Geoff Canyon's Appeal to Authority

Navigator Documentation

TABLE OF CONTENTS

Table of Contents

INTRODUCTION

Navigator is a tool for working with projects in LiveCode (formerly Revolution). When LiveCode first released the ap cation browser many years ago, there were complaints that it was too large and too slow, and didn't do enough. So spent a weekend whipping up a prototype for Navigator, with the goal of being as simple, fast, and compact as pos ble. Over time the application browser improved, and other tools were released as well. Navigator is open source, let it go to the community about ten years ago.

Over the years, Navigator has held up surprisingly well. I still use it every day in my own development work. Recent heard from several developers that they still use Navigator as well, because it retains the simplicity and robustness started with. So I decided to fix a few outstanding issues and add some much-needed updates to bring it in line wit the curren LiveCode feature set.

Below is the documentation for Navigator in its present state. The documentation is in greater need of update than Navigator itself, so I'll update this page over time.

Table of Contents

Get Navigator

Navigator comes with LiveCode, but that version is horribly out of date.

The current version of navigator is available for download here. You can replace it by dropping the new copy in Li

Code's plugins folder. It is developed with LiveCode 8.

Older versions of LiveCode: Version 4.5.3, compatible with older (LC 5.5 and greater) versions of LiveCode, is avai able here.

Table of Contents

Navigator Videos

Navigator videos are available here. I'll add a list here soon.

Table of Contents

NAVIGATOR OVERVIEW

Navigator displays lists of Revolution objects: stacks, cards, groups, and controls.

The list can be of the controls on the current card of the topStack, or of any card or group of any stack. It can also b list of cards or backgrounds in a stack, or all open stacks.

Navigator lists everything. It works with controls that are:

- Invisible
- Offscreen
- On a different card
- In a cantModify stack
- In a palette
- Nested three layers deep in a group
- Part the of the LiveCode IDE

Navigator accesses anything anywhere (except password protected scripts of course).

Navigator allows you to **edit scripts**, **edit behavior scripts, set properties, set custom properties**, or **run code** **any list of objects**.

Save references to lists of objects in any combination to make it easier to work with them later.

Filter the list based on any text in the objects' names or IDs: filter them by name, by type, by enclosing group, etc.

Search objects' scripts: find all objects that reference a particular global variable, for example.

Search for objects meeting any test you define: find all objects with a height of 23 and a backgroundcolor of rec for example.

The list supports **drag and drop** to relayer controls, move them from one place to another, move them in and out groups, place backgrounds, and more.

Copy and **paste** objects using Navigator, without having to have them visible.

Perform many commands on the objects using Navigator: **place and remove groups**, **set images as windowshap and button icons**, all with simple commands.

Select a stack to work with or bookmark a control by pointing at it.

Navigator works with LiveCode, bringing up the properties palettes, the script editor, the alignment editor, etc.

Table of Contents

Navigator offers many built-in capabilities

- modify the colors of a set of objects
- edit all the properties of a set of objects in an easy-to-use list
- align and resize a set of objects

- edit the contents of buttons and fields (even if their lockText is true) and much more.

Table of Contents

Navigator allows you to do things you wouldn't ordinarily be able to do

- align controls that are on different cards
- place a group onto multiple cards with one command
- edit the properties of objects that are on different cards or different stacks
- and much more!

Table of Contents

Navigator makes you a faster developer

- edit the scripts of controls on different cards quickly and easily
- change any property with a single menu selection rather than waiting for a palette to display
- go to the handler you want to edit in a long script with a single command.

Table of Contents

CHANGE LOG

Table of Contents

UPDATES FOR 6.0 alpha 1

Group Folding

In Navigator's control list, you can now click in the left 30 pixels of the entry for any group to fold that group, which hides the controls within that group. Clicking again will unfold the group, showing its controls.

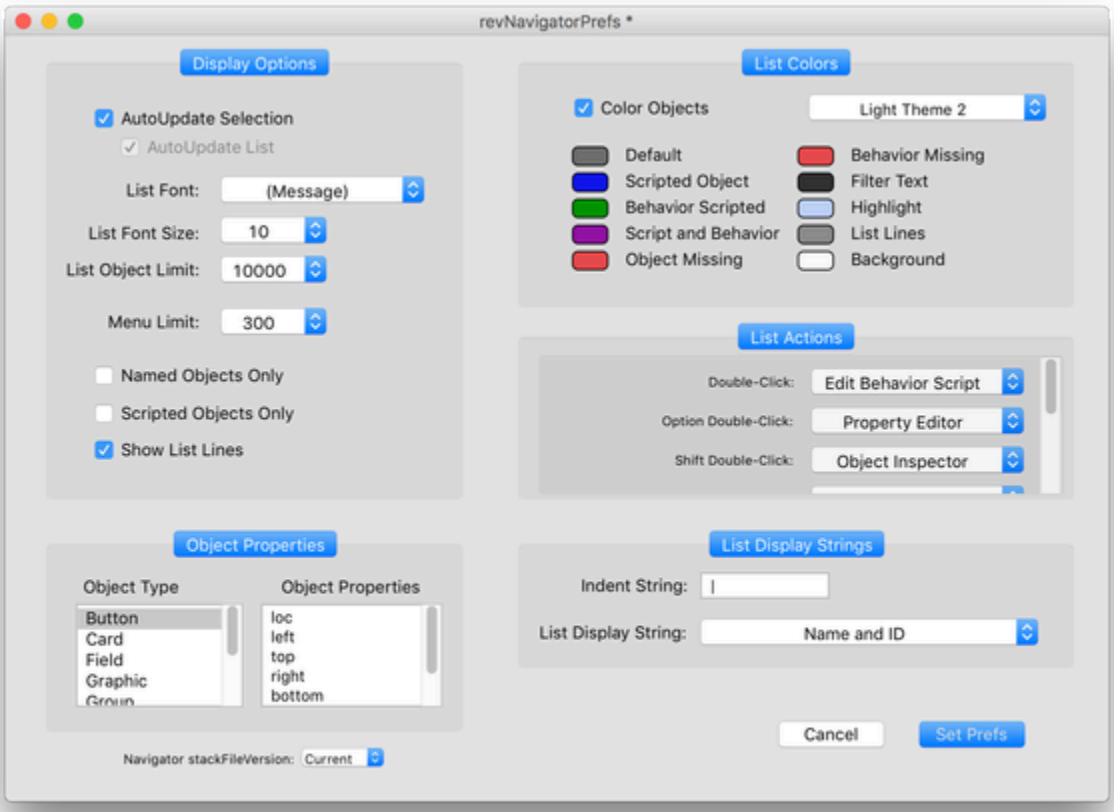
Currently there is no visual indication that folding is possible, or that it has been done. Let me know if you have suggestions.

Currently the folding settings are retained in a given Navigator window, meaning that if you are displaying controls from Stack “A” and have folded Group “X”, and switch to display Stack “B” and then later switch back to Stack “A”, th Group “X” will still be folded. Let me know if that’s a problem.

Folding settings are distinct between Navigator windows, meaning that if you are displaying controls from Stack “A’ two Navigator windows, each window has its own folding setup.

Added a Preferences Dialog

For now, the existing preference items on the Actions menu remain, but selecting “Prefs…” on the Actions menu opens this, which offers far more options:



Taking the items in order:

List Font — you can now select any font you like to use for the list in Navigator.

List Font Size — you can also select Other… and use any (reasonable) font size you like.

List Object Limit and **Menu Limit** — you can also select Other… and use any (reasonable) limit you like.

The rest of the Display Options are the same as they were/are in the menu.

The Object Properties are the same as they were/are in the menu, just easier to set.

List Colors — you can now save color sets in the menu. Note, this is currently stored in Navigator itself, so updatin to a new copy of Navigator will require re-creating your color sets, or copying the property and setting it after upda ing. Storing color sets in a preferences file is on the to-do list.

Fixed/Improved Double-Clicking — Double-clicking with the Command Key pressed has been fixed, so any comb

tion of Shift, Option, and Command can be designated to have a specific function. Also, fixed double-clicking so if y highlight any selection of items and then double-click on any item that is highlighted, it applies to the entire set of highlighted items rather than just the one you double-clicked.

Custom Double-Click Actions — You can now specify any command string to "do" for each control that is double-clicked. The string "tID" is replaced with the control's long id before "do"ing the string. As a simple example you can an action of:

set the vis of tID to not the vis of tID

And then double-clicking will show/hide the controls you double-clicked. **This is powerful:** anything you can "do" is allowed, so **be careful**, but I'm curious to see where this goes.

Also, let me know if the variables that are accessible in Navigator's Command Dialog: tCount for the number of con trols, tIndex for the index of the current control in the list of controls being manipulated, etc. would be useful for th

Custom List Display Strings — You can now specify any string to value() to determine what to display in the list. T string "tID" is replaced with the control's long id before value()ing the string. If the string "tIS" is included in the strin it is replaced with whatever the indent string is for the control. This allows you to include something before the ind string if you like. If "tIS" is not in the string, the indent string comes before everything else, as usual. As an example, the following string:

the layer of tID && tIS & "" & toUpper(char 1 of the name of tID) & ""&& Q(the short name of tID) && "loc:" & the loc of tID

Will display a list that looks like this:

23 | | **F** “Field Name” loc: 145,56

Only the first line of the result will be used, and it’s entirely possible to create strings that will error when value()ed. that happens, the entry will instead include the error string, but processing won’t halt, so the list will still display.

In the value() you can use any handler/function in the message stack. In particular, Navigator supports the followin

Q() — puts quotes around a string, so Q(“test”) is equivalent to quote & “test” & quote.

IFF() — functional if statement: iff(<boolean>,<true value>,<false value>) is equivalent to <true value> if <boolean> true, and <false value> if <boolean> is false. This is simpler than IFFV (see below), but requires both <true value> ar <false value> to be value()-able, since they get evaluated before being handed to the function.

