

## Form Edit

A form definition is the set of wd commands that define a parent window and its controls. The Form Editor takes a form definition from a script file, displays the form, supports editing, and saves the new definition back in the script. The script also contains the verbs that handle events when the form is run. The Form Editor facilitates editing these verbs.

The best way to learn about this is to try it. Create a new script and run Edit/Form Edit. Click the Design New button to add controls. Drag controls with the mouse.

Controls can be marked with red or blue borders. The red control is the selection and all marked controls are the collection. The mouse marks and unmarks controls.

|             |           |
|-------------|-----------|
| Hints       | Mouse     |
| Keyboard    | Design    |
| New Control | Control   |
| Parent      | Menu      |
| Toolbar     | Statusbar |
| Code        | Tab Order |
| Run         | Defaults  |
| Tech Notes  |           |

## Form Edit Hints

Pick a good form id at the outset, to avoid having to change it later.

Create a new script and save it as a permanent file. The form id might be a good name for the script.

Menu command Edit/Form Edit activates the form editor. It asks for a form id and a template.

Use the Design Dialog to create, position, and size controls. The New button activates the New Control Dialog used to add new controls to the form.

In dialogs such as New Control, OK performs the action and returns to the Design Dialog, and Apply performs the action and stays in the dialog.

Ctrl+C copies the collection to an internal clipboard and Ctrl+V, Ctrl+B, or Ctrl+N paste. For example, create a radiobutton, move it, Ctrl+C, and do Ctrl+N several times.

The Tab key can be used to find a control covered by another control.

You can open more than one form and copy and paste between forms.

Create the controls you want and position and size them roughly. Fine tune position, size, and alignment, after everything is on the form.

Customize control styles, captions, and ids with the Control Dialog. Give descriptive ids to controls that cause events or will be used in wd commands.

Create a menu with the Menu Dialog.

Set the tab order after the form is pretty much complete. Moving, adding, or deleting controls require redoing the tab order. Radiobuttons are automatically grouped if they are in the right tab order.

Run the form with the Design Dialog Run button. This shows the form as it will look to the user. Check the tab order and test your event handlers.

The Code Dialog shows which verbs are required to handle events and allows you to activate the form script and position the caret at the definition. Do not add handlers until the layout is settled.

Ctrl+click a control, a menu item, or the script to switch between the script and the form editor.

## Form Edit Mouse Actions

click -- select control (collection is maintained)

shift+click -- toggle mark on control

ctrl+click -- activate script (ctrl+click script to return)

double-click -- show Control Dialog

drag move -- drag collection. Ctrl+shift to move just the selection.

drag resize -- just outside the red border shows a resizing cursor. Ctrl+shift to resize just the selection.

draw select -- press the mouse outside any control and draw a box to select controls

A mouse action in a marked control affects that control. To select a control covered by a marked control, first unmark the control. The mouse selects the control with the nearest upper left corner.

The mouse displays the form menu. Select a menu item with Ctrl to activate the script.

If tab order is displayed mouse actions are different.

## **Form Edit Keyboard Actions**

Ctrl+X -- cut collection (internal clipboard)

Ctrl+C -- copy collection

Ctrl+V -- paste

Ctrl+B -- paste Beside collection

Ctrl+N -- paste beNeath collection

Ctrl+T -- toggle tab order display

Del -- delete collection

Tab -- select next in tab order

Shift+Tab -- select previous

## Form Edit Design Dialog

The New button shows the New Control Dialog for adding controls to the form.

Move collection with the cross direction buttons. The central button switches the move-grid between 1 and 4. The move is to the move-grid if not already on the grid.

Align collection with shift + a direction button.

Size collection with ctrl + a direction button.

The shift key causes the following buttons to act down rather than across.

Center -- center collection

Size -- size collection to the selection width

Space -- space 3 or more controls across the form

Touch -- space collection so they touch

Minus -- remove space

Plus -- add space

## **Form Edit New Control Dialog**

Design Dialog New button shows this dialog that is used to add controls to the form.

Select the type of control you want to create and give the control an id.

Some controls, such as edits, are often labeled by a static. If you want to label the control, fill in the label field. Check down if you want the control below the label.

To create several controls of the same type, for example several radiobuttons, fill in the copies field. Check down if you want them vertical.

Click OK or Apply to recreate the form with the new control(s).

## **Form Edit Control Dialog**

Double-click a control to show this dialog. Make changes and press OK or Apply to recreate form.

The listboxes can have multiple items selected.

Ownerdraw buttons are not drawn when designing the form. The file name is in the caption. Edit the caption to be the name of the file with the graphic for the button.

## Form Edit Parent Dialog

Design Window/Parent shows this dialog. Make changes and press OK or Apply to recreate form.

