

# **User Guide**

by

**Nick Pudar**

Card brand names and back designs used in StackView are registered trademarks of The United States Playing Card Company in the United States and other countries.

Copyright © 2006 by Nick Pudar

# Table of Contents

<b>TABLE OF CONTENTS .....</b>	<b>3</b>
<b>INTRODUCTION .....</b>	<b>11</b>
WHAT IS STACKVIEW? .....	11
CONCEPTUAL STRUCTURE .....	12
<i>Events</i> .....	12
<i>Stack Values and Position Values</i> .....	13
USER INTERFACE .....	13
BACKGROUND AND CREDITS .....	16
<i>Credits</i> .....	18
CONVENTIONS USED IN THIS USERS GUIDE .....	18
<b>WHAT'S NEW IN VERSION 5? .....</b>	<b>20</b>
<b>REFERENCE: DECK.....</b>	<b>24</b>
DECK VIEWS .....	24
<i>Default Full Deck View</i> .....	24
<i>Piles View</i> .....	24
<i>Poker Deal View</i> .....	24
<i>Point of View</i> .....	24
STACK VALUES .....	29
POSITION VALUES .....	30
SELECTED CARDS .....	30
REVERSING CARDS IN THE DECK WINDOW .....	31
<b>REFERENCE: STACKVIEW CONTROL .....</b>	<b>32</b>
CONTROL TAB .....	32
<i>Arrange Cards</i> .....	32
<i>Conventional Shuffles</i> .....	37
<i>Faro Shuffles</i> .....	53
SELECTION TAB.....	63
<i>Deck Visual Indicators</i> .....	64
<i>Force Selection</i> .....	68
<i>Free Choice From Spread</i> .....	69
<i>Free Choice with Special Handling</i> .....	71

<i>Swap Cards Selection</i> .....	77
SESSIONS TAB .....	80
<i>Recording Sessions</i> .....	81
<i>Working with Session Files</i> .....	84
<i>Controlling Sessions</i> .....	84
<b>REFERENCE: PILES CONTROL .....</b>	<b>88</b>
CREATE PILES .....	88
<i>Cut Cards</i> .....	89
<i>Deal Cards</i> .....	93
PILES VIEWPOINT .....	98
PILE CONTROL MATRIX.....	102
SELECT / RETURN .....	105
<i>Selection Pile</i> .....	105
<i>Selected Card</i> .....	106
<i>Return Pile</i> .....	106
<i>Return Position</i> .....	108
<i>Reverse Selected Card</i> .....	108
<i>Move Only</i> .....	109
COMBINE.....	109
<i>Secondary on Top</i> .....	109
<i>Primary on Top</i> .....	109
RIFFLE SHUFFLE / GILBREATH PRINCIPLE .....	109
<i>Random</i> .....	110
<i>Protect Block</i> .....	110
<i>Gilbreath View</i> .....	111
CUT PILES .....	112
<i>Cut Pile and Portion</i> .....	113
<i>Place Cut Portion</i> .....	113
<i>Reverse Cut Portion</i> .....	114
AUSTRALIAN DEAL .....	115
<i>Select</i> .....	116
<i>Start</i> .....	117
<i>Number</i> .....	117
<i>Reverse</i> .....	118
<i>Inverse</i> .....	119
SWAP PILES .....	122

<i>First Pile</i> .....	122
<i>Second Pile</i> .....	123
<i>Reverse First Pile</i> .....	128
<i>Reverse Second Pile</i> .....	128
SPECIAL.....	128
<i>Elmsley Count</i> .....	129
<i>Jordan Count</i> .....	130
<i>Turn Over Pile</i> .....	131
<i>Inverse</i> .....	131
REFRESH DECK.....	132
<b>REFERENCE: CUSTOM DECK .....</b>	<b>133</b>
USER INTERFACE.....	133
RETAINED VS. NEW STACK VALUES.....	135
CONTROL BUTTONS.....	136
<i>Import Current Deck from StackView</i> .....	136
<i>Transfer: New</i> .....	137
<i>Transfer: Retain</i> .....	137
<i>Reset: New</i> .....	137
<i>Reset: Retain</i> .....	138
<i>Reorder Retained Stack</i> .....	138
CREATE A STANYON VARIATION DECK.....	138
<i>Stanyon Variation Results Output</i> .....	139
IMPORTING A DECK .....	142
TRANSFERRING A DECK .....	143
SAVING YOUR CREATED STACK.....	143
<b>REFERENCE: CUSTOM MNEMONICS .....</b>	<b>144</b>
DEFAULT MNEMONIC FILE .....	144
CREATING CUSTOM MNEMONICS .....	144
SAVING A MNEMONIC FILE .....	145
<b>REFERENCE: BACK DESIGN .....</b>	<b>147</b>
<b>REFERENCE: JOYAL SHUFFLEMETER.....</b>	<b>148</b>
ANALYSIS TABS .....	148
<i>Spread</i> .....	149

<i>Permutation</i> .....	150
<i>Distribution</i> .....	151
<i>Group</i> .....	152
<i>Break</i> .....	153
SHUFFLE INDEX.....	154
<b>REFERENCE: STACKVIEW TEST .....</b>	<b>155</b>
ACCESSING STACKVIEW TEST .....	156
DECK RANGE.....	156
<i>Full Deck</i> .....	156
<i>Stack Value Range</i> .....	156
<i>Current Deck Position Range</i> .....	156
STACK SEQUENCE .....	157
<i>Forward</i> .....	158
<i>Backward</i> .....	158
<i>Current Order</i> .....	158
<i>Random</i> .....	159
TEST VALUE.....	159
<i>Card Value</i> .....	159
<i>Stack Value</i> .....	160
<i>Random Mix</i> .....	162
TEST CARD.....	162
<i>Current Card</i> .....	162
<i>Next Card</i> .....	163
<i>Previous Card</i> .....	163
<i>Random Card</i> .....	165
<i>Additional Note:</i> .....	165
TIMERS .....	165
<i>Durations</i> .....	165
<i>Enable Timers</i> .....	167
MNEMONICS .....	167
<i>Enable Mnemonics</i> .....	167
<i>Auto Hint</i> .....	169
CONTROL BUTTONS.....	170
<i>Test</i> .....	171
<i>Next</i> .....	173
<i>Show</i> .....	174

<i>Hint</i> .....	175
<i>Cards Remaining Progress Bar</i> .....	176
TEST RESULTS .....	176
<b>REFERENCE: STACKVIEW ADVANCED TEST .....</b>	<b>179</b>
ACCESSING STACKVIEW ADVANCED TEST .....	179
DECK ORDER .....	179
<i>Starting Deck Order</i> .....	179
<i>Current Deck Order</i> .....	180
KNOWN CARD .....	180
<i>Top Card</i> .....	180
<i>Bottom Card</i> .....	180
DESIRED CARD .....	180
<i>Random Card</i> .....	181
<i>Specified Stack Value</i> .....	181
DESIRED POSITION .....	181
<i>Random Position</i> .....	181
<i>Specified Position Value</i> .....	181
TEST QUESTIONS .....	181
<i>Cards to Cut</i> .....	182
<i>New Top Card Stack Value</i> .....	182
<i>New Bottom Card Stack Value</i> .....	182
ANSWER INPUT .....	182
<i>Answer Entry</i> .....	182
<i>Answer Response</i> .....	183
TIMERS .....	185
<i>Durations</i> .....	186
<i>Enable Timers</i> .....	187
CONTROL BUTTONS .....	188
<i>Test</i> .....	189
<i>Next</i> .....	190
<i>Show</i> .....	191
<i>Cards Remaining Progress Bar</i> .....	192
TEST RESULTS .....	193
<i>Time</i> .....	194
<i>Percent Correct</i> .....	194
<i>Score</i> .....	194

<b>REFERENCE: STACKVIEW SEARCH.....</b>	<b>196</b>
DECK SETTINGS.....	197
<i>Start Deck</i> .....	197
<i>Target Deck</i> .....	197
<i>Deck Match Setting</i> .....	197
<i>Threshold Trap</i> .....	197
MANIPULATIONS AND MOVES .....	199
<i>Maximum Moves</i> .....	199
<i>Identify Manipulations to Include in Search</i> .....	199
CONTROL BUTTONS.....	200
<i>Starting the Search</i> .....	200
<i>Interrupting a Search</i> .....	200
SEARCH PROGRESS INDICATION .....	201
<i>Estimated Possible Search Time</i> .....	201
<i>Interim Results</i> .....	201
TRANSFER LIST TO SESSIONS .....	202
<b>REFERENCE: MENUS.....</b>	<b>203</b>
FILE .....	203
<i>Open Deck</i> .....	203
<i>Save Deck As</i> ... ..	203
<i>Save Deck As Default</i> .....	204
<i>Open Session</i> ... ..	204
<i>Save Session As</i> ... ..	204
<i>Open Search</i> ... ..	205
<i>Save Search As</i> ... ..	205
<i>Open Mnemonic</i> ... ..	206
<i>Save Mnemonic As</i> .....	206
<i>Save Mnemonic As Default</i> .....	206
<i>Print</i> .....	206
<i>Exit</i> .....	207
VIEW .....	207
<i>Deck</i> .....	207
<i>Controls</i> .....	207
<i>Deck File</i> .....	208
<i>Session File</i> .....	208



<i>Mnemonic File</i> .....	208
<i>Search File</i> .....	208
<i>Threshold Trap File</i> .....	208
<b>TOOLS</b> .....	<b>209</b>
<i>Custom Deck</i> .....	209
<i>Custom Mnemonics</i> .....	209
<i>Joyal ShuffleMeter</i> .....	209
<i>Piles Control</i> .....	209
<i>Set Back Design</i> .....	210
<i>StackView Test</i> .....	210
<i>StackView Advanced Test</i> .....	210
<i>StackView Search</i> .....	210
<b>WINDOW</b> .....	<b>210</b>
<i>Cascade</i> .....	210
<i>Open Window Selection</i> .....	211
<b>HELP</b> .....	<b>211</b>
<i>Open User Guide</i> .....	211
<i>About</i> .....	211
<b>APPENDIX 1: SAMPLE EXERCISES</b> .....	<b>212</b>
SPECIFYING THE ORDER OF A STACK.....	212
ANY CARD TO ANY NUMBER WITH FAROS.....	217
WORKING WITH SESSIONS .....	221
<b>APPENDIX 2: FREQUENTLY ASKED QUESTIONS</b> .....	<b>229</b>
WHO DO I CONTACT IF I HAVE SPECIFIC QUESTIONS?.....	229
THE PROGRAM CRASHED. WHAT NOW? .....	229
I THINK I FOUND A BUG IN THE SOFTWARE. THE RESULTS DON'T SEEM RIGHT. WHAT DO I DO?.....	229
HOW DO I LEARN MORE ABOUT STACKED DECK MAGIC? .....	230
IN THE CUSTOM DECK WINDOW, WHY DON'T THE CARDS DRAG-AND-DROP CORRECTLY SOMETIMES? ..	230
WHY IS EVERYTHING SO BIG ON MY SCREEN?.....	230
WHEN WILL AN APPLE MAC OR A POCKET PC VERSION BE WRITTEN? .....	230
HOW DO I CREATE MY OWN STACKS? .....	231
HOW DO I MAKE A SPECIFIC STACK BE THE DEFAULT STACK THAT STACKVIEW USES WHEN IT LOADS?.	231
ONE (OR MORE) OF THE WINDOWS IS MISSING FROM THE SCREEN. WHERE ARE THEY?.....	232
WHAT ARE ALL THE SPECIAL FILE FORMATS THAT STACKVIEW USES?.....	233
WHAT IS A THRESHOLD TRAP?.....	234

HOW DO I RESET/RE-ASSIGN THE STACK VALUES WITHOUT DISTURBING THE CURRENT DECK ORDER?...	234
IS STACKVIEW SEARCH USEFUL? .....	235
ARE THE STACKVIEW TEST TIMERS ACCURATE?.....	235
WHY ARE THERE “INVERSE” PARAMETERS ON MANY OF THE EVENTS? .....	236
<i>Example 1</i> .....	236
<i>Example 2</i> .....	236
<i>Example 3</i> .....	237
ARE THERE VERSION INCOMPATIBILITIES? .....	237
<b>APPENDIX 3: SESSION COMMANDS .....</b>	<b>239</b>
<b>APPENDIX 4: STACKVIEW FILE STRUCTURES .....</b>	<b>269</b>
DECK FILE .....	269
SESSION FILE.....	270
MNEMONIC FILE.....	270
SEARCH FILE .....	271
THRESHOLD TRAP FILE .....	271
<b>INDEX .....</b>	<b>272</b>

## Introduction

### *What is StackView?*

StackView is a computer based tool for magicians to explore various aspects of controlling a stacked deck. There are many procedures and manipulations that magicians can do with a deck of cards. It is easy and desirable to practice and explore these techniques with cards in hand. However, the exploration of the effects these techniques have on a stacked deck becomes troublesome; it is cumbersome and time consuming to re-stack the deck each time you want to try a procedure. As a result, the percentage of innovative stacked deck material in the literature is miniscule when compared to the main body of non-stacked deck work. It is actually quite remarkable that many of the great stacked deck procedures had been developed with "deck in hand." It is a tribute to the ingenuity (and patience) of some of the great thinkers in the magic community.

The purpose of Stackview is to explore "stuff" with cards in a simulated manner. As the knowledgeable card worker will realize, the software is focused on a specialized form of stacked deck. You will note throughout this manual that many aspects of card magic are referenced by name. StackView includes several published and well known stacks for you use. However, you will not find any explanation of what any of it means in this manual; it is assumed that you already know what it all means. If there is anything unfamiliar, I urge you to learn about it from the literature or in your magic community.

Throughout this users guide I will use the Aronson Stack in my examples. Since this is the stack that I now use, it makes it quicker for me to write up the examples without having to look up the correct *Stack Values*. StackView includes several published and well-known stacks that can be selected for use in the program. In addition, you can also create your own custom stacks with an interactive deck builder.

## ***Conceptual Structure***

### **Events**

The procedures that you can do to cards are called *Events* in StackView. These *Events* are performed on the cards at three levels: 1) the deck, 2) a pile, and 3) a card.

There are *Events* that logically apply to the entire deck. An example would be a standard *Faro* shuffle. StackView provides a large range of *Events* that can be performed with the entire deck. These *Events* are found in the main *StackView Control* window among its several tabs.

Some events are performed on a subset of cards. In StackView, the subsets are called *Piles*. The *Pile Events* are found in a special *Piles Control* window, which can be accessed from the *Tools* menu. Some of the *Piles Events* are similar to those that can be done with an entire deck. However, there is more control of the features of these *Events* from within the *Piles Control* window. In addition, there are some unique *Events* that only apply to *Piles* such as the *Elmsley Count* or the *Australian Deal*. (All *Pile Events* can be applied to the entire deck by first creating a single pile of 52 cards.)

There are a few *Events* that can be done to a single card. These *Events* are limited and are applied to the *Deck* window directly. For example, you can reverse a card in the deck by double-clicking on the card with your mouse cursor. You can also have a card selected and returned to the deck, where its selected status is retained. Another example is when you double-click on a card in a visible poker hand while you are in a *Poker Deal* view. In this case, double-clicking is equivalent to performing a "discard."

Several *Events* have an "inverse" setting. This setting performs the *Event* backwards. For example, if you do a *Faro* shuffle followed by an *Inverse Faro* shuffle, the deck will be back to its starting order. Inverse *Events* are most

helpful when you are trying to construct a routine, and need to know what the starting order of the cards should be before you do an *Event* to them.

Once you have a sequence of *Events* that you like, you can record them as *Session* files. A *Session* file is equivalent to a macro that will re-play a sequence of *Events*. You can edit the *Session* sequence and play them back all at once, or one step at a time.

### **Stack Values and Position Values**

Throughout the StackView users guide and software, the phrases "Stack Value" and "Position Value" are used. It is important to understand the distinction between these terms. The term *Stack Value* refers to the secret value that is assigned to a specific card in your stack. For example, in the Aronson Stack, the Three of Spades has the *Stack Value* of 17, regardless of where the card may be in the deck. The *Position Value* refers to a cards position from the top of the deck. If the Three of Spades happens to be the second card from the top of the deck, it will have a *Position Value* of 2.

### ***User Interface***

In order to best view the various features of StackView, set your computer monitor resolution to its highest setting. (A resolution of 1024 x 768 works well.) If you use a lower resolution or a small screen, you will have to scroll around the screen at times to see all of the display windows. The individual windows are not resizable.

The user interface design was selected for ease of use. StackView allows you to perform various *Events* to the cards. Each *Event* has varying degrees of parameter control. Once you have set your *Event* parameters, you execute it by pressing its command button. The effect of the *Event* can be immediately seen in the *Deck* window.

All activities in StackView occur sequentially. While not everything that you can do with cards is a predefined *Event*, you can create your own events which are built up of sequence of existing *Events*. These sequences of *Events* can be saved in special macro files called *Sessions*.

There are several specialized task windows that are used in StackView. The key one is the *Deck* window. This window shows the full deck of cards in its current state with both the stack values and position values showing (if not disabled). The cards will be shown in whatever context is required by the *Events* you process (such as poker deals, Gilbreath view, piles control, etc.).

The *StackView Control* window is the main deck interaction area. Here, found on one of three tabs, you can control which stack you are using, have cards selected and returned, shuffled, cut, controlled, as well as establishing the back design and color. In addition, you can record your sessions for later playback.

When the *Sessions* mode is turned on, all recordable *Event* buttons change color and have red button faces.

The *Piles Control* window allows you to create subsets of cards, called *Piles*. The *Piles* related *Events* allow you to perform functions to a single *Pile*, or to a combination of *Piles*.

The *Set Back Design* window allows you to select which back design and color your cards will display when cards are reversed in the deck.

The *Custom Mnemonic* window provides you with special tables of mnemonics for the stack values and position values that are editable. You may save your own mnemonic file as the default for StackView to use.

The *Joyal ShuffleMeter* module is an analysis tool as presented in Martin Joyal's *The Six-Hour Memorized Deck* (1997 Hermetic Press).

The Custom Deck window provides you with an interactive tool to create your own stacks. You can choose to create your stack by specifying every card in the deck, or you can create a *Stanyon Variation* stack by specifying the first five cards of the stack. You can save your created stacks and even set one to be the default stack to be used by StackView.

There is a full featured *Search* tool which allows you to explore the card move sequence that is required to get from a starting card sequence to another target card sequence. The card sequences can be the whole deck or just a portion of the deck. In addition, you can set the search process "Threshold Trap" parameter to respond to a "near match." A near match is defined by the number of cards that do match. For example, you may have the target set as the entire deck, but are interested in knowing when an interim search result actually hits 45 cards or more. This ability to set *Threshold Traps* augments the brute force of computer searches with your own creativity. You can also specify if you want the *Search* process to stop or just save the interim results to a file for later use when a *Threshold Trap* is triggered.

There are also two special test tools for testing your knowledge of your stacked deck. Each test tool can be run with or without a timer running. When the timer is activated, you will get a final score to monitor your progress. The main *StackView Test* module is the primary quizzer that allows complete control of how the cards are presented, and which cards will be tested. You also have the option to view mnemonic hints during the test. The *StackView Advanced Test* module provides you with complete control for testing your ability to control desired cards to desired locations.

You can view various saved StackView user generated files as text documents with the View menu command.

As much as possible, StackView is error-proofed. When certain logical combinations are not consistent, certain choices are disabled. In other cases, pop-up dialog boxes will appear to explain certain constraints. Also, when parametric errors are input, specific error reports are provided in pop-up dialog boxes. As such, all error conditions are trapped and dealt with. In the event that you experience any crashes or incorrect event handling, please send me an email with as much detail of the experience as you can to:  
nick@stackview.com.

### ***Background and Credits***

Even though I had been a reasonably serious student of magic (and in particular, card magic) for over thirty years, it wasn't until 2002 that I "discovered" stacked deck magic. It is probably more correct to say that I subconsciously chose to ignore stacked deck magic, since my personal magic library had a healthy dose of stacked deck material.

During 2002, I learned a couple of stacks, and sought out as much material as I could get my hands on (books and internet forums). Along this new learning path, I came up with a new stacked deck idea to find multiple selected cards under seemingly impossible conditions. The basic premise for the handling was inspired by Steve Beam's routine called "Multiple Impact." My method allowed for genuine shuffling by the spectators before and after the cards were selected. While my methodology seemed to work flawlessly, I could not convince myself that it would work every time. I had an intuition that there might be some rare situation of spectator handling where the method would not work.

Since my method required a stacked deck, real spectator choices, and a lot of mental processing, the opportunity to practice the method and test its robustness was challenging. The only way I could figure out how to assure myself that my method worked or didn't work would be to write a computer simulation of the entire handling. It would have to simulate a deck of cards that



starts in stacked order. There would be random shuffling, random card selections, more random shuffling, and finally, the program would have to simulate my method of discerning the cards before comparing them to the actual selected cards.

I used to be computer literate over twenty years ago; my programming skills had seriously atrophied. So, with just a bit of determination, I re-learned how to write programs in real time in order to simulate my card routine idea. The original code was not pretty, but it worked. I was able to run several hundred thousand simulations, and found that indeed, my intuition was correct – there were some failure modes in my method. My handling would correctly identify the selected cards only 97 percent of the time. So I scrapped the idea. (And I was unable to identify any satisfactory adjustments or constraints that would eliminate the failure mode.)

In the end, I had a purpose-built simulation (entirely text-based user interface) of a specific handling with a stacked deck of cards. Around that same time (early 2003), I was rediscovering the stacked deck material in Michael Close's "Workers Number 5." In particular I was amazed that he ever discovered what he did with the effect "Tut Tut." I could just imagine Mike doing Faro shuffles and inspecting what happened to the deck, and being surprised in finding that nice sequence. I wondered if there might be an easier way to explore what might be hidden in stacked decks, as well as a way to explore what happens when you do "things" to the deck. Simon Aronson's "Try the Impossible" also contributed to my curiosity in no small way.

It occurred to me that my little simulation was in interesting starting point, and if I just put in a little more (hah!) effort, I might end up with a useful utility program for magicians. With further blind confidence, I started down that path. Once I got going, I could not stop, and the further I got, the more ideas I got on

how it should look and how it ought to work. The current product you have before you feels right (for now).

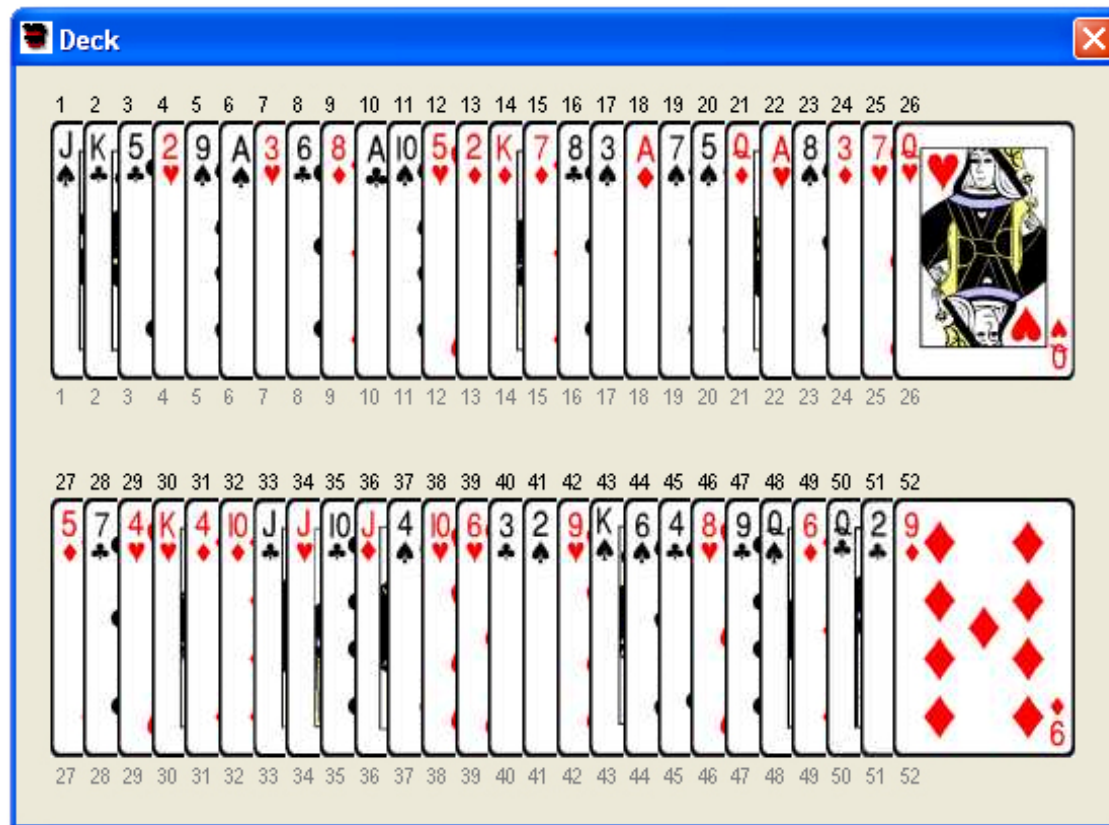
### **Credits**

Special thanks go to Simon Aronson, Martin Joyal, Richard Osterling, and Juan Tamariz for their counsel and kind permission to include their stacks in StackView. Additional thanks go out to Ron Bauer, Eugene Berger, Michael Close, Dick Dale, Dennis Loomis, Stephen Minch, Ramón Riobóo, and Jim Sisti for their guidance at various stages of this project. More special thanks to Martin Joyal for permission to include the Joyal ShuffleMeter from his book *The Six-Hour Memorized Deck*. Further thanks are extended to the US Playing Card Company for permitting the use of their back designs in StackView. Extra special thanks also go to my son Jim, who at the age of 10 coached me on Visual Basic programming and software design.

### ***Conventions Used in this Users Guide***

Throughout this Users Guide, there will be several conventions used with terminology. I want to describe some of them that I am aware of.

When referring to positions of cards in the deck, the *Top* card is the top-most face down card of the deck when held in dealing position. Similarly, the *Bottom* card is the face card of the deck. When the Aronson Stack is shown in the *Deck* window as shown in Figure 1, the Jack of Spades is the *Top* card of the deck, and the Nine of Diamonds is the *Bottom* card of the deck. For more information on the visual aspects of the *Deck* window, see page 24.



**Figure 1**

The examples used throughout this Guide tend to use the Aronson Stack for discussion purposes. I have tried to point that out wherever possible. I have chosen that stack in the examples since it is the stack that I personally use. That way, it was easier to write and edit while minimizing potential errors. As you will see, StackView is stack-agnostic.

## What's New in Version 5?

There have been many new features and adjustments made to StackView since version 4. Below is a summary for those familiar with earlier versions.

- A major new feature is the ability to create separate *Piles* of cards. There is complete control for how the *Piles* are created. Once *Piles* are created, there are many *Events* that you can execute with the *Piles*.
- In the *Deck* window, the position values of the cards are shown just below each card (in a gray color). The stack values are shown just above each card (in black). Any combination of these indicators can be turned on or off. In addition, you can specify that the position values count from the back of the deck or from the face of the deck.
- StackView now allows you to reverse cards in the deck. Just about every *Event* in StackView now has some degree of control to reverse one or more cards in the deck. When a card is reversed, its back design is shown. On the *Tools* menu is a new *Set Back Design* dialog box which provides you many current and historical brands to choose from.
- Double-clicking on the face of any card in the *Deck* window will reverse that card in the deck. If you are recording a *Session*, a `ReverseCard(X)` *Event* for that specific card will be recorded; in this case, the "X" will be the name of the card. At playback of the *Session Event*, that specific card will become reversed, regardless of where it is in the deck. If you double-click on any of the *Position Value* indicators, the card at that position will become reversed. While the effect of double-clicking on the card face or the *Position Value* is the same, the difference is in what is recorded in *Sessions*. If you double-click on the *Position value* while recording a *Session*, whatever card is at that specific position will be reversed at playback. The recorded *Event* will be `ReverseCard(X)`, where "X" will

be a numerical *Position Value*. The ReverseCard *Event* does not work while viewing a *Poker Deal*.

- In previous versions of StackView, the word “reverse” was used to perform an *Event* backwards. For example, if you did a *Faro* shuffle followed by a *Reverse Faro* shuffle, the deck would be back to its starting order. In the current version of StackView, the work “reverse” now is associated with a card that has been turned over (reversed) in the deck. The new word for reversing *Events* is “inverse.” Now, doing an *Inverse Faro* after a regular *Faro* will return the deck back to its starting order.
- In the *Poker Deal*, you can now deal up to 10 hands. While the *Deck* window is displaying the poker hands, you can double-click on any card in play, and StackView will discard that specific card. It will be replaced with the next card from the remaining talon. The discarded card is replaced to the bottom of the remaining talon. If you double click on a card’s *Position Value*, the card at that specific position will be discarded. The discard specification of double-clicking on either the card face or its position allows for more precise control when recording StackView *Sessions*. The discard feature will help the curious explore and develop rich poker routines.
- The Ireland Stack has been added to the standard list of stacks in the drop-down listbox.
- In the *StackView Test* module, you can use *Mnemonic* hints to help you learn your stack. The Aronson mnemonic pairings are the default set. However, you can customize your own set and save it for use by StackView.

