



# **CogTool 1.0b18**

## **User Guide**

**February, 2007**

**Human-Computer Interaction Institute  
School of Computer Science  
Carnegie Mellon University**

# Contributors to the CogTool Project

Bonnie E. John  
Gus Prevas  
Peter Centgraf  
Ken Koedinger  
Dario Salvucci  
Sandy Esch  
Alex Eiser  
Mike Horowitz  
Leonghwee Teo  
Jason Cornwell  
Don Morrison  
Samantha Konwinski  
Alex Faaborg  
Carmen Jackson  
Lily Cho

This research was supported by funds from the Office of Naval Research, N00014-03-1-0086. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U. S. Government.

# Conventions For This User Guide

## NOTES about CogTool 1.0b18:

This document is about CogTool 1.0b18, but other releases will follow. When some behavior is specific to CogTool 1.0b18 and is expected to change in a future release, a note appears in a gray box like this one.

## !!! KNOWN BUGs in CogTool 1.0b18:

When there is a known bug in CogTool 1.0b18, it is indicated with a note in a yellow box like this one.

This user guide is meant as a comprehensive reference to the functionality provided by CogTool. It has been structured based on the major stages of working with CogTool described below and the individual steps needed to accomplish them. If you would prefer to see a step-by-step example of how these steps fit together, please consult the *CogTool Tutorial*, which is available from the Publications page of the CogTool website.

<http://www.cogtool.org/publications.html>

Terms in this guide that are **CAPITALIZED** (in this fashion) refer to the items that are created and edited in CogTool. If the conventional English definition is intended, the words will have no special formatting.

## System Requirements

CogTool is a cross-platform application. Unless otherwise noted, instructions are identical whether CogTool is running on Mac OS X or Windows-based computers. Screen images of CogTool in this guide are taken from a Macintosh; only minor visual differences exist between these images and the PC.

CogTool is compatible with MacOS X 10.4 and above including Intel Macs (i.e., those shipped after December 2005). It is also compatible with Microsoft Windows XP. CogTool will function on older versions of Windows (down to Windows 98) but requires the GDI+ library, which you can install from Microsoft at

<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdkredist.htm>

## TABLE OF CONTENTS

1	CogTool Overview.....	7
1.1	What does CogTool do for me? .....	7
1.2	Strengths and Limitations .....	7
1.3	How do I make it work? .....	8
2	Install and Launch CogTool .....	10
2.1	System Requirements .....	10
2.2	Launching CogTool.....	10
3	Working with PROJECTS .....	11
3.1	What is a PROJECT? .....	11
3.2	PROJECT Windows.....	11
3.3	Open an Existing PROJECT .....	11
3.4	Create a New PROJECT.....	12
3.5	Close a PROJECT .....	12
3.6	Save a PROJECT .....	12
3.7	Save as... (Rename) a PROJECT .....	13
3.8	Re-opening a PROJECT window.....	13
3.9	Contextual menus available from a PROJECT window .....	14
4	Developing a DESIGN Storyboard .....	16
4.1	What is a DESIGN storyboard? .....	16
4.2	DESIGN Windows .....	16
4.3	Create the Initial DESIGN and TASK .....	17
4.4	Create a New DESIGN .....	18
4.5	Open a DESIGN for Editing .....	18
4.6	Rename a DESIGN.....	19
4.7	Close the DESIGN Window .....	20
4.8	Delete a DESIGN .....	20
4.9	Change the Order of DESIGNS using Cut and Paste .....	20
4.10	Zoom a DESIGN .....	21
4.11	Bring a DESIGN Window to the Front.....	21
4.12	Export a DESIGN to HTML for use in User Testing.....	22
4.13	Contextual menus available from a DESIGN window .....	25
5	Edit FRAMES.....	26
5.1	What is a FRAME?.....	26
5.2	FRAME windows .....	26
5.3	Create new FRAMES.....	27
5.4	Rename a FRAME.....	27
5.5	Edit a FRAME.....	28
5.6	Background Image of a FRAME.....	29
5.7	Close the FRAME Window .....	31
5.8	Delete a FRAME .....	31
5.9	Zoom a FRAME .....	31
5.10	Bring a FRAME window to the Front .....	32
5.11	Change a FRAME'S Position in the DESIGN.....	32
5.12	Contextual menus available from a FRAME window .....	32

6	Define WIDGETS.....	33
6.1	What is a WIDGET?.....	33
6.2	Create a WIDGET .....	34
6.3	Copy a WIDGET from one frame to another.....	37
6.4	Rename a WIDGET.....	38
6.5	Delete a WIDGET .....	38
6.6	Duplicate a WIDGET within the same frame .....	38
6.7	Move a WIDGET .....	38
6.8	Nudge WIDGETS into place .....	39
6.9	Overlapping WIDGETS .....	39
6.10	Aligning WIDGETS.....	39
6.11	Resize a WIDGET .....	39
6.12	Set WIDGET Layer Color.....	40
6.13	Choosing the Appropriate WIDGET Type .....	40
7	Frames via TRANSITIONS .....	45
7.1	What is a TRANSITION?.....	45
7.2	Create a TRANSITION .....	45
7.3	Delete a TRANSITION .....	47
7.4	Change the Source of a TRANSITION .....	47
7.5	Change the Destination of a TRANSITION .....	48
7.6	Edit TRANSITION Properties .....	48
8	Create and Organize TASKS .....	51
8.1	What are TASKS and TASK Groups?.....	51
8.2	Create a New TASK .....	51
8.3	Create a TASK Group.....	52
8.4	Rename a TASK or TASK Group.....	52
8.5	Delete a TASK or TASK Group .....	52
8.6	Change the Order of TASKS and TASK Groups.....	53
8.7	Change the Computed Value for a TASK Group.....	53
8.8	Organization Strategies for TASKS .....	53
9	Record a SCRIPT .....	54
9.1	What is a SCRIPT?.....	54
9.2	SCRIPT windows .....	54
9.3	Open a SCRIPT for Editing .....	56
9.4	Choose a Start Frame for the SCRIPT.....	56
9.5	Open the Demonstration Window .....	57
9.6	Zoom the Frame .....	57
9.7	Demonstrate Actions for a SCRIPT .....	58
9.8	Delete Demonstrated Action.....	59
9.9	Review Automatic SCRIPT Steps.....	59
9.10	Review Transitions Defined on a Widget.....	60
9.11	Insert a “Look at” Step .....	60
9.12	Insert a “Wait for System” Step .....	60
9.13	Insert or remove a “Think” Step .....	61
9.14	Review a Previous Step .....	61

9.15	Restart from a Previous Step.....	61
9.16	Close the SCRIPT Editor and Compute the Prediction .....	62
9.17	Re-Compute a SCRIPT.....	62
9.18	Exporting SCRIPTS to other formats.....	63
9.19	Contextual menus available from a SCRIPT window.....	64
10	Model Visualization .....	66
11	Tips.....	70
11.1	Tips for DESIGN Storyboards .....	70
12	Appendix A - How to Report a Bug and Get Support .....	71
13	Appendix B – Known Bugs in CogTool 1.0b18.....	72
14	Appendix C – Notes about CogTool 1.0b18 .....	73

# 1 CogTool Overview

## 1.1 What does CogTool do for me?

### *Predicts execution time of a skilled user.*

CogTool predicts total execution time for a skilled user performing a particular sequence of actions on a system.

The predictions made by CogTool are based on a psychological theory of human cognitive and motor capabilities, called the Keystroke-Level Model (KLM), which has been used and validated repeatedly by academics and practitioners since 1980 (Card, S. K., Moran, T. P., & Newell, A. *The keystroke-level model for user performance time with interactive systems*. Communications of the ACM, 23 (7), (1980) 396-410). CogTool automates the application of KLM to specific problems, providing an alternative to time-consuming and expensive user testing.

CogTool can only predict what KLM can predict, that is, execution time for a skilled user of a system. It cannot predict learning time, problem-solving paths, or user satisfaction with a system. It also cannot predict the path of hurricanes.

### *Produces a “reusable” design mock-up that is as useful for user testing with novices as it is for predicting skilled performance time*

When you use CogTool to predict skilled performance time, you create a storyboard of a system design that can be exported to HTML and used in user tests. See 4.12 for more information about this feature of CogTool. A discussion of multi-purpose prototypes can be found in: John, B. E. & Salvucci, D. D. (2005) Multi-Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. *IEEE Pervasive Computing* 4(4), 27-34.

## 1.2 Strengths and Limitations

Like any tool, CogTool has strengths and limitations that make it appropriate for certain kinds of use. Because CogTool depends on a computational cognitive engine called ACT-R, it can only predict behavior that ACT-R predicts. Here are some examples of areas we think CogTool has the greatest potential and areas where CogTool’s limitations might cause problems.

### ***Strength: Traditional GUI Interactions***

The features and theory embedded in CogTool are tailored to the types of interaction typically found in traditional graphical user interfaces (GUIs). Clicking on pull-down menus, check boxes, radio buttons, mouse-pointing, and simple forms are well supported.

### ***Weakness: Touch-Typing***

ACT-R models typing as a sequence of individual key-presses. This is sufficient for modeling short commands and forms, but is not a good fit for extended lengths of touch-typed prose. A more complete model of typing is planned for a future version of CogTool, but in CogTool 1.0b18 we don’t recommend making predictions for **TASKS** with a large amount of text entry.

### ***Weakness and a Work-Around: Drag and Drop***

ACT-R cannot yet drag and drop. Throughout CogTool documentation and examples, we recommend that you approximate drag-and-drop with hover-and-click. That is, have a hover action over the item that would be the recipient of the mouse button press in a drag-and-drop operation, then click on the destination of the drag-and-drop. Both operations have one down-press on the mouse button, one release of the mouse-button and one horizontal movement. Although detailed empirical studies have revealed slight differences in time for these operations, at the level of accuracy associated with KLM, hover-and-click is a sufficient approximation for drag-and-drop.

### ***Weakness: Dynamic Interfaces / Animation***

Since CogTool uses a storyboard metaphor for mocking up a user interface **DESIGN**, it is difficult to represent highly dynamic animated user interfaces. However, clever construction of the storyboard can approximate a more dynamic interface.

### ***Strength: Mobile Devices***

CogTool provides functionality to simulate interaction with mobile devices using a fingertip or stylus. CogTool can therefore predict performance with devices utilizing hardware buttons or pen-based input. Examples of mobile devices with skilled users might include personal digital assistants, cellular phones, and navigational devices.

### ***Remember: CogTool Predicts Only Skilled Performance***

The KLM theory, which CogTool uses to make predictions, is intended to simulate the performance of well-practiced users on interfaces with which they are already familiar. It does not take into account the time a new user requires to explore an interface and find functionality, nor does it predict the likelihood that a function will be understood. This is an inherent limitation of KLM theory, and therefore it is also a limitation of CogTool. However, CogTool may be combined with other techniques to understand user performance across a range of experience levels.

## **1.3 How do I make it work?**

This section gives a brief overview of the stages of using CogTool. Subsequent sections will explain each item in more detail.

### ***Install and Launch CogTool***

This guide assumes that you have downloaded and installed CogTool 1.0b18 from <http://www.cogtool.org/download.html>. If you have not yet done so, please visit the Download page of the CogTool website and follow the instructions now. Section 2 provides more details.

### ***Create a PROJECT***

In CogTool, a **PROJECT** is the center of your work and contains the user interface **DESIGN** and the **TASK** for which you would like to make predictions. Therefore, the first step in working with CogTool is to create a **PROJECT**. Section 3 has more information on how to work with **PROJECT**.

### ***Create a DESIGN Storyboard***

The interfaces that you want to make predictions about are expressed in storyboards.

Sections Developing a DESIGN Storyboard 4 thru 7 describe how to create a storyboard for the interface DESIGN upon which TASKS will be performed.

### ***Define TASKS***

Once the storyboard for your DESIGN has been created, you will define a set of TASKS whose performance you would like to predict. Section 8 of this guide will show how to create TASKS and use TASK GROUPS to organize them.

### ***Record a SCRIPT***

The steps a user would perform to accomplish a TASK are recorded in SCRIPTS. Section 9 shows how to record a SCRIPT of specific user actions by demonstrating them on the DESIGN you created in Sections 4 thru 7. Once the SCRIPTS are defined, CogTool will make its predictions.

### ***Rinse, Repeat!***

Creating a DESIGN, a set of TASKS, and SCRIPTS is the starting point for creating new DESIGNS, exploring different TASKS, and comparing the predictions. With a basic familiarity with CogTool, you now have a great new tool for improving your user interface designs. Section 9.19 discusses best practices and helpful tips for getting the most out of CogTool.

## 2 Install and Launch CogTool

### 2.1 System Requirements

CogTool is compatible with MacOS X 10.4 and above including Intel Macs (i.e., those shipped after December 2005). It is also compatible with Microsoft Windows XP. CogTool will function on older versions of Windows (down to Windows 98) but requires the GDI+ library, which you can install from Microsoft at <http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdkredist.htm>

### 2.2 Launching CogTool

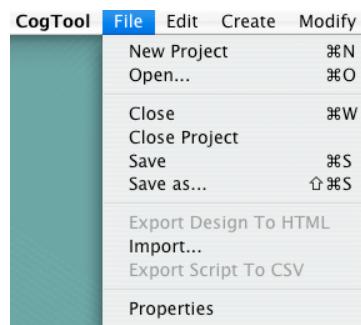
Open the CogTool application that you've previously installed. If you haven't installed it yet, follow the instructions at <http://www.cogtool.org/download.html>.

If you are working on a Windows-based PC, CogTool should have a folder in the Start Menu. If you are working on Mac OS X, CogTool will be wherever you placed it after extracting it from the download archive. When you launch CogTool, the "CogTool Start-up" box will appear, it gives you the choice of creating a new **PROJECT** or opening an existing one (Figure 1 below).



**Figure 1. The CogTool Start-up box appears every time CogTool is launched**

If you click the box closed without choosing either Create or Open you can still access CogTool commands using the menu bar. The File menu contains the commands New Project and Open.



**Figure 2. The CogTool toolbar with the File menu open**

**KNOWN BUG #1.** In Mac OS X, you cannot launch CogTool by double-clicking on a saved project file (.cgt file). You must open a saved project by launching CogTool itself and then opening the saved project file with the Open option in the CogTool Start-up Dialog Box or from the Open command in the File menu.

## 3 Working with PROJECTS

### 3.1 What is a PROJECT?

A **PROJECT** is a center of work. All the parts necessary for CogTool to make performance predictions are kept in a **PROJECT** file. The **PROJECT** name is also the name of the .cgt file that is created by CogTool. A **PROJECT** contains the user interface **DESIGNS**, the **TASKS** being studied and the **SCRIPTS** that define the human performance being predicted.

### 3.2 PROJECT Windows

The **PROJECT** window shows a matrix of cells. The column headers are the names of different storyboard **DESIGNS** and the row headers are the names of different **TASKS** and the **TASK GROUPS**. For example, in Figure 3, there are

- 2 **DESIGNS** “pad layout” and “row layout”;
- 2 **TASK GROUPS** “Use keyboard” and “Use touchscreen” each containing
- 2 **TASKS**. “1 number search” and “2 numbers search”

At the intersections of the **DESIGNS** and **TASKS** are cells called **SCRIPT** cells, which show the predicted execution time for that **TASK** using that **DESIGN**. If **SCRIPTS** have not yet been demonstrated, those **SCRIPT** cells are blank. If the **SCRIPT** has been demonstrated, but not computed, a question mark appear in the **SCRIPT** cell as in the lower right of Figure 3.

**DESIGNS** are defined in sections 3.9 through 7, **TASKS** in section 8 and **SCRIPTS** in section 9.

Tasks		pad layout	row layout
		Sum: 8.591 s	Sum: 34.677 s
Use keyboard	Search for 1 number	2.954 s	13.320 s
	Search for 2 numbers	5.637 s	21.337 s
Use touchscreen	Search for 1 number	Sum: 12.575 s	Sum: 7.528 s
	Search for 2 numbers	4.947 s	7.528 s
		7.628 s	?

Figure 3. PROJECT window “Phone Search”  
has 2 DESIGNS and 2 TASK GROUPS each containing 2 TASKS.

### 3.3 Open an Existing PROJECT

To open a **PROJECT** previously saved, a user may use the “Open...” command in the “File” menu or click on the Open button in the CogTool “Start-Up” box that appeared when you launched CogTool. You will be prompted to select a file to open by a standard Open dialog box.

### 3.4 Create a New PROJECT

To create a new **PROJECT**, you can click on the Create button in the CogTool “Start-Up” box that appeared when you launch CogTool or use the “New **PROJECT**” command in the “File” menu. Because **PROJECTS** always contain at least one **DESIGN**, you will be immediately prompted to name a new **DESIGN** for the new **PROJECT** (Figure 4).

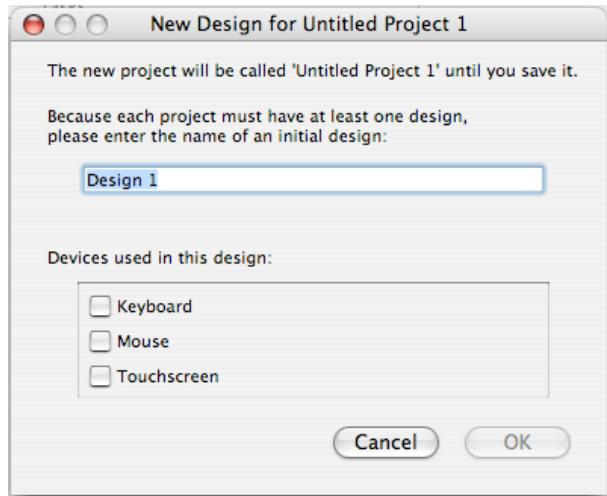


Figure 4. New **DESIGN** box appears every time a new **PROJECT** is created.

### 3.5 Close a PROJECT

To close a **PROJECT**, you must close all windows associated with the **PROJECT**. You may use the close box on each window or choose the “Close” command from the “File” menu to close a window. The “Close Project” command in the “File” menu closes all windows of the **PROJECT**. If the **PROJECT** has unsaved modifications when its last window is closed, you will be prompted if you would like to save your changes.

### 3.6 Save a PROJECT

Save a **PROJECT** by selecting the “Save” command in the “File” menu. A standard Save dialog box comes up to ask for a file name and location. You may name the file whatever you wish, by default it will be given a .cgt extension. This name (minus the .cgt extension) now appears in all **PROJECT** window title bars. The images in this guide will show the name “Tutorial Project”. When a **PROJECT** has been changed from a saved state, an asterisk appears in each window title bar. For example, Figure 5 shows the **PROJECT** window with unsaved modifications, indicated by the asterisk in front of “Project: Untitled Project 1”.

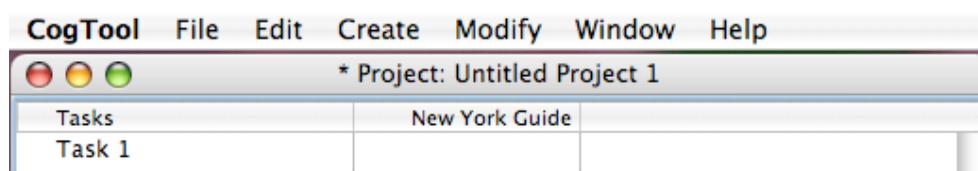
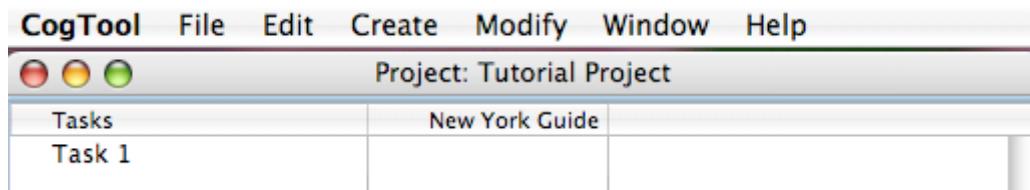


Figure 5. The asterisk next to the word **PROJECT** indicates that changes have been made since it was last saved.

In Figure 6 there is no asterisk in the title bar indicating no modifications are waiting to be saved.



**Figure 6.** This PROJECT has just been saved no changes have occurred yet. Notice that no asterisk appears in front of word PROJECT.

### 3.7 Save as... (Rename) a PROJECT

Selecting the “Save as...” command in the “File” menu will bring up the standard “Save as” dialog box for you to choose a new name and /or new location for the PROJECT.

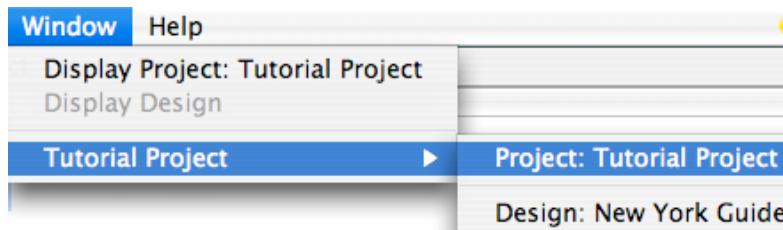
**NOTE #1.** A PROJECT should only be renamed or copied using the “Saved as” command. If the file name is changed outside of CogTool (e.g., edited using the operating system – in the folder) the PROJECT name that appears at the top of the windows WILL NOT BE CHANGED. The PROJECT must be saved under a new name from within CogTool in order to have the window names change.

### 3.8 Re-opening a PROJECT window

You can open a closed PROJECT window if one of its DESIGN, FRAME or SCRIPT windows are open by making any of its windows active then selecting the “Display Project:” command from the Window menu.

You can also bring an open PROJECT window to the front, from any window, by opening the Window menu and either by selecting the PROJECT window’s name in the top half of the drop-down menu or selecting it in the drop down list. The drop down menu lists all windows open for that project.

In Figure 7 the PROJECT window “Tutorial Project” is the only PROJECT open (and is being brought to the front). And “New York Guide” is the only DESIGN window open.



**Figure 7.** Bringing to the front the “Tutorial Project” PROJECT window

### 3.9 Contextual menus available from a PROJECT window

There are six different contextual menus available in the **PROJECT** window. These menus are accessed by right-click (PC) or control-click (Mac) on a certain area within the window.

- 1) Background (see Figure 8) – clicking anywhere below the last task line

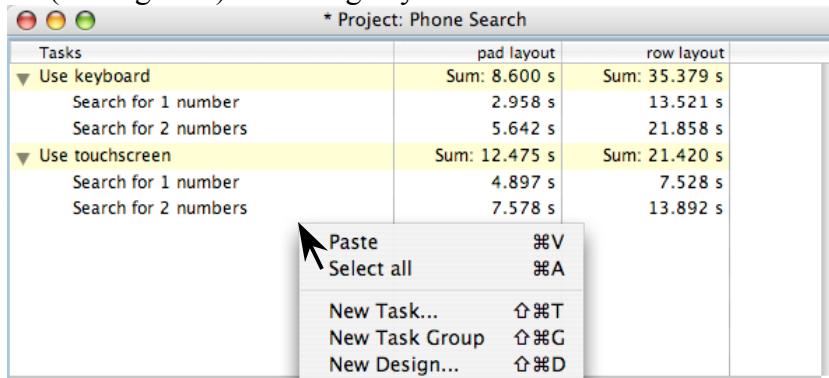


Figure 8. Background contextual menu in the PROJECT window.

- 2) TASK (see Figure 9) – clicking on a task line to the left of the DESIGN columns.

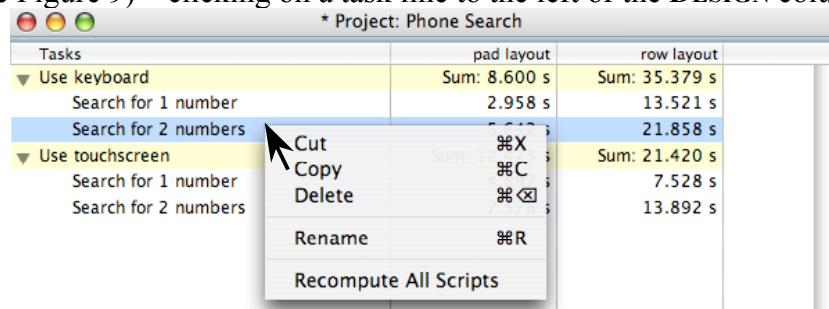


Figure 9. TASK line contextual menu

- 3) TASK GROUP (see Figure 10) -clicking in a yellow bar area to the left of the DESIGN columns.