IFFV — functional if statement with value: iffv(<boolean>,<true value>,<false value>) is equivalent to value(<true va ue>) if value(<boolean>) is true, and value(<false value>) if value(<boolean>) is false. This can be a bit more comple than IFF to set up, but it only evaluates <boolean> and the resulting output, ignoring the other value. This allows yc to do something like:

the name of tID & iffv(the behavior of tID is not empty,Q(” | “) && “&” && “the name of the behavior of” && the long of tID,””)

If button “input behavior” is the behavior of field “input field”, that will display a list that looks like this:

| field “input field” | button “input behavior”

Note that this isn’t possible using IFF(). If you tried using iff(the behavior of tID is not empty,” |” && the name of the behavior of tID,””), then for any controls that *don’t* have a behavior, the evaluation of “the name of the behavior of < control with no behavior>” would fail, and return an error string.

replaceF — functional replace statement: replaceF(<string to find>,<replacement string>,<string to replace in>) re- turns <string to replace in> with all instances of <string to find> replaced with <replacement string>. So for exampl

replaceF(“cat”,”dog”,”Outside of a cat, a book is man’s best friend. Inside of a cat, it’s too dark to read.”)

would return:

“Outside of a dog, a book is man’s best friend. Inside of a dog, it’s too dark to read.”

Updated Convert to Behaviors Dialog

Fixed a bug where controls in substacks would throw an error because the substacks have no stackfile and therefo look as if they haven’t been saved.

Added a dropdown to restore template settings from stacks for the controls in the list. Sets the behavior file name template, the space replacement character, and the behaviors folder.

Clarified Inspector Commands

When opening LiveCode inspectors for controls, Navigator can either: open a separate inspector for each control, c open a single inspector for all of the controls together. Previously, this was given unhelpful names in the name of brevity. Now the two commands are **Single Object Inspector** to open a single inspector for all of the controls toge er, and **Individual Object Inspectors** to open a separate inspector for each control.

Quick Property Setting

If you highlight a set of controls and then hold down the Command Key while selecting any property, that property will automatically be set to the value for the most-recently selected control. This does not (yet) work for custom pro erties. A more obvious user interface for this would be good.

Table of Contents

UPDATES FOR 5.2 – 5.6 alpha 1

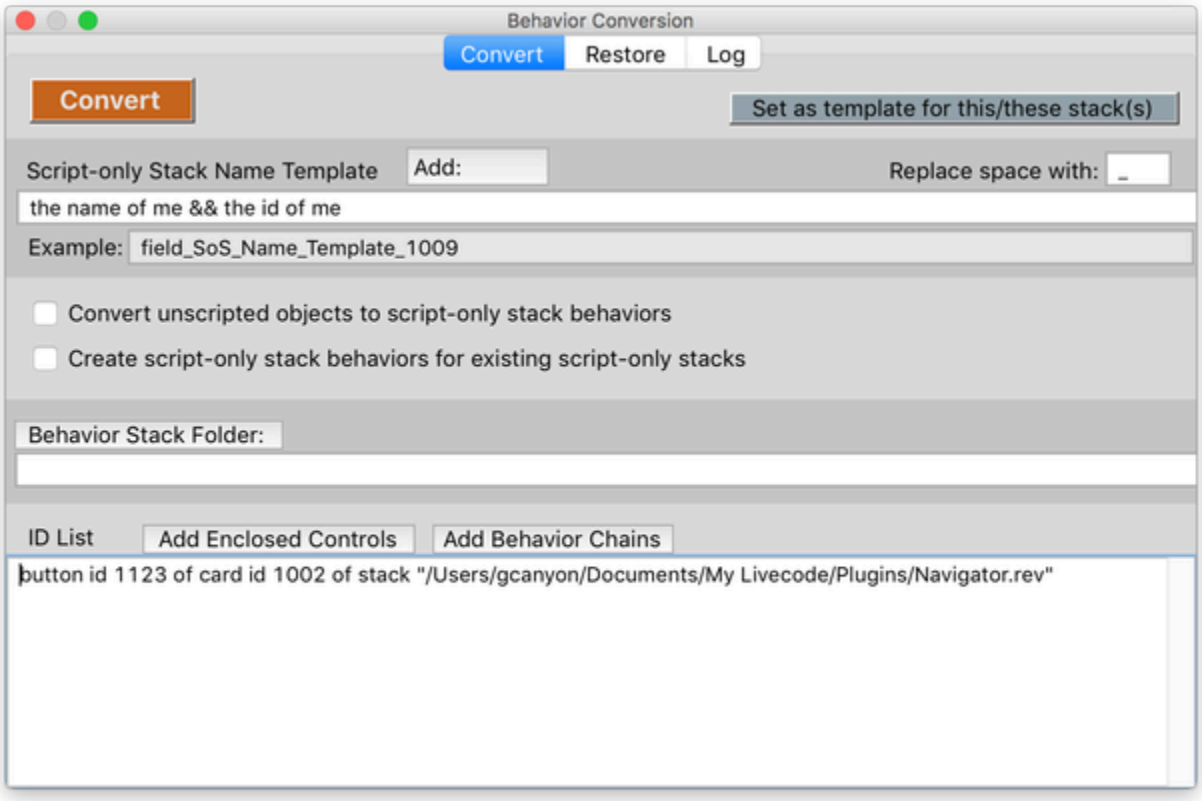
June 26, 2018

There's more, but here's a start:

Added "New Substack…" to the Substacks menu in the popup menu

Right-click a stack and select "New Substack…" on the Substacks menu on the popup menu. An Ask dialog allows yc to name the new substack. A new stack will be created with the name you provide, and set as a substack of the ma stack you right-clicked. The new substack will be the displayed in Navigator, ready for modification.

"Convert to Script-only Behaviors" Updated — Now in a dialog.
There are many updates to the conversion process, but the most fundamental is that everything has been moved t dialog:



This dramatically clarifies the process. At its most basic:

1. Select controls you want to convert in Navigator. This can be any combination of controls and bookmarks.
2. Select Convert to Behaviors… on the Script menu.
3. The dialog opens, and the long IDs of the list of controls you selected will be in the ID List field at the bottom

the dialog.

4. Click Convert, and for each control in the list:
    1. The control's script will be moved to a script-only stack.
    2. If the control already has a behavior, the script-only stack's behavior will be set to the control's behavior
    3. The control's behavior will be set to the script-only stack.

That's the basics, but there's much more to it, obviously. First, there are options. Starting from the top:

1. Spaces in the resulting script-only stack names are replaced by default. You decide what to replace them with The default is an underscore, but you can use any character(s), including a space, which will change nothing.
2. The template for the script-only stack names. The default is the name and id of the original control, but you c use anything that works with the Value() function. So if you really wanted to name the stacks for the width of source control, "the width of me" would do that. The example field shows you what the template you've creat will look like for the template field. It will highlight in red if the evaluation fails for the field. Note that the conv sion process has a built-in indexer, so if you have a name collision, it will be handled for you.
3. Whether to convert objects that have empty/no script. This is off by default. You might want to enable it if yo have defined a user interface but you haven't written any scripts yet.
4. Whether to convert existing script-only stacks. This is off by default. This seems like a bad idea, although it wi work and is there if you need it for some reason.
5. The folder to save the behavior stacks in.

Note that enclosed controls are not converted during the conversion process. If you select a card to convert, then controls on that card are not automatically converted. To add controls enclosed by the controls on the list, cli the **Add Enclosed Controls** button. Then all controls enclosed by the controls in the list will be added to the list.

Note that behavior chains are not followed during the conversion process. If you select a control that has no

script (and Convert Unscripted Objects is off), then nothing will happen, even if that control has an existing button havior. To convert the button behavior, you need to have the ID of the button in the ID List. You can easily do that b clicking the **Add Behavior Chains** button. This will add to the list all behaviors associated with the controls already the list. It will follow chains of any length. Once the behavior controls are in the list, then selecting Convert will conv them.

When you click Convert, you will have the option to perform a dry-run and see what will be done. This will convert nothing, but take you to the Log display and show the output with the current settings. If everything looks right, yo can switch back to the Convert display to perform the actual conversion.

UPDATES FOR 5.1 alpha 1
January 30, 2018 — alpha 1 notes

“Convert to Script-only Behaviors” Updated
There are many updates to the conversion code for this update. The most visible change is that the name schema t the converted stacks is now configurable. An Ask dialog lets you put in any valid LC expression, which will be evalua ed in the context of each conversion. So for example, the default:

```
the name of me && the id of me
```

will result in stacks named something like “button_start_process_1019”.

You can get creative: references to

```
the owner of me
```

will work, and

```
word -2 to -1 of the long id of me
```

will produce the stack name.

Note that simple names will work, because the code has a built-in serializer for names, so if you convert 20 control all named “bob” and choose to name the script-only files with

```
the short name of me
```

you’ll end up with stacks named bob, bob 2, bob 3, etc.

BUGFIX: the serializer would fail if a duplicate occurred of the very first control converted. This has been fixed.

“Restore from Behaviors”
The Script menu now has the command “Restore from Behaviors”, which moves scripts from behaviors back into th controls that have those behaviors, and removes the behaviors from those controls.

This is not an undo for Convert to Script-only Behaviors. If you Convert to Script-only Behaviors and then use th command, you may not end up with the same scripts in the same controls, even if the same controls were selected **You should ABSOLUTELY work on a copy with this command**.

This command should be “safe” in the sense that it will not delete any script that it has not copied to another contr or remove any behavior that needs to continue to exist. But it is up to you to ensure that you have selected all the controls that use a particular behavior before using the restore command. And if multiple controls use a single beh ior, they will (optionally, the command will ask you) all include the script of that behavior. Plus, chained behaviors w all be copied down the chain. So even if the command does the right thing, it may not be what you would want.

That said, I have used the command successfully to remove the behaviors from Navigator in order to reconvert it u ing the new naming schema function.

To get the behavior you expect, likely you will have to select all of the script-only stacks as well as the commands th use them as behaviors. To do this, select the stack list on the Stacks menu with the option-key down so that all stac are displayed. Then select the script-only stacks, right click on them and bookmark them. Then switch back to displ ing the stack with the controls that depend on the script-only stacks, and select both the bookmarked stacks and a the controls that depend on them. Then use “Restore From Behaviors.”

“Reload Script-only Stacks” on the contextual menu
Available on the contextual menu for stacks, this command is the opposite of the recently added “Save Script-only Stacks” command: it updates all of the script-only stacks from their files. This is designed for use with source contr where switching branches will update the files on disk, but LiveCode will not update automatically as a result.

Custom properties menu update
Holding down the option/alt key while selecting a custom property or custompropertyset will display a dialog askin you want to delete the custom property or custompropertyset.

BUGFIX: Display custom properties of all selected controls. Custom property menu was displaying only custom pro erties from the selected control, ignoring custom properties (in the chosen set) from other objects. Now the menu (and the Custom Property Editor) display all the custom properties in the chosen set from all the controls.