- There is a new *Advanced Test* module that will be of significant value in gaining full control of your stack. (My strong advice is that you do not use the timer when you are first using this tool.)
- Martin Joyal has provided kind permission for me to include his *ShuffleMeter* deck analysis tool. For more information on the *ShuffleMeter*, see his book *The Six-Hour Memorized Deck*, Appendix B.
- In the *StackView Control* window, on the *Control* tab, there is a new button called "Refresh Deck View." This button will force the *Deck* window to be displayed in the standard view if you currently have *Poker Hands* or *Piles* showing. If *Poker Hands* are showing, it is equivalent to assembling the poker hands with the "Assemble Poker Deal Backwards" *Event*. If *Piles* are showing, the *Refresh Deck View* button will re-stack the piles in order, so that *Pile 1* is at the top of the deck, and the last *Pile* is at the bottom.
- *StackView* has traditionally shown the deck faces without regard to what might actually be done with the cards. *StackView* now handles this situation by assuming that the cards are spread face down on the table. The standard view now assumes that you are looking at the card faces from the bottom and through the table surface. A new feature now allows you to toggle your view of the cards to look at the cards also from the top of the table. Since cards can now be reversed in the deck, and there is more flexibility with *Piles*, it became important to be able to look at the deck from both sides. The directional view feature is available in the *StackView Control* and *Piles Control* windows, but not for *Poker Deals*.
- In the *Deck* window, if you hover your mouse arrow over any face down card with your mouse, a small "tool tip" image will appear to show you what the card is.

- On the *Selections* tab of the *StackView Control* window, there is a new button that will reverse all of the currently selected cards.
- As a new card selection *Event*, you are now able to *Swap* two cards in the deck under a multitude of procedural controls. For example, you can *Swap* different colors, same suits, etc.
- In the *Piles Control* window, there is an option in the *Riffle Shuffle Event* that allows you to visually explore the *Gilbreath Principle*.
- In the *Piles Control* window, you can now fully explore the impact of the *Australian Deal* with many new customization control parameters.
- In the *Piles Control* window, you can also explore *Elmsley* and *Jordan* counts, and their affects.
- Your saved *Session* files are now macros that can be called from other *Session* files. This allows a sequence of *Events* to be referred to by a single macro *Event* that can be used in other *Sessions*.
- The *Custom Deck* module has been moved from the *StackView Control* window (as a tab) to its own window. It can be accessed from the *Tools* menu. A new feature of the *Custom Deck* window is more visual clarity of whether your custom deck is retaining the stack values of the original deck from the *Deck Window*.

## **Reference: Deck**

The *Deck* window shows you all of the cards in the deck. Every time an event is performed, the *Deck* view is updated automatically to show the effect. If enabled, the *Stack Values*, *Position Values*, and *Selected Card Status* is shown.

If the *Deck* window is closed, you may reopen it by selecting it from the *View* menu, and select "Deck."

### ***Deck Views***

#### **Default Full Deck View**

The default *Deck* view shows the full deck in two rows. The top row shows the first 26 top cards of the deck, and the second row shows the 26 bottom cards of the deck.

#### **Piles View**

The *Piles* view in *StackView* shows separate *Piles*. Each *Pile* is numbered. The *Piles* are positioned on the screen in a way to minimize the size of the window. The *Deck* window is resized for each *Pile* definition. *Piles* can have as few as 1 card, or as many as 52 cards.

#### **Poker Deal View**

On the *Control* tab of the *StackView Control* window, you can perform *Poker Deals*. You can deal from 2 to 10 poker hands. The poker hands are shown in the *Deck* window in separate piles, with the balance of the deck in the remaining pile.

#### **Point of View**

Imagine spreading a deck of cards face down on a glass table surface. Now imagine going under the table and looking up at the face of the cards through the glass. This is the point of view that *StackView* shows you in the *Deck*



window. There are two places in *StackView* where you can change your point of view to look at the cards from above the table.

The first place is on the *Selection* tab in the *StackView Control* window, where you have the option to select your view from either beneath or above the table. This option selection operates in the default *Deck* view. Also, when you choose one of these option selections, the *Deck* window will be forced to the default view regardless of what the current view was.

The second place is on the *Piles Control* window. Here, you also have the option to select your view from beneath or above the table. However, in this case, these option selections will only affect the *Piles* view.

Figure 2 shows the Aronson stack in the default full deck view when viewing from beneath the table. Note that the Jack of Spades is the left most face up card. When you change views to viewing from above, as in Figure 3, you will note that the Jack of Spades is now the right-most face down card. This is exactly as you would see it if you actually did look above and beneath the spread deck.

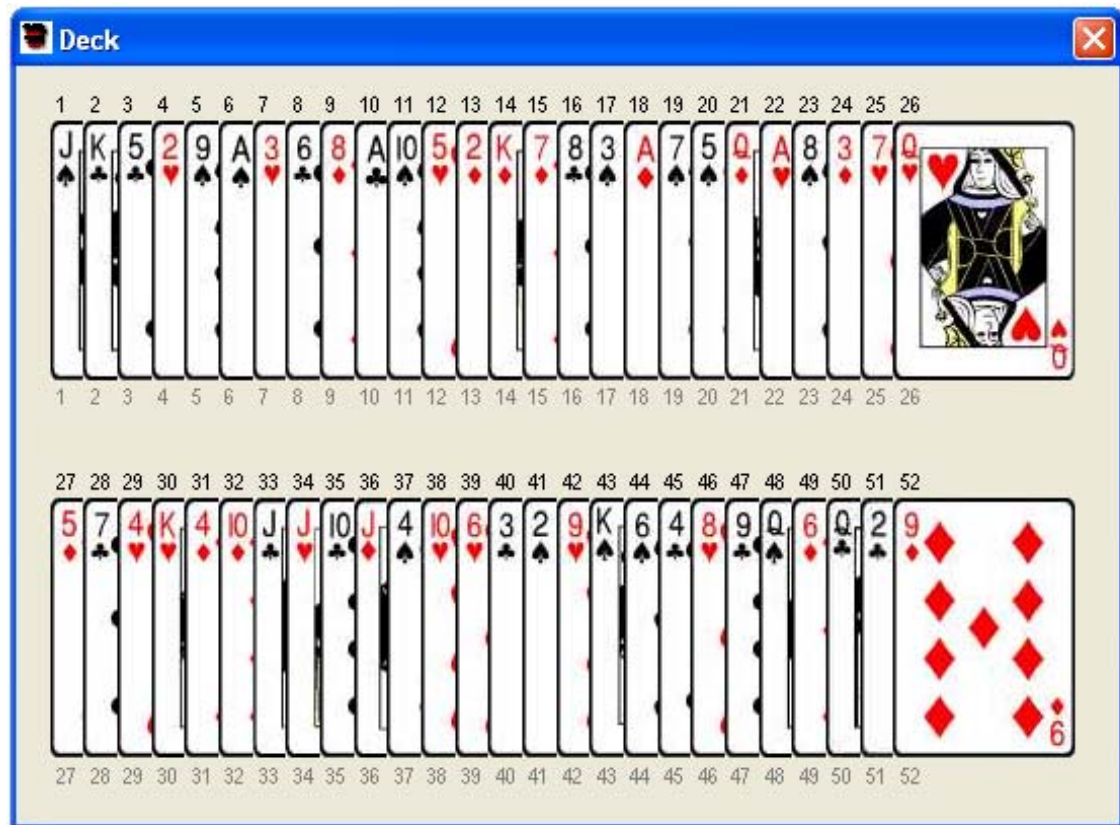
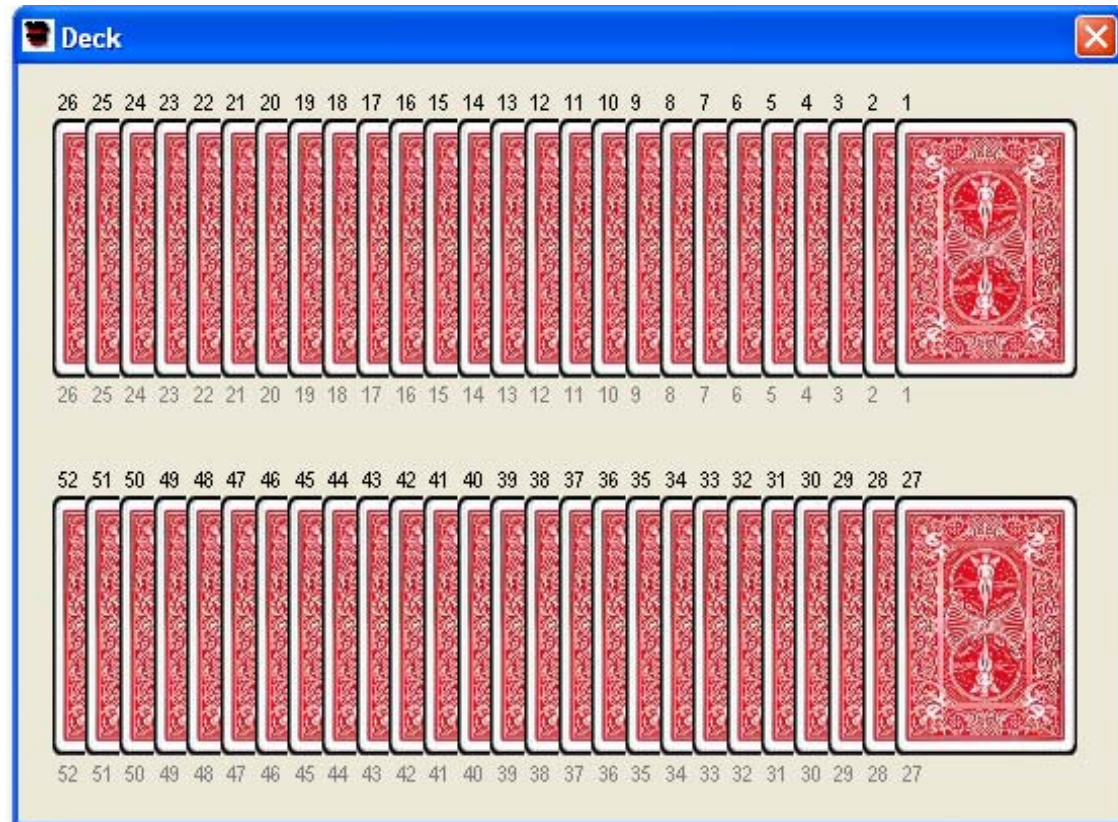


Figure 2



**Figure 3**

If you look at opposite sides of a *Piles* view as in Figure 4 and Figure 5, you will note that each individual *Pile* is correct. However, the relative positions of the *Piles* do not move. This is intentional to help you more readily keep track of which *Pile* you are looking at.

Also note that in each case, the *Stack Values* and *Position Values* are correctly placed in each view.

You may not change the *Point of View* for *Poker Deals*.

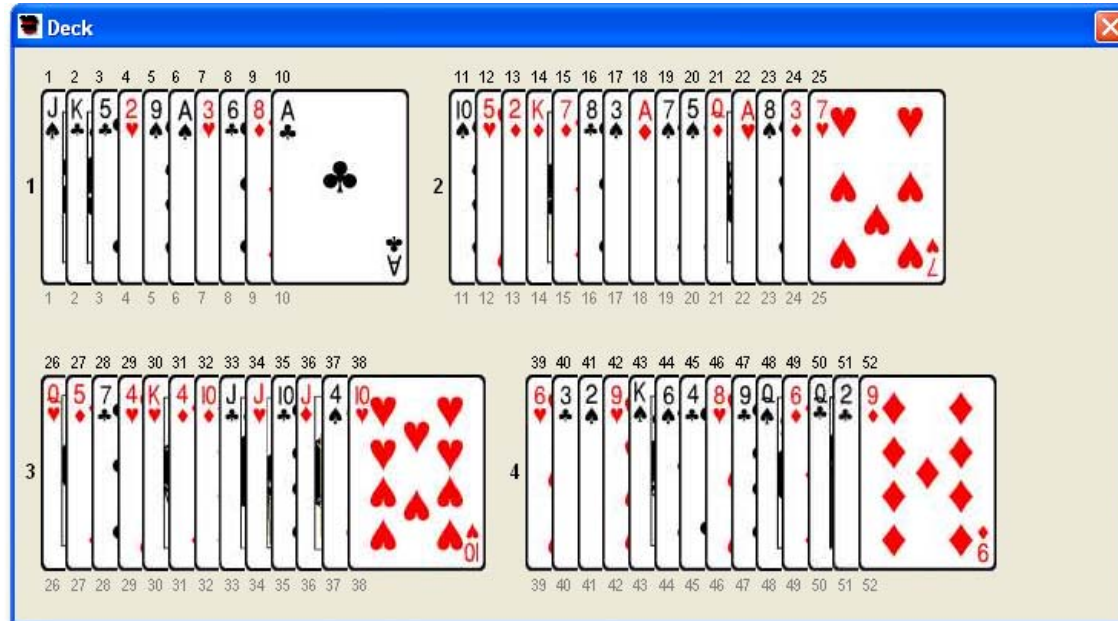


Figure 4

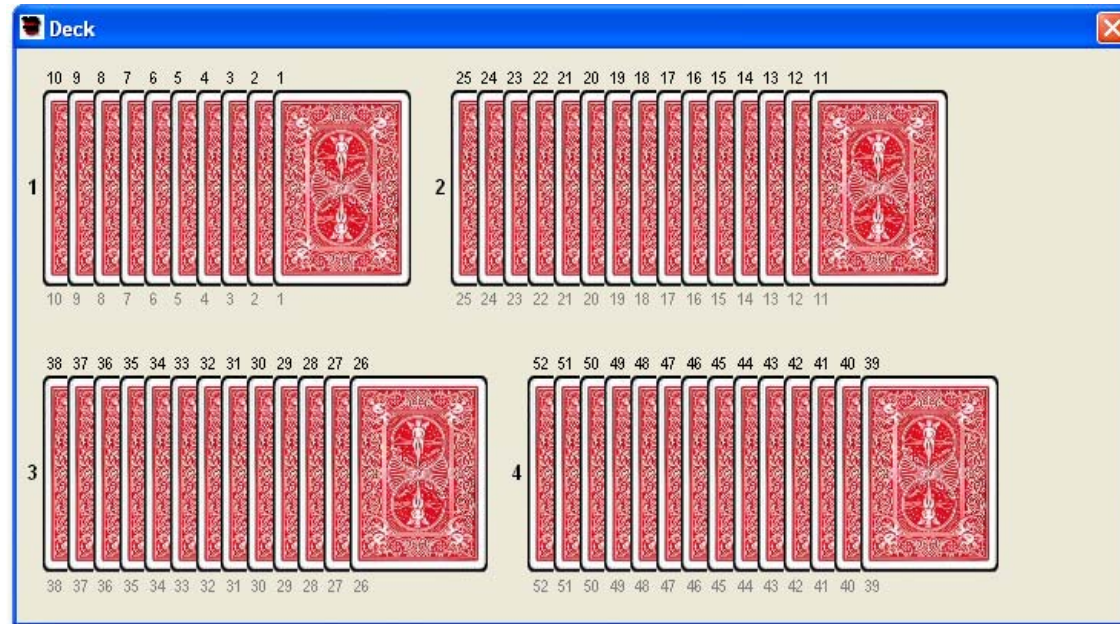


Figure 5

### ***Stack Values***

The *Stack Values* are the special numbers assigned to each card in your stack. In the Aronson stack, the *Stack Value* of the Jack of Hearts is always 34, regardless of where it is in the deck. If enabled, the *Stack Value* is always shown just above each card in the *Deck* window. The *Stack Value* is shown in a black colored font.

On the *Selection* tab of the *StackView Control* window, there is a checkbox labeled "Show Stack Values." When this checkbox is checked, the *Stack Values* are displayed in the *Deck* window. When the checkbox is not checked, the *Stack Values* are not displayed.

### ***Position Values***

The *Position Value* of any card is its ordinal position in the deck. StackView allows you to set the ordinal count from either the back or the face of the deck. The default count starts from the back, with the top card being position 1, and the bottom card being position 52. When the count starts from the face, the bottom card is position 1, and the top card is position 52. You can set from which direction you count on the *Selection* tab of the *StackView Control* window.

If enabled, the *Position Value* is always shown just below each card in the *Deck* window. The *Position Value* is shown in a gray colored font.

On the *Selection* tab of the *StackView Control* window, there is a checkbox labeled "Show Position Values." When this checkbox is checked, the *Position Values* are displayed in the *Deck* window. When the checkbox is not checked, the *Position Values* are not displayed.

### ***Selected Cards***

When a card is selected, if indicated, it is shown in three ways. First, the card image in the *Deck* window is highlighted in a yellow color. (Even if the card is reversed in the deck, if selected, its back design will have a yellow highlighted color.) The second indication of a selected card is its bolded *Stack Value* indicator. The third indicator is the name of the card in a text string field on the *Selections* tab of the *StackView Control* window.

On the *Selection* tab of the *StackView Control* window, there is a checkbox labeled "Highlight Selections." When this checkbox is checked, the selections are indicated in the *Deck* window, and in the selections text string field. When the checkbox is not checked, the selections are not indicated; the cards are not highlighted with a yellow color, the stack values are not bolded, and the card names in the selections text string field are replaced with asterisks. For more information on *Sessions*, see page 80.



### ***Reversing Cards in the Deck Window***

Double-clicking on the face of any card in the *Deck* window will reverse that card in the deck. If you are recording a *Session*, a ReverseCard(X) *Event* for that specific card will be recorded; in this case, the "X" will be the name of the card. At playback of the *Session Event*, that specific card will become reversed, regardless of where it is in the deck. If you double-click on any of the *Position Value* indicators, the card at that position will become reversed. While the effect of double-clicking on the card face or the *Position Value* is the same, the difference is in what is recorded in *Sessions*. If you double-click on the *Position Value* while recording a *Session*, whatever card is at that specific position will be reversed at playback. The recorded *Event* will be ReverseCard(X), where "X" will be a numerical *Position Value*. The ReverseCard *Event* does not work while viewing a *Poker Deal*.

In the *Deck* window, if you hover your mouse arrow over any face down card with your mouse, a small "tool tip" image will appear to show you what the card is.

## Reference: StackView Control

The *StackView Control* window is where most of the key activities are generated. The window has three tabs from which you select your actions. From the *Control* tab, you will be able to establish which stack you will be using, and do several standard (and not so standard) deck controls, including poker deals. On the *Selection* tab, you can have cards selected, and control the visible nature of the *Deck* window. On the *Sessions* tab, you can record your sessions for later playback.

If the *StackView Control* window is closed, you may reopen it by selecting it from the *View* menu, and select "Controls."

### ***Control Tab***

The *Control* tab has three main sections of *Events*. The *Arrange Cards* section allows you to establish the stack you are using, as well as reset the stack order. This section also contains the *Poker Deal* controls. The *Conventional Shuffles* section provides some standard *Events* for you to control your deck. The *Faro Shuffles* section provides you with a complete assortment of *Events* to explore full deck *Faros*. Most *Events* also give you the option to reverse cards in the deck.

### **Arrange Cards**

#### ***Set Stack***

StackView has several of the well-known stacks built in to its code. You can access these stacks with the *Set Stack Event*; there are quite a few to choose from. There is also a stack called *Default*, which you can define.

When you perform the *Set Stack Event*, the *Deck* window gets reset to the specified stack, and in that stack's original order. If you had any card selections or reversed cards in memory from prior events, they will be erased. The new stack that is displayed will only show the stack as it was saved with no selections



or reversed cards. (When stacks are saved, they only retain the stack values, card values and card positions.)

Before you press the *Set Stack* button, you must indicate which stack you want to use from the drop-down list. There are several well-know stacks to choose from.

One of the choices is labeled *Default (StackView.svf)*. If you have saved any stacks as the default, it will load automatically when you press the button (or when StackView loads initially). If there is no default file saved, then the *New Deck (Bicycle)* stack will be loaded as the default. For more information on creating *Default* stacks, see pages 204 and 231.

The standard stacks that you can choose from include: Aronson, Eight Kings, Ireland, Joyal (CHaSeD), Joyal (SHoCkeD), New Deck (Bicycle), New Deck (Fournier), Nikola, Osterlind, Si Stebbins (3), Si Stebbins (4), Stanyon, and Tamariz.

(You can also open any saved file by selecting the *Open Deck...* command from the *File* menu. For more information, see page 203.)

### ***Reset Current Deck Order***

When you perform the *Reset Current Deck Order Event*, the *Deck* window resets the order of the currently loaded deck to its original stack order. This means that *Stack Value 1* will be on top of the deck, and *Stack Value 52* will be at the face of the deck. If you had any card selections or reversed cards in the deck, they will be retained. If you would like to clear any selected cards or reversed cards, you may do so on the *Selections* tab. For more information see page 64.

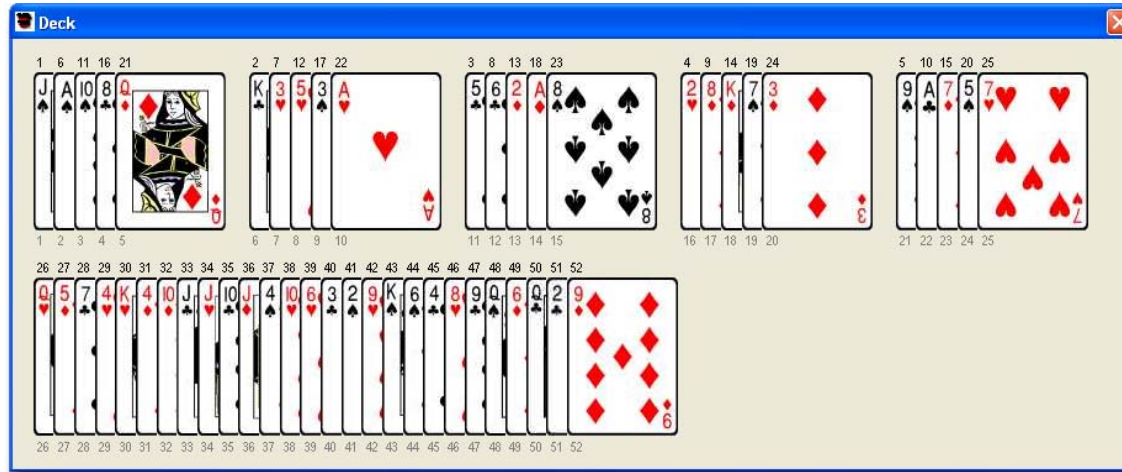
### ***Poker Deal***

The *Poker Deal Event* allows you to deal out from two to ten hands of poker. At the end of the poker deal, the *Deck* window displays the hands in separate piles. Any un-dealt remaining cards are shown in a single additional pile.

Before you press the *Poker Deal* button, you must specify the number of hands to deal. Even though the cards are shown in their respective hands in the *Deck* window, the deck is ready for additional *Events* to be performed. After a deal, the StackView deck order in the computer's memory reflects the poker hands being assembled *Backwards* as described in the next section, titled "Assemble Poker Deal."

When cards are being shown in poker hands, you may discard one of the cards by double-clicking either the card face or the card's *Position Value* indicator. When you double-click the card (or its *Position Value* indicator), that card will be sent to the bottom of the remaining talon, and the top card of the talon will be added to the poker hand.

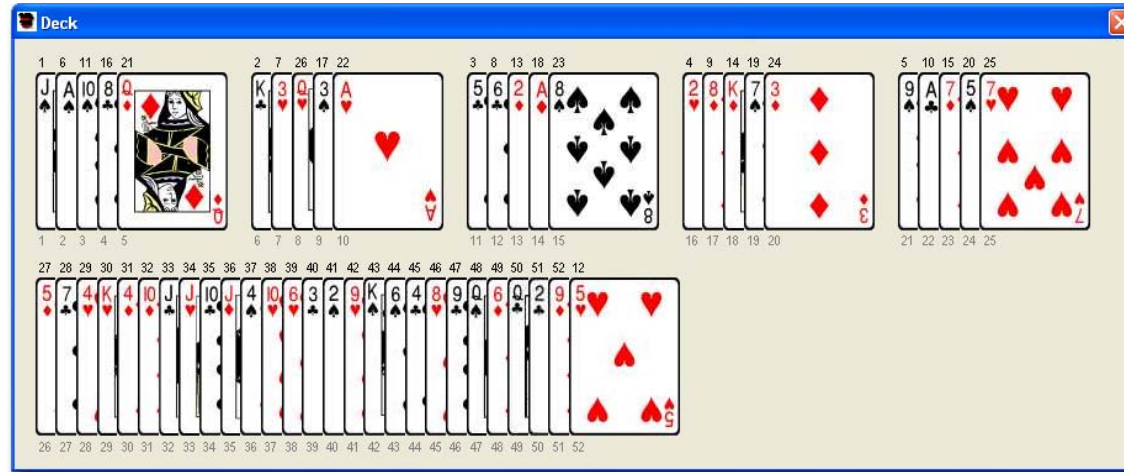
For example, if you start with the Aronson stack and do a five hand poker deal, you will see the cards as in Figure 6. If you were to double-click the Five of Hearts, you will see that the Five of Hearts is now at the bottom of the talon, and it has been replaced with the next card of the deck (talon), which is the Queen of Hearts as shown in Figure 7.



**Figure 6**

*Poker Deals* and *Discards* will retain any selected cards' or reversed cards' status.

If you are in a *Sessions* recording mode when discarding a card, the *Event* will be recorded differently depending on where you double-click. In the example above, if you double-clicked the face of the Five of Hearts, the *Event* would have been recorded as **PokerDiscard(5H)**. If you had double-clicked the Five of Hearts' *Position Value* (the number 8 beneath the card), the *Event* would have been recorded as **PokerDiscard(8)**.



**Figure 7**

When played back, the *Event* `PokerDiscard(5H)` would try to discard the Five of Hearts. However, the *Event* **`PokerDiscard(8)`**, when played back, would try to discard whichever card is in the 8<sup>th</sup> position of the cumulative poker hands.

For more information on *Sessions*, see page 80 and 221.

### ***Assemble Poker Deal***

After a *Poker Deal*, you will need to reassemble the deck. `StackView` provides you with three alternative assembly techniques.

After a *Poker Deal*, even though the default deck order is an implied *Backwards* assembly, you should use the *Assemble Poker Deal Event* button to explicitly reassemble the cards. This is particularly important if you are recording a *Session* for others to examine. To use this button, you must specify the order of assembly by selecting the appropriate choice from the drop-down list.

#### *Backwards Assembly*

The face up undealt block of cards is placed on top of the face up last hand (dealer's hand) to create a new larger block. (i.e. If you dealt a a *5 Hands Poker Deal*, the undealt cards are placed on the 5<sup>th</sup> hand.) The new larger face up block of cards is placed face up on the preceding face up hand. (Continuing the example, the new block is placed on the 4<sup>th</sup> hand.) This is repeated until all the hands are collected.

#### *Forwards Assembly*

Some poker demonstrations may require this order of reassembly.

The face up undealt block of cards is placed on top of the first dealt hand to create a new larger block. The new larger face up block of cards is placed face up on the second face up hand. This is repeated until all the cards are collected.

#### *Unwind Assembly*

Finally, you may also perform a poker demonstration that unwinds the poker deal for the next part of the routine. Unwinding a poker deal returns the deck back to the same order it was before the poker deal. Select the *Unwind* choice from the drop-down list. (It is not my intention to teach unwinding techniques, but rather to provide the capability for *Session* recording.)

#### ***Refresh Deck View***

The *Refresh Deck View* button repaints the *Deck* window to its default full deck view. (See page 24.)

### **Conventional Shuffles**

This section provides *Event* controls for some standard shuffle and card handling. Some of the *Events* give you the option to simulate the shuffles with purely random handling, or with specific control.

#### ***Riffle***

The *Riffle Shuffle Event* simulates a standard shuffle where the cards are cut, and the two halves are randomly interlaced together, as in Figure 8. You have

the option to protect a specified block of cards at either the top of the deck or at the bottom of the deck. You also have the option of reversing either half of the deck before the riffle shuffle is completed.



**Figure 8**

*Random*

When the *Random* selection is chosen, the deck is cut randomly near half, and the two sections are randomly interlaced together as they would be with a true riffle shuffle. The interlacing starts with a randomly chosen half, progresses randomly, and finishes with a random half.

### *Protect Block*

When the *Protect Block* selection is chosen, you also have to specify how many cards to protect in the block, and whether the protected block is from the top or bottom of the deck.