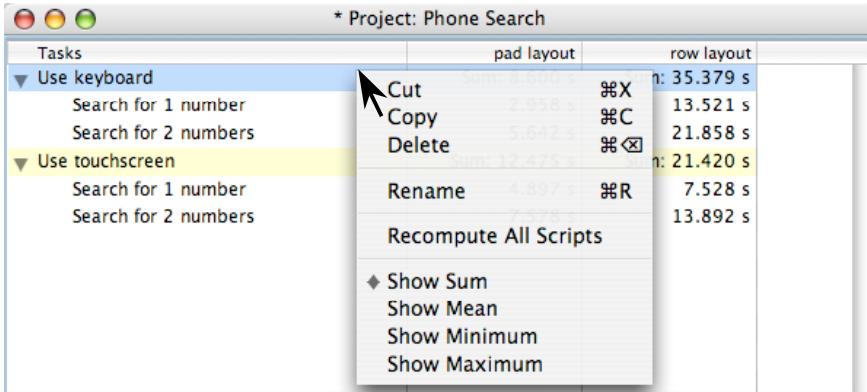


Figure 10. Task Group contextual menu

- 4) DESIGN (see Figure 11)- clicking on a DESIGN name cell such above the TASK lines.

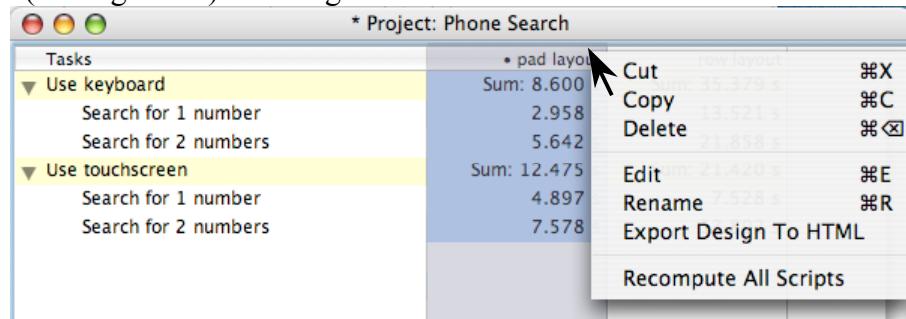


Figure 11. DESIGN cell contextual menu

- 5) SCRIPT (Prediction) cell (see Figure 12) – clicking on a SCRIPT (prediction) cell under a DESIGN name.

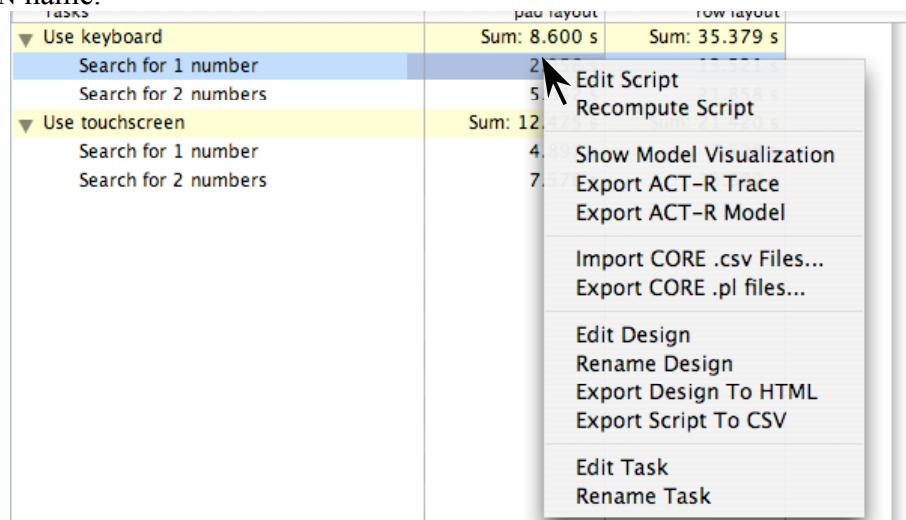


Figure 12. SCRIPT cell contextual menu

- 6) SCRIPT (Prediction) Group (see Figure 13) – clicking on a yellow cell under a DESIGN name.

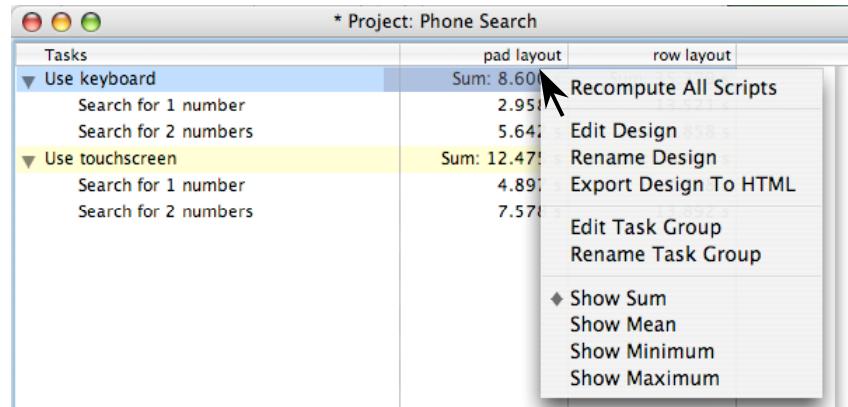


Figure 13. Script Group contextual menu

## 4 Developing a DESIGN Storyboard

### 4.1 What is a DESIGN storyboard?

A DESIGN storyboard is a series of CogTool FRAMES that represent the user interface of the system you are analyzing. Each FRAME will show an individual screen of the user interface, interactive elements on the screen (e.g., menus, list boxes, etc.) or around that screen (e.g., physical buttons), and the user actions that trigger changes from one screen to another. The FRAMES are linked together via TRANSITIONS, which shows how a particular user action causes the interface to react in one way or another.

**NOTE #2.** CogTool's storyboard approach was inspired by James Landay's SYLK system (Landay, J. A. & Myers, B. A. "Interactive Sketching for the Early Stages of User Interface Design." In *Proceedings of CHI '95*, Denver, CO, May 1995, pp. 43–50.)

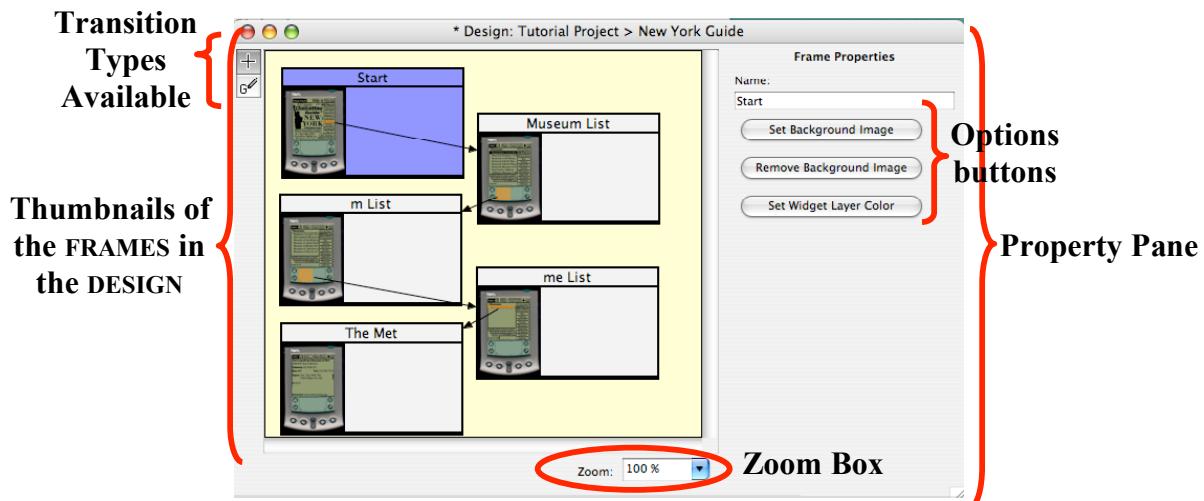
### 4.2 DESIGN Windows

DESIGN windows have a light yellow background to distinguish them from other CogTool windows and the title bar starts with the label "Design:".

The DESIGN window is where FRAMES are assembled. The window contains:

- A toolbar for working with TRANSITIONS (Section 7)
- A scrollable pane which contains thumbnails of all the FRAMES within the DESIGN and the TRANSITIONS that connect them,
- Properties that pertain to the selected object (FRAME or TRANSITION) and the available options for that object
- a Zoom box

Figure 14 is the DESIGN window for the "New York Guide" DESIGN.



**Figure 14. Storyboard for the ChoiceWay™ Guide to New York City that illustrates a museum look-up task<sup>1</sup>. Start Frame is selected.**

### 4.3 Create the Initial DESIGN and TASK

When you create a new **PROJECT**, you are immediately presented with a dialog box (Figure 4) to help you create the first **DESIGN** for this **PROJECT**. You may choose any name you wish to describe this **DESIGN**. Type the name into the text field provided and hit the “Enter/Return” key. This user guide uses the name “New York Guide.”

Also asked, is which **DEVICES** will be used with this **DESIGN**. **DEVICES** represent the ways that the **DESIGN** will receive input from its end users. At least one **DEVICE** must be selected in order to continue, since end users must be able to manipulate the **DESIGN** somehow! Confirm your choice(s) by clicking “OK” or hitting the “Enter/Return” key (Figure 15).



**Figure 15.** The dialog box presented when a new **PROJECT** is created. The default **DESIGN** name has been changed to “New York Guide”, which is a **DESIGN** for a Palm® PDA. Since the Palm® PDA is operated through a touch screen (not a mouse or keyboard), only the Touchscreen device is checked.

The new **DESIGN** appears as a column in the **PROJECT** window with the name you specified. Each column represents a specific **DESIGN** that you can compare using CogTool. Figure 16 shows the **DESIGN** named “New York Guide”. CogTool has also created the first **TASK**. In Figure 16 “Task 1” is highlighted and ready for you to type in a more appropriate name for this **TASK** (indicated by broken blue highlight).

* Project: Untitled Project 1	
Tasks	New York Guide
	Task 1

**Figure 16.** “New York Guide” appears as a column heading. “Task 1” is highlighted and ready for you to type in a **TASK** name.

<sup>1</sup> Adapted from Luo, L. & John, B. (2005) Predicting task execution time on handheld devices using the keystroke-level model. Proceedings of CHI, 2005 (Portland, OR, April 2-7, 2005) ACM, New York. This task is also used in the CogTool Tutorial.

Figure 17 shows the new name “Lookup the MET”. The solid blue highlight that extends the whole row indicates that the **TASK** name has been selected and is waiting for another command such as Cut, Copy, Edit, or Delete. **TASKS** are discussed further in Section 8.

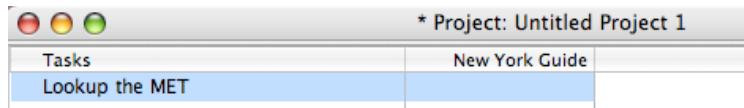


Figure 17. “Lookup the MET” appears as a **TASK** name.

#### 4.4 Create a New DESIGN

To create a new **DESIGN** in a **PROJECT**, you can

- choose the “New Design...” item from the “Create” menu, or
- right-click (PC) or control-click (Mac) on the Project window background to bring up the contextual menu then select “New Design” or
- use the keyboard shortcut  $\text{ctrl } \text{shift } \text{N}$  on Mac OS,  $\text{ctrl } \text{N}$  on Windows.

You will be presented with a dialog box in which you choose options for this **DESIGN**. The dialog box also asks which **DEVICES** will be used with the **DESIGN**. **DEVICES** represent the ways that the **DESIGN** will receive input from its end users. At least one **DEVICE** must be selected in order to continue since end users must be able to manipulate the **DESIGN** somehow! Confirm your choice(s) by clicking “OK” or hitting the “Enter/Return” key. The new **DESIGN** appears as a new column in the **PROJECT** window.

#### 4.5 Open a DESIGN for Editing

To open a **DESIGN** for editing you can either

- double-click on its name in the **PROJECT** window, or
- click on the name once to select the **DESIGN**, and then choose the “Edit” command from the “Edit” menu and select the “Edit” command (Figure 18), or

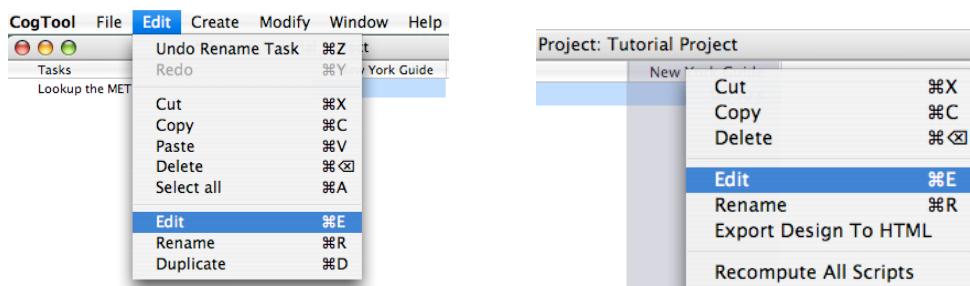
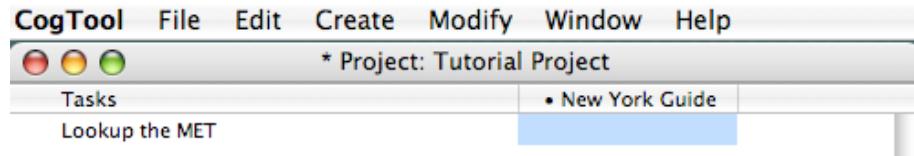


Figure 18. The **Edit** item in the **Edit** menu (on the left) or **Edit** in the contextual menu (on the right) will open the selected **DESIGN**.

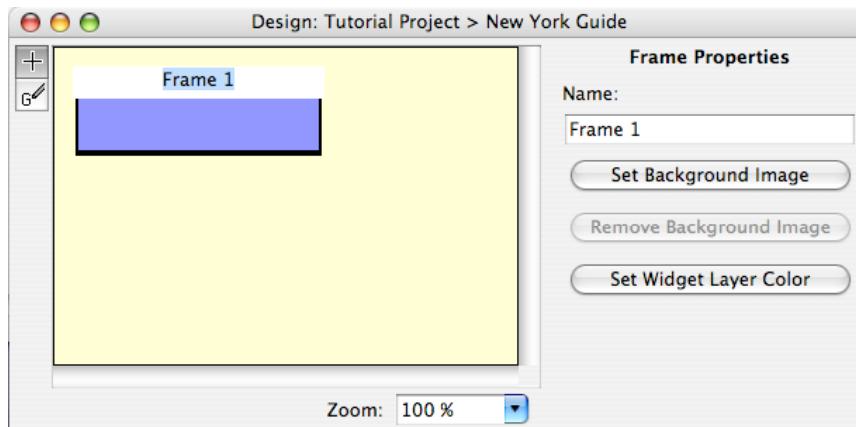
- right-click (PC) or control-click (Mac) on the **DESIGN** name to bring up the contextual menu then select “Edit”, or
- use the keyboard shortcut  $\text{ctrl } \text{E}$  on Mac OS,  $\text{ctrl } \text{shift } \text{E}$  on Windows, or
- if one of its **FRAME** or **SCRIPT** windows are open, making the **FRAME** or **SCRIPT** window active then select the “Open Design” command from the Window menu.

A selected **DESIGN** is indicated by a dot next to the selected **DESIGN**'s name and the **SCRIPT CELLS** below are highlighted in blue as shown in Figure 19.



**Figure 19.** The dot next to the DESIGN name “New York Guide” indicates that the DESIGN has been selected.

When a **DESIGN** window opens if no **FRAMES** have been created, a new **FRAME** is automatically created (Figure 20). When you select the **FRAME** title it will highlight in light blue and will be ready to have a more meaningful name entered. You can also edit the Name field in the Properties section.



**Figure 20.** “Frame 1” was automatically created when the “New York Guide” DESIGN was created, it’s title has been selected and is ready to be renamed.

## 4.6 Rename a DESIGN

To rename a **DESIGN** you can either

- 1) click on the name of the **DESIGN** once to select it, then choose the “Rename” command from the “Edit” menu, or
- 2) right-click (PC) or control-click (Mac) on the **DESIGN** name to bring up the contextual menu then select “Rename”, or
- 3) use the keyboard shortcut **⌘R** on MacOS, **CTRL ⌘R** on Windows.

A selected **DESIGN** is indicated by a dot next to the selected **DESIGN**'s name and the **SCRIPT CELLS** below are highlighted in blue as shown in Figure 19

A dialog box will appear (Figure 21), prompting you for a new name. You may use any name you wish. When you are done editing the name, press the “OK” button or the “Enter/Return” key to confirm the new name.



Figure 21. DESIGN rename dialog.

## 4.7 Close the DESIGN Window

When you are finished working with a DESIGN, close the DESIGN window to return to the PROJECT window.

To close the DESIGN window you can either

- 1) use the standard close box on the title bar of the window, or
- 2) choose "Close" from the "File" menu, or
- 3) use the keyboard shortcut ⌘W on MacOS, CTRL ⌘W on Windows.

## 4.8 Delete a DESIGN

To delete an existing DESIGN you can either

- 1) click on the name of the DESIGN once to select it, then choose the "Delete" command from the "Edit" menu, or
- 2) right-click (PC) or control-click (Mac) on the DESIGN name to bring up the contextual menu then select "Delete", or
- 3) use the keyboard shortcut ⌘⌫ on MacOS, CTRL ⌘⌫ on Windows.

You will know that the DESIGN is selected because a black dot will appear to the left of the DESIGN'S name, and the background of any SCRIPT cells in that column of the PROJECT window will become highlighted. A confirmation box will appear asking you to please confirm the deletion of the selected DESIGN. Press the "OK" button or the "Enter/Return" key to confirm the deletion.

## 4.9 Change the Order of DESIGNS using Cut and Paste

To change the order of DESIGN you must first "Cut" the DESIGN you want to move. To cut a DESIGN you can either

- 1) click on the name of the DESIGN, then choose "Cut" command from the "Edit" menu, or
- 2) right-click (PC) or control-click (Mac) on the DESIGN name to bring up the contextual menu then select "Cut", or
- 4) use the keyboard shortcut ⌘X on MacOS, CTRL X on Windows.

You will know that the DESIGN is selected because a black dot will appear to the left of the DESIGN'S name, and the background of any SCRIPT cells in that column of the PROJECT window will become highlighted. A confirmation box will appear asking you to please confirm the deletion of the selected DESIGN. Press the "OK" button or the "Enter/Return" key to confirm the deletion. Then select the DESIGN you want to move the CUT DESIGN IN FRONT OF (insert CUT DESIGN to the left.) Then "Paste" the CUT DESIGN in.

To paste the a DESIGN you can either

- 1) click on the name of the **DESIGN** once to select it, then choose the “Paste” command from the “Edit” menu, or
- 2) right-click (PC) or control-click (Mac) on the **DESIGN** name to bring up the contextual menu then select “Paste”, or
- 3) use the keyboard shortcut **⌘V** on MacOS, **CTRL V** on Windows.

**KNOWN BUG #2.** The “Cut”, “Copy”, and “Paste” commands do not allow **SCRIPTS** to be moved along with their associated **DESIGNS**. You will need to re-demonstrate any **SCRIPTS** associated with **DESIGNS** that are moved via the method described above.

## 4.10 Zoom a DESIGN

You may zoom in and out on a **DESIGN** to make working with it easier. The zoom commands are available:

- 1) in the “Modify” menu,
- 2) in the zoom box at the bottom center edge of a **DESIGN** window,
- 3) from the **DESIGN** background’s contextual menu,
- 4) The keyboard shortcuts for 

<b>Mac OS X</b>	<b>Windows</b>
“Zoom In”	<b>⌘ +</b>
“Zoom Out”	<b>⌘ -</b>
“Normal Zoom”	<b>⌘ =</b>
“Zoom to Fit”	<b>⌘ /</b>

The “Normal Zoom” command makes the contents of a **DESIGN** window (all **FRAMES**) revert to the default size.

The “Zoom to Fit” command makes the contents of the **DESIGN** window (all **FRAMES**) exactly fit the available space in the window.

## 4.11 Bring a DESIGN Window to the Front

You can activate and bring an open **DESIGN** window to the front, from any window, by

- 1) double-clicking on its name in the Project window, or
- 2) selecting its **PROJECT** name in the Window menu and then selecting the **DESIGN** window’s name in the drop down list.

In Figure 22, the “New York Guide” **DESIGN** window in the “Tutorial Project” **PROJECT** is being brought to the front. The **PROJECT** window “Tutorial Project” is the only **PROJECT** open and “New York Guide” is the only **DESIGN** window open.

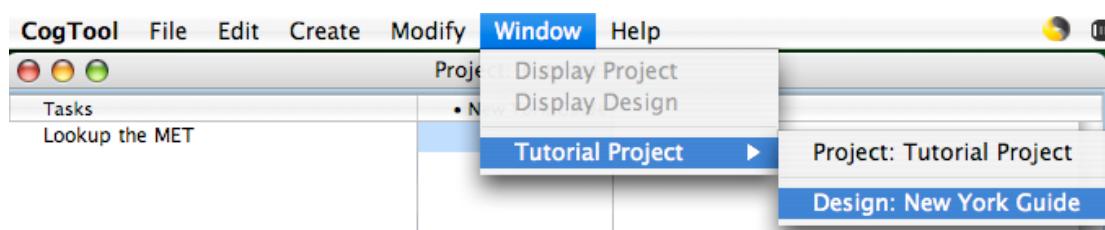
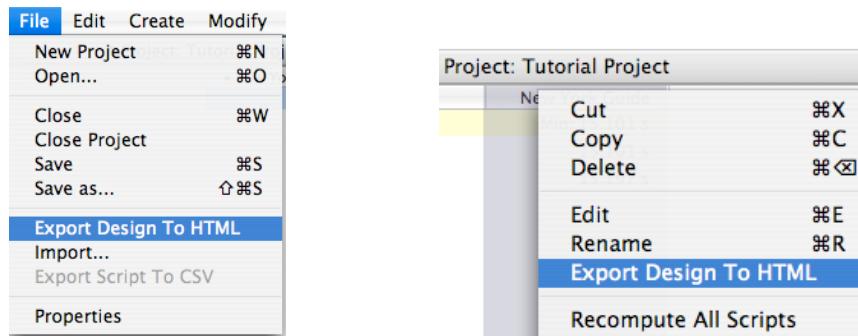


Figure 22. Bringing the New York Guide DESIGN window to the front.

## 4.12 Export a DESIGN to HTML for use in User Testing

You may want to use your completed DESIGN for purposes other than human performance modeling. For example, you might use it to communicate your design ideas to other members of the project team or to do user tests. CogTool lets you export your DESIGN to HTML to use for other purposes.

