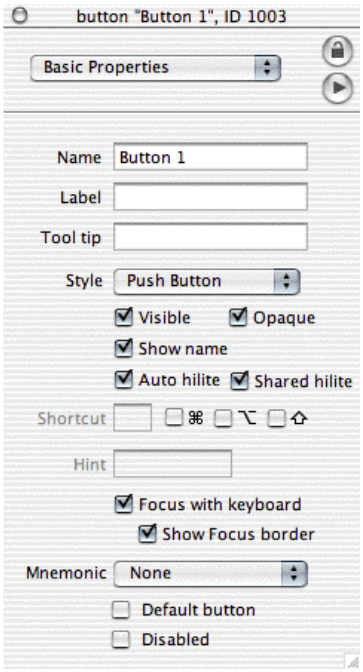
Computers & the Humanities 281

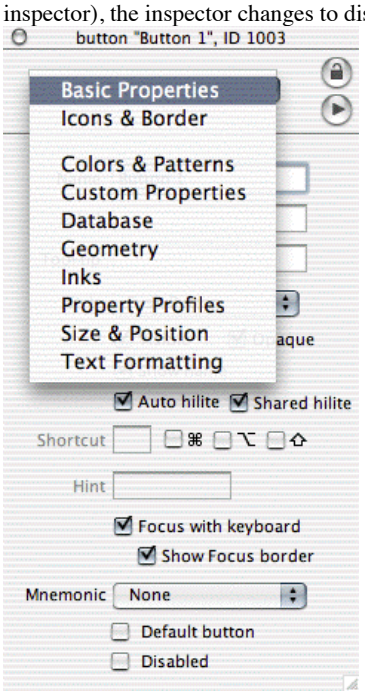
Properties of Objects in Revolution

As stated in the previous lecture, objects are the basic building blocks in the Revolution authoring environment. There are several different types of objects: stacks, cards, buttons, fields, graphics, etc. Apart from being fundamentally dissimilar, they also have characteristic variances within object types. Just like Lego blocks, these objects all come in different shapes, sizes, and colors. In Revolution, these characteristics are called "properties."

All objects in Revolution have properties or distinct characteristics that determine how they appear and behave. As with Legos the objects can be very generic or very specialized, all depending upon what the specific properties of the object are. The properties of an object can be accessed easily by double-clicking on the desired object with the pointer tool, or by selecting the appropriate properties under the **Object** menu. This will cause the properties of the particular object to be displayed in the Property Inspector.



Once the Property Inspector is visible, each time a different object is selected with the pointer tool (or with the selection menu in the top right of the inspector), the inspector changes to display the properties for that object.

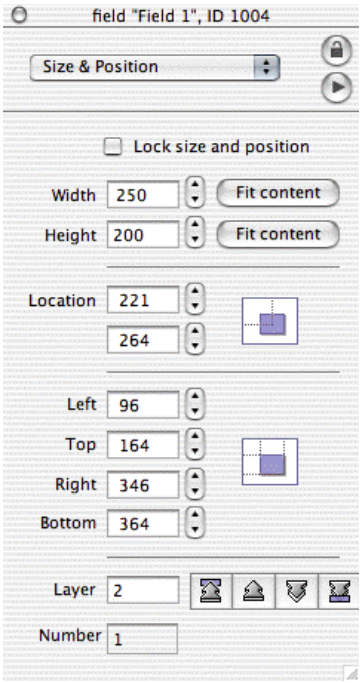


The properties for each object have been organized into different sections. At the top of the inspector is a menu through which one can jump between sections. By default the **Basic Properties** is selected every time the inspector is opened (as displayed above). The different sections available depend upon the object selected. We will first explore the common properties, and then move into important properties that are unique to various objects.

- **Name:** Every object has a name by which it can be identified. Revolution will assign a name to each object upon its creation by default. To avoid confusion it is good practice to give important objects meaningful names (never give a stack a name starting with "rev"). An object name can be several words with spaces. However, such a name can be somewhat unwieldly.
- **Visible:** This will determine if an object can be seen or not.

- **Disabled:** This determines if the object's functions are available.
- **Opaque:** This determines if the object is transparent or not.

You do need to be aware that these are the basic properties which most objects share. Some of the other common properties are found in other sections.



Under **Size & Position** we find the following common properties:

- **Lock size and position:** Checking here will make it impossible to change the location or dimensions of the object.
- **Width/Height:** The dimensions of the button are given in pixels. Altering the numbers here (rather than using the handles) gives one a greater degree of precision in altering the dimensions of the object.
- **Location:** The x and y values of the center of the object are given in reference to the top left corner of the stack. Changing the values here (rather than dragging the object to a new location) gives one a greater degree of precision in determining the location of the object.
- **Layer:** This number identifies the object in relation to other objects (buttons, fields, images, etc.) on the card, again in the order in which they were created. This can be changed by either typing a new number or by using the arrow buttons to the right.
- **Number:** This number identifies the object in relation to other objects of its type essentially in the order in which they were created. This can be changed by using the arrow buttons to the right.

The other common properties are addressed in Inheritance below.

Properties for Particular Objects

A careful observation of the Property Inspector reveals that the properties displayed change according to the object selected, sometime quite drastically. We will cover only certain properties of which you need to be aware. A number of the properties have names that are more or less self-explanatory and will not be covered here. Exploration on your part is strongly encouraged.

Stack

There are just a few stack properties about which you need to be concerned.