### **Number**

Enter the number of cards to protect in the *Number* field. You may enter as few as 1 card, or as many as 52 cards to protect. StackView will try to randomly cut the deck in the middle of the unprotected cards, and then randomly interlace the two halves while protecting the appropriate block of cards.

### **Location**

Using the *Location* drop down list, select either the *Top* or *Bottom* block for protection. When StackView executes the riffle shuffle, the appropriate block of cards will remain unshuffled at the requested part of the deck.

### *Reverse Top/Bottom Block*

You may check only one of the two boxes to reverse one of the blocks of cards before the shuffle is executed. If you are protecting a block of cards, the protected block is determined after the appropriate half is reversed. For example, starting with the Aronson Stack, Figure 9 shows the result of protecting ten cards at the top of the deck, but first reversing the top cut portion. You can see that in this random shuffle, the initial cut was made 33 cards deep. The top half was reversed and shuffled while protecting the top ten cards (*Stack Values* 24 through 33).

Number Location

Rifle  Random  Protect Block

Overhand  Random  Protect Block

10 Top

Reverse Top Block  
 Reverse Bottom Block

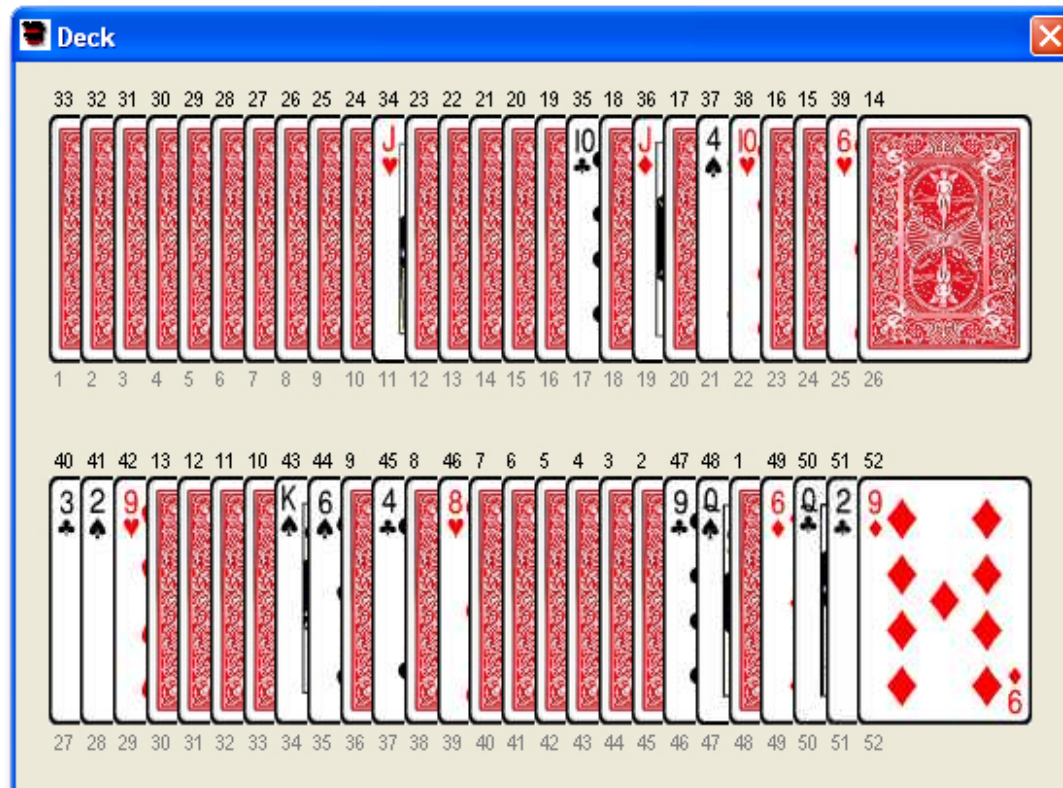


Figure 9



### ***Overhand***

The *Overhand Shuffle Event* simulates a standard hand to hand shuffle as demonstrated in Figure 10. The right hand holds the deck from below, and the left thumb draws small groups of cards from the top of the right hand portion. Each successive group of cards settles on the prior group, until all of the cards have been shuffled off into the left hand. You have the option to protect a specified block of cards at either the top of the deck or at the bottom of the deck.

### ***Random***

When the *Random* selection is chosen, the deck is shuffled in a purely random manner. The groups of cards that are drawn by the left hand will vary in size randomly and separately.

### ***Protect Block***

When the *Protect Block* selection is chosen, you also have to specify how many cards to protect in the block, and whether the protected block is from the top or bottom of the deck. In either case, (*Top* or *Bottom*), the protected block will need to be cut back to its correct position after the overhand shuffle.

### **Number**

Enter the number of cards to protect in the *Number* field. You may enter as few as 1 card, or as many as 52 cards to protect. StackView will protect exactly the amount of specified cards, and then randomly overhand shuffle the balance of the deck. A final cut is executed to get the protected block to the correct end of the deck.

### **Location**

Using the *Location* drop down list, select either the *Top* or *Bottom* block for protection. When StackView executes the overhand shuffle, the appropriate block of cards will remain unshuffled at the requested part of the deck.



**Figure 10**

***Cut***

The *Cut Event* is just a straight forward complete cut of the deck. A portion of the top of the deck is cut and placed to the bottom of the deck to complete the cut. StackView provides some options in the *Cut Event*.

### *Random*

When the *Random* option selection is made, the portion of cards that is cut from the top of the deck is purely random. The random cut can be anywhere from 1 to 52 cards in size.

### *Precise*

When the *Precise* selection is made, you must also enter a number from 1 to 52 in the provided field to specify how many cards are to be cut.

### *Special Random*

When the *Special Random* selection is made, you must also select one of the options from the dropdown list box. The choices include:

#### **Quarter**

The random cut will be approximately a *Quarter* deep from the top of the deck. The actual random range that is cut is anywhere from 8 to 17 cards.

#### **Third**

The random cut will be approximately a *Third* deep from the top of the deck. The actual random range that is cut is anywhere from 11 to 22 cards.

#### **Half**

The random cut will be approximately a *Half* deep from the top of the deck. The actual random range that is cut is anywhere from 16 to 35 cards.

#### **Two Thirds**

The random cut will be approximately *Two Thirds* deep from the top of the deck. The actual random range that is cut is anywhere from 28 to 39 cards.

#### **Three Quarters**

The random cut will be approximately *Three Quarters* deep from the top of the deck. The actual random range that is cut is anywhere from 34 to 43 cards.

### **Shallow**

The random cut will be less than half of the deck deep from the top of the deck. The actual random range that is cut is anywhere from 5 to 24 cards.

### **Deep**

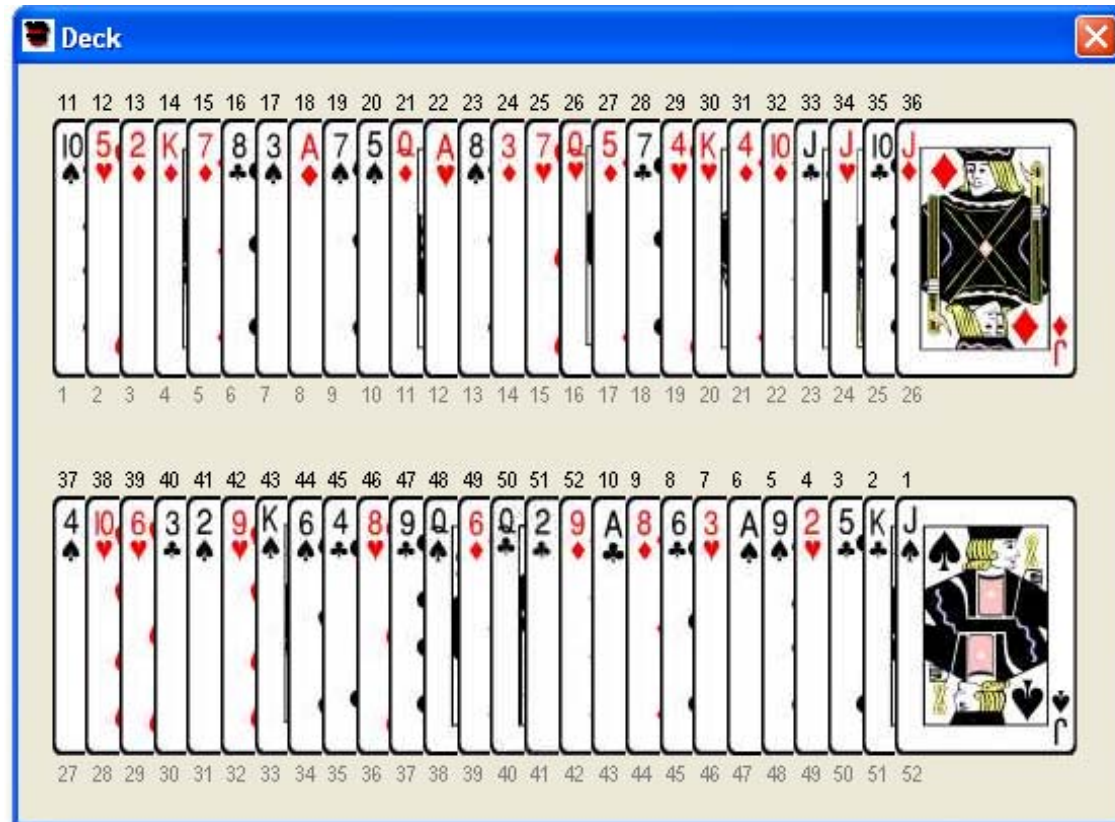
The random cut will be more than half of the deck deep from the top of the deck. The actual random range that is cut is anywhere from 28 to 47 cards.

### *Reverse Top/Bottom Block*

You may check only up to one of the two boxes to reverse one of the blocks of cards before the cut is executed. If the *Top Block* is reversed, it refers to the original top portion of the deck that is reversed before the cut is completed. Similarly, if the *Bottom Block* is reversed, it is the original bottom portion of the deck before the cut is completed.

### ***Run Single Cards***

The *Run Single Cards Event* is similar to an *Overhand Shuffle*, except that the cards that are being shuffled in to the left hand are being run singly (one at a time). You specify the number of single cards that will be run, and then the balance of the deck is tossed as a block onto the shuffled cards. As an example, starting with the Aronson Stack, if you specify 10 cards in the input box, and press the *Run Single Cards Event* button (leave the other two settings unchecked), you will get the new deck order as shown in Figure 11. Essentially, the top ten cards were set in the reverse order, and cut to the bottom of the deck.



**Figure 11**

*Inverse*

When the *Inverse* checkbox is checked, the *Run Single Cards Event* will be performed backwards. Doing an *Inverse Event* gets the deck to the order it would have been in before doing the same regular *Event*.

*Reverse Run Cards*

When this option is selected, the single cards that have been run into the left hand are turned over as a block before the final block is thrown on top of them. For example, starting in Aronson order, doing the *Run Single Cards Event* with 12 cards, with the *Reverse Run Cards* option selected will result in the deck order shown in Figure 12.

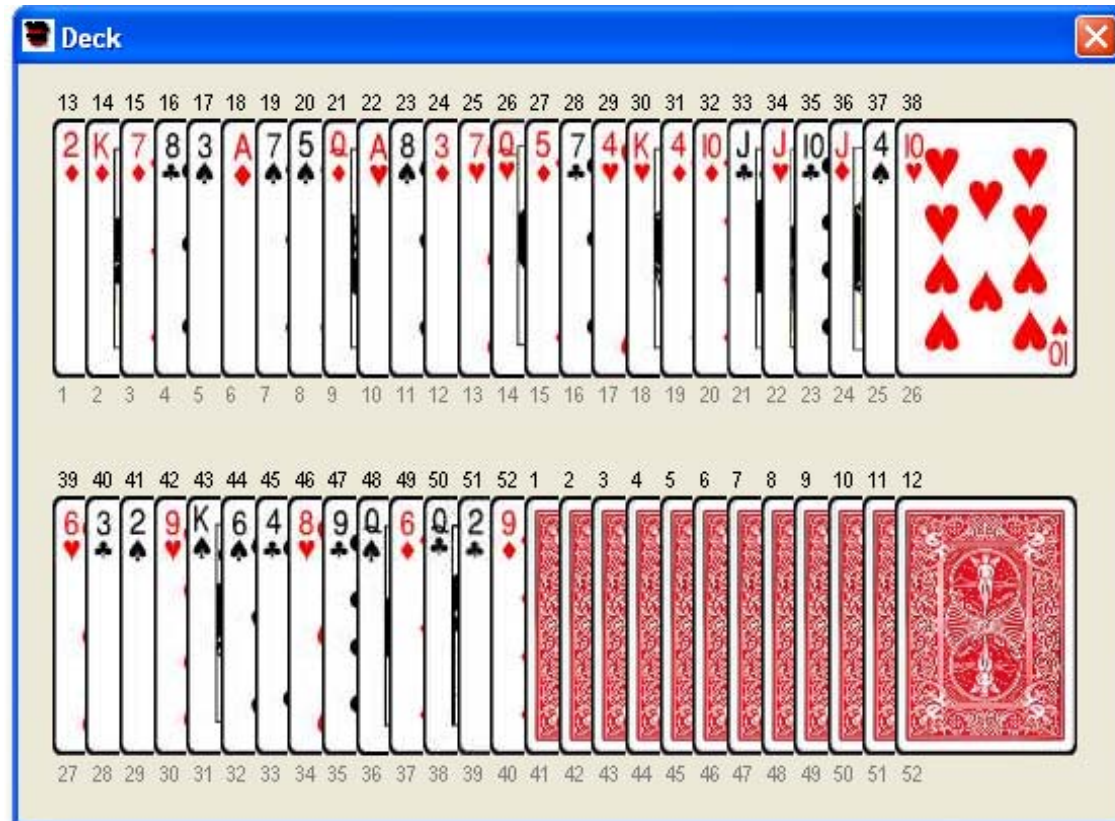


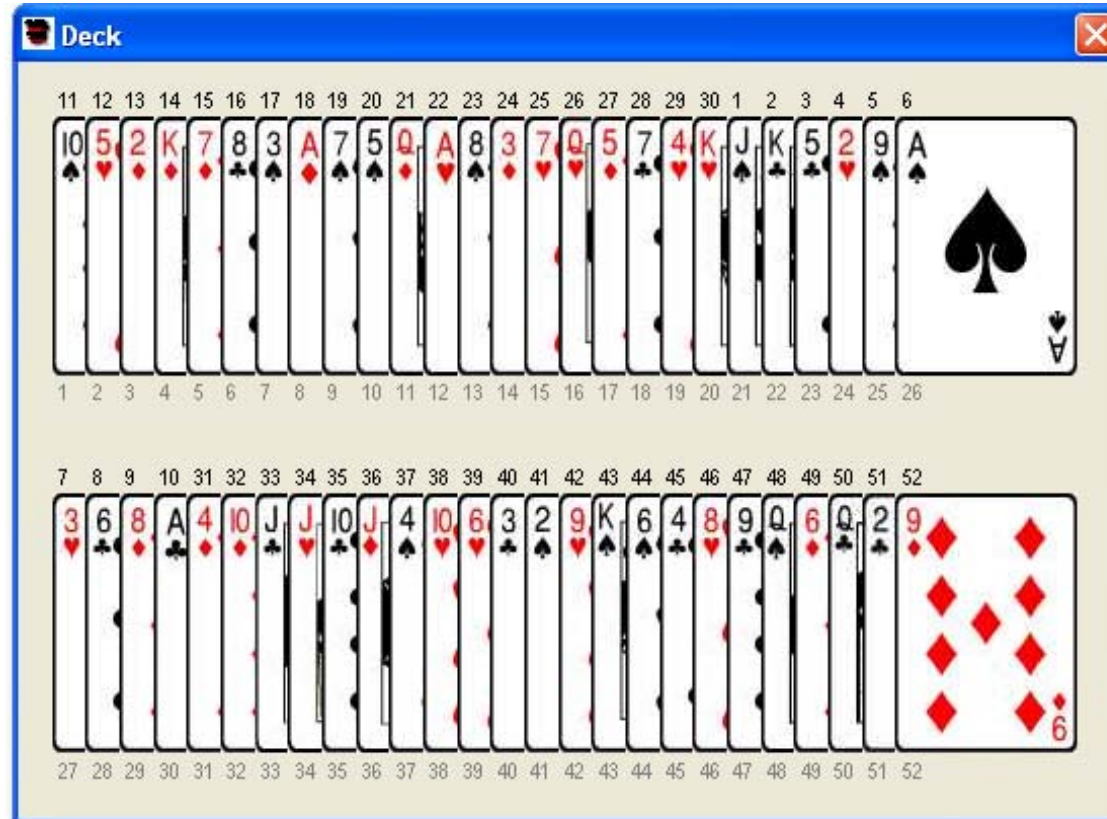
Figure 12

***Shift Top Block***

The *Shift Top Block Event* moves a specified number of cards from the top of the deck to a specified position in the balance of the deck. You must specify the number of cards to move in the "Block" field, and you must specify the depth of the insertion point in the "Depth" field. For example, starting with the Aronson



Stack, doing a *Shift Top Block Event* with a *Block* setting of 10, and a *Depth* of 20, you can see the resulting deck as shown in Figure 13.



**Figure 13**

You can see that the top 10 cards were cut, and moved to an insertion point 20 cards deep into the remaining block. When you specify *Block* and *Depth*, the



sum of your two numbers can not exceed 52. If the sum was greater than 52, there would not be enough cards in the remaining *Block* for the specified *Depth*.

#### *Inverse*

When the *Inverse* checkbox is checked, the *Shift Top Block Event* will be performed backwards. Doing an *Inverse Event* gets the deck to the order it would have been in before doing the same regular *Event*.

#### *Reverse Top Block*

When this option is selected, the top specified block will be turned over as a block before it is inserted at the specified depth. For example, starting in Aronson order, doing the *Shift Top Block Event* with a *Block* of 5 and a *Depth* of 35, with the *Reverse Top Block* option selected will result in the deck order shown in Figure 14.

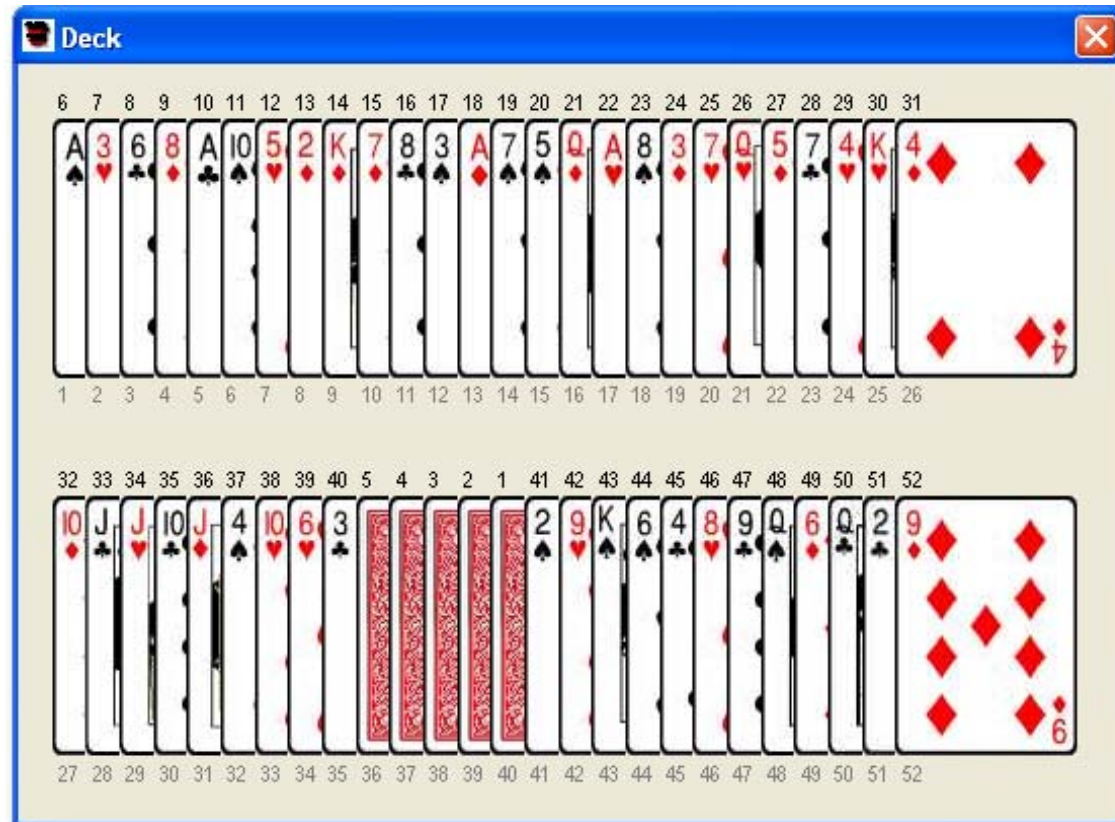


Figure 14

***Move Card***

The *Move Card Event* moves a single card from a specified position to another specified position in the deck. You must specify the originating position in the "From" field, and the destination position in the "To" field before pressing the

*Event* button. For example, starting in Aronson order, doing a *Move Card* with a "From 20" and a "To 5" will result in the deck order as shown in Figure 15.

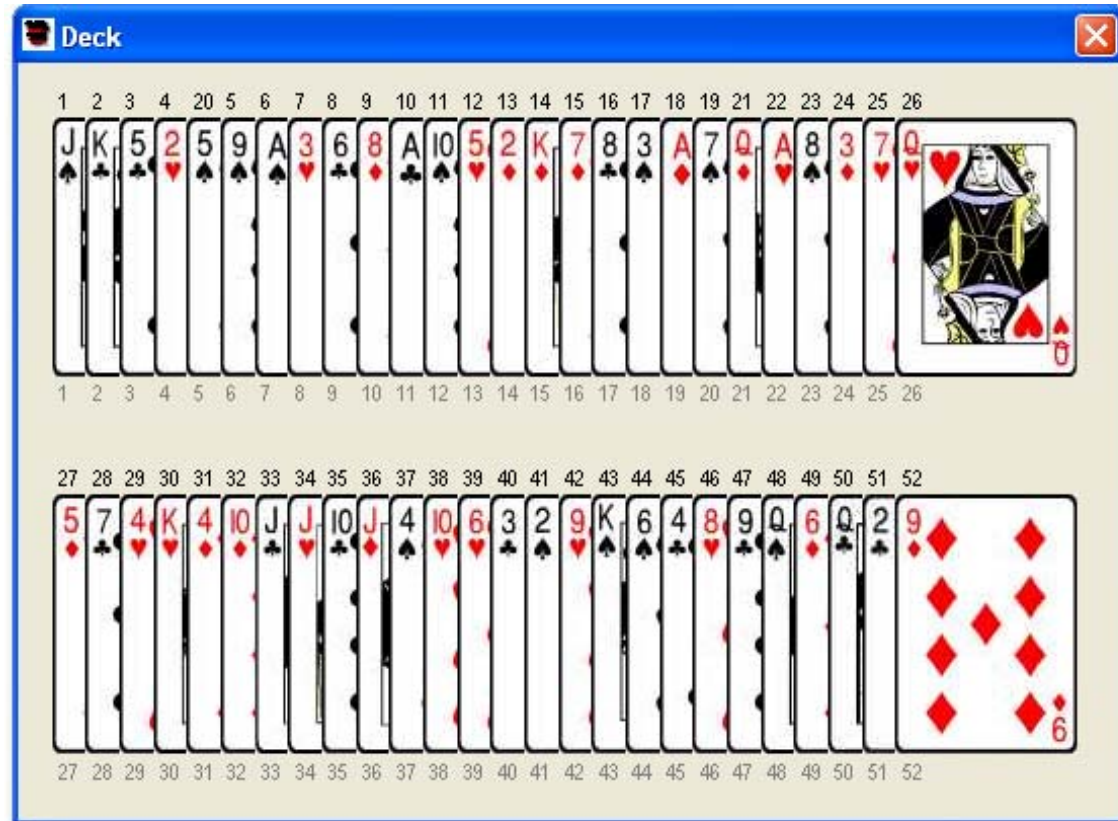


Figure 15

### *Inverse*

When the Inverse checkbox is checked, the *Move Card Event* will be performed backwards. Doing an *Inverse Event* gets the deck to the order it would have been in before doing the same regular *Event*.

### *Reverse Card*

When this option is selected, the single card that is being moved will be turned over during the *Event*. For example, starting in Aronson order, doing the *Move Card Event* with "From 10" and a "To 40" settings, with the *Reverse Card* option selected, will result in the deck order shown in Figure 16.

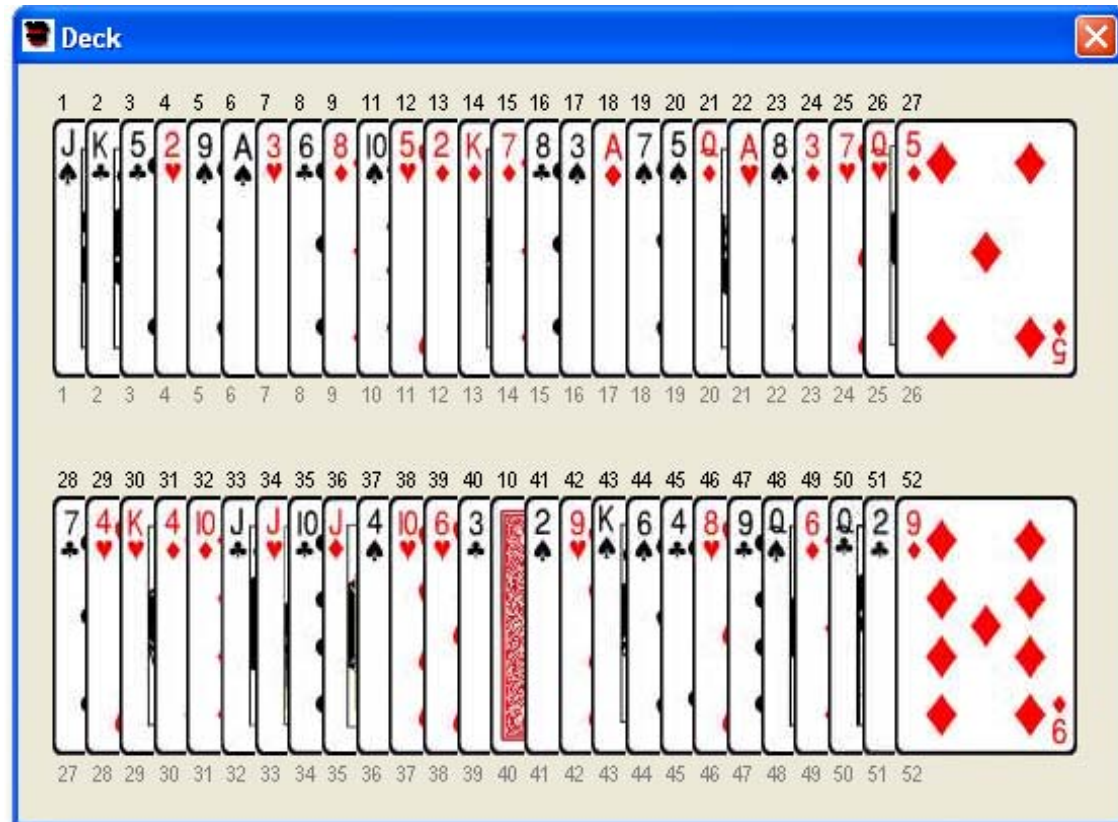


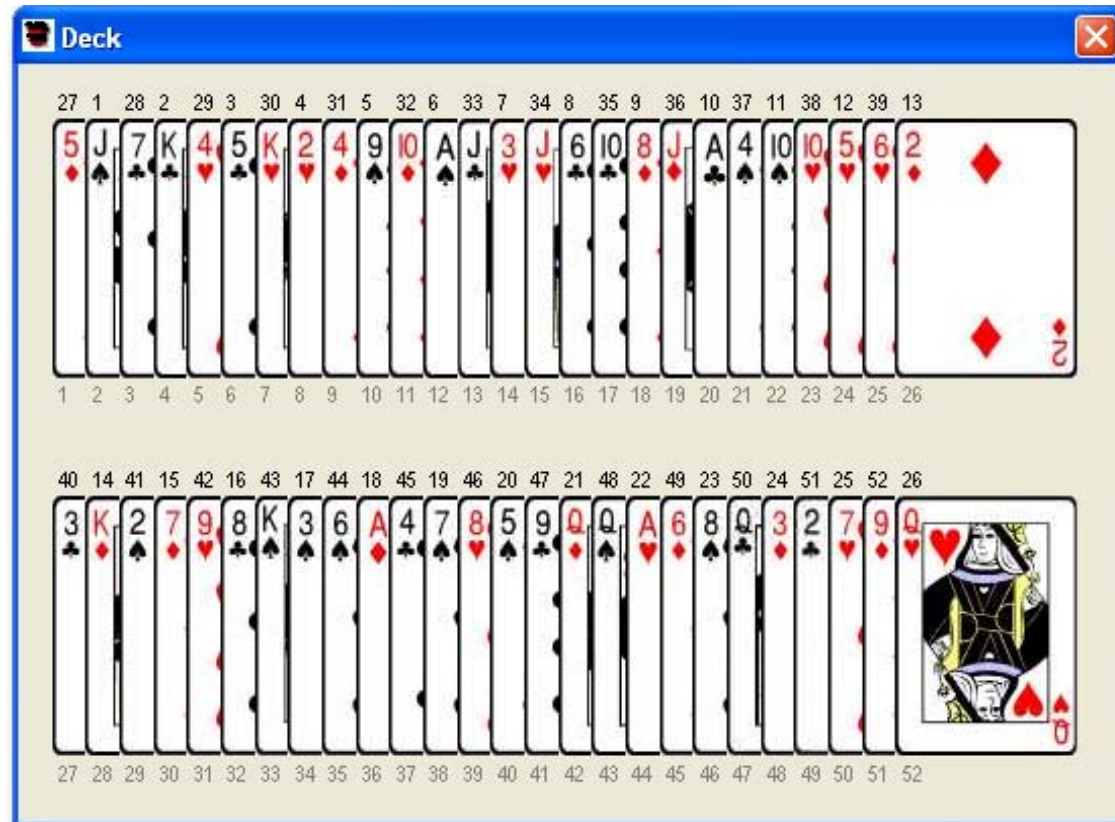
Figure 16

### Faro Shuffles

A *Faro* shuffle is a perfect interleaving of cards. StackView offers many variations and controls in simulating *Faro* shuffles.