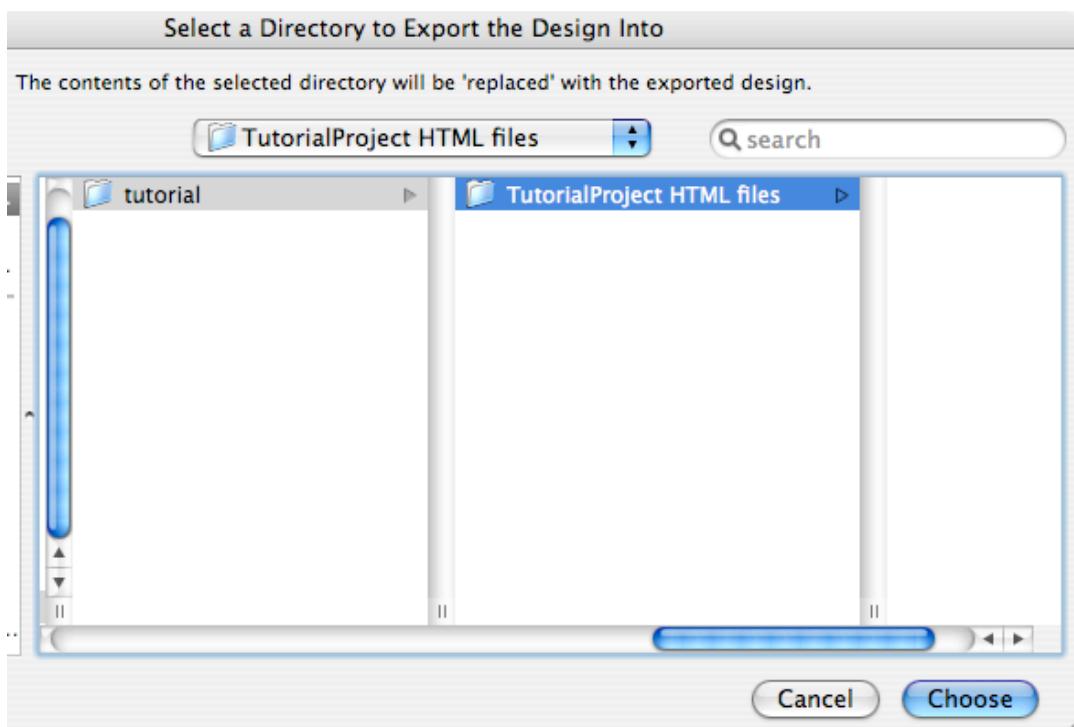
To export to HTML, first select a DESIGN in the PROJECT window or make a DESIGN window active. Then select the “Export Design To HTML” item in the File menu or in the contextual menu from the DESIGN name. (Figure 23)



**Figure 23. Export Design to HTML in the File menu on left. On right, in the contextual menu from the DESIGN name.**

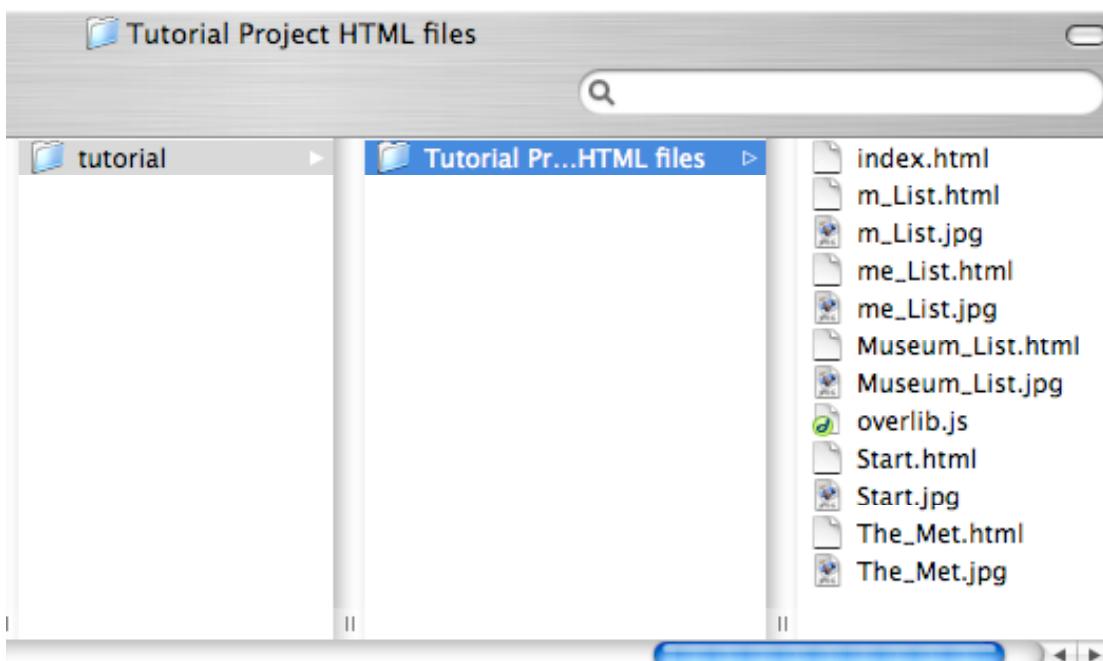
You will get a standard folder-selection dialog box where you can navigate to the desired place in your file hierarchy, or create a new folder in which to place the HTML files. After creating a new folder or selecting an existing one, press the “Choose” button to start the export process. (Figure 24)

**NOTE #3.** If you do not create a new folder and just select an already existing one, the contents of the folder selected will be REPLACED with the exported design.



**Figure 24. The “Export Design to HTML” dialog box. The exported files WILL REPLACE any files currently in the folder!**

Two files will be created for each FRAME, one will be an HTML file and the other will be an image in .jpg format. An index.html file will also be created that lists all the frames in alphabetical order. (See Figure 25)



**Figure 25. Files created by “Exporting Design to HTML” command**

The images will appear in each HTML file and the **WIDGETS** that you created in CogTool will be available as links that transition to the HTML file associated with the **FRAME** that results from that **TRANSITION** in CogTool. The HTML **WIDGET** will not be visible in the HTML files because for user tests, you do not want to give the user hints about what actions will accomplish the task you have asked them to do. For the same reason, the cursor will not change when a user moves it over the **WIDGET** in the HTML files. If the user clicks on a **WIDGET**, the types of actions available on that **WIDGET** appear as a pop-up menu and the user can select which one to perform. If the user clicks on an area of the image that does not have a **WIDGET** associated with it, a message pops up saying, “This item/object will not help accomplish your goal”.

The HTML will be sufficient for users to complete the **TASK** you defined in CogTool, according to the specific **SCRIPT** you demonstrated in CogTool. Of course, users are likely to attempt actions other than the script, and will get the message above, but these “errors” give you information about what other paths users may attempt. If multiple paths through an interface can accomplish a task, then you can augment your **DESIGN** in CogTool with as many alternative paths as you wish, and users will be able to follow those paths in the HTML.

Thus, CogTool provides a “reusable” **DESIGN** that is as useful for user testing with novices as it is for predicting skilled performance time.

## 4.13 Contextual menus available from a DESIGN window

There are 3 different contextual menus available in the DESIGN window. These menus are accessed by right-click (PC) or control-click (Mac) on a certain area within the window.

- 1) Background (see Figure 26) – clicking anywhere on the yellow area of the background

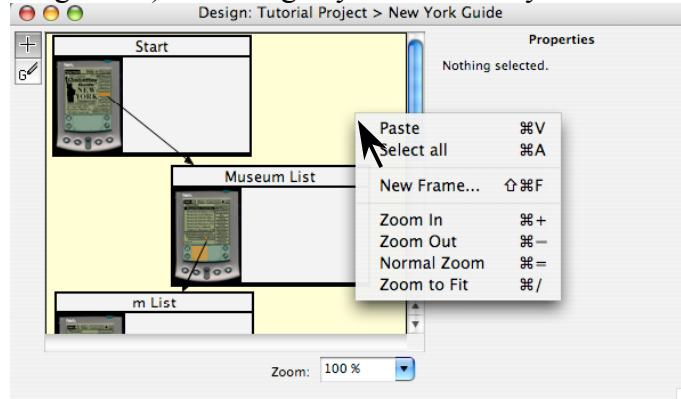


Figure 26. Background contextual menu in the DESIGN window.

- 2) Frame (see Figure 27) – clicking within a Frame.

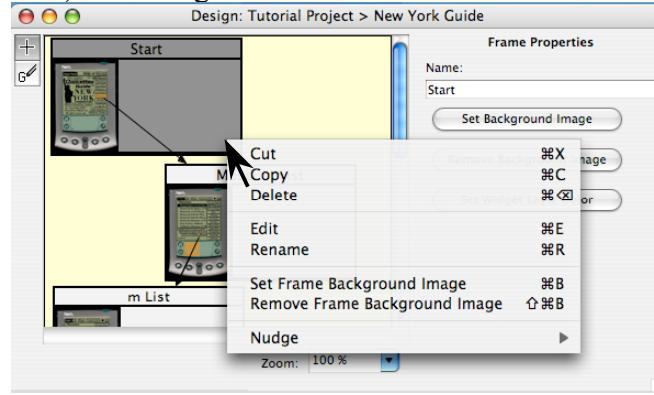


Figure 27. Frame contextual menu in the DESIGN window.

- 3) Transition (see Figure 28) – clicking on an arrow (transition).

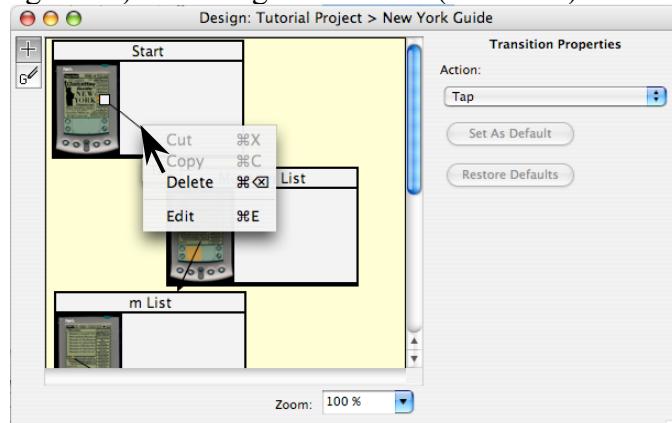


Figure 28. Transition contextual menu in the DESIGN window.

## 5 Edit FRAMES

### 5.1 What is a FRAME?

A **FRAME** represents a single screen of a user interface. Multiple **FRAMES** can be linked together to form a complete **DESIGN** storyboard. **FRAMES** may contain background images (section 5.11) and **WIDGETS** (section 6) to represent the look and interactivity of the interface you are designing. **FRAMES** will be connected by **TRANSITIONS** (section 7) from their **WIDGETS** to other **FRAMES**.

### 5.2 FRAME windows

The title bar of a **FRAME** window begins with the label “Frame:”. (See Figure 29) The areas of a **FRAME** window are:

- a toolbar down the left edge that contains all the widget types available
- the Interface representation area.
- Widget properties – name, label, and type
- options available for widgets
- a Zoom Box

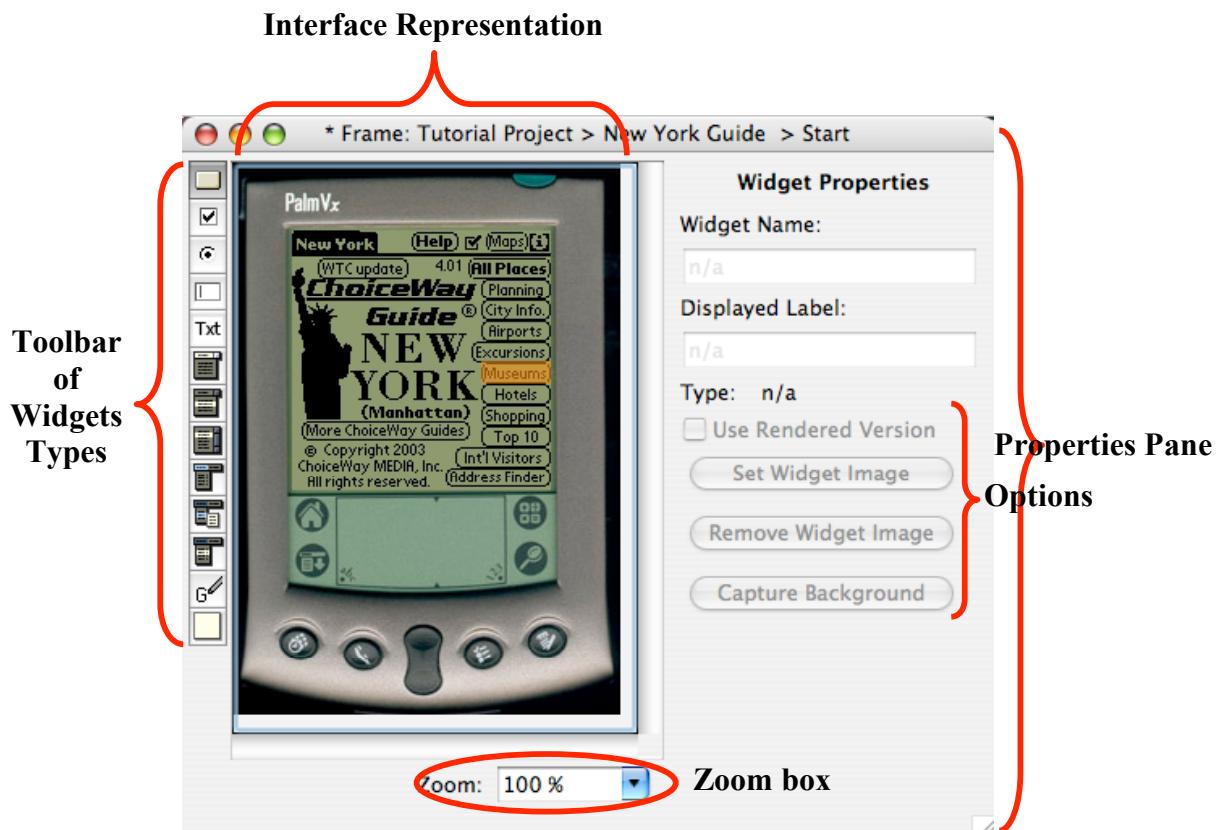


Figure 29. FRAME window “Start” with one WIDGET named “Museums” Type: Button.

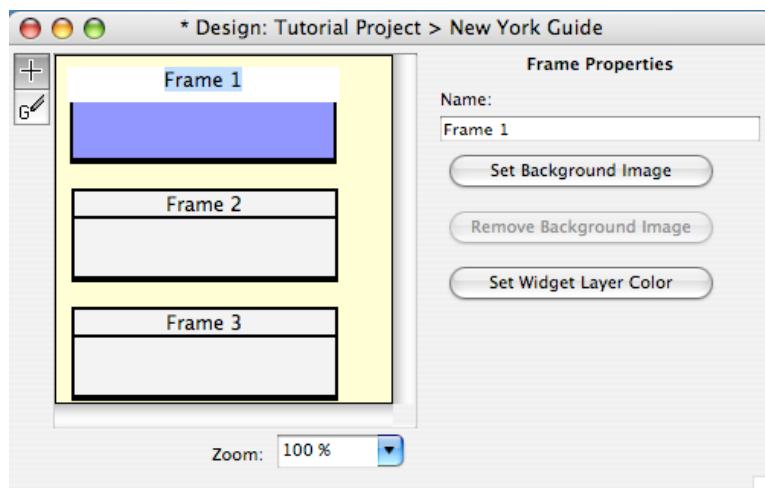
### 5.3 Create new FRAMES

When a **DESIGN** window opens, if no frames have been created, a new **FRAME** is automatically created.

You can create additional **FRAMES** by using either

- the “New Frame” command in the “Create” menu or
- “New Frame” in the **DESIGN** background’s contextual menu, or
- the keyboard shortcut  $\text{⌘F}$  on Mac OS X,  $\text{CTRL F}$  on Windows.

You may wish to rearrange the **FRAMES** in the **DESIGN** window to make the storyboard more visually pleasing. You can do this by simply pressing the left mouse button inside the center rectangle of a **FRAME**, dragging the mouse to a new location, and releasing the mouse button. A storyboard consisting of blank **FRAMES** might look like Figure 30.



**Figure 30.** Three empty **FRAMES**. **FRAME 1**’s default name is highlighted and ready to be replaced by a more meaningful name.

**KNOWN BUG #3.** When **FRAMES** are stacked, the stacking order is mysterious and may even seem to change over time. Work around: move the **FRAMES** apart so they are not stacked.

### 5.4 Rename a FRAME

To change the name of an existing **FRAME**, you can

- 1) use the “Rename” command in the “Edit” menu, or
- 2) use “Rename” in the **FRAME**’s contextual menu.
- 3) You may also double-click the top area of the **FRAME**, where the name is located. Or use
- 4) the keyboard shortcut  $\text{⌘R}$  on Mac OS X,  $\text{CTRL R}$  on Windows.

The light blue highlight around the name indicates that it is ready to be renamed.(Figure 30).

A name may be used only once in a particular **DESIGN**. After you type a name for the **FRAME**, press the “Enter/Return” key or click in the window, to confirm it. (Figure 31).

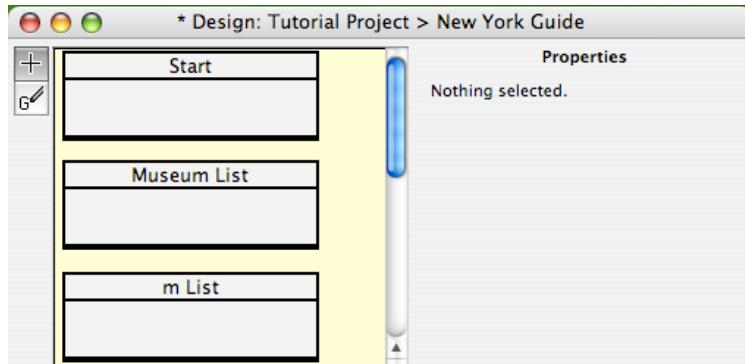


Figure 31. Default names have been replaced with meaningful names.

## 5.5 Edit a FRAME

FRAMES must contain information about the interface in order for CogTool to make predictions. To enter the necessary information open a **FRAME** for editing by

- 1) double-clicking on the rectangular body of the **FRAME** you wish to edit. (Be careful not to double-click on the top area, which allows you to edit the name of the **FRAME**.)
- 2) You may also click the **FRAME** to select it, and then choose the “Edit” command from the “Edit” menu, or
- 3) choose “Edit” from the **FRAME**’s contextual menu.
- 4) the keyboard shortcut **⌘E** on Mac OS X, **CTRL E** on Windows.

The **FRAME** window shown in Figure 32 is ready to be edited with a background image and **WIDGETS**.

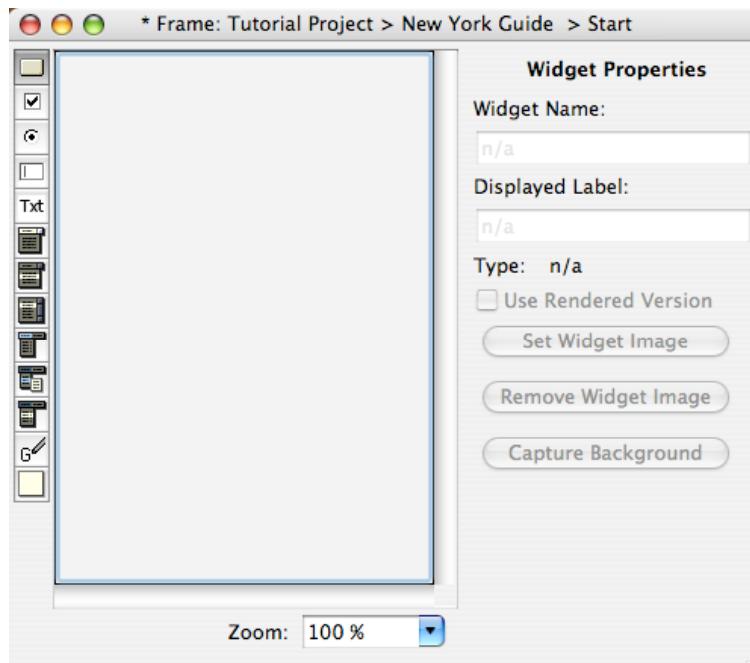


Figure 32. The **FRAME** editing window for an empty **FRAME** named “Start”.

## 5.6 Background Image of a FRAME

### 5.6.1 Set the Background Image of a FRAME

FRAMES may have background images. This is particularly useful when building a storyboard based on design drawings that have been made into images, or when baselining the performance of an existing interface and you have screen captures available. To select a background image for a FRAME, make the FRAME active in the DESIGN window then either:

- choose the “Set Frame Background Image” command from the “Modify” menu, or
- press the “Set Background Image” in the Frame Properties of the DESIGN window. (Figure 33), or
- choose the “Set Frame Background Image” command from the FRAME’S background contextual menu . Or
- use the keyboard shortcut  $\text{⌘B}$  on Mac OS X,  $\text{CTRL B}$  on Windows.

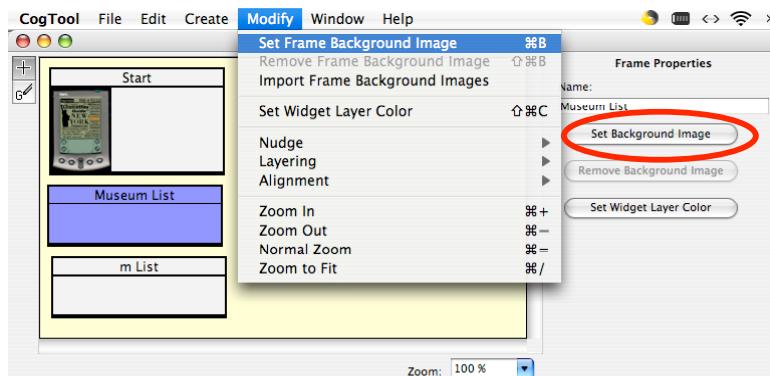


Figure 33. The “Set Frame Background Image” command in the “Modify” menu is highlighted in blue and the “Set Background Image” button is circled in red.

Then choose the desired image from the “Open File” dialog box that appears. Acceptable formats for a background image include JPEG, PNG, GIF, TIFF, and BMP. When confirmed, the background will appear in the FRAME window (Figure 34).

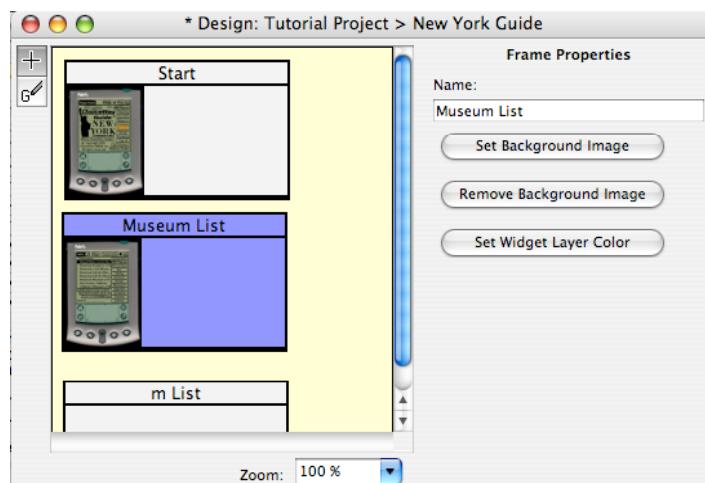


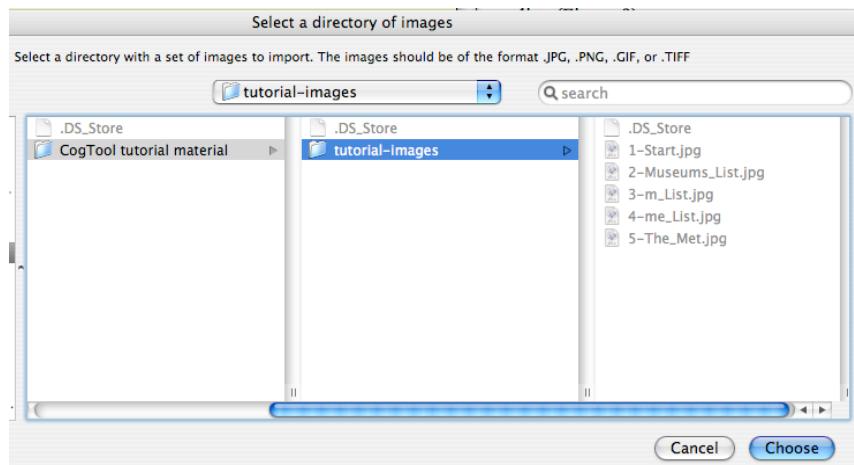
Figure 34. The “Museum List” FRAME now has a background image.