### Styles:

- closeok -- parent closes without causing an event
- dialog -- heavy frame
- nomax -- no maximize button
- nomenu -- no system menu
- nomin -- no minimize button
- nosize -- no sizing frame
- owner -- owned by currently selected form (owner disabled until close)

Options do not affect the form during design.

The pmove button toggles use of a pmove command. When set, the current design location is shown to the right of the button. The form will be moved to that location when it is created.

The right and bottom margins set the margins used in creating the form.

## Form Edit Menu Dialog

Design Window/Menu shows this dialog. Edit the definition and press OK or Apply to recreate the form.

A popup menu item definition has just a caption. For example:

```
File
```

A command menu item definition has an id, caption, and optional shortcut, tooltip, and statushelp. For example:

```
idopen caption _ shortcut _ tooltip _ statushelp
```

The \_ character separates fields that may contain blanks. The shortcut appears at the right side of the menu list. The tooltip does not affect the menu, but a toolbar button with the same id will show the tooltip when the mouse rests on the button. The statushelp shows in the statusbar when the mouse is on the menu item.

+ ends a popup menu, - is a separator. and & underlines the next character.

For example:

```
&File
```

```
new &New
```

```
open &Open _ Ctrl+O _ tool tip _ status help text
```

```
-
```

```
idquit Quit
```

```
+
```

```
&Help
```

```
idhelp &Contents
```

Use the mouse to show the menu on the form. Press Ctrl and select a menu item to go to the code in the script for that event.

Use the Code Dialog to define verbs for menu ids and shortcuts. A menu shortcut usually calls the verb defined for the menu id. For example, shortcut Ctrl+O could define a handler for fkey 'o ctrl' that called `formid_idopen_button`.

## Form Edit Toolbar Dialog

Design Window/Toolbar shows the Toolbar Dialog. This dialog manages the form toolbar.

A toolbar is a set of buttons that show across the top of form, below the menu.

The buttons are drawn from a bitmap. The File button allows you to select the bitmap for your toolbar. The selected bitmap shows across the top of the dialog with the buttons numbered.

A line in the edit box gives the id for the toolbar button, the index of the bitmap picture to use, a tooltip to display when the mouse rests on the button, and statushelp to display in the status bar.

A line with just an index is a separator where the index is the width.

For example:

```
idnew 0 new file _ Create a new file.
```

```
idsave 2 save file _ Save file.
```

```
25
```

```
idprint 6 print file _ Print file
```

A toolbar button is usually a shortcut for a menu item and in this case they should have the same id.

## Form Edit Statusbar Dialog

Design Window/Statusbar shows the Statusbar Dialog. This dialog manages the form statusbar.

A statusbar is a set of panes across the bottom of a form.

The first pane is the help pane and its width varies from a minimum up to the space remaining. It has default text set by the sbarset command. It shows menu and toolbar statushelp when the mouse is on a menu item or toolbar. When this temporary help is not shown the pane reverts to its default contents.

A line in the edit box give the id, width, and initial text for each pane in order.

For example:

```
idstatushelp 100 Press F1 for help.  
idstatusrdy 50 Ready  
idstatusoption 75 option
```

## Form Edit Code Dialog

Design Window/Code shows this dialog.

This dialog helps you define and edit verbs in the script that handle events.

The button event type lists the ids of controls that cause button events.

Items are indented if a verb is not defined for that event.

Double-click an item in the listbox to activate the script with the caret located at the verb. If there is no definition, a default definition is added. Ctrl+click the script to return to the form editor.

The fkey event type lists keys that cause events. Alphabetic and numeric keys with Ctrl or Ctrl+Shift as well as function keys cause events. Fkey events Ctrl+Z,X,C, or V are not supported. Menu shortcuts are handled as fkey events.

The other event type lists general form events.

The orphan event type lists verbs defined in the script for which there is no control that would cause the event. For example, create a button and create a handler verb in the script, then delete the button.

Ctrl+click a control, a menu item, or the script to switch between the script and the form editor.

## **Form Edit Tab Order**

Ctrl+T in the form toggles display of the tab order.

The tab order determines how the user moves around on the form with the tab key.

Mouse actions with the tab order displayed are as follows:

click -- select control (red border)

shift+click -- put new selection after previous selection in the tab order

ctrl+click -- put new selection as first in tab order

## **Form Edit Run**

Design Dialog Run button saves the form in the script, saves the script, runs the script, and runs the verb `formid_run` to create and show the form.

Code Dialog other event type entry run takes you to the form run verb definition.

## **Form Edit Defaults**

Design Dialog File/Defaults displays the sizes used for controls created with New Control. The defaults can be changed and can be recorded in the ini file.

The default form font can be set by the main J menu with View/Set Form Font. Use File/Save jsm.ini to save a new form font as the default in the ini file.

## **Form Edit Tech Notes**

A form definition in a script begins with a line that starts with a pc command and ends with the line <rem form end>.

Scripts in system\packages\forms are used as templates for new forms.