### ***InFaro***

When the *InFaro* shuffle is executed with the *Standard* option button selected, first the deck is divided exactly in half (26 cards in each half). The cards are then perfectly interleaved so that the original top card is shuffled “in” (under) the top card of the bottom half block. Starting in Aronson order and doing a *Standard InFaro* shuffle will result in the deck order as shown in Figure 17. You can see that the original top card is now in the second position, and the rest of the deck is perfectly interleaved.



**Figure 17**

An interesting fact about this shuffle is that if you do 26 consecutive *InFaro* shuffles, the deck will end up in completely reversed order. Another 26 *InFaro* shuffles will return the deck to its original order. (Not very practical – however, see the similar information regarding the *OutFaro* shuffle on page 56.

### ***OutFaro***

When the *OutFaro* shuffle is executed with the *Standard* option button selected, first the deck is divided exactly in half (26 cards in each half). The cards are then perfectly interleaved so that the original top card is shuffled "out" (over) the top card of the bottom half block. Starting in Aronson order and doing a *Standard OutFaro* shuffle will result in the deck order as shown in Figure 18. You can see that the original top card is still in the top position, and the rest of the deck is perfectly interleaved.

An interesting fact about this shuffle is that if you do eight consecutive *OutFaro* shuffles, the deck will end up in its original order.



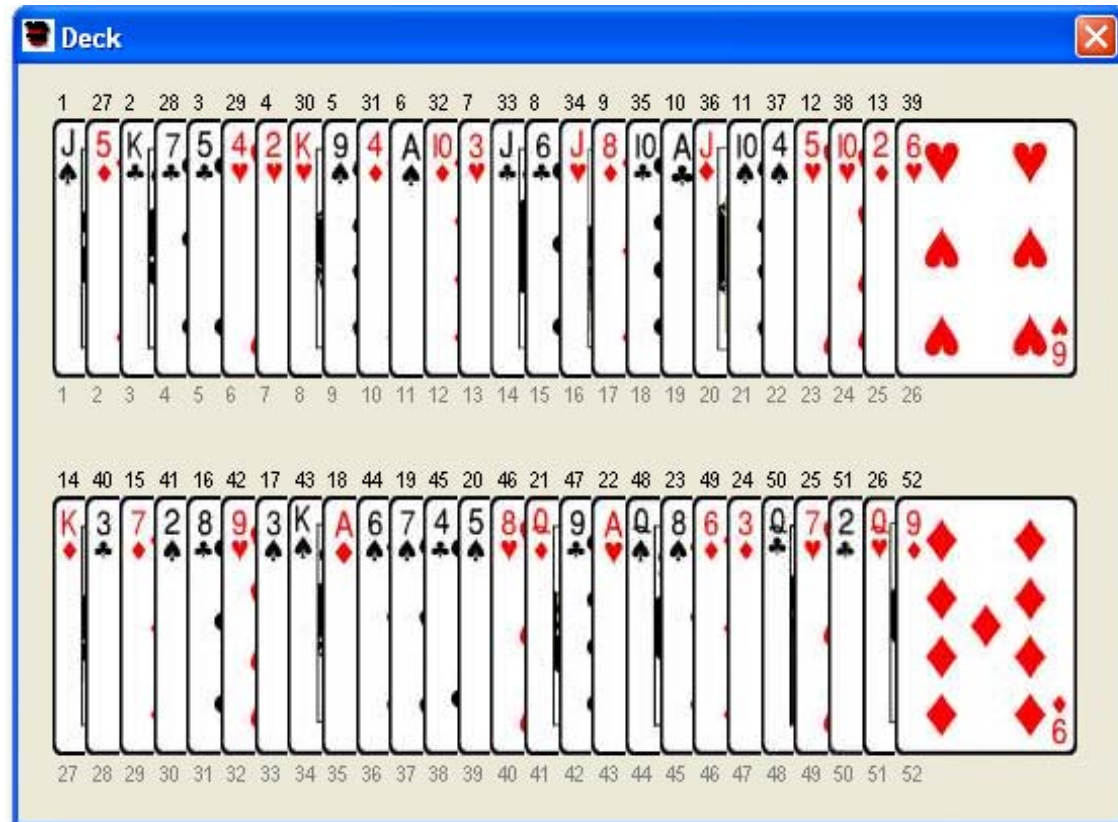


Figure 18

***Special Faros***

In both the *InFaro* and *OutFaro* shuffles, you have additional parameters to control by selecting the *Special* option button. (Special Faros are also referred to as Partial Faros.)

### *From Top*

The *From Top* field allows you to specify how many cards to cut in the top half before doing the *Faro* shuffle. You may specify from 1 to 51 cards to cut. The sum of *From Top* and *Interior Position* may not exceed 52.

### *Start Weave*

You must specify whether the *Faro* weave should start at the *Top* or the *Bottom* of the deck. See the examples at the end of this section.

### *Interior Position*

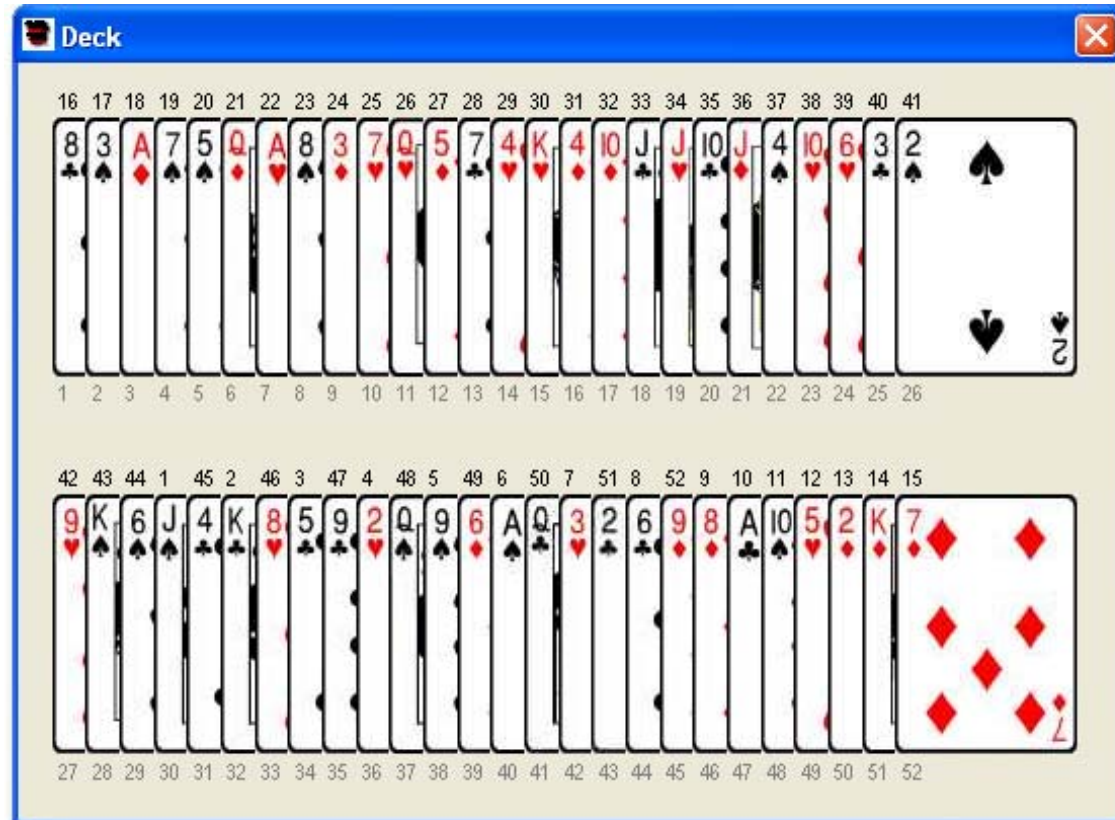
The *Interior Position* field allows you to specify how deep into the bottom half the *Faro* weave should start. If you had set the *Start Weave* option to *Top*, then the *Interior Position* depth is counted from the top of the remaining portion of the deck. If you had set the *Start Weave* option to *Bottom*, then the *Interior Position* depth is counted from the bottom of the remaining portion of the deck. The sum of *From Top* and *Interior Position* may not exceed 52.

If the *Interior Position* field is left empty, the implied value is 1, which is the top card if you set the *Start Weave* option to *Top*. Similarly, the implied value of 1 would be the bottom card if you set the *Start Weave* option to *Bottom*.

### *Example 1*

Starting in Aronson order, and doing a *Special OutFaro* shuffle with a *From Top* set to 15 cards, *Start Weave* set to *Top*, and *Interior Position* set to 30, will result in a deck order as shown in Figure 19. With these settings, the top card of the cut portion is the Jack of Spades, and the top card of the remaining portion is the Eight of Clubs. Therefore the 30<sup>th</sup> card down in the remaining portion is the Four of Clubs. You will note that the Jack of Spades is to the left of the Four of Clubs since it is an *OutFaro*. (Had it been an *InFaro*, the Jack of Spades would have been to the right of the Four of Clubs.) Also note that positions 46 through

52 are cards from the original top portion that did not have any cards to weave into. They just finish at the bottom of the deck.



**Figure 19**

*Example 2*

Starting in Aronson order, and doing a *Special InFaro* shuffle with a *From Top* set to 20 cards, *Start Weave* set to *Bottom*, and *Interior Position* set to 18, will

result in a deck order as shown in Figure 20. With these settings, the bottom card of the cut portion is the Five of Spades, and the top card of the remaining portion is the Queen of Diamonds. Therefore the 18<sup>th</sup> card from the bottom in the remaining portion is the Ten of Clubs. You will note that the Five of Spades is to the right of the Ten of Clubs since it is an *InFaro*. (Had it been an *OutFaro*, the Five of Spades would have been to the left of the Ten of Clubs.) Also note that positions 1 through 5 are cards from the original top portion that did not have any cards to weave into. They just finish at the top of the deck.

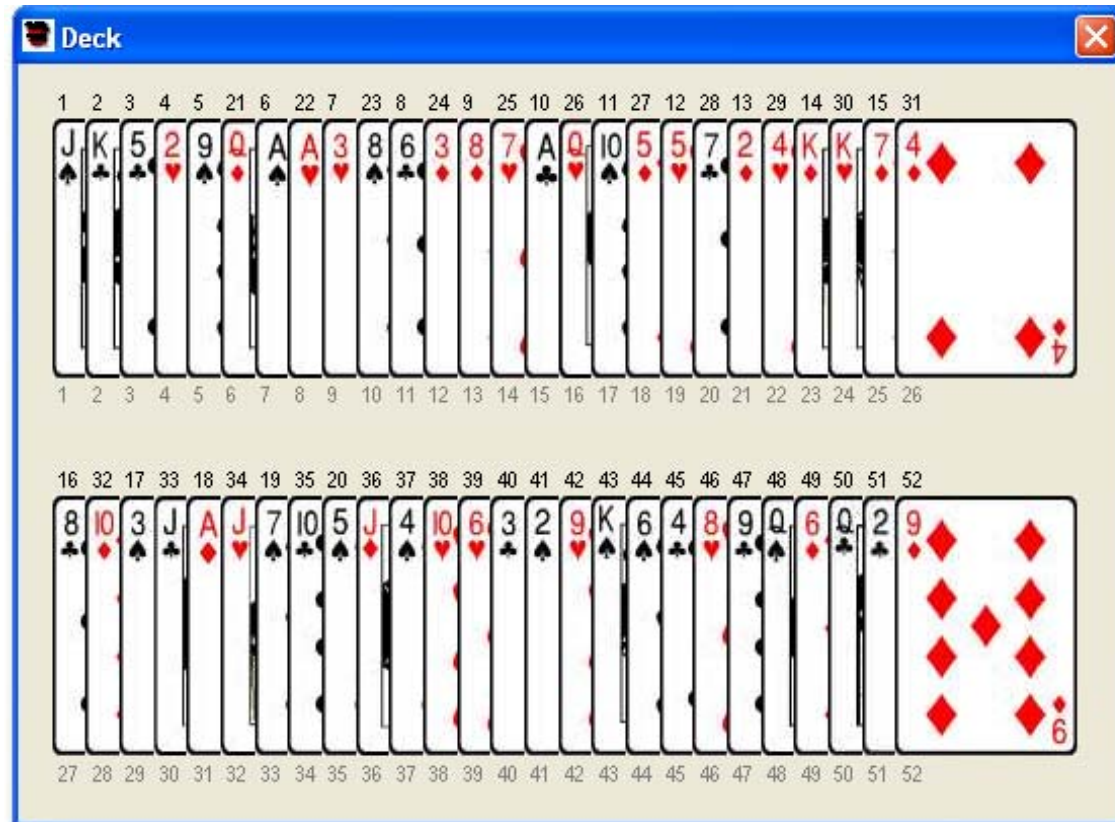


Figure 20

***Inverse***

When the *Inverse* checkbox is checked, the *Faro Shuffle Event* (*Standard* or *Special*) will be performed backwards. Doing an *Inverse Event* gets the deck to the order it would have been in before doing the same regular *Event*.

This feature can be useful in preparing a stack that you want to openly (*Faro*) shuffle before you begin performing. For example, you would use your favorite stack, but you also want to see you shuffle the deck before you begin. You can use StackView to do two *Inverse OutFaro* shuffles. Arrange the deck as shown in the *Deck* window, and put the cards in the case. When you are ready to perform, take the cards out of the case, perform two *OutFaro* shuffles, and your stack will now be in the correct order.

### ***Reverse Top Block***

When this option is selected, the *From Top* specified block will be turned over as a block before it is *Faro* shuffled into the remaining block. For example, starting in Aronson order, doing a *Special OutFaro* with a *From Top* specification of 10, a *Start Weave* from the *Top*, and an *Interior Position* of 12, with the *Reverse Top Block* option selected will result in the deck order shown in Figure 21.

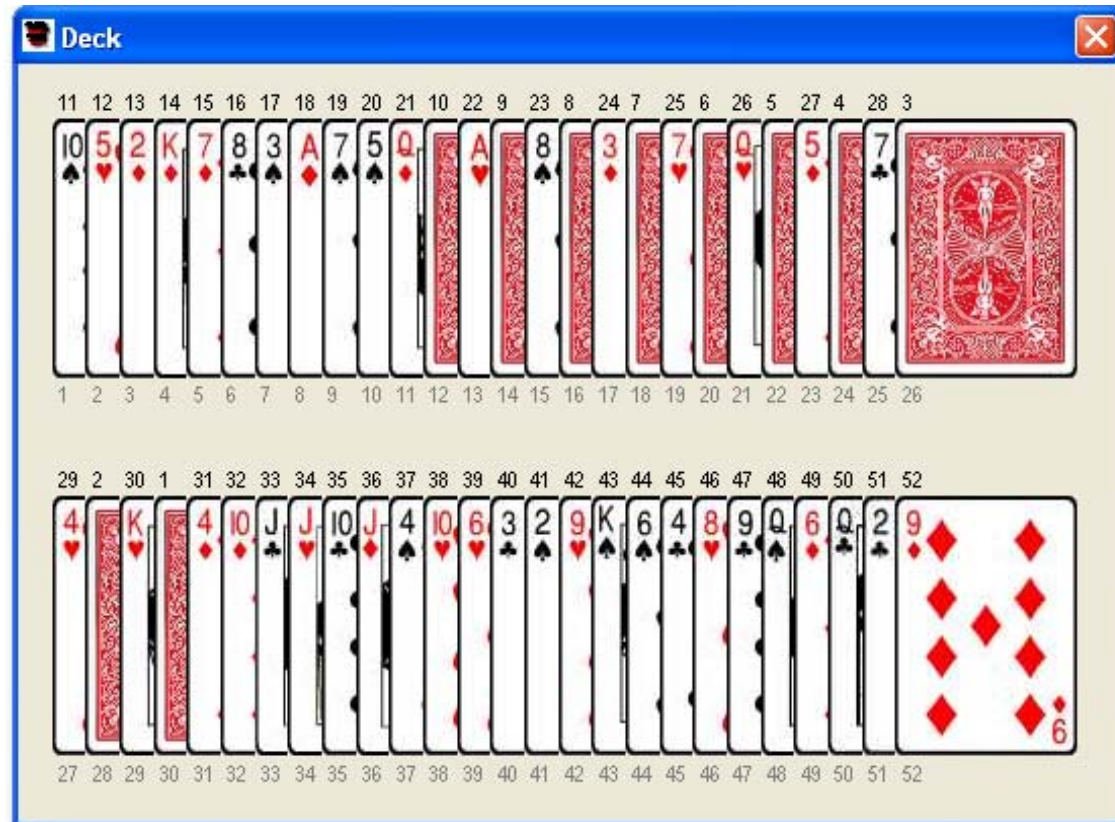


Figure 21

***Selection Tab***

As the name implies, the *Selection* tab allows you to simulate the selection of cards. In addition, there are several settings to control how the deck displays in the *Deck* window.

## **Deck Visual Indicators**

Whenever you perform any *Events* with StackView, the results of your actions are shown in the *Deck* window. The *Deck* window will also show several pieces of information: the *Position Values*, the *Stack Values*, and whether a card has been selected, are all things that you may or may not want to be visible. You can also change whether the *Position Values* are counted from the face or the back of the deck. You are also able to change the view of the deck from between the top and the bottom of the table. Figure 22 shows a close-up of a couple of selected cards.





**Figure 22**

When a card is selected, if indicated, it is shown in three ways. First, the card image in the *Deck* window is highlighted in a yellow color. (Even if the card is reversed in the deck, if selected, its back design will have a yellow highlighted color.) The second indication of a selected card is its bolded *Stack Value* indicator. The third indicator is the name of the card in a *Selections* text string field.

### ***Highlight Selections***

When the *Highlight Selections* checkbox is checked, any selected cards will be fully highlighted. The selected card will be highlighted with a yellow color. The *Stack Value* will be in a bold font. And the *Selections* text field will contain the name of the card.

When the *Highlight Selections* checkbox is unchecked, the card is not highlighted, and the *Stack Value* font is not bolded. Also, the *Selections* text field data will be replaced with asterisks (\*\*\*) .

### ***Show Stack Values***

When the *Show Stack Values* checkbox is checked, the *Stack Values* of each card are displayed above the cards. When it is unchecked, the *Stack Values* are not displayed. Having this ability to turn the *Stack Values* on or off is useful in practicing certain effects where you must test your ability to discern certain cards.

### ***Show Position Values***

When the *Show Position Values* checkbox is checked, the *Position Values* of each card are displayed below the cards. When it is unchecked, the *Position Values* are not displayed.

### ***Count From Back/Face***

The default position count is to count from the top/back of the deck to the bottom/face of the deck. The top card is position 1, and the bottom card is position 52. However, in some routines and practice sessions, it is practical to count the deck in the opposite direction. StackView gives you the option to display the *Position Values* in either direction.

When the *Count From Back* option is selected, the *Position Values* are shown counting from back to face. When the *Count From Face* option is selected, the *Position Values* are shown counting from face to back.

### ***Selections Text String Field***

The *Selections* text string field shows the names of the cards that have been selected in earlier activities. If the *Highlight Selections* checkbox is unchecked, the card names will be replaced with asterisks (\*\*\*) .

Every time a card is selected, its name will appear in the *Selections* field. If the same card is selected more than once, its name will appear more than once. The *Selections* field will retain the order of the cards as they are selected, and duplicate selections will be recorded.

There is no practical limit to the number of cards that can be selected.

### ***View Deck From Beneath/Above Table***

The default view of the deck is from the bottom of the table. StackView allows you to change the view as though you were looking at the deck from the top of the table. You do this with the option buttons provided.

When you select the *View Deck From BENEATH Table (main)* option, you will be looking at the cards as though they were spread face down on a glass table, and you were looking at the cards from beneath the table.

When you select the *View Deck From ABOVE Table* option, you will be looking at the cards as though they were spread face down on the table, and you were looking at the cards from above the table.

For more detail on the *Point of View*, see page 24.

### ***Clear All Selections***

When the *Clear All Selections* button is pressed, all selected cards are returned to their unselected mode. The highlighting is turned off. The *Stack Value* font is no bolded. And the *Selections* text field is cleared.

When the selections are cleared, the order and orientation (reversed or not) of the cards are retained.

### ***Clear Reversed Cards***

When the *Clear Reversed Cards* button is pressed, all reversed cards are turned over to the original orientation. (If the deck is viewed from the bottom of the table, as in the default mode, all the cards will be face up when this button is pressed.)

When the reversed cards are turned back over, the order of the cards, and their selection status will be retained.

### ***Reverse All Selections***

When the *Reverse All Selections* button is pressed, all of the selections will be individually turned over in the deck.

### **Force Selection**

With StackView you may influence the card that is selected and returned to the deck. Once you establish the parameters of your force, you press the *Force* button, and the *Event* is executed.

A card that is forced will become a selected card, and will be highlighted. For more information on controlling the view of selected cards, see page 66.

When a *Force Selection Event* is recorded in a *Session*, it will appear as two *Commands*. The first *Command* is the *Force* portion of the *Event*, and the second *Command* is the *Return* portion of the *Event*. For more information see page 85.

### ***Force Card***

You indicate the *Force Card* by specifying the *Position Value* of the card you want forced. The reason the *Position Value* is used (and not the *Stack Value*) is that you must know where the card is before you can force it.

For example, starting with an Aronson stack, perform two *OutFaro* shuffles, and use StackView to *Force* with a *Position* parameter of 10. The 10<sup>th</sup> physical card

from the top of the deck, the Eight of Clubs, will be forced, even though its *Stack Value* is 16. If you had actually wanted to *Force* the card with the *Stack Value* of 10 (the Ace of Clubs), you would have to use a *Position* parameter of 37 because the Ace of Clubs is the 37<sup>th</sup> card from the top.

### ***Return Card***

Once a card is forced, you can have it returned to the deck randomly or in a controlled location.

#### *Anywhere*

When the *Anywhere* option is selected, the forced card will be returned to a random place in the deck.

#### *Controlled Position*

When the *Controlled Position* option is selected, you must also specify the *Position Value* for the card to be returned to in the provided text box. For example, if you want the forced card returned to the top of the deck, you enter the number 1 in the text box.

### ***Reverse Forced Card***

If the *Reverse Forced Card* is checked, the card will be turned over before it is returned to the deck.

### **Free Choice From Spread**

With StackView you may have a card freely selected and returned to the deck. Once you establish the parameters of your free choice selection, you press the *Free Choice From Spread* button, and the *Event* is executed.

A card that is chosen will become a selected card, and will be highlighted. For more information on controlling the view of selected cards, see page 66.

When a *Free Choice Event* is recorded in a *Session*, it will appear as two *Commands*. The first *Command* is the *Free Choice* portion of the *Event*, and the

second *Command* is the *Return* portion of the *Event*. For more information see page 85.

### **Select Card**

The selected card can be completely random, or partially constrained.

#### *Any Card*

If the *Any Card* option is selected, the free choice will be completely random, and any card in the deck is equally likely to be selected.

#### *Specific Section*

If the *Specific Section* option is selected, the selected card will still be a free choice, but the selection will be constrained to a specific portion of the deck.

You must specify whether the constrained portion of the deck is the *Top Third*, *Middle Third*, or *Bottom Third* from the dropdown list.

### **Return Card**

Once a card is freely selected, you can have it returned to the deck randomly or in a controlled location.

#### *Original Position*

If the *Original Position* option is selected, the freely chosen card will be returned to its exact same position it was in before it was chosen.

#### *Anywhere*

When the *Anywhere* option is selected, the freely chosen card will be returned to a random place in the deck.

#### *Controlled Position*

When the *Controlled Position* option is selected, you must also specify the *Position Value* for the card to be returned to in the provided text box. For example, if you want the freely chosen card returned to the tenth position in the deck, you enter the number 10 in the text box.

### *Specific Section*

If the *Specific Section* option is selected, the return position will still be a random outcome, but the return placement will be constrained to a specific portion of the deck.

You must specify whether the constrained portion of the deck is the *Top Third*, *Middle Third*, or *Bottom Third* from the dropdown list.

### ***Reverse Selected Card***

If the *Reverse Selected Card* is checked, the card will be turned over before it is returned to the deck.

### **Free Choice with Special Handling**

In stacked deck magic, and in particular some of Simon Aronson's material, cards are selected by spectators with special handling. StackView provides some dedicated events to simulate the special handling. These special handlings allow the spectators to select one or more cards in a seemingly fair and free way with no touching of cards by the magician.

You must first select one of the special handlings from the dropdown list. When you select one of the choices, the graphic box will change to a depiction of what the special handling is. The picture shows the "before and after" condition of the deck with each handling. Each of them is described below.

In each of these handlings, you must instruct the spectator to cut specified approximate sizes. For example, you may ask the spectator to cut "approximately a third of the deck."

### ***Cut Select Next***

The *Cut Select Next* special handling has options for one, two, or three card selections. This description will describe the selection of one card in detail.

For the *Cut Select Next 1 Event*, the spectator first randomly cuts about half of the deck. The cut portion is called Block A. Block A is placed on the table. Then next card – the top card of the remainder of the deck – is the selected card. It is placed on top of Block A. The new balance of the deck is called Block B. Block B is placed on top of the selected card, and the deck is squared up. See Figure 23 for the graphical depiction.

To execute the *Cut Select Next 2 Event*, the same sequence is followed, but the random cuts should be approximately a third of the deck, and the cuts are increased to select two cards. See Figure 24 for the graphical depiction.

To execute the *Cut Select Next 3 Event*, the same sequence is followed, but the random cuts should be approximately a quarter of the deck, and the cuts are increased to select three cards. See Figure 25 for the graphical depiction.

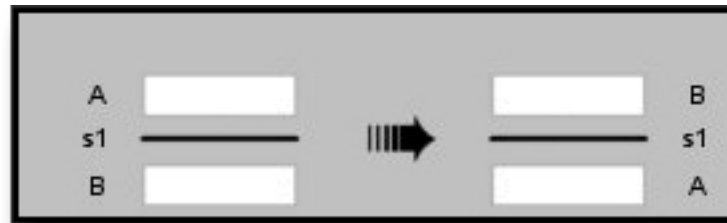


Figure 23



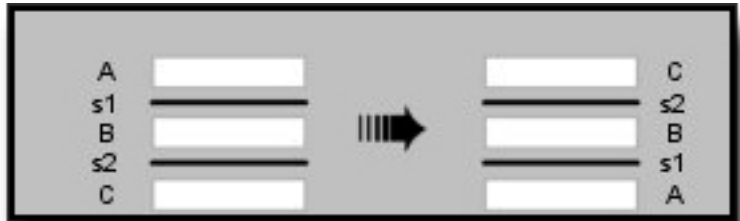


Figure 24

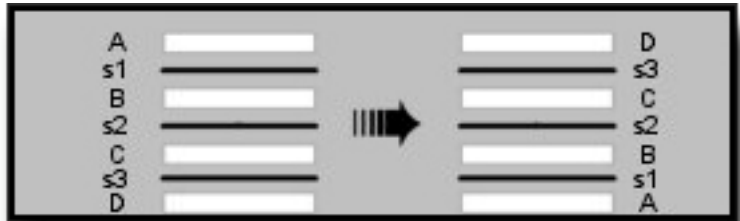


Figure 25

***Cut Select Next Face***

The *Cut Select Next Face* special handling has options for one, two, or three card selections. This description will describe the selection of one card in detail.

