

# Application Framework Contents

## Overview

[Welcome to the Application Framework](#)

[Application Framework components](#)

## How To

[Creating and running an application](#)

[Creating forms, reports, and scripts](#)

[Customizing your application Menu Bar](#)

[Changing Application Framework properties](#)

## Tools

[Project Viewer](#)

[Application Menu](#)

[Application Event Expert](#)

## Reference

[Application Framework aliases](#)

[Application Framework files](#)

[Application Framework properties](#)

## **Welcome to the Application Framework**

Welcome to the Corel Paradox Application Framework. The Application Framework allows you to quickly create an executable application without writing a single line of code.

Integrated seamlessly with the Project Viewer, the Application Framework enables you to create the shell, or framework, of your application with a few clicks of the mouse. Even before you create a single form or report, you can run your application and view its framework.

The framework of your application includes the main window with the name of your application in the title bar, a customizable Menu Bar, and an About box (that users can click to view the name and copyright information of your application). The default framework also contains built-in properties that you can change with a click of the mouse. For example, by default, whenever you click File, Exit, your application displays a modal dialog box that asks users to confirm exiting the application. From within the Project Viewer, you can change this property (called App On Close) with a click of the mouse.

Once you have created the framework of your application, you create the tables, forms, reports, and other database objects using the Project Viewer. The Application Framework integrates each database object (form, report, etc) with your application as you create it. For example, whenever you create a form, report, or script, the Application Framework allows you to add its name to the Menu Bar of your application. You can specify the menu item under which it appears and its position in the list.

The Application Framework is so easy to use that even the most novice database user can create a professional, integrated application with no knowledge of ObjectPAL.

## **Application Framework components**

The Application Framework is a small set of powerful design tools that create and manage your application. The following is a brief description of its main components.

### **Project Viewer**

The Project Viewer enables you to create and open applications, as well as forms, reports, and so on. You use the Project Viewer to manage all the objects in your application. The Application Framework also expands the capabilities of the Project Viewer and enables you to set application properties that control the look and behavior of your application.

### **Application Event Expert**

The Application Event Expert helps you add specially-named methods (such as `prefilterMouseClickedBefore` which allows you to specify the action that occurs when a user clicks the mouse) to your application. You can use the Application Event Expert to add events and apply them to all objects on a form, all forms in an application, or all applications you create within the Application Framework. You can also specify when the event takes place. The Application Event Expert is seamlessly integrated into the Application Framework and allows you to customize your application without writing a single line of code.

### **Application Menu**

The Application Menu, accessible through the Tools menu, allows you to design the Menu Bar that ships with your application. Use it to lay out the menu and connect menu items to the forms, reports and other objects in your application.

### **Framework Launcher**

The Application Framework Launcher opens Corel Paradox and runs your application when you double-click the application .EXE file.

## Creating and running an application

Creating an application with the Application Framework is easy. You can run your application and view its framework before you create a single database object (table, form, report and so on).

Once you create the framework of your application, you can create all of the forms, reports, scripts and other objects you want to include. You must create these objects within the Application Framework. You can, however, use existing tables.

As you create each form, report, or script, the Application Framework allows you to include the name of the object in the Menu Bar and specify its position. You can further customize your application's menu bar using the Application Menu dialog box (Tools menu).

Using the Project Viewer, you can change the properties for each type of object. If you want to add form-level, application-level, or Application Framework-level events to your application, use the Application Event Expert (Tools menu).

### To create an application

1. In the Project Viewer, right-click the Applications icon and click New.
2. In the New Application dialog box, type the name of your application in the Title box.
3. Type the path and folder name for your new application in the Exe Path box. You must specify a folder that does not currently exist.
4. Tab to the Tables Path box. Corel Paradox automatically specifies a Tables folder within the folder you specified in step 3.

The Application Framework creates your application.

You can run your application as soon as you create it.

### To run your application

1. From the Project Viewer, choose your application's working directory from the list box at the top of the window.
2. Click the Run button on the Project Viewer Toolbar.

## Creating forms, reports, and scripts using the Application Framework

Once you create your application using the Application Framework, you need to create all of the forms, reports and scripts that you want to include in your application. As you create each form, report, or script, the Application Framework prompts you to specify such things as which application you want to use the form in, whether you want to include the object in the specified application or all applications, and where the file is listed in your application's Menu Bar. After that, you create forms, reports and scripts the same way as you do when you create these objects outside of the Application Framework.

