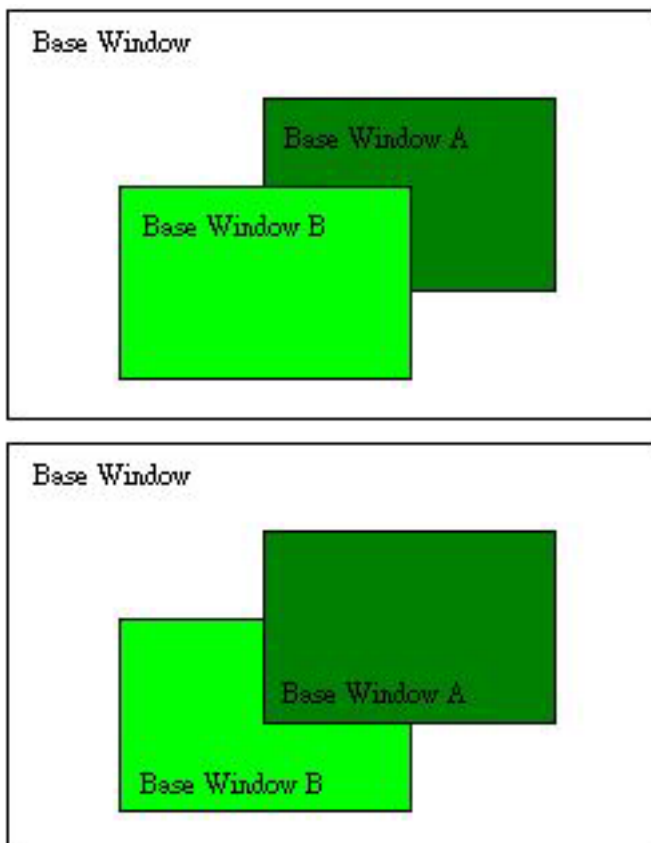
These parameter types have pointed out that one spring relation between the window and window in close proximity to it. If there is " monopolizing " nature at a window, then when it springs, its father will be frozen at window, and spring this at the window and show it all the time in its upper strata the most. If there is " cutting and getting rid of " nature at a window, be then limited by its father's window at the border of this window, that is to say that shows that the part outside its father's border of the window will be cut down! If there is " following " nature at a window, then when the father in this window moves at window, it will follow to move too.

Both the Clipping and Tracking must be selected on the same time.

Coherence means that there can be no windows between the calling window and this window. One cannot be brought to the foreground without the other; both will be brought forward.

For example, window A and window B are on the base window. Generally speaking, window A is displayed on the upper level if it is touched. However, if you mark it as Coherence, even if you touch it, it doesn't go to the up level and cohere to it's calling window.

Note: The window number is not changeable after a window is created, but name, frame and background are changeable.



Security: Set the corresponding security level to prevent unauthorized access to windows. See Ch10. Security for further details.

Underlay window: Set up corresponding Underlay Window in the present window. Underlay window is set only for created window. Generally speaking, shared objects can be put on it.



Frame: It's suggested to choose frame for popup window. The width of the frame can be selected:0, 4, 6, 8, 10, 12, 14, 16. If the Frame is not set to 0, 8 colors are available for the frame.

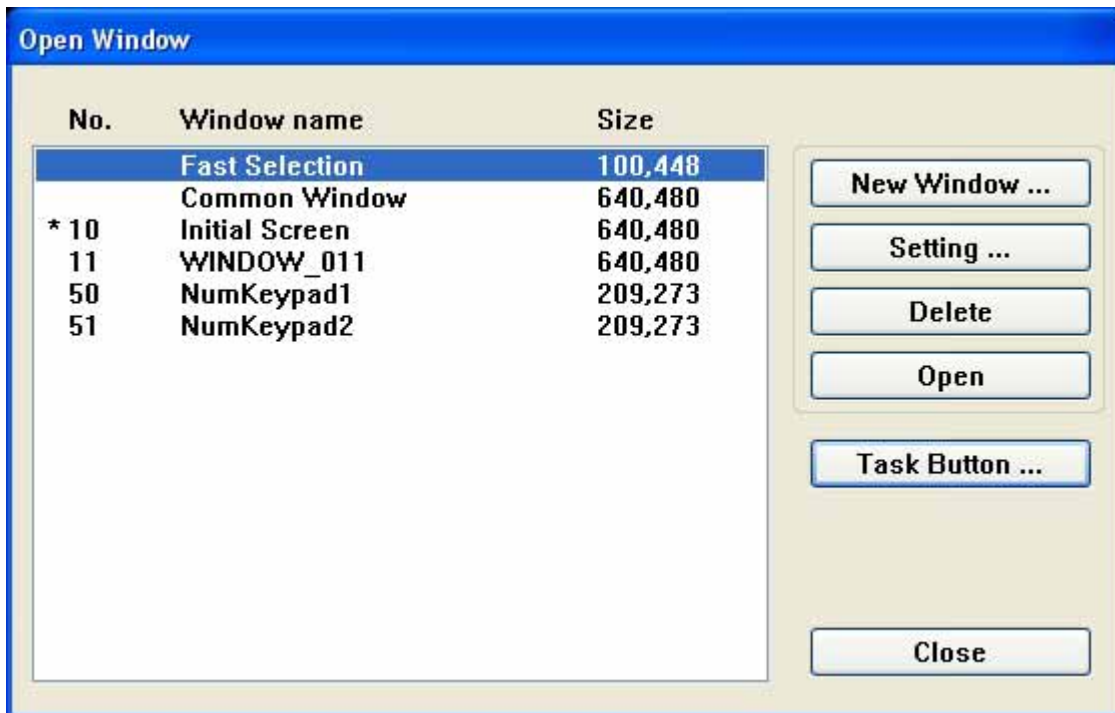
Background: Select the background color and pattern for the window.

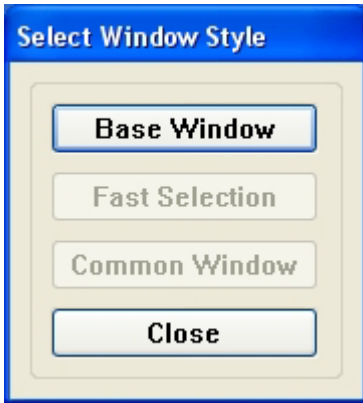
Modifying window properties

- 1). Select [Open Window] from the window menu. Highlight the target window, click on Setting to change the properties.
- 2). Right click the selected window from the window Treebar and choose setting.