Custom Property Editor…
The property editor now supports custom properties. It displays a list of all the custom properties in the current cu tompropertyset for any of the selected controls. The default value provided will be the existing value of that prope for the last object you clicked. For example, if you click on one control and then shift-click another to select a range the value displayed will be for the control you shift-clicked. The name and id of the object, along with the number o other controls being edited, is displayed at the top of the list. The custom property editor always displays the text e try field for all properties. Either click OK or hold Option/Alt and press return to set the value, or Cancel to discard your change to the value. **Important**: when you click OK, all values are set immediately. There is no cancel or undo the Custom Property Editor.

(Custom) Property Editor Updates

The Property Editor displays inherited values in gray, and the customized properties and most recent properties at the top with the property labels bolded. This has been the case for several updates of Navigator, but it wasn't in the documentation before this.

The property editor compares values for the displayed properties, and shows the value in bold if the controls don't have the same value for that property. This can be particularly helpful for finding the differences when you have two controls that are similar but not identical.

If you are displaying properties for multiple controls and you click on the property name, the editor displays a pop-menu with the control names and their values for that property, with the default at the top. Selecting on the menu sets the property for all the controls.

Script Menu: Send: now supports private commands and functions

If you select a private command or function, Navigator will temporarily insert a helper command/function into the handler to pass along your call. That means Navigator can now easily test any code other than before and after behavior messages.

Stacks Menu: Script-only stacks are hidden by default.

As with other stacks, open the menu with the Option/Alt key pressed to see all stacks, including LiveCode IDE stacks and script-only stacks.

UPDATES FOR 5.0 alpha 3

January 25, 2018 — alpha 3 notes

Added a "Save StackFiles" menu item to the popup menu when right-clicking on a stack or multiple stacks. This is especially useful for stacks using script-only behaviors. With one command it saves all the stacks referenced in the stack's stackfiles property. **Note: this does not save the stack itself — use the existing "Save" menu item for that.**

The Scripts menu now supports unlimited chained behaviors. It will display the handlers in all chained behaviors, ir submenus under each behavior's name, with the object's own handlers, if any, at the bottom of the menu. The Sen menu is organized similarly, and can send any public on, command, function, getprop or setprop handlers.

January 25, 2018 — alpha 2 notes

Fixed a bug where if you selected to put the converted script-only stack files in the same folder as one of the stacks being converted, the code would loop endlessly.

Also changed the menu item to "Convert to Behaviors" because "Convert to Script-only Stack Behaviors" made the menu unwieldy

January 24, 2018

Converted Navigator to script-only stack behaviors. This will allow Navigator to be a GitHub project. Further, this wa done using Navigator's new Convert to Script-only Stack Behaviors command, available on the Script menu.

Convert to Script-only Stack Behaviors command

THIS FEATURE IS DANGEROUS. ONLY USE ON COPIES OF YOUR PROJECTS. I've tested this feature on multiple tes projects without incident, but it does change the scripts of every scripted object you use it on. You should definitely have backups in case something funny happens.

Highlight any controls, including bookmarked controls, and select Convert to Script-only Stack Behaviors on the Scr menu. Navigator will provide stern warnings that you should only perform this action on backed-up projects (or copies of projects). The command has the following features:

1. Works on any selection of controls, including bookmarked controls, so you can select controls on multiple car or in multiple stacks.
2. Moves the scripts of all selected objects into new script-only stacks.
3. Sets the behavior of the objects to the new script-only stacks that contain the objects' scripts.
4. If any other selected objects have the object as their behavior, their behaviors will be updated to point to the new script-only stack.
5. You select the folder to create script-only stacks in.
6. Sets the stackfile properties of stacks to include the newly-created stacks, with relative paths based on the cu rent location of the stack and the selected script-only stack folder.
7. Deletes the objects' scripts.
8. If an object with a script already has a behavior, Navigator sets the behavior of the new script-only stack to th behavior.
9. Optionally, if objects with no scripts are selected, will create new script-only stacks for those as well.
10. Generates a log file documenting everything it does.

11. If any new script-only stacks have behaviors themselves, the log file will include code that will set those behav iors. You'll need to include that code in your startup routines for the script-only stacks to continue to have tho behaviors.

That's a lot to take in, so here are some use-case examples:

1. You have a bunch of controls with scripts, and you want to convert them to script-only stack behaviors. The c trols can be anything you can select in Navigator, including bookmarks of controls in multiple stacks. Select th controls, select Convert to Script-only Stack Behaviors on the Script menu, and choose a folder to create the stacks in. Navigator performs the conversion, sets the behaviors, and updates the stackfiles property(ies) of t stack(s) to make everything work properly.
2. You have a new project interface and you want to use script-only stacks for everything from the start. Select t controls in Navigator that you want to work with and select Convert to Script-only Stack Behaviors on the Scri menu. Navigator will ask if you want to convert empty scripts to script-only stack behaviors. Select Convert Them, and Navigator will create script-only stacks as behaviors for all the controls, and set the stackfiles prop ty to make everything work.
3. You have a bunch of controls with buttons as behaviors, and you want to switch to script-only stacks. (This wa the status of Navigator itself). Select the controls *and* the behavior buttons — this required the use of book marks in Navigator — and select Convert to Script-only Stack Behaviors on the Script menu. Navigator will ask you want to convert empty scripts to script-only stack behaviors — select Skip Them. Navigator will convert al the behavior buttons to script-only stacks and set the stackfiles property to make everything work. Navigator update all the selected controls that refer to the buttons for their behaviors to point to the new script-only stacks. The buttons will also have the script-only stacks as their behaviors, so if you missed any controls that r fer to the buttons, they will continue to work, as long as you don't delete the buttons.
4. You have controls that have scripts and behaviors, and you want to move everything to script-only stack beha iors. Select the controls and select Convert to Script-only Stack Behaviors on the Script menu. Navigator will c

ate new script-only stacks for the scripted objects and set the stackfiles property to make everything work. Th objects will have the script-only stacks as their behaviors, and the script-only stacks will have the objects' orig behaviors as their behaviors. Navigator's log for the update will include lines of script that set the script-only stacks' behaviors; you will need to run that code whenever you first open your project for the script-only stacl to have the correct behavior.

You can download Navigator 5 here. You'll need to decompress the archive, and then put both Navigator and the N igator_Behaviors folder in your My Livecode/Plugins folder. If you have any questions about the new function, feel free to ask on the LiveCode Mailing List.

Table of Contents

UPDATES FOR 4.5.4

January 7, 2018

Fixed a bug where selecting objects of multiple types would cause the Property Editor to fail to open, because the add-on properties for each type weren't necessarily common to all the types. Navigator now properly checks the ac ons against representatives of each type to ensure that all add-ons from any type that apply to all types are displa

Table of Contents

UPDATES FOR 4.5.3

January 7, 2018

Unless a bug makes it necessary, this will be the last update to Navigator in LC5.5 format. The compatible version is available here.

From this point on, I'm developing Navigator in LiveCode 8. Going forward, updates will be to the current version o unless something comes up that really needs to be fixed.

In the script menu, added support for "private command" and "private function" handlers.

Fixed an issue with the script menu not properly scrolling scripts to the selected handler. In LC 6.7.3, line 1 of the openstacks was reliably the script editor when an object's script was edited. This seems not to be the case in LC 8. I changed the code to parse through the openstacks until it finds a script editor window, and then go with that. It's a

hacky, but it seems to work.

Table of Contents

UPDATES FOR 4.5.2

January 6, 2018

From this point on, Navigator is being updated for LiveCode 8. I'll post a link to a version that works back to 5.5 tom row, but going forward updates will be to the current version, unless something comes up that really needs to be fixed.

I just checked, and stacks used as behaviors work fine, yay!

Modified the Handlers menu to support the use of "command" as a synonym of "on". This synonym has been in the engine since the beginning, but (amazingly) I had never come across it. So now Navigator recognizes "command" ha dlers, and can send messages to them.

Added small icons to the right side of Navigator's title bar that collapse and open Navigator, and the same for all in stances of Navigator. This functionality was previously (and is still) available by double-clicking Navigator's title bar, but I wanted to make sure it was surfaced to the user.

Fixed the icon for the Stack menu. There was always something wonky about the icon setting of that button. To fix had created an equally wonky stack image to use as an icon. I took the time to recreate the button, and replaced th icon.

I re-enabled the option to “paste” controls after “copying” them. I don’t know why I had commented out that code, it seems to work. Note that Navigator tries to copy using standard LC commands, but also keeps a list of the object “copied”. The “Paste Objects” command goes off this list, so you can paste directly into a stack, card, or group, even they are not currently in front.

I changed the font of Navigator’s list to be one more compatible with the horizontal rule provided by LiveCode fielc

TBD: In the script menu, add support for “private command” and “private function”

TBD: In the “dark” color scheme, some buttons are currently obscured. I’m going to have to figure out what to do about that, since the colors are configurable, and I don’t want to have to code something that interprets a reasonal color for the buttons based on whatever the user chooses for the color scheme.

TBD: The opening of locked property inspectors is broken in LC 8. It calls a handler that may not exist in the curren dev environment. I’m going to have to figure out what happened with that.

Table of Contents

UPDATES FOR 4.5.1

January 4, 2018

Modified the Handlers menu to include sending messages in a new submenu, to make it clearer. (Previously this w accomplished by holding down the option/alt key while selecting a handler name. This is no longer available)

Modified the Handlers menu to fix calls to no-argument handlers, like “on mouseUp”

Modified the Handlers menu to use “dispatch” instead of “send” for messages. This makes messages a little harder send if they have many arguments (and limits the arguments to 9) but makes it possible to arguments with quotes, commas, and returns. TBD: using a custom dialog to simplify this.

Modified the Handlers menu to support calling functions using “dispatch.” The limit for arguments is 9. The functio return value is displayed, but if it’s an array, that will fail. TBD: unroll arrays before display.

Modified the Handlers menu to support calling setProp and getProp. getProp's return value is displayed, but if it's a array, that will fail TBD: unroll arrays before display.

Modified the Handlers menu to fix the editing of the behavior scripts. The code to jump to the handler name in behavior scripts was broken.

Modified the Property Editor... in the Properties menu to have a small header that includes the name and id of the source object for the property values and a count of the additional objects being modified.

Table of Contents

UPDATES FOR 4.5

All of Navigator's code has been moved to behaviors. The "New Navigator" command on the Actions menu now works, and works a treat. You can have as many copies of Navigator open as you want/need, each viewing its own of controls.

Navigator understands behaviors. If you double-click a control with a behavior, Navigator will open the script edito for the behavior. If the object has its own script, option-double-click to open the object's script. Or use the popup menu to access either.