### To create a form, report, or script using the Application Framework

1. Create your application using the Application Framework and ensure your application's working directory is selected in the Project Viewer.
2. Do one of the following:
  - To create a form, right-click the Form icon in the Project Viewer and click New.
  - To create a report, right-click the Report icon in the Project Viewer and click New.
  - To create a script, right-click the Script icon in the Project Viewer and click New.
3. Type a title in the Title box.
4. Do one of the following:
  - Enable the This Application button to use the form, report, or script with the specified application.
  - Enable the All Applications button to use the form, report, or script with all applications that you create with the Application Framework.
5. Enable the Include In Application Menu check box to include the name of the object in your application's Menu Bar; otherwise, go to step 7.
6. If you chose to include the name of the object in the Menu Bar, choose a menu item from the list box in the center of the dialog box, and then click the arrow buttons to position the item within the list.
7. If you are creating a form or report, choose a style sheet from the Style Sheet dialog box.
8. If you are creating a form, enable the Dialog check box if you want the form to be a dialog box.
9. If you are creating a form and enabled the Dialog check box in step 8, enable the Modal check box to create a modal dialog box for the form.

## Customizing your application Menu Bar

When you create a form, report, or script using the Application Framework, the Application Framework allows you to add the name of the object to the application's Menu Bar under any menu item in the APP MENU (the Menu Bar that appears when your application opens).

If you want to further customize your application's Menu Bar, you can use the Application Menu dialog box (Tools menu).

### To customize a Menu Bar

1. With your application working directory open in the Project Viewer, click the Menus icon.
2. Double-click menu you want to customize. The Application Menu dialog box opens and displays the structure of the specified menu.
3. Change the structure of the menu as required using the procedures outlined below. For more information about the Application Menu dialog box, click the Help button.
4. Click Menu, Save to save your changes. Run your application to view your changes.

### To delete an item from a menu

- Click the item in the left panel of the Application Menu dialog box and click MenuItem, Delete.

### To move an item to a new position

1. Click the item in the left panel of the Application Menu dialog box and do one of the following:
  - Click MenuItem, Move Up to move the item up one item in the list.
  - Click MenuItem, Move Down to move the item down one item in the list.
  - Click MenuItem, Move Left to move a nested menu item up one level.
  - Click MenuItem, Move Right to nest a menu item within the item directly above it.

### To add a divider within a menu

1. Click the item below where you want to add the divider.
2. Click MenuItem, Insert.
3. In the Menu Text box, type a hyphen (-).

### To add an ObjectPAL menu action item

1. Follow steps 1 and 2 of the preceding procedure.
2. In the Menu Text box, type the text that you want to appear in the Menu Bar.
3. In the Menu Type box, enable the Paradox button.
4. In the Menu Id list box, choose the action item you want to add.

### To create a menu item for a form, report, script, executable or dialog

1. Click the left panel of the Application menu dialog box and click MenuItem, Insert.
2. Use the arrow keys to move the item to the required position.
3. Type the text that you want to appear in the menu in the Menu Text box.
4. In the Menu Type box, enable the Application button.
5. If the item you are adding is a main menu item, or is within a menu item that merges with another menu, type a number in the Merge Id box. For information about Merge Id numbers, click the Help button in the Application Menu dialog box.
6. From the Object Path list box, choose the working directory in which the object you are adding is located.
7. From the Object Name list box, choose the object that you want to open when the user clicks on that item in the Menu Bar.

### To attach a shortcut key to the menu item

1. Click the item in the left panel of the Application Menu dialog box.
2. Click the ellipses (...) button beside the Menu Key box.
3. In the Menu Key dialog box, choose a key from the list box.
4. Enable one of the following check boxes (not required, but highly recommended):
  - Control
  - Alt
  - Shift



## **Changing Application Framework properties**

When you are using the Application Framework to create an application, the Project Viewer allows you to set properties for each object (for example, form or report).

### **To change Application Framework properties**

1. Create your application using the Application Framework.
2. In the left panel of the Project Viewer, click the object whose properties you want to change. Corel Paradox displays the properties you can change for that object in the bottom section of the right panel.
3. Click the property you want to change to select it.
4. Choose the behavior you want from the property list box.



## Application Menu

The Application Menu allows you to fully customize your application's Menu Bar. By default, the Application Framework provides your application with the following menus:

App Menu: The Menu Bar that appears when your application opens.

Form Merge Menu: A Menu Bar that merges with the App Menu. By default, the Application Framework sets the Menu property for each form so that this menu appears whenever you open a form.

Report Merge Menu: A Menu Bar that merges with the App Menu. By default, the Application Framework sets the Menu property for each form so that this menu appears whenever you open a report or script.

### To change the menu that appears when a form, report or script opens

1. In the Project Viewer, open your application working directory.
2. Select a form, report, or script in the right panel of the Project Viewer. The properties at the bottom of the Project Viewer display the properties for the selected object.
3. Choose a menu from the Menu property list box.

Using the Application Menu, you can add and delete menu items. Move menu items to new positions within the menu, Create nested menus, and add ObjectPAL menu actions to your menus. For more information about customizing your application menus, see [Customizing your application Menu Bar](#).



