



Interface Objects:

(buttons, text fields and menus)
the building blocks of an application

This topic introduces you to the different types of object that can be created using the tools palette and what they are used for. You'll learn how to lay these objects out on a card to create an interface for an application and also how to use the editing and runtime modes to build and test as you go.

Key topics covered in this tutorial

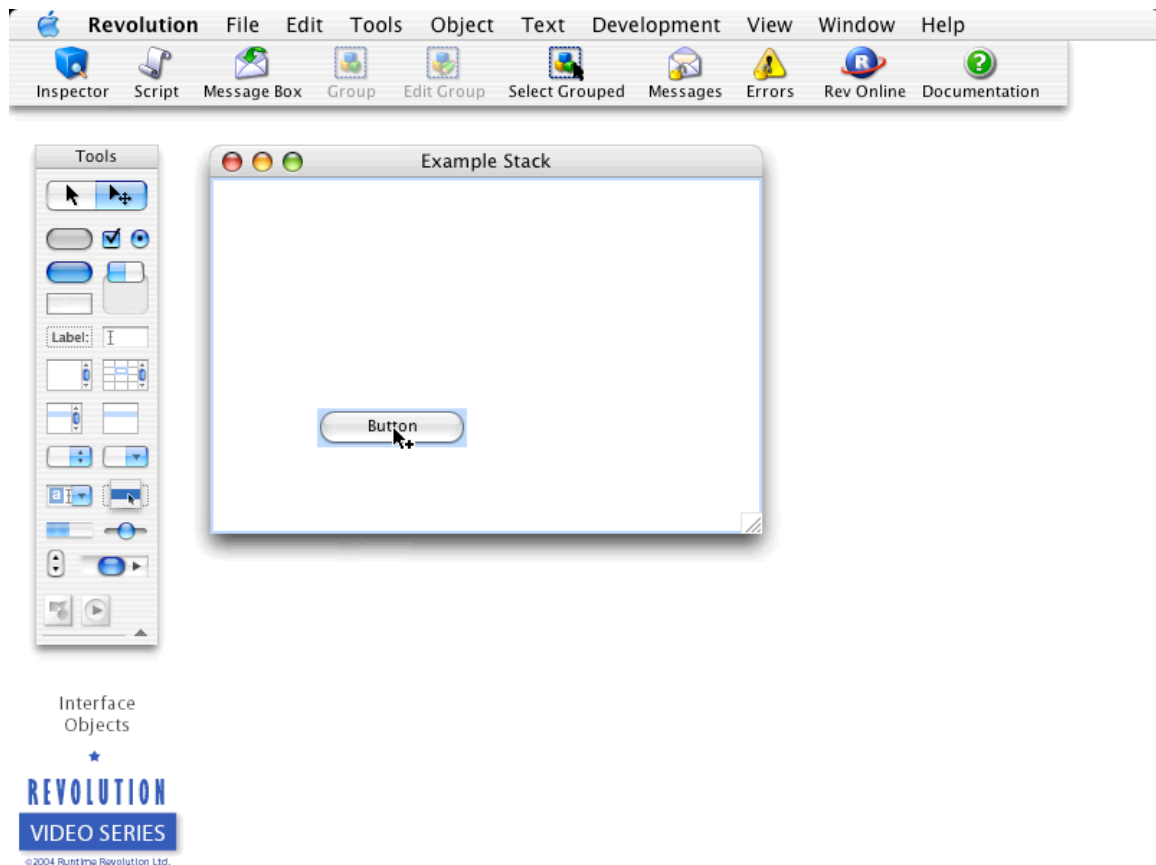
- Using the tools palette
- Laying out an application interface
- Modes – editing and runtime

See also: [Reference: Object types and object references](#)

The tools palette allows you to create any standard user interface object, such as a button, checkbox, text field or graphic. Let's take a look at each object and what it is for. In this tutorial we're concentrating on making standard objects – in other words objects that look and behave in the same standard way they do in other applications on your computer. In the tutorials on customizing objects, we cover how to customize objects to create your own unique look and set of behaviors.