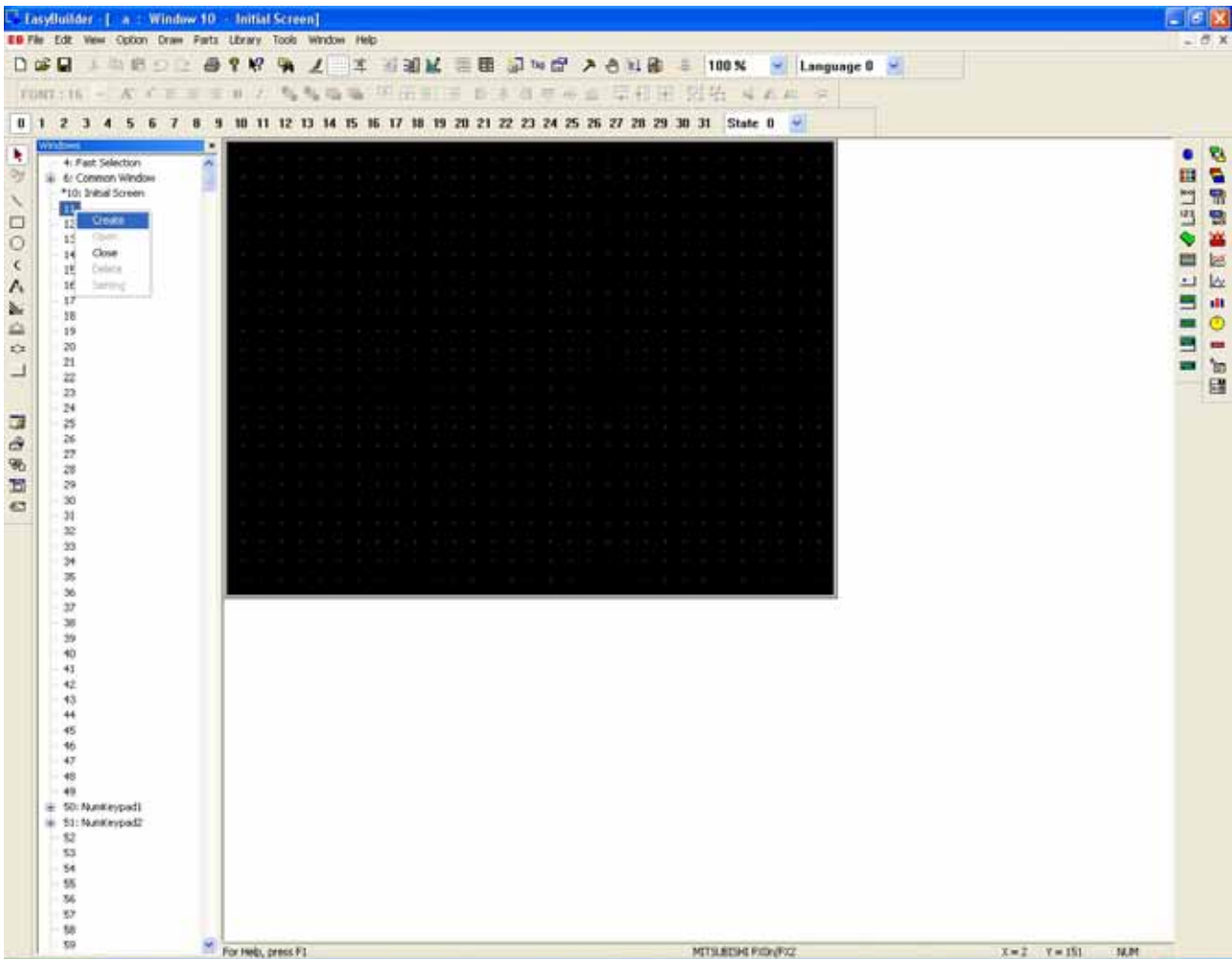
### 4.3 Creating New Windows

In the window menu,select the Open Window command/ New window. Select the window type:Base Window or Fast selection or Common Window to create a new window or right click on an undefined window number in the Treebar and then select Create.





Selecting appropriate types of window pops up window properties dialog. Base window is selected here. Or right click directly on an undefined window number in the Treebar to create a new window as the graph below.



The popup dialog of window properties shows as follows:

**Window Setting**

Name :

Window no. :  Start Pos. : X:  Y:

**Size**

Width :  Height :

**Style**

Tracking  Monopoly  Clipping  Coherence

**Security level**

**Underlay window**

1:  2:  3:

**Frame**

Width :  Color :

**Background**

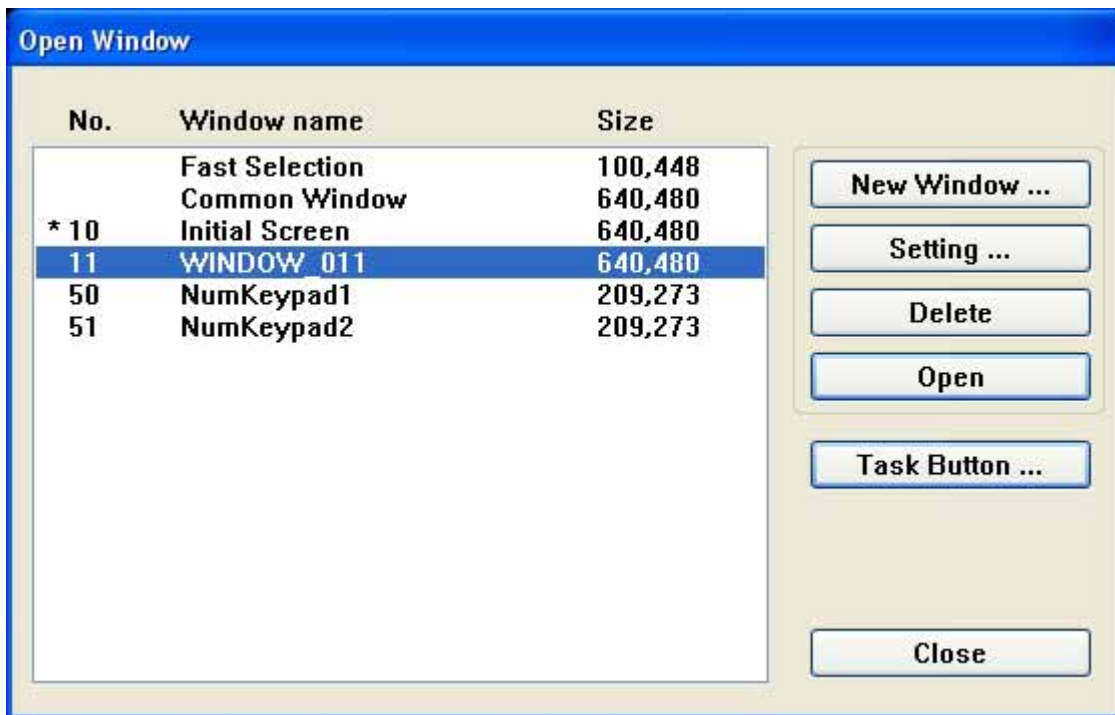
Color :  Pattern :

Filled Pattern color :

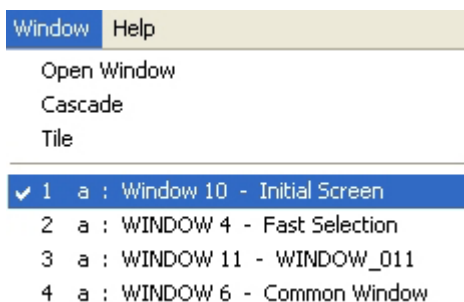
Fill in all properties and then click OK.

## 4.4 Opening Windows

After a window is created, with the open window dialog box select it from the list and double click it or press [Open] to open the window. The windows with an \* mark are already opened.