### Note

- Because the Application Framework provides you with three different menus, and prompts you to add the name of each form, report, or script as you create it, you may never need or want to change the structure or layout of the menus in your application.

## **Application Event Expert**

Using the Application Event Expert, you can add ObjectPAL events to any form in your application. After selecting an event, you can specify that the event apply at the following levels:

- For a selected form
- For all forms within your application
- For all forms within all applications

### **To run the Application Event Expert**

1. In the Project Viewer, change your working directory to your application's working directory.
2. Click Tools, Application Event.

## Application Framework aliases

The Application Framework uses a variety of aliases to manage the locations of different files. Each alias supports a particular need. For example, the framework (:fw:/fwTables:) and application (:app:/appTables:) aliases separate the code you use across applications from the code you use for a particular application. The code (:fw:/app:) and table (:fwTables:/appTables:) aliases separate things that are usually local (code) from things that are usually shared (tables) in a multi-developer environment. The tables alias (:tables:) separates application-specific tables (like the one that contains your menu structure) from your data so that multiple applications can share the same data.

### **:FW:**

Points to the Application Framework directory that contains all of the code shared across applications.

### **:FWTABLES:**

Points to the Application Framework tables directory, that contains all of the tables shared across applications.

### **:EXE:**

Points to the executable directory, that contains the .EXE and .INI files, and the startup script for an application.

### **:APP:**

Points to your application directory, that contains all of the code used by your application.

### **:APPTABLES:**

Points to the application tables directory, that contains all of the code-related tables used by an application.

### **:TABLES:**

Points to the tables directory, that contains all of the data tables used by an application.

## Application Framework files

### **.INI files**

[APP.INI](#)

[FW.INI](#)

### **Scripts**

[APP.SSL](#)

### **Forms**

[APPABOUT.FSL](#)

[APPDESK.FSL](#)

[APPMAP.FSL](#)

[APPSPLSH.FSL](#)

[FWERR.FSL](#)

### **Libraries**

[APP.LSL](#)

[APPTOOLS.LSL](#)

[FW.LSL](#)

[FWDM.LDL](#)

[FWEVENT.LDL](#)

[FWON.LDL](#)

[FWTOOLS.LDL](#)

## **APP.INI**

This .INI file contains the list of aliases your application uses. The Application Framework renames this file to match your application. For example, if you name your application MY APP, the Application Framework renames the script file as MYAPP.INI.

## **FW.INI**

This .INI file, which is located in the windows directory, contains the list of Application Framework aliases and the list of Application Framework applications.

## **APP.SSL**

This script file launches your application. The Application Framework renames this file to match your application. For example, if you name your application MY APP, the Application Framework renames the script file as MYAPP.SSL.

## **APPABOUT.FSL**

This form file contains the information displayed in your application's About box. Default About box information is located in the Application Framework directory (:FW:).

To create your own About box form and override the default style, create a form with this name and place it in the application directory (:APP:).



## **APPDESK.FSL**

This form file contains the Desktop form (main window) used by your application and is located in the Application Framework directory (:FW:).

To create your own Desktop window and override the default style, create a form with this name and place it in the application directory (:APP:).

## **APPMAP.FSL**

This form file contains the map form used by the application. A map form contains a series of buttons that open forms and reports in the application and is a shortcut to using menus. When you close the default map form, you close your application.

The default map form is located in the Application Framework directory (:FW:). To create your own map form and override the default style, create a form with this name and place it in the application directory (:APP:).

### **See Also**

[App Show Map](#)

## APPSPLSH.FSL

This form file contains the splash screen used by the application. The default splash screen is located in the Application Framework directory (:FW:). To create your own default form and override the default style, create a form with this name and place it in the application directory (:APP:). You do not have to create your own form to change the splash screen (for example, the image it displays).



### Note

- The splash form is not an Application Framework form. Therefore, it doesn't contain all of the form-level code that integrates a form with the rest of the Application Framework. This is because the splash form is generally displayed while the Application Framework is being initialized.

### See Also

[App Show Splash](#)

## **FWERR.FSL**

This form file contains the error dialog box used by your application. The error dialog is located in the Application Framework directory (:FW:).

## APP.LSL

This library, located in the application directory (:APP:), contains the application-level events that control behavior in a single application. By default, each application contains empty APP.LSL file.

To create application-level events, you create specially named methods within this library. The Application Framework calls those methods at the appropriate time. You can add these events using the Application Event Expert.

For example, to have each form in the application open in edit mode, you'd create this method in APP.LSL:

```
method formOpenAfter(  
    var fwFormInfo FWForm,  
    var eventInfo Event,  
    var fwEventInfo FWEvent  
)  
  
    self.edit()  
  
endmethod
```

This method runs after the form's default open behaviour and is equivalent to including the following code in each form:

```
method open(  
    var eventInfo Event  
)  
  
    if eventInfo.isPrefilter() then  
        ;// This code executes for  
        ;//each object on the form:  
  
    else  
        ;// This code executes only  
        ;//for the form:  
  
        doDefault  
        self.edit()  
  
    endif  
  
endmethod
```

### See Also

[Application Event Expert](#)

## APPTOOLS.LSL

This library file, located in the application directory (:APP:), contains support methods used by a particular application. By default, each application contains an empty APPTOOLS.LSL.