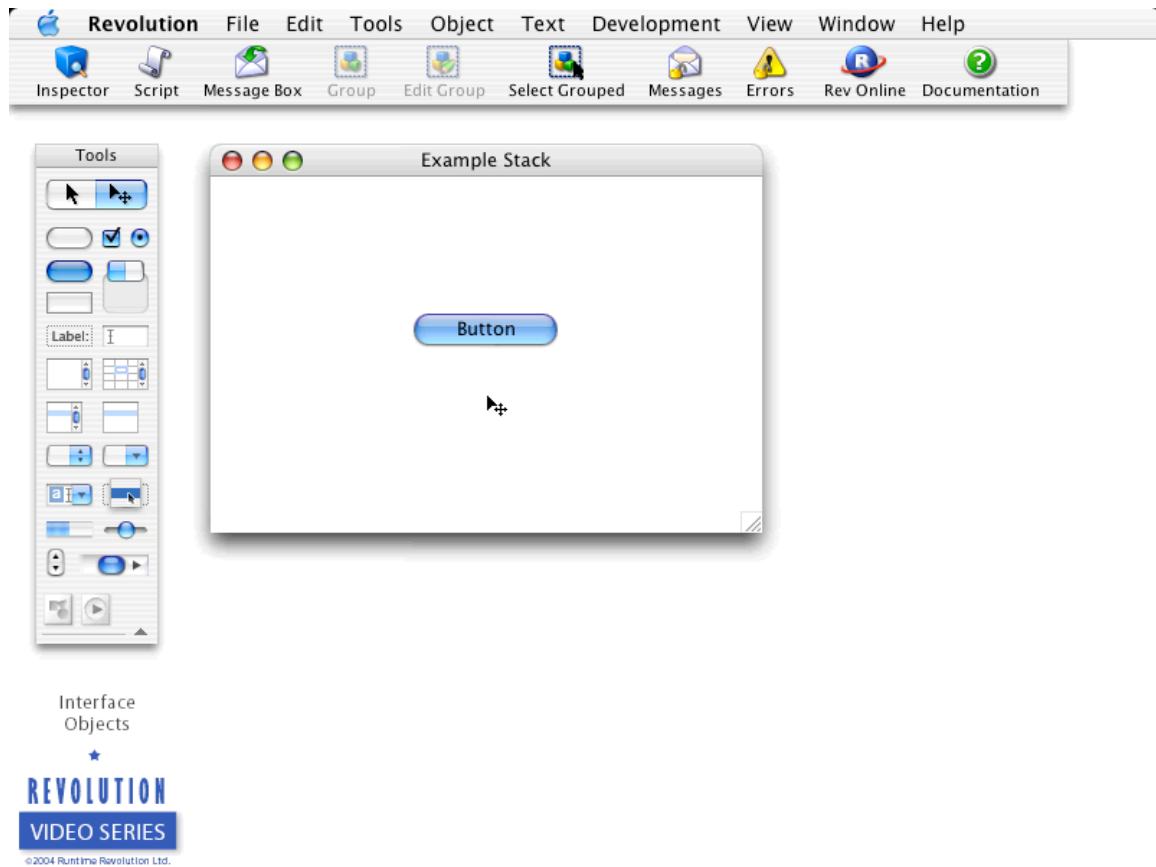
You create an object by holding down and dragging the object from the tools palette to your card window.

A button is one of the most simple objects you can create. Buttons are used for a variety of purposes - allowing a user to start something happening, to respond to a question or to bring up a dialog box.



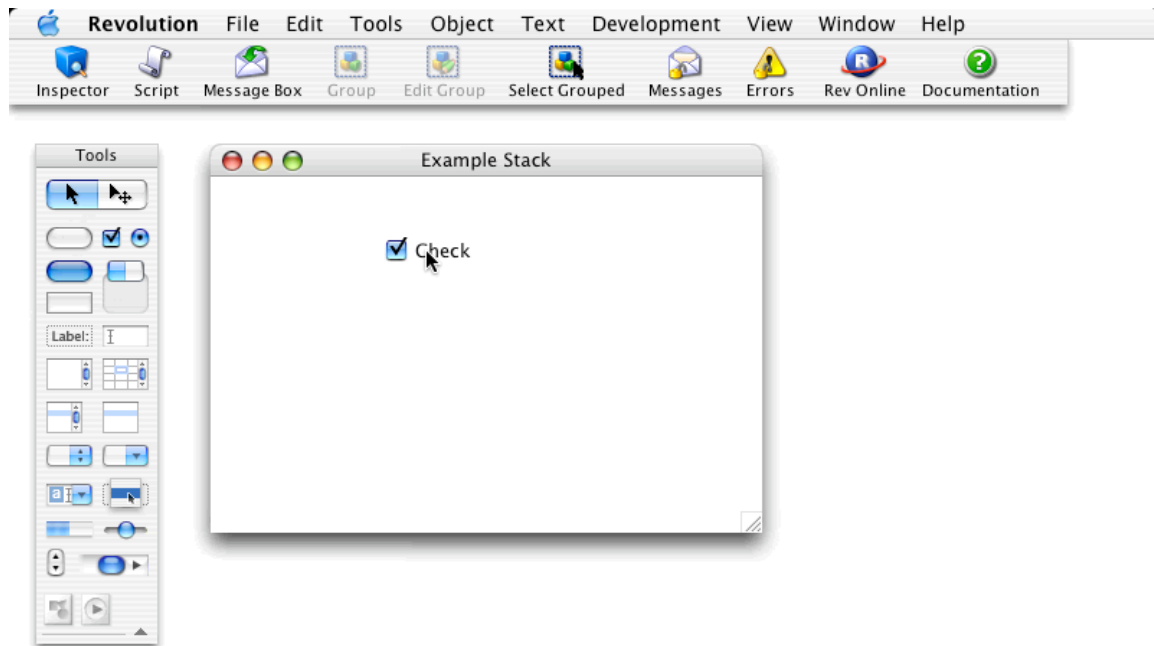
To edit an object after you have created it, click on the pointer tool (the arrow at the top right of the tools palette) to switch to editing mode. The pointer tool then allows you to select, move and resize objects. Clicking the browse tool (the arrow at the top left of the tools palette) switches to runtime mode and allows you to try out the application you are working on. In runtime mode, the button we created will behave as you would expect - if we click on it, it will highlight. This button doesn't do anything yet, because we haven't added a script to it to tell it what to do.

