Lab 2a: Additional capabilities

Many of you have already completed the programming that was assigned last week. Those of you who have not yet finished last week’s work can continue with those problems, but will also be required to complete the work assigned here.

The script you have been working on for last week’s lab (when completed) will display a dialog box with a list of themes from a selected view. The user can select many themes from that list, and those themes will be made visible in the view. Themes that have not been selected will be made not visible in the view.

Once you have finished writing last week’s script, try this:

A – Run the script
B – When the dialog window appears, click cancel.

You will get an error message that says something like this:
A(n) Nil object does not recognize request FindByValue
Write down the exact message that you get:

Describe what you think this message means, and why you are getting it.

For this lab, you are asked to add the following capabilities to the script you are working on:

**Capability 1:**
Fix the problem that is pointed out above. This is called error handling. You will need to figure out a way to prevent your program from passing the user a message like the one above. Instead, you want to pass a message to the user that he or she can make sense out of. You also want to ensure that the script does not end unexpectedly.

*Hint:*
One way to approach this problem would be to search for the condition that causes this error (in this case, an improperly set variable). When searching for that condition, remember that the error occurs because you have canceled the dialog box! If that
condition exists, you can execute a message box telling the user what is wrong, and then gracefully end the script using the `exit` key word (look it up in ArcView help).

**Capability 2:**
Create a dialog box that allows the user to select one theme, and then set that theme active. Ensure that the same type of error handling techniques that you used in the previous capability also work with this capability.

_Hint:_
There are two steps here: selecting the theme, and then setting that theme active. Look up help on message boxes and find a request that allows you to select one choice from many. Remember that the message box will then return that choice, so you can then reference it with a variable. Once you have a variable referencing the theme, see if you can find a request (in Avenue help) that will set that theme to active. Also remember that you will want to set all other themes to not active.

**Capability 3:**
Call the script that allows the user to zoom to the selected theme.

_Hint:_
This script can be found by looking at the View GUI. Find the button that allows you to “Zoom to active Theme(s)” and then select the project window. Select the project: customize option from the pull down menu. This will open the following dialog box:

![Customize: lab2.apr](image-url)
Select “View” in the “Type” combo box (upper left-hand corner), and then select “Buttons” in the “Category” combo box. The Customize dialog box should now look like this:

![Customize dialog box](image)

In the graphical section displaying the buttons, move the scroll bar to the right until you see the button you are looking for. Select that button, and then look at the “Click” property (the topmost line in the bottom section of the dialog window). Copy the name of the script listed here (the script in the window above is “Project.Save”). This is the script that you will want to call from your script.

Additional help on using the “Run” request can be found in ArcView Help.