



# **CogTool**

## **User Guide—DRAFT**

**August 2008**

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

# CONTENTS

<b>CONVENTIONS FOR THIS USER GUIDE.....</b>	<b>6</b>
<b>CHAPTER 1: OVERVIEW .....</b>	<b>7</b>
1.1 Understanding CogTool .....	7
1.1.1 What is CogTool?.....	7
1.1.2 How CogTool Works .....	7
1.1.3 Uses of CogTool.....	8
1.2 Downloading CogTool.....	8
1.3 Installing CogTool .....	9
1.3.1 Installing on Mac OS X .....	9
1.3.2 Installing on Windows XP and Vista.....	9
1.4 Launching CogTool.....	10
<b>CHAPTER 2: MOCK-UP OF THE INTERFACE.....</b>	<b>11</b>
2.1 Projects.....	11
2.1.1 What is a Project? .....	11
2.1.2 The Project Window .....	11
2.1.3 Creating a New Project.....	12
2.1.4 Opening an Existing Project.....	12
2.1.5 Saving a Project.....	13
2.1.6 Closing a Project .....	13
2.1.7 Re-opening a Project Window.....	14
2.1.8 Project Properties .....	14
2.2 Designs .....	14
2.2.1 What is a Design? .....	14
2.2.2 The Design Window .....	15
2.2.3 Creating a New Design .....	15
2.2.4 Editing a Design .....	16
2.2.5 Saving a Design .....	17
2.2.6 Closing a Design .....	17
2.2.7 Moving a Design .....	17
2.2.8 Copying/Pasting a Design.....	17
2.2.9 Deleting a Design .....	18
2.2.10 Duplicating a Design.....	18
2.2.11 Renaming a Design .....	18
2.2.12 Bringing a Design Window to the Foreground .....	19
2.3 Frames .....	19
2.3.1 What is a Frame?.....	19
2.3.2 The Frame Window .....	19
2.3.3 Creating a New Frame .....	21
2.3.4 Renaming a Frame .....	21
2.3.5 Moving a Frame .....	21
2.3.6 Copying/Cutting/Pasting a Frame .....	22
2.3.7 Duplicating a Frame .....	22
2.3.8 Deleting a Frame.....	22
2.3.9 Editing a Frame.....	23
2.3.10 Zooming a Frame .....	23

2.3.11 Setting the Background Image of a Frame .....	23
2.3.12 Removing the Background Image of a Frame.....	24
2.3.13 Importing Frame Background Images.....	24
2.3.14 Bringing a Frame to the Foreground.....	25
2.3.15 Closing the Frame Window .....	25
<b>2.4 Widgets.....</b>	<b>25</b>
2.4.1 What is a Widget? .....	25
2.4.2 Types of Widgets.....	26
2.4.3 Creating a New Widget.....	26
2.4.4 Naming a Widget.....	27
2.4.5 Displaying the Widget Label.....	28
2.4.6 Setting a Widget's Appearance.....	28
2.4.6.1 Setting a Widget Layer Color .....	28
2.4.6.2 Rendering a Widget Skin .....	28
2.4.6.3 Setting an Image for a Widget .....	29
2.4.7 Moving a Widget.....	31
2.4.8 Layering Widgets.....	31
2.4.9 Modifying Widgets .....	32
2.4.10 Copying Versus Duplicating a Widget.....	32
<b>2.5 Transitions.....</b>	<b>33</b>
2.5.1 What is a Transition?.....	33
2.5.2 Creating a New Transition .....	34
2.5.3 Editing a Transition .....	35
2.5.3.1 Setting a Delay for a Transition .....	36
2.5.4 Changing the Source and Destination of a Transition .....	36
2.5.5 Deleting a Transition .....	36
<b>2.6 Export Design to HTML.....</b>	<b>37</b>
<b>CHAPTER 3: QUANTITATIVE ANALYSIS .....</b>	<b>39</b>
<b>3.1 Tasks and Task Groups.....</b>	<b>39</b>
3.1.1 What are Tasks and Task Groups?.....	39
3.1.2 Creating a New Task .....	39
3.1.3 Creating a New Task Group .....	39
3.1.4 Copying/Cutting/Pasting a Task or Task Group.....	40
3.1.5 Duplicating a Task or Task Group.....	40
3.1.6 Renaming a Task or Task Group.....	41
3.1.7 Deleting a Task or Task Group .....	41
3.1.8 Changing the Order of Tasks and Task Groups .....	41
3.1.9 Changing the Computed Value for a Task Group.....	42
3.1.10 Organizing Tasks .....	42
3.1.10.1 Categorical Grouping.....	42
3.1.10.2 Task Breakdown Grouping.....	42
3.1.10.3 Alternative Grouping .....	42
<b>3.2 Scripts .....</b>	<b>43</b>
3.2.1 What is a Script?.....	43
3.2.2 The Script Windows.....	43
3.2.2.1 The Select the Start Frame Window.....	43
3.2.2.2 The Script Edit Window.....	43

3.2.3 Opening or Editing a Script .....	44
3.2.4 Selecting a Start Frame for the Script.....	44
3.2.5 Demonstrating Actions for a Script.....	45
3.2.6 Editing Demonstration Actions .....	46
3.2.7 Deleting Demonstration Actions .....	46
3.2.8 Reviewing Your Demonstration.....	47
3.2.8.1 Reviewing Automatic Script Steps .....	47
3.2.8.2 Reviewing the Summed Task Group .....	47
3.2.8.3 Reviewing Transitions Defined on a Widget.....	47
3.2.8.4 Reviewing a Previous Step.....	47
3.2.9 Inserting Items into a Script.....	48
3.2.9.1 Inserting a Self-Transition .....	48
3.2.9.2 Inserting a “Look At” Step .....	48
3.2.9.3 Inserting or Removing a “Think” Step .....	48
3.2.10 Restarting a Script from a Previous Step .....	49
3.2.11 Computing a Prediction.....	49
3.2.12 Recomputing Scripts .....	50
3.2.13 Exporting Scripts to CSV.....	50
3.2.14 Exporting All Results to CSV.....	51
3.3 Model Visualization .....	52
<b>APPENDIX A: IMPROVEMENTS TO COGTOOL SINCE THE LAST PUBLIC RELEASE .....</b>	<b>56</b>
<b>APPENDIX B: TYPES OF WIDGETS .....</b>	<b>57</b>
Button .....	57
Link.....	57
Checkbox .....	58
Radio Button.....	58
Text Box.....	58
Text.....	59
Menu.....	59
Submenu .....	59
Menu Item .....	60
Context Menu .....	60
Pull-Down List.....	60
Pull-Down Item.....	61
List Box Item .....	61
Graffiti® .....	62
Non-Interactive .....	62
<b>APPENDIX C: STANDARD VERSUS CUSTOM WIDGETS .....</b>	<b>63</b>
Button Widget.....	63
Link Widget.....	63
Checkbox Widget.....	63
Radio Button Widget.....	64
Text Box Widget.....	65
Text Widget.....	65
Menu Widget.....	65
Context Menu Widget .....	66
Pull-Down List Widget.....	66

List Box Item Widget .....	66
Non-Interactive Widget.....	66
<b>APPENDIX D: TRANSITION PROPERTIES .....</b>	<b>67</b>
Mouse .....	67
Keyboard .....	68
Touchscreen .....	68
Graffiti® .....	69
Microphone.....	69
Speaker.....	70
<b>APPENDIX E: SEMANTIC KEYS AND SYMBOLS.....</b>	<b>72</b>
<b>APPENDIX F: REPORTING A BUG.....</b>	<b>73</b>
<b>APPENDIX G: COGTOOL SHORTCUT KEYS .....</b>	<b>75</b>
<b>APPENDIX H: COGTOOL CONTEXT MENUS .....</b>	<b>77</b>
<b>APPENDIX I: ADDITIONAL INFORMATION.....</b>	<b>78</b>
References .....	78
Learn More.....	79
<b>APPENDIX J: GLOSSARY .....</b>	<b>80</b>
ACT-R.....	80
Demonstration.....	80
Design .....	80
Frame.....	80
Keystroke-Level Model (KLM).....	80
Project.....	80
Script .....	80
Task.....	81
Task Group .....	81
Transition.....	81
Widgets .....	81

## CONVENTIONS FOR THIS USER GUIDE

This user guide documents the functionality of CogTool, a software package that allows technicians to prototype user interfaces and perform quantitative analysis. Video tutorials are available from the CogTool Project documentation page: <http://www.cs.cmu.edu/~bej/cogtool/documentation.html>.

Instructions for each Task are given by having the user navigate to CogTool's menu system. Users may also bring up context menus for the Tasks or use shortcut keys. Although context menus are not explicitly mentioned in the main section of the documentation, they are recorded in the Appendices. Accessing commands through a context menu will be mentioned in the main section of the user guide when it is the only way to accomplish a Task.

Shortcut keys are listed in the sidebar when they are applicable to a Task. They are also collected together in an Appendix section, which can be used as a quick reference sheet.

The following items are included in the Appendices:

- Improvements to CogTool Since the Last Public Release
- Types of Widgets
- Standard Versus Custom Widgets
- Transition Properties
- Semantic Keys and Symbols
- Reporting a Bug
- CogTool Shortcut Keys
- CogTool Context Menus
- Additional Information (References and Learn More)
- Glossary

Special graphics and boxes are used to denote the following items:



Notes that explain the behavior of CogTool



Tips



Warnings of potential problems



Documented bugs in CogTool (bugs are constantly being fixed and may not be found in later versions of CogTool)



Things to remember while using CogTool

## **CHAPTER 1: OVERVIEW**

### 1.1 Understanding CogTool

#### **1.1.1 What is CogTool?**

CogTool is a software application that runs on both Windows and Mac operating systems. Utilizing ACT-R and KLM (Keystroke-Level Modeling), CogTool allows users to easily prototype Designs and predicts the time an expert user will take to interact with an interface. With CogTool, an evaluator can create user interface Designs that can then be used to:

- Communicate ideas to others
- Make the Design available to others for review
- Evaluate Design and user interface ideas

By making prototyping easy and by predicting the behavior of expert users, CogTool can eliminate the need for extensive user testing. You may use CogTool, for instance, to evaluate several Design options so you can settle on the two or three Designs you wish to evaluate further through user testing.

This release of CogTool only makes predictions based on the performance of an expert user. Future releases of CogTool may be expanded to make predictions on new user behavior.

#### **1.1.2 How CogTool Works**

CogTool bases its predictions on prototyping according to the storyboard metaphor utilized by James A. Landay and Brad Myers (see the References section of Appendix I: Additional Information). CogTool is built on the 30 years of cognitive psychology research in human and computer interactions, and it utilizes ACT-R and KLM to make its predictions. By using background images in JPG, PNG, or GIF format, you can create Design interfaces with the look and feel of the application or product being evaluated.

In order to produce valid cognitive models, you must correctly name or type Widgets, or interface elements, in CogTool. For example, if you use a radio button, simply type it as a radio button. Since CogTool is keyed to the types of Widgets, all the cognitive work is performed in the background.

KLM has never claimed to have an accuracy rate better than within 20% of a large group sampling. Testing has shown that CogTool has an accuracy rating within 10% of performance. If you receive predictions in CogTool that are within 10% to 20% of each other, you cannot assume that the one with the better rating is the better prediction. Only predictions greater than 20% can be efficiently evaluated.

### **1.1.3 Uses of CogTool**

CogTool can be used for the following:

- Prototypes: Model human performance. With CogTool, you can benchmark or create prototypes to compare interfaces that perform the same Task. This allows the evaluator to compare numerous ideas to find the one that is better for skilled or experienced users.
- Requirements: Determine the specifications for the Tasks the application or product should perform. You can then compare products to your specifications to see which is better.
- Acceptance testing: If you have contracted for an application or a product, you can perform acceptance testing to confirm if the product meets your specifications.
- Many metrics of performance: CogTool currently makes predictions on skilled execution time. However, exploratory behavior and power consumption are currently being researched for future releases of CogTool.

At this time, CogTool can only model interfaces that work in reaction to human interactions. You can use CogTool to model the following:

- Any interfaces that utilize a keyboard or a mouse (although KLM was primarily designed for command line, it makes valid predictions with graphical user interfaces (GUI) )
- Any device with buttons, straight-line gestures, or Graffiti®, such as PDAs or cell phones
- Any physical system that has a display with which the user can interact, such as an airplane cockpit
- Systems with voice interfaces

Systems that work on their own or volunteer information are not currently specified in CogTool.

## **1.2 Downloading CogTool**

The latest public release of CogTool is available from the CogTool Project's website on the download page (<http://www.cogtool.org/download.html>).

Installers are available for Windows XP/Vista and Mac OS X versions greater than 10.3 (including Intel Macs).

## 1.3 Installing CogTool

### 1.3.1 Installing on Mac OS X

1. When you download the Mac installer, your browser will launch archive software to unpack the software. If it does not, double-click the archive file to expand it.

 The file is expanded in the same location as the archive.

2. Place the expanded CogTool file in the **Applications** folder. If the file cannot be placed in the **Applications** folder, it will still run from any location on your hard drive.



CogTool

### 1.3.2 Installing on Windows XP and Vista

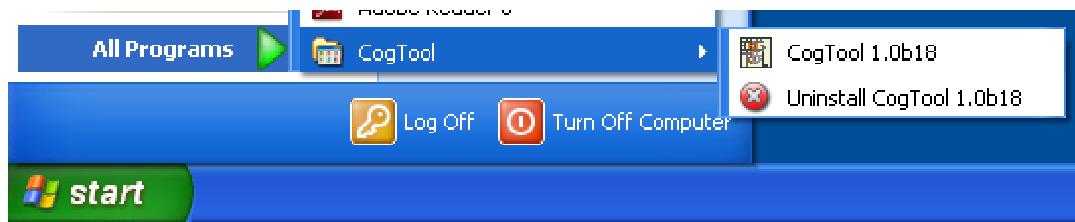
1. You must have Java Runtime Environment installed in order to use CogTool. The latest release of Sun's Java Runtime Environment is available at <http://java.com/en/download/>. Follow the instructions provided by Java Runtime Environment to install the software.

 Windows 2000 users will also need to download the GDI+ library from Microsoft's web site and install it.

2. Double-click the CogTool installer to launch the install wizard.

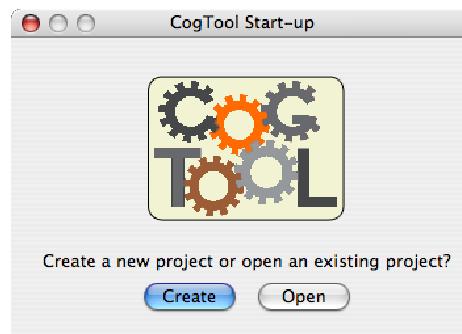


3. Follow the instructions in the wizard to complete the CogTool installation.
4. Once CogTool is installed, it can be launched from the Windows Start Menu.



## 1.4 Launching CogTool

You can launch CogTool by double-clicking the application icon or by choosing the application from the listing in the Windows Start Menu.



In Mac OS X, you cannot launch CogTool by double-clicking a saved project file (.cgt). You must first launch CogTool. Then click the **Open** button and navigate to your saved project file.



With Mac OS X, you can launch the CogTool application from any location to which you save it.

## CHAPTER 2: MOCK-UP OF THE INTERFACE

With CogTool, you can create a mock-up for an interface. This allows you to rapidly compare many different interface ideas to see which works best for your situation. You can perform benchmarks on the current interface design and new interface recommendations. In order to create a mock-up, you must first create a Project and a Design.

### 2.1 Projects

#### 2.1.1 What is a Project?

A Project is central to CogTool. All the parts necessary for CogTool to make performance predictions are kept in a Project file. The Project contains the user interface Designs, the Tasks being studied, and the Scripts that define the human performance being predicted.



The Project name is also the name of the .cgt file that is created by CogTool.

#### 2.1.2 The Project Window

The Project window is laid out as a matrix of cells and titles. The title bar starts with the name “Project” and contains the name of the Project.

The column headers, starting with the second column, contain the names of the Designs associated with the Project.

The row headers contain the names of the different Tasks and Task Groups.

Script cells show the predicted execution time for the Tasks under a listed Design.

Project: Phone Search		
Tasks	pad layout	row layout
▼ Use keyboard	Sum: 8.591 s	Sum: 34.677 s
Search for 1 number	2.954 s	13.320 s
Search for 2 numbers	5.637 s	21.357 s
▼ Use touchscreen	Sum: 12.575 s	Sum: 21.420 s
Search for 1 number	4.947 s	7.528 s
Search for 2 numbers	7.628 s	13.892 s

Figure 2-1: The Project window: This example has two Designs and two Task groups, each containing two Tasks.