The default button is similar to a regular button, except that it is used to indicate a default choice such as 'OK', or 'Save'.

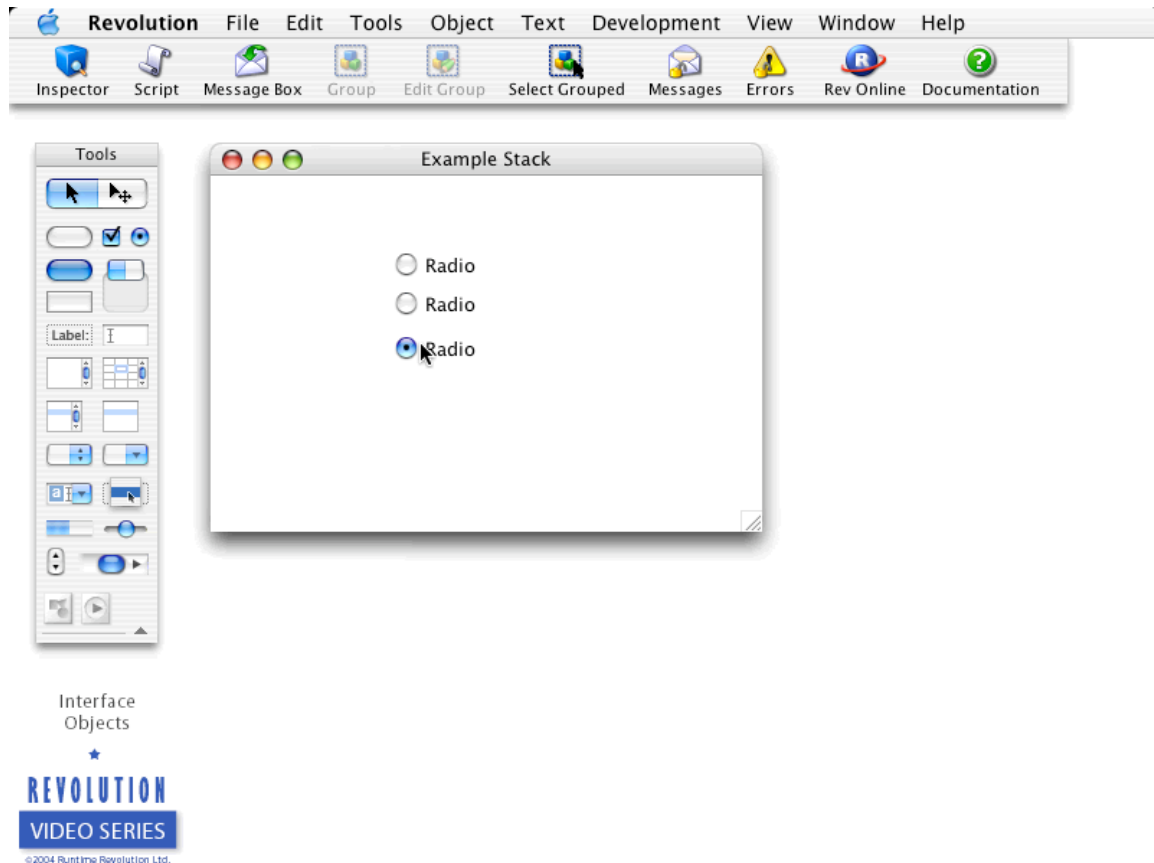


The square button is similar to a normal button but can be used to give a different look to the design of your application.

Next, let's take a look at a checkbox. A checkbox is a button which can either be checked or unchecked. You can use checkboxes to allow a user to specify a preference or a setting in your application.