For the *Cut Select Next Face 1 Event*, the spectator first randomly cuts about a third of the deck. The cut portion is called block A. Block A is placed on the table. Another cut is made of approximately a third of the deck. This is called Block B. The selected card is the face card (bottom card) of Block B. Block B is placed on top of Block A. The new balance of the deck is called Block C. Block C is placed on top of the stack of Blocks A and B, and the deck is squared up. See Figure 26 for the graphical depiction.

To execute the *Cut Select Next Face 2 Event*, the same sequence is followed, but the random cuts should be approximately a quarter of the deck, and the cuts are increased to select two cards. See Figure 27 for the graphical depiction.

To execute the *Cut Select Next Face 3 Event*, the same sequence is followed, but the random cuts should be approximately a fifth of the deck, and the cuts are increased to select three cards. See Figure 28 for the graphical depiction.

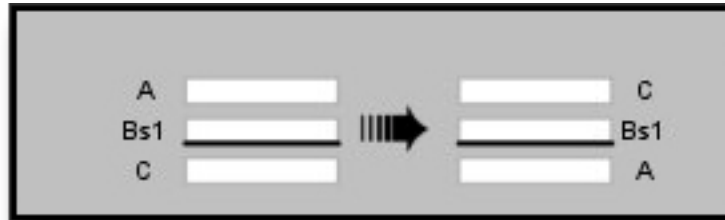


Figure 26

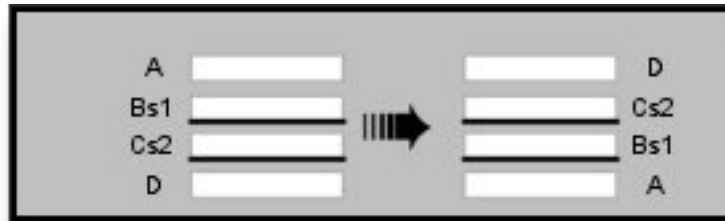


Figure 27

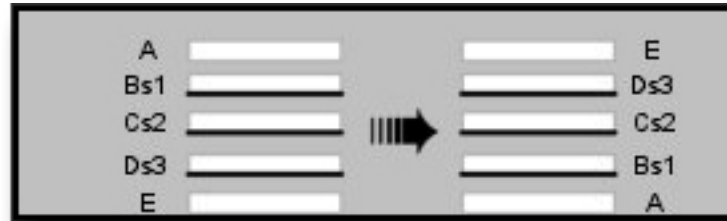


Figure 28

***Cut Select Next Repeat***

The *Cut Select Next Repeat Event* is identical to doing two *Cut Select Next Events* in sequence, but in each case, approximately a third of the deck is cut each time at the start. This description will describe the selection of two cards in detail.

For the *Cut Select Next Repeat Event*, the first spectator first randomly cuts about a third of the deck. The cut portion is called Block A. Block A is placed on the table. Then next card – the top card of the remainder of the deck – is the first selected card. It is placed on top of Block A. The new balance of the deck is placed on top of the selected card, and the deck is squared up. The sequence is repeated. The second spectator randomly cuts about a third of the deck. This cut portion is called Block B. Block B is placed on the table. Then next card – the top card of the remainder of the deck – is the second selected card. It is placed on top of Block B. The new balance of the deck is called Block C. Block C is placed on top of the selected card, and the deck is squared up. See Figure 29 for the graphical depiction.

To execute the *Cut Select Next Repeat 2 Event*, the same sequence is followed, but the random cuts should be approximately a quarter of the deck, and the sequence is increased by one round to select a total of three cards. See Figure 30 for the graphical depiction.

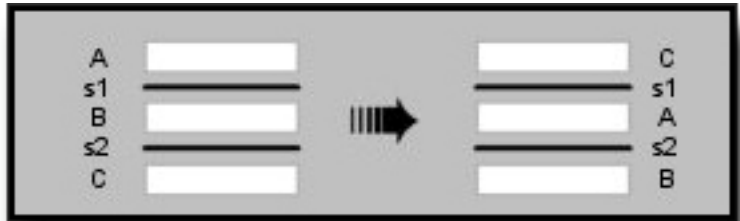


Figure 29

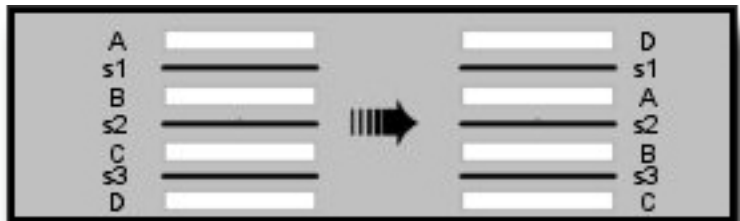


Figure 30

***Reverse Selected Card(s)***

There are three checkboxes (one for each possible special handling selection) that stipulate whether any or all of the selected cards should be reversed in the deck.

By checking one of the checkboxes, that corresponding card will be reversed in the deck during the selection *Event*.

Only the relevant checkboxes will be enabled. For example, if you are selecting a two-card selection special handling, only the first two checkboxes will be enabled.

You may specify any number or combination of cards to reverse.

## **Swap Cards Selection**

The *Swap Cards Event* allows you precise control of swapping the positions of two cards in the deck. You have the option of having the swapped cards be *Selections* or not. Before pressing the *Swap Cards* button, you must select one of the control options and any additional parameter settings.

### ***Random***

When the *Random* option is selected, the positions of two completely random cards are swapped. Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Specified Card Stack Values***

When the *Specified Card Stack Values* option is selected, you must also indicate the *Stack Values* of two different cards. Those two cards' positions will be swapped. Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Specified Card Positions***

When the *Specified Card Positions* option is selected, you must also indicate the *Position Values* of two different cards. The two cards at those positions will be swapped. Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Different Colors***

When the *Different Colors* option is selected, the positions of two completely random cards are swapped. However, one of the cards will be a red card, and the other will be a black card. Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Same Color***

When the *Same Color* option is selected, the positions of two completely random cards are swapped of the same color (red or black). You must also specify

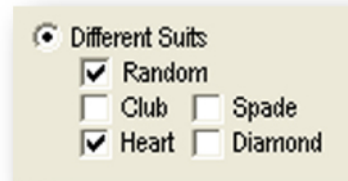
something about the color. If you check the *Random* checkbox, StackView will randomly select the color for you. If you check the *Red* or *Black* checkboxes, the appropriate color will be set for the random swap. Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Different Suits***

When the *Different Suits* option is selected, the positions of two completely random cards are swapped with different suits (clubs, hearts, spades or diamonds). You must also specify something about the suits.

If you check only the *Random* checkbox, StackView will randomly select both suits for you.

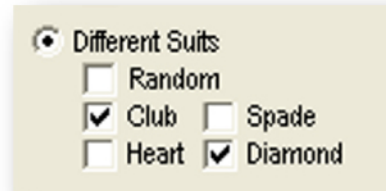
You may also check the *Random* checkbox and any one of the other suits. In that case, one of the cards will be randomly selected among the specified suit cards, and the other card will be from a random other suit. For example, if you select the *Random* checkbox and the *Heart* checkbox as shown in **Error! Reference source not found.**, the two swapped cards will be a randomly selected *Heart* card, and a randomly selected non-*Heart* card.



**Figure 31**

You may also specify two suits for the swapped cards. For example, if you select the *Diamond* and *Club* checkboxes as shown in Figure 32, the two swapped

cards will be a randomly selected *Diamond* card and a randomly selected *Club* card.



**Figure 32**

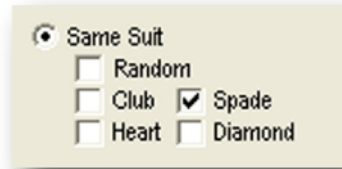
Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***Same Suit***

When the *Same Suit* option is selected, the positions of two completely random cards are swapped with the same suit (clubs, hearts, spades or diamonds). You must also specify something about the suit.

If you check only the *Random* checkbox, StackView will randomly select the suit for you.

You may also check any one of the *Suit* checkboxes. and any one of the other suits. In that case, both cards will be randomly selected among the specified suit. For example, if you select the *Spade* checkbox as shown in Figure 33, the two swapped cards will both be a randomly selected *Spade* card.



**Figure 33**

Depending on the status of the *No Selections* checkbox, the swapped cards may or may not become *Selections*.

### ***No Selections***

The default condition of the *No Selections* checkbox is the unchecked mode. When it is unchecked, the two swapped cards will become *Selections*. When the checkbox is checked, the two cards are swapped, but they do not become *Selections*.

### ***Sessions Tab***

A *Session* is a sequence of *Events* that you have recorded for later playback. This feature helps you practice your routines by simulating them for you and leaving the results for further study. You can play back the recorded *Session Events* one *Event* at a time, or all at once. You can also send your *Session* files to others to play back with their own copy of StackView.

*Session* files that have been recorded with earlier versions of StackView will not work with version 5. You must re-record them for them to work with the latest version. Similarly, *Sessions* recorded with version 5 will not work with earlier versions of StackView. If you are sharing *Session* files with others, make sure that everyone is using the correct version of StackView.



## **Recording Sessions**

Recording a *Session* is similar to creating a macro. It is a sequence of *Events* that can be saved and played back later. Once the *Session* is initiated, all *Recordable Events* that are executed will be listed in the *Session* list box to the left of the control buttons.

### ***Record Button***

The Record button toggles from the "Record" mode to the "Stop" mode. Its label changes to indicate what will happen when you press it next.

When you press the Record button, your subsequent actions in StackView will be recorded as Events in the Sessions list box. When the *Sessions* record mode is turned on, all recordable *Event* buttons change color and have red button faces.

If you are in recording mode, the text above the button will read "Currently Recording." If you are not in recording mode, the text will read "Not Recording."

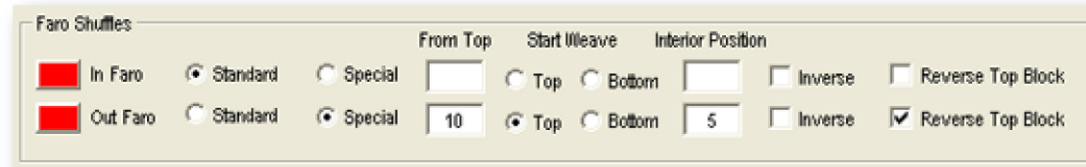
When you have finished recording your *Session*, you should press the same button, now labeled "*Stop*," to prevent additional *Events* from being added to your *Session*.

### ***Recordable Events***

With your *Session* in record mode, all *Recordable Event* buttons change color and have red button faces. When you press one of the *Recordable Events*, the *Event* and its parameters are entered in the *Session* list as *Session Commands*. Newly recorded *Session Commands* are always added to the bottom of the list. You will find more information on rearranging *Session Commands* below. When dealing with *Sessions*, the terms "*Event*" and "*Command*" are interchangeable. However, the term "*Event*" refers to what happens to the cards, and the term "*Command*" refers to what the *Event* is called in the *Session* file. The list in the *Session* list box can be called the *Session Events* list and also the *Session Commands* list.

For example, if you specify an *OutFaro* shuffle with special parameters as shown in Figure 34, you will get a *Session Command* entered in the *Session* list that reads:

```
OutFaroSpecialTopReverse(10, 5)
```



**Figure 34**

This particular *Command* title contains some of the parameter information in its name. The additional data appears as parameters in the parentheses.

Another example is the *Swap Cards Event* as shown in **Error! Reference source not found.** You will get a *Session Command* entered in the *Session* list that reads:

```
SwapDifferentSuits(X, S, 0)
```

Swap Cards

No Selections

Random

Specified Card Stack Values

Specified Card Positions

Different Colors

Same Color

Random

Red

Black

Different Suits

Random

Club  Spade

Heart  Diamond

Same Suit

Random

Club  Spade

Heart  Diamond

Figure 35

In this case, the parameter "X" represents the *Random Suit*, the "S" represents *Spades*, and the "0" represents the unchecked *No Selections* condition.

For complete details on the *Session Command* names and parameters, see *Appendix 3: Session Commands* on page 239.

### **Working with Session Files**

Once you have recorded a *Session*, you may save them for later use. For more details on saving and opening *Sessions*, see the detailed information on page 204.

If there is no active *Session* in StackView's memory, the text above the *Session* list box will read "No current session." If there are *Session Commands* in the *Session* list box, but the *Session* has not been saved (or it has been saved, but new changes exist that have not been saved), the text above the *Session* list box will read "Current Session NOT saved." If the *Session* is newly saved or newly opened from an existing file, the text above the *Session* list box will show the file name.

### **Controlling Sessions**

When you have *Session Commands* in your *Session* list box, you can control them individually or as a group. To control an individual *Session Command*, you must first highlight it by clicking it with your mouse pointer. A highlighted *Session Command* appears as shown in Figure 36.

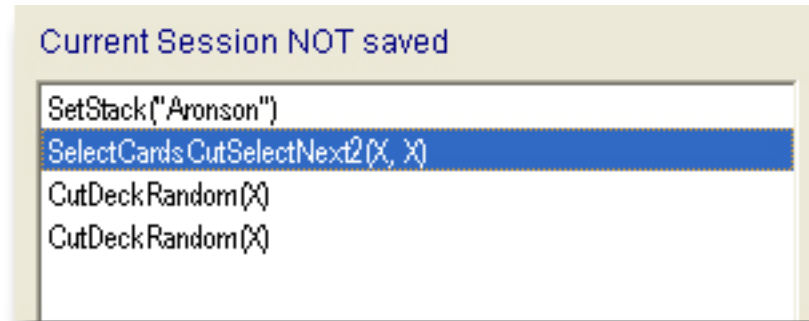


Figure 36

***Play ALL Current Session***

This button will automatically scroll down the entire list of *Session Commands* and perform each *Event*. The *Deck* window will update after all the *Events* are completed. After the whole *Session* has completed playing, the final *Session Command* will remain highlighted.

***Play Current Event***

This button will perform the single *Event* that is selected and highlighted in the *Session Events* list. Since *Select/Return Events* happen in pairs, selecting either the *Select* or the *Return Event* before pressing the *Play Current Event* button will result in the pair of *Commands* getting executed in the correct order as a single *Event*.

After a single *Event* is executed from the *Session* list, the next *Command* is highlighted. If the last *Command* was the one that was executed, it will remain highlighted, but you will get a pop-up dialog box to remind you that you are at the end of the list.

### ***Move Event Up***

This button will move the selected *Event* up one position in the *Session Events* list. (You select an *Event* by clicking on it, and it will become highlighted.) Since *Select/Return Events* happen in pairs, they will move together, and be treated as a single *Event*.

### ***Move Event Down***

This button will move the selected *Event* down one position in the *Session Events* list. (You select an *Event* by clicking on it, and it will become highlighted.) Since *Select/Return Events* happen in pairs, they will move together, and be treated as a single *Event*.

### ***Clear ALL Session Events***

This button will prompt you to clear all of the *Events* in the *Session Events* list. If you answer "Yes" to the question, it will clear all of the *Events*. Be sure to confirm that your *Session* has been saved by checking the text above the *Session* list box. If there is a file name in the text, then the file has been saved. For more information on this text indicator, check on page 84.

### ***Delete Current Event***

This button will delete the highlighted *Event* in the *Session Event* list. (You select an *Event* by clicking on it, and it will become highlighted.) There is no undo option to revert back, and there is no confirmation requested for deleting a single *Event*.

### ***Insert Session File as Macro***

You can have a *Session Command* call another *Session* file as a separate macro sequence. This is useful when you have created a special sequence of *Events* that you would like to use in another routine. When you press this button, you will get a pop-up dialog box requesting a *Session* file to use as a macro. *Session* files have a `.sys` extension. (For more information on *Session* files, see page

205.) To use a *Session* file as a macro, the file must be in the same directory as the StackView program.

As an example, if you want to use the file `MyMacro.svs` as a macro, the *Session Command* will appear in the *Session* list box as `Macro(MyMacro.svs)`.

When the macro event is run, the entire called *Session* file is run at once. Macro files can contain other nested macro files. However, you can only call macro files up to ten levels deep. In other words, if `Macro1.svs` calls a `Macro2.svs` which calls a `Macro3.svs`, etc., you can only go as far as `Macro10.svs`. If you try to nest one more level, you will get an error message.

When a macro call is complete, the next *Command* in the calling *Session* will be executed (unless you are playing back one *Event* at a time).

## Reference: Piles Control

The *Pile Control* window provides versatility of control by letting you perform *Events* on subsets of cards called *Piles*. When *Piles* are created, either randomly or under your control, you have a great deal of control with individual *Piles* or combinations of *Piles*.

The general sequence of actions with *Piles* is that you must first create the *Piles* you are interested in. The *Deck* window will show the cards in their respective *Piles*, and each one will be numbered. You can create up to eight *Piles*. Once the *Piles* are created, you have many *Events* to choose from to perform on individual *Piles* or with pairs of *Piles*.

You specify which *Piles* you want to work with through the *Pile Control Matrix*, which is described on page 102.

You access *Piles Control* from the Tools menu on the menu bar. When the *Piles Control* menu selection is checked, the *Piles Control* window is visible. When the *Piles Control* menu selection is unchecked, the window is hidden. When the *Piles Control* window is closed, the *Piles Control* menu item will become unchecked.

### **Create Piles**

Broadly speaking, there are two ways to create *Piles*: by cutting the cards, or dealing the cards. Each way of creating *Piles* can be done with precise control, or varying degrees of randomness.

You must first identify how many *Piles* you want created by entering a single digit from 1 to 8 in the *Create Piles* field as shown in Figure 37. Before you can press the *Create Piles* button, you must specify how you want the *Piles* created.



Figure 37



## Cut Cards

When *Piles* are created by cutting the cards, the cards are cut from the top (back) of the deck in sequence. For example, if you want four *Piles* created by randomly cutting the cards, the result you will get is four *Piles* that are each approximately a quarter of the deck. *Pile 1* is the first quarter cut, *Pile 2* the second, *Pile 3* the third, and *Pile 4* the balance of the deck. The *Piles* will display in the *Deck* window as shown in Figure 38. The *Piles* shown in Figure 38 were created from the Aronson Stack in its starting order. You can see that *Pile 1* came from the first quarter cut of the deck, and so on.

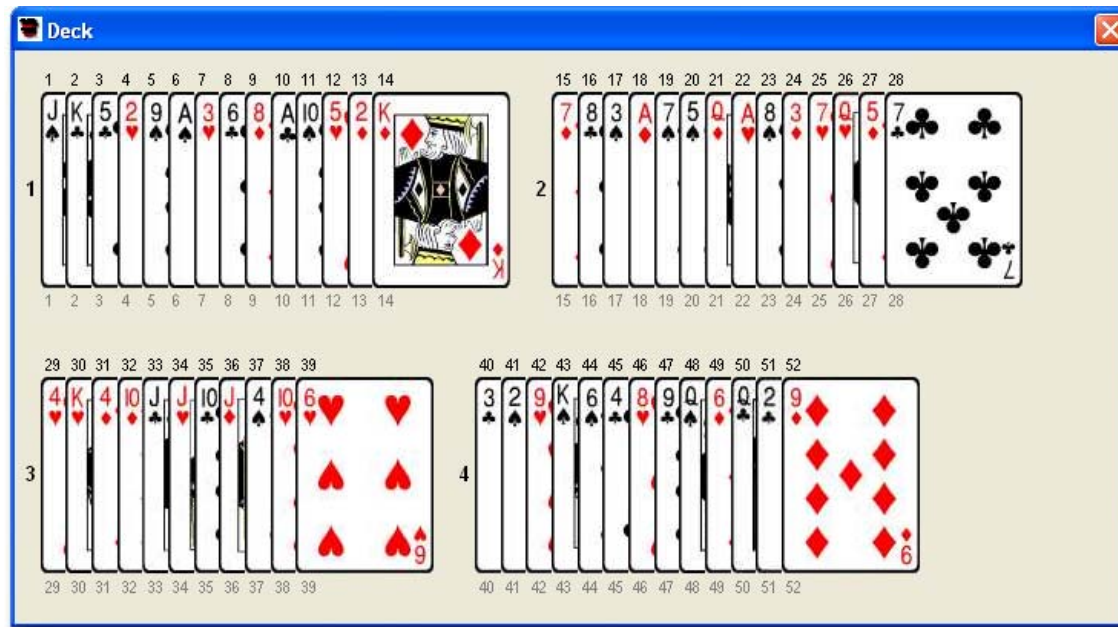


Figure 38

### ***Pile Specifications***

You have complete control of how the cards will be cut through the *Pile Specifications* frame as shown in Figure 39. Each *Pile* specification row has specific controls. The *Random* option will result in a *Pile* with a random number of cards. The *Specified* option will result in a precise number of cards to be cut for that *Pile*. The text field is used to indicate the number of cards to cut for the *Pile*. The single letter at the right of the text field is either an "A" for "approximate" or "E" for "exact." (You can not edit the single letter – it is there as an indicator.)

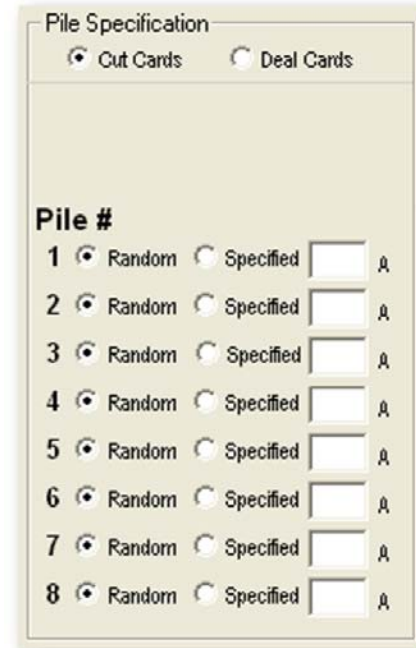
To cut *Piles*, you must select the *Cut Cards* option in the *Pile Specifications* frame.

If all of the specifications are set to *Random*, and all the text fields are empty, StackView will cut the *Piles* so that they are of random, but approximately equal size.

If a *Pile* specification is *Random*, and you specify a number in the text field, that *Pile* will be cut randomly, but the number of cards will approximate the number you entered in the text field.

If a *Pile* specification is *Specified*, the *Pile* will be cut precisely to the number you enter in the text field.

If all of the *Pile* specifications are *Specified*, then the numbers in the text fields must add up to 52.



**Figure 39**

You can use any combination of *Random* and *Specified Pile* settings, with any numbers entered in the text fields. As long as one of the *Piles* has a *Random* setting, the *Piles* will be scaled logically so that the entire deck is used.

For example, the settings shown in Figure 40 call for 5 *Piles*. The first *Pile* is *Specified* to 10 cards. The second *Pile* is *Specified* to 2 cards. The third *Pile* is set to *Random* with an approximate number set to 40 cards. The fourth and fifth *Piles* are also *Random*, but both of their approximate numbers are set to 20 cards. Clearly, the numbers do not add up to 52 cards.

However, since at least one of the *Piles* is *Random*, StackView will logically adjust the *Piles* while preserving the implied *Pile* ratios by scaling the random numbers. Starting with the Aronson Stack, these specifications result in the five *Piles* shown in Figure 41.

(This is a random outcome, and you may not get the exact same result from the same specifications.) Note that *Pile* 1 has exactly 10 cards, and *Pile* 2 has exactly 2 cards – just as specified. Since *Piles* 3 through 5 had 40, 20, and 20 cards specified respectfully, StackView scaled the remaining 40 cards (12 were already consumed in the first two *Piles*) while preserving the requested ratios. In this particular random outcome, as shown in Figure 41, *Pile* 3 got 20 cards, *Pile* 4 got 10 cards, and *Pile* 5 got 10 cards.

Pile #	Random	Specified	Value	Label
1	<input type="radio"/>	<input checked="" type="radio"/>	10	E
2	<input type="radio"/>	<input checked="" type="radio"/>	2	E
3	<input checked="" type="radio"/>	<input type="radio"/>	40	A
4	<input checked="" type="radio"/>	<input type="radio"/>	20	A
5	<input checked="" type="radio"/>	<input type="radio"/>	20	A
6	<input checked="" type="radio"/>	<input type="radio"/>		A
7	<input checked="" type="radio"/>	<input type="radio"/>		A
8	<input checked="" type="radio"/>	<input type="radio"/>		A

Figure 40

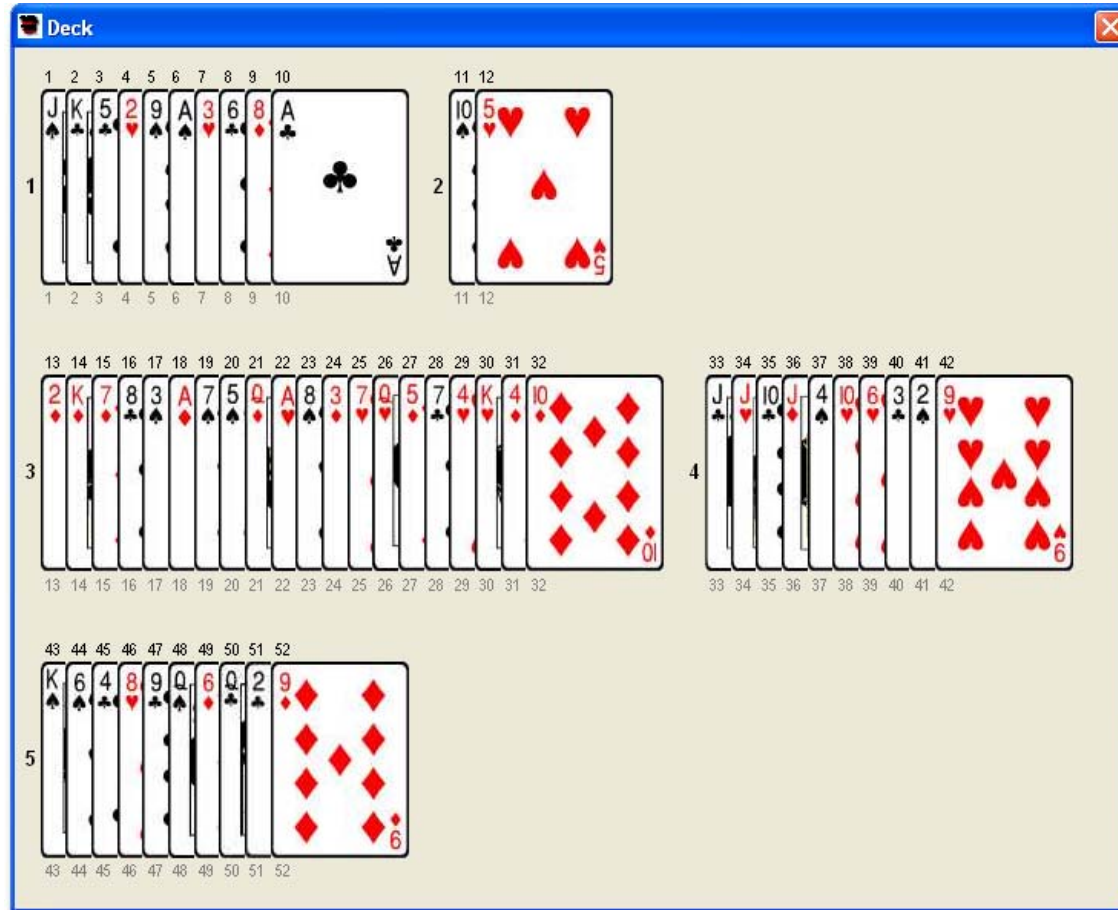


Figure 41

Also note the *Stack Values* of the cards in each *Pile*. You can see that the *Piles* have been cut to the table one *Pile* at a time.

When you specify the number of *Piles* you want in the *Create Piles* text field, the *Pile Specifications* frame will disable the rows of the *Piles* that are not relevant. (Note that Piles 6 through 8 in Figure 40 are not enabled.) The enabled/disabled rows will get updated when the *Create Piles* text field has been changed, and you have shifted focus away from it by clicking on any other control. To reset the *Piles Specification* frame to enable all of the rows, press the *Refresh Deck* button in the lower left corner of the *Piles Control* window. See more information about the *Refresh Deck* button on page 131.

### **Deal Cards**

When *Piles* are created by dealing the cards, the cards are dealt one at a time from the top of the deck onto *Piles* on the table as specified in the *Pile Specification* frame. There are three ways in which the *Piles* can be dealt. The first way is that the *Piles* are dealt so that each *Pile* is completed before the next one is started. The second way is alternating the deal so that each *Pile* gets one card at a time, in the same manner as a poker deal would occur. The third way is that the cards are randomly dealt in no particular order, with one or more cards being dealt onto a *Pile* at a time. As with cutting the *Piles*, you have specific control of how the dealing will occur with the *Pile Specifications*.