### 2.1.3 Creating a New Project

Mac: ⌘N

Win: CTRL+N

To create a Project, do one of the following:

- In the CogTool startup window, click the **Create** button.

OR

- From the **File** menu, choose **New Project**.

Because a Project must contain at least one Design, you will be prompted to name a new Design for the new Project (see Figure 2-2). The name you choose should be descriptive of the interface being modeled. Since each Design must have at least one input and one output device, you will be prompted to select devices in the **New Design** dialog box. See **Section 2.2.2 Creating a New Design** for more information on the **New Design** dialog box.

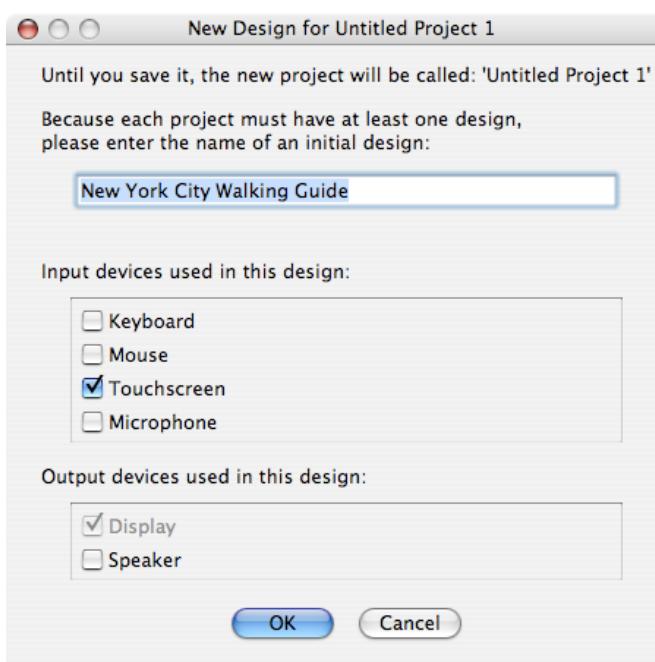


Figure 2-2: The New Design dialog box



By default, the **Display** is selected as an **Output device**. It is not possible to deselect the display in this version of CogTool.



Once you have defined the devices used in a Design, you cannot change them. Consider your Design needs thoroughly before selecting devices.

### 2.1.4 Opening an Existing Project

Mac: ⌘O

Win: CTRL+O

To open an existing Project, do one of the following:

- In the CogTool startup window, click the **Open** button.

OR

- From the **File** menu, choose **Open**.

You will be prompted to choose a file to open.

To reopen recent Projects:

- From the **File** menu, choose **Open Recent**.

### 2.1.5 Saving a Project

**Mac:** ⌘S

To save a Project:

- From the **File** menu, choose **Save**.

The first time you save a file, you will be prompted for the filename and location. By default, the file will be given a .cgt extension. The filename (minus the .cgt extension) will appear in the title bar of all CogTool windows for the Project.



An asterisk (\*) in front of the word “Project” in the title bar of a window signifies that the Project has unsaved modifications (see Figure 2-3). When the file has been saved, the asterisk disappears.

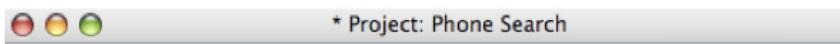


Figure 2-3: The asterisk denotes a Project with unsaved modifications.

**Mac:** ⌘S

To save a Project with a new name:

- From the **File** menu, choose **Save As**.

**Win:**  
CTRL+SHIFT+S

### 2.1.6 Closing a Project

**Mac:** ⌘W

To close a Project:

- From the **File** menu, choose **Close Project**.

**Win:** CTRL+W

To close a window:

- From the **File** menu, choose **Close Window**.



If a Project has unsaved modifications, you will be prompted to save the Project when you close it.



You can also close an open window by clicking the close button on the window title bar.

## 2.1.7 Re-opening a Project Window

To open a closed Project window when one of its Design, Frame, or Script windows is open:

- From the **Window** menu, choose **Display Project**.



If you have multiple windows open for a Project, you can bring the Project window to the front by selecting it from the **Window** menu. Choose the Project's name from the menu listing.

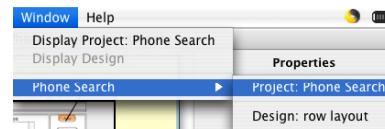


Figure 2-4: View of the Window menu

## 2.1.8 Project Properties

To see the version number of the CogTool application last used to save the Project:

- From the **File** menu, choose **Properties** (see Figure 2-5).



Figure 2-5: A sample Project Properties window

## 2.2 Designs

### 2.2.1 What is a Design?

The CogTool Design approach was inspired by James Landay's SYLK system (see the References section of Appendix I: Additonal Information). A Design is a series of Frames that represent the user interface of the system being analyzed. Each Frame represents an individual screen of the interface, the interactive elements on the screen (e.g. menus, list boxes, etc.) and around the screen (e.g. physical buttons), and areas that trigger changes. Transitions, which show how a particular user action causes the interface to react, link the Frames to each other.

## 2.2.2 The Design Window

Design windows have a light yellow background to distinguish them from the other CogTool windows. The title bar starts with the word “Design” (see Figure 2-6).

Frames are assembled and Transitions are inserted in the Design window.

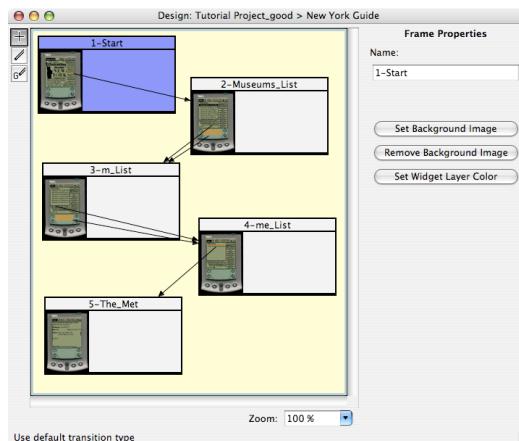


Figure 2-6: The Design Window

## 2.2.3 Creating a New Design

When you create a Project, the **New Design** window appears on the screen (see Section 2.2.2).

**Mac:** ⌘⌘D

**Win:**  
CTRL+SHIFT+D

To create additional Designs for your existing Project:

- From the **Create** menu, choose **New Design**.

When you create a new Design, do all of the following:

- Choose a name unique to the Project and descriptive of the interface being modeled. This will help you to identify the purpose of the Design if you have multiple Designs in your Project window.
- Select at least one input and one output device. You will not be able to click **OK** until you have done so.



By default, the **Display** is selected as an **Output device**. It is not possible to deselect the display in this version of CogTool.

- Think of all possible inputs and outputs to your Design and select as many devices as you will need (see Figure 2-7).



Once you have defined the devices used in a Design, you cannot change them. Consider your Design needs thoroughly before selecting devices.

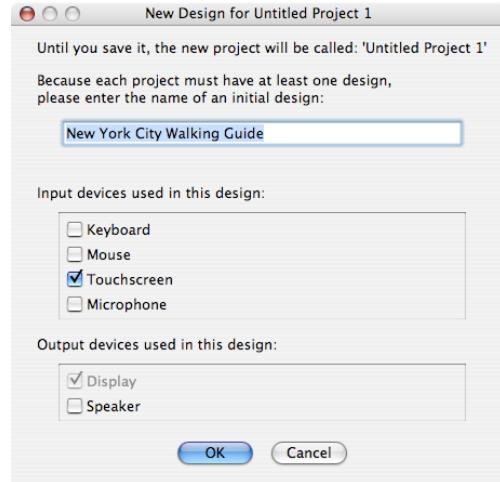


Figure 2-7: The New Design window: Since this Design is for a PDA, only the touchscreen is selected as an input device.

When Designs are created, the Design columns appear in the order in which they are created with new columns taking the right-most position.

If a Design column is selected when a new Design is created, the new Design will appear immediately to the right of the selected Design column (see Figure 2-8).

* Project: New York PDA Guides			
s	New York City Walking Guide	• Design 3	Design 2
1			

Figure 2-8: The New York City Walking Guide Design column was selected when Design 3 was created and placed.

## 2.2.4 Editing a Design

**Mac:** ⌘E

**Win:** CTRL+E

To edit a Design, do one of the following:

- In the Project Window, select the Design.
- From the **Edit** menu, choose **Edit**.

OR

- Double-click the Design name.



A Design is selected when a dot appears next to the Design name and the Script cells below the name are highlighted in blue (see Figure 2-9).

Project: New York PDA Guides	
tasks	• New York City Walking Guide
task 1	

Figure 2-9: Selected Design

## 2.2.5 Saving a Design

Changes to your Design are saved when you save the Project.

## 2.2.6 Closing a Design

**Mac:** ⌘W

**Win:** CTRL+W

To close a Design window:

- From the **File** menu, choose **Close**.

## 2.2.7 Moving a Design

You can change the order of the Design columns by clicking on the Design name and dragging the column to the desired position.

You can also change the order of the Design columns by cutting and pasting the Design columns into the desired location.

**Mac:** ⌘X

**Win:** CTRL+X

To cut a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Cut**.

**Mac:** ⌘V

**Win:** CTRL+V

To paste a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Paste**.



If an existing Design is selected, the pasted Design will be added to the right of the selected Design. If no Design is selected, the pasted Design will be added to the right-most column.

## 2.2.8 Copying/Pasting a Design

You can copy a Design and paste it into another location of the same Project or into a different Project.

**Mac:** ⌘C

**Win:** CTRL+C

To copy a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Copy**.

**Mac:** ⌘V

**Win:** CTRL+V

To paste a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Paste**.



If an existing Design is selected, the pasted Design will be added to the right of the selected Design. If no Design is selected, the

pasted Design will be added to the right-most column.

If the Design's name is already in use in the Project, a unique suffix is appended to the pasted Design's name (see Figure 2-10).

* Project: New York PDA Guides		
	New York City Walking Guide	New York City Walking Guide [2]

Figure 2-10: Example of unique suffix added to a Design

If one or more of the Task names already exist in the Project, you will be asked if the pasted Tasks should be treated the same as the existing Tasks. If they are, the pasted Tasks are placed in the same row as the existing Tasks. If they are not, the pasted Tasks are placed in a new row and a unique suffix is appended to the names of the pasted Tasks.

## 2.2.9 Deleting a Design

**Mac:** ⌘⌫

**Win:**  
CTRL+DELETE

To delete a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Delete**.

You will have to confirm the deletion of the Design. It will not be deleted until you click **OK**.

## 2.2.10 Duplicating a Design

**Mac:** ⌘D

**Win:** CTRL+D

To duplicate a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Duplicate**.



The Design is duplicated to the right of the original Design, and its name will be given a unique suffix (e.g. "MyDesign[1]").

## 2.2.11 Renaming a Design

**Mac:** ⌘R

**Win:** CTRL+R

To rename a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Rename**.

Enter the new name for the Design when prompted. The name will be changed when you click **OK** (see Figure 2-11).



Figure 2-11: The Rename Design dialog box



Each Design name must be unique within a Project.

### 2.2.12 Bringing a Design Window to the Foreground

To bring a Design window to the foreground, do one of the following:

- In the Project window, double-click the Design name.

OR

- From the **Window** menu, choose the Project name.
- From the drop-down list, choose the Design name.

## 2.3 Frames

### 2.3.1 What is a Frame?

A Design consists of Frames and Transitions. A Frame represents a single screen of a user interface. Multiple Frames can be linked together to form a complete Design.

Frames may contain background images and Widgets to represent the look and interactivity of the interface you are designing. Frames are connected by Transitions from their Widgets to other Frames.

### 2.3.2 The Frame Window

The Frame window contains a toolbar with Widgets, the interface representation, Widget properties, and zoom capabilities.

The Widget types listed on the toolbar to the left of the window are available depending on whether the Standard (see Figure 2-12) or Custom (see Figure 2-13) toolbar option is selected (see **Section 2.4 Widgets** for more information on Widgets).

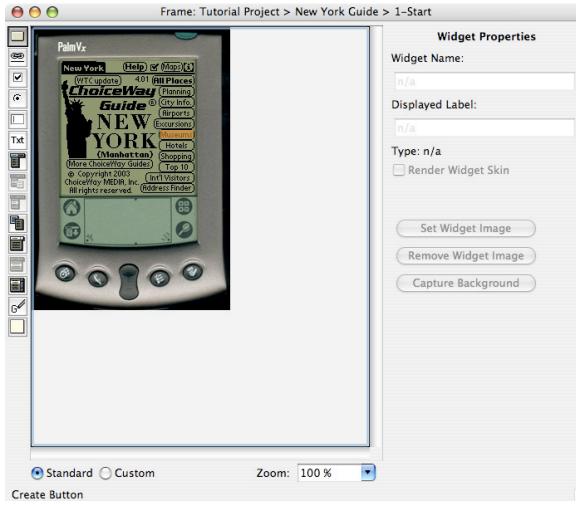


Figure 2-12: The standard Frame window

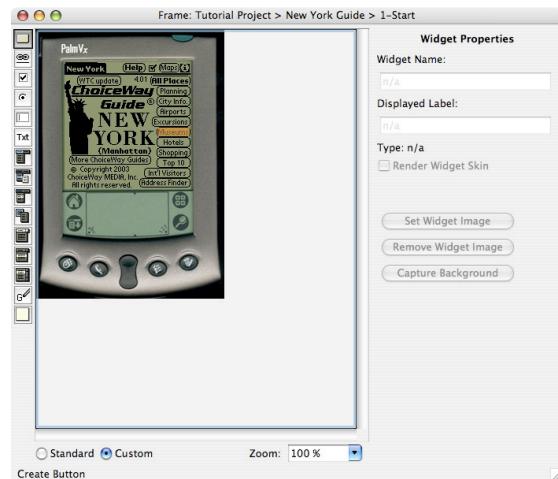


Figure 2-13: The custom Frame window



When a Frame window is made too narrow, the Standard, Custom, and Zoom labels run into each other (see Figure 2-14).

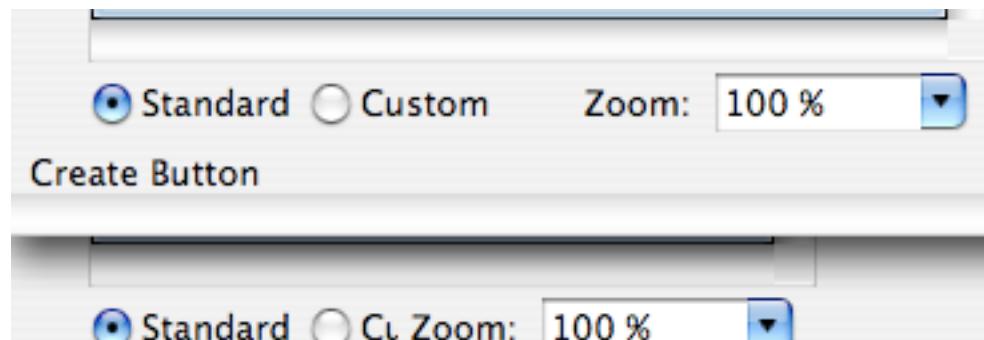


Figure 2-14: Example of normal-sized and undersized Frame windows

### 2.3.3 Creating a New Frame

When a Design window is opened for the first time, a new Frame is automatically created.

**Mac:** ⌘F

**Win:**

CTRL+SHIFT+F

To create a new Frame:

- From the **Create** menu, choose **New Frame**.

### 2.3.4 Renaming a Frame

**Mac:** ⌘R

**Win:** CTRL+R

To change the name of a Frame, do one of the following:

- From the Edit menu, choose **Rename Frame**.

OR

- Double-click the Frame's name. The Frame title bar turns blue when the name can be changed.



A Frame name may only be used once in a particular Design.

### 2.3.5 Moving a Frame

You may arrange the Frames to make the Design more visually pleasing and easier to follow. A Design consisting of new blank Frames might look like the figure below (see Figure 2-15).

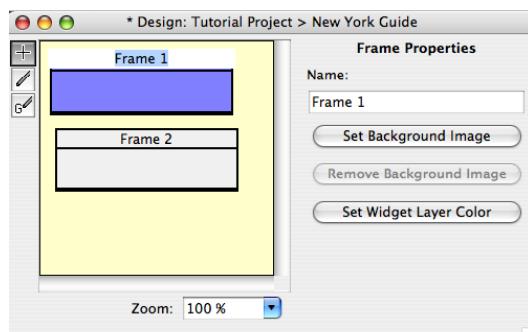


Figure 2-15: Two empty Frames: The default name, “Frame 1,” is highlighted and ready to be replaced by a more meaningful name.