Table of Contents

UPDATES FOR 4.0 RC1

"Clone Navigator" has been disabled. I never heard from anyone who was using it, it was slightly buggy, and it was built before behaviors, so it wholesale copied everything, making Navigator 3x as large. Changing everything over t behaviors, and thus allowing unlimited cloning of Navigator, is on some theoretical to-do list. Let me know if this is more important than I realize.

Drag and Drop has mostly been fixed. The original implementation was based on the built-in messages in Livecode which was challenging and didn't survive the intervening versions. The new implementation uses a highly modified version of Scott Rossi's example code. The new implementation does not support dragging anything out of or into Navigator, but does work better to support shift- and command- selections in the list. Most things supported by dr and drop work: rearranging layers, moving things into and out of groups, rearranging cards in a stack, copying con trols to new cards/stacks, placing groups onto multiple cards.

Navigator is not under active development, but I still use it all the time, and I've made minor modifications over the years. If you find any bugs or issues, or have feature suggestions, feel free to email me at gcanyon@gmail.com.

Table of Contents

NAVIGATOR FEATURES IN DETAIL

Table of Contents

Small and Resizeable

- List the controls in the current card, or any card or background in any stack. The list can be indented to show group control hierarchy.
- Quick access to a list of the cards or backgrounds/groups in any stack.
- List all open stacks and substacks.
- From any card, group, or stack reference, jump directly to a list of the controls in that card, group, or stack.

Edit scripts, going directly to any handler in a script without having to edit the script first.

Edit properties of controls **using LiveCode palettes**; either individually or in groups.

Edit properties of controls **using a simple scrolling list**.

Table of Contents

Actions on controls

- **rename** controls.
- **clone** controls.
- **delete** controls.
- **group** controls.
- **Edit the colors** of a set of controls.
- **Edit the alignment** of a set of controls.
- **Call any handler in any script**, with a template for the arguments.

Count the length in lines and characters of an object's script, or of all objects contained within the object's hierarch group and all of its controls, or even everything in a stack).

When viewing "this card" of "the topStack":

- Selecting controls in the IDE is immediately reflected in Navigator.
- Selecting controls in Navigator's list selects them in the IDE in real time.

Select controls in the list and have Navigator select the actual controls in the LiveCode environment, regardless of whether you are in edit mode, or if the controls are on the current card, even if the controls are on different cards in different stacks.

Display controls with scripts in a different color. (actions menu > prefs).

Filter for named controls (actions menu > prefs).

Filter for controls with scripts (actions menu > prefs).

See short names, long names, short ids or long ids (actions menu > prefs).

Filter controls for arbitrary text by entering it in the filter box at upper right. Type “fie” to get all fields, for example

Table of Contents

Copy and Paste

- **Copy** a list of references to the selected controls; either the long id, or whatever is currently displayed. Also p the list into the Message Box.
- **Copy the selected controls**. This copies the actual controls, which allows pasting copies somewhere else.
- **Paste the copied controls into any set of cards, stacks, or groups**.

Set the **windowShape** of a stack.

Set the **icon** of a button.

Table of Contents

Drag and Drop

- **Relayer controls** by dragging them.
- **Move controls into or out of groups by dragging them**.

Table of Contents

Navigation/Stack Commands

- **Go directly to any card of any stack**. Also go to any bookmarked stack, open or not.
- **TopLevel any card** of any stack.
- **Save any stack** directly.
- **Show a stack**, forcing its visible to true and centering it on the display.
- **Purge any stack** from memory directly.

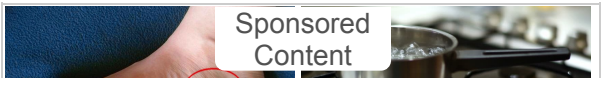


Table of Contents

Edit Properties

- **Easily change standard boolean properties** for all selected controls, cards, backgrounds, or stacks.
- **Easily change other standard properties** for all selected controls, cards, backgrounds, or stacks.
- Set any list of properties (standard or custom) for all selected controls, cards, backgrounds, or stacks.
- **Run any code against all selected controls**, cards, backgrounds, or stacks.
- **Save any code** you run as a Navigator plugin for future use on any controls.
- Navigator plugins can offer the user a choice on one or two popups, or a listbox.

Table of Contents

CONTROL LIST

Controls are displayed in a simple list. Bookmarked controls are shown at the top of the list, indented. Below the bookmarked controls is the stack reference in bold. Below the stack reference is either: a single card or backgroun also in bold, and a list of all the controls on the card/background; or a list of cards or backgrounds in the stack. Alte natively, below the bookmarks is a list of stacks and substacks. If displaying a list of controls, the controls are inder to reflect enclosure within groups on the card. The indent is configurable on the Actions > Prefs menu.

Table of Contents

Selection

- **Click** an object to select it.
- **Shift-click** another object to select a range from the one you clicked to the one you shift-clicked.

- **Command-click** to toggle the selection of an object.
- **Click and drag** to select a range of controls.

Table of Contents

Contextual Menus

- **Control-click** to pop a contextual menu with various commands. See Contextual Menu below for details.
- **Control-Option-click** to pop a contextual menu of standard properties. You can set them from this menu se Properties Menu below for details.
- **Control-Option-Command-click** to pop a contextual menu of custom properties. You can set them from this menu see Custom Properties Menu below for details.
- **Control-Command-click** to pop a contextual menu of the handlers in the object's script. You can only do this with a single control highlighted. Select any handler in the menu to edit the object's script and go directly to t handler. Option-select any handler to call that handler in the object's script. See Handlers Menu below for details.

Table of Contents

CONTROL LIST DRAG AND DROP

The list supports drag and drop for a wide variety of commands. All of them involve the simple action of selecting o or more items in the list, clicking on one of the selected items, and dragging to somewhere else. The context allows for many different results.

Listed below are the various drag and drop contexts, and the results. Generally, you can mix and match the contex and Navigator will do the right thing for each item you drag. An example of this is that if you select bookmarks and controls, and drag to another place on the card, Navigator will copy the bookmarks to the current card and rearrar their layers with the controls you dragged, as appropriate.

ANYTIME

Drag bookmark items within the bookmark list to rearrange the list. This has no effect on the bookmarked objects.

Drag references to objects into the list from elsewhere to create bookmarks to the objects. There is no error check on this: if the objects can't be found, Navigator will still make bookmarks.

WHEN DISPLAYING THE CONTROLS ON A CARD OR GROUP

Drag controls in the control list to relayer them.

Drag controls in the control list into a group to move the controls into the group. You must be holding down the op tion/alt key to do this; otherwise the control will be layered just above or below the group. Note that there is no vis distinction based on this Objects dragged into a group will simply be layered above or below it if the option/alt key isn't down.

Drag controls in a group out of the group to remove the controls from the group. You must be holding down the o tion/alt key to do this. Otherwise the control will be layered at the bottom or top of the group. **NOTE:** if the control field that has different text on different cards, this will remove the field from all cards but the current one and the t of the field on those cards to be lost.

Drag controls in the control list into the bookmark area of the list to create bookmarks for the selected controls. If bookmarks already exist for a control, no duplicate will be created. This depends on the name of the bookmark, nc the underlying ID, so if you have renamed bookmarks, duplicates can be created.

Drag bookmarks of controls (not groups, cards, or stacks; see below for details on them) into the control list to cop

the controls to the current card/group. **NOTE**: if the bookmarks represent controls on the current card, they will sti be copied to the card, duplicating the controls.

Drag bookmarks of cards into the control list to copy the cards to the current stack. Has no effect if the card is alre in the current stack.

If a bookmark is of a group in another stack, drag the bookmarks of groups into the control list to copy the group t the current card.

If a bookmark is of a group (background) in the current stack, drag the bookmarks of the groups into the control lis to place the existing backgrounds onto the current card.

Drag bookmarks of stacks into the control list to set the mainstack of the bookmarked stacks to the current stack. T makes the dragged stacks substacks of the current stack. Note that you can easily undo this by switching the displa to the substack and dragging the bookmark into the display again. This will make the substack its own mainstack, making it a mainstack again.

WHEN DISPLAYING THE CARDS IN A STACK

Drag the cards in the list to set their layer in the stack. This changes the order of the cards in the stack.

Drag cards into the bookmark area to create bookmarks to them.

This one's a bit tricky and perhaps counterintuitive: select one or more cards, and a single bookmark to a group. Cl and drag on the bookmark of the group. Drag it anywhere in the card list. This will place the group onto all of the cards you selected in a single command. If the bookmarked group was in another stack, it will first be copied to the current stack, and then placed on all the cards. **NOTE:** with version 3, this can be done more intuitively by copying t group, then selecting the cards and using the contextual menu. See the documentation on the contextual menu fo more information.

WHEN DISPLAYING THE GROUPS (BACKGROUNDS) IN A STACK, OR THE STACK LIST

Drag groups or stacks into the bookmark area to create bookmarks to them.

Table of Contents

FILTER BOX

Text typed in the Filter box at the upper right of Navigator acts as a filter on the list. Type “button” to see only buttc “field” to see only fields, “bob” to see only controls with “bob” in their name, etc. Any text will work, so you can see only objects named “xyz” if you want. The filter acts on whatever is displayed, so you can filter on object names if names are being displayed, or on object IDs if IDs are being displayed. Indentation based on groups goes away if th group objects themselves are filtered.

The selected objects in the list remain constant throughout filtering and de-filtering of the list. This is especially har if you are looking for particular controls: filter the list, highlight the objects, and then de-filter the list. The objects w now be highlighted in the list of all controls.

The filter has no effect when displaying the stack list.

Table of Contents

BOOKMARKS

Save references to controls/cards/groups/stacks permanently by bookmarking them.

What is a bookmark? A bookmark is a static reference to a given object. It can reference any control, card, group,

stack. Bookmarked objects don't need to be currently displayed, or even in a stack that is opened (you can't work w them if they're not open, but the reference stays intact as long as the stack file isn't moved or renamed). Bookmark controls don't need to be from the same card, or the same stack. Bookmarked controls can be used much like regu items in the list: edit their scripts or properties, save them (if they are stack references) etc.

Save bookmark sets to quickly switch between projects.

Bookmarks and bookmark sets can contain any controls controls from multiple cards, stacks, backgrounds, etc. Als any cards, groups/backgrounds, or stacks.

Anything that can be done to a live reference can also be done to a bookmark: delete the object, rename it, se properties on it, edit its script, etc.