## 5.6.2 Import FRAME Background Images

If you already have many images that you want to use for FRAME backgrounds, and they're in an acceptable format (JPG, JPNG, or GIF) you can import an entire folder of images at once. Each image will come in as a FRAME with the image as its background and the file name as the FRAME's title.

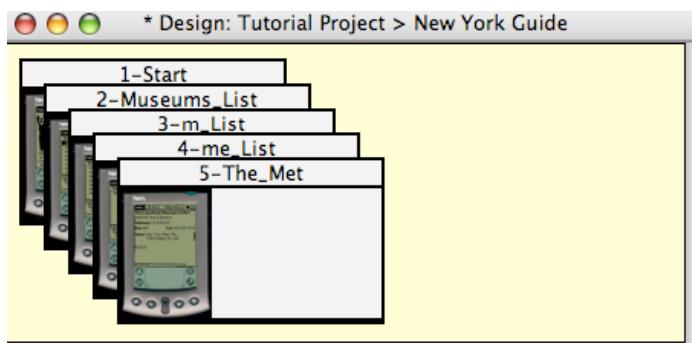
To import backgrounds you must be in the DESIGN window, select the Modify menu then choose the "Import Frame Background Images" command. A standard folder-selection dialog box will appear. Navigate to the folder that contains the images, click on the folder, then click the "Choose" button.

Figure 35 shows the folder "tutorial-images" being selected. This folder contains 6 files, 5 of which are .JPG format.



**Figure 35.** "Select a directory of images" dialog box, which lists the acceptable formats: **JPG, JPNG, or GIF.**

In order to see the frames fully after they've been created by "Import Background Images" it may be necessary for you to move the frames apart (Figure 36). See section 5.11 on how to reposition a frame.



**Figure 36.** The "New York Guide" Design window after 5 frames had been created using the "Import Background Images" command.

### **5.6.3 Remove the Background Image from a FRAME**

To remove the background image from a FRAME that currently contains one, you can either

- choose “Remove Background Image” from the “Modify” menu, or
- press the “Remove Background Image” button in the properties section of the DESIGN window, or
- choose “Remove Background Image” from the FRAME’s background contextual menu.
- Or use the keyboard shortcut  $\text{⌘}\text{F}$  on Mac OS X,  $\text{CTRL}\text{F}$  on Windows.

The image will be removed immediately.

## **5.7 Close the FRAME Window**

To close a FRAME window, you may use

- the standard close box on the title bar of the window, or
- choose the “Close” item from the “File” menu, or
- choose “Close” from the FRAME’s background contextual menu.
- the keyboard shortcut  $\text{⌘W}$  on Mac OS X,  $\text{CTRL W}$  on Windows.

**The DESIGN window will update and reflect the changes made to the FRAME**

## **5.8 Delete a FRAME**

To delete a Frame, first select the Frame you wish to delete by clicking on it once. You will know that the Frame is selected because it will be highlighted in blue. Then you can either

- 1) select the “Delete” command from the “Edit” menu, or
- 2) choose “Delete” from the FRAME’s contextual menu. Or use
- 3) the keyboard shortcut  $\text{⌘⌫}$  on Mac OS X,  $\text{CTRL⌫}$  on Windows

A confirmation box will appear asking you to confirm the deletion of the selected Frame. Click the “OK” to confirm.

## **5.9 Zoom a FRAME**

You may zoom in and out on a FRAME to make working with it easier. This is often necessary when using a large background image. The zoom commands are available

- in the “Modify” menu,
- in a zoom box at the bottom center edge of a FRAME window, or
- in the contextual menu for the FRAME window, or
- The keyboard shortcuts for 

<b>Mac OS X</b>	<b>Windows</b>
<b>“Zoom In”</b>	$\text{⌘+}$
<b>“Zoom Out”</b>	$\text{⌘-}$
<b>“Normal Zoom”</b>	$\text{⌘=}$
<b>“Zoom to Fit”</b>	$\text{⌘/}$

The “Normal Zoom” command makes the contents of a FRAME revert to the default size. The “Zoom to Fit” command makes the contents of a FRAME, including the background image; exactly fit the available space in the window.

## 5.10 Bring a FRAME window to the Front

If a **FRAME** window is already open but behind other windows, you can bring it to the front by selecting its **PROJECT** in the Window menu, then selecting the **FRAME** window you want in the drop down menu that lists all windows open in the following order: the **PROJECT** window is first, then the **DESIGN** windows, **FRAME** windows, then **SCRIPT** windows.

In Figure 37, the only **PROJECT** window open is “Tutorial Project”. The “New York Guide” **DESIGN** window is the only **DESIGN** open. There are three **FRAMES** open: “New York Guide > Museum List”, “New York Guide > m list”, and “New York Guide > Start”. And “New York Guide > Lookup the MET” is the only **SCRIPT** window open.

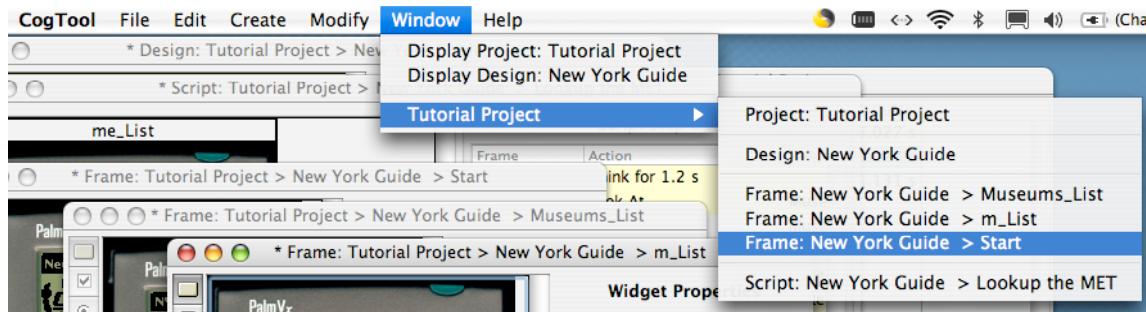


Figure 37. Bringing to front the window “Frame: New York Guide> Start”.

## 5.11 Change a FRAME’S Position in the DESIGN

To rearrange the **FRAMES** in the **DESIGN** window press the left mouse button inside the center rectangle of a **FRAME**, then dragging the mouse to the new location, and releasing the mouse button.

## 5.12 Contextual menus available from a FRAME window

There are 2 different contextual menus available in the **FRAME** window. These menus are accessed by right-click (PC) or control-click (Mac) on a certain area within the window.

- 1) Background – clicking anywhere in the background area that is NOT covered by a **WIDGET**.
- 2) **WIDGET** – clicking in an area defined by a **WIDGET**.

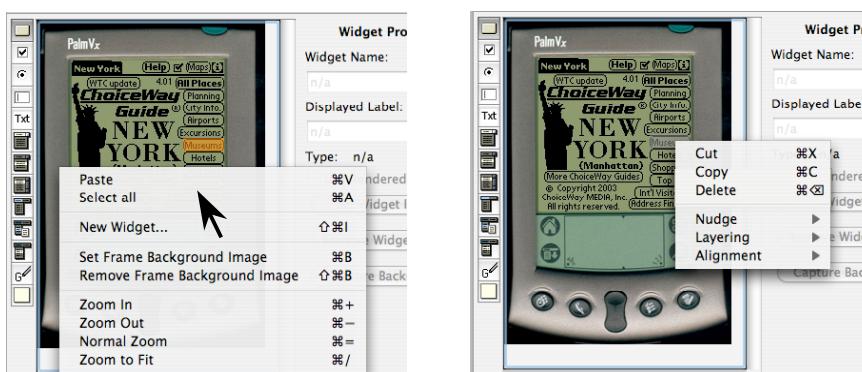


Figure 38. On the left the Background contextual menu within the FRAME  
On the right the Widget contextual window within the Frame.

## 6 Define WIDGETS

### 6.1 What is a WIDGET?

A **WIDGET** represents an interactive element on a **FRAME**. Define a **WIDGET** for each interface element that the user will interact with when demonstrating a **TASK** (section 9) e.g. buttons, check boxes, radio buttons, menus, displays, etc.. There is no limit to the number of **WIDGETS** that can be placed on a **FRAME**. In general, there is no need to create **WIDGETS** for every element of the original interface only for those needed for the **TASKS** you want to investigate.

**WIDGETS** appear as colored boxes over the background image. Figure 39 shows a **WIDGET** placed over the word “Museum” on the background. The white corner handles show that the **WIDGET** has been selected.

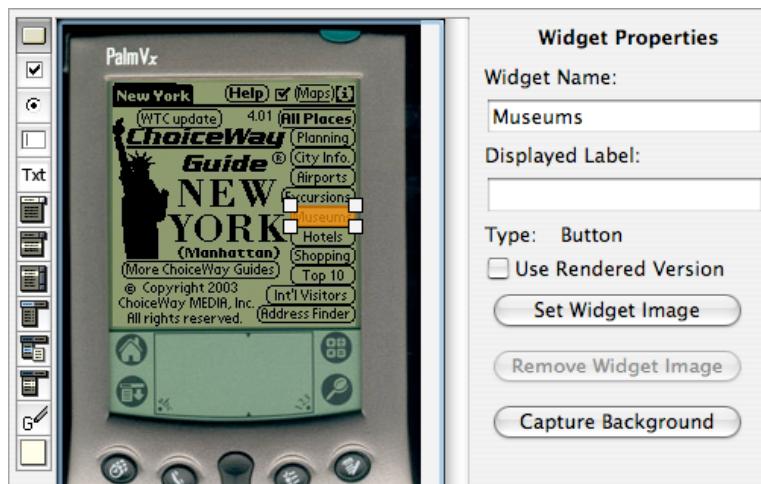


Figure 39. A **WIDGET** has been placed over the image of the Museums button. The text in the “Widget Name” field is currently selected.

**WIDGETS** interacted with during a task demonstration must have **TRANSITIONS** assigned to them in the **DESIGN** window. **TRANSITION** may link to a new **FRAME** (Figure 40) or not. See section 7 for further detail about assigning **TRANSITIONS**.

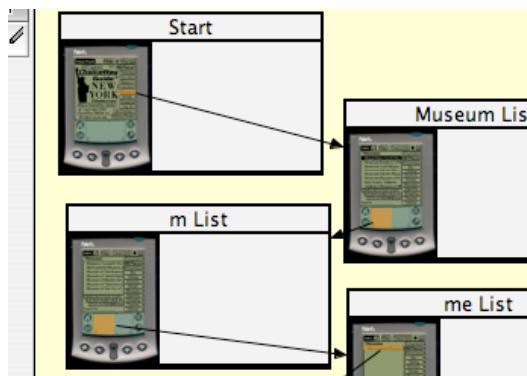


Figure 40. Black arrows represent which frames a widget links to.

## 6.2 Create a WIDGET

To create a **WIDGET**:

- 1) Select a **WIDGET** type from the toolbar along the left side of the **FRAME**'S window. This **WIDGET** type will be the default type for all future **WIDGETS** until a new tool is chosen.

**NOTE #4.** Once a widget has been created, its type cannot be changed. To change a **WIDGET**'S type you must create a new **WIDGET** then delete the previous one. See section 6.2.1 for further instructions.

- 2) Insert a **WIDGET** by either dragging a rectangle across an area of an open **FRAME** or by using the “New Widget” command (by selecting it in the “Create” menu, selecting it from the **FRAME**'S background contextual menu, or using the keyboard shortcut  $\text{⌘}\text{I}$  on Mac OS X,  $\text{CTRL}\text{I}$  on Windows). The “New Widget” command places a square **WIDGET** at the upper left corner of the frame. (When you have two actions in a row that act on the same interface “widget” it is important for the two CogTool **WIDGETS** to be placed and sized identically. The easiest way to ensure that two **WIDGETS** have identical positioning is to create the **WIDGET** on one **FRAME**, copy it, and then paste it onto all other **FRAMES** where the **WIDGET** appears. See Section 6.3.)
- 3) Figure 41). You can then move the **WIDGET** to the appropriate position and resize as necessary. (See section 6.7 for how to move a **WIDGET** and section 6.11 for how to resize **WIDGETS**.) The **WIDGETS** properties will then appear to the right of the window. A default name is inserted into the **WIDGET** Name field; we suggest that you replace this default with a meaningful name.



**Figure 41.** The **WIDGET** on the left was made by selecting a desired area on the **FRAME**. The **WIDGET** on the right was made using the “New Widget” command in the “Create” menu. It will need to be repositioned and resized.

**NOTE #5.** When you have two actions in a row that act on the same interface “widget” it is important for the two CogTool **WIDGETS** to be placed and sized identically. The easiest way to ensure that two **WIDGETS** have identical positioning is to create the **WIDGET** on one **FRAME**, copy it, and then paste it onto all other **FRAMES** where the **WIDGET** appears. See Section 6.3.

- 4) Each **WIDGET** must have a unique name in the “**WIDGET Name**” field; we strongly advise that you type in a meaningful name (as opposed to “Widget\_1”), as these names are used to refer to the widgets in other areas of CogTool. Figure 42 shows where a **WIDGET** has been placed on the **FRAME’S** background image. A **WIDGET** can also have its own background image, which can come from an external file or be captured from the **FRAME’S** background. The following sub-sections contains more information concerning setting these properties.



**Figure 42.** The default settings for a newly created **WIDGET**.

### 6.2.1 Set the Widget’s Name

When you create a **WIDGET**, its properties appear to the right of the window. A default name is inserted into the **Widget Name** field, but we advise that you replace it with a meaningful name.. Each **WIDGET** on a **FRAME** must have a unique name made up of alphanumeric characters. You may use spaces and punctuation like dash and underscore in a **WIDGET** name.



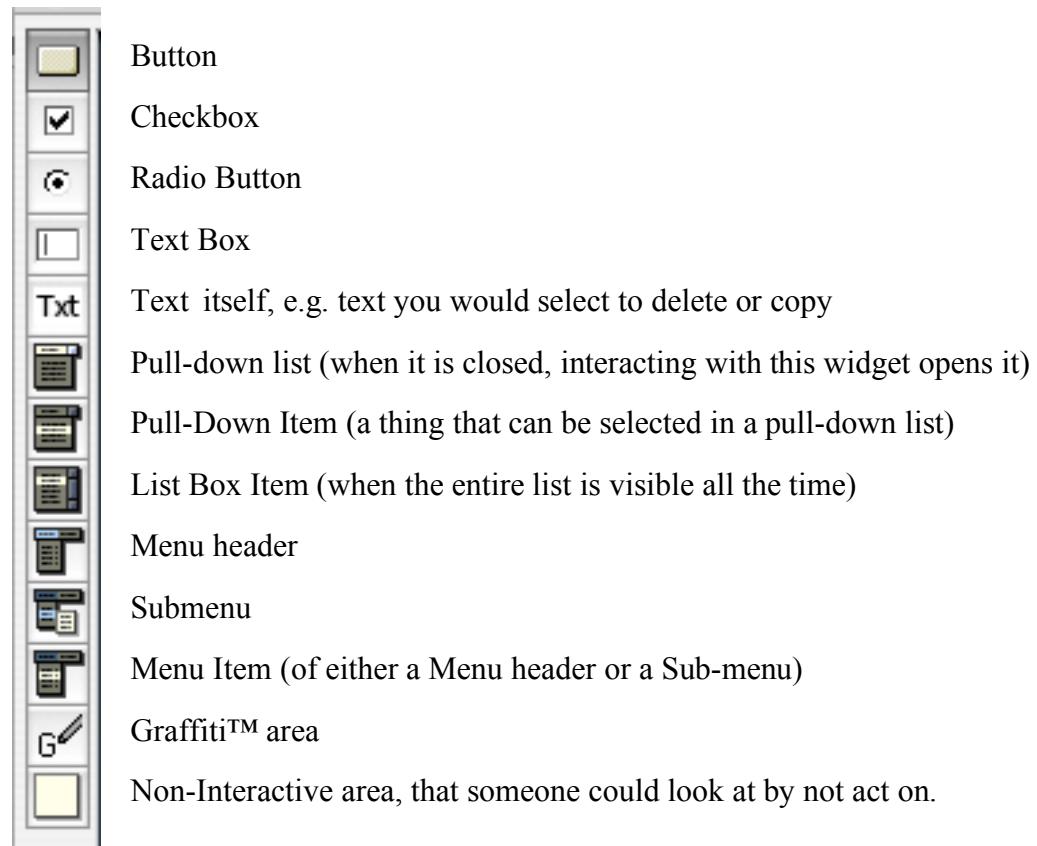
**Figure 43.** The **Button** widget type icon was selected to create the **WIDGET**,which has the unique widget name “**Museum**”.

### 6.2.2 Select the Widget’s Type

A **WIDGET**’s type is assigned by selecting a icon from the toolbar along the left edge of the Frame’s window. Assigning the appropriate type to each **WIDGET** is critical for CogTool to produce valid predictions of user behavior. Therefore, please refer to section 6.13 for a full discussion of **WIDGET** types.

**NOTE #6.** Once a widget has been created, its type cannot be changed. To change a **WIDGET**’s type you must create a new **WIDGET** then delete the previous one. Please refer to section 6.13 for a full discussion of **WIDGET** types so you can choose the right type for each **WIDGET** you create.

**KNOWN BUG #4.** Tool tips do not work on the toolbars for some versions of the Macintosh.



**Figure 44. Types of WIDGETS and the tools that insert them. See section 6.13 for a full discussion of WIDGET types**

### 6.2.3 Set the Widget's Label to Be Displayed

If the background image for a FRAME does not provide enough information to identify the purpose of a WIDGET, you may provide a “Displayed Label”. This text will appear centered within the highlighted area of the WIDGET whenever a FRAME is displayed. Type the text you wish to appear within the “Displayed Label” text box.

### 6.2.4 Set the Widget's Image

Like FRAMES, WIDGETS can have an image. This can be useful when creating a new layout that is not based on an existing system. To set the image for a WIDGET, first select it by left-clicking on it. Next, click the “Set Widget Image” button. A standard Open dialog will appear. Select the image you would like to use. Acceptable formats for a image include JPEG, PNG, GIF, TIFF, and BMP. Once you have confirmed your file selection, the image should appear underneath the highlight for the selected WIDGET.

### 6.2.5 Capture the Image for a Widget from the Frame's background

You may choose to set the image for a **WIDGET** to match the background image of the **FRAME** beneath it. This will allow you to move the **WIDGET** to a new area of a **FRAME** while keeping a consistent image for the **WIDGET**. To do this, select the **WIDGET** by left-clicking on it, and then press the “Capture Background” button at the bottom of the **FRAME** window. See Figure 45.

This feature can be used to create totally new layouts from one screenshot of a **FRAME**. Simply create a **FRAME** with all the desired **WIDGETS** and capture the background for each one. Copy each **WIDGET** and paste it onto a **FRAME** with a blank or neutral background and move the **WIDGETS** wherever you want them to be. You can also use the “Capture Background” function, and then remove the image from the **FRAME**, leaving only the **WIDGETS** behind.

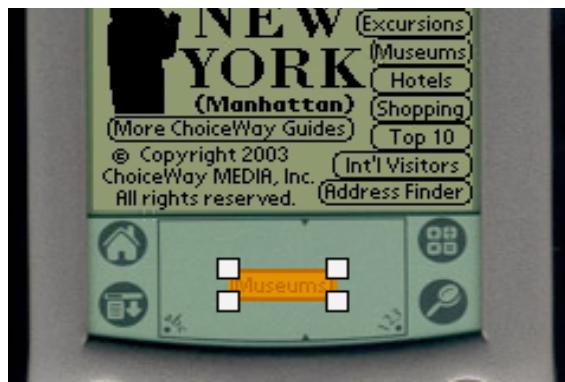


Figure 45. The image of the “Museums” button **WIDGET** has been captured and the **WIDGET** has been moved to the center of the Graffiti® area.

### 6.2.6 Remove the Image from a Widget

To remove the image for a **WIDGET**, first select the **WIDGET** by left-clicking on it. Next, press the “Remove Widget Image” button. The image for the **WIDGET** will be immediately removed. Note that this function will not remove the image of the **FRAME** behind a **WIDGET**, if there is one.

## 6.3 Copy a WIDGET from one frame to another

Select the **WIDGET** by clicking on it in the **FRAME** window. You will know that the **WIDGET** is selected because square, white resize handles will appear at the corners of the **WIDGET**,

**NOTE #7.** If you paste a **WIDGET** into the same **FRAME** it was copied from, the new **WIDGET** will pasted DIRECTLY ON TOP OF THE ORIGINAL ONE and you won't be able to see the original **WIDGET**. Move the newly-pasted **WIDGET** off of the original to be able to see both.

**NOTE #8.** If the same interface **WIDGET** appears in different **FRAMES** in a **DESIGN**, it is **important** for the CogTool that the **WIDGETS** be **placed and sized identically**. The only way to ensure that two **WIDGETS** have identical positioning is to create the **WIDGET** on one **FRAME**, copy it, and then paste it onto all other **FRAMES** where the **WIDGET** appears. This becomes important for correct prediction of human performance when there are several actions in a row on the same widget (e.g., clicking the “back” button or a “page down” button several times in a row).

and the properties panel at the bottom of the window will show the current options selected for that **WIDGET**. Then select the “Copy” command from the “Edit” menu or the **WIDGET**’s contextual menu, or use the standard keyboard shortcut for Copy. Go to the frame you want to paste the copied **WIDGET** into then select “Paste” from the “Edit” menu or the **WIDGET**’s contextual menu, or the standard keyboard shortcut for Paste. The new **WIDGET** will be placed at the same location as the original **WIDGET**. To move the new **WIDGET** click-and-drag it to a new location and resize if desired.

## 6.4 Rename a WIDGET

To rename a **WIDGET**, first select it by clicking on it in the **FRAME** window. You will know that the **WIDGET** is selected because square, white resize handles will appear at the corners of the **WIDGET**, and the properties panel to the right of the window will show the current options selected for the **WIDGET**. The current name of the **WIDGET** will be highlighted in the **WIDGET** name: text field ready for you to type in the new desired name. Once you click out of the text field, or hit ENTER/RETURN the new name will be set.

## 6.5 Delete a WIDGET

To delete a **WIDGET**, first select it by clicking on it in the **FRAME** window. You will know that the **WIDGET** is selected because square, white resize handles will appear at the corners of the **WIDGET**, and the properties panel at the bottom of the window will become enabled. To delete the **WIDGET**, after selecting it, you can either:

- select the “Delete” command from the “Edit” menu, or
- select the “Delete” command from the **WIDGET**’s contextual menu. Or
- the keyboard shortcut  $\text{⌘⌫}$  on Mac OS X,  $\text{CTRL⌫}$  on Windows.

A confirmation box will appear asking you to confirm the deletion of the selected **WIDGET**. Confirm by selecting the “OK” button.

## 6.6 Duplicate a WIDGET within the same frame

Select the **WIDGET** by clicking on it in the **FRAME** window. You will know that the **WIDGET** is selected because square, white resize handles will appear at the corners of the **WIDGET**, and the properties panel at the bottom of the window will show the current options selected for the **WIDGET**. Then either

- 1) select the “Duplicate” command from the “Edit” menu. Or use
- 2) the keyboard shortcut for  $\text{⌘D}$  on Mac OS X,  $\text{CTRL D}$  on Windows.

An identically sized **WIDGET** will appear to the lower right of the original **WIDGET**. This **WIDGET** will have the same properties as the original, but will be given a new unique name.

## 6.7 Move a WIDGET

Move a **WIDGET** by pressing with the left mouse button on the translucent area of the **WIDGET** and dragging the mouse to the new location. Release the mouse button when the **WIDGET** is positioned where you would like it.

## 6.8 Nudge WIDGETS into place

To help position a **WIDGET** precisely in a Frame, the widget can be nudged into place by either:

- using the arrow keys on the keyboard while holding the **⌘** (Mac OS X) or **CTRL** key (Windows). Or
- selecting the “Nudge” command from the Modify menu. Or
- selecting the “Nudge” command from the **WIDGET**’s contextual menu.