APPTOOLS.LSL is used so frequently that it is automatically opened by each Application Framework form and library that you create.



### Tip

- The Application Event Expert enables you to quickly add methods to your application. Click Tools, Application Event to open the Application Event Expert.

## FW.LSL

This library file, located in the Application Framework directory (:FW:), contains events that control behavior across all applications you create with the Application Framework.

To create Application Framework-level events, you insert methods within this library which the Application Framework calls at the appropriate time.

For example, to have each form in every application open in edit mode, you can insert this method in FW.LSL:

```
method formOpenAfter(  
    var fwFormInfo FWForm,  
    var eventInfo Event,  
    var fwEventInfo FWEvent  
)  
  
    self.edit()  
  
endmethod
```

This method runs after the form's default open behavior and is equivalent to inserting the following code in each form:

```
method open(  
    var eventInfo Event  
)  
  
    if eventInfo.isPrefilter() then  
        ;// This code executes for  
        ;//each object on the form:  
  
    else  
        ;// This code executes only  
        ;//for the form:  
  
        doDefault  
        self.edit()  
  
    endIf  
  
endmethod
```



### Tip

- The Application Event Expert enables you to quickly add methods to your application. Click Tools, Application Event to open the Application Event Expert.

### See Also

[Application Event Expert](#)

## **FWDM.LDL**

This library file, located in the Application Framework directory (:FW), contains support methods that the Application Framework uses to generate Application Framework data model actions.



## **FWEVENT.LDL**

This library file, located in the Application Framework directory (:FW:), contains support methods that the Application Framework uses to dispatch Application Framework events.

## **FWON.LDL**

This library file, located in the Framework directory (:FW:), contains support methods which the Application Framework uses to respond to Application Framework actions.

## **FWTOOLS.LDL**

This file, located in the Application Framework directory (:FW:), contains support methods used across all of your applications.

FWTOOLS.LDL is used so frequently that it is automatically opened by each Application Framework form and library that you create.

## **Application Framework properties**

[App On Close](#)

[App Show Hint](#)

[App Show Key Viol Msg](#)

[App Show Map](#)

[App Show Right Click Menu](#)

[App Show Splash](#)

[App Show Toolbar](#)

[App Title](#)

[Author Company](#)

[Author Copyright](#)

[Author Name](#)

[Author Version](#)

[Description](#)

[Form Close Means](#)

[Form Color](#)

[Form On Close](#)

[Form On Enter Button](#)

[Form On Esc Key](#)

[Form On Open](#)

[Form On Open Edit](#)

[Form On Open Show](#)

[Form On Resize](#)

[Help File](#)

[Help Hint](#)

[Help Topic](#)

[Help Topic Contents](#)

[Help Topic Index](#)

[Menu](#)

[Menu Merge Into](#)

[Report On Open](#)

[Report On Open Show](#)

[Report On Open Zoom](#)

[Table On Delete](#)

[Table On Insert](#)

[Table On Modify](#)

## **App On Close**

### **Values**

Allow	The application closes immediately.
Confirm	The application asks the user for permission to close.

### **Applies To**

Applications

## App Show Hint

App Show Hint determines whether or not the application shows application-defined messages (hints) when the mouse passes over an object like a field or a button.

### Values

Yes	Show fly-over help (hints).
No	Don't show fly-over help (hints).

### Applies To

Applications

### See Also

[Help Hint](#)

## App Show Map

App Show Map determines whether or not the application displays the work flow map.

### Values

Yes	Show the work flow map.
No	Don't show the work flow map.

### Applies To

Applications

### See Also

[APPMAP.FSL](#)

## App Show Splash

App Show Splash determines whether or not the application shows the opening splash screen as it launches.

### Values

Yes

Show the splash screen.

No

Don't show the splash screen.

### Applies To

Applications

### See Also

[APPSPLSH.FSL](#)



## **App Show Toolbar**

### **Description**

App Show Toolbar determines whether or not the application displays the Toolbar.

### **Values**

Yes	Display the Toolbar.
No	Hide the Toolbar.

### **Applies To**

Applications

## **App Show Key Viol Msg**

App Show Key Viol Msg determines whether or not the application displays a special explanatory message when it detects a key violation.

### **Values**

- Yes            Show the key violation message.
- No            Don't show the key violation message.

### **Applies To**

Applications

## **App Show Right Click Menu**

App Show Right Click Menu determines whether or not the application displays a generic, context-sensitive pop-up menu when the user right-clicks on an item.