Color bookmarks any way you like to help organize them.

Table of Contents

ACTIONS MENU

About Navigator... Displays this information about Navigator, and help. Click the close button, or anywhere in the about/help display, to make it go away.

Prefs Submenu with options for Navigator:

Use Names If checked, the list shows object names. Otherwise it shows IDs.

Color Scripted Objects If checked, objects in the list with scripts are shown in a different color (set on the prefs > c ors submenu see below). Slows list display slightly, so if you are on a slower machine or working with very long lists and want to maximize performance, uncheck this.

Named Objects Only If checked, the list shows only objects that have been given names. Excellent for filtering out bel fields, etc.

Scripted Objects Only If checked, the list shows only objects whose scripts are not empty. Works whether or not scripted objects are colored separately.

Use Long Name or ID If checked, shows the long name or id. Otherwise, the short name or abbreviated id.

Show Tooltip If checked, holding the pointer over an item in the list displays a tooltip with the long id of the item. Very annoying. 😉

AutoUpdate List If checked, the control list is updated whenever an object is created or deleted, whenever the cur rent card changes, and whenever the selection changes. If it is unchecked, you must manually update the list see **U date List Now** below. Note that it is possible for the list to be out of sync even if this is checked. The list is updated response to messages from the development environment, and there are some changes that will not trigger these messages. One example is if you change the name of a control using the Properties Contextual Menu. Many other things trigger a potential list update, so for example switching stacks should cause the list to update.

AutoUpdate Selection If checked, the selection of objects will be updated in Revolution as entries are highlighted the list in Navigator. This only takes place if Navigator is displaying “this card” of “the topstack.”

Indent String... Navigator indents controls in the list based on the groups they are in. You can have Navigator use any string to do this. Use an empty string to prevent any indentation. The default “| ” provides for indentation and vertical lines showing the beginning and ending of groups. Three spaces makes for nice indentation with no vertica lines.

Add Properties... Navigator displays a list of properties for the selected objects when you Control-Option click the see **Contextual Menu (Properties)** below. The default list of properties to display does not include every possible property.

You can add property names to the list of properties to display. The properties to add are defined separately for ea type of control. The property names to add are stored as custom properties of the Navigator stack.

Selecting this command opens the Command Panel, with the current values of the custom properties displayed. Th
list will look something like this:

```
Card=
Image=loc
left
top
right
bottom
Button=loc
left
top
right
bottom
Group=backgroundBehavior
layer
loc
Graphic=loc
left
top
right
bottom
fillfore
Scrollbar=loc
left
top
right
bottom
Field=loc
left
top
right
bottom
tProp=
Player=loc
left
top
```

```
right
bottom
Stack=loc
left
top
right
bottom
```


Note how for Buttons the properties “loc, left, top, right, and bottom” have been added in the example above. Simp enter the names of any other properties you want to add to the popup, one per line. The properties do not have to in any particular order they’ll be alphabetized when the popup is displayed.

When you are done, click the “Set” button to set the properties and save your changes, or “Close” to cancel the process and ignore your changes. Changes you make are immediate.

Colors Submenu with color options for Navigator. Displays the color picker to select a color. The color options are:

- **Default…** The color objects will display in by default. (blue by default)
- **Scripted Object…** If you check Color Scripted Objects, the color objects will display in if they have a script. (da green by default)
- **Object Missing…** The color objects will display in if Navigator looks for them (to edit their scripts or properties and can’t find them. (red by default)

Menu Limit Sets the maximum number of items to display in the Stack and Card menus. You might want to use th you are working with an extremely large number of stacks, or a stack with thousands of cards, or cards or groups v

a large number of controls. **NOTE:** if you cannot display all the cards in a stack because of the menu limit, you can s lect **Card List** to put a list of the cards in the stack into the control list. That list is subject to the **Object Limit**, whicl might be set to a larger value.

Object Limit Sets the maximum number of items to display in the control list. You might want to use this is if you a working with a card with an extremely large number of controls.

Table of Contents

Double-Click Options

Each of these determines what is done when an item is double-clicked in the list. You can set the action for:

- **Double-Click**
- **Option-Double-Click**
- **Command-Double-Click**
- **Option-Command-Double-Click**

The five possible actions are:

- **Copy ID** Copies the long id of the double-clicked object to the clipboard. You can then paste it anywhere to th message box, for example.
- **Edit Script** Edits the script of the double-clicked object. This is the default for option/alt-double-click.
- **Edit Behavior Script** Edits the script of the double-clicked object. This is the default for double-click.
- **Property Editor** Displays a list of all properties of the objects. Click any item in the list to edit the value for th property. This is the default for command/control-double-click.
- **Object Inspector** Displays the LiveCode object inspector palette for the double-clicked object. This is the defa for option/alt-command/control-double-click
- **Bookmark-Remove** If the double-clicked object is a bookmark, removes it; otherwise bookmarks it.

New Navigator Creates a copy of the Navigator in which you select the command. You can open as many Navigato palettes as you like/need. Each copy can show a different list and different bookmarks.

Update List Now Updates the list of displayed objects. If **AutoUpdate List** isn't checked, select **Update List Now** t update the list. Also, there are a few circumstances where a change happens and it isn't reflected in the list.

Find in Scripts... Displays a dialog for you to enter text. **Searches all items in the list, including bookmarks, to s**

if their script contains the text. Items that match are bookmarked. **Bookmarks that match are removed and r bookmarked**, placing them at the top of the bookmark list.

Find by Test... Displays a dialog for you to enter a boolean test. Enter any test, using **tID** to refer to the control bei tested. For example, put "the height of tID is 23," to find objects 23 pixels tall. Put "the backgroundcolor of tID is 255,0,0 and the vis of tID is true," to find visible red objects. **Searches all items in the list, including bookmarks.** Matching items are bookmarked. **Bookmarks that match are removed and re-bookmarked**, placing them at the top of the bookmark list.

Rename Displays an answer dialog for each highlighted object in turn. Whatever you type will be the name of the object.

Delete Presents a dialog to confirm, then deletes highlighted objects. Hold down the alt/option key to delete witho a confirmation dialog. Note that this is not undo-able.

Clone Clones the highlighted objects.

Edit Properties Opens a new properties window for each of the selected objects. Windows are arranged, and lock to the objects they are opened for. NOTE: this will work even for controls that are not currently displayed. You can amine the properties for a control in any stack by selecting the stack on the stack menu, selecting the control you want, and selecting Edit Properties. This feature works only with Revolution.

Edit Scripts Edits the script of each of the selected objects. Windows are arranged. Like Edit Properties, the contro does not need to be displayed. You can edit the script of any control in any open stack.

Show Properties Opens one property window for all the selected objects. In Revolution, only properties all the ob-

jects have in common are active. In MetaCard, if more than one control is selected the alignment palette will be displayed.

NOTE: this works on the selected controls in the LiveCode environment you can only use the Show Properties com mand for controls in “this card” of the topStack. In LiveCode, you can use Edit Properties to edit the properties for controls that are not currently displayed. You can always use Navigator’s tools: the Properties menu, the Custom Properties menu, the Properties list, and the Set command in the Command Panel, to change any object’s properti

Command… Displays the **Command Panel**. See the section on the Command Panel below. Note that this disables the control list until you close the Command Panel.

Bookmark>Selected Lines Bookmarks all entries highlighted in Navigator’s list copies full references to them to th top of the list.

Bookmark>Selected Objects Bookmarks whatever objects are currently selected in Revolution.

Bookmark>MouseControl… Hold down the **option/alt key** while selecting this menu item. Then move the pointe over whatever control you would like to bookmark. Release the option/alt key, and the control under the pointer w be bookmarked.

Remove Removes selected bookmarked objects.

Remove All Removes all bookmarked objects.

Save Control Set… Saves currently bookmarked objects to a named set. The named set is displayed at the bottom the Actions menu. This can be very handy if you are working on multiple tasks: save a control set that is relevant to each task you are working on, and open the control sets as you switch tasks. Control sets persist across launches o Navigator; they stay until you delete them.

ControlSets this is where Control Sets are listed when they are created. You can have as many as your menu syste will allow. Option-select one to delete it.

Table of Contents

CONTEXTUAL POPUP MENU

If you open a contextual menu by clicking on an item that isn’t currently selected, that item will be selected and the

contextual menu will apply only to that item. To apply a contextual menu to a selection of items, open the contextu menu by clicking on one of the selected items. **Control-click** the list (**right click** on PCs and Unix) to activate.

Control-Option-click to open the **Properties** menu as a contextual menu. This is identical to the Properties menu the top of Navigator (documented below).

Control-Option-Command-click to open the **Custom Properties** menu as a contextual menu. This is identical to t Custom Properties menu at the top of Navigator (documented below).

Control-Command-click to open the **Handlers** menu as a contextual menu. This is identical to the Handlers menu the top of Navigator (documented below).

The contextual menu displays different items depending on what type of objects you have highlighted. For ample, if you highlight a card object (or bookmark) the contextual menu will contain items to “Go” to the card, to “T Level” the card, to “Browse Controls” of the card, etc. The object-specific items are:

Table of Contents

STACKS

- **Save —** Saves the referenced stack to disk. **NOTE:** this will not save a stack that has not been previously save you create a new stack and try to save it for the first time using this command, nothing will happen.
- **Purge Stack… —** Purges the stack and any associated mainstacks/substacks completely from memory. Any changes since the last save are lost. Offers a confirmation dialog. If the stack is not available, turns the refere to the error color.
- **MainStack/SubStacks —** If the stack is a mainstack, displays the substacks if any of the mainstack. If the stac

is a substack, displays the mainstack of the substack. In either case, browses the controls in the stack chosen.

- **Substacks > New Substack... —** An Ask dialog allows you to name the new substack. A new stack will be crea ed with the name you provide, and set as a substack of the mainstack you right-clicked. The new substack wil displayed in Navigator, ready for modification.
- **Set Copied Stack as MainStack — Available if a Stack has been copied —** Sets the copied stack as the Mai Stack for the highlighted stacks. Note that the stack must be copied by itself, and it must be copied using Nav tor's **Copy>Objects** command.
- **Set Copied Image as WindowShape — Available if an Image has been copied —** Sets the copied image as the WindowShape for the highlighted stacks. Note that the image must be copied by itself, and it must be cop using Navigator's **Copy>Objects** command.