## 6.9 Overlapping WIDGETS

As in other drawing applications, objects can be drawn on top of each other. In CogTool, this means that **WIDGETS** can partially or even fully occlude other **WIDGETS**. Drawing overlapped **WIDGETS** has implications for defining **TRANSITIONS** (Section 7.2) and demonstrating **SCRIPTS** (Section 9.7). As in other drawing applications, you can change the depth of the overlapped **WIDGETS** with standard commands Bring to Front, Bring Forward, Send Backward and Send to Back. These commands are available by selecting the Layering item in the Modify menu in the FRAME window, or from the contextual menu available when you select a **WIDGET**.

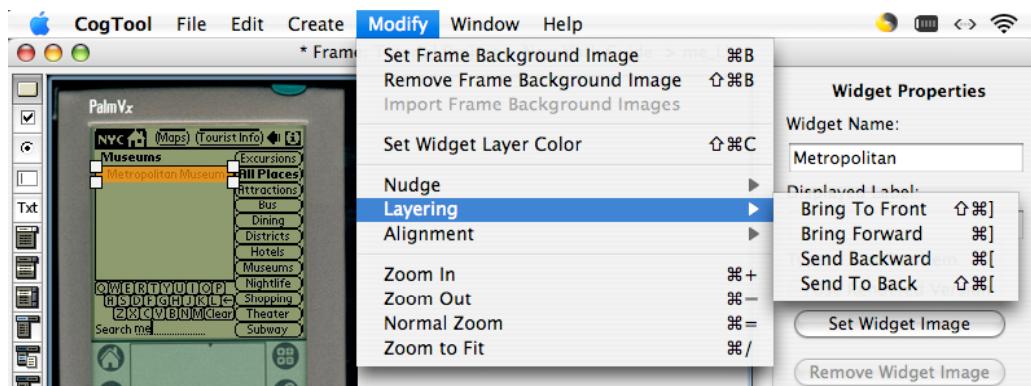


Figure 46. Layering menu item for moving layered WIDGETS.

## 6.10 Aligning WIDGETS

To help position a **WIDGET** precisely on the background the **WIDGET** can be aligned or spaced out with other **WIDGETS** selected. To align or space out, select multiple **WIDGETS** either by shift clicking on each **WIDGET** or by dragging a bounding box around the desired **WIDGETS** (i.e., holding down the (Mac OS X) SHIFT **⌘** or (Windows) SHIFT **CTRL** keys while dragging the mouse). Then select the modification to be used in the “Alignment” command in the “Modify” menu or in the **WIDGET**’s contextual menu.

## 6.11 Resize a WIDGET

Resize a **WIDGET** by clicking and dragging the small square handles at the corners of a selected **WIDGET**. The highlighted area will change to indicate the new size.

## 6.12 Set WIDGET Layer Color

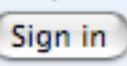
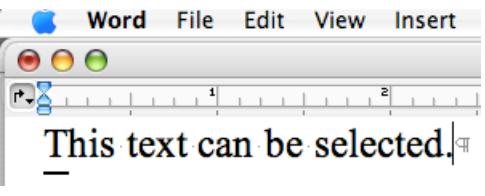
By default, **WIDGETS** are orange. However, orange may not be visible in all **FRAMES**, i.e., if the **FRAME'S** background image is orange. Therefore, you can change the color of all **WIDGETS** in a **FRAME**. Choose the “Set **WIDGET** Layer Color” command in either the:

- Modify Menu
- by pressing the “Set **WIDGET** Layer Color” button
- or us the keyboard shortcut  $\text{⌘}\text{C}$  on Mac OS X,  $\text{CTRL}\text{ }\text{⌘}\text{C}$ .

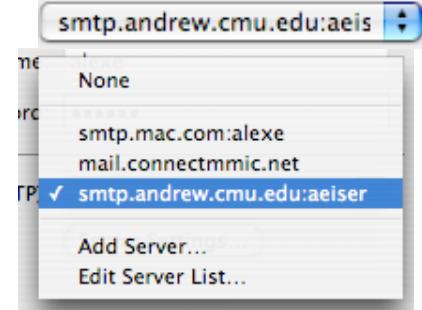
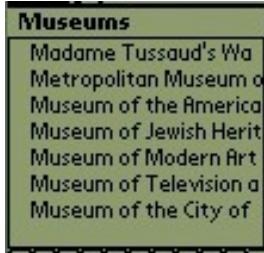
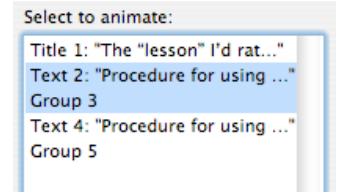
A standard color wheel dialog box will appear for you to choose the desired color. All **WIDGETS** on the **FRAME** will be changed.

## 6.13 Choosing the Appropriate WIDGET Type

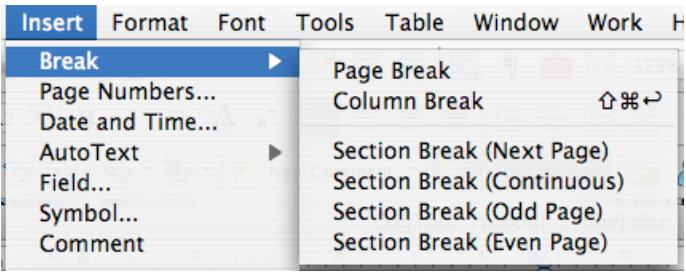
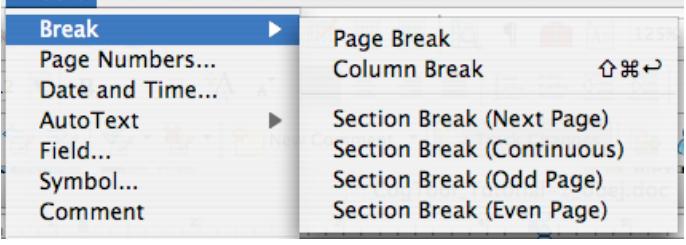
Properly marking the type of a **WIDGET** is necessary for CogTool to make accurate predictions. The type of a **WIDGET** is based on its function in the interface. Different values are used for different types of user interface elements. See Table 1 for a list of the **WIDGETS** supported by CogTool and guidelines for when to use each **WIDGET**. These **WIDGET** types are selected in the Frame window from the toolbar on the left side of the window.

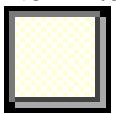
Widget	Examples	When to use this Widget
Button	   	Any time a widget looks like a button as defined by current conventions. The button can be a hardware button (like the Palm® Datebook button to the left), a static button (like the calculator button), or a button that appears dynamically on a screen (like the New and Sign In buttons).
Check Box	 <b>Private:</b> <input type="checkbox"/> <input checked="" type="checkbox"/> Allow anonymous posts. <input type="checkbox"/> Allow author to modify message after posting <input type="checkbox"/> Allow author to remove own posted messages <input checked="" type="checkbox"/> Allow file attachments. <input checked="" type="checkbox"/> Allow new threads.	Any time there is an interactive box that can be toggled on or off (checked or not checked) independently of any other check boxes. The top example on the left is a single checkbox from a Palm® calendar entry setting. The bottom example is a set of related checkboxes encountered when setting up a discussion forum on Blackboard®.
Radio Button	  	Any time there is a set of visible options where (1) the options remain visible after clicking on one, (2) the state of the option is visible and remains visible, and (3) only one option can be selected at a time. These often have circles beside them (like the text style selections to the left) that show the state of the options, but may also take other shapes (like the days of the week in the Palm® Date Book, to the left).
Text Box	 <b>Every: ..... 1 Day(s)</b> Username: <input type="text"/> Password: <input type="password"/>	Any time there is a place to enter text (letters or numbers). These can have different looks, like the text area for the number of days on the Palm® or the Username and Password boxes on Gmail® (to the left).
Text	 	Any text that can be selected and copied, deleted, pasted over, or typed over. It can be the text in a textbox, the text someone is editing in a word processor, text on the web that you want to copy, etc. The widget goes directly over the text you want to manipulate. For example, if you select a large amount of text by double-clicking the first word of the passage and shift-clicking the last word of the passage, you would place a text widget on the first word and the last word because these would be the targets of your mouse actions.

*Table continued on next page.*

Pull-Down List	 	Any time there is a widget where a list drops down in response to the click (or tap) on the displayed item, only one item on the list can be selected, and the selected item replaces the original displayed item as the label of the list. Use the Pull-down List <b>WIDGET</b> for the displayed item (before the list is pulled down). An example is the list of types of ToDo items to display on the Palm®. In the example on the far left, the “All” with the down arrow indicates a pull-down list; if this were the image in the <b>FRAME</b> , the arrow+All would be a Pull-down List <b>WIDGET</b> . In the example from Apple’s Mail, the object with the address+arrows would be a Pull-down List <b>WIDGET</b> .
Pull-Down Item	 	Use the Pull-Down Item <b>WIDGET</b> for the items in the pull-down list that could be selected after the list is pulled down. In the example on the left from Palm®, if this were the image in the <b>FRAME</b> , each of the items would be Pull-down Item <b>WIDGETS</b> . Likewise, all the items in Apple’s Mail pull-down would be Pull-down Item <b>WIDGETS</b> .
<b>Using Pull-Down Lists and Items:</b>	When demonstrating a task, you can <i>only</i> interact with a Pull-Down Item <b>WIDGET</b> <i>after</i> having pulled down the list by interacting with its associated Pull-Down List <b>WIDGET</b> . (This makes sense because a user cannot interact with a list item until they are visible and they are not visible until the list is pulled down.) If you demonstrate a task that interacts with a Pull-Down Item <b>WIDGET</b> without first interacting with the associated Pull-Down List Widget, CogTool will crash.	
List Box Item	  	When there is a visible list of items (which does not have to be pulled down or popped up) from which one or more can be selected. The Museums list to the left is from the ChoiceWay™ Guide to New York City; the animation list box is from the “Custom Animation” dialog in PowerPoint™.

*Table continued on next page.*

Menu			The Menu <b>WIDGET</b> is used for the menu headers. In the Palm® menu to the left, “Record”, “Edit” and “Options” would be Menu <b>WIDGETS</b> . In the Microsoft Word™ menu below it, “Insert”, “Format”, and all the words across the top would be Menu <b>WIDGETS</b> .
Submenu			The Sub Menu Widget is used for menu items that produce another menu. The Palm® menu to the left does not have a Sub Menu widget. The Microsoft Word™ menu on the left would have two Sub Menu <b>WIDGETS</b> , “Break” and “AutoText”.
Menu Item			The items that can be selected from a menu (which do NOT produce other menus) are Menu Item <b>WIDGETS</b> . In the Palm® menu to the left, “Undo”, “Cut” and all the words below them would be Menu Item <b>WIDGETS</b> . In the Microsoft Word™ menu below it, “Page Break”, “Column Break”, and all the words below them would be Items.
<b>Using Menu, Submenu and Menu Item:</b> When demonstrating a task, you can <i>only</i> interact with a Submenu Item <b>WIDGET</b> or a Menu Item <b>WIDGET</b> <i>after</i> having interacted with its associated Menu <b>WIDGET</b> . (This makes sense because a user cannot interact with a submenu or menu item until they are visible and they are not visible until the menu is pulled down.) If you interact with a Sub Menu Item or a Menu Item <b>WIDGET</b> without first interacting with its associated Menu <b>WIDGET</b> , CogTool will crash.			
Graffiti			The Graffiti® <b>WIDGET</b> is used to designate an area on a touchscreen that can interpret Graffiti® strokes. To mock-up a Palm interface, we suggest using one Graffiti® <b>WIDGET</b> on the left side called Graffiti_Letters and one on the right side called Graffiti_Numbers because the user must target one or the other side when using a Palm. Other areas of touch screen can also receive Graffiti® strokes and can be designated with Graffiti® <b>WIDGETS</b> .
<i>Table continued on next page.</i>			

Non-Interactive			A non-interactive <b>WIDGET</b> is one where a user might have to look to get information, but cannot manipulate that information from that <b>WIDGET</b> . For example, to the left is a picture of part of the Flight Management System of a Boeing 777 airplane. The Altitude display (currently displaying the number 10,000) would be defined as a non-interactive <b>WIDGET</b> . The task could involve looking at that <b>WIDGET</b> , but the display cannot be changed directly (the knob below it would have to be defined as an interactive <b>WIDGET</b> and separately manipulated).
<i>End of Widget Selection Table.</i>			

## 7 Frames via TRANSITIONS

### 7.1 What is a TRANSITION?

After you have created the **FRAMES** and **WIDGETS** of the storyboard for your **DESIGN**, you must show CogTool how those **FRAMES** should be linked together into a multi-step **TASK**. In CogTool, the links between **FRAMES** are called **TRANSITIONS** and are represented by black arrows in the **DESIGN** window. (Figure 47) A **TRANSITION** records how a particular action on a particular **WIDGET** causes the storyboard to flip from one **FRAME** to another. The next section will explain how to draw **TRANSITION** from a **WIDGET** to a **FRAME** and how to associate a **TRANSITION** with a particular user action.

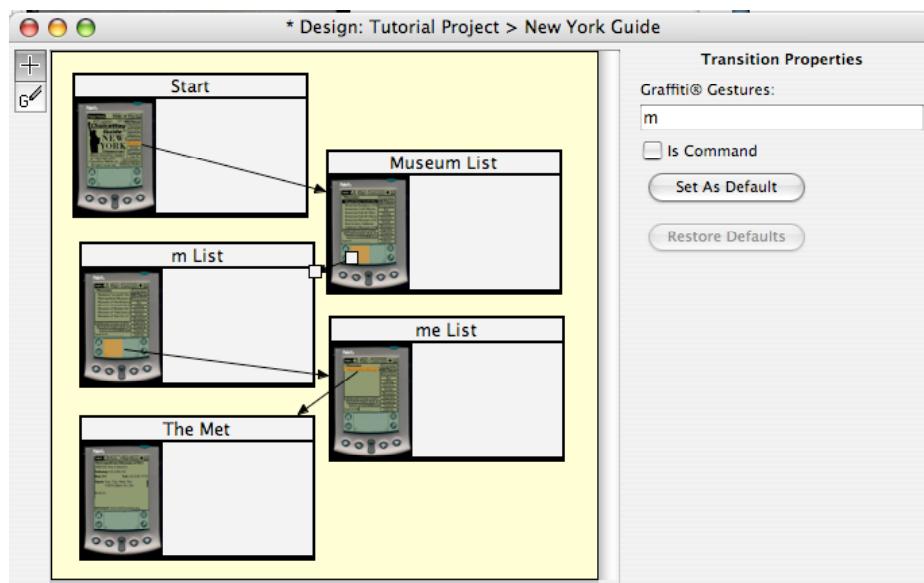


Figure 47. Frames in the Design: Tutorial Project > New York Guide are linked together by TRANSITION arrows.

### 7.2 Create a TRANSITION

TRANSITIONS are created in the DESIGN window. A TRANSITION can go from a WIDGET to a different FRAME, or from a WIDGET to the same FRAME (a *self-transition*).

To create a TRANSITION from a WIDGET to a different FRAME, first select the appropriate TRANSITION type from the toolbar along the left edge. The icons available indicate which input devices are applicable for this DESIGN as specified by the Devices selected as “used in the design” in the dialog box that appeared when the new Design was created. (See Figure 4 on page 12)

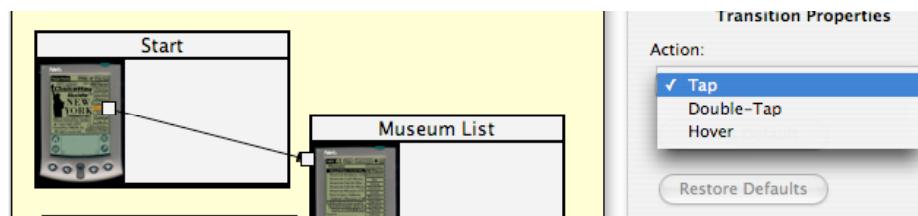
The available transition types are:

- = default for the type of widget from which the TRANSITION emanates
- = mouse
- = keyboard
- = touch screen

In the vast majority of cases, you will want the default type of transition for the **WIDGET**, and will leave the top tool (default) selected.

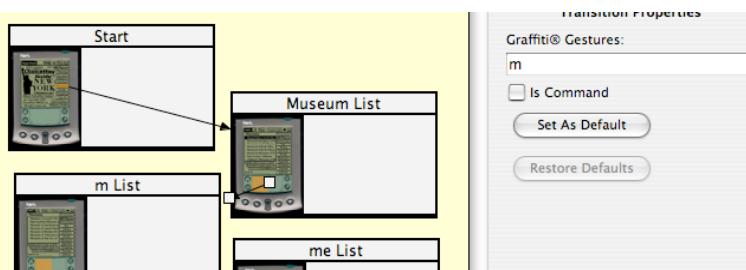
In a rare cases, you may want to specify a transition other than the default transition. For example, if you have both a touch screen and a keyboard available as devices in your **DESIGN**, and a Graffiti® **WIDGET** on a **FRAME**, the default transition for that **WIDGET** would be the touch screen. However, it is possible that a user could move the mouse to that area and use the mouse to make a Graffiti® gesture. If you want to evaluate a task using that strategy, you would select the mouse tool before drawing a transition from the Graffiti® **WIDGET**.

When the desired tool is selected, press the left mouse button on the **WIDGET**, drag the mouse to the destination **FRAME** until it highlights (turns gray), and then release the mouse button. An arrow will appear from the **WIDGET** to the destination **FRAME**, showing the **TRANSITION** you just created. (Figure 48)



**Figure 48** Creating a transition by dragging from the “Museums Button” WIDGET in the “Start” FRAME to the “Museums List” FRAME

To create a **SELF-TRANSITION** from a **WIDGET** to the same **FRAME**, press the left mouse button on the **WIDGET**, drag the mouse just outside of the **WIDGET** but still in the same frame, and then release the mouse button. An arrow will appear from the **WIDGET** to the edge of the **FRAME**, showing the **SELF-TRANSITION** you just created. In Figure 49 a **SELF-TRANSITION** is selected.



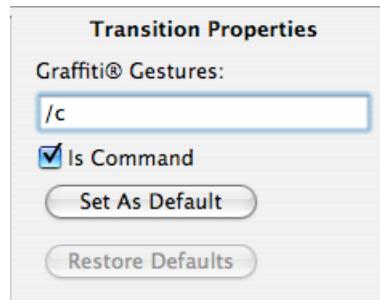
**Figure 49** A self-TRANSITION arrow is selected. The arrow was created by dragging the mouse from inside the WIDGET to just outside the WIDGET.

The mouse cursor changes to a crosshair when it is hovering over a **WIDGET**, to help you see when the cursor is pointing at a small **WIDGET**. If you have trouble clicking on **WIDGETS** because they are too small, use the “Zoom” commands in the “Modify” menu or from the “Zoom” box located at the bottom center of the **DESIGN** window.

If you have trouble clicking on a **WIDGET** because it is behind another **WIDGET**, there are two remedies. If the desired **WIDGET** is only partially occluded by the **WIDGET** in front of it, carefully select the edge of the desired **WIDGET** that is not covered by the second **WIDGET** and draw the **TRANSITION** from that edge. If the desired **WIDGET** is completely covered by the second **WIDGET** or the edge is too small to click on, change the order of the **WIDGETS** by selecting the second **WIDGET** and use the layering commands available in the Modify menu or the contextual menu (Section 6.9).

You are not allowed to create two **TRANSITIONS** from the same source with the same properties because it is impossible for a **DESIGN** to be in two different states at the same time. If you attempt to create two **TRANSITIONS** from the same source that have the same properties, an error message will appear, and you will have a chance either to select different properties or cancel the creation of the **TRANSITION**.

When you create a Graffiti® or Keyboard **TRANSITION**, CogTool prompts you for the text associated with the **TRANSITION**. The values you give to this dialog box show up in the Transition Properties Pane. If the keys or gestures you specify are a command to the system (like the keyboard shortcut for “copy” or the Graffiti® gesture for “paste”, as opposed to plain text), check the **Is-Command** checkbox in the Transition Properties Pane (Figure 50).



**Figure 50.** The **TRANSITION** Properties pane for the Graffiti(r) gestures that invoke the copy command. The gestures are "/c" and the **Is Command** checkbox is checked.

### 7.3 Delete a **TRANSITION**

To delete a **TRANSITION**, first select it by clicking on it in the **DESIGN** window. (You may find it easier to click on the head of the arrow rather than the line.) You will know that the **TRANSITION** is selected because square, white resize handles will appear at the endpoints of the arrow. Finally, choose the “Delete” command from the “Edit” menu, or the **TRANSITION**’s contextual menu. A confirmation box will appear asking you to confirm the deletion of the selected Transition. Press the “OK” to confirm.

### 7.4 Change the Source of a **TRANSITION**

You may change the source of a **TRANSITION** to any **WIDGET** (on any **FRAME**) that supports the action type of the **TRANSITION**. For example, only a Graffiti® **WIDGET** can be the source of a **TRANSITION** with a Graffiti® action type. First, select the **TRANSITION** by clicking on it. (You may find it easier to click on the head of the arrow rather than the line.) Drag the square, white resize handle at the current source to a new **WIDGET**. The mouse cursor will change to a crosshair when it is pointing at a **WIDGET** that can accept

the **TRANSITION**. When you release the mouse button, the **TRANSITION** will start at the new source **WIDGET**.

## 7.5 Change the Destination of a TRANSITION

You may change the destination of a **TRANSITION** to any **FRAME**. First, select the **TRANSITION** by clicking on it. (You may find it easier to click on the head of the arrow rather than the line.) Drag the square, white resize handle at the current destination to a new **FRAME**. When you release the mouse button, the Transition will point at the new destination **FRAME**.

## 7.6 Edit TRANSITION Properties

Like a **WIDGET**, a **TRANSITION** has properties that are initially set using the toolbar on the left and can be further configured in the panel to the right of the window. Unlike widgets, the type of a transition can be changed. To change the type of action represented by a **TRANSITION**, first select the **TRANSITION** by clicking on it. Then select the Edit command from the Edit Menu, or type apple-E (Mac) or cntrl-E (PC). A dialog box will appear with the types of transitions applicable to the devices associated with the Design. For example, a **DESIGN** with no keyboard will not have the option of a Keyboard action type, and a **DESIGN** with no Touchscreen will not have the option of a Graffiti® action type. Click the radio button for the type of action desired and complete any additional information asked for by CogTool. The properties pane will change to reflect the properties appropriate for that action type.

Once in the desired type is set, the possible actions are displayed in the Transitions Properties pane. The possible actions are as follows.

### Mouse

This action type represents acting on a particular on-screen location using a mouse cursor. Two properties can be configured: the “Mouse Button:” property represents the choice of buttons on a multi-button mouse (Left, Right, Middle) and the “Action:” property represents the physical motion (Figure 51). “Click” actions represent button actions on a mouse. “Hover” is when a mouse is used to position the cursor at a location, but not click at that location (sometimes called a “rollover”).

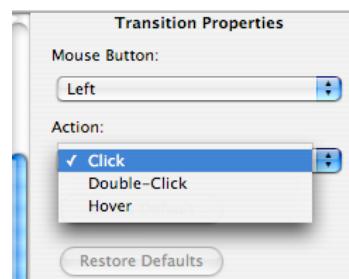


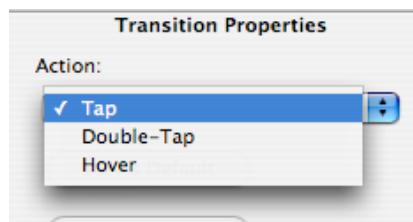
Figure 51. A Design using only a Mouse input device has three Action types to choose from: Click, Double-Click and Hover.

**KNOWN BUG #5.** In scripts, a hover action is supposed to appear as a yellow Look-at step, a yellow Move-Mouse step, and a white Hover step. Sometimes the hover step does not appear in the script, but the model still produces a correct prediction because the hover itself does not contribute time to the task (only the move-mouse contributes motor time).