### **Values**

- Yes Show the right-click menu.
- No Don't show the right-click menu.

### **Applies To**

Applications

## **App Title**

App Title is the text that appears in the application's title bar.

### **Type**

String

### **Applies To**

Applications

## **Author Company**

Author Company is the name of the company that wrote the application.

### **Type**

String

### **Applies To**

Applications

### **See Also**

[Author Copyright](#)

[Author Name](#)

[Author Version](#)

## **Author Copyright**

Author Copyright is the copyright of the company or person that wrote the application.

### **Type**

String

### **Applies To**

Applications

### **See Also**

[Author Company](#)

[Author Name](#)

[Author Version](#)

## **Author Name**

Author Name is the name of the person who wrote the application.

### **Type**

String

### **Applies To**

Applications

### **See Also**

[Author Company](#)

[Author Copyright](#)

[Author Version](#)

## **Author Version**

Author Version is the version of the application.

### **Type**

String

### **Applies To**

Applications

### **See Also**

[Author Company](#)

[Author Copyright](#)

[Author Name](#)



## **Description**

Description is a short description of an object. It's generally used for documentation, although the framework will sometimes use this text in confirmation dialogs.

### **Type**

String

### **Applies To**

All

## Form Close Means

Form Close Means determines what happens when the user closes a form by selecting 'Close' from the control menu, or by typing Control+F4 (forms)/Alt+F4 (dialogs). Closing the form can be set to mean the same thing as hitting the Enter key (which is usually the OK button) or the same thing as hitting the Esc key (which is usually the Cancel button). This property is particularly useful in modal dialogs.

### Values

Same as On Enter Button      Call the pushButton event of the object defined in Form On Enter Button.

Same as On Esc Button      Call the pushButton event of the object defined in Form On Esc Button.

### Applies To

Dialogs  
Forms

### See Also

[Form On Enter Button](#)

[Form On Esc Button](#)

## Form Color

Form Color determines the colors of your application forms. By default, the Application Framework is set to System, which means that when you run your application, your forms will use the colors set in your Windows Control Panel. Alternately, you can choose Form, in which case your forms will use the colors you apply as you create the form.

### Values

System	Forms will open using the system colors set in the Windows Control Panel
Form	Forms will open using the colors set when you created the form.

### Applies To

Dialogs

Forms

## Form On Enter Button

Form On Enter Button determines what happens when the user presses the Enter key. If the form contains a pushButton with the same name as is given by this property, then that pushButton's pushButton event is called. If the form contains no such button, then the Enter key acts as usual. This property is particularly useful in modal dialogs.

### Type

UIObject Name (PushButton)

### Applies To

Dialogs

Forms

### See Also

[Form Close Means](#)

[Form On Esc Button](#)

## Form On Esc Button

Form On Esc Button determines what happens when the user presses the Esc key. If the form contains a pushButton with the same name as is given by this property, then that pushButton's pushButton event is called. If the form contains no such button, then the Esc key acts as usual. This property is particularly useful in modal dialogs.

### Type

UIObject Name (PushButton)

### Applies To

Dialogs

Forms

### See Also

[Form Close Means](#)

[Form On Enter Button](#)

## Form On Close

Form On Close determines whether or not the form asks the user to save unposted changes before closing.

### Values

Allow	The form closes immediately.
Confirm	If there are any unposted changes the form asks the user whether or not they should be saved. If the user hits the Cancel button the form won't close.

### Applies To

Dialogs

Forms

## Form On Open

### Description

Form On Open determines what happens when a form is opened.

### Values

View	View the form.
View One	If the form is already open, simply bring it to the top. If it isn't, view it.
View Modally	View the form modally. (For truly modal forms use a modal dialog; otherwise the form will be modal only to the form that opened it.)

### Applies To

Dialogs

Forms

### See Also

[Form On Open Edit](#)

[Form On Open Show](#)

## Form On Open Edit

Form On Open Edit determines whether or not the form automatically enters edit mode as it opens.

### Values

- Yes            When the form opens put it in edit mode.  
No             When the form opens don't put it in edit mode.

### Applies To

Dialogs  
Forms

### See Also

[Form On Open](#)  
[Form On Open Show](#)



## Form On Open Show

Form On Open Show determines how the form will be shown when it opens.

### Values

Normal	The form opens as it normally would.
Full Screen	The form opens just big enough to fit within the full size of the screen
Maximized	The form opens maximized.
Minimized	The form opens minimized.
Hidden	The form opens hidden.

### Applies To

Dialogs  
Forms

### See Also

[Form On Open](#)

[Form On Open Edit](#)

## Form On Resize

Form On Resize determines what happens when a form is resized.

### Values

Allow	The form can be resized.
Disallow	The form can't be resized.
Size To Fit	When the form is resized the contents of the form are adjusted to fit.