Table of Contents

STACKS AND CARDS

- **Go>Go —** Goes to the card or stack referenced. If the card/stack is not available, turns the reference to the er color.
- **Go>TopLevel —** Goes to the card or stack referenced using the TopLevel command. This makes palettes and other usually not-editable stacks editable. If the card/stack is not available, turns the reference to the error color.
- **Go>LM Go —** The same as the Go command, but locks messages before going to the card or stack
- **Go>LM TopLevel —** The same as the Toplevel command, but locks messages before going to the card or stac
- **Go>Show —** Forces the card or stack to be visible by setting its visible to true and centering it on screen.

Table of Contents

STACKS, CARDS, AND GROUPS

- **Paste Objects —** Pastes the copied objects into the container. You can select multiple containers to paste int all at once. Note that this only works with objects copied within Navigator. Paste is intelligent. If the object to pasted is a group and it's being pasted into multiple cards in a stack, it will be first copied to the target stack, a then placed onto the selected cards.
- **Browse Controls —** Displays the controls that are part of the referenced stack/card/group. For stacks, displa the controls for the current card of that stack.

Table of Contents

CARDS

- **Place Backgrounds — Available if Backgrounds have been copied —** Places the copied backgrounds onto the highlighted cards. If any of the target card(s) are in a different stack than the background, the backgrounc first copied to the stack containing the target card(s) and then placed on each target card.

Table of Contents

GROUPS

- **Remove Group Available for Groups** removes whatever groups are highlighted from the current card. Does not delete the background unless it is being removed from the last card in the stack on which it appears.

Table of Contents

IMAGES

- **Set as WindowShape —** Sets the image as the WindowShape for the stack that contains it.

Table of Contents

BUTTONS

- **Set Copied Image as Icon — Available if an Image has been copied —** Sets the copied image as the Icon fc the highlighted buttons. Note that the image must be copied by itself, and it must be copied using Navigator's **Copy>Objects** command.

Table of Contents

BOOKMARKS

- **Remove** Removes selected bookmarked object(s).
- **Remove All** Removes all bookmarked objects. **NOTE:** this command removes all bookmarks, not just the currently selected bookmarks.
- **Color...** Displays an ask dialog allowing you to enter any color you like (by name or triplet) and colors the sele ed bookmarked lines that color.
- **Bookmark Text...** Displays a dialog allowing you to enter any name you like and sets the text of the bookmarked lines to that text.

Table of Contents

GENERAL

Commands available on the contextual menu regardless of the selected controls are:

- **Select Objects** Selects the highlighted objects in Revolution.
- **Copy>Objects** Copies the currently highlighted objects to the clipboard. This means you can go to any card o group in Revolution and paste the objects using Revolution's Paste command. Also copies the references inte nally to Navigator. This means you can paste the objects using Navigator's commands, and perform numerou other actions with them as well.
- **Copy>Names** Copies the currently selected list items to the clipboard. Copies names if names are displayed, IDs if IDs are displayed. You can then paste anywhere to the message box, for example.
- **Copy>IDs** Copies the Long IDs of the selected objects to the clipboard. Long ID references are guaranteed unique, and will therefore always work.
- **Copy>Text** Copies the actual text highlighted, whatever it is.
- **Put** Puts the currently selected list items (into the Message Box).
- **Rename...** For each highlighted object in the list, displays a dialog asking for the new name for the object. Yo can cancel on any object and stop the process.
- **Delete** Deletes the highlighted objects in the list. It first presents a warning dialog. You can avoid the confirm tion dialog by holding down the option/alt key while you select this menu item.
- **Clone** Clones the highlighted objects in the list.
- **Group** Groups the highlighted objects in the list.
- **Find>In Scripts...** Displays a dialog for you to enter text. Searches highlighted items in the list, including book marks, to see if their script contains the text. Items that match are bookmarked. Bookmarks that match are r

moved and re-bookmarked, placing them at the top of the bookmark list.

- **Find>In Enclosed Scripts…** Displays a dialog for you to enter text. Searches highlighted items in the list, inclu ing bookmarks, to see if their script contains the text. Items that match are bookmarked.
- **NOTE**: Bookmarks that match are **not** removed. They will remain where they are in the list, but will be re-boo marked at the top of the list. This may be addressed in an update to Navigator. Let me know if this is the way you prefer it.
- **Find>By Test…** Displays a dialog for you to enter a boolean test. Enter any test, using **tID** to refer to the cont being tested. For example, put "the height of tID is 23," to find objects 23 pixels tall. Put "the backgroundcolor tID is 255,0,0 and the vis of tID is true," to find visible red objects. Searches highlighted items in the list, includ bookmarks. Matching items are bookmarked. **Bookmarks that match are removed and re-bookmarked**, placing them at the top of the bookmark list.
- **Find>In Enclosed By Test…** Displays a dialog for you to enter a boolean test. Enter any test, using tID to refe the control being tested. For example, put "the height of tID is 23," to find objects 23 pixels tall. Put "the back- groundcolor of tID is 255,0,0 and the vis of tID is true," to find visible red objects. Searches highlighted items i the list, including bookmarks**.** Matching items are bookmarked.
- **NOTE**: Bookmarks that match are **not** removed. They will remain where they are in the list, but will be re-boo marked at the top of the list. This may be addressed in an update to Navigator. Let me know if this is the way you prefer it.
- **Resize…** Opens a resizing dialog. The highlighted objects are resized relative to the object you clicked on to open the dialog. This means that the object you click on must be visible and accessible. There are four option
- **Align** the other objects' sides are aligned with the object you resize. If you resize the right side of the object, t right sides of all the other objects will align with it.
- **Match** the other objects are resized to be the same as the object you resize. This only affects the measureme you change. If you change only the right side of the object, the other objects' widths will match the object, but their heights will be unchanged.
- **Proportional** the other objects resize in proportion to their size relative to the object you resize.
- **Scale** the other objects are resized the same amount as the object you resize.
- **Property List…** Displays a list of all the properties of the highlighted objects, with the current values of the ob ject you clicked on.
    - Clicking on a boolean property flips its value from true to false and false to true.
    - Clicking on the Colors property displays the Navigator Color Editor (see below) to let you set the objects' colors.
    - Clicking on any other property displays an entry field to edit the property.
- **Edit Colors…** Displays the Navigator Color Editor. The color editor lets you select which colors you want to ed and then set those colors using simple sliders for Red, Green, and Blue.
    - When using the sliders, you can hold down the shift key to have all the sliders move together, giving you different shades of the same color, or hold down the option key to have all the sliders match, giving you shades of gray.

- **Edit Properties** Opens a new properties window for each of the selected object(s). Windows are arranged, ar locked to the object they are opened for. **NOTE:** this will work even for controls that are not currently displaye You can examine the properties for a control in any stack by selecting the stack on the stack menu, selecting t control you want, and selecting Edit Properties. This feature works only with Revolution.
- **Edit Scripts** Edits the scripts of all selected object(s). Windows are arranged. Like Edit Properties, the control does not need to be displayed. You can edit the script of any control in any open stack.
- **Show Properties** Opens one properties window for all selected object(s). Only properties all the objects have common are active. **NOTE:** this works on the selected controls in the Revolution/MetaCard environment you only use the Show Properties command for controls in "this card" of the topStack.
    - For other controls, you can use Edit Properties, the Properties contextual menu, the Custom Properties contextual menu, the property list, or the Set command in the Command Panel.
- **Count>Scripts** Counts the character length and the number of lines in the scripts of the highlighted objects. Note that this command simply counts the number of lines. Comments count the same as regular code. So d blank lines. Also note that it is impossible to overcount the command automatically avoids re-counting group objects that appear on multiple cards. If you select an object twice (the object itself and a bookmark) it will no be duplicated in the count.
- **Count>Enclosed Scripts** Counts the character length and total number of lines in the scripts of the highlighte objects and all the objects they enclose. Using this on a mainstack will count every line of every script of every control in every card or background of every substack as well as the mainstack.
- **Bookmark** Bookmark all selected object(s) copies full references to them to the top of the list. Bookmarked o jects don't need to be currently displayed, or even in a stack that is opened (you can't work with them if they'r not open, but the reference is still intact). Bookmarked controls don't need to be from the same card, or the same stack. Bookmarked controls can be used much like regular items in the list: edit their scripts or properti save them (if they are stack references) etc.
- **Command...** Displays the Command Panel. See the section on the Command Panel below. Note that this dis- ables the control list until you close the Command Panel.
- **Command submenu** Lists available saved command plugins. Command plugins are text files in the Navigato Commands folder in the Plugins folder. Any valid code for the Do button in the Command Panel can be savec a command plugin. NOTE: there is nothing to prevent you from saving property sets as commands, but they won't work. Let me know if this would be particularly valuable to you.

Table of Contents

PROPERTIES MENU

The Properties menu is available at the top of Navigator. It is marked with a "P." This menu affects all highlighted en tries, and displays every standard property that is common to all the objects represented. The Properties menu can

also be displayed as a contextual popup menu by holding down the Option/Alt key while clicking on the control list call up the contextual menu.

The properties contextual menu will work on all the selected controls. You can select bookmarked controls and reg lar displayed controls, and set properties for all the controls at once.

The properties menu remembers the most recent five properties you have set for any given object type; those pro erties appear in the menu alphabetically, but also at the top of the menu for easy access.

Commands available on the properties menu are:

- **Property Editor…** Displays a list of properties common to the selected controls. The default value provided v be the existing value of that property for the last object you clicked. For example, if you click on one control a then shift-click another to select a range, the value displayed will be for the control you shift-clicked. The nam and id of the object, along with the number of other controls being edited, is displayed at the top of the list. Clicking boolean values in the list changes their values. Colors will bring up the color selector. All other values display a text edit field with the current value in it. For values that cannot contain a return, hitting return/pres ing enter will set the value; for values that can contain a return, either click OK to set the value, or Cancel to d card your change to the value. **Important**: all values you change are set immediately. **There is no cancel or undo to the Property Editor… dialog**.
- **Object Inspector (1)** Displays **one** built-in LiveCode object inspector **for all of the selected control(s)**. This means that if more than one control is selected, the object inspector will display the alignment panel.
- **Object Inspector (N)** Displays one instance of the built-in LiveCode object inspector for **each** of the selected control(s). This means that if more than one control is selected, you will get more than one object inspector. There is no limit to this, Navigator will happily try to open 100 object inspectors if you have 100 controls selected.
- The next section includes the additional properties set in the preferences, and the most recent five properties for the object type selected.
- The final section is a list of all the relevant properties for the selected objects.