### ***Complete Specified Piles***

With *Complete Specified Piles*, the cards are dealt singly onto each *Pile* before the next *Pile* is started. You have complete control of how the cards will be dealt through the *Pile Specifications* frame as shown in Figure 42. Each *Pile* specification row has specific controls. The *Random* option will result in a *Pile* with a random number of cards. The *Specified* option will result in a precise number of cards to be dealt for that *Pile*. The text field is used to indicate the number of cards to deal for the *Pile*. The single letter at the right of the text field is either an "A" for "approximate" or "E" for "exact." (You can not edit the single letter – it is there as an indicator.)

To deal complete *Piles*, you must select the *Deal Cards* option in the *Pile Specifications* frame. You must also select the *Complete Specified Piles* option.

If all of the specifications are set to *Random*, and all the text fields are empty, StackView will deal the *Piles* so that they are of random, but approximately equal size.

If a *Pile* specification is *Random*, and you specify a number in the text field, that *Pile* will be dealt randomly, but the number of cards will approximate the number you entered in the text field.

If a *Pile* specification is *Specified*, the *Pile* will be dealt precisely the number of cards you entered in the text field.

If all of the *Pile* specifications are *Specified*, then the numbers in the text fields must add up to 52.

You can use any combination of *Random* and *Specified Pile* settings, with any numbers entered in the text fields. As long as one of the *Piles* has a *Random* setting, the *Piles* will be scaled logically so that the entire deck is used.

For example, the settings shown in Figure 43 call for 4 *Piles*. The first *Pile* is *Specified* to 10 cards. The second *Pile* is *Specified* to 2 cards. The third *Pile* is set to *Random* with an approximate number set to 40 cards. The fourth *Pile* is

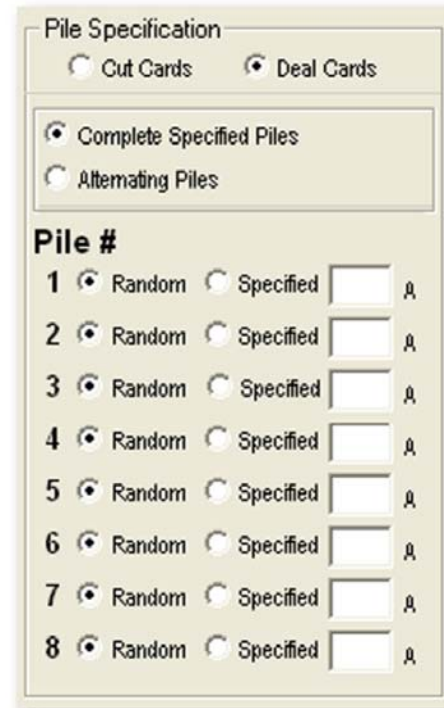


Figure 42

also *Random*, but its approximate number is set to 10 cards. Clearly, the numbers do not add up to 52 cards. However, since at least one of the *Piles* is *Random*, StackView will logically adjust the *Piles* while preserving the implied *Pile* ratios by scaling the random numbers. Starting with the Aronson Stack, these specifications result in the four *Piles* shown in Figure 44. (This is a random outcome, and you may not get the exact same result from the same specifications.) Note that *Pile* 1 has exactly 10 cards, and *Pile* 2 has exactly 2 cards – just as specified. Since *Pile* 3 and *Pile* 4 had 40 and 10 cards specified respectively, StackView scaled the remaining 40 cards (12 were already consumed in the first two *Piles*) while preserving the requested ratios. In this particular random outcome, as shown in Figure 44, *Pile* 3 got 31 cards, and *Pile* 4 got 9 cards.

Also note the *Stack Values* of the cards in each *Pile*. You can see that the *Piles* have been dealt to the table one card at a time until each *Pile* is complete.

When you specify the number of *Piles* you want in the *Create Piles* text field, the *Pile Specifications* frame will disable the rows of the *Piles* that are not relevant. (Note that Piles 5 through 8 in Figure 43 are not enabled.) The enabled/disabled rows will get updated when the *Create Piles* text field has been changed, and you have shifted focus away from it by clicking on any other control. To reset the *Piles Specification* frame to enable all of the rows, press the *Refresh Deck*

Figure 43



button in the lower left corner of the *Piles Control* window. See more information about the *Refresh Deck* button on page 131.

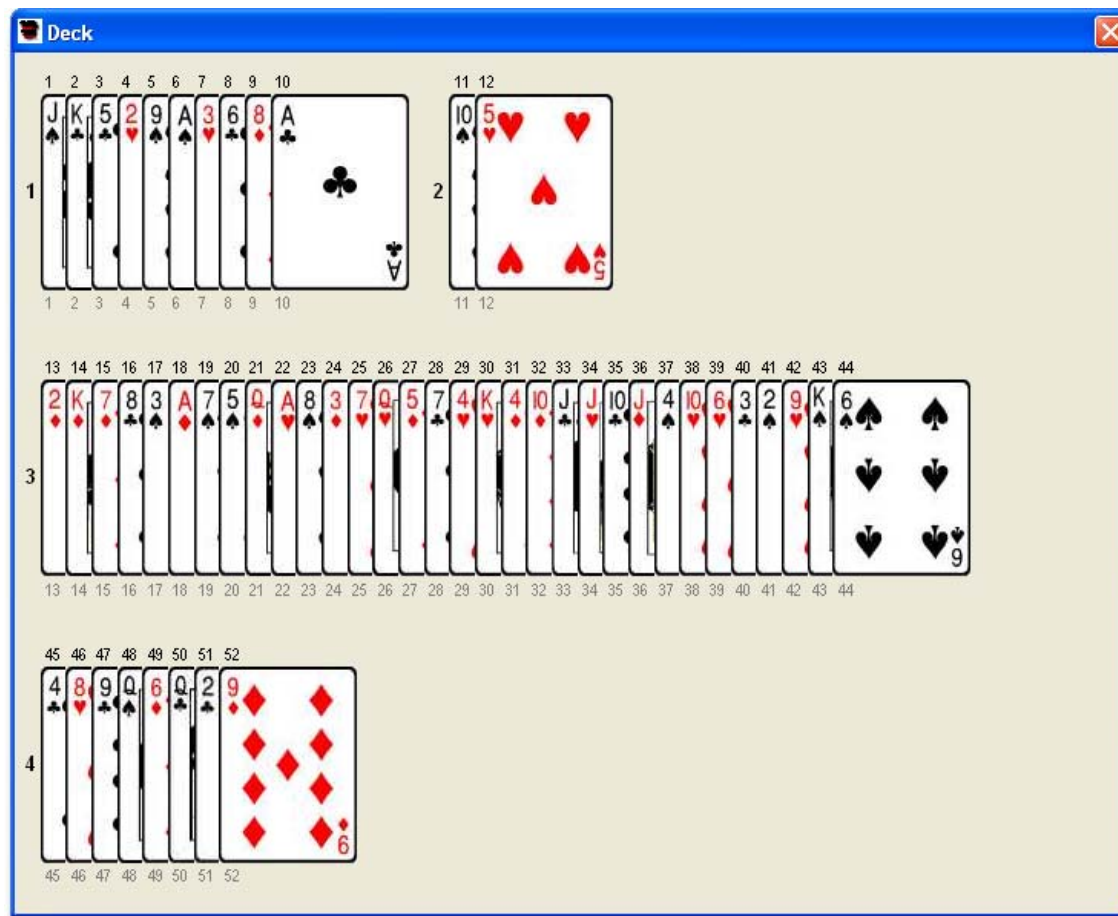


Figure 44



### ***Alternating Piles***

When you specify *Deal Cards*, one of the further options you have is to select *Alternating Piles* as shown in Figure 45. When you deal *Alternating Piles*, the cards are dealt singly to each *Pile* that you specify, in sequence.

#### *Number of Cards to Deal*

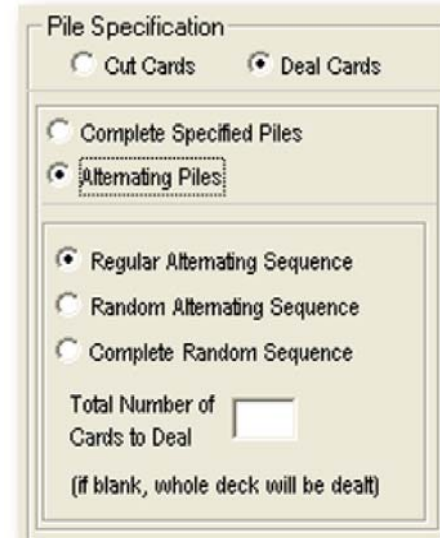
You may specify the number of cards to deal in the text field provided. If you leave the text field blank, all 52 of the cards will be dealt into the number of *Piles* you specified in the *Create Piles* field. If you specify the number of cards to be dealt, StackView will deal that many cards into the correct number of *Piles* as specified, and the balance of the deck will become an additional *Pile*. (This final *Pile* is not dealt – it is

just placed on the table.) For this reason, if you specify the number of cards as less than 52, you may only request up to seven *Piles*, since StackView needs the eighth *Pile* for the balance of the deck.

You may not specify fewer cards to be dealt than then number of requested *Piles*; each *Pile* needs at least one card.

#### *Regular Alternating Sequence*

When you select the *Regular Alternating Sequence*, the cards will be dealt just as they would in a poker deal. Cards are dealt one at a time, and one per *Pile*, starting at the first *Pile*. Once every *Pile* gets a card, the dealing returns back to



**Figure 45**

the first *Pile*, and continues one card per *Pile*. This continues until the specified number of cards are all dealt.

#### *Random Alternating Sequence*

The *Random Alternating Sequence* is similar to the *Regular Alternating Sequence* with the exception that the order of the *Piles* that the cards are dealt to is random for each round of cards. For each round of dealing, each *Pile* gets one card, but the sequence will be different each time.

For example, Figure 46 shows the outcome of a *Random Alternating Sequence* deal of all the cards into four *Piles*. The starting deck order was the Aronson Stack. You will note that the set of first card from the face of each *Pile* contains the first four *Stack Values*. The set of second cards from the face of each *Pile* contains *Stack Values* 5 through 8. However, the order of the deal was different for each round. As you study the *Piles* further, you will see that the order of *Piles* dealt in each round is random.

#### *Complete Random Sequence*

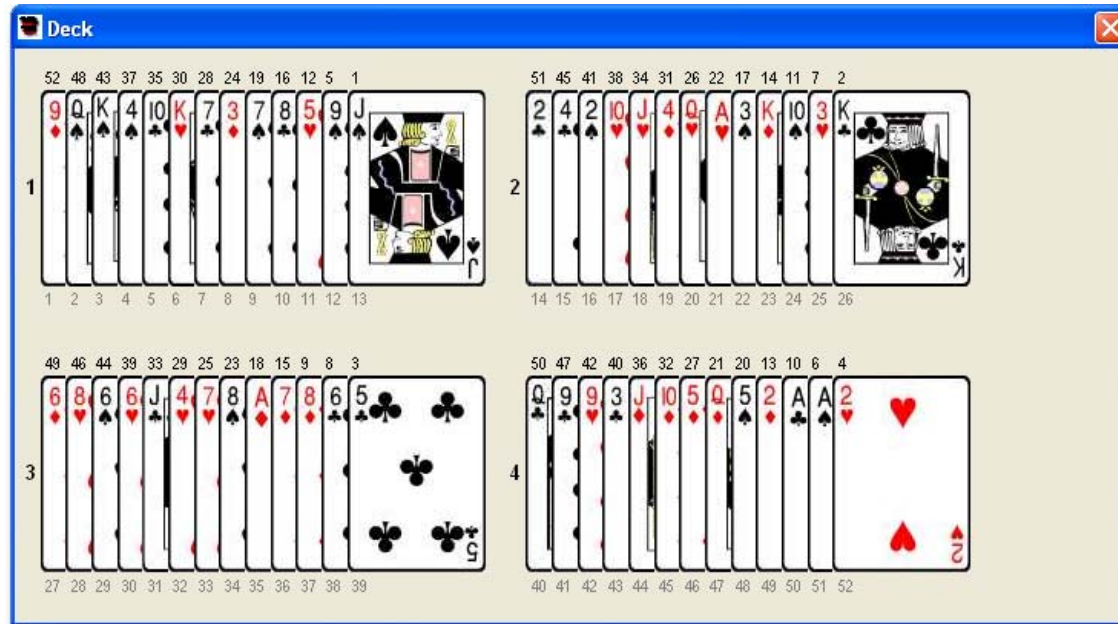
The *Complete Random Sequence* is similar to the *Random Alternating Sequence* except that it is not limited to one card per *Pile* for each round. In this deal, one or more cards are placed in each *Pile* until all the specified cards are exhausted.

There is one constraint with this deal. For the first round of cards dealt, one card will go to each *Pile* in a random sequence. After the first round, one or more cards will be dealt to the *Piles* randomly. The purpose for this constraint is to ensure that each *Pile* will get at least one card.

#### ***Piles Viewpoint***

Imagine spreading a deck of cards face down on a glass table surface. Now imagine going under the table and looking up at the face of the cards through the glass. This is the point of view that StackView shows you in the *Deck*

window. There are two places in StackView where you can change your point of view to look at the cards from above the table.



**Figure 46**

The first place is on the *Selection* tab in the *StackView Control* window, where you have the option to select your view from either beneath or above the table. This option selection operates in the default *Deck* view. Also, when you choose one of these option selections, the *Deck* window will be forced to the default view regardless of what the current view was. In other words, if you are in a *Piles* view, and you change the viewpoint from the *Selection* tab in the *StackView Control* window, you will lose your *Pile* display; it will default back to the standard *Deck* view.

The second place in on the *Piles Control* window. Here, you also have the option to select your view from beneath or above the table. However, in this case, these option selections will only affect the *Piles* view. When you are in the *Piles* viewing mode, you must use the *Piles Viewpoint* options on the *Piles Control* window to change the point of view.

It is possible to confuse yourself if the *Point of View* settings are different in the two locations. When you switch between regular *Deck* view and *Piles* view, the cards would appear to turn over without your intervention.

Figure 47 shows the Aronson Stack in the *Piles* view (after being cut to four random piles) when viewing from beneath the table. Note that the Jack of Spades is the left most face up card in the first *Pile*. When you change views to viewing from above, as in Figure 48, you will note that the Jack of Spades is now the right-most face down card in the first *Pile*. This is exactly as you would see it if you actually did look above and beneath first *Pile*.

When you compare the opposite sides of a *Piles* view as in Figure 47 and Figure 48, you will note that each individual *Pile* is correct. However, the relative positions of the *Piles* do not move. This is intentional to help you more readily keep track of which *Pile* you are looking at.

Also note that in each case, the *Stack Values* and *Position Values* are correctly placed in each view.

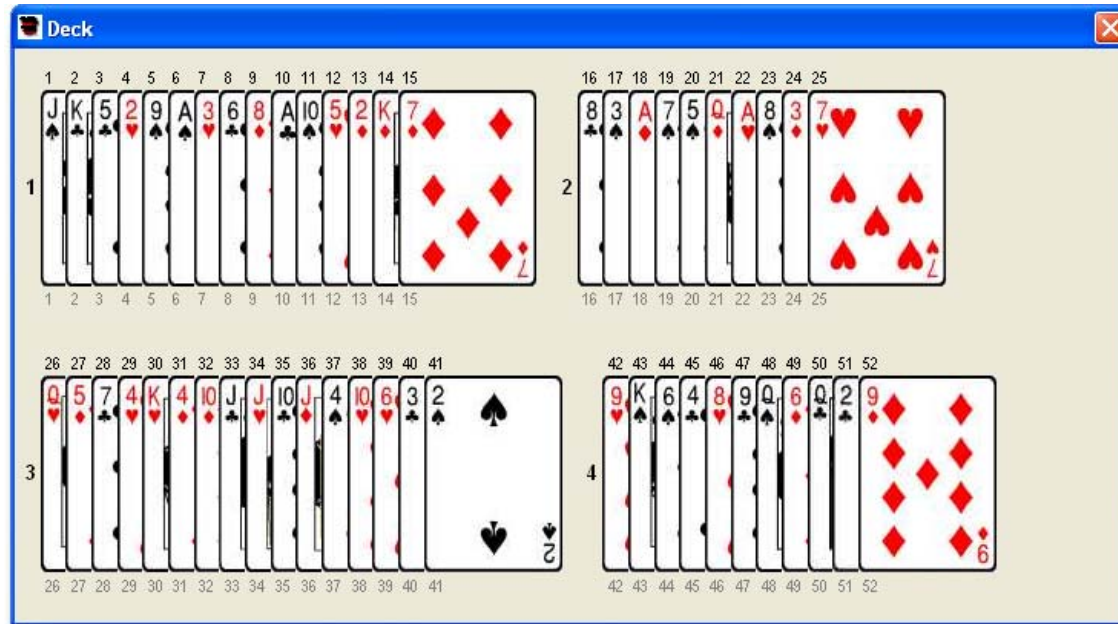


Figure 47

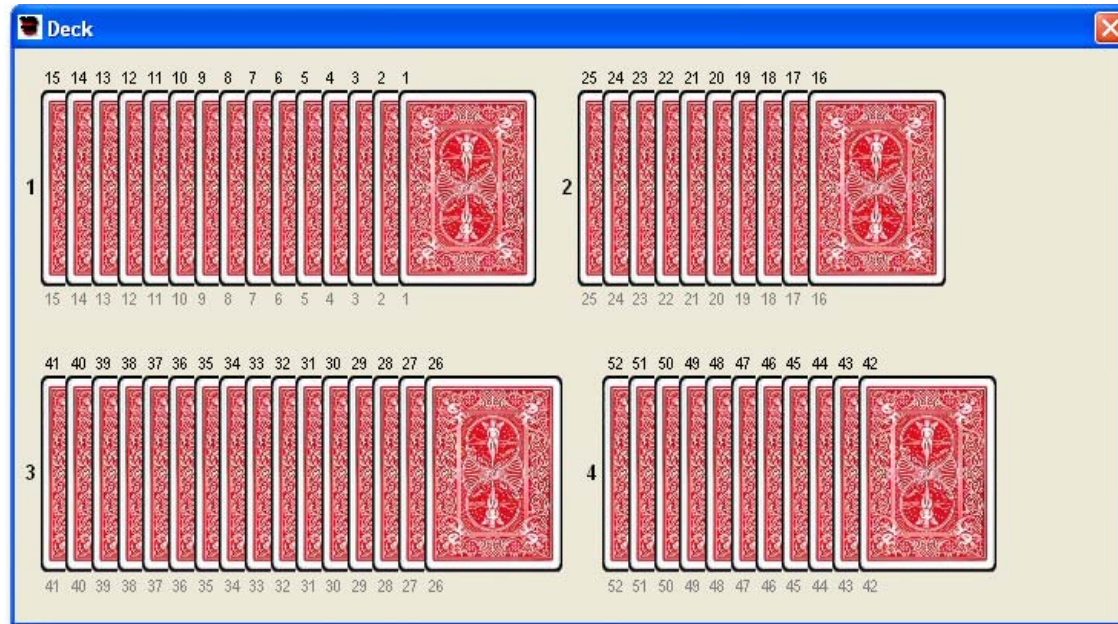


Figure 48

### ***Pile Control Matrix***

The *Pile Control Matrix* may look intimidating at first. Once you understand its functionality, it will become second nature to use. Each of the seven *Pile Events* use the *Matrix* to specify which *Piles* the *Event* will operate on.

The heart of the *Matrix* is an eight by eight grid of checkboxes. When you create a number of *Piles*, the *Matrix* will disable a portion of the *Matrix*, since it can not be used. For example, before you create any *Piles*, the *Matrix* will appear as shown in Figure 49. As an example, after you create five *Piles*, the *Matrix* will appear as shown in Figure 50. Essentially, the *Matrix* will resize itself based on the number of *Piles* that are present.

Figure 49

the *Secondary Pile*.

All *Pile Events* require a *Primary Pile* to be identified. Some *Pile Events* may or may not require a *Secondary Pile* to be identified. Some *Pile Events* require that both the *Primary* and *Secondary* are identified. When only the *Primary Pile* is used by an *Event*, the check can be made in any box in the desired row of the *Matrix*.

Some *Pile Events* allow for both the *Primary* and *Secondary Piles* to be the same, while others do not. If it is not allowed, you will get an appropriate error message.

Each *Row* of the *Matrix* represents the *Primary Pile* that an *Event* will use, and the *Columns* of the *Matrix* represent the *Secondary Pile* for the *Event*.

You may place a check in any one of the active *Matrix* boxes. If you check another box, the first one will be cleared automatically. Figure 51 shows a check that indicates the 2<sup>nd</sup> *Pile* as the

*Primary Pile*, and the 4<sup>th</sup> *Pile* as

Figure 50



		Secondary							
Rev		1	2	3	4	5	6	7	8
Primary	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 51

it will be *Reversed* before the *Event* executes. The *Secondary Pile* is *Pile 5*, and it will not be *Reversed* during the *Event*.

When a *Pile Event* allows use of the *Reverse* checkboxes in the *Piles Control Matrix*, there is a blue bold letter "R" in the frame of the *Event*. Figure 53 shows the letter "R" in the frame of the *Combine Event*.

There are two additional sets of checkboxes as part of the *Matrix*. There is a column of boxes to the left of the *Matrix*, and a row of boxes above the *Matrix*. These boxes are used to identify which *Pile* should be *Reversed* (turned over) before the event is executed. These additional boxes are used by only two of the *Pile Events* (*Combine* and *Riffle Shuffle*).

The *Matrix* example in Figure 52 indicates that there are six *Piles*. The

*Primary Pile* is *Pile 3*, and

		Secondary							
Rev		1	2	3	4	5	6	7	8
Primary	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 52



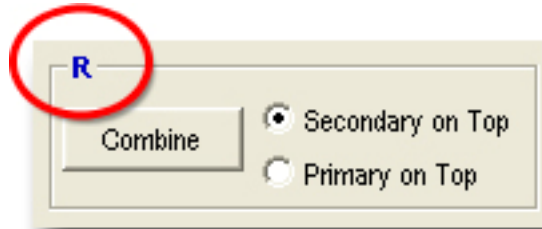


Figure 53

### ***Select / Return***

The *Select/Return Event* allows you to have a card selected (random or forced) from a *Pile* (specified or random), and then returned to a *Pile* (same or different, specified or random). You have the option to have the selected card take on the StackView *Selected* status, and also the option to *Reverse* the selected card.

### **Selection Pile**

The *Selection Pile* is the *Pile* from which the card will be selected. You have two options to choose from.

#### ***Primary***

You select this option if you want to specify the *Selection Pile*. You must indicate the *Primary Pile* in the *Pile Control Matrix* by checking a box in the appropriate row.

#### ***Random***

If you want the *Selection Pile* to be a *Random* choice, you select this option. If there is a check in the *Pile Control Matrix*, the *Primary* row information will be ignored.

### **Selected Card**

There are four options to specify which card from the *Selection Pile* will be the *Selected Card*.

#### ***Top***

This option results in the *Top* card of the *Selection Pile* becoming the *Selected Card*.

#### ***Bottom***

This option results in the *Bottom* card of the *Selection Pile* becoming the *Selected Card*.

#### ***Random***

This option results in a *Random* card from the *Selection Pile* being chosen as the *Selected Card*.

#### ***Specified***

For this option, you must specify the position of the card you want selected from the *Selection Pile*. The card positions are counted from the top of the *Selection Pile*. You must enter the position number of the card you want selected in the text field box provided. If you enter a number that is greater than the number of cards in the *Pile*, you will get an error message, and the *Event* will be blocked.

### **Return Pile**

The *Return Pile* is the *Pile* to which the card will be returned. You have six options to choose from.

#### ***Secondary***

You select this option if you want to specify the *Return Pile*. You must indicate the *Secondary Pile* in the *Pile Control Matrix* by checking a box in the appropriate column.

***Primary/Same***

You select this option when you want the *Return Pile* to be the same as the *Selection Pile* (regardless of how the *Selection Pile* was specified). For this option, the *Secondary Pile* information in the *Pile Control Matrix* is ignored.

***Random Not Same***

You select this option when you want the *Return Pile* to be randomly selected, but different than the original *Selection Pile* (regardless of how the *Selection Pile* was specified). For this option, the *Secondary Pile* information in the *Pile Control Matrix* is ignored.

***Random Any***

If you want the *Return Pile* to be a *Random* choice (including the original *Selection Pile* as a possibility), you select this option. If there is a check in the *Pile Control Matrix*, the *Secondary* row information will be ignored.

***New Pile Random***

This option has the *Selected Card* being placed as a *New Pile* (single card) somewhere among the existing *Piles*. The *New Pile* will take on a random *Pile* number. Since StackView can only handle a maximum of eight *Piles*, this option requires that you are starting with at most seven *Piles*, since it creates an additional *Pile*.

***New Pile Specified***

This option has the *Selected Card* being placed as a *New Pile* (single card). The *New Pile* will be assigned the specified *Pile* number, which is entered in the text field box provided. If the number entered in the field is greater than the number of *Piles* that will exist after the *Event*, the *Pile* number will be assigned to the next value. Since StackView can only handle a maximum of eight *Piles*, this option requires that you are starting with at most seven *Piles*, since it creates an additional *Pile*.

## **Return Position**

You may specify where in the *Return Pile* the *Selected Card* will be placed. If the *Return Pile* is a *New Pile*, the *Return Position* options will be ignored since there is only one card in the *New Pile*.

### ***Top***

This option results in the *Selected Card* being returned as the *Top* card of the *Return Pile*.

### ***Bottom***

This option results in the *Selected Card* being returned as the *Bottom* card of the *Return Pile*.

### ***Same***

This option results in the *Selected Card* being returned to the *Same* position in the *Return Pile* as it was in the *Selection Pile*. This option could potentially generate an error if the *Same* position in the *Return Pile* does not exist.

### ***Random***

This option results in the *Selected Card* being returned to a *Random* position in the *Return Pile*.

### ***Specified***

For this option, you must specify the position the *Selected Card* will occupy in the *Return Pile*. The card positions are counted from the top of the *Return Pile*. You must enter the position number in the text field box provided. If you enter a number that is greater than the number of cards in the *Pile*, you will get an error message, and the *Event* will be blocked.

## **Reverse Selected Card**

If this box is checked, the *Selected Card* will be *Reversed* during the *Event*.

### **Move Only**

If this box is checked, the *Selected Card* will only be moved and will not take on the StackView *Selected* card status. (In other words, this option does not record the card as a *Selected Card* for highlighting, and it will not appear in the *Selected Cards* text string on the *Selection* tab in the *StackView Control* window.)

### **Combine**

The *Combine Event* simulates picking up one *Pile* and putting it on top of another *Pile* to combine them. The destination *Pile* is always defined by the *Primary Pile*.

You must specify the *Primary* and *Secondary Piles* with the *Pile Control Matrix*. You may also specify reversing either or both of the *Piles*. For more information on using the *Pile Control Matrix*, see page 102.

### **Secondary on Top**

When you specify *Secondary on Top*, the *Secondary Pile* will be placed on top of the *Primary Pile*.

### **Primary on Top**

When you specify *Primary on Top*, the *Secondary Pile* will be placed beneath the *Primary Pile*.

### **Riffle Shuffle / Gilbreath Principle**

The *Riffle Shuffle Event* simulates riffle shuffling two Piles together into a combined Pile. The destination *Pile* is always defined by the *Primary Pile*. You may protect a portion (top or bottom) of one of the *Piles*. You also have an option to view the results of your shuffle in a special *Gilbreath* mode to explore the Gilbreath Principle.

You must specify the *Primary* and *Secondary Piles* with the *Pile Control Matrix*. You may also specify reversing either or both of the *Piles*. For more information on using the *Pile Control Matrix*, see page 102.

## **Random**

When you select the *Random* option, the two specified *Piles* are riffle shuffled together in a random manner. The destination *Pile* is always defined by the *Primary Pile*.

## **Protect Block**

While *Riffle Shuffling*, you may protect a specified block of cards from either *Pile* at either the *Top* or the *Bottom* of the deck. The protected block of cards will remain intact and will not get shuffled.

You must select the *Protect Block* option to enable the rest of the parameters.

### ***Block***

#### *Primary*

When you specify the *Primary* option, the protected block will be in the *Primary Pile*.

#### *Secondary*

When you specify the *Secondary* option, the protected block will be in the *Secondary Pile*.

### ***Portion***

#### *Top*

When you specify the *Top* option, the protected block will be the *Top* portion of the specified *Pile*.

You must also indicate the number of cards to protect in the text field provided. You may not enter a number larger than the number of cards in the specified *Pile*.