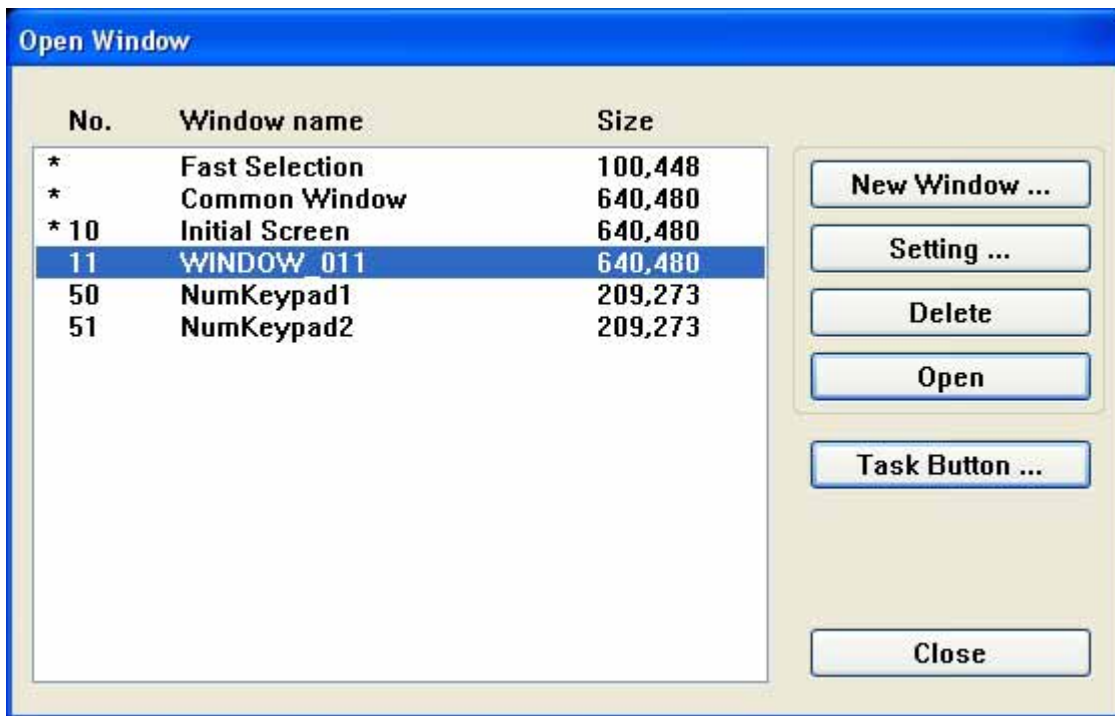
Select the window name from the window menu to enter an opened window.



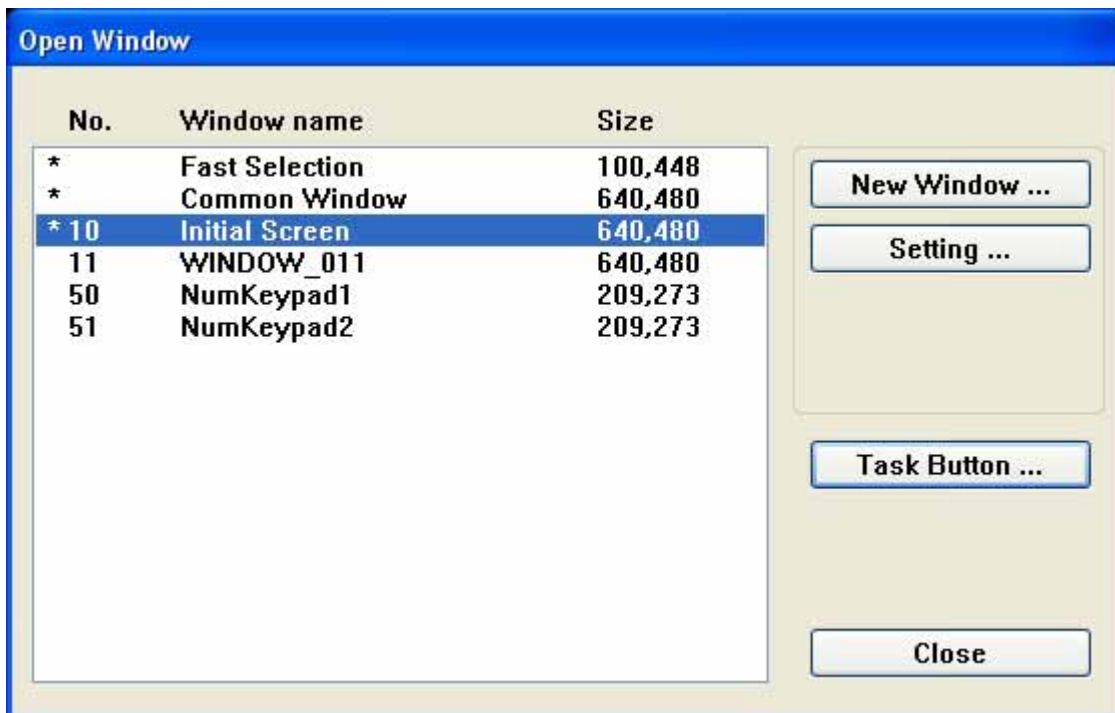
### Deleting windows

1. On the [Window] menu, select [Open Window], then choose the window from the list and press [Delete].
2. Right click the window number in the window Treebar. Selecting Delete to delete the window.

Note: A window Delete can't be executed when it's active. For example, window 11 is closed so that you can find the Delete selection:



Window 10 is active so that you can't find the Delete selection.