### **Touchscreen(Graffiti®)**

This action type represents acting on a particular on-screen location using a finger or stylus. For “Graffiti®” touchscreens the action type represents gesture input using a stylus and the Palm Graffiti® recognition system.

For normal touchscreen devices the “Action:” property represents the physical motion. (See Figure 52) “Tap” actions correspond to physical finger taps on a touch-sensitive screen. “Hover” is when a cursor is moved to a location, but does not tap at that location; this is only meaningful for touchscreen hardware that can detect movements above the surface.

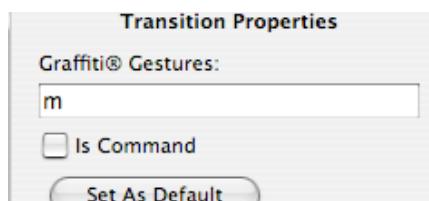


**Figure 52.** The transitions available for a DESIGN with a “Touchscreen” input device.

For “Graffiti®” touchscreens the “Graffiti® Gestures” represent the characters input via a stylus

**NOTE #9.** Hover only has meaning in touchscreen hardware that can detect hovering above the surface of the screen without touching the screen. E.g., some versions of SmartBoard technology can detect hovering; PDAs typically cannot.

and the Palm Graffiti® recognition system. The “Is Command” property distinguishes whether the gestures should be recognized as a command that triggers an active behavior in the system, or simple text input.

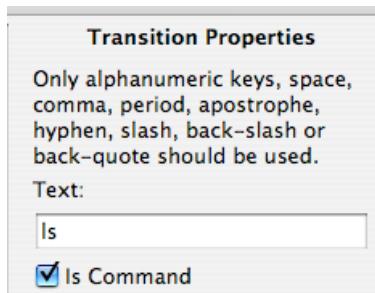


**Figure 53.** Graffiti® gestures are either text or commands. Both types are entered into the text box. Checking the checkbox to the left of “Is Command” indicates a command.

## Keyboard

This action type represents keyboard text input. The “Text” property represents the actual characters typed. The “Is Command” property distinguishes whether the key presses should be recognized as a command, triggering an active behavior in the system, or simple text input. The “Action” represents the physical motion that will trigger the Transition, e.g. a full “Press” or just the “Down” or “Up” portions of the motion. CogTool can only accept certain keystrokes for a keyboard transition; use only alphanumeric characters, space, comma, period, apostrophe, hyphen, slash and back-slash.

**NOTE #10.** In many applications that use Graffiti®, commands are preceded by a diagonal slash gesture, and you would enter “/” as the first character when you define this type of command, e.g., “/c” for Copy, “/v” for Paste. To allow designers to expand the definitions of commands in novel designs, CogTool 1.0b18 does not enforce this convention and any string can be defined as a command by checking the “is Command” box. If you want to use traditional Graffiti® commands, you must put both the “/” and check the box saying that the string is a command to ensure valid time predictions.



**Figure 54. Keyboard actions available. The “is Command” box should be checked when the text entry issues an explicit command to the computer. This figure shows how you would simulate entering the “ls” command to a UNIX terminal window.**

**NOTE #11.** CogTool currently only supports the “Press” motion (down AND up) and does not allow separate “Down” and “Up”. In the original KLM, holding down a shift or control key while pressing other keys was approximated by a press of the modifier key before pressing the other keys; CogTool follows this tradition. In addition, modifier keys (e.g. “Shift” or “Ctrl”) and special keys (e.g. Tab or Backspace) are also not currently supported. You can use a nearby key on the keyboard to simulate the press of an unsupported key. For example, to simulate “Tab”, tell CogTool to press the “q” key.

**NOTE #12.** Even though CogTool will allow you to enter a capital letter by holding the “shift” key when defining a keyboard transition, and it will appear as a capital letter in the text field and context menus, CogTool will not give valid time predictions because it will not account for the time to hit the “shift” key. Use a nearby key to simulate the press of the “shift” key. For example, to get an accurate prediction for “J”, tell CogTool to press “zj”; for “F” tell CogTool to press “/f”.

## 8 Create and Organize Tasks

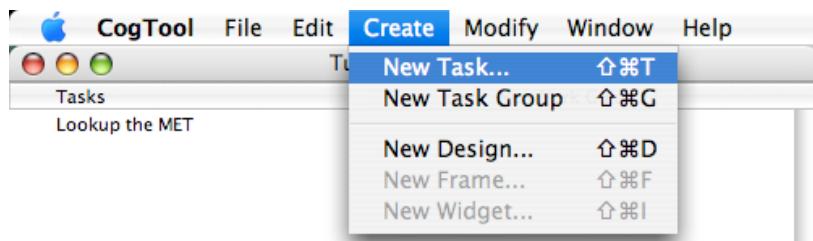
## 8.1 What are TASKS and TASK Groups?

CogTool uses the concept of a **TASK** to represent the purpose of a specific sequence of actions. A **TASK** has a name and can have a different **SCRIPT** of actions for each **DESIGN**. A **TASK GROUP** is used to organize **TASKS** and compute simple statistics about them

**NOTE #13.** The currently implemented methods for reorganizing **TASKS** and **TASK GROUPS** are quite awkward, so we suggest that you carefully plan the organization of **TASKS** before demonstrating any **SCRIPTS**. There is no ungrouping of **TASK GROUPS**. You can cut or copy and paste a **TASK** name, but if there is a **SCRIPT** associated with that **TASK** it will not copy and will have to be re-demonstrated.

## 8.2 Create a New TASK

To create a new **TASK** chooses the “New **TASK...**” command from the “Create” menu (Figure 55), or from the **DESIGN** window’s background contextual menu. You may also use the keyboard shortcut for this command **⌘T** on Mac OS X, **CTRL T** on Windows.



**Figure 55.** The “New Task...” command.

This creates a new row in the **PROJECT** window, ready to accept a name. The new **TASK** will appear immediately below the bottom-most selected **TASK** or **TASK GROUP**. You may name the **TASK** anything you wish, but there can only be one **TASK** with a particular name in each **TASK GROUP** or at the top level of a **PROJECT**. It is best to use names that express all of the unique features of a particular **TASK**, to contrast with similar **TASKS**. Try to name **TASKS** so that they will continue to apply as the **DESIGN** evolves over time.

**NOTE #14.**     **ONCE AGAIN**, the currently implemented methods for reorganizing **TASKS** and **TASK GROUPS** are quite awkward, so we suggest that you carefully plan the organization of **TASKS** before demonstrating any **SCRIPTS**. There is no ungrouping of **TASK GROUPS**. You can cut or copy and paste a **TASK** name, but if there is a **SCRIPT** associated with that **TASK** it will not copy and will have to be re-demonstrated.

### 8.3 Create a TASK Group

You can create a **TASK GROUP** by selecting the **TASKS** or other **TASK GROUPS** you wish to add to the new group, then choosing “New **TASK GROUP...**” from the “Create” menu, or from the **DESIGN** window’s background contextual menu. You may also use the keyboard shortcut for this command  $\text{⌘G}$  on Mac OS X,  $\text{CTRL G}$  on Windows.

Any **TASKS OR TASK GROUPS** that are selected (hold down the shift key while selecting all desired tasks) when the “New **TASK GROUP...**” command is chosen will be placed inside the new **TASK GROUP**. (Figure 56). There is no limit to the number of **TASK GROUPS** that may be nested within other groups.

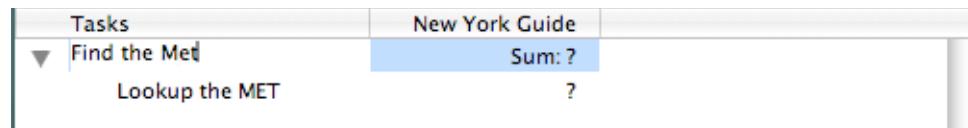


Figure 56. **TASK GROUP** named “Find the Met” contains one **TASK** “Lookup the MET”.

**NOTE #15.** **YET A THIRD TIME**, the currently implemented methods for reorganizing **TASKS** and **TASK GROUPS** are quite awkward, so we suggest that you carefully plan the organization of **TASKS** before demonstrating any **SCRIPTS**. There is no ungrouping of **TASK GROUPS**. You can cut or copy and paste a **TASK** name, but if there is a **SCRIPT** associated with that **TASK** it will not copy and will have to be re-demonstrated.

### 8.4 Rename a TASK or TASK Group

To rename a **TASK**, you may double-click its name. You may also select the **TASK** by clicking on its name once, and then choose the “Rename” command from the “Edit” menu. Both methods will cause the name to be replaced by an editable text field. When you have finished editing the name, press the “Enter/Return” key to accept the changes. You may name the **TASK** anything you wish, but there can only be one **TASK** with a particular name in each **TASK GROUP** or at the top level of a **PROJECT**.

### 8.5 Delete a TASK or TASK Group

To delete a **TASK OR TASK GROUP**, first select it by clicking on its name in the **PROJECT** window. You will know that the **TASK OR TASK GROUP** is selected because its row in the **PROJECT** window will have a highlighted background. Then choose the “Delete” command from the “Edit” menu. A “Confirm Deletion” box will appear, select the OK button. (Figure 57) **Deleting a Task Group will delete all tasks that are within that Group!**

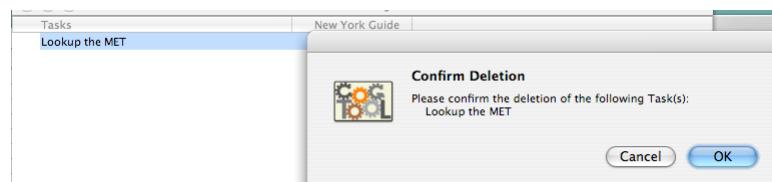


Figure 57. A **TASK** in the process of being deleted.

## 8.6 Change the Order of TASKS and TASK Groups

CogTool 1.0b18 does not provide a direct method for changing the order of **TASKS** in the **PROJECT** window. However, it is possible to rearrange **TASKS** using the “Cut” and “Paste” commands in the “Edit” menu. The “Paste” command will place any pasted **TASKS** below the bottom-most selected **TASK** or **TASK GROUP** in the **PROJECT** window.

**KNOWN BUG #6.** The “Cut”, “Copy”, and “Paste” commands in CogTool do not allow **SCRIPTS** to be moved along with their associated **TASKS**. You will need to re-demonstrate any **SCRIPTS** associated with **TASKS** that are moved via the method described above.

## 8.7 Change the Computed Value for a TASK Group

It is often useful to group **TASKS** together and apply simple computations. For example, you may wish to see the total time for a suite of benchmarks **TASKS**, or the minimum time among all methods of completing a **TASK**. Select the type of value you wish to compute for each group via the “Modify” menu (Figure 58).

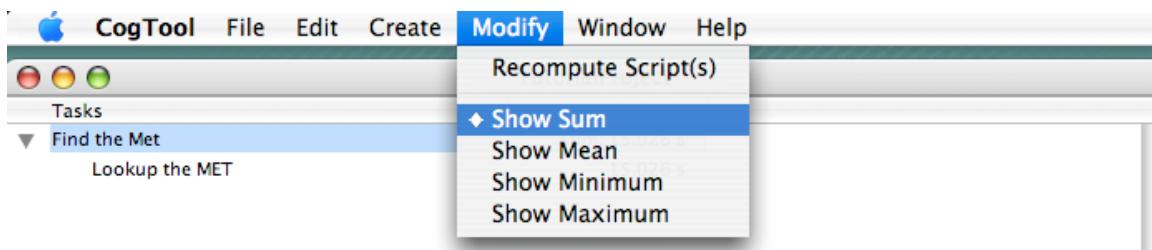


Figure 58. Use the “Modify” menu to set the value for the Task Group.

## 8.8 Organization Strategies for TASKS

CogTool can make predictions for many **TASKS** using the same **DESIGN**. It can be very useful to organize **TASKS** to simplify making use of the results.

### *Categorical Grouping*

The simplest arrangement of **TASKS** is a grouping by category of **TASK**, allowing all but the category in focus to be collapsed to minimum size. In this case, the computation type chosen is not important. If the categories are different aspects of a benchmark suite, “Show Sum” may be appropriate.

### *Task Breakdown*

In some cases, it is useful to break a long **TASK** into multiple segments so that each component can be analyzed separately. Choose “Show Sum” for this kind of **TASK GROUP**, so that the total time can be calculated automatically.

### *Alternate Methods*

In other cases, different **TASKS** represent alternative methods for achieving the same goal. Choose “Show Minimum” or “Show Maximum” to display the value of the fastest or slowest alternative. A **TASK GROUP** used in this way can be nested within another **TASK GROUP** representing a benchmark suite, so that the total benchmark will represent either a best or worst case analysis, respectively.

## 9 Record a SCRIPT

### 9.1 What is a SCRIPT?

In order to make a prediction about a Task, CogTool needs to know the specific sequence of user actions that compose it. In this section, you will learn how to record that sequence of actions, called a **SCRIPT**, by demonstrating the actions using the **DESIGN** storyboard described in Section 3.9.

**NOTE #16.** Before you start editing **SCRIPTS**, make sure you have carefully organized your **TASK** names. The currently implemented methods for reorganizing **TASKS** and **TASK GROUPS** are quite awkward and if there is a **SCRIPT** associated with a **TASK** it will not move with that **TASK** and will have to be redemonstrated.

### 9.2 SCRIPT windows

A **SCRIPT** window has 2 sections. The left section is devoted to the demonstration of the task. It shows the frame that is available for interaction and the bottom part of the section has three buttons that allow for special interaction timings to be inserted into the **SCRIPT**. The right section is where the steps that have been demonstrated are listed and there are two buttons in the bottom section, “Delete Step” and “Close and Compute”. (See Figure 59) The title bar of a **SCRIPT** window begins with the label “Script:”.

**List of demonstrated**

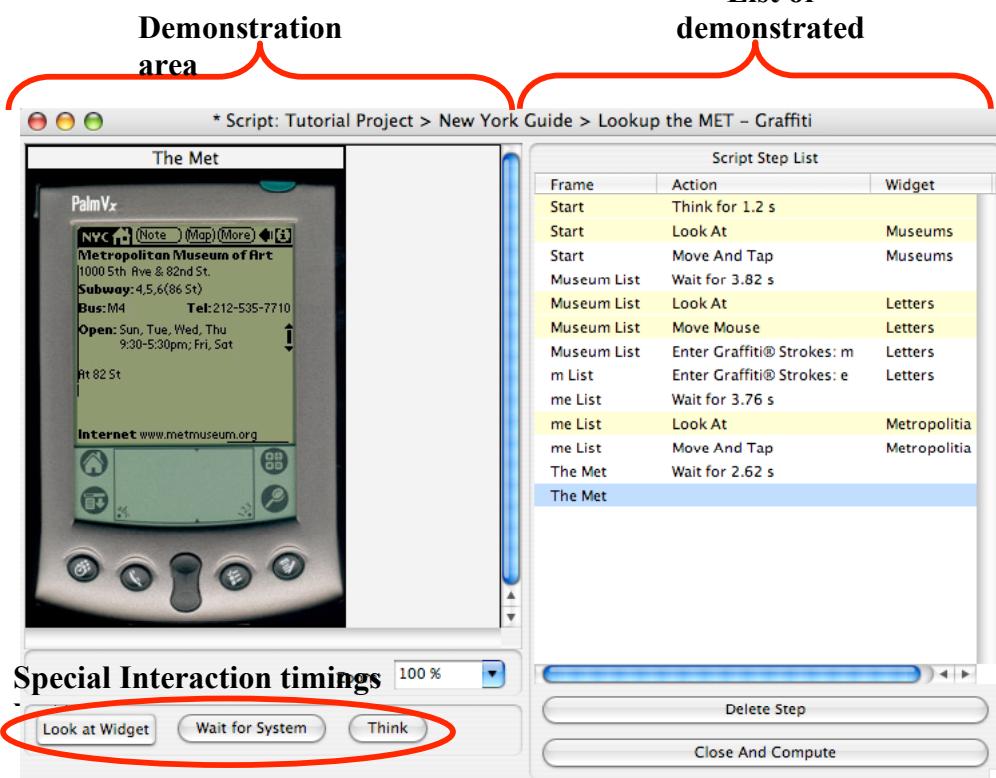


Figure 59. A **SCRIPT** window.

If the lines in a script ever appear highlighted in red (Figure 60), the script has become invalid. All steps in this script must be deleted and the script must be demonstrated again. This often happens when you have changed the mock-up after demonstrating the script. To be safe, always re-demonstrate scripts after changing a mock-up.

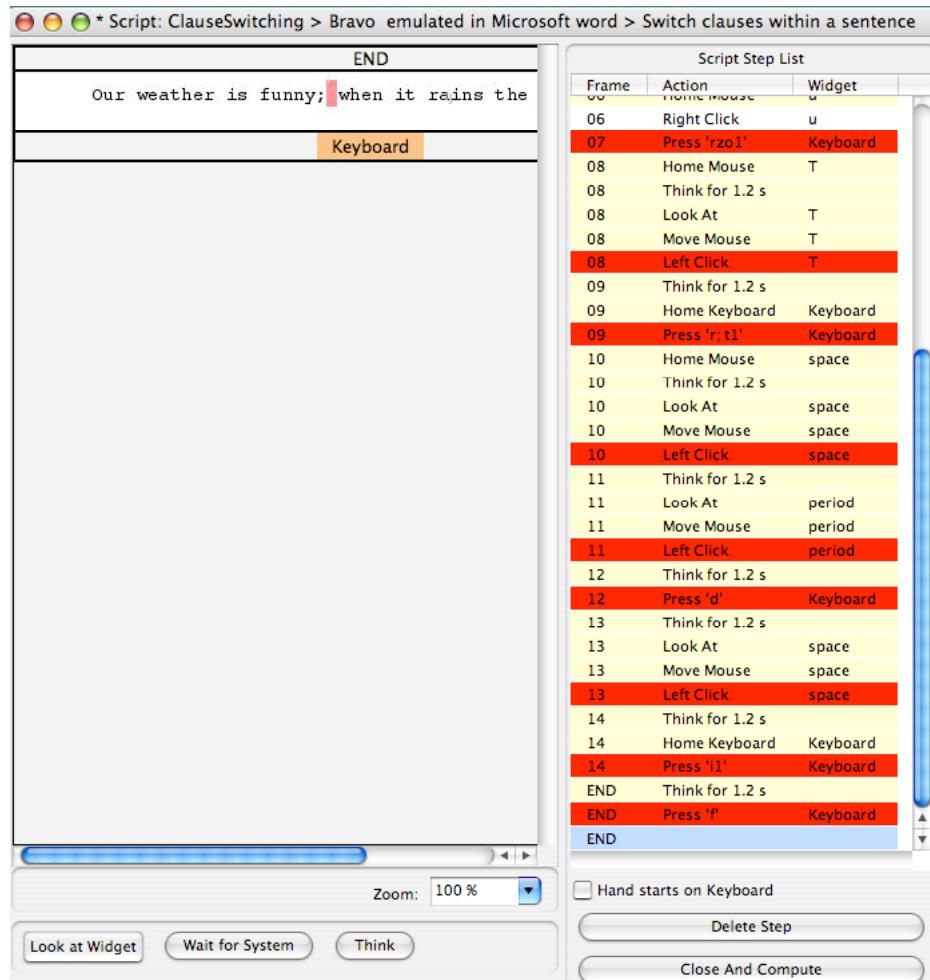


Figure 60. Red lines within a SCRIPT means that the steps are no longer valid.

### 9.3 Open a SCRIPT for Editing

To open the **SCRIPT** window for a **DESIGN**, you double-click in the **SCRIPT** cell under the **DESIGN** name and across from the **TASK** you want to demonstrate. You may also open the contextual menu for a cell by clicking the right mouse button. (On Mac OS X, you can also click while holding the CTRL key.) Then, choose the “Edit **SCRIPT**” command from the menu (Figure 61).

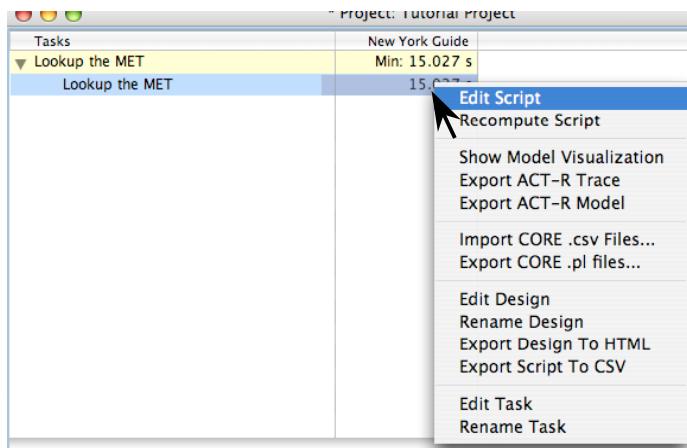


Figure 61. Open a **SCRIPT** to edit by using the contextual menu at the intersection of a **DESIGN** and **TASK**.

### 9.4 Choose a Start Frame for the SCRIPT

Since a **SCRIPT** may begin at any **FRAME** in the **DESIGN**, you must select the start **FRAME** separately for each **SCRIPT**. The first time you edit a particular **SCRIPT**, you will be prompted with the **FRAME** chooser window. To select the start **FRAME**, double-click on a **FRAME** or its name in the list at the left. You may also click the desired **FRAME** to select it, and then press the “Start Demonstrating Task” button or the “Enter/Return” key. The main demonstration window will replace the **FRAME** chooser window.

In the Figure 62 below, the first **FRAME** in the **TASK** happens to be named “Start”. In general, you may use any name you wish for the first **FRAME** of a **TASK**.

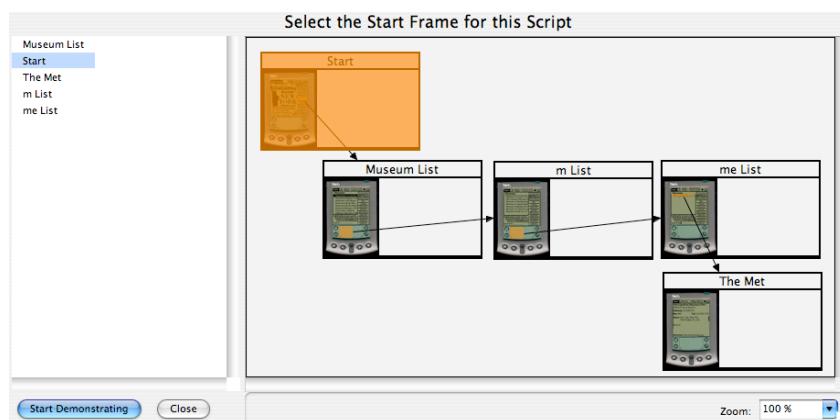
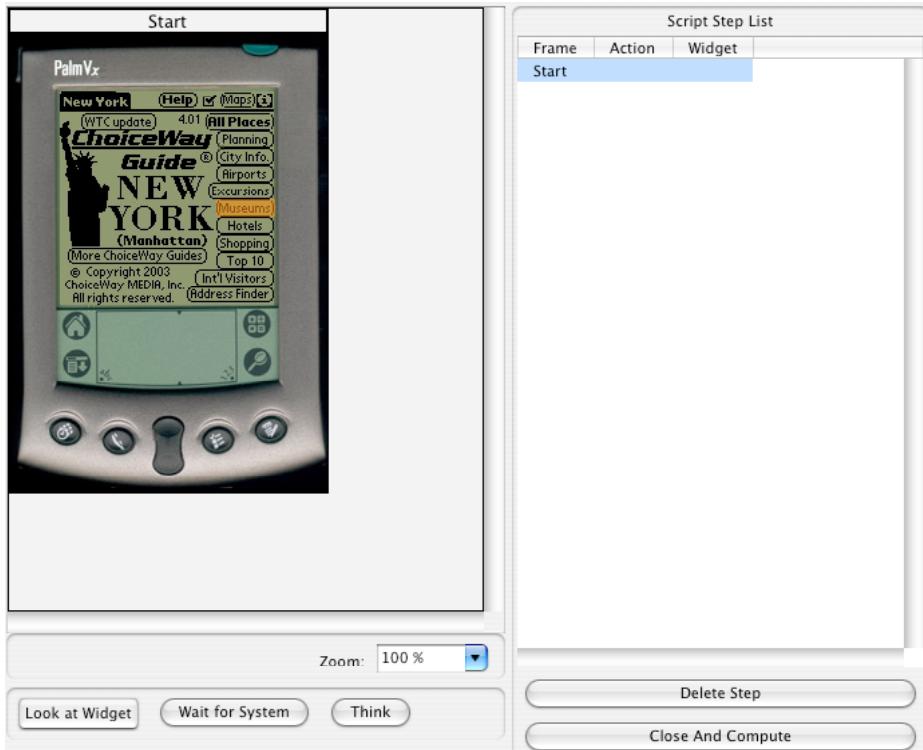


Figure 62. The start **FRAME** chooser window for the **SCRIPT** editor.