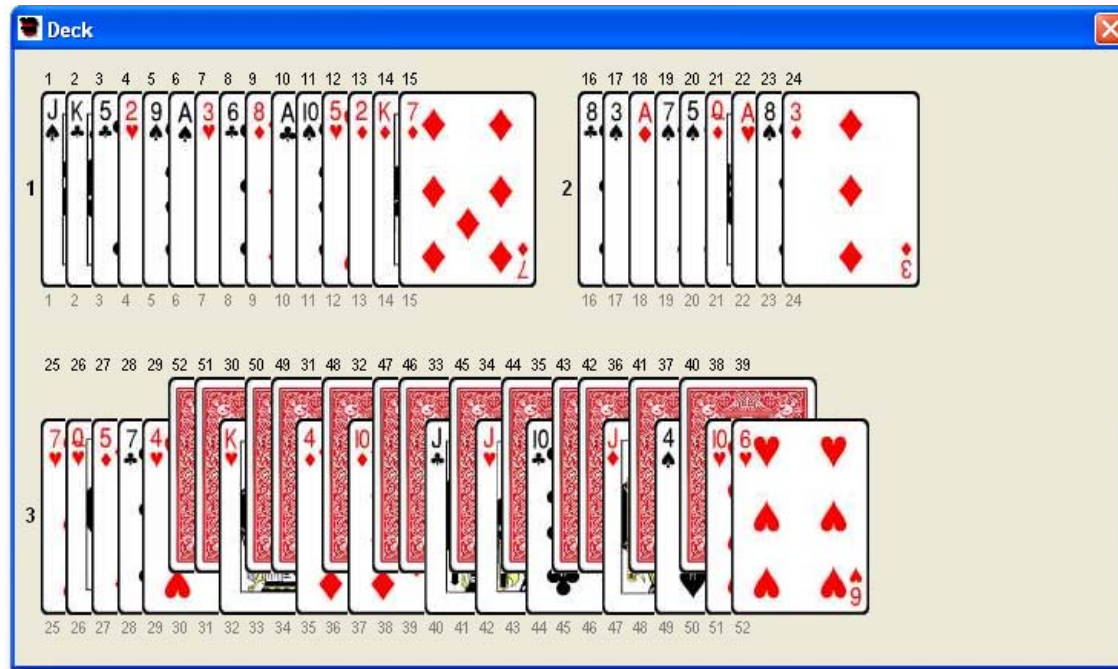
#### *Bottom*

When you specify the *Bottom* option, the protected block will be the *Bottom* portion of the specified *Pile*.

You must also indicate the number of cards to protect in the text field provided. You may not enter a number larger than the number of cards in the specified *Pile*.

### **Gilbreath View**

If you select the *Gilbreath View* option by placing a check in the provided checkbox, the *Deck* window will show the results of the *Riffle Shuffle* in a special way. The shuffled *Piles* will show their new order, but the *Piles* will not have been pushed flush. One of the *Piles* will be upjogged in the display. Figure 54 shows the results of a Gilbreath View when two piles are shuffled with one of the piles reversed.



**Figure 54**

In the *Gilbreath View*, the upjogged cards are always from the *Secondary Pile*.

### ***Cut Piles***

The *Cut Piles Event* simulates cutting one of the *Piles*, or cutting a portion of a *Pile* and placing it onto another existing or new *Pile*. The *Primary Pile* is the one that gets cut, and the *Secondary Pile* is where the cut portion gets placed. You may also specify reversing the cut portion of the *Primary Pile*.

You must specify the *Primary* and *Secondary Piles* with the *Pile Control Matrix*. For more information on using the *Pile Control Matrix*, see page 102.



## **Cut Pile and Portion**

### ***Cut Pile***

#### *Primary Pile*

Select this option when you want to specify which *Pile* will initiate the *Cut*.

#### *Random Pile*

Select this option when you want StackView to randomly select which *Pile* will initiate the *Cut*.

### ***Cut Portion***

#### *Cut Random*

Select this option when you want the *Cut Portion* to be a random number of cards.

#### *Complete Pile*

Select this option when you want the *Cut Portion* to be the entire *Pile* of cards.

#### *Cut Specified*

Select this option when you want the *Cut Portion* to be a specified number of cards. You must also enter the number of cards in the text field provided. The number you enter must not be more than the number of cards in the *Pile*. If you selected a *Random Cut Pile*, and specify a large number of cards, you may get an error message if there are not enough cards in the randomly selected *Pile*.

## **Place Cut Portion**

### ***Complete Cut Primary/Same***

You select this option when you want the *Primary Pile* to get a *Complete Cut* (regardless of how the *Secondary Pile* was specified). For this option, the *Secondary Pile* information in the *Pile Control Matrix* is ignored.

### ***Top of Secondary***

Select this option if you want to place the *Cut Portion* onto the specified *Secondary Pile*. For this option, the *Pile Control Matrix* must be used.

***Top of Same***

Select this option if you want to place the *Cut Portion* back onto the same *Pile*. This is useful if you also use the *Reverse Cut Portion* option described below.

***Top Random Any***

Selecting this option will have the *Cut Portion* placed on any of the existing *Piles*. There is a possibility that the cards may be replaced on the originating *Pile*.

***Top Random Not Same***

Selecting this option will have the *Cut Portion* placed on any of the existing *Piles*, except for the originating *Pile*.

***New Pile Random***

When you select this option, the *Cut Portion* will be used to create a new *Pile* in a random position. Normally for this option, you need to have seven *Piles* to begin with since the eighth *Pile* will be used for the *Cut Portion*. However, if the entire originating *Pile* is cut (either specified or randomly), then a new eighth *Pile* can be created.

***New Pile Specified***

When you select this option, the *Cut Portion* will be used to create a new *Pile* in the specified *Pile* position. Normally for this option, you need to have seven *Piles* to begin with since the eighth *Pile* will be used for the *Cut Portion*. However, if the entire originating *Pile* is cut (either specified or randomly), then a new eighth *Pile* can be created.

**Reverse Cut Portion**

When this option is selected, the *Cut Portion* is turned over (reversed) before it is put in its destination position.

### ***Australian Deal***

The *Australian Deal*, also known as the *Down/Under Deal* has traditionally stipulated one card for both the *Down* and the *Under* card movements. StackView provides significantly more control for this handling. You may now incorporate specified *Downs* and *Unders*, as well as random values for each. You can control whether the cards are reversed as they are handled in a specified or a random way. You can also have one of the cards become a selected card. Finally, you can perform the *Inverse* function on your specifications to assist you in creating new routines with this handling.

You must specify the *Pile* you want to work with by placing a check in the appropriate row in the *Pile Control Matrix*. The *Secondary Pile* information will be ignored. For more information on using the *Pile Control Matrix*, see page 102.

The *Pile* you are using will always be held in the orientation that is represented by the *Point of View* being above the table. For example, in Figure 55, the *Point of View* is the standard *Beneath the Table* view. The top card of the *Pile* is the Jack of Spades. The *Australian Deal* will start with the *Pile* face down in your left hand, and the first card handled will be the Jack of Spades. For more information on *Point of View*, see page 24.

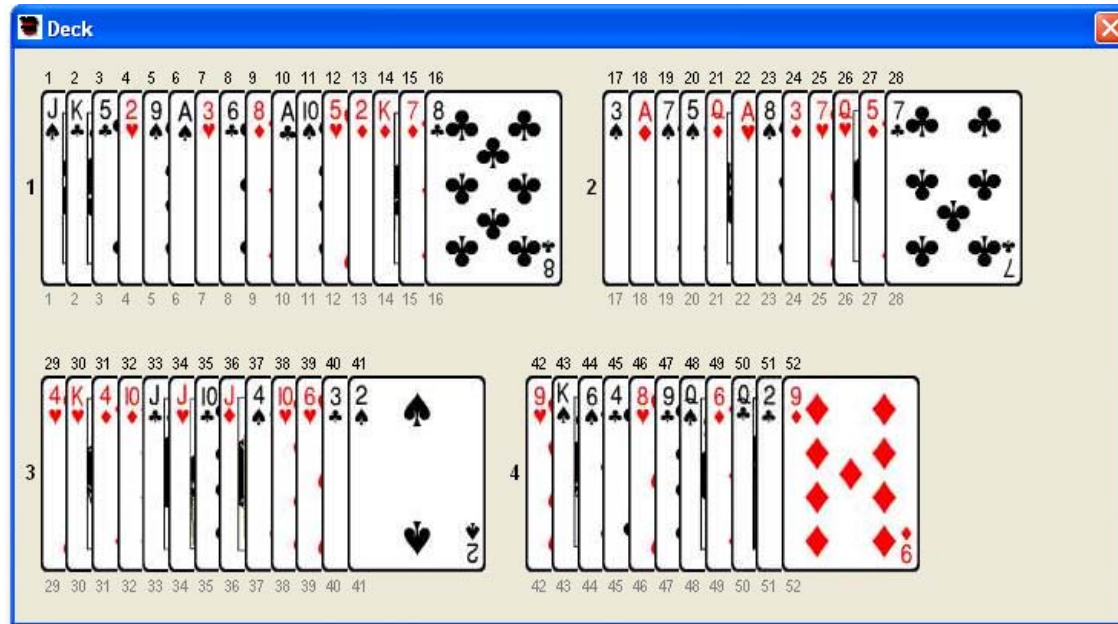


Figure 55

### Select

You may specify one of the *Down* cards to become the elected card by checking one of the checkboxes.

### *Final Card*

When you specify the *Final Card* option, the final *Down* card will be the selected card.

### *Random Down Card*

When you specify the *Random Down Card* option, a random *Down* card will be the selected card.

**Start**

You may specify the starting condition of the *Australian Deal* by selecting one of the *Start* options.

***Down / Under***

When the *Down/Under* option is specified, the first card will be placed *Down* on the table.

***Under / Down***

When the *Under/Down* option is specified, the first card will be placed *Under* the cards in the remaining packet.

**Number**

You may specify how many cards are handled for each of the *Downs* and *Unders* in the deal. When there are more *Downs* to deal than there are cards left in your hand, the deal will finish when the last card is placed on the table. When there are more *Unders* to deal than there are cards left in your hand, the *Unders* will continue looping until the *Under* count is done.

***Standard***

When you select the *Standard* option, exactly one card will be dealt for each *Down* and *Under*. This is the standard handling for the *Australian Deal*.

***Specified***

When you select the *Specified* option, you must also enter values for the number of cards for each *Down* and *Under* in the text fields provided.

You may enter any combination of *Exact* and *Random* settings, but you may not have both an *Exact* and *Random* value set for the same *Down* or the same *Under*.

If you have selected the *Specified* option and have left both the *Exact* and *Random* fields empty for the *Down* or *Under* handling, it will default to an "*Exact 1*" setting for the respective *Down* or *Under*.

### Exact

When you place a value in either of the two *Exact* fields, that is how many cards will be dealt for the respective *Down* or *Under*.

For example, if you enter a 2 and a 1 in the *Exact* fields as shown in Figure 56, the deal will always place 2 cards *Down* followed by 1 card *Under*.

### Random

When you place a value in either of the two *Random* fields, that is the upper limit how many random cards will be dealt for the respective *Down* or *Under*. Each time a *Random Down* is encountered within the same deal, a new random number is calculated. Likewise, each random *Under* within the same deal will have a new random number within the limit specified.

For example, if you enter a 3 and a 2 in the *Random* fields as shown in Figure 57, the deal will place 1, 2, or 3 cards *Down* followed by 1 or 2 cards *Under*. Each successive *Down* will generate a new random number up to a value of 3. Each successive *Under* will generate a new random number up to a value of 2.

### Reverse

You have the option to control how any of the cards may be *Reversed* while the *Australian Deal* is being performed.

	Down	Under
Exact	2	1
Random		

Figure 56

	Down	Under
Exact		
Random	3	2

Figure 57

### ***No Reverse***

When you select the *No Reverse* option, none of the cards will be turned over during the deal.

### ***Reverse Selected***

When you select the *Reverse Selected* option, the selected card will be reversed when it is dealt *Down*. For this option, you must also specify whether the *Final* or *Random Down* card will become the selected card as described on page 116.

### ***Reverse Cards***

When you select the *Reverse Cards* option, you have the opportunity to specify how the *Down* and *Under* cards are reversed, by filling in the appropriate checkboxes. Either *All* the *Down* and/or *Under* cards are reversed, or they are *Randomly* reversed. You may specify any combination of *All* and *Random* allowed.

#### ***All***

When a check is placed in the *All* checkbox beneath the *Down* label, all of the *Down* cards will be reversed. When a check is placed in the *All* checkbox beneath the *Under* label, all of the *Under* cards will be reversed.

#### ***Random***

When a check is placed in the *Random* checkbox beneath the *Down* label, the *Down* cards will be reversed randomly. When a check is placed in the *Random* checkbox beneath the *Under* label, the *Under* cards will be reversed randomly.

### **Inverse**

The *Inverse* option performs the requested sequence in reverse. This is helpful when designing sequences that employ the *Australian Deal*.

For example, Figure 58 shows *Pile 1* with the first ten cards of the Aronson Stack. When the *Australian Deal* specifications are executed as shown in Figure 59, the new *Pile 1* order is shown in Figure 60. With that *Pile* sequence, adding the

*Inverse* check as shown in Figure 61 will undo the first deal and get you back to the *Pile* as shown in Figure 58.

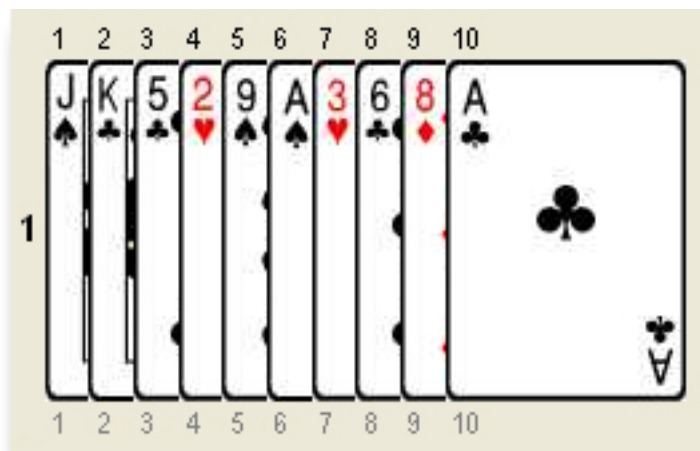


Figure 58



Start		Number		Reverse	
Australian Deal	<input checked="" type="radio"/> Down / Under <input type="radio"/> Under / Down	<input type="radio"/> Standard <input checked="" type="radio"/> Specified			
<input checked="" type="checkbox"/> Select Final Card <input type="checkbox"/> Select Random "Down" Card <input type="checkbox"/> Inverse	Down Under Exact <input type="text" value="2"/> <input type="text" value="1"/> Random <input type="text"/> <input type="text"/>		<input type="radio"/> No Reverse <input type="radio"/> Reverse Selected <input checked="" type="radio"/> Reverse cards	Down Under All <input type="checkbox"/> <input checked="" type="checkbox"/> Random <input type="checkbox"/> <input type="checkbox"/>	

Figure 59

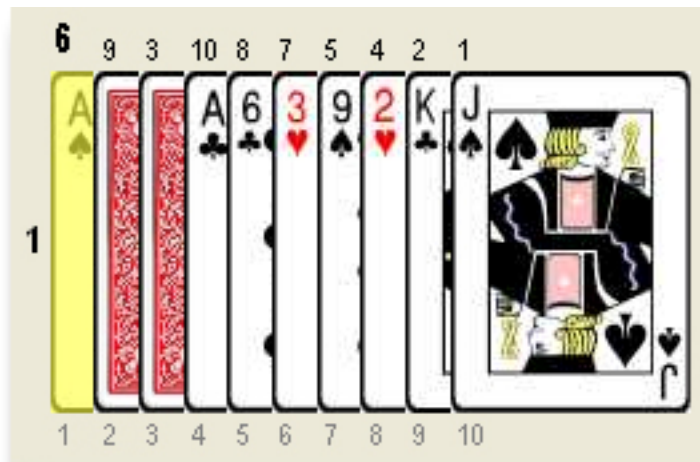


Figure 60

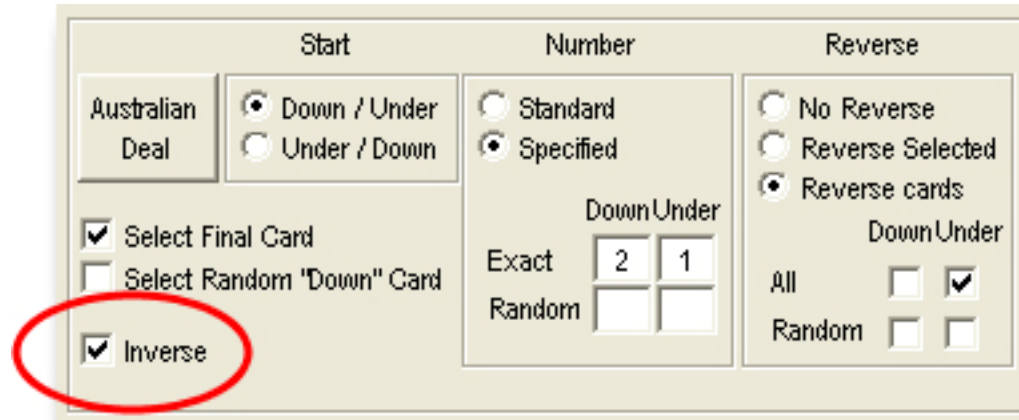


Figure 61

### ***Swap Piles***

The *Swap Piles Event* allows you to switch the positions of two of the *Piles* with some specifications.

You must specify the *Piles* you want to work with by placing a check in the appropriate row and column in the *Pile Control Matrix*. For more information on using the *Pile Control Matrix*, see page 102.

The two *Piles* that will be switched are called the *First* and *Second Piles*. You specify the nature of the *Pile* selection by choosing the appropriate options for each *Pile*.

#### **First Pile**

##### ***Primary***

When you select the *Primary* option, the *First Pile* will be the one specified in the row information of the *Pile Control Matrix*.

### ***Random***

When you select the *Random* option, the *First Pile* will be chosen randomly.

### ***Includes Selected Card***

When you chose the *Includes Selected Card* option, the *First Pile* will include a selected card. If no *Pile* has a selected card, you will get an error message. If more than one of the *Piles* has a selected card, one of those *Piles* will be chosen randomly.

### ***Random with no Selected Card***

When you chose the *Random with no Selected Card* option, the *First Pile* will not include a selected card. If there are no *Piles* without a selected card, you will get an error message. If more than one of the *Piles* is without a selected card, one of those *Piles* will be chosen randomly.

## **Second Pile**

### ***Secondary***

When you select the *Secondary* option, the *Second Pile* will be the one specified in the column information of the *Pile Control Matrix*.

### ***Random***

When you select the *Random* option, the *Second Pile* will be chosen randomly.

### ***Includes Selected Card***

When you chose the *Includes Selected Card* option, the *Second Pile* will include a selected card, if such a *Pile* is available.

If no available *Pile* has a selected card, you will get an error message. If more than one of the *Piles* has a selected card, one of those *Piles* will be chosen randomly.

The *Swap Piles Event* establishes the *Piles* in order; the *First Pile* is established before the *Second Pile*. Because of this, there is the possibility of a logical error

condition for the *Second Pile*. For example, the *Swap Piles* specifications as shown in Figure 62 may generate a logical error. The *First Pile* has a *Random* setting, and the *Second Pile* has an *Includes Selected Card* setting. Note in Figure 63 that only *Pile 3* has selected cards. Since the *First Pile* is selected randomly, it is possible that it will be assigned *Pile 3*. In that situation, since there are no other *Piles* with selected cards, an error message will appear as shown in Figure 64.

The image shows a dialog box titled "Swap Piles". On the left side, there are two unchecked checkboxes: "Reverse First Pile" and "Reverse Second Pile". In the center, there is a section for "First Pile" with four radio button options: "Primary", "Random" (which is selected), "Includes Selected Card", and "Random with no Selected Card". On the right side, there is a section for "Second Pile" with four radio button options: "Secondary", "Random", "Includes Selected Card" (which is selected), and "Random with no Selected Card".

Figure 62

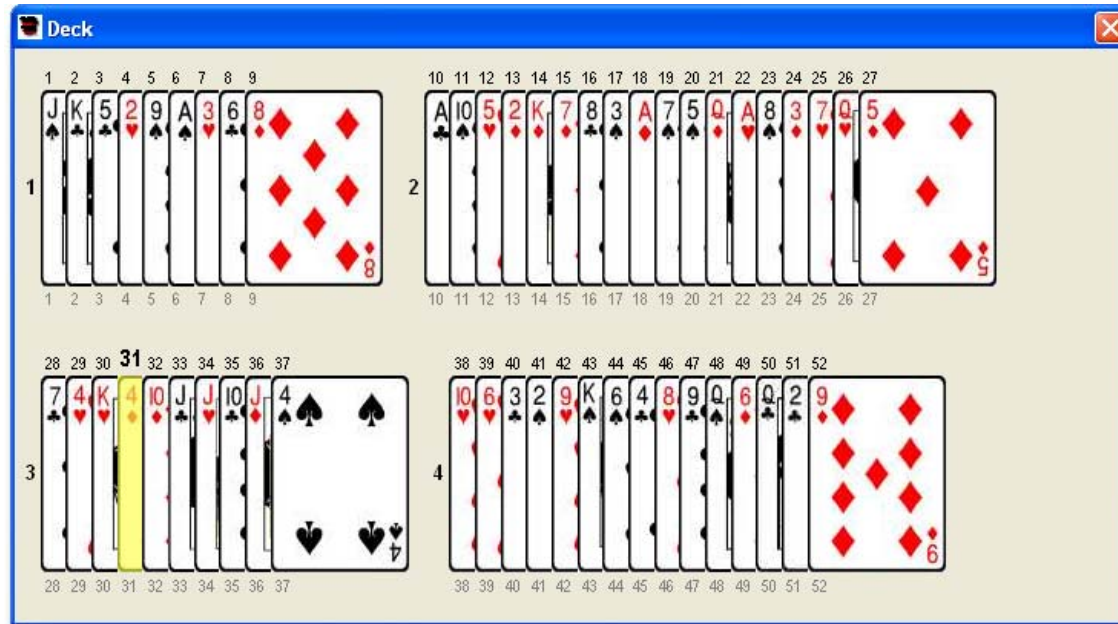


Figure 63



Figure 64

***Random with no Selected Card***

When you chose the *Random with no Selected Card* option, the *Second Pile* will not include a selected card.

If there are no *Piles* without a selected card, you will get an error message. If more than one of the *Piles* is without a selected card, one of those *Piles* will be chosen randomly.

The *Swap Piles Event* establishes the *Piles* in order; the *First Pile* is established before the *Second Pile*. Because of this, there is the possibility of a logical error condition for the *Second Pile*. For example, the *Swap Piles* specifications as shown in Figure 65 may generate a logical error. The *First Pile* has a *Random* setting, and the *Second Pile* has an *Random with no Selected Card* setting. Note in Figure 66 that only *Pile 2* has no selected cards. Since the *First Pile* is selected randomly, it is possible that it will be assigned *Pile 2*. Since there are no other *Piles* without any selected cards, an error message will appear as shown in Figure 67.

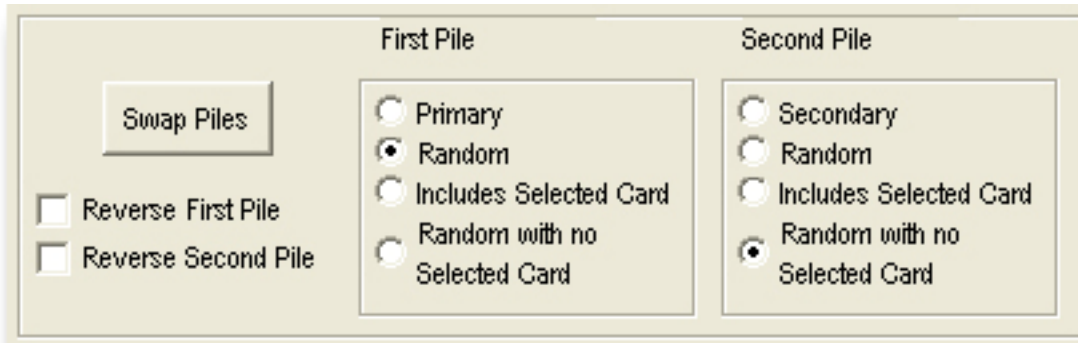


Figure 65

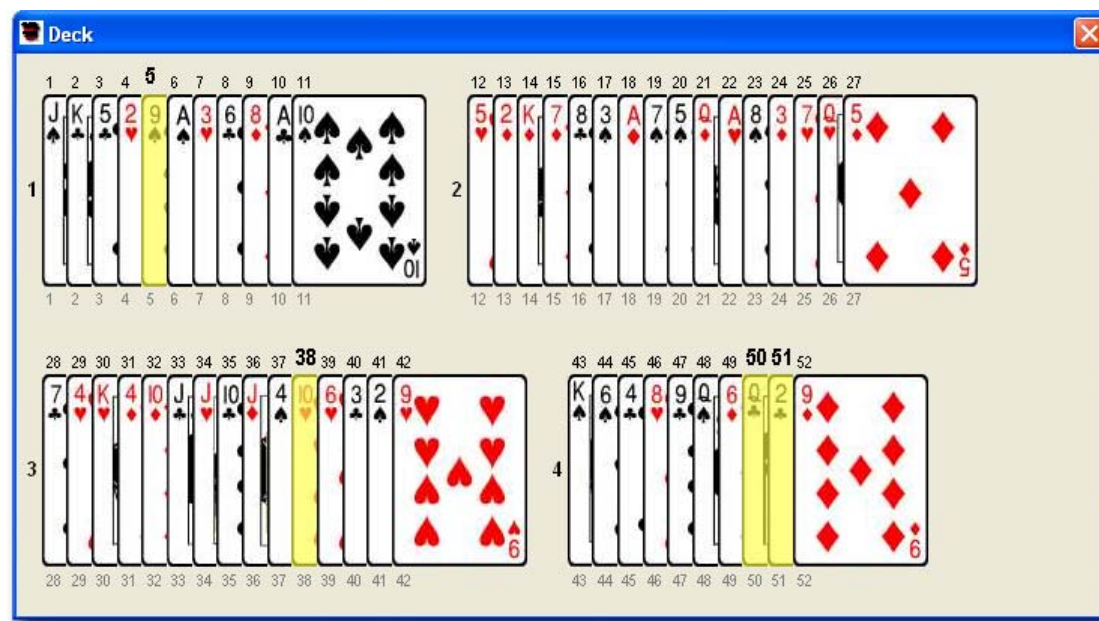


Figure 66



Figure 67

### **Reverse First Pile**

When you select the *Reverse First Pile* option, the *First Pile* will be turned over before it is swapped with the *Second Pile*.

### **Reverse Second Pile**

When you select the *Reverse Second Pile* option, the *Second Pile* will be turned over before it is swapped with the *First Pile*.

### ***Special***

The Special section in the Piles Events allows you to do a few special handlings to a single Pile.

You must specify the *Pile* you want to work with by placing a check in the appropriate row in the *Pile Control Matrix*. The *Secondary Pile* information will be ignored. For more information on using the *Pile Control Matrix*, see page 102.



## Elmsley Count

An *Elmsley Count* will be performed on the specified *Pile*. The *Elmsley Count* will be performed with the *Pile* held in the *Above the Table Point of View*. (For more information on *Piles Viewpoint*, see page 98.) For example, Figure 68 shows the *Above the Table* view of the four Aces just before the first *Elmsley Count* in Dai Vernon's famous *Twisting the Aces* routine. After the *Elmsley Count*, the *Above the Table* view of the four Aces is shown in Figure 69.



Figure 68



Figure 69

### Jordan Count

A *Jordan Count* will be performed on the specified *Pile*. The *Jordan Count* will be performed with the *Pile* held in the *Above the Table Point of View*. (For more information on *Piles Viewpoint*, see page 98.) For example, Figure 70 shows the *Above the Table* view of the Ace through Four of Hearts with the Four of Hears reversed. After the *Jordan Count*, the *Above the Table* view of the four cards is shown in Figure 71.



Figure 70

### Turn Over Pile

When you select the *Turn Over Pile* option, the specified *Pile* will be turned over and returned in its starting position.

### Inverse

The *Inverse* option performs the requested sequence in reverse. This is helpful when designing sequences that employ the *Elmsley* or *Jordan Counts*. (If you use the *Inverse* option with the *Turn Over Pile Event*, you will undo the turning over action, so there will be no visible change.)



Figure 71

### ***Refresh Deck***

When you press the *Refresh Deck* button, the *Piles* will be reassembled in order, from the top to the bottom of the deck. The *Deck* window will return to the standard view.