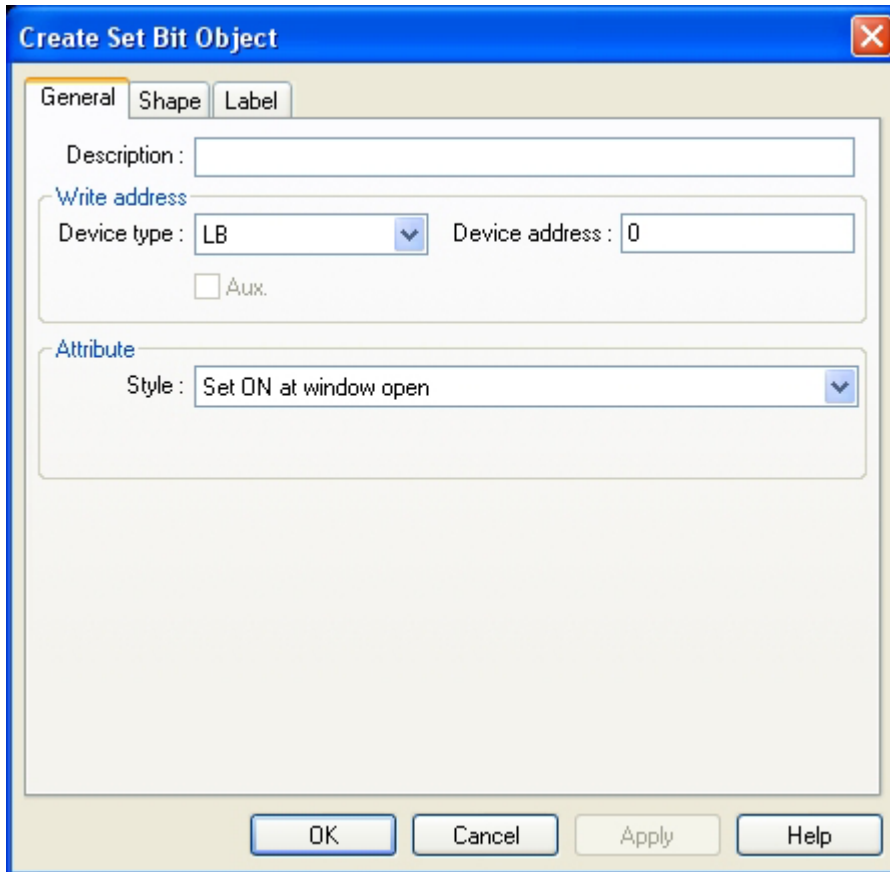


## 4.5 Examples

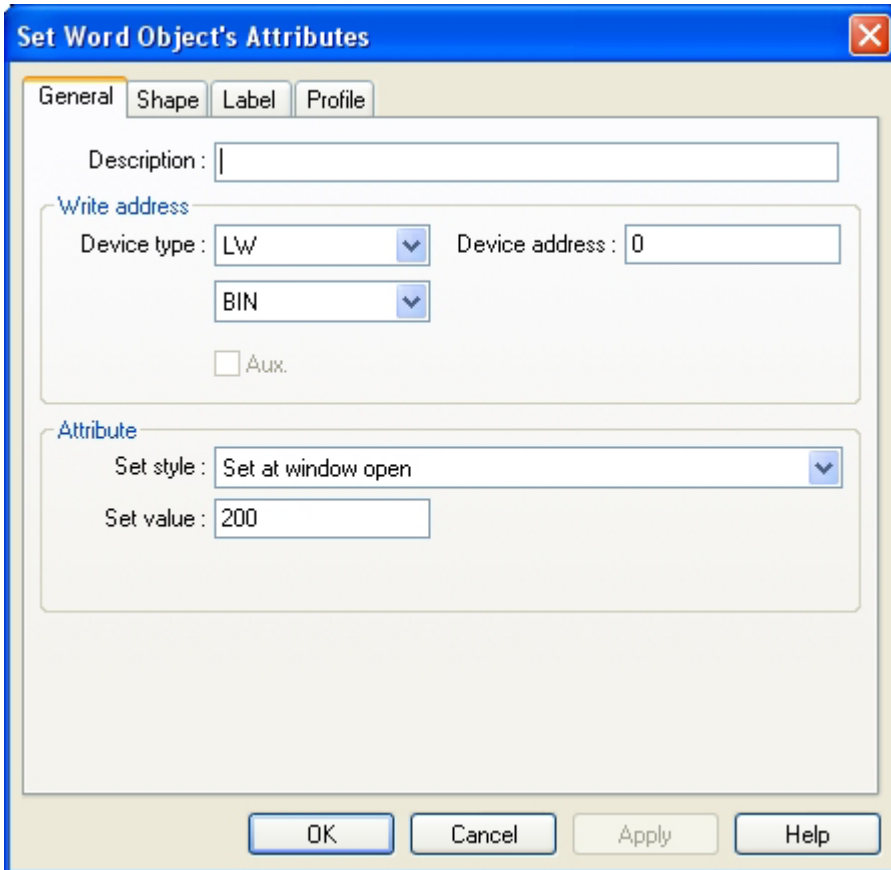
The followings are some simple examples of operating the windows.

Example 1: We need to set up certain data to the register when initializing a project or an application. We set LB0 to On and LW1 to 200 when start up window 10.

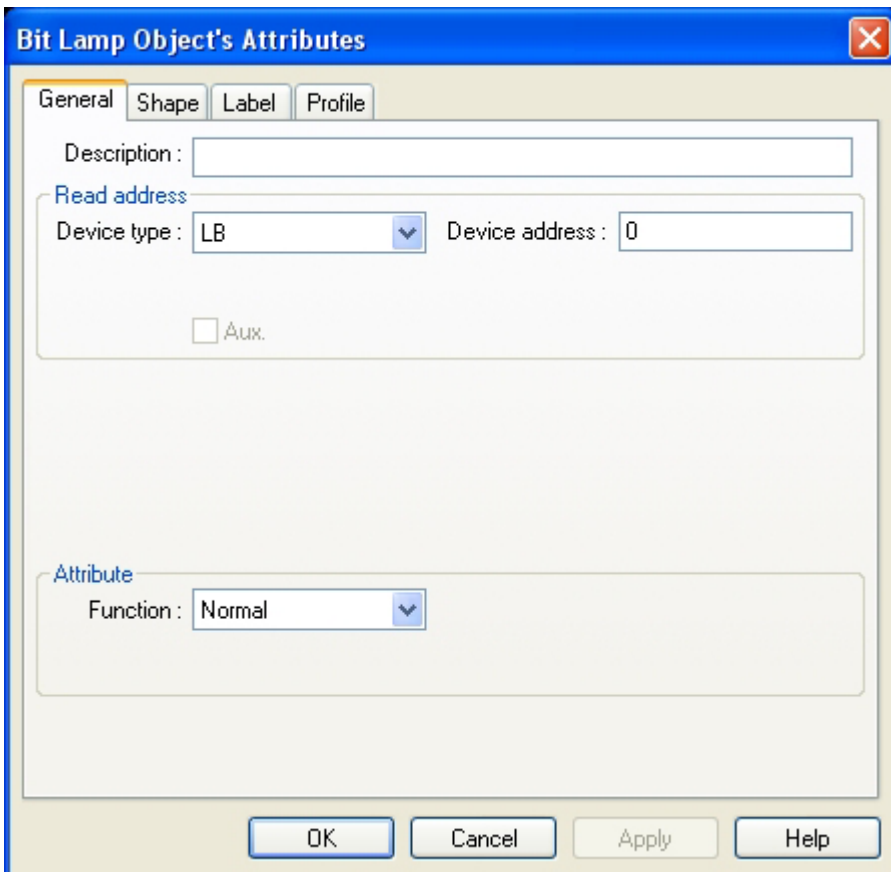
1. Create a (Set Bit Object), Device type for LB, Device address for 0. Select style [Set On at window open]



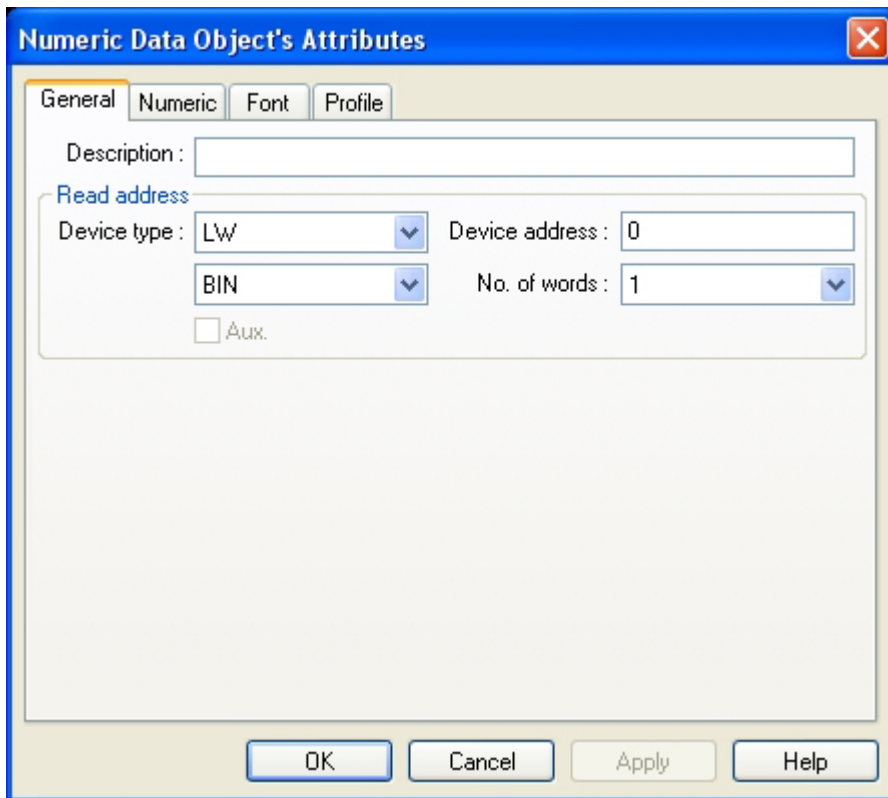
2. Create a Set Word object , Device type for LW, Device address for 1. Select style [Set On at window open]