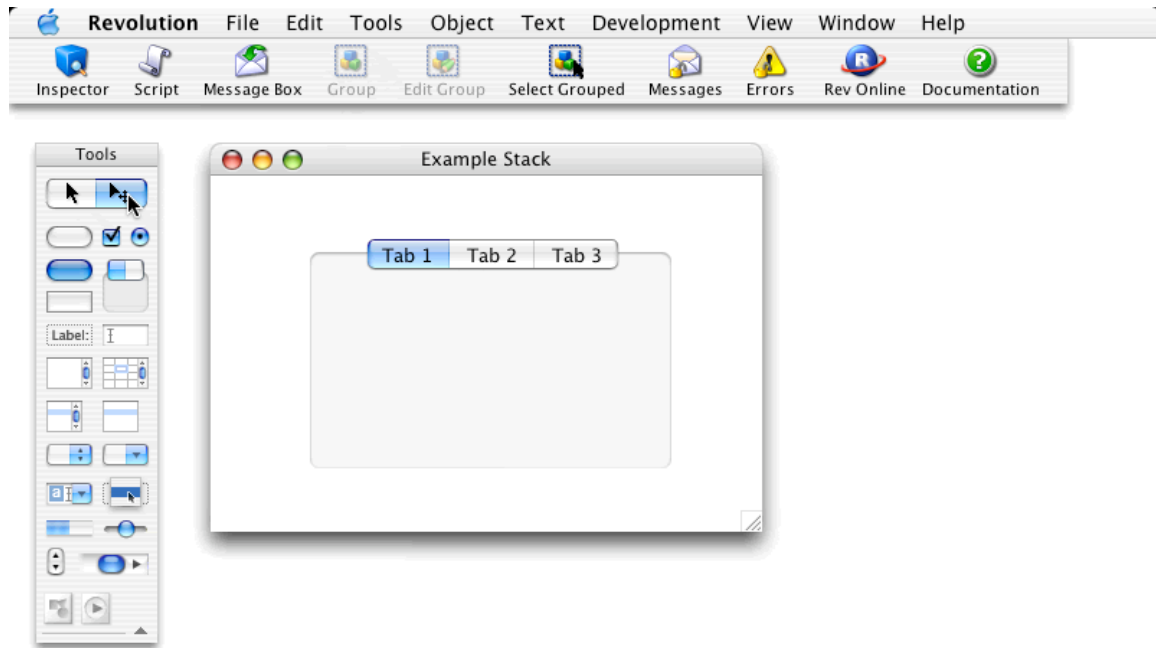


A radio button is named after the frequency buttons on a radio – where only one of the buttons can be depressed at once. While checkboxes are simply on or off, a set of radio buttons allows only one to be on at any time. To tell Revolution which set of radio buttons go together, you place them in a group. Let's select the buttons we've created and group them together and try it out. You can see that only one of the buttons is highlighted at a time.



You can group all kinds of objects together. When you group objects, they become an object in their own right with a shared set of properties and behaviours. For example, you can hide or show a group all at once, rather than having to hide and show each object separately. You can also place instances of the same group onto different cards in your application, allowing you to reuse groups of objects quickly. To create a group, simply select some objects you have already created and use the group button to group them together. We'll talk more about this in the tutorial 'Cards and backgrounds'.

A tabbed button can be used to create an application that displays different sets of information in the area underneath the tabs. Like radio buttons, only one tab can be selected at a time. However while radio buttons can give the user a choice, tabbed buttons are usually used to navigate to different screens in your application. We talk more about creating a tabbed button interface in the tutorial 'Cards, stacks and backgrounds'.