To move a Frame, simply click on the center of the Frame and drag it to a new location.

To move a Frame in fractional increments, use **Nudge**:

- From the **Modify** menu, select **Nudge**.



To reduce confusion, move the Frames apart so they are not overlapping or stacked.

### 2.3.6 Copying/Cutting/Pasting a Frame

**Mac:** ⌘C

**Win:** CTRL+C

To copy a Frame:

- Select the Frame.
- From the **Edit** menu, choose **Copy Frame**.



When you copy a Frame, only the contents of the Frame are copied. Transitions are not.

**Mac:** ⌘X

**Win:** CTRL+X

To cut a Frame:

- Select the Frame.
- From the **Edit** menu, choose **Cut**.

A Frame or selection of Frames can be copied or cut and pasted into the same or different Design in the same or different Project.

**Mac:** ⌘V

**Win:** CTRL+V

To paste a Frame:

- Select the Frame.
- From the **Edit** menu, choose **Paste Frame**.

If the name of the pasted Frame already exists in the Design, the pasted Frame is given a unique suffix (e.g. “LastFrame[1]”).

### 2.3.7 Duplicating a Frame

When you copy a Frame, you only copy the content held within the Frame. No Transitions to other Frames are copied. To make an exact duplicate of a Frame, including Transitions, duplicate the Frame.

**Mac:** ⌘D

**Win:** CTRL+D

To duplicate a Frame:

- Select the Frame.
- From the **Edit** menu, choose **Duplicate Frame**.



In CogTool for Windows, you can also duplicate a Frame by holding Control while dragging it.

Duplicated Frames are slightly down and to the right of the original Frame, and their names are given unique suffixes (e.g. “LastFrame[1]”).

### 2.3.8 Deleting a Frame

**Mac:** ⌘⌫

**Win:**  
CTRL+DELETE

To delete a Frame:

- Select the Frame.
- From the **Edit** menu, choose **Delete Frame**.



If you have already demonstrated your Design, you will have to redo your Demonstration after deleting Frames from your Design.

### 2.3.9 Editing a Frame

A Frame must contain information about the interface so that CogTool can make predictions.

**Mac:** ⌘E

**Win:** CTRL+E

To edit a Frame, do one of the following:

- Select the Frame.
- From the **Edit** menu, choose **Edit Frame**.

OR

- Double-click the body of the Frame.

### 2.3.10 Zooming 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.

**Zoom In**

**Mac:** ⌘+=

**Win:** CTRL+=

To zoom a frame, do one of the following:

- From the Modify menu, choose the Zoom option you want.

OR

- Choose your zoom settings from the box at the bottom of the Frame window.

**Zoom Out**

**Mac:** ⌘-

**Win:** CTRL+-

**Normal Zoom**

**Mac:** ⌘0

**Win:** CTRL+0



**Normal Zoom** makes the contents of the Frame revert to the default size. **Zoom to Fit** makes the contents of a Frame fit the available space in the window.

**Zoom to Fit**

**Mac:** ⌘/

**Win:** CTRL+/

### 2.3.11 Setting the Background Image of a Frame

You may use a background image for your Frame. Background images are useful in helping you to build an interface based on Design drawings or screen captures of an existing interface.

There are two ways in which you can select the background image for a Frame.

**Mac:** ⌘B

**Win:** CTRL+B

From the Design window, do one of the following:

- Select the Frame.
- From the **Modify** menu, choose **Set Background Image**.

OR

- In the Properties panel, click the **Set Background Image** button (see Figure 2-16).

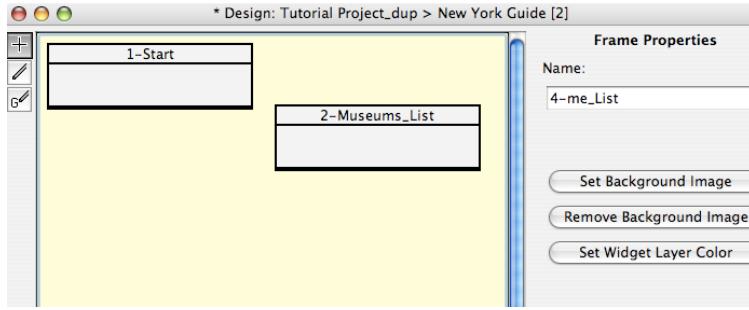


Figure 2-16: Buttons on right pane to set and remove background image.

From the Frame window:

- From the **Modify** menu, choose **Set Background Image**.

Choose the image from the **Open File** dialog box. JPG, PNG, and GIF are acceptable file formats.



You can also paste images from the clipboard into your Frame.

### 2.3.12 Removing the Background Image of a Frame

There are several ways to remove the background image of a Frame.

**Mac:** ⌘B

**Win:**

CTRL+SHIFT+B

From the Design window, do one of the following:

- Select the Frame.
- From the **Modify** menu, choose **Remove Background Image**.

OR

- In the Properties panel, click the **Remove Background Image** button (see Figure 2-16).

From the Frame window:

- From the **Modify** menu, choose **Remove Background Image**.

### 2.3.13 Importing Frame Background Images

You can import an entire folder of images for use in your Frames. Each image will become the background of its own Frame. The Frame's title will be the same as the corresponding file name, without the extension.



Only JPG, PNG, or GIF are acceptable file formats for CogTool.

To import background images:

- Open the Design window.
- From the **Modify** menu, choose **Import Background Images**.
- Navigate to the folder with the images.
- Click **Choose**.



If the Design contains only the default Frame, the default Frame will be deleted when the images are imported.

The imported Frames will be stacked on top of each other. In order to see them properly, you will need to move the Frames apart.

### 2.3.14 Bringing a Frame to the Foreground

To bring your open Frame to the foreground:

- From the **Window** menu, choose the desired **Project**.
- From the drop-down menu, choose the **Frame**.

### 2.3.15 Closing the Frame Window

**Mac:** ⌘W

**Win:** CTRL+W

To close a Frame:

- From the **File** menu, choose **Close**.

## 2.4 Widgets

### 2.4.1 What is a Widget?

A Widget is an element on a Frame with which a user may interact. You will want to define a Widget for each interface element with which a user will interact when performing a Task (see **Section 3.1 Tasks and Task Groups**), such as buttons, check boxes, menus, etc. There are no limits to the number of Widgets that can be placed on a single Frame.



You will not need to create a Widget for every element on the original interface. You will only need the elements for performing the Tasks you want to investigate.

Widgets appear as colored boxes over the background image or on the Frame. The Widget is selected when white boxes appear in the corners of the Widget (see Figure 2-17).

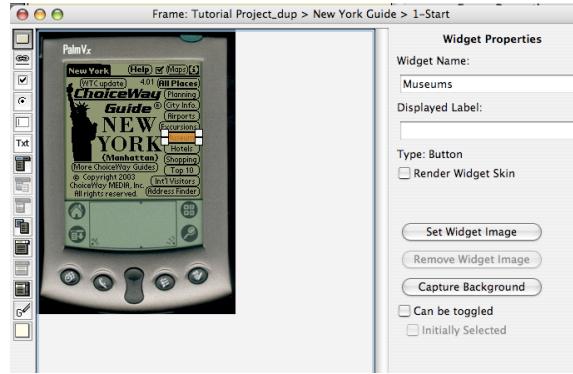


Figure 2-17: The Widget is selected in this example.

Widgets can have Transitions assigned to them in the Design window. Each Transition may link to a new Frame or the same Frame (see **Section 2.5 Transitions** for more information).

### 2.4.2 Types of Widgets

Think carefully when you choose the Widget for your Frame because there will be implications for your Demonstrations and modeling. Research has shown that people take different amounts of time to use different Widgets. Choosing the appropriate Widget has an impact on your predictive modeling.



A Widget cannot be changed to a different type once it is created.  
Think carefully about your Design, needs, and users before making a decision about Widget type.

You may use either a standard Widget or a custom Widget.

**Standard Widgets** are the default, and they assume you want standard Widget behavior and layout. **Custom Widgets** allow you to use non-standard behavior and layout. Once you have created a Widget, you cannot change its type.

You can change the placement, sizing, naming, and appearance of a Widget after creating it (see **Appendix B: Types of Widgets** for a complete list of Widget types).

### 2.4.3 Creating a New Widget

After you have decided which Widget to use and whether your interface will require standard or custom mode, you can create your Widget.

**Mac:** ⌘I

**Win:** CTRL+SHIFT+I

To create a Widget, do one of the following:

- From the toolbar, choose the Widget.
- Drag a rectangle across an area of the open Frame.

OR

- From the **Create** menu, choose **New Widget**.



When you use the New Widget command from the menu or short-cut keys, a square Widget will be placed in the upper left corner of the Frame. You can move and resize the Widget as appropriate for your interface.

When a Widget is created, a white strip appears in the middle of the Widget so that you may type the label to be displayed on the Widget. The Widget will return to a solid color when the Return key is pressed or a mouse click occurs.

You can move the Widget by clicking on its body and dragging it to the desired location. You can resize the Widget by clicking in one of the boxes in the corners of the selected Widget and dragging to the appropriate size (see Figure 2-18).

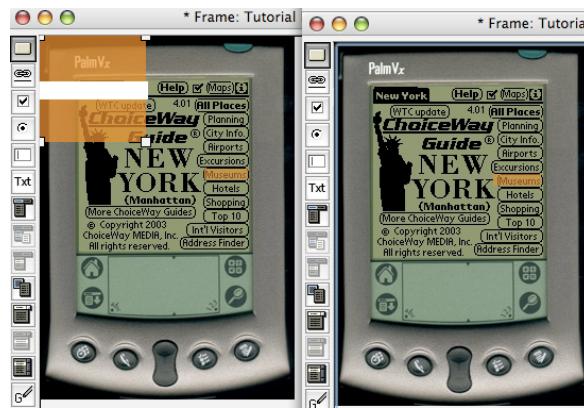


Figure 2-18: Widget created by the click-and-drag method and menu command: The Widget label can be typed in the white strip.



When you have two actions performed in a row on the same interface Widget, you will want to identically place and size the Widget. To do this, create the Widget in one Frame, copy it, and paste it into all the Frames where the Widget appears.

#### 2.4.4 Naming a Widget

When you create a Widget, a default name is inserted into the **Widget Name** field, located in the Widget Properties to the right of the Frame. You should change this default to a more meaningful name. Each Widget on a Frame must have a unique name.

You many use alphanumeric characters, spaces, dashes, and underscores to name your Widget (see Figure 2-19).

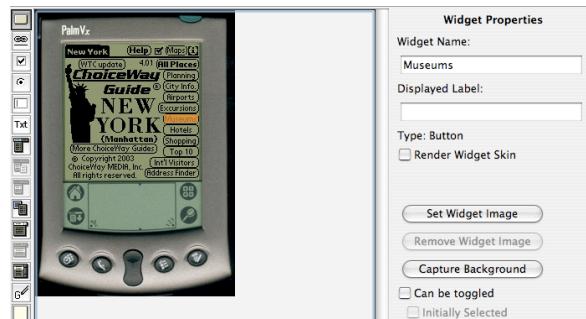


Figure 2-19: This Widget has been named “Museums.”



Some words have special meaning and are considered terminators of a Task. Special meaning words are: cancel, yes, no, exit, abort, quit, save, done, and ok. Name the Widget with the display name that would appear to the user in these cases.

## 2.4.5 Displaying the Widget Label

If the background image of a Frame does not provide enough information to identify the purpose of the Widget, you can provide a label for the Widget. The text will appear within the highlighted area of the Widget when the Frame is displayed.

To label a Widget, do one of the following:

- Type the name in the white box displayed when the Widget is created.

OR

- Open the Frame.
- Click on the Widget.
- In the Properties area to the right of the Frame, go to the **Displayed Label** field.
- Enter the text to be displayed.

## 2.4.6 Setting a Widget’s Appearance

### 2.4.6.1 Setting a Widget Layer Color

A Widget’s default color state is orange. Depending on a Frame’s background color, orange may not always be visible.

**Mac:**  $\text{⌘C}$

**Win:**  
CTRL+SHIFT+C

To change the color of all the Widgets in a Frame:

- From the **Modify** menu, choose **Set Widget Layer Color**.

### 2.4.6.2 Rendering a Widget Skin

A Widget skin can be rendered to resemble objects, such as buttons, on different systems (Mac OS X, Windows XP, and Palm PDA).



CogTool does not have the full set of images for the Mac OS X, Windows XP, and Palm skins. When you choose to render a skin from one of these platforms, you may experience varying results.

To render a Widget skin:

- In the Widget Properties pane, select **Render Widget Skin**.
- From the **Modify** menu, choose **Set Skin**.



Once the Widget skin is set, it will be used for all rendered Widgets.

The available Widget skins are as follows:

**Wireframe:** Each Widget is outlined in black. Each Widget retains its original color and continues to be transparent.

**Mac OS X, Windows XP, Palm:** Each Widget has the appearance of the corresponding interactive Widget. All skins except the wireframe are opaque.

Widget Type	Toolbar Icon	Not Rendered	Wire Frame	MacOSX	Windows XP	Palm
BUTTON		Button	Button	Button	Button	Button
LINK		Link	Link	Link	Link	Link
CHECK BOX		Check Box	Check...	Check...	Check...	Check...
RADIO BUTTON		Radio But...	Radio ...	Radio...	Radio...	Radio...
TEXT BOX			Text Box			
TEXT		Text	Text	Text	Text	Text
PULL-DOWN LIST		PD List	PD L... ▾	PD List ▾	PD List ▾	PD List ▾
PULL-DOWN ITEM		PD Item	PD Item	PD Item	PD Item	PD Item ▾
LIST ITEM		List Item	List Item	List Item	List Item	List Item ▾
CONTEXT MENU		Context ...	Context...	Context...	Context...	Context...
MENU		Menu	Menu	Menu	Menu	Menu
SUBMENU		Submenu >	Sub... ▶	Sub... ▶	Sub... ▶	Sub... ▶
MENU ITEM		MenuItem	Menu It...	Menu It...	Menu It...	Menu It...
GRAFFITI		Graffiti	Graffiti	Graffiti	Graffiti	Graffiti
NON-INTERACTIVE		Non-Inte...	Non-Inte...	Non-Inte...	Non-Inte...	Non-Inte...

Figure 2-20: Rendered versions of Widgets

#### 2.4.6.3 Setting an Image for a Widget

Widgets can have an image. This is useful for creating a new layout that is not based on an existing system.

To set an image for a Widget:

- Open the Frame.

- Select the Widget.
- In the Widget Properties, click **Set Widget Image**.
- From the standard Open File Dialog box, choose your file.



Only JPG, PNG, or GIF are acceptable formats for CogTool.

You may also choose to have the Widget match the background image of the Frame beneath it. You will then be able to maintain a consistent image for the Widget as you move it around the Frame (see Figure 2-21).

To capture the image from the Frame's background:

- Open the Frame.
- Select the Widget.
- In the Widget Properties, click **Capture Background**.



Figure 2-21: The Museum button has been captured for the Widget from the Frame background. The Widget has been moved to the Graffiti area to illustrate how the captured image looks.



You can use this feature to create entirely new layouts from one screenshot of a Frame. Create a Frame with all the desired Widgets and capture the background for each Widget. You can then remove the Frame background or copy and paste the Widgets into a new blank or neutral background.

To remove the image from a Widget:

- Open the Frame.
- Select the Widget.
- In the Widget Properties, click **Remove Widget Image**.



Removing the Widget image will not remove the background image on the Frame, if there is one.

## 2.4.7 Moving a Widget

To move a Widget:

- Open the Frame.
- Select the Widget.
- Click and hold on to the translucent body of the Widget.
- Drag the Widget to the desired location.

You may want to make slight changes in placement to have the Widget precisely placed. You can do this by “nudging” the Widget.

To nudge the Widget, do one of the following:

- From the **Modify** menu, choose **Nudge**.

OR

- While holding the **⌘** key(Mac) or the Control key (Win), use the arrow keys on the keyboard to move the widget.

You can align individual or multiple Widgets precisely on the Frame background.

To align Widgets:

- Open the Frame.
- Select the Widgets by holding the Shift key as you click on the Widgets or drag the cursor over a group of Widgets.



When you select one Widget, you will see white boxes in the corners of the Widget. When you select more than one Widget, the translucency of all those selected will be darker.

- From the **Modify** menu, choose **Alignment**.
- Select the desired alignment.

## 2.4.8 Layering Widgets

In CogTool, objects can be drawn on top of each other. Widgets can partially or fully occlude other Widgets. You can change the depth of overlapping Widgets with the standard drawing program commands to “Bring to Front,” “Bring Forward,” “Send Backward,” and Send to Back.”