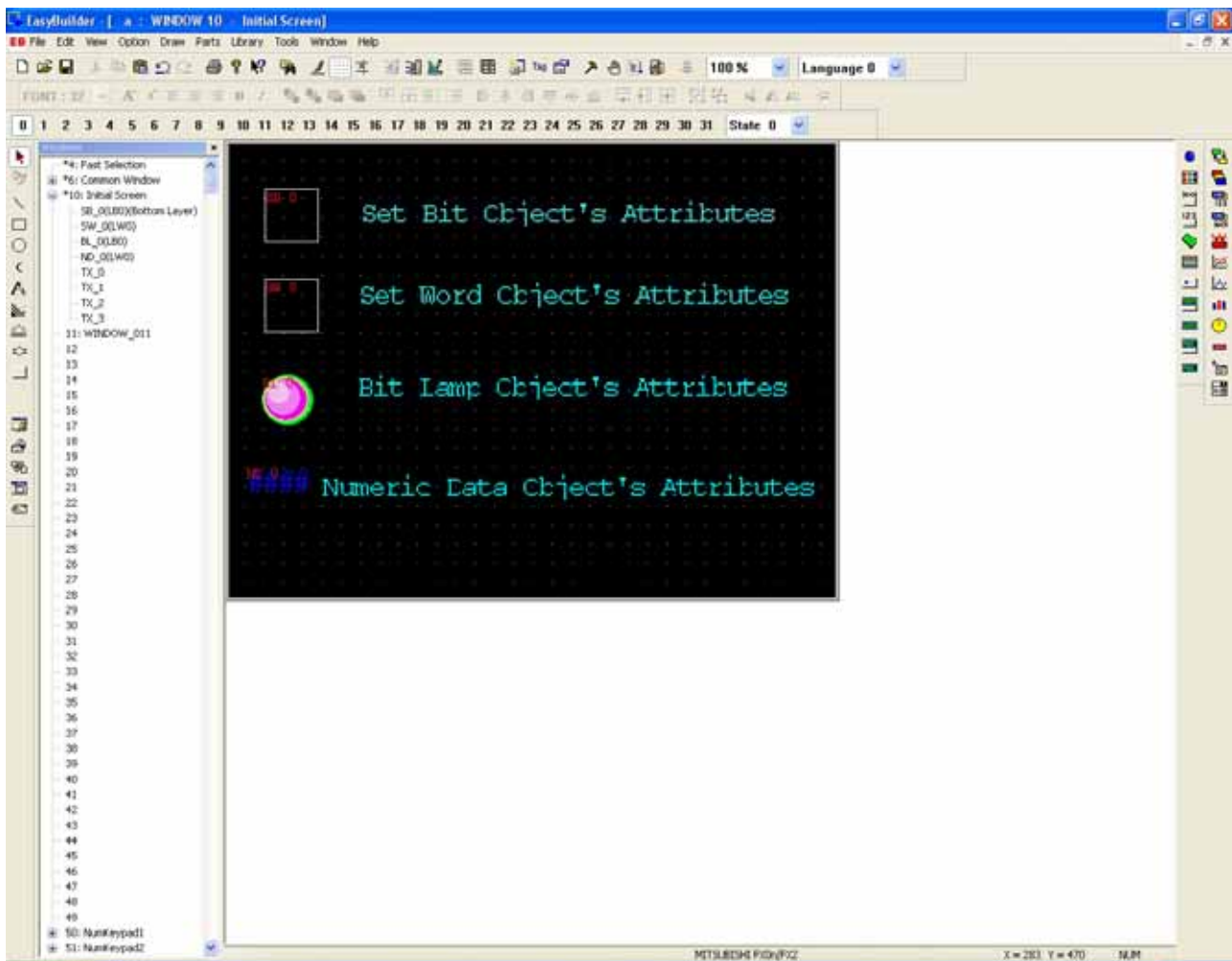
3. Create a Bit Lamp object, Device type for LB, Device address for 0. Select Function as Normal to display at State 0.



4. Create a numeric data object, Device type for LW, Device address for 1.

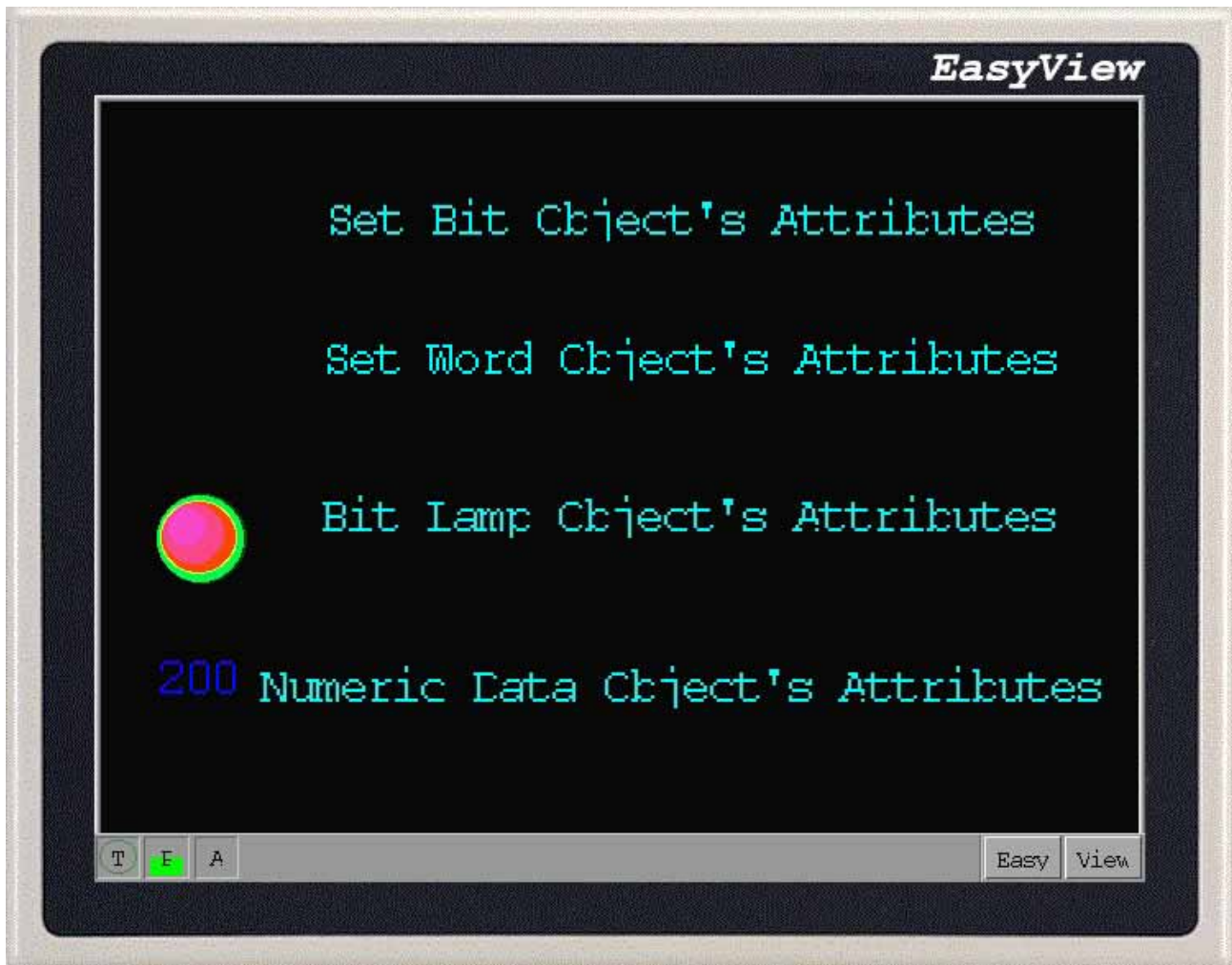


5.The complete project shows as below:



6.Save , Compile , Off-line simulation to run the project .LB0 is set for ON, LW1 is set for200.