### Applies To

Dialogs  
Forms

## Help File

Help File is the name of the application help file.

### Type

Help File Name

### Applies To

Applications

### See Also

[Help Topic](#)

[Help Topic Contents](#)

[Help Topic Index](#)

## Help Hint

Help Hint is the application-defined message (hint) which is displayed when the mouse passes over an object like a field or a button.

### Type

String

### Applies To

Dialogs

Forms

UIObjects

### See Also

[App Show Hint](#)

## Help Topic

Help Topic is the topic name which is opened if a user asks for help.

### **Type**

Help Topic Name

### **Applies To**

Dialogs

Forms

UIObjects

### **See Also**

[Help File](#)

[Help Topic Contents](#)

[Help Topic Index](#)

## Help Topic Contents

Help Topic Contents is the topic name of the help file's contents page.

### **Type**

Help Topic Name

### **Applies To**

Applications

### **See Also**

[Help File](#)

[Help Topic](#)

[Help Topic Index](#)

## Help Topic Index

Help Topic Index is the topic name of the help file's index page.

### **Type**

Help Topic Name

### **Applies To**

Applications

### **See Also**

[Help File](#)

[Help Topic](#)

[Help Topic Contents](#)

# Menu

## Description

Menu is the menu associated with an object.

## Type

Menu Name

## Applies To

Forms

Reports

## See Also

[Menu Merge Into](#)



## Menu Merge Into

### Description

Menu Merge Into is the menu which the current menu merges into. For instance, App Menu has the standard File, Report, Window, and Help menus on it. Form Merge Menu has Edit and Record menus. The default menu for all forms is the Form Merge Menu, which merges into the App Menu. After they've merged, you have the full set of menus: File, Edit, Record, Report, Window, and Help.

Merging menus is an advanced topic.

### Type

Menu Name

### Applies To

Menus

### See Also

[Application Framework Menu](#)

[Menu](#)

## Report On Open

Report On Open determines what happens when a report is opened.

### Values

View	View the report.
View One	If the report is already open, simply bring it to the top. If it isn't, view it.
View Modally	View the report modally. (Currently the report will be modal only to the form that opened it.)
Print	Print the report.

### Applies To

Reports

### See Also

[Report On Open Show](#)

[Report On Open Zoom](#)

## Report On Open Show

Report On Open Size determines how the report will be shown when it opens.

### Values

Normal	The report opens as it normally would.
Full Screen	The report opens just big enough to fit within the full size of the screen.
Maximized	The report opens maximized.
Minimized	The report opens minimized.
Hidden	The report opens hidden.

### Applies To

Reports

### See Also

[Report On Open](#)

[Report On Open Zoom](#)

## Report On Open Zoom

### Description

Report On Open Size determines what zoom setting the report will use when it opens.

### Values

25%	The report is shown at 25% of actual size.
50%	The report is shown at 50% of actual size.
100%	The report is shown at actual size.
200%	The report is shown at 200% of actual size.
400%	The report is shown at 400% of actual size.
Best Fi	The report is scaled so that an entire page fits within the window.
Fit Height	The report is scaled so that its height fits within the window.
Fit Width	The report is scaled so that its width fits within the window.

### Applies To

Reports

### See Also

[Report On Open](#)

[Report On Open Show](#)

## Table On Delete

Table On Delete determines what happens when the user tries to delete a record from the given table.

### Values

Allow	The record can be deleted.
Confirm	The application asks the user for permission to delete the record.
Disallow	The record can't be deleted.

### Applies To

Tables

### See Also

[Table On Insert](#)

[Table On Modify](#)

## Table On Insert

### Description

Table On Insert determines what happens when the user tries to insert a record into the given table.

### Values

Allow	The record can be inserted.
Confirm	The table asks the user for permission to insert the record.
Disallow	The record can't be inserted.

### Applies To

Tables

### See Also

[Table On Delete](#)

[Table On Modify](#)

## Table On Modify

### Description

Table On Delete determines what happens when the user tries to modify a record in the given table.

### Values

Allow	The record can be modified.
Confirm	The table asks the user for permission to modify the record.
Disallow	The record can't be modified.

### Applies To

Tables

### See Also

[Table On Delete](#)

[Table On Insert](#)

## **New Application dialog box**

Use this dialog box to specify the name and location of your new application.

### **Title**

The name and title of the new application.

### **Exe Path**

The folder where you want Corel Paradox to save all application objects. You must specify a folder that does not currently exist. The Application Framework copies files into the folder.

### **Tables Path**

The folder where you want Corel Paradox to save all the tables that you will be creating within the new application. By default, Corel Paradox specifies a Tables folder within the folder specified in the Exe Path box. This can be an existing folder.



## **New Form dialog box**

Use the New Form dialog box to specify properties for the form.