To layer Widgets:

- Select the Widgets.
- From the **Modify** menu, choose **Layering**.



Drawing overlapping Widgets has implications for defining Transitions and demonstrating Scripts (see **Chapter 3: Quantitative Analysis** for more information).

## 2.4.9 Modifying Widgets

To resize a Widget:

- Open the Frame.
- Select the Widget.
- Resize by clicking-and-dragging in the small white boxes at the corner of the Widget.

**Mac:** ⌘R

**Win:** CTRL+R

To rename a Widget:

- Open the Frame.
- Select the Widget.
- In the Widget properties panel, edit the text for the **Widget Name**.

**Mac:** ⌘⌫

**Win:**  
CTRL+DELETE

To delete a Widget:

- From the **Edit** menu, choose **Delete**.

## 2.4.10 Copying Versus Duplicating a Widget

In CogTool, you can copy or duplicate a Widget.

When you copy or cut a Widget, the Widget is placed in the clipboard and can be pasted into the same Frame or another Frame. The properties of the Widget are copied, but Transitions are not.

When you duplicate a Widget, it is duplicated into the same Frame as the original Widget. The properties and the Transitions associated with the Widget are also duplicated.



If the same interface Widget appears in different Frames of the Design, it is important that the Widgets be placed and sized identically. This is imperative to ensure the correct prediction of human performance when there are several actions in a row on the same Widget. To make sure that all Widgets are identical, you should create the Widget on one Frame and copy and paste it into all the other Frames.

**Mac:** ⌘C

**Win:** CTRL+C

To copy a Widget:

- From the **Edit** menu, choose **Copy Widget**.

**Mac:** ⌘X

**Win:** CTRL+X

To cut a Widget:

- From the **Edit** menu, choose **Cut Widget**.

**Mac:** ⌘V

**Win:** CTRL+V

To paste a Widget:

- From the **Edit** menu, choose **Paste Widget**.



When you paste a Widget into the same Frame, it is pasted directly on top of the original Widget and is given a new unique suffix

(“MyWidget[1]”). Move the newly pasted Widget off of the original to see both Widgets.

**Mac:** ⌘D

**Win:** CTRL+D

To duplicate a Widget::

- From the **Edit** menu, choose **Duplicate Widget**.



When you duplicate a Widget, an identically sized Widget appears to the lower right of the original Widget and is given a new unique suffix.

## 2.5 Transitions

### 2.5.1 What is a Transition?

A Transition is the link between Frames that form a Task. After you have created Frames and Widgets for your Design, you must show CogTool how those Frames should be linked together in a multi-step Task. A Transition records how a particular action on a particular Widget causes the Design to change from one Frame to another.

CogTool Transitions are created in the Design window and are represented by black arrows. Transitions can go from a Widget or from an input device, such as a keyboard or a microphone, to a different Frame or the same Frame. When a Transition is made to the same Frame, it is called a Self-Transition.

Transition action types are the devices that would generate a Transition. Transition action properties are the specific actions that the device makes, such as a click for a mouse.

You can create multiple Transitions from a single Frame. Depending on the action of a Widget or device, a user may take any number of paths and connect to different Frames.

The available Transition action types consist of the following:

	= default Transition type (goes from source to result)
	= mouse action (e.g. clicks and hover)
	= keyboard
	= touchscreen (e.g. taps and hover)
	= Graffiti®
	= Microphone

## 2.5.2 Creating a New Transition

Before you create a Transition, you must have created the Frames and Widgets for your Design.

To create a Transition:

- Open the Design.
- From the toolbar on the left side of the window, choose the Transition type. In most cases, you will want the default Transition type.



Transitions will only be available for the input devices that were selected at the time the Design was created. You cannot change your selection of devices after creating the Design. Make sure to select all devices that will be needed for your Design.

- Click the Widget or keyboard device and drag the mouse to the destination Frame. The Frame will be highlighted when you have successfully made the connection with the Transition.



A black arrow will appear from the Widget or keyboard device to the destination Frame.



If you drag a Transition from a Widget into open space outside of the source Frame, a new Frame will be created as the destination of the Transition.

At times, you will want to specify a Transition other than the default Transition. For example, you may have a device with a touchscreen or Graffiti® Widget. In those cases, you will want to select the Transition appropriate to your device and needs.

After you release the mouse button to complete a Graffiti® Transition, a dialog box will appear. Enter the Graffiti® gestures that will initiate the Transition (see Figure 2-22).



When the Graffiti® gesture is a command, check the box for **Is Command**.

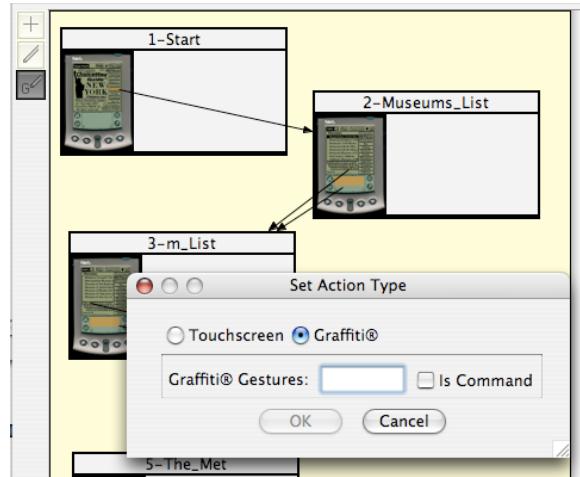


Figure 2-22: The dialog box that sets the Graffiti® action type

To create a Self-Transition:

- Click the Widget that initiates the Transition.
- Drag and release the mouse just outside of the Widget, but in the same Frame.

 A black arrow will appear from the Widget or keyboard device to the edge of the Frame.

 If you forget to create a Self-Transition, you can still insert one when you demonstrate a Script. For more information, see **Section 3.2 Scripts**.

 You cannot create two Transitions from the same source with the same action properties because it is impossible for a Design to be in two different states due to the same action. If you try to do this, an error message will appear. You will have to select different action properties or cancel the Transition creation.

### 2.5.3 Editing a Transition

Transitions have properties that are initially set using the toolbar on the left of the Frame window. These properties can be further configured in the panel to the right of the window. The action type of a Transition can be changed.

**Mac:** ⌘E

**Win:** CTRL+E

To edit a Transition:

- Select the Transition.
- From the **Edit** menu, choose **Edit Transition**.
- You can choose from the actions applicable to the devices associated with the Design. For example, if the Design does not have a keyboard device, you will not have an option for a keyboard action type. See **Appendix D: Transition Properties** for a complete list.

### 2.5.3.1 Setting a Delay for a Transition

When you set a delay for a Transition, the delay will be enacted on the Transition to the next Frame.

To set a delay for a Transition:

- Select the Transition.
- In the Transition Properties pane, set the delay time in seconds.

### **2.5.4 Changing the Source and Destination of a Transition**

You can change the source of a Transition to any other source on the Frame that supports the action type of the Transition. A keyboard Transition may be the only source with an action type of keyboard. However, several Widgets on the Frame may have the action type of mouse.

To change the source of a Transition:

- Select the Transition.
- Drag the white square located at the current source to the new source.

 The mouse cursor will change to a crosshair when it is pointing at a Widget that can accept the Transition.

To change the destination of a Transition:

- Select the Transition.
- Drag the white square located at the current destination Frame to the new destination Frame.

### **2.5.5 Deleting a Transition**

To delete a Transition:

- Select the Transition.

 The Transition is selected when a white square appears at the endpoints of the Transition arrow.

Then do one of the following:

- Press the keyboard Delete key.
- From the **Edit** menu, choose **Delete**.

## 2.6 Export Design to HTML

Export Design to HTML converts your Design into HTML pages, which can be viewed from any web browser. This allows you to share your Design with individuals who do not use CogTool or to run simple user tests where the Tasks from your Design can be explored through think-alouds.

To export your Design to HTML:

From the Project window:

- Select the Design to export.
- From the **File** menu, choose **Export**.
- Choose **Export Design to HTML**.

From the Design window:

- From the **File** menu, choose **Export**.
- Choose **Export Design to HTML**.

CogTool saves the exported Design files in a folder of your choosing. These files include HTML pages of each individual Frame from your Design and an index.html page from which you can navigate to these other pages (see Figure 2-23).

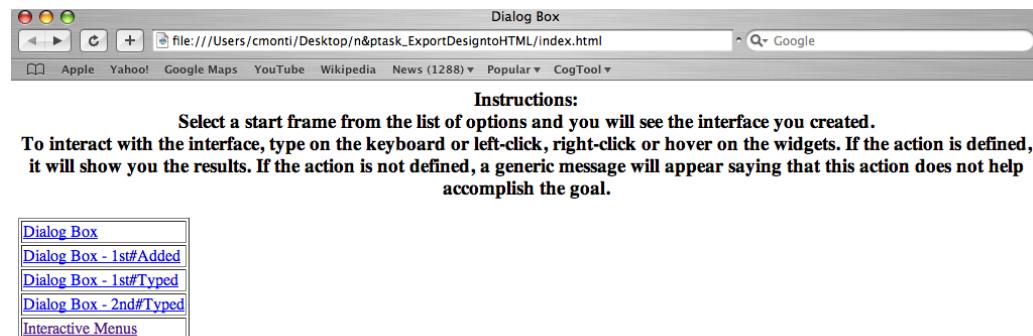


Figure 2-23: An index page of Frames exported from a Design to HTML

You can view the individual Frames from your Design by clicking the Frame name on the index page. If your Frames include interactive Widgets (e.g. drop-down menus, buttons, etc.), you can interact with the Frames directly from their corresponding HTML pages just as you would normally. If you view an exported Frame with drop-down menus, for instance, simply click the menus to choose your desired option (see Figure 2-24).

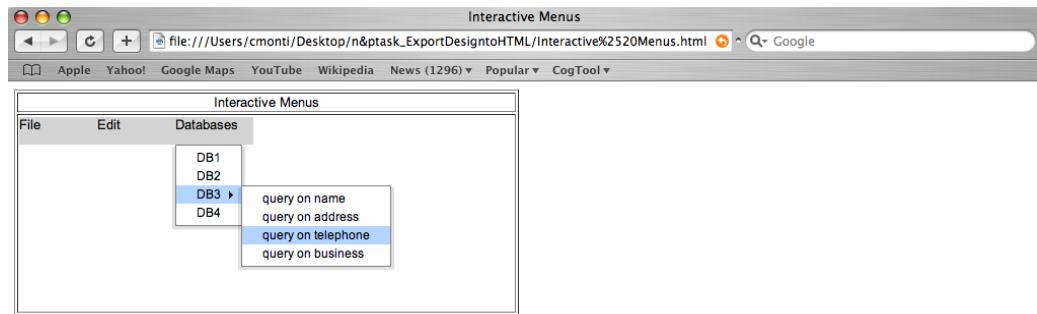


Figure 2-24: An exported Frame in HTML with drop-down menus

Similarly, if you view an exported Frame with a form that includes text boxes and buttons, simply type text in the boxes and click the buttons as you would normally (see Figure 2-25).

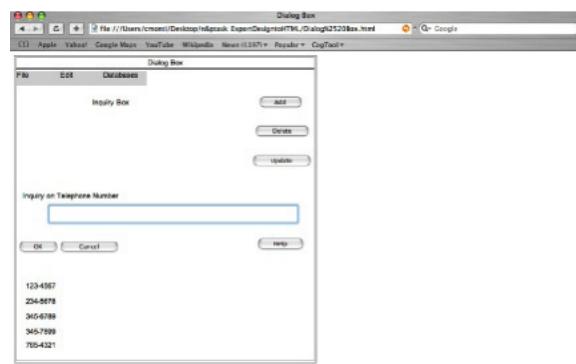


Figure 2-25: An exported Frame in HTML with text boxes and buttons



If you click an area on an HTML page that does not transition to another Frame (e.g., an area that is non-interactive), a pop-up message is displayed to inform you that “This action will not help accomplish your goal” (see Figure 2-26).



Figure 2-26: A pop-up message for a non-interactive area of the page

## CHAPTER 3: QUANTITATIVE ANALYSIS

Once you have created your Design, you can perform quantitative analysis on it. For example, you may have a cell phone screen and want to analyze how quickly certain Tasks can be performed. You may also find that there is more than one way to perform a Task and wonder how quickly an expert user can perform each Task. With this information in hand, you can recommend the fastest interface Design.

In order to perform quantitative analysis with CogTool, you must create Tasks and Scripts. Model visualization illustrates how ACT-R produced predictions on your Design.

### 3.1 Tasks and Task Groups

#### 3.1.1 What are Tasks and Task Groups?

CogTool uses the concept of a Task to represent the purpose of a specific sequence of actions, such as making a purchase on an online website or setting a ring tone on your cell phone.

A Task Group organizes the Tasks and computes statistics on the sum, mean, minimum, and maximum prediction times for the Tasks in that group.

#### 3.1.2 Creating a New Task

When you create a Project, a Task named “Task 1” is automatically inserted into the Project.

**Mac:** ⌘T

**Win:**

CTRL+SHIFT+T

To create additional Tasks:

- From the **Create** menu, choose **New Task**.

The new Task will appear below the Task that is currently selected.



When the new Task is created, the Task name is already selected and a replacement name can be entered. The Task name can be anything you wish, but it should be unique in the Task Group and descriptive of the Task. Try to use a name that will continue to apply as the Design evolves over time.

#### 3.1.3 Creating a New Task Group

**Mac:** ⌘G

**Win:**

CTRL+SHIFT+G

To create a Task Group:

- Select the Tasks that should be part of the Task Group.
- From the **Create** menu, choose **New Task Group**.

A Task Group can be comprised of Tasks and other Task Groups. There is no limit to the number of Tasks or Task Groups that may be nested within a Task Group.

### 3.1.4 Copying/Cutting/Pasting a Task or Task Group

**Mac:** ⌘C

**Win:** CTRL+C

To copy a Task or Task Group:

- Select the Task or Task Group. When a Task or Task Group is selected, it is highlighted in solid blue.
- From the **Edit** menu, choose **Copy Task** or **Copy Task Group**.



When you copy a Task or Task Group, it does not copy any Demonstrations or computed execution times associated with the Task or Task Group.

**Mac:** ⌘X

**Win:** CTRL+X

To cut a Task or Task Group:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Cut Task** or **Cut Task Group**.

**Mac:** ⌘V

**Win:** CTRL+V

To paste a Task or Task Group:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Paste**.

A Task or Task Group can be pasted into the same Project or a different Project.



When you paste a Task into a Task Group or Project that already has a Task of the same name, the pasted Task will be given a unique suffix (e.g. “MyTaskGroup[1]”). When you paste a Task Group into a Project that already has a Task Group of the same name, the pasted Task Group will be given a unique suffix.

If the Task name exists in one Task Group but you paste the Task into another Task Group in the same Project, the pasted Task will not be given a unique suffix.

### 3.1.5 Duplicating a Task or Task Group

When you copy a Task, the Task name is copied, but the demonstrations and computed execution times are not. If you want to make an exact copy of the Task, including demonstrations and computed execution times, you should duplicate the Task.

**Mac:** ⌘D

**Win:** CTRL+D

To duplicate a Task or Task Group:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Duplicate Task** or **Duplicate Task Group**.



Duplicated Tasks are placed beneath the Task that has been duplicated. The Duplicated Task is given a unique suffix (“MyTask[1]”).



In CogTool for Windows, you can also duplicate a Task by holding Control while dragging it.

### 3.1.6 Renaming a Task or Task Group

**Mac:** ⌘R

**Win:** CTRL+R

To rename a Task or Task Group:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Rename Task** or **Rename Task Group**.



You may name the Task or Task Group anything you wish. However, you can have only one Task with that name in a Task Group or at the top level of a Project. Make the name meaningful and descriptive of the Task or Task Group.

### 3.1.7 Deleting a Task or Task Group

**Mac:** ⌘⌫

**Win:**  
CTRL+DELETE

To delete a Task or Task Group:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Delete Task** or **Delete Task Group**.



You will be asked to confirm the deletion before the Task or Task Group is removed from the Project.



When you delete a Task Group, you will also delete all Tasks that are within that Group.

### 3.1.8 Changing the Order of Tasks and Task Groups

You can change the order of Tasks or Task Groups by dragging the Tasks or Task Groups to the desired location. To create a hierarchy of Tasks, you can also promote a Task or Task Group, which moves the Task or Task Group up one level in the overall schema, or demote a Task or Task Group, which moves the Task or Task Group down one level. Promotion or demotion can move a Task or Task Group into or out of a Task Group.