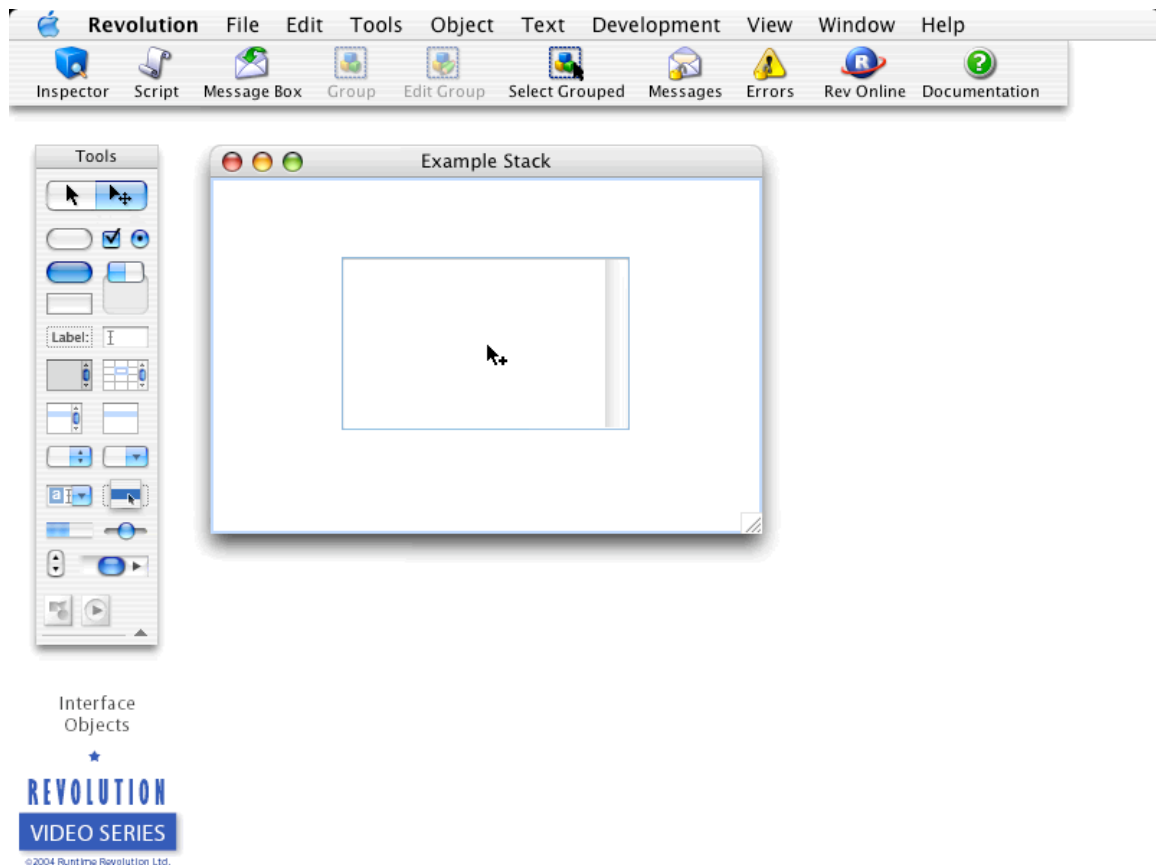


Now let's take a look at a different class of interface objects, called fields. Fields are used to display text on the screen or to allow a user to enter or edit text.

The label field is used to label objects in your application. A label field can be edited by you when you create the application but can't be edited by the user when the application is running.

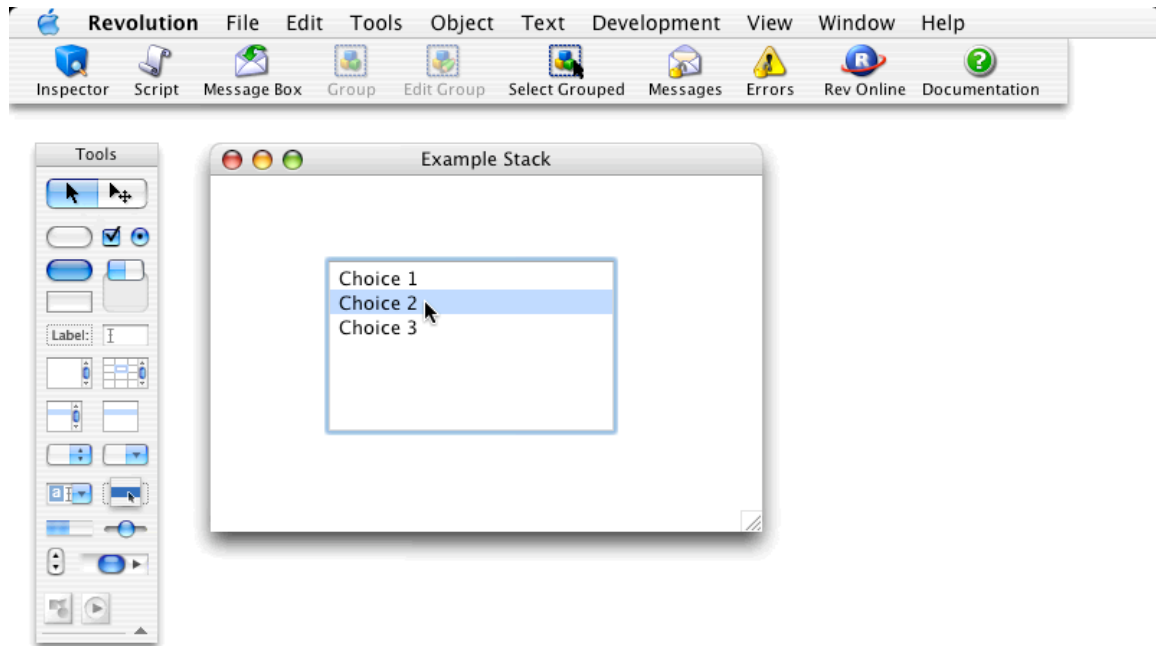
The text field is used when you want to allow the user to type or edit text. We've already used a text field to allow the user to enter text for different purposes in the tutorial 'Making your first application'.

The scrolling field is the same as the regular text field but includes a scroll bar so that the user can work with text that overruns the length of the field itself. This is the kind of field you have seen in word processor applications.