### **Title**

The name of the form.

### **Use With**

Enable This Application to use the form only with the current application.

Enable All Applications to use the form in multiple applications.

### **Include in Application Menu**

Enable to include the form name in the default application Menu Bar, APP MENU.

### **Sample 1 area**

Use the list box and the arrow buttons to specify the position of the form within the Menu Bar of the application. Choose a menu item from the list box, and then click the up arrow to move the item up, and the down arrow to move the item down. Typically, you will place forms in File, and reports and scripts in Reports.

### **Dialog**

Enable to make the new form a dialog. Dialogs are separate windows that you can move around the Corel Paradox Desktop.

### **Modal**

When Dialog is enabled, enable to create a modal dialog box. Modal dialogs must be closed before you can continue working in an application. Typically, modal dialogs prompt the user for responses which affect what happens next in an application.

### **Style Sheet**

Specifies a style sheet for the new form.

### **OK**

Saves the specified settings and closes the dialog box. Corel Paradox creates the form.

### **Cancel**

Closes the dialog box without saving any settings.

### **Help**

Opens this Help topic.

## **New Report dialog box**

Use the New Report dialog box to specify properties for the report.

### **Title**

The name of the report.

### **Use With**

Enable This Application to use the report only with the current application.

Enable All Applications to use the report in multiple applications.

### **Include in Application Menu**

Enable to include the report name in the default application Menu Bar, APP MENU.

### **Sample 1 area**

Use the list box and the arrow buttons to specify the position of the report within the Menu Bar of the application. Choose a menu item from the list box, and then click the up arrow to move the item up, and the down arrow to move the item down. Typically, you will place forms in File, and reports and scripts in Reports.

### **Style Sheet**

Specifies a style sheet for the new form.

### **OK**

Saves the specified settings and closes the dialog box. Corel Paradox creates the report.

### **Cancel**

Closes the dialog box without saving any settings.

### **Help**

Opens this Help topic.

## **New Script dialog box**

Use the New Script dialog box to specify properties for the script.

### **Title**

The name of the script.

### **Use With**

Enable This Application to use the script only with the current application.

Enable All Applications to use the script in multiple applications.

### **Include in Application Menu**

Enable to include the script name in the default application Menu Bar, APP MENU.

### **Sample 1 area**

Use the list box and the arrow buttons to specify the position of the script within the Menu Bar of the application. Choose a menu item from the list box, and then click the up arrow to move the item up, and the down arrow to move the item down. Typically, you will place forms in File, and reports and scripts in Reports.

### **Style Sheet**

Specifies a style sheet for the new script.

### **OK**

Saves the specified settings and closes the dialog box. Corel Paradox creates the script.

### **Cancel**

Closes the dialog box without saving any settings.

### **Help**

Opens this Help topic.

## Application Menu dialog box

Use the Application Menu dialog box to customize your application's Menu Bar. The Application Framework stores this information in your application's APPMENU.DB file.

### Application Menu list box

The list box at the top of the dialog box that contains the following items:

APP MENU: The Menu Bar that appears when your application opens.

FORM MERGE MENU: A Menu Bar that merges with the APP MENU. By default, the Application Framework sets the Menu property for each form so that this menu appears whenever you open a form.

REPORT MERGE MENU: A Menu Bar that merges with the APP MENU. By default, the Application Framework sets the Menu property for each form so that this menu appears whenever you open a report or script.

### Merge ID

An identification number that specifies the position of an item in the Menu Bar relative to items in a menu that merges with that menu. The Application Framework only refers to the Merge Id number to determine how to merge the App Menu with either the Form Merge Menu or the Report Merge Menu. For example, suppose you design the App Menu to have two main menu items called File and Report and assign merge id of 10 and 30 to them, respectively. Then, you design your Form Merge Menu with one item called Edit and assign to it a merge id of 20. When you run your application, the Application Framework refers to these numbers and positions the Edit menu item between the File and Report menu items. If you accidentally use the same Merge ID number for two menu items on the same level, the Form and Report Merge Menu items always override the App Menu.

### Menu Text

The text of the menu item as it appears in the Menu Bar of the application. Often the menu text is the same as the Object Name.

### Menu Key

Allows you to specify (by clicking the ellipses (...) button) a shortcut key for the menu item.

### Menu Type

- Application: Enable if the menu item is a Corel Paradox object (form, report or script).
- Paradox: Enable if the menu item is an ObjectPAL menu action.

### Menu ID

When the Application Menu Type is enabled, type a unique identification number for that item in the menu.

When the Paradox Menu Type is enabled, select the ObjectPAL method you want to assign to the menu item.

### Object Type

- None: Enable when you are adding an item that is not a Corel Paradox object.
- Form: Enable when you are adding a form as a menu item.
- Script: Enable when you are adding a script as a menu item.
- Dialog: Enable when you are adding a dialog as a menu item.
- Report: Enable when you are adding a report as a menu item.
- Exe: Enable when you are adding an executable file as a menu item.