To promote or demote Tasks or Task Groups:

- Select the Task or Task Group.
- To promote a Task, press the **Tab** key.
- To demote a Task, press the **Shift** and **Tab** key.

### 3.1.9 Changing the Computed Value for a Task Group

Creating a Task Group allows you to group Tasks together and apply simple computations. You may wish to see the total time for a suite of benchmark Tasks, or you may want to see the minimum, maximum, or mean value for all the Tasks (see Figure 3-1).

To change the computed value for a Task Group:

- Select the Task Group.
- From the **Modify** menu, choose the type of value you wish to compute.

Tasks	New York Guide	All Devices	New York Guide [2]
Lookup the MET	Mean: 15.239 s	Mean: 2.540 s	Mean: 15.239 s
Lookup the MET ~ Graffiti	15.167 s	2.756 s	15.167 s
Lookup the MET ~ Soft Keyboard	15.311 s	2.324 s	15.311 s

Figure 3-1: The computed value has been set to the mean and is displayed for all Designs in the Project.

### 3.1.10 Organizing Tasks

CogTool can make predictions for many Tasks using the same Design. Organizing your Tasks helps to simplify using the results.

#### 3.1.10.1 Categorical Grouping

Tasks can be grouped by category. This allows you to collect all the Tasks that belong to a particular grouping. For example, you may be analyzing login times using a variety of different methods. You can then group all login Tasks together and analyze the mean or the minimum and maximum times. Categorical grouping is also useful when the different aspects of a benchmarking suite are grouped together. In that instance, you may want to show the sum for the group.

#### 3.1.10.2 Task Breakdown Grouping

A long Task, or a Task with multiple segments, can be placed in a Task Group to allow each component to be analyzed separately. You will want to choose to show the sum for such a Task Group to find the total Task time.

#### 3.1.10.3 Alternative Grouping

A Task Group may be nested within another Task Group. As an example, you may want to place a login Task Group within a Task Group representing a benchmark suite. You will want to change the computed value for the Task Group to best address your needs.

## 3.2 Scripts

### 3.2.1 What is a Script?

In order to make a prediction about a Task, CogTool needs to know the specific sequence of user actions that comprise it. This sequence of actions is called a Script, which demonstrates the actions that can be performed with the Design.

### 3.2.2 The Script Windows

When you open a Script window, you will see one of two possibilities: the Select the Start Frame window or the Script Edit window. If you are opening a Project's Script window for the first time, you will see the Select the Start Frame window, which asks you to select the starting Frame for the Script. This window will appear only if a starting Frame has not been selected for the Demonstration; otherwise, you will see the Script Edit window.

#### 3.2.2.1 The Select the Start Frame Window

The Select the Start Frame window has two panes. The left pane lists the available Frames for that Design. The right pane contains a view of the Design window (see Figure 3-2).

If the Design's input devices include the keyboard and mouse or touchscreen, you will have an option to select the Mouse Hand and Initial Hand Location.

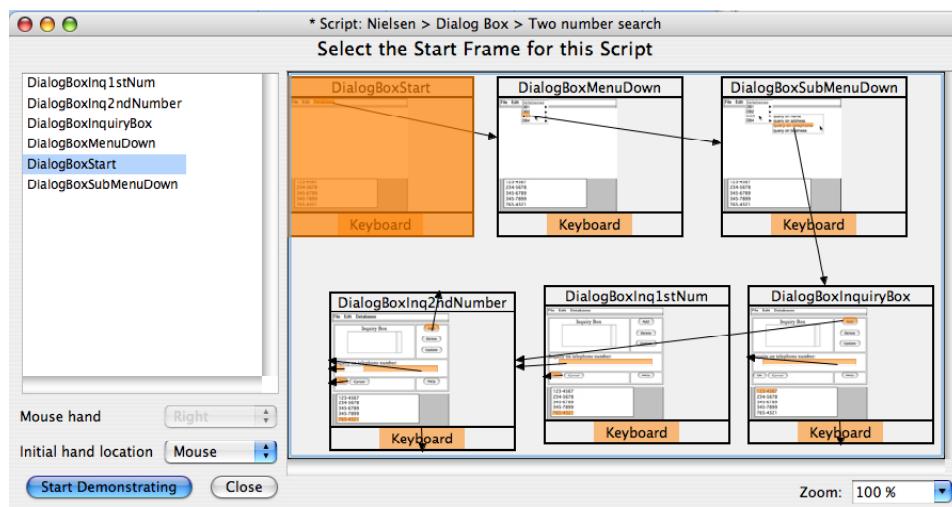


Figure 3-2: The Select the Start Frame window

#### 3.2.2.2 The Script Edit Window

The Script Edit window appears once a starting Frame has been selected. This window has two panes (see Figure 3-3).

The left pane is devoted to the Demonstration of the Task. It shows the current Frame

with its Widgets that are available for interaction. Under the Frame are two buttons used to insert special timing into the Script. One inserts a **Look at** Widget and the other inserts a **Think**.

The right pane shows the Script Step List. The steps that have been demonstrated are listed here. Buttons at the bottom of the right pane allow you to Delete Step and Compute. If the Design's input devices include the keyboard and mouse or touchscreen, you will have an option to select the Mouse Hand and Initial Hand Location.

Details on working with Scripts start with **Section 3.2.5 Demonstrating Actions for a Script**.

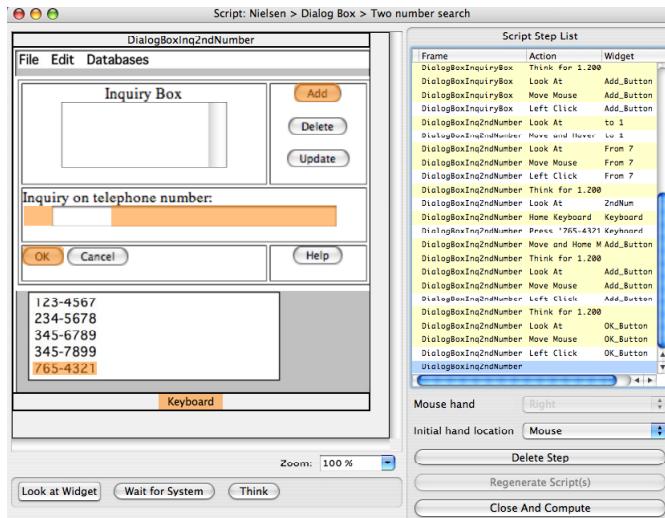


Figure 3-3: The Script Edit window

### 3.2.3 Opening or Editing a Script

**Mac:** ⌘E

**Win:** CTRL+E

To open or edit a Script, do one of the following:

- Select the Task or Task Group.
- From the **Edit** menu, choose **Edit Script**.

OR

- Double-click the Script cell.

### 3.2.4 Selecting a Start Frame for the Script

Since a Script may begin at any Frame in the Design, you must select the Start Frame for each Script. The first time you open or edit a Script, you will be prompted with the Select the Start Frame window.

To select the starting Frame, do one of the following:

- Double-click on the Frame in the right pane.

OR

- Choose the Frame from the list in the left pane.

If the keyboard and the mouse or touchscreen are input devices in your Design, buttons at the bottom of the left pane will allow you to choose the Mouse Hand and the Initial hand location. These options are also available in the Demonstration Script window.

Once the starting Frame has been selected, the Script Edit window will appear. The starting Frame appears in the left pane of the Script Edit window and the current list of steps appears in the right pane.

You can use the same zoom commands that you used when you created and edited the Frames to allow you to see the Frame better.

#### Zoom In

Mac:  $\text{⌘}+=$

Win:  $\text{CTRL}+=$

#### Zoom Out

Mac:  $\text{⌘}-$

Win:  $\text{CTRL}+-$

#### Normal Zoom

Mac:  $\text{⌘}0$

Win:  $\text{CTRL}+0$

#### Zoom to Fit

Mac:  $\text{⌘}/$

Win:  $\text{CTRL}/$

To use the zoom commands, do one of the following:

- From the Modify menu, choose the Zoom option you want.

OR

- Choose your zoom settings from the box at the bottom of the Frame window.



**Normal Zoom** makes the contents of the frame revert to the default size. **Zoom to Fit** makes the contents of a frame fit the available space in the window.



The CogTool for Windows zoom shortcut keys do not follow standard Windows functionality.

### 3.2.5 Demonstrating Actions for a Script

You can record steps for the Script by interacting with the Frame in the left pane as a user would interact with the actual device. If you have added all the Widgets needed to demonstrate the Task, you just need to click on the appropriate Widget to make the action occur. 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 (see Figure 3-4).

Script Step List		
Frame	Action	Widget
1-Start	Think for 1.2 s	
1-Start	Move and Tap	Museums
2-Museums_List		

Figure 3-4: CogTool demonstrating a touchscreen action

You can also demonstrate steps by selecting them from the context menu for a Wid-

get. Some actions (such as keyboard and Graffiti® actions) must be demonstrated by selecting the action from a context menu. If you click in the Graffiti® area of a Frame illustrating a Palm® device, the context menu will appear (see Figure 3-5).

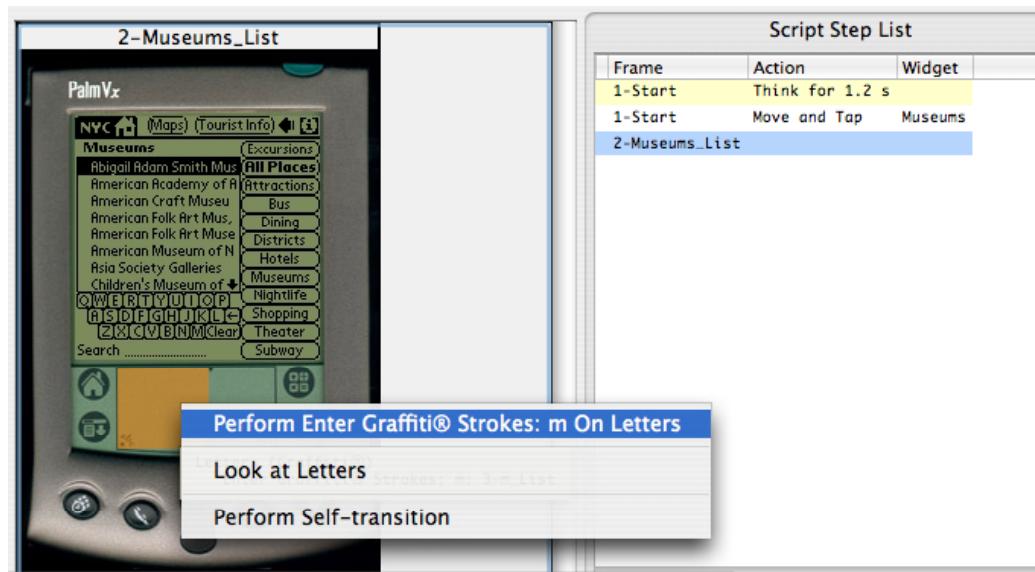


Figure 3-5: Context menu appears for Graffiti® action Widget



If your Frame has overlapping Widgets, a click or double-click will automatically follow the Transition for the topmost Widget. Actions on all occluded Widgets are accessible through the context menu.

### 3.2.6 Editing Demonstration Actions

You may edit steps in your Demonstration by double-clicking them and then editing the duration and label for **Wait** or **Think**. Currently, you can only edit **Wait** and **Think** steps.

### 3.2.7 Deleting Demonstration Actions

It is not advisable to delete arbitrary steps in the middle of a Script. Although CogTool does not prevent you from deleting an arbitrary step, the resulting Script is not guaranteed to produce a psychologically valid prediction.

To delete a demonstrated action:

- Select the last step you demonstrated.
- Click **Delete Step** until you have deleted back to the undesired step.
- Restart your Demonstration from the deleted step.



You will not be able to delete steps that are generated by a Speaker device. If you want to delete a Speaker step in a Script, you must

edit the Frame containing the Speaker information.

### 3.2.8 Reviewing Your Demonstration

#### 3.2.8.1 Reviewing Automatic Script Steps

In the Script Step List (see Figure 3-5), CogTool has recorded two steps in response to the single demonstrated action, clicking on the Museums Button. They are a Think for 1.2 seconds and a Move and Tap action on the Museums Button. The Think step is 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, and then they tap that button. The tap causes the current Frame to change to the Museums List Frame, which is now shown on the left and listed as the last line of the Script Steps List. Automatically inserting such 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, Move-mouse, and Home-hand steps is supported by decades of psychology research and HCI data and is at the core of the research that produced CogTool.

CogTool shades in yellow the steps it automatically generates to distinguish them from steps directly demonstrated or inserted by the analyst, which have a white background. All the yellow generated steps above a white demonstrated step are associated with that demonstrated step. If a demonstrated step is deleted, all generated steps associated with it will be deleted as well.

#### 3.2.8.2 Reviewing the Summed Task Group

You may want to review the sum for a Task Group. When you view the sum, you will see the steps for all the Tasks in the group combined into a single Script Edit window.

To view the sum for a Task Group:

- Double-click the cell which contains the summed value for the group.

#### 3.2.8.3 Reviewing Transitions Defined on a Widget

If you have difficulty remembering which Widgets have Transitions or actions to trigger them, hover the mouse over the Widget to display a tool tip. The context menu for the Widget will also have information about Transitions.

#### 3.2.8.4 Reviewing a Previous Step

When you click on the Script step in the Script list, the associated Frame appears in the left pane of the Script Edit window.

### 3.2.9 Inserting Items into a Script

#### 3.2.9.1 Inserting a Self-Transition

If you forget to insert a Self-Transition into your Design as you demonstrate the Design, you can still add a Self-Transition during the Demonstration process.

To insert a Self-Transition:

- Right-click the Widget.
- From the context menu, select **Perform Self-transition**.

#### 3.2.9.2 Inserting a “Look At” Step

At times, you may want a modeled user to look at a Widget for visual information but not to physically manipulate it. This action does not cause a Transition to be followed. To cause a Transition to be followed, you will need to insert a “Look At” Script step.

To insert a “Look At” step:

- In the Script Edit window, click the **Look At** button.
- Click the Widget at which the user will look. A new “Look At” step is added to the Script list.



The **Look At** button is a toggle button. Once it is clicked, CogTool expects you to select the target Widget for the “Look At.” The only action for a target non-interactive Widget is “Look At.”

#### 3.2.9.3 Inserting or Removing a “Think” Step

In some cases, you may wish to change the way 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.

To remove a “Think” operator:

- Select the line in the Script with the “Think.”
- Click the Delete Step button at the bottom of the pane.

If you believe that you have sufficient justification, you may add additional “Think” operators.

To insert a “Think” operator:

- Select the point in the Script to add the “Think.”
- Click the **Think** button.
- In the dialog box that appears, enter 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.

### 3.2.10 Restarting a Script from a Previous Step

If you have made a mistake in demonstrating a Task, you can return to a previous step and continue again from that point. Click a step in the Script 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. Click **OK** to continue demonstrating the corrected sequence of steps.



It is not always safe to add, remove, or change steps in a Script. Performing these actions may cause ACT-R to hang or for the predictions to be calculated incorrectly. If you want to make changes to the Design or Scripts, delete all the steps and redemonstrate them.

### 3.2.11 Computing a Prediction

Once you have finished creating and editing a Script, click the Compute button at the bottom of the right pane to compute a prediction. The result will appear as a value at the top of the right pane in the Script window (to the right of the “Prediction” label) and also in the Script cell for the Task in the Project window.

The number that appears in these locations is the calculated prediction for execution time. If you do not have numbers in these locations, you might have one of the following:

- Nothing ( ): the cell has not been demonstrated or lacks a Script.
- Double dash (--): the Script has been demonstrated but not computed.
- Question marks (??): there is an invalid step, such as clicking on a Menu Item Widget before clicking on a Menu Widget. The Script steps affected will be colored red and will be marked by an X on the left-hand side of the “Script Step List.” The Script will need to be edited before it can be recomputed (see Figure 3-6).

t > New York Guide > Lookup the MET		
Script Step List		
Frame	Action	Widget
1-Start	Think for 1.2 s	
1-Start	Move and Tap	Museums
2-Museums_List	Move Mouse	Letters
2-Museums_List	Enter Graffiti® Strokes: m	Letters
X 3-m_List	Think for 1.2 s	
X 3-m_List	Enter Graffiti® Strokes: e	Letters
X 4-me_List	Move and Tap	Metropolitan
X 5-The_Met		

Tasks	New York Guide
Look up the MET	??

Figure 3-6: Question marks in the Script cell and red lines in the “Script Step List”

- Double XX (X): a problem exists with a Transition that causes the prediction time for the Script to be greater than 600 seconds (10 minutes). CogTool cannot

compute prediction times greater than 600 seconds. The Script will need to be edited or broken into separate Tasks before it can be recomputed.