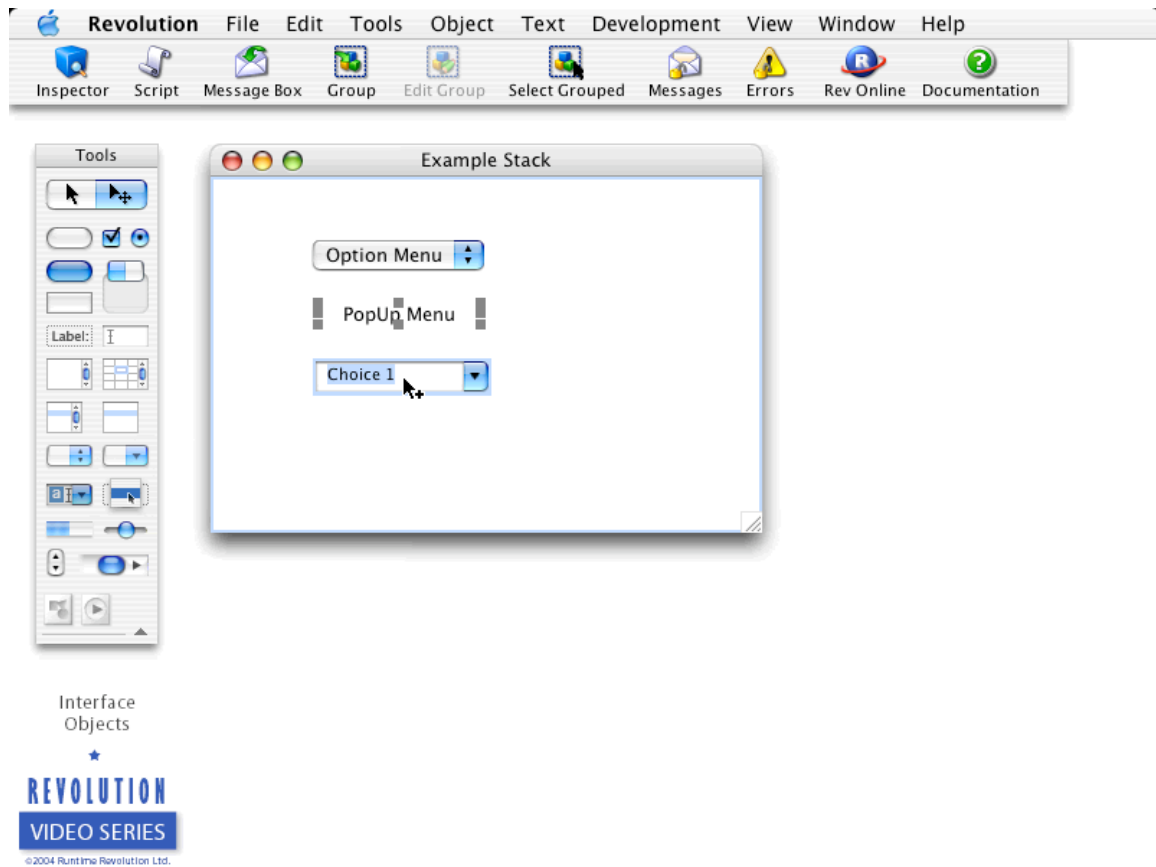
The table field allows you to create a field which behaves like a spreadsheet. You can edit individual cells, as well as set the cell width, height, number formatting and other properties.

The list field is used to provide a set of choices that a user can choose from. You can put choices into it when you create your application. The user cannot edit the field when the application is running, but can click on one line or another to indicate a choice. The scrolling list field is similar to the list field but allows scrolling as well.



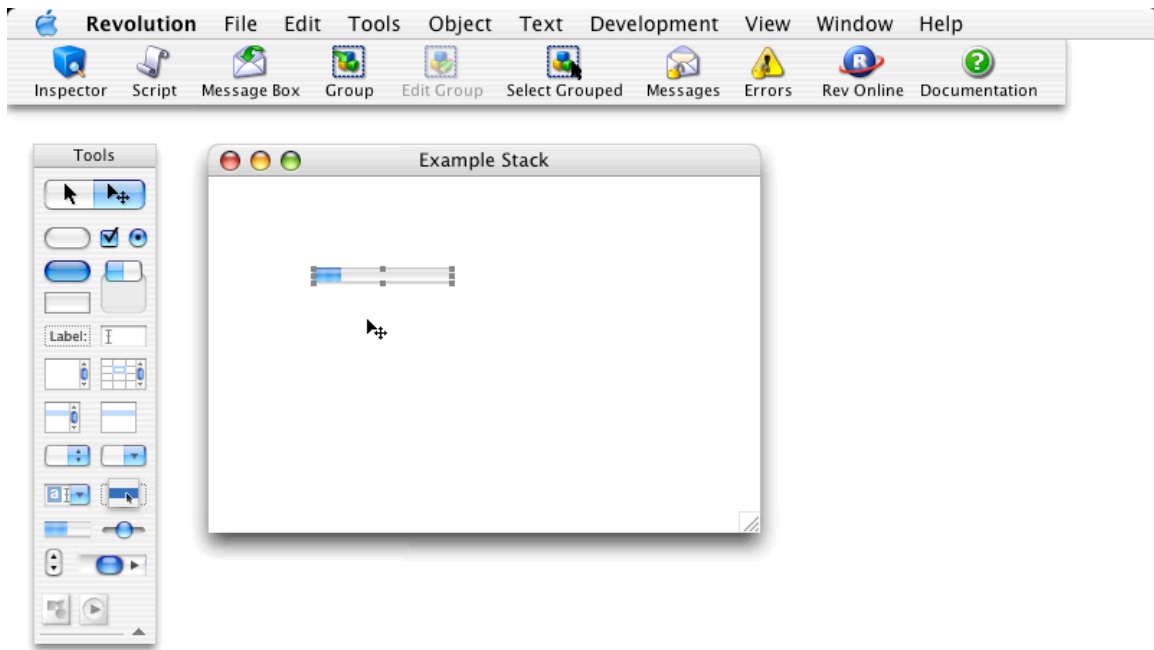
The next class of interface objects are menus. Menus are used to provide a list of choices that pop up when the user clicks on them – the advantage is that the choices are only displayed when the user clicks on the menu, saving space the rest of the time.