Table of Contents

Setting Properties Using the Properties Menu

Boolean Properties (True/False) Each boolean property is an item on the menu, with a submenu for true/false. If a the selected objects have the same value for that property, the value (true or false) is checked. If the property is tru for some objects and false for others, neither item is checked.

To set a boolean property, select true or false on the submenu of the property you want to set.

Non-Boolean Properties All other properties are simple menu items.

To set a non-boolean property, select the property you want to set. Navigator will display a text entry field for you t enter a value for the property.

If you used the contextual menu, the default value provided will be the existing value for that property for the obje you clicked on to get the contextual menu. NOTE: this means, for example, if you want to set the rect of several cor trols to the rect of one of them, you can select all the controls in the list, and then control-option-click on the one whose rect you want to set the others to. Select rect from the menu, and the text entry field will contain the rect of the control you control-clicked. When you click OK without modifying the value, the rect of all the controls will be se to the rect of the control you control-option-clicked.

If you use the properties menu at the top of Navigator, the default value provided will be the existing value for that property for the last object you clicked. For example, if you click on one control and then shift-click another to selec range, the value displayed will be for the control you shift-clicked.

Example 1: to hide a set of controls, select them. Then select false on the “visible” sub-menu on the properties me Or control-option-click one of them, and select false on the visible submenu of the contextual menu.

Example 2: you want to set the location of several controls to the location of one of them. Select all the controls in list. Control-Option-click on the one whose location you want to set the others to. Select loc from the menu. The te entry field will contain the loc of the control you control-option-clicked as a default. Just click OK, and the controls w get the loc of the control you control-option-clicked. You could do this with the properties menu at the top of Navig tor, but you would have to be certain that the last control you clicked was the one you wanted to set them all to.

Add properties to the menu: the properties contextual menu includes all the properties in the “propertyNames” o the objects. Not all standard properties are listed in the propertyNames, and some properties are compound: thei component properties can be accessed by name, but aren’t in the propertyNames function. For example, the rect o an object is also the left, top, right, and bottom, for example. The colors property is another example. A complete li of all the properties of an object is displayed in the Property List in the contextual menu.

It is possible to add other property names to this menu in addition to the standard properties: the revNavigator sta has custom properties in the custom property set gSBAddPropsProp as follows:

- Button
- Card

- Field
- Graphic
- Group
- Image
- Player
- Scrollbar
- Stack

Each of these custom properties contains the list of additional property names to include on the menu for that typ object. For exmaple, “loc” is included in most of the above. Use the **Add Properties…** item in the **Actions** menu to modify these values. See the Add Properties… entry in the Actions Menu section for details.

Table of Contents

CUSTOM PROPERTIES MENU

The Custom Properties menu is available at the top of Navigator. It is marked with a “U.” This menu affects all high- lighted entries. It displays a menu with every custom property set any of the selected objects has, with a checkmar next to the current custom property set, followed by every custom property in the selected CustomPropertySet tha any of the selected objects has. It can also be displayed as a contextual popup menu by holding down the Commar and Option keys (Control and Alt keys on PCs) while calling up the contextual menu.

NOTE: The custom properties contextual menu will work on all the selected controls. You can select bookmarked c trols and regular displayed controls, and set properties for all the controls at once.

Add New… Displays the Ask dialog to get the name of the new custom property, then displays an entry field to get the value for it.

———-

Setting Custom Properties Using the Properties Contextual Menu:

Boolean Properties (True/False) If any of the items on the menu has the custom property set to either true or fals the property will be displayed as a boolean property, with a submenu for true/false. If all the selected objects have the same boolean value for that property, the value (true or false) is checked. If the property is true for some objec and false for others, neither item is checked. Only the objects where the value is boolean will be considered when checkmarking the menu.

To set a boolean property, select true or false on the submenu of the property you want to set. If you want to chan a boolean custom property of an item to a non-boolean value, use the Set button on the Command Panel instead.

Non-Boolean Properties All other properties are simple menu items.

To set a non-boolean property, select to the property you want to set. Navigator will display a text entry field for yo to enter a value for the property.

If you used the contextual menu, the default value provided will be the existing value for that property for the obje you clicked on to get the contextual menu. NOTE: this means, for example, if you want to set a custom property of several controls to the same value one of them already has, you would select all the controls in the list, and then co trol-option-command-click on the one whose property you want to set the others to. Select the custom property's name from the menu, and the text entry field will contain the value for that property for the control you control-op tion-command-clicked. Just click OK, and the controls will have that property set to the value of the property for the control you control-option-command-clicked.

If you used the custom properties menu at the top of Navigator, the default value provided will be the existing valu for that property for the last object you clicked. For example, if you click on one control and then shift-click another select a range, the value displayed will be for the control you shift-clicked.

Table of Contents

HANDLERS MENU

The Handlers menu is available at the top of Navigator. It is marked with a script icon. If more than one control is highlighted, then the only item on the menu will be "Edit Scripts." Selecting it will open a script editor window for ea control. This menu can also be displayed as a contextual popup menu if only one control is highlighted by holding down the Command key (Control key on PCs) while calling up the contextual menu. Commands available on the Ha dlers menu are:

- **Copy Script** Places the entire script of the control onto the clipboard.
- **Paste Script** Sets the script of the control to the clipboard. **Note:** this command **does not validate the scrip** when setting it. If the script on the clipboard has a syntax error, it will still be set as the script
- **Put Handler List** Puts a list of the handlers in the object's script and behavior into the message box.
- **Edit Script** Opens the script editor with the object's script. TBD: add a command to edit the behavior script if object has a behavior set.
- **Send:** This sub-menu lists all the on, function, setProp, and getProp handlers in the object's script and behavi

with a separator between them. Select any item from this menu to trigger that handler. If the handler takes n arguments, this will happen immediately. If the handler takes arguments, a series of prompts will allow you t specify up to nine arguments. This will be improved with a single dialog at some point. If the handler returns value, it will be displayed in a dialog. This fails with arrays at present.

- **A list of all the handlers in the object's script and behavior**, with a separator between them. Selecting any handler opens the script editor with either the object's script or behavior open, scrolled to the selected handl

If the object's script is empty, "Edit Script" will display.

Select any handler in the menu to edit the script of the object and display the handler selected. If the object has no script, select "Edit Script" to edit its script and get started.

TBD: a way to get a template for calling a handler.

Table of Contents

STACK MENU

Allows you to select any user stack to examine with Navigator. The length of the menu is limited to the **Menu Limit** you set in the prefs on the **Actions** menu. Use the **Stack List** to display more stacks.

the topStack The default "the topStack" makes it reflect the topStack always. As you click from stack to stack, the c rent topStack is reflected in Navigator.

Select a stack by pointing to it: hold down the Option/Alt key while selecting "the topStack"; continue to hold dow the Option/Alt key and place the pointer over the stack you want to browse. Release the Option/Alt key to browse t controls in the stack.

the mouseStack hold down the Option/Alt key while selecting "the mouseStack"; continue to hold down the Option/Alt key and place the pointer over the stack you want to browse. Release the Option/Alt key to browse the con trols in the stack. If you are not holding down the Option/Alt key when you select this menu choice, an alert display to tell you to hold down the Option/Alt key. You can hold down the key as you close the alert, and then select a sta

Stack List Displays a list of your open stacks in the control list. Substacks are grouped with their mainStacks. Hold down the Option/Alt key to include development environment stacks.

The menu is alphabetized by default; you can display the menu in the order returned by Revolution/MetaCard by

holding down the Shift key while clicking the menu.

Inspect all open stacks by holding down the **Option/Alt** key when clicking the menu.

Go to the stack selected (instead of setting it as the stack to examine) by holding down the **Command/Control** k as you select it.

Table of Contents

CARD MENU

Allows you to select any card to examine with Navigator. The length of the menu is limited to the **Menu Limit** you s on the **Actions** menu. If you have a large number of cards (or groups), you can display them in the list instead of in the menu by selecting "**card list**" ("**background list**"). Then you can use the contextual menu to display controls fo the card (or group).

The default "**this card**" makes the list the controls of whichever card is showing in the stack.

By default, the menu displays the cards in their order within the stack. You can **alphabetize** the card menu by holc down the **Shift** key while clicking the menu.

The card menu shows **backgrounds** (**groups**) instead if you hold down the **Option/Alt** key. Only backgrounds with the **backgroundBehavior** property set to true are displayed. You can see a list of **all** the backgrounds/groups in th control list by selecting "**background list**" on the menu. You can then browse the controls of a particular group by control-clicking it and selecting "**Browse Controls**."

Go to the card selected (instead of setting it as the card to examine) by holding down the **Command/Control** key a you select it on the menu.

Table of Contents

COMMAND PANEL

Select **Command...** on the **Actions** menu or the contextual menu to display the **Command Panel**. At least one line must be highlighted. Opening the Command Panel disables the control list. You have four buttons in the Commanc Panel: Close, Set, Do, and Save.

Table of Contents

CLOSE

Closes the Command panel and enables the control list again.

Table of Contents

SET

Uses the entries in the Command field to set properties on the selected objects. Any properties can be set to any v ues, including custom properties (within the current custom property set). The format for property/value pairs is:

propertyName=value

No quotes are required. Multi-line values are allowed as long as they don't include an equals sign. You can set as many properties at a time as you like, so this would be fine:

```
fillfore=red
rect=40,40,90,90
zMyCustomProperty=some value I use a lot
to keep track of stuff.
enabled=false
```

In this example, note that "some value I use a lot to keep track of stuff" does not need to be quoted. If you quote it, the quotes will be included in the value of the property. Also note that it is multi-line. This command would change the controls selected to red, make them the same size, disable them, and set a silly custom property on them.

Table of Contents

DO

Runs the code in the Command field against each of the selected objects. This is extremely flexible. All looping and references are taken care of for you.

The following variables are available:

- **tCount** the number of selected controls.
- **tIndex** the number of the current control as the code executes on one control after another.
- **tID** the long id of the current control as the code executes
- **tIDList** An array of the long IDs of the selected controls. The index is the numbers 1, 2, 3...tCount.
- **tOwnerID** the long id of the owner of tID (the current object).
- **tCardID** the long id of the card that contains tID (the current object).
- **tStackID** the long id of the stack that contains tID (the current object).
- **tLastID** the long id of the previous control the code executed against. For the first control, it references the la
- **tNextID** the long id of the next control the code will execute against. For the last control, it references the firs

Note: **tLastID** and **tNextID** wrap. For the first object in the list, tLastID will be the id of the last object in the list. Fo the last object in the list, tNextID will be the id of the first object in the list.