Reference : Ch6.1:Bit Lamp Object,6.4:Set Bit Object,6.5:Set Word object,6.11:Numeric Data Object.

Example 2:A popup window with a keypad.

In many situations, we need to input the data to the register. If we place the keyboard on the active window, not only is it too dull to be artistic, but it also eliminates the space for designing the active display. We use a popup direct window for solving this problem well. If we would like to input data to the LW0.

1.Create a numeric input extend object, set Device type for LW, Device address for 0,Trigger device for LB and address for 9000.

**Create Numeric Input Extend Object**

General Numeric Shape Font

Description :

Read address

Device type : LW  Device address :

BIN No. of words :  1

Aux.

Trigger address :

Device type : LB  Device address :

Aux.

OK Cancel Apply Help

2. Create window 11, width is 180, height is 180.

**Window Setting**

Name :

Window no. :  Start Pos. : X:  Y:

**Size**

Width :  Height :

**Style**

Tracking  Monopoly  Clipping  Coherence

**Security level**

**Underlay window**

1:  2:  3:

**Frame**

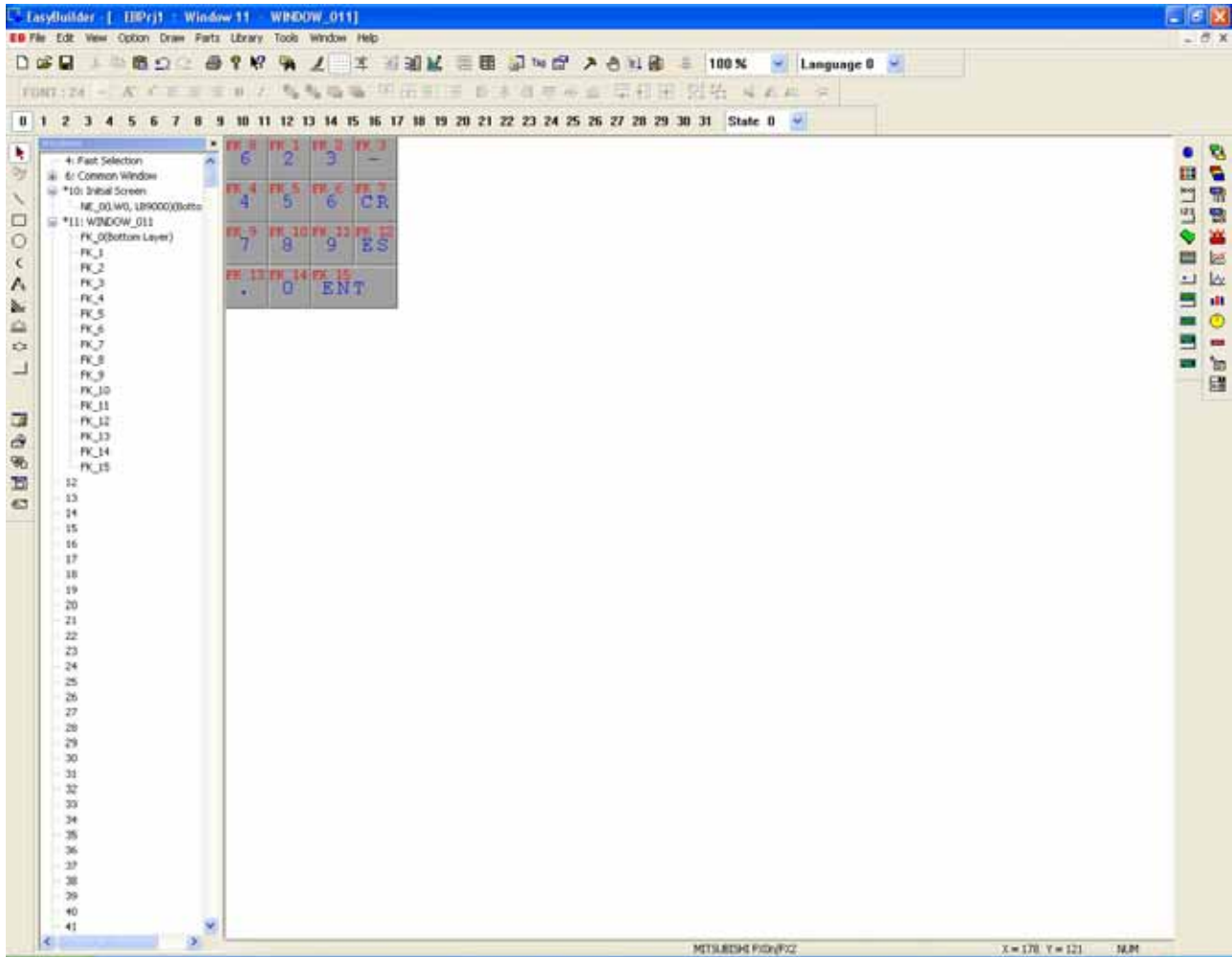
Width :  Color :

**Background**

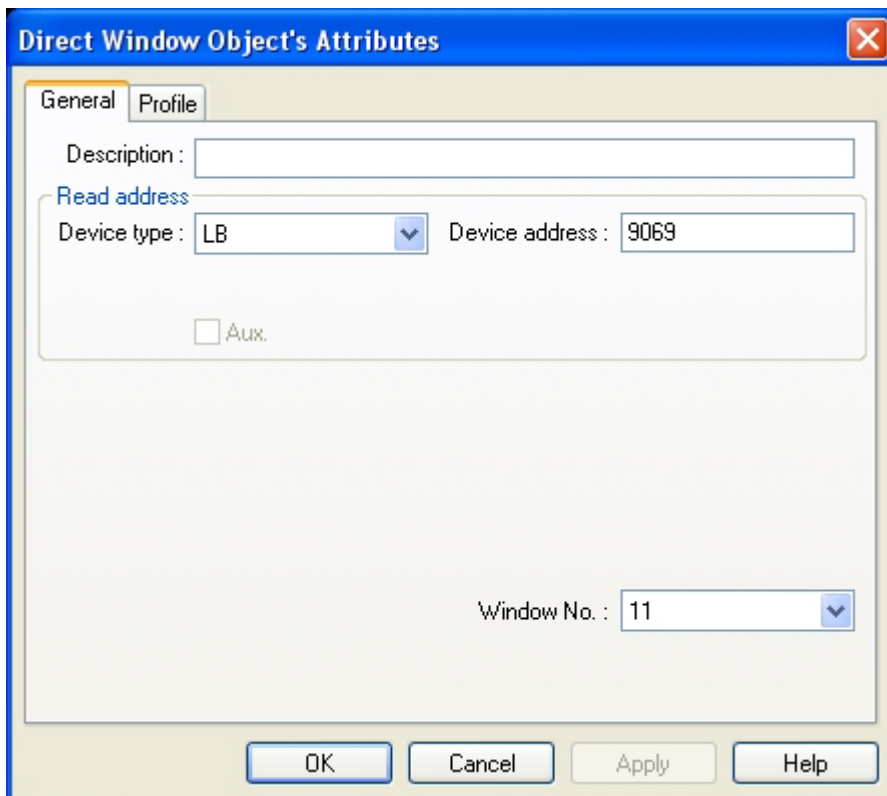
Color :  Pattern :