The main types of menu are the option menu, the popup menu, and the combo box. You can see each menu type looks a little different. The option menu and the popup menu contain a list of options. The combo box allows you to see a list of options or allows the user to type in a choice of their own.



The pulldown menu [drag one out] can be used to create a standard menu bar at the top of the screen. We'll cover using menus and creating menu bars in the tutorial on 'Menu bars'.

The progress bar allows you to display feedback to a user about how long a task will take.

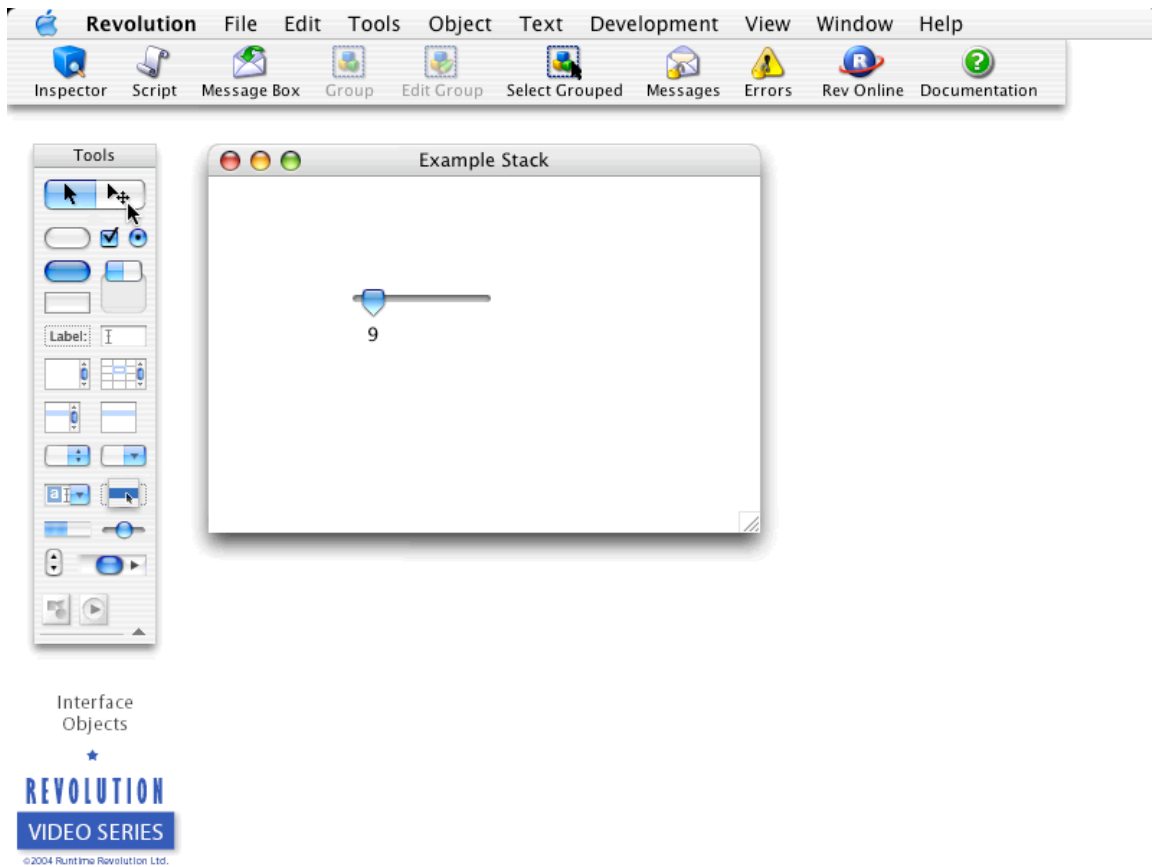


Interface
Objects

★
REVOLUTION
VIDEO SERIES

©2004 Runtime Revolution Ltd.

The scale bar allows the user to choose a setting from a range of possible options, for example as a volume control.