Also, the **defaultStack** is set for each object to the stack that contains it, so for a series of cards this will work:

```
place background “menubar1” onto tID
```

Your code is wrapped in a **try...end try**, so an error on one object won't stop the code from working for other controls. That also means that your code can fail without letting you know.

It is very useful to know that a long id in a variable can be used as a complete reference to an object. Here is an exa ple of code you might run against a set of selected controls:

```
set the loc of tID to 50,(tIndex*30)
set the enabled of tID to not the enabled of tID
set the name of tID to ("another radio button" && tIndex)
set the zSomeCustomProp of tID to gSomeGlobalArray[tIndex]
```

etc.

When you are writing code to execute with the Do button, you can display a dialog to offer the user (yourself) a choice. Here's an example of how to do that:

```
if tIndex is 1 then
    setOptionLabel "Backgrounds:"
    put the backgroundNames of this stack into tNames
    setOptionListItems tNames
    showOptions
end if
```

The first thing to notice is that the code is wrapped in an **if** statement. tIndex is the number of the loop iteration we on. Putting the prompt code into an **if** based on "**tIndex is 1**" means this code will be executed only once no matte how many controls are selected. It will run the first time through, before the other loop code.

- **setOptionLabel** sets a label in the dialog to whatever text you like. Use it as a prompt.
- **setOptionListItems** will cause the prompt to display a list with the choices you give it. The list allows multiple selections.
- **setOptionOneItems** will cause the prompt to display a popup menu with the items you provide.
- **setOptionTwoItems** will cause the prompt to display a second popup menu with the items you provide.
- **showOptions** actually displays the prompt. Execution of your code halts until the user (you) clicks OK or Can

When **showOptions** finishes, local variables will be set to the values chosen by the user (you) as follows:

- **tOptionList** will be the items selected in the list, if you displayed the list. Each item is a separate line. You mig use it like this:

```
if word 1 of tID is "card" then
    repeat for each line tBGName in tOptionList
    place background tBGName onto tID
    end repeat
end if
```

- **tOptionOne** is the choice made in the first popup if you showed it.
- **tOptionTwo** is the choice made in the second popup if you showed it.

SAVE saves code in the Command field as a plugin. Navigator supports any number of plugins, which are just text f that store the code to run. Plugin commands are listed at the bottom of the contextual menu.

Table of Contents

ABOUT/SPLASH SCREEN

When Navigator first opens, it displays the About… screen. If you have checked the “Don’t Show” checkbox, the About… screen will go away automatically as quickly as it appeared. If you haven’t checked the “Don’t Show” checkb Navigator waits for you to dismiss the About…screen.

If you don’t see the About… screen when Navigator starts up, try closing Navigator and re-opening it. Navigator sho the About… screen as the last step in its initialization process, so if you don’t see it, something went wrong in the in tialization process.

You can dismiss the About… screen by clicking the close button at the upper left, or clicking anywhere in the About screen.

The about screen used to contain this documentation. As of LiveCode 7/Navigator 4.5, the documentation is here, i stead of in Navigator.

Table of Contents

KNOWN ISSUES

(minor) Choosing specific cards/backgrounds depends on what the name of the card/background is to determine whether the name is an id or an actual name. In either case I get text, so a card with id 1002 and a card named “car id 1002” would both show with the name “card id 1002” A card named “card id 1002” with an actual id of some othe number would cause a problem.

(possible) Show Properties doesn’t perform the same checks the Livecode development environment does before sending the command to open a properties palette. This might lead to problems, or maybe it won’t…

Table of Contents

PLANNED IMPROVEMENTS

Make the Show Properties command first select the highlighted controls, so that it should work properly even if you’re not viewing “This Card” of “The Topstack.”

Make the property list work with custom properties.

Add a command to the Custom Properties menu to display the property list with the custom properties.

Add a command to the Custom Properties menu to delete custom properties/sets.

Drag and drop between Navigators.

Drag and drop to/from controls in stacks.

Command to place selected controls onto the current card.

Command to convert scripts/button behaviors to script-only stacks.

Support for widget-specific properties (e.g. timezone on clock widget).

Improve handling of custom property sets.

Automate calls to private commands/functions by appending a public calling handler/function to the script (and th removing it).

Thanks, and let me know if you find any bugs or add any code.

Table of Contents

NAVIGATOR IS FREE TO USE

If it helps you develop in Livecode or MetaCard, you can donate to your favorite charity, or recommend me on LinkedIn, or...

Regards,

Geoff Canyon

gcanyon@gmail.com

Table of Contents

OLDER VERSION UPDATES

Table of Contents

UPDATES FOR 3.0 RC1

Worked around “Clone Navigator” limitations. Now have up to three copies of Navigator open at once, each with its own display.

Table of Contents

UPDATES FOR 3.0 B1

Fixed an issue with the “Property List” menu command in the main (not contextual) menu.

Table of Contents

UPDATES FOR 3.0 A3

Fixed an issue with Command-Double-Clicking where Navigator didn’t register the Command Key, and instead did whatever action was selected for double-clicks.

Corrected the behavior of the “Copy ID” item in the preferences for double-clicking. It was broken.

Added Option-Command-Double-Click to the preferences, so an action can be chosen for that. The default action f it is Bookmark/Remove.

Corrected an issue with editing the contents of a field or button. Where there were no styles in the text of the field button, the editing field might not have a textfont that matched that of the source field or button. This could result the text not being displayed correctly. The textfont of the editing field is now set to the effective textfont of the sou button or field, which should fix this issue.

Changed the behavior when selecting a new stack previously Navigator would retain the setting of displaying the li of cards if that were the setting, which always annoyed me. Navigator now defaults to displaying the controls of th active card.

Corrected behavior when Purging a stack. Previously, if you purged the stack Navigator was listing, you would get e

ror messages. Now, if the stack Navigator is listing is not available, Navigator defaults to the topStack and the curre card.

Table of Contents

NEW FEATURES IN 3.0

Navigator is clone-able. You can create a clone of Navigator and work with up to three copies at once.

Drag and Drop uses a new method to show you where the objects you are dragging will end up.

New menus: Properties, Custom Properties, and Script Handlers menus are available directly at the top of Navigat

New Property Editor: when editing properties that are not boolean, text entry is now through a scrolling field rath than an answer dialog, allowing much greater ease of editing.

Convenient Property Editor List: edit all the standard properties of a set of controls in a scrolling list.

Type-sensitive contextual menus now have extra entries depending on the object types selected.

Easily set button icons: Copy an image using Navigator's contextual menu. Select any number of buttons. The cor textual menu will have an item "Set Copied Img as Icon."

Easily set stack windowshape: Copy an image using Navigator's contextual menu. Select any number of stacks. T contextual menu will have an item "Set Copied Img as WindowShape."

New contextual menu items:

Rename displays a dialog for each of the highlighted items, renaming them.

Delete deletes the highlighted items.

Clone clones the highlighted items.

Copy Objects copies the highlighted items. Read the documentation for more information.

Paste Objects pastes the previously copied objects into any card, group, or stack.

Select selects the highlighted items in Revolution.

Edit Button and Field Contents contextual item for buttons and fields allows you to edit the contents of fields and buttons, including labels.

Resize any objects: resize contextual menu item allows live resizing of any group of objects in various ways.

New Color Editor: Colors contextual menu allows live modification of the colors of objects in various ways.

The selection of objects is retained while resizing.

Resizing Navigator is live.

The selection of objects is retained through the application/removal of filters. This makes it easy to find objects in a long list.

Bookmark by pointing at an object.

When bookmarking an object or selecting a stack by pointing, the hand pointer is used.

New Hide/Show method for Navigator: double-click the title bar.

Find in Scripts... command finds and bookmarks objects based on the content of their scripts.

Find in Enclosed Scripts... command searches all objects contained in the highlighted objects and bookmarks the jects based on the content of their scripts.

Find by Test... command finds and bookmarks objects based on a boolean test you supply.

Find Enclosed by Test... command searches all objects contained in the highlighted objects and bookmards the ob jects based on a boolean test you supply.

Edit Colors... contextual menu item displays a color dialog that allows setting any (and any combination of) colors a set of objects.

Align Objects... contextual menu item displays Revolution's alignment dialog for the selected objects.

Resize... dialog allows resizing objects based on the selected object.

Browse Controls contextual menu command causes Navigator to display only the controls in the selected group.

Bookmark the selected objects regardless of what is displayed in Navigator.

Copy objects directly from the list in Navigator.

Set button icons and stack windowShapes using the new copy command.

Place groups onto cards using the copy command.

Table of Contents

NEW FEATURES IN 2.5

The list can now show the open stacks.

Edit the custom properties of controls; either individually or in groups. Also, create new custom properties.

Contextual menus are smarter. They pop on mouseDown rather than mouseUp. If you pop a contextual menu by clicking on a selected item, the selection is unchanged and the contextual menu applies to all the controls. If you p a contextual menu by clicking on an unselected item, the selection will change to just the clicked item, and the cont tual menu applies to just the clicked item.

Hold down the option(alt) key while selecting “the topStack” Move the pointer over any stack. Let go of the option(a key, and Navigator lists the controls for the stack the mouse is over.

Rename and **Recolor** bookmarks.

Count Scripts reports line count and character count.

Count Enclosed Scripts counts the scripts for the highlighted objects and all the objects they contain.

Edit custom properties through the contextual menu, just like the regular properties.

Drag and drop support:

Drag controls to set their layer

Drag controls into and out of groups

Drag controls into the bookmark list to create bookmarks

Drag bookmarks to rearrange them

Drag bookmarks into the list of controls to add them to the current card or group.

Mix and match: drag bookmarks and controls into the bookmark list to rearrange the bookmarks and create new ones for any controls not already present as bookmarks

Mix and match: drag bookmarks and controls into the control list to rearrange the controls and add the bookmark controls

Drag cards in a card list to rearrange them in a stack

Drag controls/bookmarks to the desktop or other drag targets to save the long id of the controls.

Drag long id text into the list from other sources to create bookmarks. **NOTE:** this does not test the drag data. It wi create bookmarks for whatever text you drag in.

Drag a bookmark of a group into the control list is smart: if the group is in the same stack, it places it. Otherwise it copies it.

Drag a card bookmark into the control list to copy the card to the display stack.

Drag a stack bookmark into the control list to set the dragged stack’s mainstack to the display stack.

Select any number of cards in a card list, and a bookmark for a group. Click the group bookmark and drag into the card list. Places the group onto all the card references highlighted.