## 9.5 Open the Demonstration Window

Once the start **FRAME** has been selected, the **FRAME** chooser window will be replaced by the main demonstration window. (Figure 63) Here you will interact with the storyboard for your **DESIGN** to record a sequence of **SCRIPT STEPS** for the **SCRIPT**. The start **FRAME** appears on the left of the window, and the current list of **STEPS** in the **SCRIPT** appears on the right. A set of buttons for manipulating the **SCRIPT** is located at the bottom of the window.



**Figure 63. The demonstration window for the SCRIPT editor. The FRAME associated with the selected SCRIPT step is displayed in the FRAME pane (on the left) and the SCRIPT steps are displayed in the SCRIPT pane (on the right).**

## 9.6 Zoom the Frame

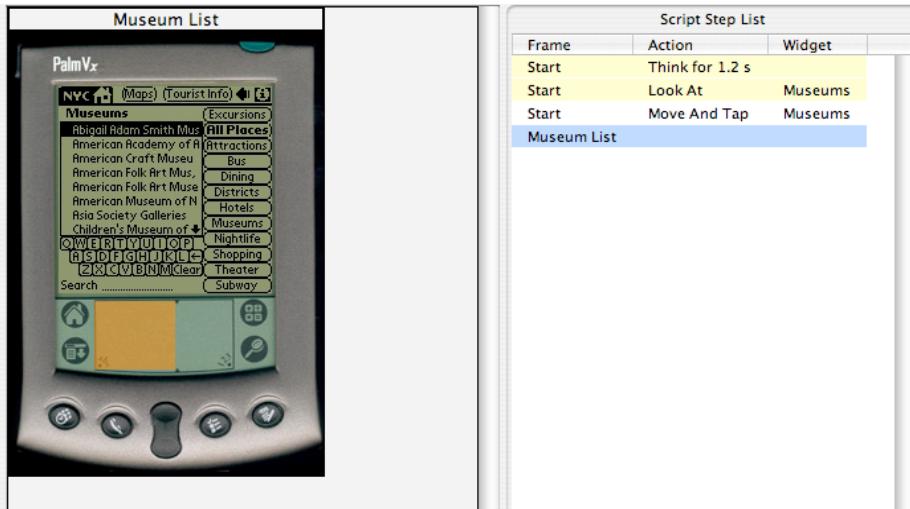
The zoom commands are available in the “Modify” menu, in a zoom box at the bottom right of a **FRAME** pane, in the contextual menu for the **FRAME** pane, or by using the keyboard shortcut : The keyboard shortcuts for

	<u>Mac OS X</u>	<u>Windows</u>
“Zoom In”	⌘ +	CTRL +
“Zoom Out”	⌘ -	CTRL -
“Normal Zoom”	⌘ =	CTRL =
“Zoom to Fit”	⌘ /	CTRL /

The “Normal Zoom” command makes the contents of the **FRAME** pane revert to the default size. The “Zoom to Fit” command makes the contents of a **FRAME**, including the background image; exactly fit the available space in the window. The “Zoom to Fit” command is particularly useful when demonstrating a **SCRIPT**, so you can see an entire **FRAME** at once.

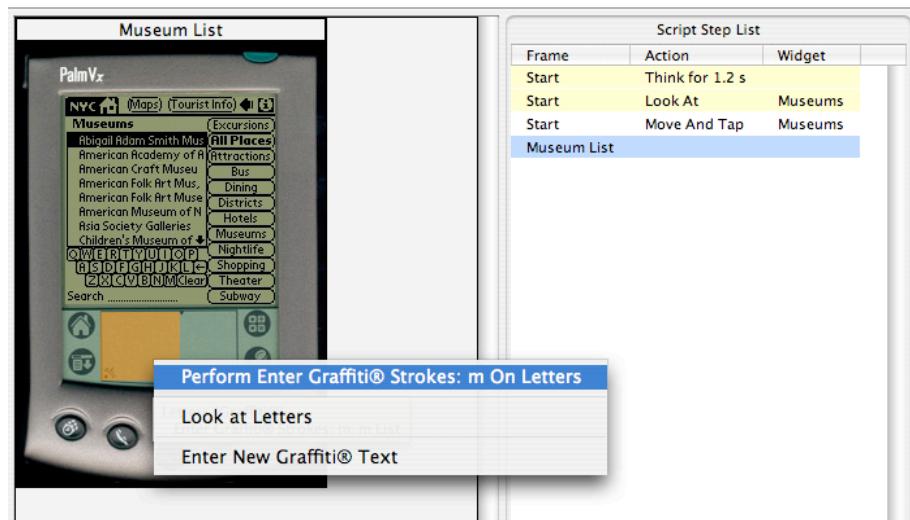
## 9.7 Demonstrate Actions for a SCRIPT

There are two ways to record new **STEPS** for the **SCRIPT**. To record simple actions, you just interact with the storyboard as a user would interact with an actual device. This will work for single- and double-clicks. In the case of a touchscreen interface like a Palm OS®-based handheld, CogTool automatically interprets mouse clicks as taps and double-clicks as double-taps (Figure 64).



**Figure 64. Result of demonstrating the first STEP in the TASK  
“Lookup the MET”**

The other way to demonstrate an action is to open the contextual menu for a **WIDGET** and choose an action from the menu. Some actions (such as Keyboard, Graffiti®, and Voice actions) must be demonstrated via the contextual menu (Figure 65).



**Figure 65. Contextual menu on the Graffiti® WIDGET.**

**NOTE #17.** If your frame has overlapping **WIDGETS**, a click or double-click will automatically follow the transition for the topmost **WIDGET**. All actions on all occluded **WIDGETS** are accessible through the context menu.

The **TRANSITIONS** associated with your actions will be followed automatically when the actions are demonstrated. That is, the current **FRAME** will be replaced with the destination **FRAME** of the **TRANSITION** in the demonstration window. For each action you perform, a new **STEP** is also added to the **SCRIPT**.

**KNOWN BUG #7.** In scripts, a hover action is supposed to appear as a yellow Look-at step, a yellow Move-Mouse step, and a white Hover step. Sometimes the hover step does not appear in the script, but the model still produces a correct prediction because the hover itself does not contribute time to the task (only the move-mouse contributes motor time).

## 9.8 Delete Demonstrated Action

It is not advisable to delete arbitrary steps in the middle of a script. CogTool does not prevent you from selecting an arbitrary step and deleting it, but the resulting script is not guaranteed to produce a psychologically valid prediction. Thus, to safely delete a Demonstrated Action, select the last step you demonstrated and hit the “Delete Step” button repeatedly, “backing up” the script, until you have deleted the undesired step. You can then restart your demonstration from that point.

## 9.9 Review Automatic **SCRIPT Steps**

In the “**SCRIPT STEP List**” on the right, in Figure 65 CogTool has recorded three **STEPS** in response to the single demonstrated action: clicking on “Museums Button”. They are a “Think” for 1.2 seconds (s), a “Look At” to find “Museums Button”, and a “Move and Tap” action on “Museums Button”. Think and Look-at **STEPS** are inserted automatically because prior research in psychology and human-computer interaction has shown that when people tap on a button, they think to remember which button to press, they look at the button, and then they tap that button. The tap results in the **FRAME** changing to the “Museums List” **FRAME**, which is now shown on the left and listed as the last line of the **SCRIPT STEP List**. Automatically inserting Think and Look-at **STEPS** is an important part of CogTool’s functionality. By doing this automatically, CogTool allows analysts to produce valid models whether they have studied psychology or not. The placement of Think and Look-at **STEPS** is backed by decades of psychology research and HCI data and is at the core of the research that produced CogTool.

The **STEPS** that are inserted automatically are shaded in yellow to distinguish them from **STEPS** deliberately demonstrated by the analyst (white **STEPS**). All of the yellow **STEPS** above a white **STEP** are associated with that white **STEP**. If a white **STEP** is deleted, all its yellow **STEPS** will be deleted as well.

The **STEPS** that are inserted automatically are shaded in yellow to distinguish them from **STEPS** deliberately demonstrated or inserted by the analyst in white. All of the yellow automatic **STEPS** above a white demonstrated **STEP** are associated with that demonstrated **STEP**. If a demonstrated **STEP** is deleted, all automatic **STEPS** associated with it will be deleted as well.

## 9.10 Review Transitions Defined on a Widget

While you are demonstrating **STEPS**, if you have difficulty remembering which **WIDGETS** have associated **TRANSITIONS** or the actions to trigger them, hold the mouse over a **WIDGET** to display a tool-tip. You can also use the contextual menu on a **WIDGET** and select a specific action from the menu.

## 9.11 Insert a “Look at” Step

One type of user action can be demonstrated in a third way. If a user looks at a **WIDGET** for visual information but does not physically manipulate it, the appropriate action is a “Look at” **SCRIPT STEP**. This type of action never causes a **TRANSITION** to be followed. To perform a “Look at” **STEP**, press the “Look at **WIDGET**” button, and then click the **WIDGET** at which the user will look. A new “Look at” **STEP** is added to the **SCRIPT STEP List**.

## 9.12 Insert a “Wait for System” Step

An accurate prediction of user task time must include time spent waiting for the system to complete requested actions. To simulate this system delay, you may insert special **SCRIPT STEPS** to represent the delay time. To insert one of these special **STEPS**, press the “Wait for System” button. A dialog box will prompt for the amount of time in seconds the user must wait. You can insert system delay **STEPS** while you are demonstrating the **SCRIPT** or after the other **STEPS** have been completed. When you have entered the desired value, press the “OK” button or the “Enter/Return” key to confirm it. (Figure 66)

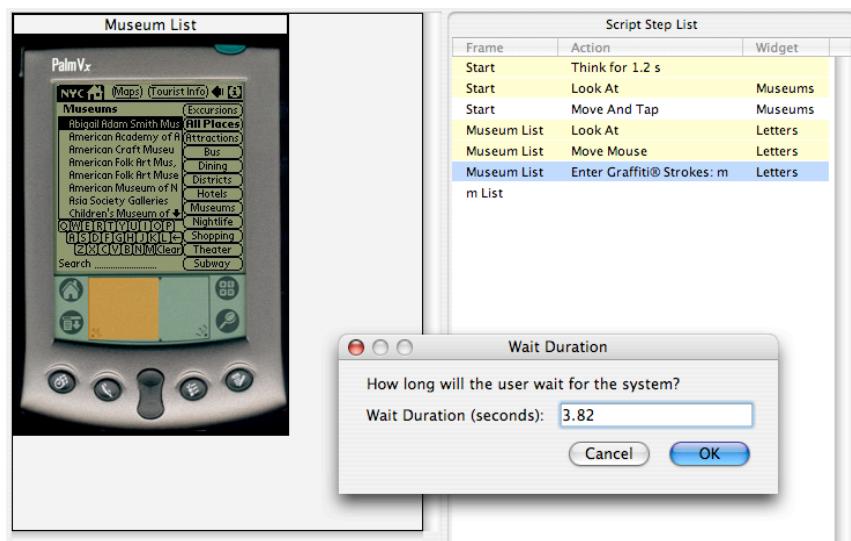


Figure 66. Entering the duration of a “Wait for System” **SCRIPT step**.

The system delay is entered into the **SCRIPT STEP List** immediately before the currently selected **FRAME**, as shown in Figure 67.

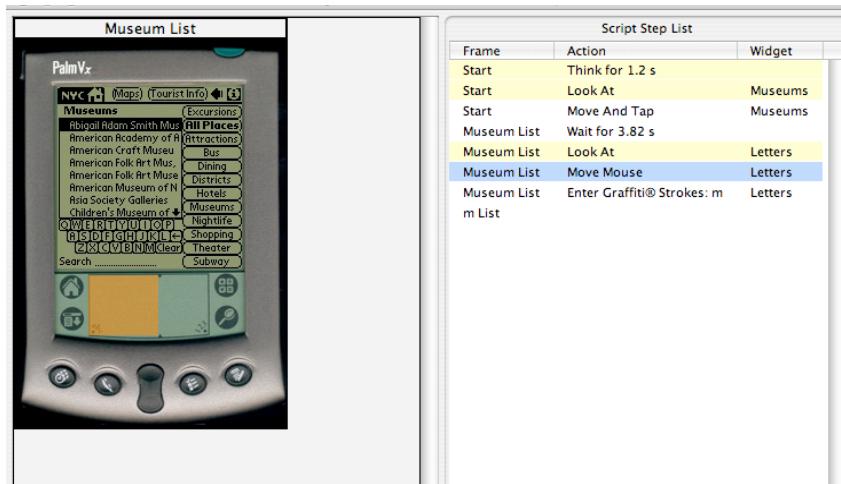


Figure 67. The result of entering system delay before the Museums List FRAME.

### 9.13 Insert or remove a “Think” Step

In some cases, you may wish to change the way that CogTool automatically inserts “Think” STEPS in the SCRIPT. This is highly discouraged, unless you have empirical evidence to support the change. The rules CogTool uses to place “Think” operators have been validated through many experiments spanning several decades. If you believe that you have sufficient justification, you may add additional “Think” operators by selecting the point in the SCRIPT where the new STEP will be added, and then pressing the “Think” button. A dialog box will prompt for the amount of time in seconds the user will pause to think. You can insert “Think” STEPS while you are demonstrating the SCRIPT or after the other STEPS have been completed. You may also remove a “Think” operator by selecting the line and hitting the Delete Step button at the bottom of the pane.

### 9.14 Review a Previous Step

You may review the FRAME being used in a particular SCRIPT STEP by clicking on the STEP in the SCRIPT STEP List. The FRAME will appear in the left pane.

### 9.15 Restart from a Previous Step

If you have made a mistake in demonstrating the TASK, you can return to a previous STEP and continue again from that point. Click on a STEP in the history list to return to that point in the demonstration. If you demonstrate a new action while viewing a previous STEP, CogTool will warn you that this will delete all STEPS after the one being demonstrated. Press “OK” to continue demonstrating the corrected sequence of STEPS.

**KNOWN BUG #8.** Editing scripts is buggy in CogTool 1.0b18. It is safe to add and remove Wait for System steps and to add and remove Think steps, and the resulting predictions will be calculated correctly. However, it is not safe to add, remove or change any other steps; the resulting predictions are almost guaranteed to hang ACT-R or be calculated incorrectly. Do not edit **SCRIPT** steps as described in this section. If you want to make changes to the **DESIGN** or the **SCRIPTS**, delete all steps and redemonstrate the steps.

## 9.16 Close the **SCRIPT** Editor and Compute the Prediction

Once you have finished creating and editing the **SCRIPT**, press the “Close and Compute” button at the bottom of the right pane, to return to the **PROJECT** window and compute a prediction. The result will appear as a value in the **SCRIPT CELL** (Figure 68).

Tasks	New York Guide
Lookup the MET	15.026 s

Figure 68. Predicted task execution time appears in the **PROJECT** window.

**NOTE #18.** If the computation process has an error, it will often return either “0.000” or “600.000” as the answer. This is NOT a correct prediction. You should re-check your design and script.

## 9.17 Re-Compute a **SCRIPT**

The prediction results that are calculated for a **SCRIPT** can become invalid if you make changes to the **DESIGN** it acts upon. CogTool will notify you that a **SCRIPT** result has become invalid by showing a question mark “?” in the **SCRIPT** cell where a **SCRIPT** result would normally appear (Figure 69). To recompute a **SCRIPT**, open the contextual menu on the “?”, or click on the Modify menu in the toolbar, and then choose the “Recompute **SCRIPT**” command

Tasks	New York Guide
Lookup the MET	?

Figure 69. Project window with invalid **SCRIPT** result.

**NOTE #19.** Some changes to a **DESIGN** can prevent a **SCRIPT** from being recomputed properly. For example, deleting a **FRAME** or **WIDGET** that is used in the **SCRIPT** will prevent the **SCRIPT** from working. If the “Recompute Script” command produces an erroneous result (see NOTE above), you may need to demonstrate the **SCRIPT** again from the beginning.

## 9.18 Exporting SCRIPTS to other formats

Sometimes it is useful to have a **SCRIPT** in a format outside of CogTool. We have supplied two commands to export scripts to other formats. The most useful additional format for most CogTool uses is the comma separated value (CSV) document. We also supply an export to ACT-R files for the CogTool developers themselves, other cognitive researchers, or anyone interested in the cognitive engine underlying the predictions.

### 9.18.1 Export Script To CSV

To export a **SCRIPT** listing to a comma separated value (CSV) document use the “Export Script To CSV” command available only in the Script cell contextual menus. To bring up the contextual menu for a desired **SCRIPT**, control-click (Mac) or a right-click (PC), on the desired **SCRIPT**’s cell in the **PROJECT** window, then select the “Export Script To CSV” command from the menu. See Figure 70.

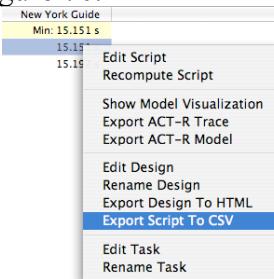


Figure 70. “Export Script to CSV” available through the **SCRIPT** cell contextual menu.

The CVS file can be opened in MS Excel (Figure 71). It contains the date and time of the export, the names of the **PROJECT**, **DESIGN**, and **TASK** as identifying information. The **SCRIPT** itself is recorded in three columns. The left column is the **FRAME** where the action takes place, the middle column is the Action taken in that frame, and the rightmost column is the **WIDGET** on which the action was taken. The resulting **FRAME** starts the next row.

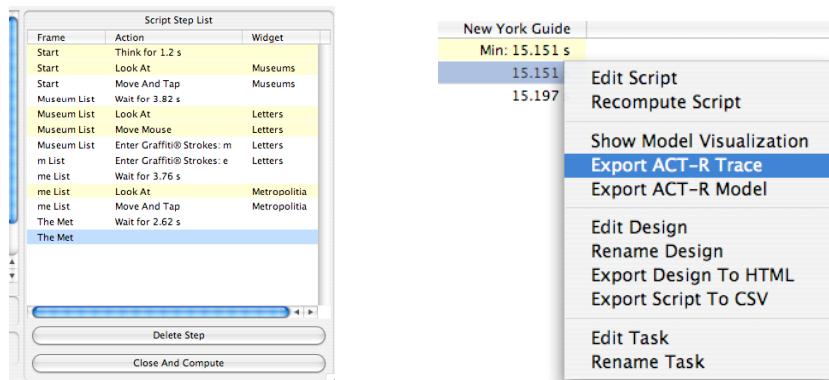
As an example, Figure 71 shows a **SCRIPT** that starts on a **FRAME** named “Start”, where the model thinks for 1.2 seconds. It then looks at the Museums **WIDGET** and moves and taps on that **WIDGET**. This results in the change to a new **FRAME** called Museum List, but the model must wait for 3.82 seconds for the system to display that **FRAME**. The model continues until the script is complete on The Met **FRAME**.

	A	B	C	D
1	Date and Time:	7/8/06 11:48		
2	Project Name:	TutorialProject		
3	Design Name:	New York Guide		
4	Task Hierarchy:	Lookup the MET	Lookup the MET - Soft Keyboard	
5				
6	Frame	Action	Widget	
7	Start	Think for 1.2 s		
8	Start	Look At	Museums	
9	Start	Move And Tap	Museums	
10	Museum List	Wait for 3.82 s		
11	Museum List	Think for 1.2 s		
12	Museum List	Look At	softkeyboard_m	
13	Museum List	Move And Tap	softkeyboard_m	
14	m List	Think for 1.2 s		
15	m List	Look At	softkeyboard_e	
16	m List	Move And Tap	softkeyboard_e	
17	me List	Wait for 3.76 s		
18	me List	Look At	Metropolitnia	
19	me List	Move And Tap	Metropolitnia	
20	The Met	Wait for 2.62 s		
21	The Met			
22				

Figure 71. Result of the “Export Script to CSV” command opened in MS Excel.

### 9.18.2 Exporting ACT-R files

ACT-R is the underlying cognitive modeling engine that predicts times using the steps in a **SCRIPT**. It is not necessary to inspect the inner workings of ACT-R to use CogTool, but we have provided a way to see both the ACT-R commands and the specific ACT-R trace that produces the predictions. CogTool developers use this information to debug Cogtool itself and validate its predictions. To see the ACT-R commands and traces use the menu items available only in the contextual menu of a prediction time in the Project Manager (see Figure 72). These commands create text files that are named and placed in the desired folder with a standard file-save dialog box. Information about ACT-R can be found at <http://act-r.psy.cmu.edu/>.

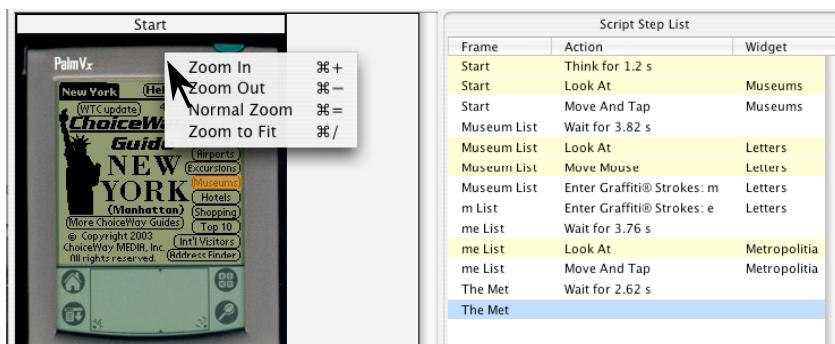


**Figure 72.** On the left is the **SCRIPT** that leads to the prediction listed in the Project window on the right. Selecting “Export ACT-R Trace” or “Export ACT-R Model” from the contextual menu produces text files associated with this particular **SCRIPT**

### 9.19 Contextual menus available from a **SCRIPT** window.

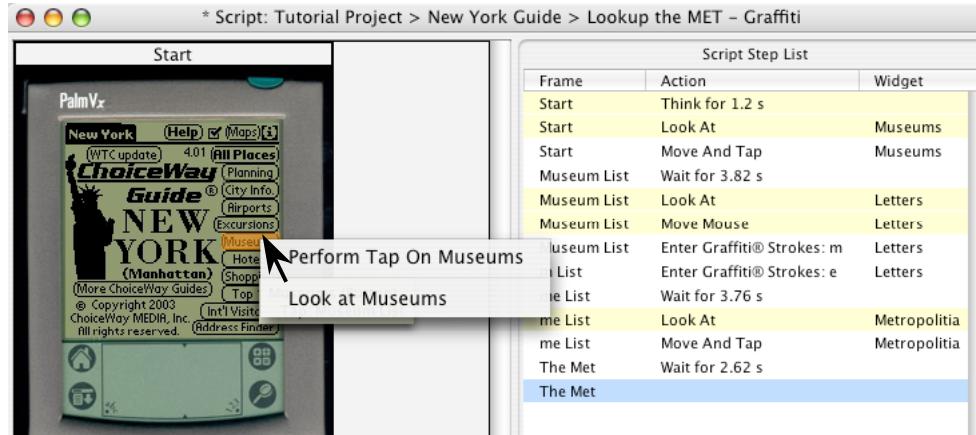
There are 2 different contextual menus available in the **SCRIPT** window. These menus are accessed by right-click (PC) or control-click (Mac) on a certain area within the window.