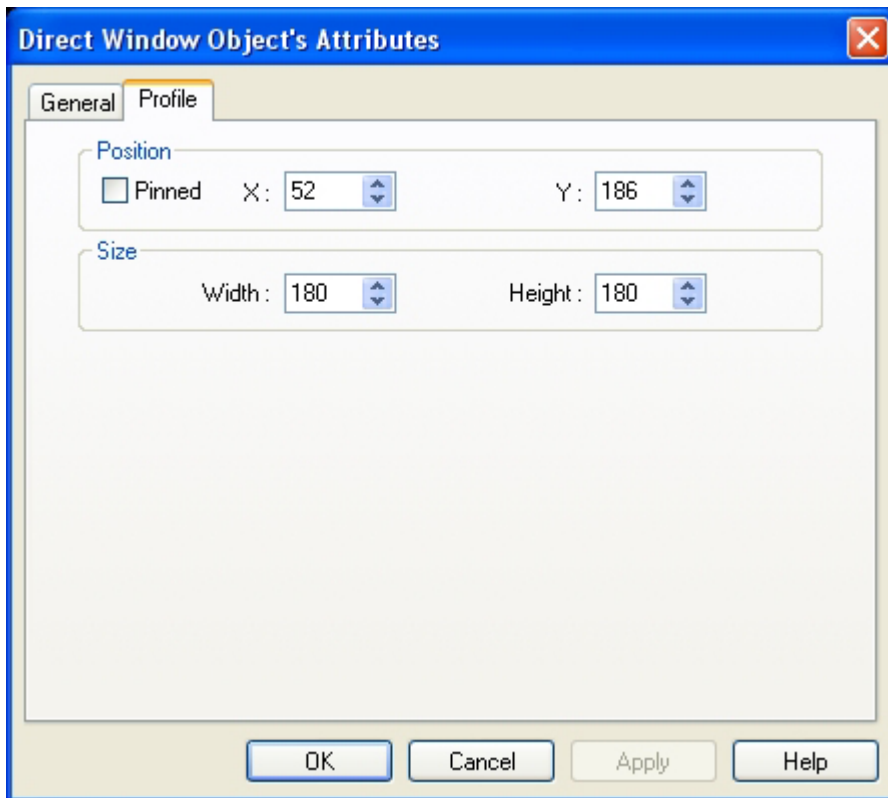
Filled Pattern color :

3. Edit window 11, place a keypad in the middle of the window.

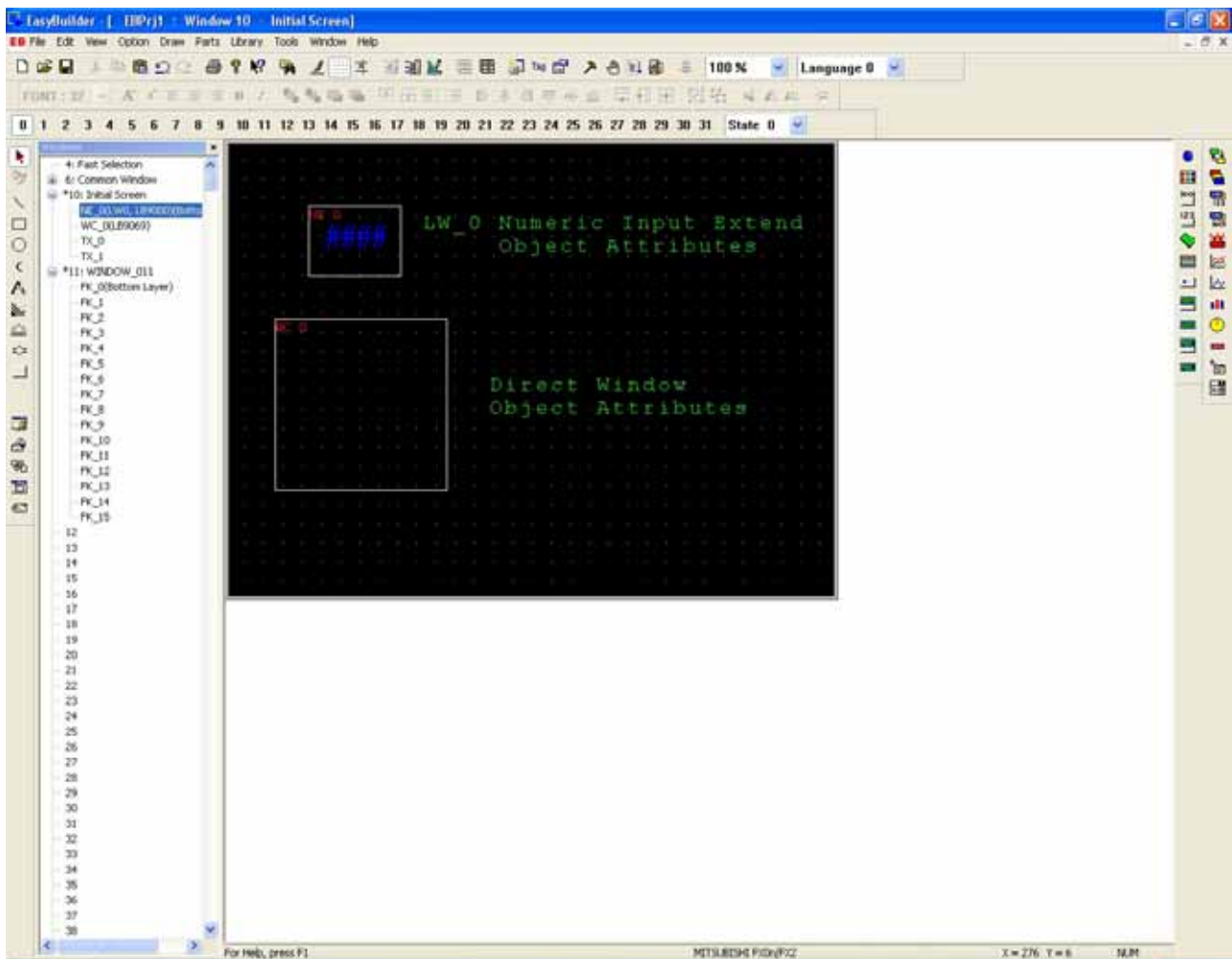


4. Create a direct window object, Device type for LB, Device address for 9069. In size, width for 180 and height for 180.