### 3.2.12 Recomputing Scripts

When you make changes to your Script, it will automatically regenerate. However, you should always recompute a changed Script to ensure an accurate, updated prediction time.

To recompute a Script:

From the Script window:

- Click the **Compute** button.

From the Project window:

- Select the Script cell.
- From the **Modify** menu, choose **Recompute Script**.



You can recompute all Scripts in a Project window by selecting all the Scripts. To select all the Scripts, from the **Edit** menu, choose **Select All**.



Some changes to a Design can prevent a Script from being recomputed properly. 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, you may need to redemonstrate the Script from the beginning.

Invalid Scripts must be redemonstrated. The red areas in the Script window indicate from which point the associated Script must be redemonstrated.

### 3.2.13 Exporting Scripts to CSV

You may want to export your Script to a CSV, or comma separated value, file to view the steps in a clearer format or to use them in documentation. CSV files can be used in many ways. Usually, you will want to open the CSV file in a spreadsheet program, such as Microsoft Excel.

To export a Script:

- Select the Script cell from the Project window.
- From the **File** menu, choose **Export**.
- Choose **Export Script to CSV**.

The default file name for the exported Script follows the standard of **Project Name\_Design Name\_Task Name.csv**.



If the Script window is already open, you can still export to CSV by going to the **File** menu, choosing **Export**, and then choosing **Export Script to CSV**.

The CSV file contains information on the version of CogTool, the Project and Design names, and all the information from the Script window (the Frames, Actions, Widget Names and Types, and Displayed Labels) (see Figure 3-7).

A	B	C	D	E	F
1 Date and Time:	6/25/08 11:23				
2 Project Name:	Tutorial Project				
3 Design Name:	New York Guide Tutorial				
4 Task Hierarchy:	Lookup the MET	Lookup the MET - Graffiti			
5					
6 Frame	Action	Widget-Name	Displayed-Label	Widget-Type	
7 1-Start	Look At	Museums		Button	
8 1-Start	Think for 1.200 s				
9 1-Start	Move and Tap	Museums		Button	
10 2-Museums_List	Move Mouse	Letter Input Area		Graffiti/E	
11 2-Museums_List	Enter Graffiti/E Strokes: m	Letter Input Area		Graffiti/E	
12 3-m_List	Enter Graffiti/E Strokes: e	Letter Input Area		Graffiti/E	
13 4-me_List	Move and Tap	Metropolitan Museum		List Box Item	
14 5-The_Met					
15					
16					
17					

Figure 3-7: The information from the Script window exported into Microsoft Excel



When you export a Script from a Task Group, the Task Hierarchy is displayed under the Design name (see Figure 3-7).



When you open the file in a spreadsheet application, such as Microsoft Excel, symbols for the semantic keys (e.g.  $\uparrow$  for Shift) may not be displayed. In these cases, the symbols are replaced by the key name, which is capitalized and in parentheses (e.g. (SHIFT)) (see Figure 3-8).

A	B	C
1 Date and Time:	6/27/08 11:06	
2 Project Name:	Untitled Project 3	
3 Design Name:	Design 1	
4 Task Hierarchy:	Task 1	
;		
5 Frame	Action	Widget-Name Disj
6 Frame 1	Think for 1.200 s	
7 Frame 1	Type ('SHIFT') <sup>1</sup>	Keyboard
8 Frame 1	Think for 1.200 s	
9 Frame 1	Get	
10 Frame 1	Microphone	
11 Frame 1	Think for 1.200 s	
12 Frame 1	Type ('CTRL') <sup>1</sup>	Keyboard
13 Frame 1		

Figure 3-8: An exported Script with semantic keys: For a complete list of the semantic keys and their corresponding symbols, see **Appendix E: Semantic Keys and Symbols**.

### 3.2.14 Exporting All Results to CSV

You can export all the results from demonstrated Scripts to a CSV file. You might choose this function to manipulate (add, subtract, etc.) numbers in Microsoft Excel or to more easily track and work with multiple Tasks instead of viewing them in a large CogTool file.

To export all results to CSV:

- Display the Project window.
- From the **File** menu, choose **Export**.

- Choose **Export All Results to CSV**.

The default file name for the exported script follows the standard of **Project Name – Date\_Time.csv**.

The CSV file contains information on the version of CogTool, the Project Name, the date and time stamp, and all the information from the Project window (the Tasks and the computed times, in seconds) (see Figure 3-9).

A	B	C	D
1	Project:	Tutorial Project	
2	Date and Time:	6/26/08 14:23	
3	All times are in seconds		
4	Tasks	New York Guide Tutorial	
5	Lookup the MET	Min: 4.052	
6	Lookup the MET - Graffiti	4.052	
7	Lookup the MET - Soft Keyboard	5.82	
8			
9			
10			
11			

Figure 3-9: The information from the Project window exported into Microsoft Excel

### 3.3 Model Visualization

The underlying computation cognitive engine that makes the predictions in CogTool is called ACT-R and is very complex. To learn more about it, visit the ACT-R website: <http://act-r.psy.cmu.edu/>.

We have built a visualization tool into CogTool so that you can see what ACT-R does to produce the predictions. Although this visualization is meant for a 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:

From the Project window:

- Select the Script cell of interest.
- Right-click the cell to bring up the context menu.
- Choose **Show Model Visualization** (see Figure 3-10).

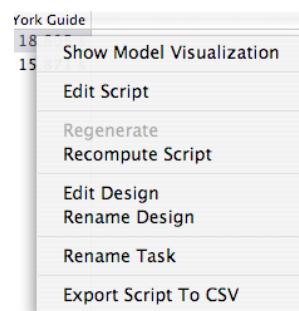


Figure 3-10: Accessing Model Visualization

From the Script window:

- Click the Show Visualization button (see Figure 3-11).

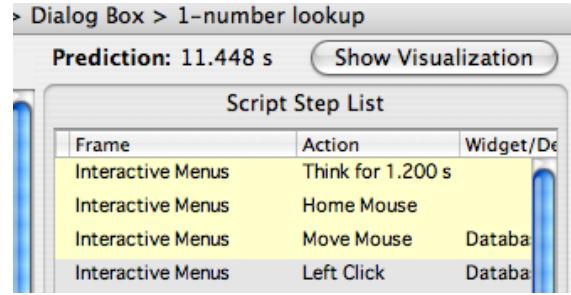


Figure 3-11: The Show Visualization button in the Script window

You can open as many visualization windows as you wish and align them to compare the execution traces and see where their times differ.

The Visualization window has two panes (see Figure 3-12). In the left pane is a timeline showing the different activities ACT-R goes through to make the predictions. The left pane is split into two, with the top portion of the pane presenting the entire model in miniature and the bottom portion showing a blow-up of a portion of the timeline selected from the top portion. The right pane shows either single operator information or a textual trace of the model's run, depending on whether the **Trace** or **Operator Info** tab is selected.

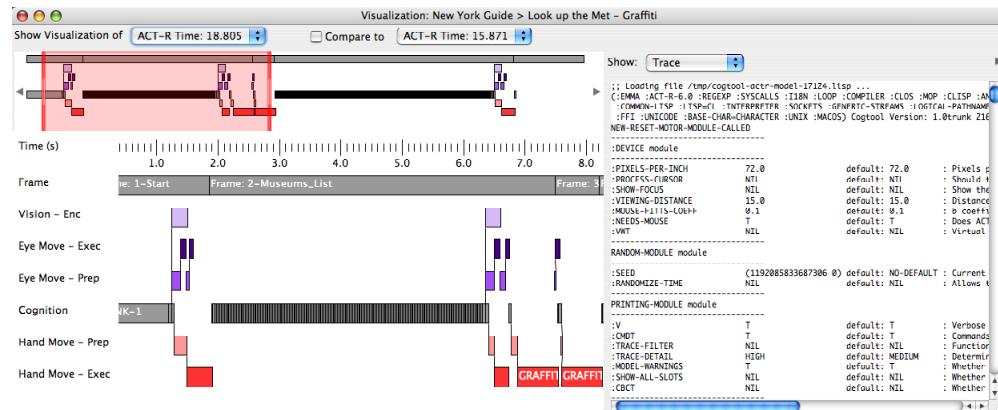


Figure 3-12: The Model Visualization window



The light red highlighted area in the overview pane represents the information shown in the detail pane.

In the left pane, time runs from left to right and the widths 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 operators are grouped into three categories:

- Eyes - 1st, 2nd, and 3rd row (shades of purple)
  - \* Vision - Enc (light purple) - represents the eyes seeing objects on the Frame

- \* Eye Move – Exec (darkest purple)
- \* Eye Move – Prep (mid purple)
- Cognition - 4th row (gray) - represents the elements of a production system, the type of cognitive engine that ACT-R employs. The production system represents the thoughts the model has when performing the Task. The long boxes in this row are the Think operators. Short boxes are other types of cognitive operators that initiate motor movements, and motor and visual attention shifts.
- Hands - 5th & 6th row (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 manual.
  - \* Hand Move – Prep (light red)
  - \* Hand Move – Exec (darker 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 this example, only horizontal movements of the stylus are shown, since a Palm® device is illustrated (see Figure 3-12).

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



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 does not 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 seconds into the Task (see Figure 3-12). However, it is not visible until about 5.7 seconds 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 seconds and 5.7 seconds. This pattern of operators represents the eyes trying to see the next button, but since the Frame is not visible, the vision system returns a “failure” signal to cognition. As soon as the Frame is visible, the vision operator is much longer, as it finds the next button to press and returns its location to cognition.

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 light 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 clicking the red area and dragging it, or you can move one end by clicking and dragging a vertical end bar. You can also scroll smoothly by pressing down on the arrows ( $\blacktriangleleft$  or  $\triangleright$ ) at the edges of the overview

timeline, or you can step through by clicking on those arrows.

The figure below (see Figure 3-13) shows a comparison between two different models for a Palm® device. One shows the tap/keyboard method while the other shows Graffiti® gestures. 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 seconds and 7.2 seconds), while 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.

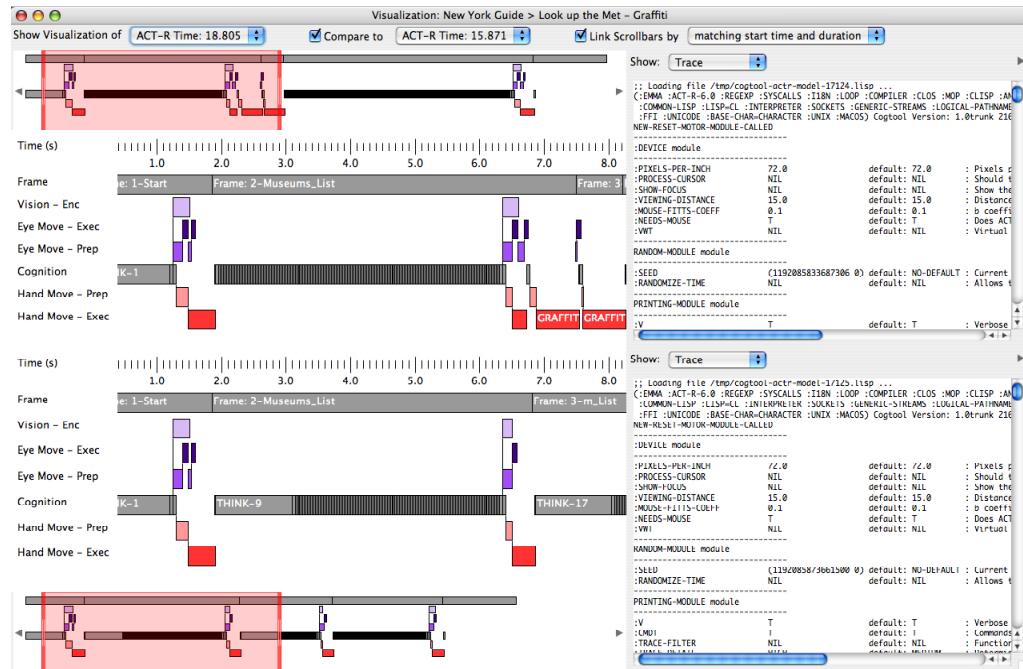


Figure 3-13: Comparison of Graffiti® to Soft-keyboard methods in ACT-R visualization windows

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 compute.



Deleting and inserting Think steps and System Wait steps are the only safe modifications you can make to Scripts without redemonstrating.

## **APPENDIX A: IMPROVEMENTS TO CogTOOL SINCE THE LAST PUBLIC RELEASE**

The last public release of CogTool was version 1.018b, released in February 2007. The new public release of CogTool has improved functionality in the following areas:

- The speaker output device represents audible warnings and feedback.
- The microphone input device represents voice commands from the user.
- Individual cells can be selected in the Project window.
- Designs can be opened by double-clicking the Design name.
- Tasks and Designs can be moved around in the Project window by dragging-and-dropping.
- Tasks can be promoted and demoted by pressing the Tab key when the Task name is highlighted.
- Hovering tool tips are available.
- A status bar at the bottom of the CogTool window records the last command processed or the text that would appear in a tool tip.
- Designs can be exported to HTML.
- All Script results can be exported to CSV.
- Visualizations of the Script results can be viewed.

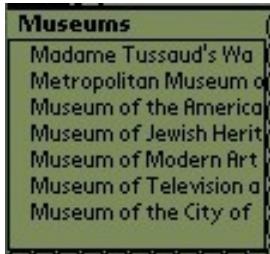
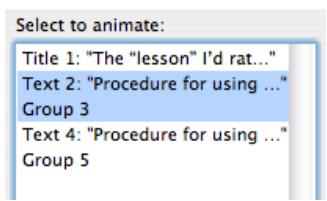
## APPENDIX B: TYPES OF WIDGETS

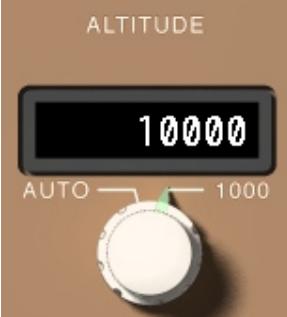
Widget	Examples	Use this Widget...
Button	     	<p>...anytime a Widget looks like a button as defined by current conventions. The button can be a hardware button (like the Palm® Datebook button to the left), a static button (like the Palm® calculator button), or a button that appears dynamically on a screen (like the New and Sign In buttons). A button Widget is also used for things like the Close, Hide, and Full Size buttons that control Mac windows; the back, forward, reload, stop, and home buttons on the Firefox toolbar; and tabs in Firefox (three of which are shown). The Close button within each Firefox tab would also be represented by a button Widget.</p>
Link	 <a href="http://www.cogtool.org">http://www.cogtool.org</a> <u>Carnegie Mellon University</u> <ul style="list-style-type: none"> <li>● <b>About Us</b></li> <li><b>Contact Info</b></li> <li><b>Facilities</b></li> <li><b>Visit HCII</b></li> </ul> <b>What's New</b>  <p>Available for order: Updated 2006 Research-Based Web Design &amp; Usability Guidelines Book</p>	<p>...anytime a Widget represents a link on a web page. This can be a URL, a text link, or anything else that acts as a link in a web page. For example, both the blue words and the picture of the book on the “What’s New” section of Usability.gov are links.</p>

<b>Checkbox</b> 	<b>Private:</b> <input checked="" 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.	...anytime there is an interactive box that can be toggled on or off (checked or not checked) independently of any other checkboxes. The top example 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®.
<b>Radio Button</b> 	  	...anytime 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 they may also take other shapes (like the days of the week in the Palm® Date Book, to the left).
<b>Text Box</b> 	Username: <input type="text"/> Password: <input type="password"/>  <b>Every: ..... 1 Day(s)</b>	...anytime there is a place to enter text (letters or numbers). These can have different looks, such as the Username and Password boxes on Gmail® or the text area for the number of days on the Palm®.  NOTE: A text box is different from the text <i>inside</i> the text box. The only interaction with a text box Widget is clicking on it to put the focus in the text box. If you want to represent the text inside a text box to highlight it, insert into it, or otherwise edit it, put a separate text Widget inside the text box Widget.