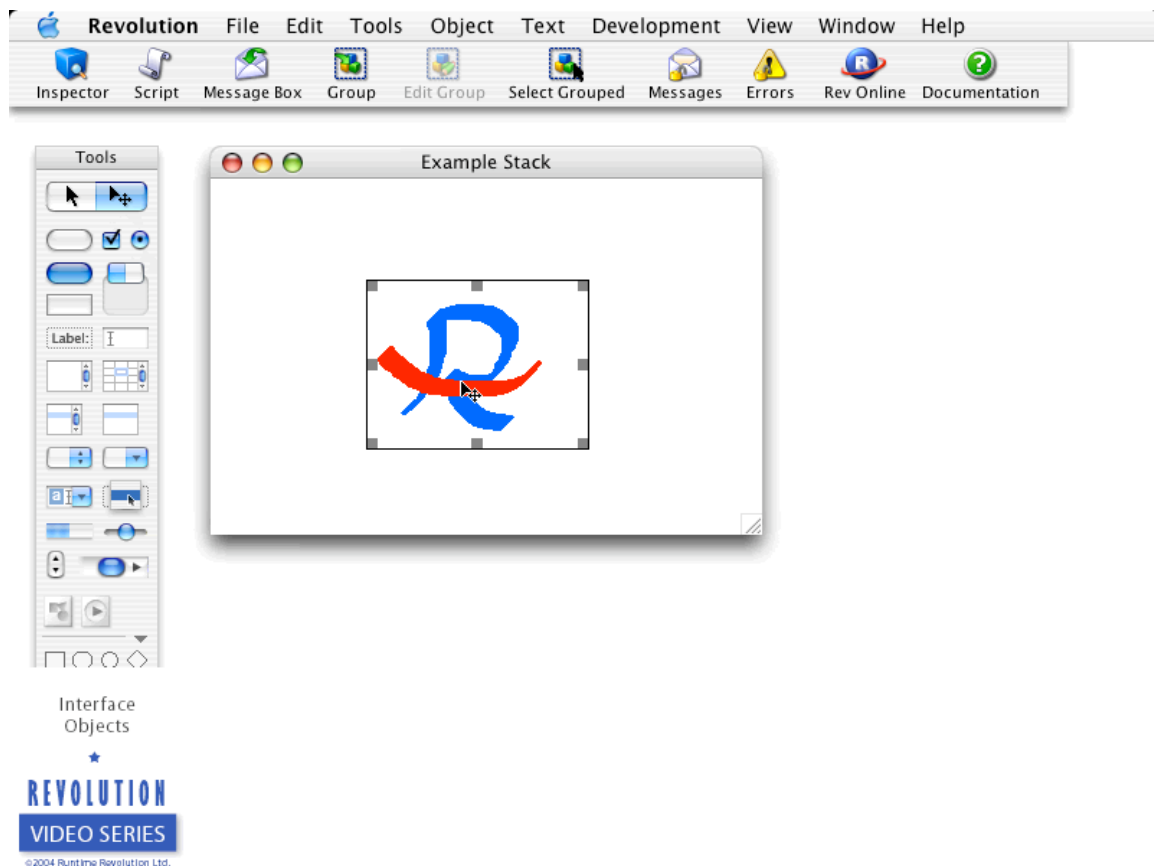


The image object allows you to place an image on the application interface. Only the frame of the image shows initially – you can either allocate an image file (e.g. JPG) on your computer to be displayed here, or you can use the built-in paint tools to paint an image of your own.

The player object allows you to play back media using QuickTime, such as QuickTime movies, sound, a virtual reality panorama, or an interactive Flash animation. We'll cover using the player object in more detail in the tutorial on 'Media and QuickTime movies'.

Folding out the tools palette displays the drawing and painting tools. The first set of tools are the graphic tools. Use graphic tools to draw geometrical shapes or freehand graphics directly onto the window. Each line, box or other graphic you create is an object in its own right – although you can group them as before to create single objects.

The set of tools below are the paint tools. To use them, first add an image frame. [drag out an image object] You can only paint within an image frame - if you do try to paint without placing an image frame on the card first, Revolution creates an image frame that is the size of the entire card as soon as you start to paint. Whatever you paint in the image frame, the image itself is treated as a single object.



The last object is the scroll bar, normally you will scroll objects using scrollbars attached to groups, or text using scrollbars attached to fields. However, the scroll bar object can be used to create customized scrolling behavior, for example scrolling several fields or a group of objects and text together.

To conclude this tutorial, we want to draw your attention to a very powerful feature of Revolution. We have shown you how you can manually add and manipulate objects on the interface of your application. The real power of Revolution is that you can also do this using scripts. So you can write a script that will add a button to an interface, or change its name, or change the size of a text field or even plot graphics. This feature is crucial as you start to develop more sophisticated applications.

Here's a simple example. First, create a button. Now let's add a script:

```
Set the width of me to the width of me + 50
```

```
Set the name of me to "splat!"
```

The script tells the button to increase its width by 50 pixels and change its name to splat when we click on it.