- **Basic Properties: Title.** Give every stack you make a meaningful Title. This is the text displayed in the title bar of your stak. If you do not, it displays the file name you have assigned the stack with an asterisk reminding you that you have not given it a title. In other words, it looks unprofessional to leave it titleless.
- **Size & Position: Resizable.** This is checked by default, allowing you to grab the lower right corner and change the dimensions of the stack (the cards assume the same size as the stack). Once you have the desired dimensions for your stack, it is best to uncheck this box as to not allow the user to alter the appearance of your stack.

Card

There is just one card property that concerns us.

- **Basic Properties: Number.** Here you can alter the order in which the cards are organized. Changing the number here changes the placement of the card in relation to the other cards in the stack.

Group

The essential properties of groups will be covered in a later lecture. Be very afraid.

Button

Since buttons are one of the main control objects we will utilize, there are a few properties with which we need to be familiar.

- **Basic Properties: Label.** Any text placed here will be displayed by the button. If left blank, the button's name is utilized instead. This all hinges upon whether the **Show name** property is checked or not.
- **Basic Properties: Tool tip.** Any text placed here will be displayed as a type of help when the user hovers the mouse over the object.
- **Icons & Border: Icon.** Here we may give the button a meaningful icon to indicate its function.

Field

Fields are another control object we will use quite frequently.

- **Basic Properties: Don't wrap.** By default all text typed in a field will automatically wrap to the next line when the end of the field is reached (i.e., it will insert soft returns). Checking this will cause all text to be confined to one line unless interrupted by a hard return.
- **Basic Properties: Lock text.** By default a field is not locked and a person can enter text into it (as evidenced by the i-beam). When checked the user cannot type in the field, i.e., they cannot alter whatever you may have put in that field, so it is a **great** idea to always lock your information fields in your stack.
- **Basic Properties: Focusable.** Enables a user to tab into the field for editing, if the field is not locked. If the text is locked the user can select text and scroll within the field if this box is checked. If not checked, a user cannot type or select text.
- **Basic Properties: Scrollbars H/V.** Selecting either one of these will give the field a scrollbar that allows the user to scroll through the information in the field.
- **Contents.** The field at the bottom of the palette displayed the text in the field. Text may be added/edited here (even if the field is locked).
- **Table: Text baselines.** Checking this option will give the field the appearance of lined paper.
- **Table: Tab stops.** Providing a value here will allow one to indent text using the tab key. The number given is the indent value in pixels.

Image

These objects have one useful property we'll address.

- **Basic Properties: Blend.** If checked, this will allow you to determine the degree of opacity of an image.

Graphic

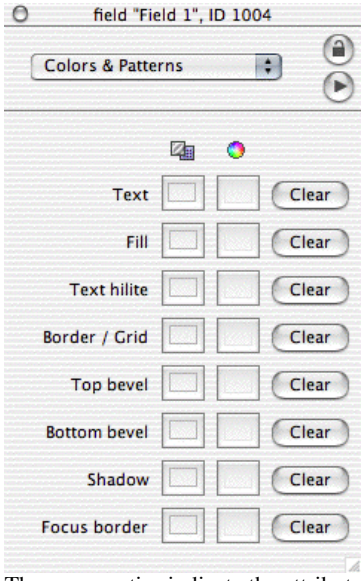
The available properties change depending upon the type of graphic object. All of them have to do with physical attributes of the graphic and allow you to change those properties manually here. **Basic Properties: Type.** will allow one to alter the type of graphic without creating a new one.

Inheritance

There is an object hierarchy in Revolution through which properties are passed or "inherited," much like physical characteristics and personality traits within a biological family. Properties for a particular object are "inherited" from another object higher in the hierarchy. It may be viewed in terms of ownership: Stacks contain and therefore own cards and groups, which two in turn contain and own controls objects such as buttons, fields, images, etc. In biological terms, stacks are grandparents, cards and groups are parents, and control objects are the children. Properties are passed or inherited through that hierarchy. Consequently, if a stack has a particular property value, then all objects owned by that stack inherit the same property value. Changing that particular characteristic subsequently changes the property for all objects below that object in the hierarchy (here the biological metaphor breaks down). Property values are passed only down the hierarchy. They never move up, nor do they move laterally within the same level.

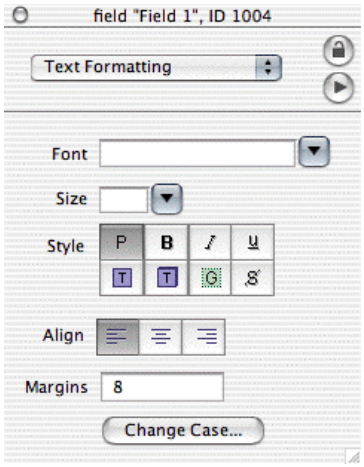
There are only a few inherited properties which concern us:

Color



These properties indicate the attributes of an object that may receive color. Here one can choose colors other than those received through inheritance. There are numerous and various colors that may be utilized for any given object and many attributes to color. The most relevant properties you would want to change with an object would be the **Text** and the **Fill**. For most objects the text color determines the color of the text being displayed by the object. The fill color, of course, indicates the color of the object itself. There are variations between objects and within object types which will be discovered upon experimentation. Click the **Clear** button to reset the color value to what the object had inherited through the hierarchy.

Text



These properties determine how the text for the object is displayed (e.g., font, size, and alignment within the object). The margins surronding that text can also be edited. As mentioned before, when these properties are altered, all objects in the hierarchy below are altered accordingly.

As hinted to previously, there are so many other properties that we cannot adequately cover them here without causing you drastic overload. Please take the time to experiment with the various objects and determine how these and other properties alter the appearance and function of the various objects.

Property Exercise/Group Intro
Course Schedule
Main Page