Text	<p>The screenshot shows a Microsoft Word window with a toolbar at the top. Below the toolbar is a text box containing the text "This text can be selected." A mouse cursor is positioned over the text, indicating it is selected.</p>	<p>...for any text that can be selected and copied, deleted, pasted over, or typed over. It can be the text in a text box, the text someone is editing in a word processor, text on the web you want to copy, etc. The text Widget is usually used when a picture of the text you want to edit is included in the Frame. Then the text Widget goes directly over the text you want to manipulate. For example, suppose 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 another text Widget on the last word, because these would be the targets of your mouse actions.</p>
Menu	<p>The screenshot shows a Palm OS-style menu bar with three items: "Record," "Edit," and "Options." "Edit" is currently highlighted. To the left of the menu bar is a small icon of a computer monitor.</p>	<p>...for the menu headers. In the Palm® menu to the left, “Record,” “Edit,” and “Options” would be menu Widgets. In the Microsoft Word™ menu below in the next section, “Insert,” “Format,” and all the words across the top would be menu Widgets.</p>
Submenu	<p>The screenshot shows a Microsoft Word menu bar with several items: Insert, Format, Font, Tools, Table, Window, Work, and F. The "Break" option under the "Insert" menu is expanded, showing sub-options like "Page Break," "Section Break (Next Page)," "Section Break (Continuous)," "Section Break (Odd Page)," and "Section Break (Even Page)."</p>	<p>...for menu items that produce another menu. The Palm® menu in the previous section does not have a submenu Widget. The Microsoft Word™ menu on the left would have two submenu Widgets, “Break” and “AutoText.”</p>

<h3>Menu Item</h3> 		<p>...for items that can be selected from a menu (and do not produce other menus). In the Palm® menu to the left, “Undo,” “Cut,” and all the words below them would be menu item Widgets. In the Microsoft Word™ menu in the previous section, “Page Break,” “Column Break,” and all the words below them would be menu items.</p>
<b>Using Menu, Submenu, and Menu Item:</b> When demonstrating a Task, you can only interact with a submenu item Widget or a menu item Widget after having interacted with its associated menu Widget. This is true 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 submenu item or a menu item Widget without first interacting with its associated menu Widget, CogTool will give you a warning message.		
<h3>Context Menu</h3> 		<p>...for a menu that is accessible by right-click (PC) or CTRL-click (Mac).</p>
<h3>Pull-Down List</h3> 	  	<p>...anytime 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 Widget for the displayed item (before the list is pulled down). An example is the list of types of “To Do” items to display on the Palm®. In the example to the left, the “All” with the down arrow indicates a pull-down list; if this were the image in the Frame, the arrow+All would be a pull-down list Widget. In the example from Apple’s Mail, the object with the server_name+arrows would be a pull-down list Widget.</p>

<b>Pull-Down Item</b>	 	<p>...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 Frame, each of the items would be pull-down item Widgets. Likewise, all the items in Apple's Mail pull-down would be pull-down item Widget.</p>
<p><b>Using Pull-Down Lists and Items:</b> When demonstrating a Task, you can only interact with a pull-down item Widget after having pulled down the list by interacting with its associated pull-down list Widget. This is true because a user cannot interact with list items 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 Widget without first interacting with the associated pull-down list Widget, CogTool will give you a warning message.</p>		
<b>List Box Item</b>	  	<p>...when there is a visible list of items, from which one or more can be selected, that does not have to be pulled down or popped up. 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™.</p>

Graffiti®			<p>...to designate an area on a touchscreen that can interpret Graffiti® strokes. To mock-up a Palm® interface, we suggest using one Graffiti® Widget 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 the touchscreen can also receive Graffiti® strokes and can be designated with Graffiti® Widgets.</p>
Non-Interactive			<p>...where a user might have to look to get information but cannot manipulate that information from the Widget. The 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 Widget. The Task could involve looking at that Widget, but the display cannot be changed directly (the knob below it would have to be defined as an interactive Widget and separately manipulated).</p>

## APPENDIX C: STANDARD VERSUS CUSTOM WIDGETS

There are two modes for creating Widgets: standard and custom. Standard mode is the default and assumes the user wants standard Widget behavior and layout, as presented by prevalent user interfaces such as Macintosh and Windows. This assumption lets CogTool aid you in specifying the layout and behavior of standard Widgets. If you wish to present Widgets using non-standard behavior and/or layout, then the custom mode should be selected. This version of CogTool offers a few Widgets in standard mode, but some are no different in standard mode than they are in custom mode. The following section helps you choose which Widgets creation mode is appropriate for each Widget.



The following section mentions details of CogTool use, such as specifying behavior, demonstrating Tasks, setting skins, and rendering skins, which are defined elsewhere in this document.

### Button Widget

Button Widget layout and behavior are the same in either mode.

The screenshot shows the CogTool interface with the 'Widget Properties' pane open for a 'Button' widget named 'OK\_Button'. The pane includes fields for 'Widget Name' (set to 'OK\_Button'), 'Displayed Label' (set to 'OK'), and 'Type' (set to 'Button'). It also contains several buttons and checkboxes: 'Render Widget Skin' (unchecked), 'Set Widget Image' (button), 'Remove Widget Image' (button), 'Capture Background' (button), 'Can be toggled' (unchecked), and 'Initially Selected' (unchecked). To the left of the pane, a preview window shows a simple orange rectangular button labeled 'OK' with small square handles at its corners.

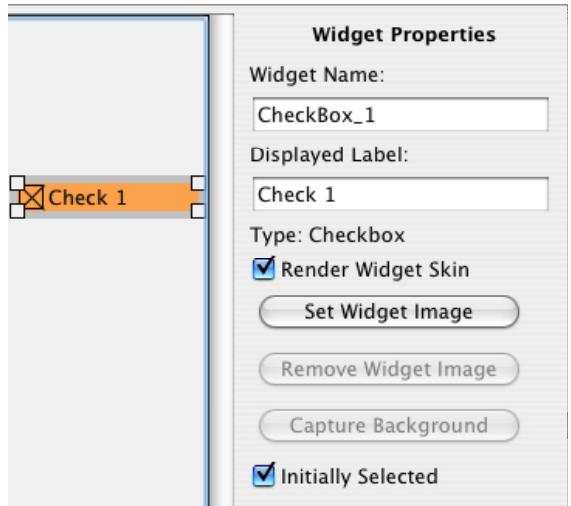
The only difference you will see between standard and custom mode for a button Widget is that the Widget Properties pane for a standard button has two checkboxes at the bottom (can be toggled and initially selected). These checkboxes do nothing in this version of CogTool and should be ignored.

### Link Widget

Link Widget layout and behavior are the same in either mode.

### Checkbox Widget

Standard checkboxes display check and uncheck behavior automatically. If you create a checkbox in custom mode you must explicitly specify that behavior and use different background images to get the same effect.



**Widget Properties**

Widget Name: CheckBox\_1

Displayed Label: Check 1

Type: Checkbox

Render Widget Skin

**Set Widget Image**

**Remove Widget Image**

**Capture Background**

Initially Selected

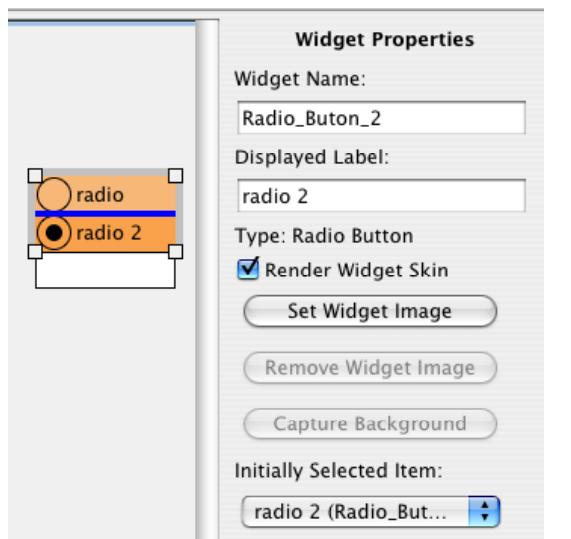
To get the biggest benefit, create a standard checkbox Widget, render its skin, and set its skin to Wireframe (from the **Modify** menu, choose **Set Skin**). Now you can indicate whether it is initially selected or not in this Frame by checking the checkbox at the bottom of the Widget Properties pane. You do not need to specify the behavior of this checkbox further. When you demonstrate your Task and you left-click on this Widget, it will automatically toggle between checked and unchecked.



Standard checkboxes only display automatic behavior when the Widget is rendered and the skin is set to Wireframe (from the **Modify** menu, choose **Set Skin**). Other skins or unrendered standard checkboxes will perform the automatic behavior, but because they do not change visually, you may not notice that behavior in a Demonstration.

## Radio Button Widget

Radio buttons belong to a group in a logical sense, in that only one member of the group can be selected at a time, as well as in a graphical sense, in that they are almost always arranged in rows and/or columns. When creating a standard radio button Widget, CogTool provides support for building a group of radio buttons in both of these senses. It allows you to easily create aligned columns and rows, and it only allows one of the radio buttons to be selected at a time. If you create radio button Widgets in custom mode, you must explicitly specify that correct behavior, use different background images, and carefully align multiple Widgets to get the same effect.



**Widget Properties**

Widget Name: Radio\_Buton\_2

Displayed Label: radio 2

Type: Radio Button

Render Widget Skin

**Set Widget Image**

**Remove Widget Image**

**Capture Background**

**Initially Selected Item:**

radio 2 (Radio\_But...)

To get the greatest benefit, create a standard radio button Widget, render its skin, and set its skin to Wireframe (from the **Modify** menu, choose **Set Skin**). Now you can select which of the radio buttons in the group is initially selected. You do not need to specify the behavior of these radio buttons further. When you demonstrate your Task and you left-click on one of the unselected radio buttons in the group, it will automatically change to the selected state, and whichever one was selected before will automatically change to the unselected state.



Standard checkboxes only display automatic behavior when the Widget is rendered and the skin is set to Wireframe (from the **Modify** menu, choose **Set Skin**). Other skins or unrendered standard checkboxes will perform the automatic behavior, but because they do not change visually, you may not notice that behavior in a Demonstration.

## Text Box Widget

Text box Widget layout and behavior are the same in either mode.

## Text Widget

Text Widget layout and behavior are the same in either mode.

## Menu Widget

Hierarchical menus are complex groups of interactive Widgets that CogTool supports very well in standard creation mode. In custom mode, you have to use background images of menu systems expanded in various ways to depict different selections of items and sub-items. You also have to make sure that Demonstrations do not interact with a menu item or sub-item before it would be visible in a real interface. You cannot select a menu item before its menu header has been selected. Keeping track of this behavior was the biggest source of errors in previous versions of CogTool.

		<p>Because CogTool will automatically make submenus and items as you type in the hierarchical menu structure, the standard creation mode tools include just the menu tool itself. The submenu and menu item tools are grayed out.</p> <p>In the custom mode, submenu and menu items are active and must be explicitly set.</p>
Standard Creation Mode Tools	Custom Creation Mode Tools	

	<p>To create a standard menu, you simply draw the position of the first menu header and type in each header, submenu, or item, navigating around the hierarchy with the arrow keys.</p> <p>You can set certain properties of the menu system in the Widget Properties pane. You can set what action invokes the opening of a submenu. In the figure, submenus are opened by hovering the cursor over them. Since the mouse might have to hover for a while before the submenu opens, you can select a submenu Transition delay (typically 0.0 seconds for a Mac and 0.5 seconds for a PC).</p> <p>In addition to this behavioral support, all the menu items line up automatically, and you can insert horizontal separators simply by typing “---” as you type in your menu items.</p>
--	---

## Context Menu Widget

Standard context menus have similar benefits as standard menus. CogTool provides substantial behavioral and layout support that you must provide if you use custom context menus. Context menus are created in a way similar to menus, and they have similar behavior and parameters.

## Pull-Down List Widget

Standard pull-down lists have similar benefits as standard menus. CogTool provides substantial behavioral and layout support that you must provide if you use custom pull-down lists.

## List Box Item Widget

List box item Widget layout and behavior are the same in either mode.

## Non-Interactive Widget

Non-interactive Widget layout and behavior are the same in either mode.

## APPENDIX D: TRANSITION PROPERTIES

### Mouse

This action type represents acting on a particular on-screen location using a mouse cursor. Three properties can be configured (see Figure A-1):

- **Mouse Button:** represents which button is used on a multi-button mouse (Left, Right, Middle).
- **Action:** represents the physical motion taken by the mouse.
  - \* **Click** actions represent button actions on a mouse.
  - \* **Hover** represents when the mouse cursor is positioned at a location but not clicked.
- **Modifiers:** represents which keyboard modifier keys are pressed when a click mouse action is performed.

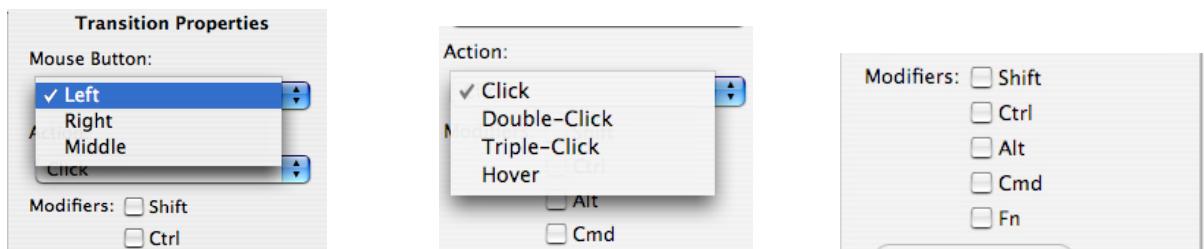


Figure A-1: Mouse Transition Properties: (from left to right) Button properties, action properties, and modifiers

To model the drag-and-drop action performed with a mouse, 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, and 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.



Hover only has meaning in touchscreen hardware that can detect hovering above the surface of the screen without touching the screen. Some versions of SMART Board™ technology can detect hovering; PDAs typically cannot.



CogTool currently supports the **Click** motion and separate down and up motions, but it does not check for down and up motions in pairs. For example, when you click down on the mouse, the next action should be an up, not another down. Although CogTool does not crash, the same repeated mouse action does not make sense when demonstrated. 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.

## Keyboard

This action type represents text input from the keyboard. In the Transition properties, **Text** represents the actual characters typed. **Is Command** distinguishes whether the key presses should be recognized as a command, which triggers an active behavior in the modeled system, or as simple text input. It is not possible to input certain keyboard characters directly. To insert key presses such as Shift and Escape, position the text cursor where the key press is to occur and press the corresponding “Modifiers” or “Specials” button. You can also set the text and command sequence as default behaviors for the device (see Figure A-2).



The **Is Command** box should be checked when the text entry issues an explicit command to the computer.

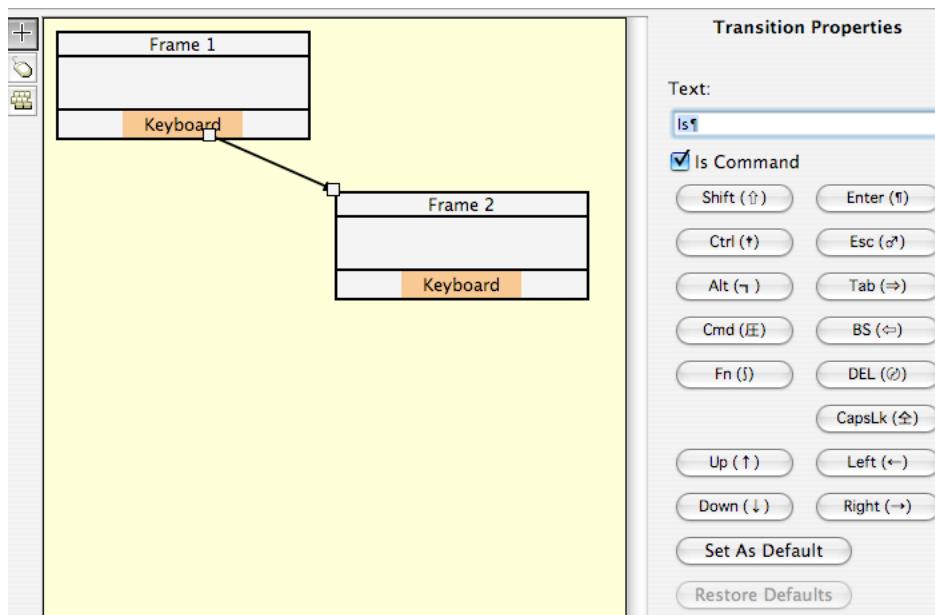


Figure A-2: Keyboard Transition Properties: This figure shows how to simulate entering the “ls” command to a UNIX terminal.

## Touchscreen

A touchscreen device 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** represents the physical motion. **Tap** actions correspond to physical finger taps on a touch-sensitive screen. **Hover** is the act of moving a cursor to a location without tapping at that location. **Hover** is only useful for touchscreen hardware that can detect movements above the surface (see Figure A-3).