### Object Path





The working directory in which the form, report, script, or executable file you are adding is stored.






### Object Name

The name of the object you are adding as a menu item.

### Application Menu Icons

The following icons control the object tree displayed on the left side of the Application Menu.


-  Inserts a new item in the menu. Use the arrow buttons to move the new item to the desired place.
-  Deletes the selected menu item from the Menu Bar
-  Moves the selected menu item up one place in the list
-  Moves the selected menu item down one place in the list

-  Moves the selected menu item one place to the left
-  Moves the selected menu item one place to the right
-  Expands or collapses the selected main menu item in the object tree
-  Expands all main menu items in the object tree
-  Collapses all main menu items in the object tree

## Open Application Menu dialog box

Use the Open Application Menu dialog box to open an APPMENU.DB file. The APPMENU.DB file stores the structure of the Menu Bar for any application created using the Application Framework. Once you open an APPMENU.DB file, you can use the Application Menu dialog box to customize that application's Menu Bar by adding, deleting or moving menu items.

The APPMENU.DB file for an application is stored in the application's APPTBLS folder. For example, if your application is stored on your C: drive, in a MYAPP folder, the APPMENU.DB file is located in C:\MYAPP\APPTBLS.

Click  and click an item for Help on that item.

# Project Viewer Application Framework

Use the Project Viewer to manage the objects in your application.

The Project Viewer shows the objects in your application directory. The list box at the top of the Viewer displays the application that you are working with.

## Managing files

Choose the type (Tables, Forms, Queries, SQL, Reports, Scripts, Libraries, or Data Models) of object you want to see or choose Custom to see all objects in your application. Click Applications to display the application's executable (.EXE) file. The icon shown to the left of each object indicates the type of object displayed.

Right-click the name of an object for a menu of commands you can use on the object. The commands available depend on the object. Double-click an object to perform the default action (first item on the menu).



### Note

- In the Application Framework, the Project Viewer allows you to select any alias, including your working and private directories. The Application Framework does not support references as the Project Viewer does when you are not working with the Application Framework.

## Framework properties

Depending on the object, you can set one or more of the following properties.

[App On Close](#)

[App Show Hint](#)

[App Show Key Viol Msg](#)

[App Show Map](#)

[App Show Right Click Menu](#)

[App Show Splash](#)

[App Show Toolbar](#)

[App Title](#)

[Author Company](#)

[Author Copyright](#)

[Author Name](#)

[Author Version](#)

[Description](#)

[Form Close Means](#)

[Form Color](#)

[Form On Close](#)

[Form On Enter Button](#)

[Form On Esc Key](#)

[Form On Open](#)

[Form On Open Edit](#)

[Form On Open Show](#)

[Form On Resize](#)

[Help File](#)

[Help Hint](#)

[Help Topic](#)

[Help Topic Contents](#)

[Help Topic Index](#)

[Menu](#)

[Menu Merge Into](#)

[Report On Open](#)

[Report On Open Show](#)

[Report On Open Zoom](#)

[Table On Delete](#)

Table On Insert

Table On Modify



**Exit command (File menu)**

Closes the Application Menu dialog box.

**New command (Menu menu)**

Creates a new Menu Bar for your application.

**Open command (Menu menu)**

Displays the App Menu for your application.

**Save command (Menu menu)**

Saves any changes you have made to a menu.

**Revert command (Menu menu)**

Discards unsaved changes to a menu.

**Duplicate command (Menu menu)**

Enables you to duplicate the structure of the current menu under a new name.

**Delete command (Menu menu)**

Deletes the current menu.

**Close command (Menu menu)**

Closes the APPMENU.DB file for the application.



**Import command (Menu menu)**

Enables you to import a menu from another application.

**Expand All command (Menu menu)**

Displays all menu items in the current menu.

**Collapse All command (Menu menu)**

Closes the menu structure so that only main menu items are displayed.

**Insert command (MenuItem menu)**

Inserts a new menu item.

**Delete command (MenuItem menu)**

Deletes the selected menu item.

**Up command (MenuItem menu)**

Selects the menu item above (the same as pressing the Up Arrow).

**Down command (MenuItem menu)**

Selects the menu item below (the same as pressing the Down Arrow).

**Move Up command (MenuItem menu)**

Moves the selected menu item up one place.



**Move Down command (MenuItem menu)**

Moves the selected menu item down one place.

**Move Left command (MenuItem menu)**

Moves the selected menu item one place to the left.

**Move Right command (Menuitem menu)**

Moves the selected menu item one place to the right.

**Expand/Collapse command (MenuItem menu)**

Expands or collapses the selected menu item.

**Help Topic command (Help menu)**

Opens the Application Framework Help file.