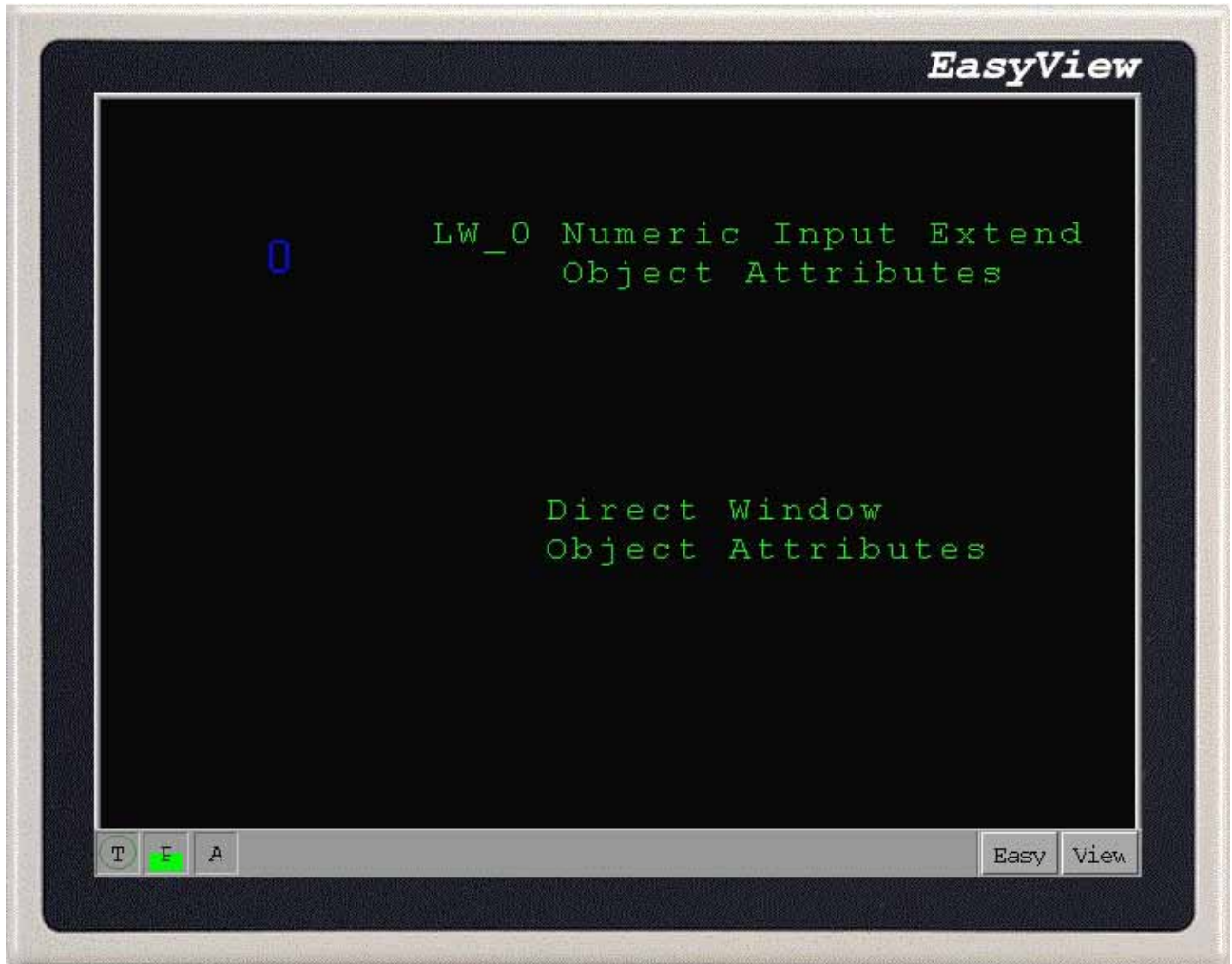




The complete project shows as below:



5. Save, compile and Off-line simulation to run the project:



6. When clicking on the numeric input extend object to input LW0, a popup keypad appears under the object. After inputting, the keypad disappears.



Reference:Ch.13,system reservation for register address B9000,LB9069.

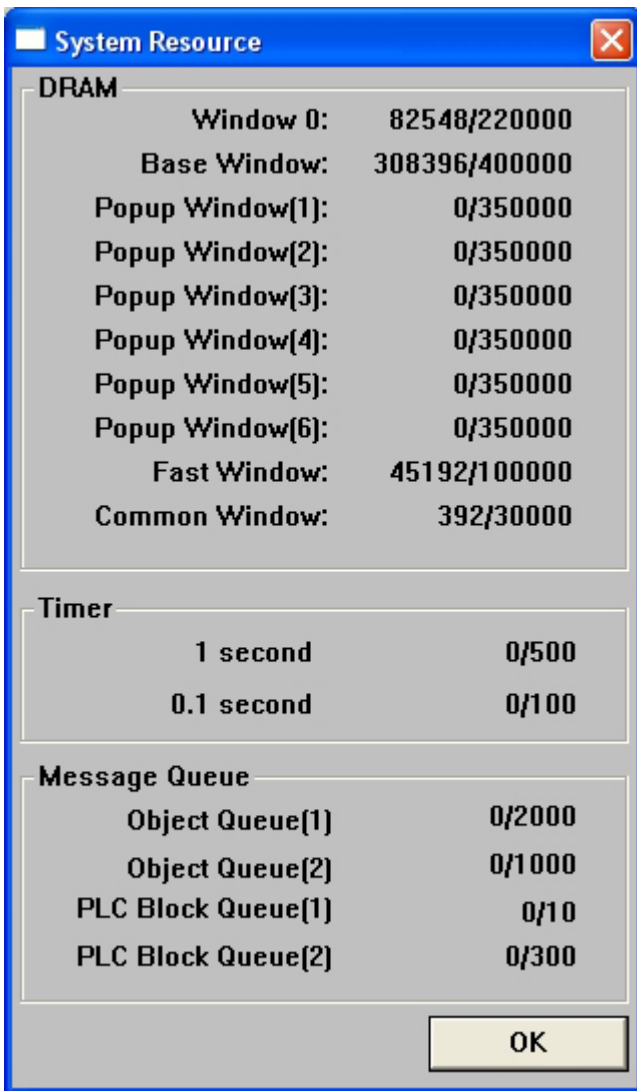
Note : In EB500, there is a default 2-digit keypad.

## 4.6 Windows related object

Directly-related function keys of windows: Change Window, Return to Previous, Change Common Window, Popup Window, Jog FS-Window. Related function keys of direct window: Direct and Indirect window. Please refer to corresponding content of [Parts].

## 4.7 Resource allocation of windows

MT500 uses static distribution memory to allocate the resource. Each window occupies certain fix resource. Right click to select [System Resource] under the off-line simulation to look up the distribution of the resource.



The screenshot shows a dialog box titled "System Resource" with a close button (X) in the top right corner. The dialog is divided into three sections: DRAM, Timer, and Message Queue. Each section contains a list of resource types and their current usage relative to their total capacity.

Resource Type	Current Usage / Total Capacity
<b>DRAM</b>	
Window 0:	82548/220000
Base Window:	308396/400000
Popup Window(1):	0/350000
Popup Window(2):	0/350000
Popup Window(3):	0/350000
Popup Window(4):	0/350000
Popup Window(5):	0/350000
Popup Window(6):	0/350000
Fast Window:	45192/100000
Common Window:	392/30000
<b>Timer</b>	
1 second	0/500
0.1 second	0/100
<b>Message Queue</b>	
Object Queue(1)	0/2000
Object Queue(2)	0/1000
PLC Block Queue(1)	0/10
PLC Block Queue(2)	0/300

An "OK" button is located at the bottom right of the dialog box.

A base window occupies at most 400kb, which means if you place too many objects on the base window to exceed 400kb, error occurs. A popup window occupies at most 350kb, fast selection window occupies 100kb and common window occupies 30kb. Please refer to [Help] in Ch.8.3.

The basic principle of window design is to select an object and set up the attributes. Place on the display and then save the project. Please refer to Ch. 5 for basic design of windows.