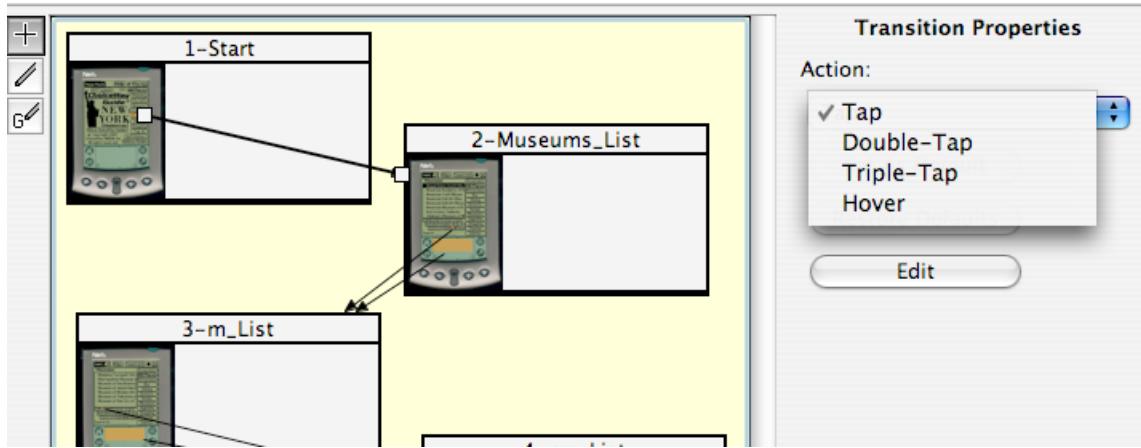


Figure A-3: Touchscreen Transition Properties: Hover only has meaning in touchscreen hardware that can detect hovering above the surface of the screen without touching the screen. Some versions of SMART Board™ technology can detect hovering; PDAs typically cannot.

## Graffiti®

For Graffiti® touchscreens, Graffiti® Gestures represent the characters' input via a stylus 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 as simple text input. Both types are entered into the text box. Checking the checkbox to the left of “Is Command” indicates a command (see Figure A-4).

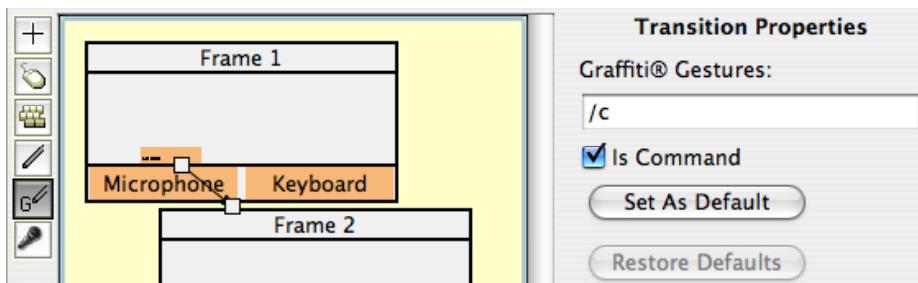


Figure A-4: Graffiti® Gestures Transition Properties

## Microphone

Just as Transitions from the keyboard represent characters typed by a user, Transitions from the microphone represent voice input from a user. When you draw a Transition from the microphone, a dialog box appears and asks for the words the user will speak. You can also enter these words by selecting the Transition and entering them in to the Transitions Properties pane (see Figure A-5).



Entering numbers or symbols as voice input may cause CogTool to crash or fail to compute. To avoid these problems, only enter words.

The “Is Command” property functions the same way for microphone Transitions as it does for keyboard Transitions; it distinguishes whether spoken words should be recognized as a command, which triggers an active behavior in the modeled system, or as simple voice input.

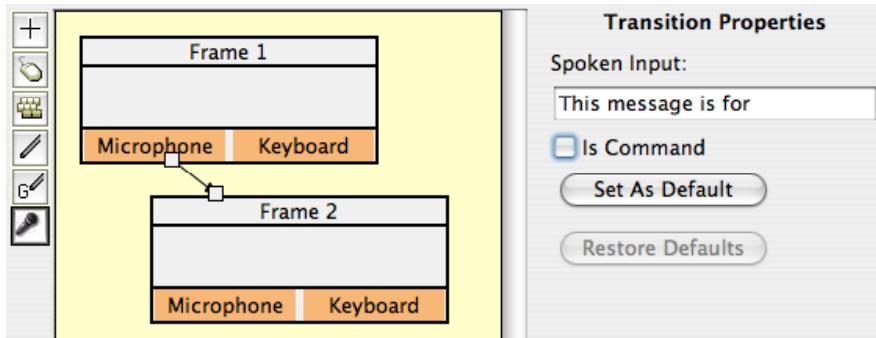


Figure A-5: Microphone Transitions are used to enter spoken words.

## Speaker

Just as Transitions from the microphone represent voice spoken by a user, Transitions from the speaker represent voice spoken by the system. A speaker is an output device.



Microphone Transitions are input from the user. Speaker Transitions are output from the system.

Speaker text is placed in the Speaker box at the top of the Frame and is generated from an initiating Frame in a Transition. As an example, a voice or sound alarm will cause a user to perform an action in response to the alarm.

You can specify the amount of time an individual will listen to the audible message by placing a number that represents the time in seconds in the small box to the right of the speaker text box (see Figure A-6).

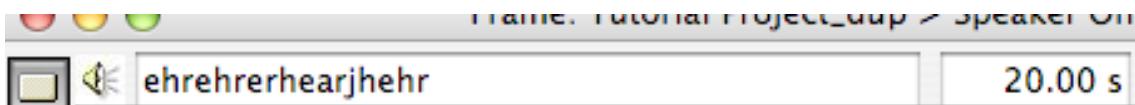


Figure A-6: The longer box to the left is the speaker text box. The smaller box on the right is the listening time.

In the absence of listening time, CogTool uses the speaker text to compute the amount of time the speaker is active, providing a prediction by counting 50 milliseconds/character. If the listening time is set, it overrides the speaker text (see Figure A-7).



Figure A-7: Text typed in the speaker box is audible at the Transition.

Although the prediction time displayed in the Script cell accounts for the speaker interaction, nothing appears in the Script Step List. Speaker actions can be viewed in the model visualization, appearing as a “Listen in the Cognition” row.

## APPENDIX E: SEMANTIC KEYS AND SYMBOLS

A **semantic key** is a key that performs a special function (as opposed to standard letter and number keys). While CogTool uses symbols to represent these keys, other applications (e.g. Microsoft Excel) may not be able to display the symbols and must therefore use substitutes. Listed below are all fifteen semantic keys, the symbols used to represent them in both the Windows and Macintosh versions of CogTool, and the symbols' corresponding substitutes when they cannot be displayed.

Semantic Key	Win Symbol	Mac Symbol	Substitute for Symbol
Shift	↑	↑	(SHIFT)
Control	†	†	(CTRL)
Alt	⊸	⊸	(ALT)
Command	田	压	(CMD)
Function	ƒ	ƒ	(FN)
Enter	CR	¶	(ENTER)
Escape	ESC	♂	(ESC)
Tab	→	⇒	(TAB)
Backspace	BS	←	(BS)
Delete	DEL	⌚	(DEL)
Caps Lock	⇧	仝	(CAPS)
Up	↑	↑	(UP)
Down	↓	↓	(DOWN)
Left	←	←	(LEFT)
Right	→	→	(RIGHT)

## APPENDIX F: REPORTING A BUG

Though fully functional, CogTool is still a work in progress and will sometimes crash. A CogTool crash generates an error window like the one seen below.

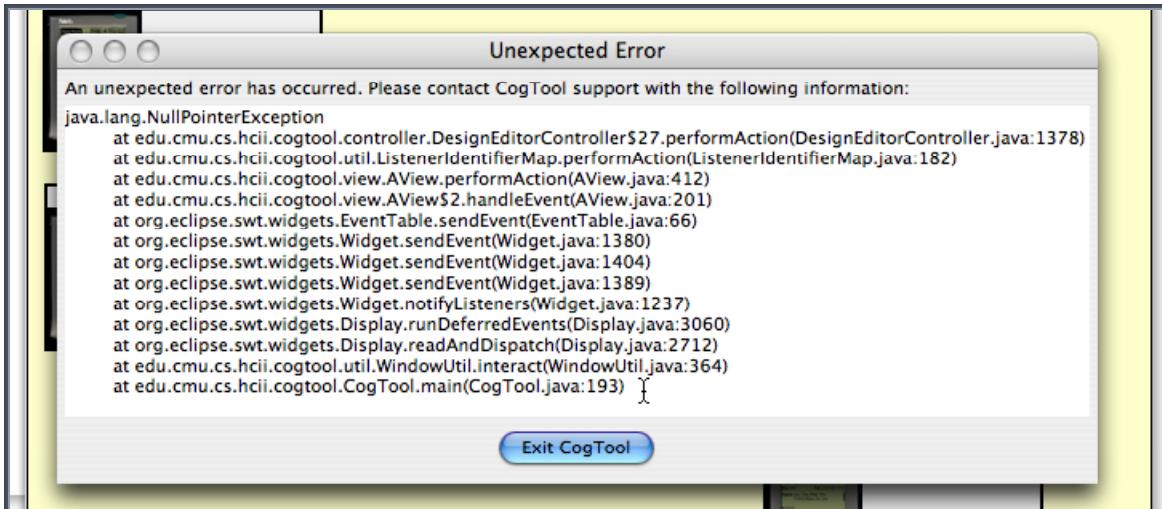


Figure A-8: Error message window

CogTool crashes result from bugs in the program that have not yet been discovered or fixed. Reporting these bugs alerts us to problems we can address for future releases of CogTool.

To report a bug:

1. Copy the text in the error window by highlighting the text and copying with CTRL+C (Windows) or ⌘C (Mac OS).
2. Identify the version of CogTool you have.

**Mac OS X:** Under the CogTool menu listing, choose About CogTool.

**Windows:** Under the Help menu listing, choose About CogTool.



3. Send an email to cogtool@cs.cmu.edu with the text from the error message, the version of CogTool you are using, and a description of the action you performed immediately before the error appeared. If we need additional information, we will contact you.

**Note:** If you have a data file and are willing to share it, please send the file as an attachment.

We will do our best to resolve the problem and will contact you with a fix or workaround as soon as one is available.

## APPENDIX G: COGTOOL SHORTCUT KEYS

The following shortcut keys can be used whenever the action they perform is required.

Save	Save As	Open Project	New Project	Close Window
Mac: ⌘S	Mac: ⌘+S	Mac: ⌘O	Mac: ⌘N	Mac: ⌘W
Win: CTRL+S	Win: CTRL+SHIFT+S	Win: CTRL+O	Win: CTRL+N	Win: CTRL+W

Save	Copy	Cut	Paste	Duplicate
Mac: ⌘S	Mac: ⌘C	Mac: ⌘X	Mac: ⌘V	Mac: ⌘D
Win: CTRL+S	Win: CTRL+C	Win: CTRL+X	Win: CTRL+V	Win: CTRL+D

Delete	Rename	Edit	Undo	Redo
Mac: ⌘⌫	Mac: ⌘R	Mac: ⌘E	Mac: ⌘Z	Mac: ⌘Y
Win: CTRL+DELETE	Win: CTRL+R	Win: CTRL+E	Win: CTRL+Z	Win: CTRL+Y

The remaining shortcut keys perform actions that are specific to the CogTool window you are using.

Design
New Design
Mac: ⌘+D
Win: CTRL+SHIFT+D

Frame		
New Frame	Setting Background Image	Removing Background Image
Mac: ⌘+F	Mac: ⌘B	Mac: ⌘+B
Win: CTRL+SHIFT+F	Win: CTRL+B	Win: CTRL+SHIFT+B

Widget	
New Widget	Setting Layer Color
Mac: ⌘+I	Mac: ⌘+C
Win: CTRL+SHIFT+I	Win: CTRL+SHIFT+C

Task/Task Group	
New Task	New Task Group
Mac: ⌘+T	Mac: ⌘+G
Win: CTRL+SHIFT+T	Win: CTRL+SHIFT+G

Zoom for Design, Frame, Script			
Zoom In	Zoom Out	Normal Zoom	Zoom to Fit
Mac: ⌘+	Mac: ⌘-	Mac: ⌘0	Mac: ⌘/
Win: CTRL+=	Win: CTRL+ -	Win: CTRL+0	Win: CTRL+/

## **APPENDIX H: COGTOOL CONTEXT MENUS**

## **APPENDIX I: ADDITIONAL INFORMATION**

### **References**

Landay, J. A., & Meyers, B. A. (1995). Interactive Sketching for the Early Stages of User Interface Design. Proceedings of CHI 1995, 43-50.

Lin, J., Newman, M. W., Hong, J. I., & Landay, J. A. (2000) DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. CHI Letters: Human Factors in Computing Systems, CHI 2000, 510-517.

Luo, L., & John, B. (2005) Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model. Proceedings of CHI 2005, 1605-1608.

### **Original CogTool (the paper to reference when talking about CogTool)**

John, B. E., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. Proceedings of CHI 2004 (Vienna, Austria, April 2004) ACM New York. 455-462.

### **CogTool-Explorer—predicting exploratory behavior of novice users**

Teo, L., & John, B. E. (2008) Towards Predicting User Interaction with CogTool-Explorer. Proceedings of the Human Factors and Ergonomics Society 52nd Annual Meeting (New York City, New York, Sept 22-26, 2008).

Teo, L., & John, B. E. (2008) CogTool-Explorer: Towards a Tool for Predicting User Interaction. Extended Abstracts of CHI 2008 (Florence, Italy, April 5 – 10, 2008) ACM, New York.

Teo, L., John, B. E., & Pirolli, P. (2007) Towards a tool for predicting user exploration. Extended Abstracts of CHI 2007 (San Jose, USA, April 28 – May 3, 2007) ACM, New York.

### **CogTool for PDA**

Luo, L., & John, B. E. (2005) Predicting task execution time on handheld devices using the Keystroke-Level Model. Extended Abstracts of CHI 2005 (Portland, Oregon, April 2-7, 2005) ACM, New York.

### **Use of CogTool's export to HTML and on off-the-desktop devices**

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.

### **Predicting power consumption for mobile device**

Luo, L., & Siewiorek, (2007) D.P. KLEM: A Method for Predicting User Interaction Time and System Energy Consumption during Application Design. 11th International Symposium on Wearable Computers/, Boston, MA, October 2007.

## Learn More

### ACT-R:

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

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004) An integrated theory of the mind. *Psychological Review*, 111 (4), 1036-1060.

### KLM:

Card, S. K., Moran, T. P., & Newell, A. (1980) The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM*, 23(7), 396-410.

## **APPENDIX J: GLOSSARY**

### **ACT-R**

ACT-R is a cognitive theory that explains how human cognition works. It provides a framework for how humans will interact when they perform complex tasks. You can collect quantitative analyses from ACT-R models and compare them to data collected from human participants to measure performance time and accuracy. For more information on ACT-R, go to : <http://act-r.psy.cmu.edu/about/>.

### **Demonstration**

A Demonstration follows a series of Transitions from the initiating Frame to the ending Frame. CogTool computes a predicted execution time for the Demonstration based on ACT-R and lists the actions and cognitive steps that the user performs to move from the initiating Frame to the ending Frame. A Demonstration is performed on a Task.

### **Design**

A Design is a series of Frames that represent the user interface of the system being analyzed. The Frames are linked by Transitions.

### **Frame**

A Frame represents a single screen of a user interface. Multiple Frames can be linked together to form a complete Design.

### **Keystroke-Level Model (KLM)**

Keystroke-level model (KLM) is one of the frameworks used by CogTool to make predictions on how long it would take to perform a Task using a keyboard and mouse.

### **Project**

A Project is central to CogTool. All the parts necessary for CogTool to make performance predictions are kept in a Project file. The Project contains the user interface Designs, the Tasks being studied, and the Scripts that define the human performance being predicted.

### **Script**

A Script demonstrates the specific sequence of user actions that are needed to predict the execution time of a Task.

## Task

A Task represents a specific sequence of actions a user will take to complete an operation.

## Task Group

A Task Group organizes the Tasks and computes statistics on the sum, mean, minimum, and maximum prediction times for the Tasks in that group.

## Transition

A Transition is the link between Frames that form a Task. When the interface is demonstrated, interaction with a Widget will follow the Widget's Transition to the next Frame.

## Widgets

A Widget is an element on a Frame with which a user may interact. A Widget could be a button, link, checkbox, radio button, text box, text, menu, submenu, menu item, context menu, pull-down list, pull-down item, list box item, or Graffiti®. In CogTool, Widgets can also be created for non-interactive elements to construct a full representation of an interface.