- 1) Background (see Figure 73) – clicking anywhere in the **FRAME** window area on the left side.



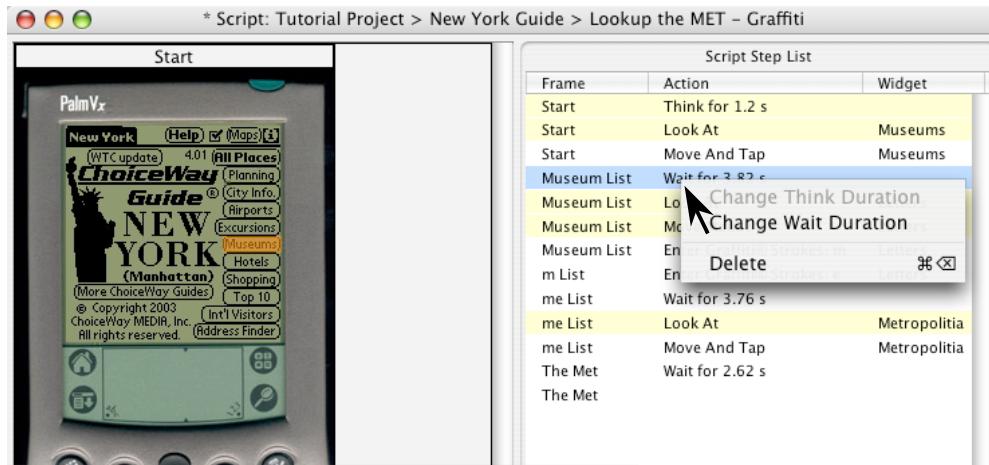
**Figure 73.** Background contextual menu with in the **FRAME** area of the **SCRIPT** window.

- 2) **WIDGET** (see Figure 74) – clicking on a **WIDGET** within the **FRAME** window area on the left side.



**Figure 74. WIDGET contextual menu within the SCRIPT window.**

- 3) Step (see Figure 75) – clicking on a step line in the **SCRIPT** area on the right side.



**Figure 75. Step contextual menu within the SCRIPT window.**

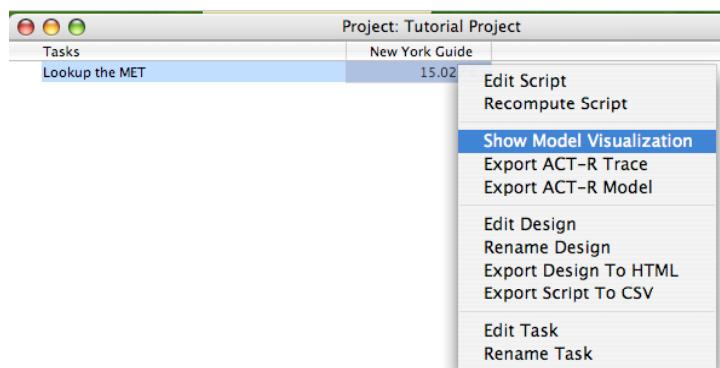
## 10 Model Visualization

The underlying computation cognitive engine that makes the predictions is called ACT-R and is very complex. To learn more about it, you can go to the ACT-R website.

<http://act-r.psy.cmu.edu/>

We have built a visualization tool into CogTool so you can see what ACT-R is doing to produce the predictions. Although this visualization is meant for cognitive psychology researcher to validate CogTool's and ACT-R's predictions, it is fun and may entice you to learn more about computational cognitive modeling.

To see the visualization, in the **PROJECT** window use the contextual menu on a **SCRIPT** cell to select “Show Model Visualization” (Figure 76). You can open as many visualization windows as you wish, and align them to compare the execution traces and see where their times differ.



**Figure 76 The Contextual menu on a SCRIPT cell for the TASK “Lookup the MET – Graffiti”**

The Visualization window has two panes (Figure 77). In the left pane is a timeline showing the different activities ACT-R goes through to make the predictions. This pane uses an overview+details presentation of information where the entire model is shown in miniature at the bottom and a blow-up of a selected portion of the timeline is at the top. The right pane shows either single operator information or a textual trace of model's run, depending on which tab is selected (“Operator Info” or “Model Trace”).

**KNOWN BUG #9.** The detailed area displayed is slightly larger than the area indicated by the selection box on the overview timeline.

In the left pane, time runs from left to right and the width of the boxes are proportional to the time they take to execute. The rows of boxes are different types of “operators” that happen in the course of performing these tasks.

The top row is labeled “Frame” and shows the duration that each frame is visible as the task is performed.

**KNOWN BUG #10.** The frames are not actually visible as long as these boxes indicate. When there is a system wait, one frame disappears and the next one doesn't show up for the duration of the system wait. This is not shown correctly in the current visualization. For example, frame 2-Museum\_List is shown as visible at about 1.6 s into the task. It is really not visible until about 5.7 s into the task. A person well-versed in ACT-R would be able to see this discrepancy in the visualization because the vision row and the production row have many short pairs of operators between 1.6 s and 5.7 s. This pattern of operators represents the eyes trying to see the next button, but the frame not being visible, so the vision system returns a "failure" signal to cognition. As soon as the frame is visible, the vision operator is much longer, as it actually finds the next button to press and returns its location to cognition.

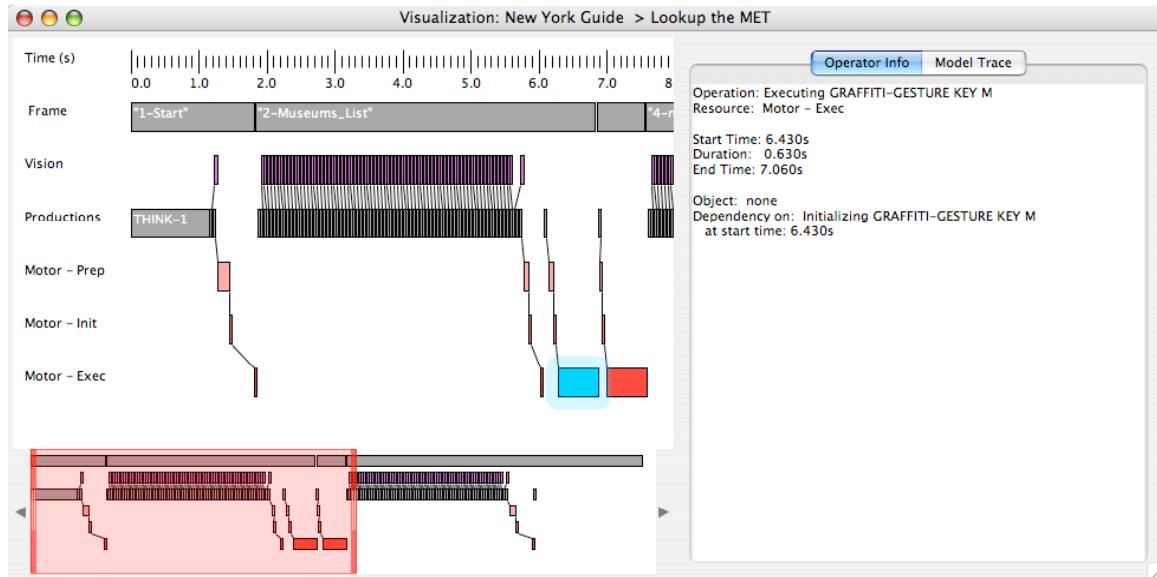
The next row (purple) is Vision, which represents the eyes seeing objects on the frame.

The next row (grey) is labeled "Productions" (the elements of a production system, which is the type of cognitive engine that ACT-R employs), which represent the thoughts the model has when performing this task. The long boxes in this row are the Think operators. Short boxes are other types of cognitive operators that initiate motor movements motor and visual attention shifts.

The next three rows (shades of red) represent different aspects of the motor system employed by ACT-R. ACT-R's motor system is complex and beyond the scope of this tutorial, but the bottom row (darkest red) shows the motor movement that would be observable on a video: horizontal movements of a finger, stylus, or mouse, clicks of a mouse button or key presses. In these models, only horizontal movements of the stylus is shown, because it uses a Palm device that uses a stylus (not a mouse or keyboard).

If you click on a box in the left pane, it highlights the box in bright blue and either gives information about the operator you just selected (in the Operator tab) or highlights the part of the model trace that corresponds to it in the right pane (in the Model Trace tab).

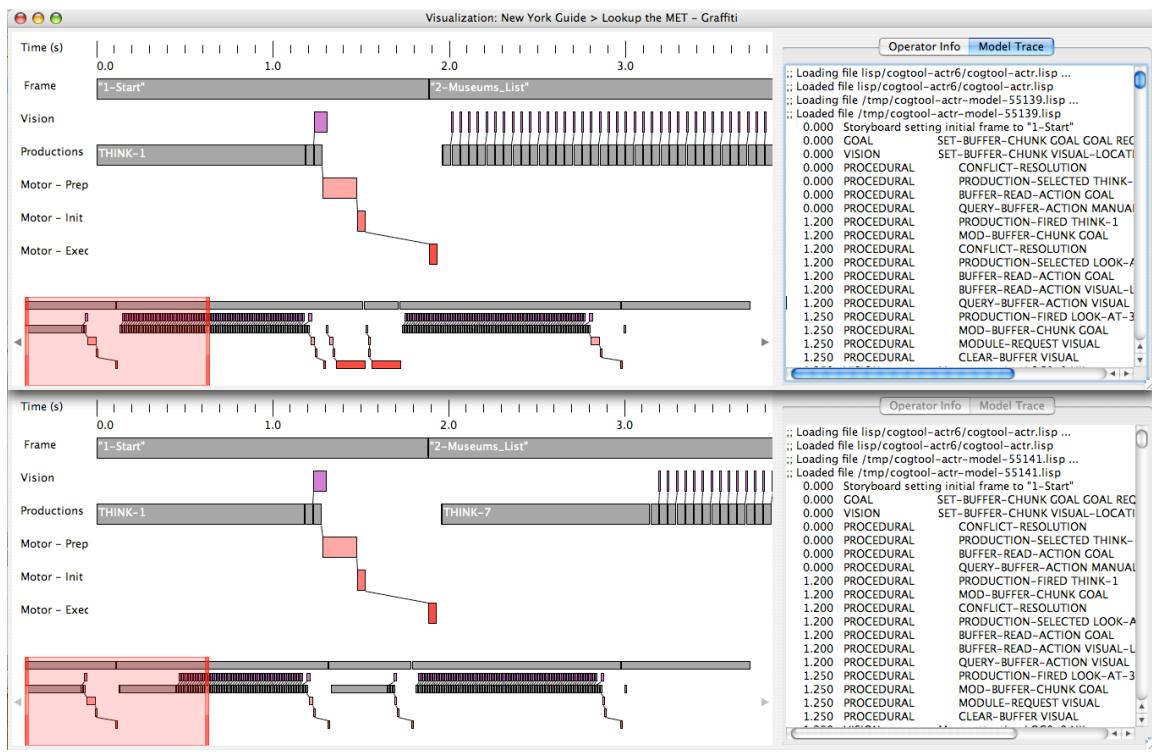
The area shaded in red on the overview timeline represents what is currently showing in the detailed timeline. The detailed timeline can be manipulated by moving the selection box on the overview timeline. You can resize and move the red bars at the ends of the red area. You can move the whole area by click-and-drag from within the red area, or just move one end by clicking-and-dragging a vertical end bar. You can also scroll smoothly by pressing down on the ◀ or ▶ at the edges of the overview timeline, or step by clicking on those arrows.



**Figure 77 A visualization window, showing the operator information tab.**

Figure 78 shows a comparison between two different ways of accomplishing the same goal. This is from the “Look-up the MET” task in the CogTool Tutorial. The biggest difference between the two models evident in the timelines is that the Graffiti® model has long red motor operators where the Graffiti® gestures are (around 6.4 s and 7.2 s) and the Soft-keyboard model has much shorter motor movements to poke the M-key and the E-key. This is because Graffiti® gestures take about 0.5 seconds to complete, on average and the soft-key poking time is calculated from Fitts’s law. Offsetting this savings in motor time, the Soft-keyboard model has an additional Think operator between the M-key and the E-key that the Graffiti® model does not have between the m-gesture and the e-gesture. By default, “me” is treated as a “cognitive unit” by CogTool when entered as successive Graffiti® gestures that are not commands, and successive soft-keyboard presses are not because each key is modeled as a button. We are currently collecting data to validate these approximations. If you would like to see what the times would be if the Think operator was not inserted between the M-key and the E-key, simply delete it from the script and close and compute. (Deleting and inserting Think steps and System Wait steps are the only safe modifications you can make to scripts without re-demonstrating.)

Have fun with this visualization.



**Figure 78. Comparison of Graffiti to Soft-keyboard methods in ACT-R visualization windows**

# 11 Tips

## 11.1 Tips for DESIGN Storyboards

### *Simulating Drag and Drop*

The drag-and-drop interaction technique is often used in graphical direct-manipulation interfaces. CogTool can be used to simulate this behavior with multiple **FRAMES** and **TRANSITIONS**. First, create three **FRAMES**: one **FRAME** for the starting state, one **FRAME** for during the drag action, and one final **FRAME** for the state after the drop occurs. Create a **WIDGET** at the position of the item to be dragged in the start **FRAME** and another in the second **FRAME** over the target location. Create **TRANSITIONS** between the **FRAMES**. Set the first to be triggered by a “Down-click” button action and the second triggered by an “Up-click” button action. With all of these components in place, you can now demonstrate the full sequence of the drag-and-drop operation from start to finish.

### *How much is enough?*

In the example of the Choice Way™ Guide used in the images above, only one **WIDGET** was defined on each **FRAME**, although many buttons are apparent in the background image. This can be appropriate when a **DESIGN** is being used to make a prediction for a single **TASK**. Because each **FRAME** has only a single area to click upon, it is impossible to make an error during a demonstration of that **TASK**. However, if multiple **TASKS** will be performed on the same interface, it is preferable to add as many **WIDGETS** as necessary to the same **FRAME**, rather than creating multiple **FRAMES** with the same background. This makes it possible to demonstrate many unique sequences of actions on the same **DESIGN** without the need for special-purpose adjustments.

### *Remove Background for Layout Experimentation*

If the layout of an interface is still being modified, it can be helpful to use the “Capture Background” command for all **WIDGETS**, then “Remove Background Image” for the **FRAME**. This produces a “cut out” effect for the **WIDGETS** that can be helpful when mocking up new layout variations.

## 12 Appendix A - How to Report a Bug and Get Support

Despite our best efforts, CogTool 1.0b18 is not perfect, and sometimes it will crash. If this happens to you, we would appreciate it if you let us know. If you see a window like Figure 79, please click and drag over the text to select it, then use the CTRL C (on Windows) or ⌘ C (on Mac OS) keyboard shortcuts to copy the text. Paste the text into an email message to Jason Cornwell [jcorn@cs.cmu.edu](mailto:jcorn@cs.cmu.edu) along with a short description of what you did immediately before the crash occurred. If you are willing to share your data file with us, please attach it to the email. We will do our best to resolve the problem and contact you about any workaround or fix that might be available.

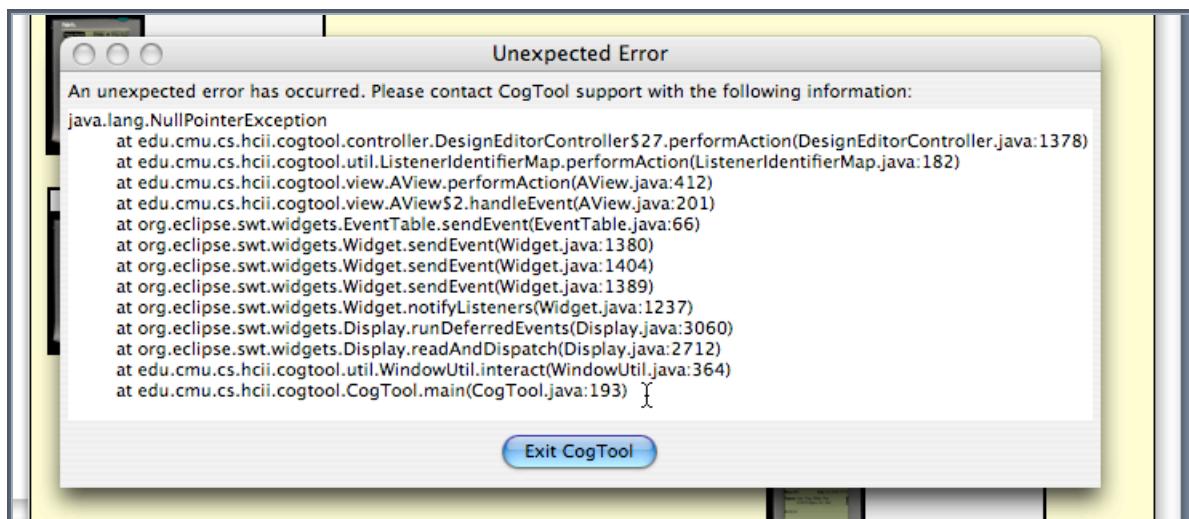


Figure 79. Crash window

## 13 Appendix B – Known Bugs in CogTool 1.0b18

KNOWN BUG #1. In Mac OS X, you cannot launch CogTool by double-clicking on a saved project file (.cgt file). You must open a saved project by launching CogTool itself and then opening the saved project file with the Open option in the CogTool Start-up Dialog Box or from the Open command in the File menu.

KNOWN BUG #2. The “Cut”, “Copy”, and “Paste” commands do not allow **SCRIPTS** to be moved along with their associated **DESIGNS**. You will need to re-demonstrate any **SCRIPTS** associated with **DESIGNS** that are moved via the method described above.

KNOWN BUG #3. When frames are stacked, the stacking order is mysterious and may even seem to change over time. Work around: move the frames apart so they are not stacked.

KNOWN BUG #4. Tool tips do not work on the toolbars for some versions of the Macintosh.

KNOWN BUG #7. In scripts, a hover action is supposed to appear as a yellow Look-at step, a yellow Move-Mouse step, and a white Hover step. Sometimes the hover step does not appear in the script, but the model still produces a correct prediction because the hover itself does not contribute time to the task (only the move-mouse contributes motor time).

KNOWN BUG #8. Editing scripts is buggy in CogTool 1.0b18. It is safe to add and remove Wait for System steps and to add and remove Think steps, and the resulting predictions will be calculated correctly. However, it is not safe to add, remove or change any other steps; the resulting predictions are almost guaranteed to hang ACT-R or be calculated incorrectly. Do not edit **SCRIPT** steps as described in this section. If you want to make changes to the **DESIGN** or the **SCRIPTS**, delete all steps and redemonstrate the steps.

KNOWN BUG #9. The detailed area displayed is slightly larger than the area indicated by the selection box on the overview timeline.

KNOWN BUG #10. The frames are not actually visible as long as these boxes indicate. When there is a system wait, one frame disappears and the next one doesn't show up for the duration of the system wait. This is not shown correctly in the current visualization. For example, frame 2-Museum\_List is shown as visible at about 1.6 s into the task. It is really not visible until about 5.7 s into the task. A person well-versed in ACT-R would be able to see this discrepancy in the visualization because the vision row and the production row have many short pairs of operators between 1.6 s and 5.7 s. This pattern of operators represents the eyes trying to see the next button, but the frame not being visible, so the vision system returns a “failure” signal to cognition. As soon as the frame is visible, the vision operator is much longer, as it actually finds the next button to press and returns its location to cognition.

## 14 Appendix C – Notes about CogTool 1.0b18

NOTE #1. A Project should only be renamed or copied using the “Saved as” command. If the file name is changed outside of CogTool (e.g., edited using the operating system – in the folder) the Project name that appears at the top of the windows WILL NOT BE CHANGED. The Project must be saved under a new name from within CogTool in order to have the window names change.

NOTE #2. CogTool’s storyboard approach was inspired by James Landay’s SYLK system (Landay, J. A. & Myers, B. A. “Interactive Sketching for the Early Stages of User Interface Design.” In Proceedings of CHI ‘95, Denver, CO, May 1995, pp. 43–50.)

NOTE #3. If you do not create a new folder and just select an already existing one, the contents of the folder selected will be REPLACED with the exported design.

NOTE #4. Once a widget has been created, its type cannot be changed. To change a widget’s type you must create a new widget then delete the previous one. Please refer to section 6.13 for a full discussion of widget types so you can choose the right type for each widget you create.

NOTE #5. When you have two actions in a row that act on the same interface “widget” it is important for the two CogTool Widgets to be placed and sized identically. The easiest way to ensure that two Widgets have identical positioning is to create the Widget on **ONE** Frame, copy **IT**, and then paste it onto all other Frames where **THE** Widget appears. SEE Section 6.3.

NOTE #7. If you paste a widget into the same frame it was copied from, the new widget will pasted DIRECTLY ON TOP OF THE ORIGINAL ONE and you won’t be able to see the original widget. Move the newly-pasted widget off of the original to be able to see both.

NOTE #8. If the same interface widget appears in different frames in a design, it is important for the CogTool **THAT** the widgets be placed and sized identically. The only way to ensure that **TWO** Widgets have identical positioning is to create **THE** Widget on **ONE** Frame, copy it, and then paste it onto all **OTHER** Frames where **THE** Widget appears. This becomes important for correct prediction of human performance when there are several actions in a row on the same widget (e.g., clicking the “back” button or a “page down” button several times in a row).

NOTE #9. Hover only has meaning in touchscreen hardware that can detect hovering above the surface of the screen without touching the screen.

**NOTE #10.** In many applications that use Graffiti®, commands are preceded by a diagonal slash gesture, and you would enter “/” as the first character when you define this type of command, e.g., “/c” for Copy, “/v” for Paste. To allow designers to expand the definitions of commands in novel designs, CogTool 1.0b18 does not enforce this convention and any string can be defined as a command by checking the “is Command” box. If you want to use traditional Graffiti® commands, you must put both the “/” and check the box saying that the string is a command to ensure valid time predictions.

**NOTE #11.** CogTool currently only supports the “Press” motion (down AND up) and does not allow separate “Down” and “Up”. In the original KLM, holding down a shift or control key while pressing other keys was approximated by a press of the modifier key before pressing the other keys; CogTool follows this tradition. In addition, modifier keys (e.g. “Shift” or “Ctrl”) and special keys (e.g. Tab or Backspace) are also not currently supported. You can use a nearby key on the keyboard to simulate the press of an unsupported key. For example, to simulate “Tab”, tell CogTool to press the “q” key.

**NOTE #12.** Even though CogTool will allow you to enter a capital letter by holding the “shift” key when defining a keyboard transition, and it will appear as a capital letter in the text field and context menus, CogTool will not give valid time predictions because it will not account for the time to hit the “shift” key. Use a nearby key to simulate the press of the “shift” key.

**NOTE #13. (NOTE #14. NOTE #15. )** The currently implemented methods for reorganizing tasks and task groups are quite awkward, so we suggest that you carefully plan the organization of tasks before demonstrating any scripts. There is no **UNGROUPING** of task groups. You can cut or copy and paste A task name, but if there is A script associated with THAT task it will not copy and will have to be re-demonstrated.

**NOTE #16.** Before you start editing scripts, make sure you have carefully organized your task names. The currently implemented methods for reorganizing tasks and task groups are quite awkward and if there is a script associated with a task it will not move with that task and will have to be redemonstrated.

**NOTE #17.** If your frame has overlapping widgets, a click or double-click will automatically follow the transition for the topmost widget. All actions on all occluded widgets are accessible through the context menu.

**NOTE #18.** If the computation process has an error, it will often return either “0.000” or “600.000” as the answer. This is NOT a correct prediction. You should re-check your design and script.

**NOTE #19.** Some changes to a Design can prevent a Script from being recomputed properly. For example, deleting a Frame or Widget that is used in the Script will prevent the Script from working. If the “Recompute Script” command produces an erroneous result (see NOTE above), you may need to demonstrate the Script again from the beginning.