## Reference: Custom Deck

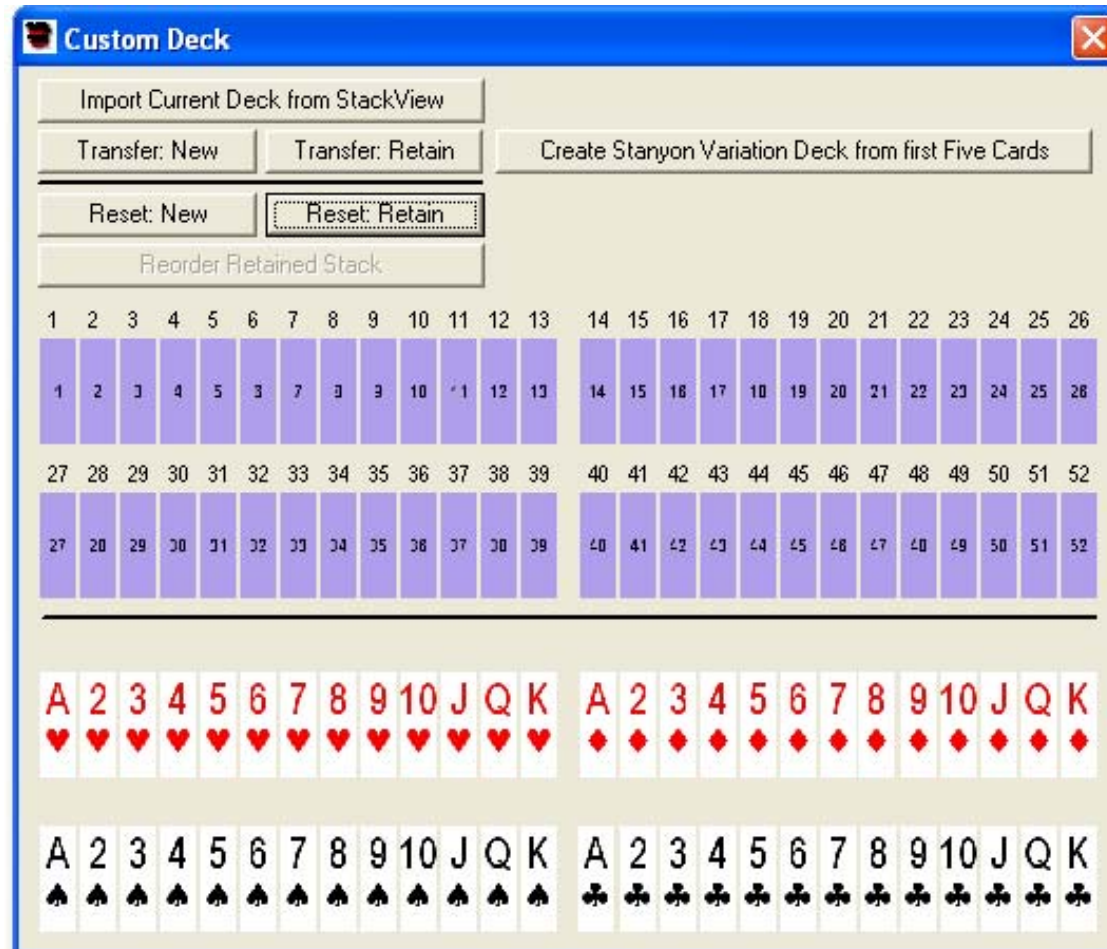
StackView has a *Custom Deck* module that will give you much greater control of creating and setting up stacks. Whether you use a stack that is not already provided in StackView's built in set, or you want to rearrange your existing stack in a special order, you will find the manual process in the *Custom Deck* module useful in your explorations. In addition to the manual process, there is an option to automatically create *Stanyon Variation* stacks by specifying the first five cards of the stack.

### ***User Interface***

You access the *Custom Deck* module from the *Tools* menu. When the *Custom Deck* menu selection is checked, the *Custom Deck* window is visible. When the *Custom Deck* menu selection is unchecked, the window is hidden. When the *Custom Deck* window is closed, the *Custom Deck* menu item will become unchecked. The initial card layout is shown in Figure 72. The *Custom Deck* layout has two sections separated by a long horizontal dividing line. Above the line is your custom deck design area. The *Stack Positions* are represented by open position slots. The slots are numbered from 1 to 52, and the numbers correspond to the positions from the top to the bottom of the face down deck. Below the dividing line is the set of card indices in their home position. The cards that are below the dividing line are those that have not been placed into a stack position yet. You can move the card indices around on the screen to build and edit your stack.

It is important to note that the *Custom Deck* you are building does not affect the official stack in the *Deck* window until you finalize your stack by transferring it.

A card index can be in one of two places: 1) below the dividing line in its own home position, or 2) above the dividing line in one of the stack position slots.



**Figure 72**

You move the cards around by dragging-and-dropping them. To do this, you place your mouse pointer over the card index you want to move. Click-and-hold

the mouse button down. You may now move the card index around. (Actually, you will see the card index outline move.) When you have positioned the card index where you want it, release the mouse button, and the card index will drop into that place. However, the dragging will require some specific knowledge of how StackView handles your actions

1. StackView identifies the target drop area by the exact tip (point) of the mouse's arrow icon. Depending on the drop area, you will get different behavior. If the tip of the arrow slightly misses your desired target, you may get unexpected behavior.
2. If you try to drop a card index onto another existing card index, the attempt will be ignored, and there will be no change.
3. If you try to drop a card index onto an open card slot, it will move to that location.
4. If you try to drop a card index onto any gray background space, the card index will return to its original home position below the dividing line. This can happen unexpectedly while trying to drop a card index onto an open slot if the tip of the arrow is on the small sliver of gray between slots.
5. The best handling to minimize these types of errors is to have the tip of the mouse arrow in the center of the card index while you are moving it, and to position it carefully before releasing the mouse button.
6. If the card index disappears completely, you can always press the Reset: New button to return all of the cards to their original position, or re-import your stack.

### ***Retained vs. New Stack Values***

While you are building a *Custom Deck*, the *Stack Values* associated with the cards may have one of two states, *New* or *Retained*.

The *New* state has no *Stack Values* associated with any cards until the *Custom Deck* has been arranged and finalized. At that point, the *Stack Positions* are assigned as the new *Stack Values* during the transfer to the main Deck window. Whenever the *Custom Deck* is in the *New* state, the *Stack Value* labels will all be above the dividing line, and above each of the card slot positions. When the card indices are moved around while in the *New* state, the *Stack Value* labels will stay above the dividing line (because they have not been finalized).

There are only three ways to have *Retained Stack Values*: 1) you imported a deck into the *Custom Deck* module, 2) you created a *Stanyon Variation Deck*, and 3) you completely arranged a *Custom Deck* from the *New* state, and then pressed the: *Reset: Retain* button, as described on page 138.

When the *Custom Deck* is in the *Retained* state, the *Stack Value* labels will stay with the card indices when they are moved around.

You will want to use the *New* state when you are creating a brand new stack, and you want to assign new *Stack Values* to each card. You will want to use the *Retained* state when you want to rearrange an existing stack and maintain each card's *Stack Values*.

## ***Control Buttons***

### **Import Current Deck from StackView**

This button will *Import* the current deck from the *Deck* window to the *Custom Deck* design area. When the deck is imported to the *Custom Deck* area, the *Stack Values* are retained by StackView, and you will be left in the *Retained* state. In addition, the *Selected* and *Reversed* states of each card are also imported, but not displayed. If the cards are transferred back to the Deck window with retained *Stack Values* (see more on page 137), the *Selected* and *Reversed* states of each card will be also transferred, but in the newly defined positions from your *Custom Deck* work.



**Transfer: New**

This button will *Transfer* the completed stack above the dividing line to the *Deck* window for further manipulation in StackView. All position slots above the dividing line must be filled before you can press this button. When the *Custom Deck* is transferred with this button, the *Stack Values* that are transferred will be reset to the *Stack Position* values, regardless of whether you are in the *Retained* or *New* state. In other words, when you press this button, the card occupying the first slot will get a *Stack Value* of 1, the second slot a 2, and so on, all the way to 52.

**Transfer: Retain**

This button will *Transfer* the completed stack above the dividing line to the *Deck* window for further manipulation in StackView. All position slots above the dividing line must be filled before you can press the button. When the *Custom Deck* is transferred to the *Deck* window, the sequence of cards will be the same as in the custom stack design area. However, the *Stack Value* designations that go with the cards will be the same as when they were retained. In addition, if you imported the *Custom Deck* cards before doing the *Transfer*, any *Selected* or *Reversed* card conditions will be kept and transferred.

You may only use this button if the cards have true *Retained Stack Values* associated with them. Otherwise, you must use the *Transfer: New* button to transfer a newly created *Custom Stack*. As described earlier, there are only three ways to have *Retained Stack Values*: 1) you imported a deck into the *Custom Deck* module, 2) you created a *Stanyon Variation Deck*, and 3) you completely arranged a *Custom Deck* from the *New* state, and then pressed the *Reset: Retain* button.

**Reset: New**

This button will cause all of the card indices above the dividing line to return to their home positions below the line. If you had imported the deck, pressing this

button will clear the *Stack Value* assignments from each card. The *Stack Value* labels will be positioned above the open card slots above the dividing line. You will be left in the *New* state.

### **Reset: Retain**

This button will cause all of the card indices above the dividing line to return to their home positions below the line. *Stack Values* assigned to the cards will be *Retained* and the *Stack Value* labels will position themselves above the card indices. You will be left in the *Retained* state, and all cards with *Retained Stack Values* will have their *Stack Value* labels move with the indices.

If you completely arrange a *Custom Deck* from the *New* state, and press this button, the *Custom Deck* will behave as though it was imported. The *Stack Values* of the cards will be retained.

### **Reorder Retained Stack**

When you press this button, all of the card indices will be positioned above the dividing line in the *Retained* stack order that you had previously imported or established. This button is only enabled when *Retained Stack Values* exist in the *Custom Deck*.

### ***Create a Stanyon Variation Deck***

To create a *Stanyon Variation* stack, you must fill out the first five position slots with card indices before you press this button. There are two constraints on the five cards:

1. The first four cards must include one of each of the suits in any order.
2. The value of the fifth card must not be the same as the value of the first card. (For example, the first and fifth card may not be both Aces.)

The fifth card should technically match the suit of the first card, but StackView uses only the value of the card to build the stack. If the suit of the fifth card

does not match the suit of the first card, StackView will automatically put the correct card index there after you press the button.

Only the first five cards matter. The rest of the card indices can be anywhere (above or below the dividing line) when you press the button to build the *Stanyon Variation* stack. If you violate any of the constraints, you will get an appropriate error prompt. (For those of you who may be interested, there are 8,225,568 different possible *Stanyon Variation* stacks. A few of them are quite interesting.)

Once you press the button, the rest of the deck will automatically be positioned above the dividing line. If you create a *Stanyon Variation* stack from the *New* state, the *Stack Value* labels will reflect a new stack numbered sequentially from top to bottom. If you create a *Stanyon Variation* stack from the *Retained* state, each card index will retain its original *Retained Stack Value* label.

### **Stanyon Variation Results Output**

After you have successfully built a *Stanyon Variation* stack, a special line of text will appear below the button.

For example, if you enter the first five cards of the original *Stanyon Stack*, as shown in Figure 73, and press the button to build the rest of the stack, you will get a results output as shown in Figure 74.

The first part of the output tells you what the first card in the stack is. The second part of the output tells you the cycling order of the suits. The third part of the output shows the card increments that you need to use to calculate the next card in the new stack. Each suit has its corresponding increment value. (The first suit with the first increment value, the second with the second, etc.)

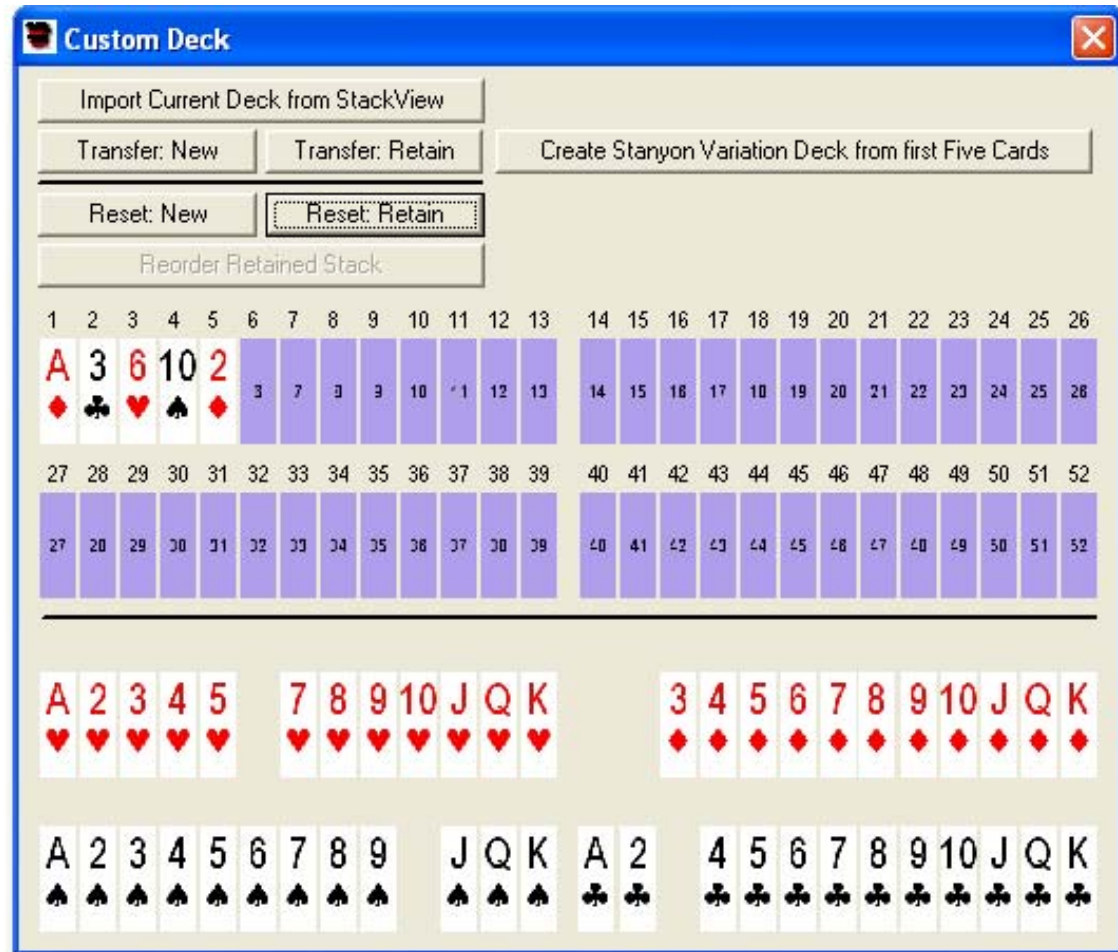


Figure 73

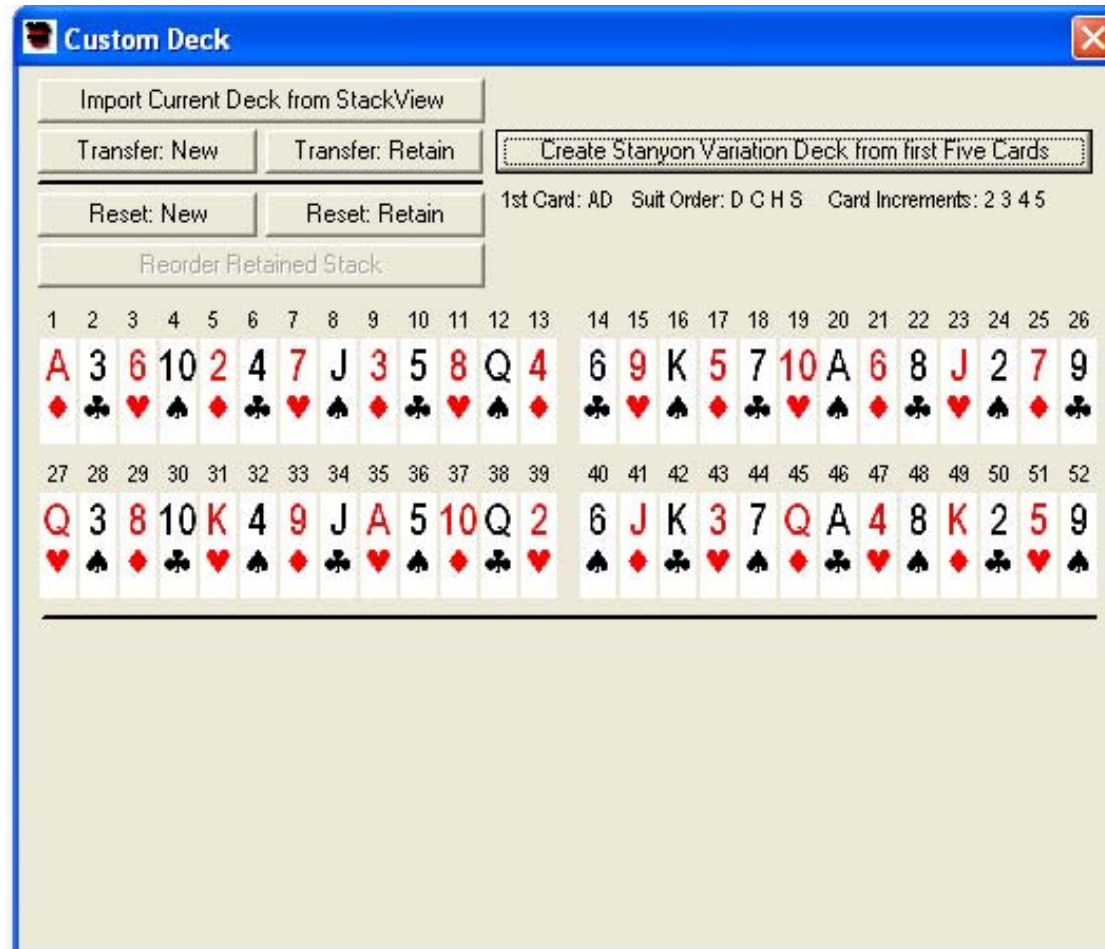


Figure 74

In the above example, to determine the next card after a *Diamond* card, you need to add 2 to the *Diamond's* value, and then increment the suit based on the

suit cycle. So, the card after the *Nine of Diamonds* would have to be the *Jack of Clubs* ( $9 + 2 = 11$  for the *Jack*, and *Clubs* because it comes after *Diamonds* in the suit cycle). Continuing the example, to determine the value of the next card after a *Club*, you add an increment of 3, and its suit will be a *Heart*. To determine the value of the next card after a *Heart*, you add an increment of 4, and its suit will be a *Spade*. And to determine the value of the next card after a *Spade*, you add an increment of 5, and its suit will be a *Diamond* since you have to loop back to the beginning of the suit cycle.

The card increments can also have negative values based on the first five cards.

When you calculate the value of the next card, if the value goes above 13 (*King*), you just cycle around to the *Ace*. For example, a calculated value of 16 cycles around to a *Three*. (This is also known as "casting out the 12s.") Likewise, if you have negative increments and end up with a calculated negative number, you just cycle back around towards the *King*. For example, if you get calculated value of -2 cycles back to a *Queen*.

### ***Importing a Deck***

You *Import* a deck from the *Deck* window by pressing the *Import Current Deck from StackView* button as described on page 136. When you *Import* a deck, the *Stack Value* of each card is *Retained* for the duration of your *Custom Deck* work unless you press the *Reset: New* button (page 137). (For more information on the full detail of *Retained Stack Values*, see the information on page 135.)

The typical use of an imported deck is to re-order it for a particular routine, while still retaining the original *Stack Values*. After the deck is re-ordered, it is transferred back to the *Deck* window by pressing the *Transfer: Retain* button. From there, you can save the file in that new order for later access as described on page 143.

### ***Transferring a Deck***

You transfer your custom deck to the Deck window by pressing either the Transfer: New button or the Transfer: Retain button as described 137. The difference is whether the Stack Values are Retained. (For more information on the full detail of *Retained Stack Values*, see the information on page 135.)

### ***Saving Your Created Stack***

After you have successfully transferred your *Custom Deck* to the *Deck* window, you can save your stack by using the *Save Deck As* menu item as described on page 203.

If you would like to save your custom stack (or any stack) as the default stack to start with when the program loads, you need to save it as your *Default Stack*. The *Default Stack* must be named `StackView.svf` and it must be saved in the same directory as the StackView program. There are two ways to do this:

1. The easiest way to create a default stack file is to select the *Save Deck As Default* item from the *File* menu (page 204). This will create the `StackView.svf` file using the current deck and order that is in the *Deck* window. It will save the file to the correct directory automatically.
2. The other way to create a default stack is to use the *Save Deck As* item from the *File* menu (page 203). When prompted for the filename, enter the name "StackView" and make sure to save it in the same directory as the StackView program.

## Reference: Custom Mnemonics

Many magicians use mnemonics to learn their stacks. Once a stack has been learned with confidence, most people quickly forget the mnemonic aids that were so instrumental at the beginning of the learning process. StackView provides a Custom *Mnemonics* capability that is integrated with the *StackView Test* module as described on page 167.

### ***Default Mnemonic File***

StackView always opens a *Default Mnemonic File* when the program loads. If you have not yet defined your own *Default Mnemonic File*, StackView uses the *Aronson Mnemonic* set as the default. If you do create your own mnemonic file, you can save it as the default. After that, whenever StackView loads, it will also load your *Default Mnemonic File* into memory.

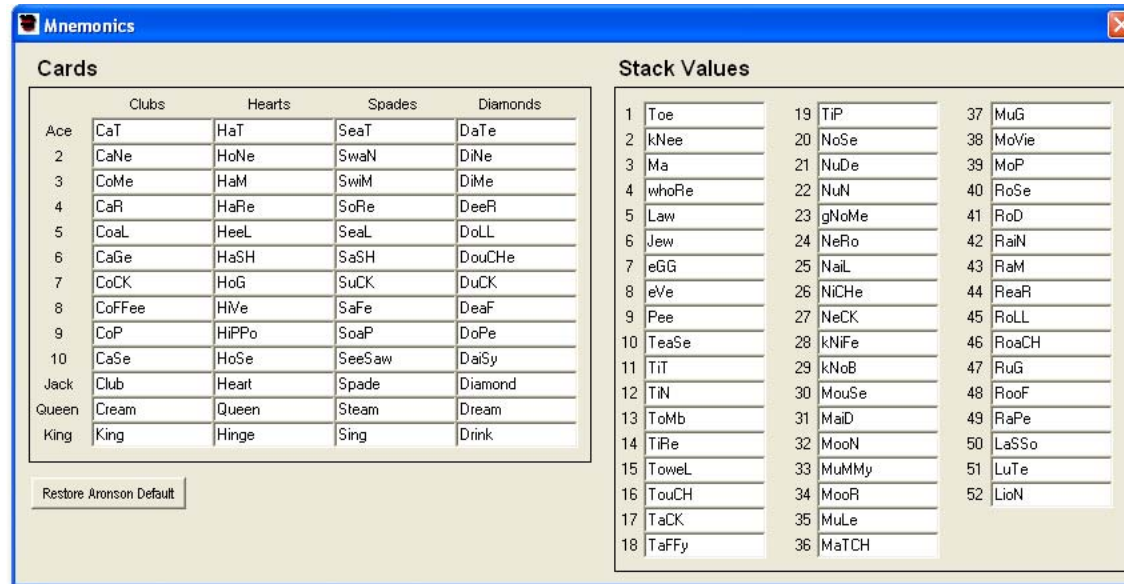
### ***Creating Custom Mnemonics***

To view and edit the current mnemonic file, select the *Custom Mnemonics* item from the *Tools* menu (page 209). You will see the *Custom Mnemonics* editing module as shown in Figure 75. When the *Custom Mnemonics* menu selection is checked, the *Custom Mnemonics* window is visible. When the *Custom Mnemonics* menu selection is unchecked, the window is hidden. When the *Custom Mnemonics* window is closed, the *Custom Mnemonics* menu item will become unchecked.

There are two tables: one for the *Card Value* mnemonics and one for the *Stack Value* mnemonics. By editing the text in the fields, you can customize your mnemonic associations.

There is a button labeled *Restore Aronson Default*, which, as its name suggests, will reset the mnemonics to the Aronson set.





**Figure 75**

If you would like to start your editing from another existing mnemonic file, you use the *Open Mnemonic* item from the *File* menu (page 206).

When you edit the fields, your entries are only saved to memory if you move out of the field or press the Enter key. If you edit an entry without changing fields or pressing Enter, the new data will not register in the appropriate memory variables. You can change fields by pressing the Tab key, or by clicking in a new field with your mouse.

### ***Saving a Mnemonic File***

Once you have established your custom mnemonic settings, you can save your file. Just select the *Save Mnemonic As* item from the *File* menu (page 206) and you will be prompted for a file name. You can also set your custom mnemonic

file as the default mnemonic set to use when StackView loads by selecting *Save Mnemonic As Default* item from the *File* menu (page 206).

## Reference: Back Design

Many of StackView's *Events* allow you to reverse cards in the deck. When a card is reversed, a *Back Design* will show. StackView uses a red *Bicycle Rider* back design. However, StackView allows you to select a *Back Design* from a predefined set. You can also select the color of the *Back Design*.

From the Tools menu, select the *Set Back Design* item (page 210) to open the *Back Design* module. When the *Back Design* menu selection is checked, the *Back Design* window is visible.

When the *Back Design* menu selection is unchecked, the window is hidden. When the *Back Design* window is closed, the *Back Design* menu item will become unchecked.

Select one of the back designs from the dropdown box as shown in Figure 76. Finally, specify either Red or Blue by clicking on one of the options. Finally press the OK button, and the back design of all of the cards will be set to your specification.

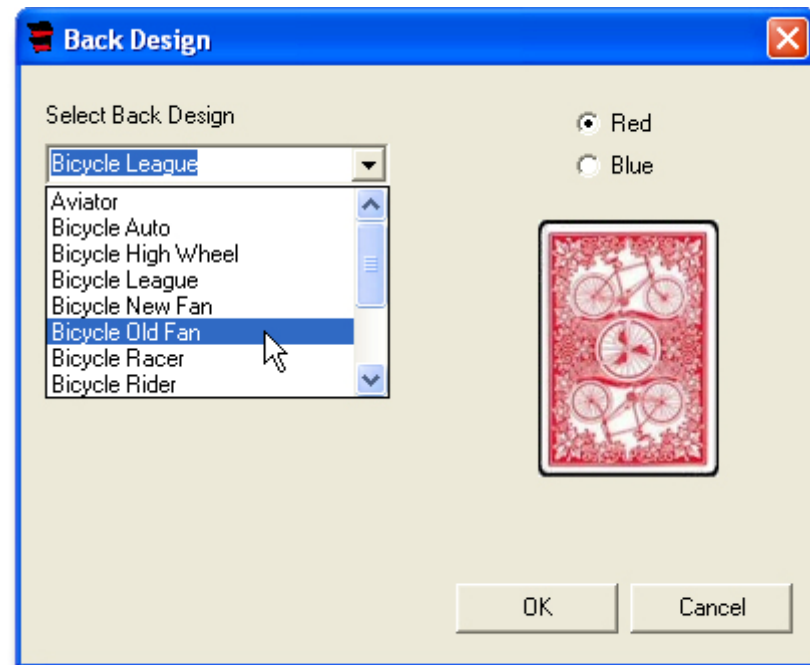


Figure 76

## Reference: Joyal ShuffleMeter

The Joyal ShuffleMeter is a stack analysis tool used to assess the randomness of a deck's order. It is not my intention to describe the full details of the ShuffleMeter; for its details, you are referred to Martin Joyal's book, *The Six-Hour Memorized Deck*, Appendix B. It was published in 1997 by Hermetic Press, Inc.

You access the *Joyal ShuffleMeter* from the *Tools* menu on the menu bar. When the *Joyal ShuffleMeter* menu selection is checked, the *Joyal ShuffleMeter* window is visible. When the *Joyal ShuffleMeter* menu selection is unchecked, the window is hidden. When the *Joyal ShuffleMeter* window is closed, the *Joyal ShuffleMeter* menu item will become unchecked.

The *Joyal ShuffleMeter* will automatically update itself whenever the *Deck* window contents change.

### ***Analysis Tabs***

There are five tabs in the *ShuffleMeter* window. Each provides a different perspective on the degree of randomness of the shuffled state of the deck.

Each tab contains a set of numbers labeled *average*. These numbers refer to the average value for that parameter from a large sample of well-shuffled decks.

The cumulative differences of the actual to the average values indicate the degree of randomness of the deck.

Martin did ask that I reprint a small disclaimer from the book:

"I do not claim that the ShuffleMeter is the only means of measuring the shuffled qualities of a deck, nor the best means. It is even possible to trick the ShuffleMeter. However, the indications it provides are useful, particularly in determining if the characteristics of a card setup are close to those of a well-shuffled deck."

## Spread

The *Spread* tab (Figure 77) shows a visual representation of the state of the deck. The parameters at the top of the tab indicate the cyclical nature of the deck.

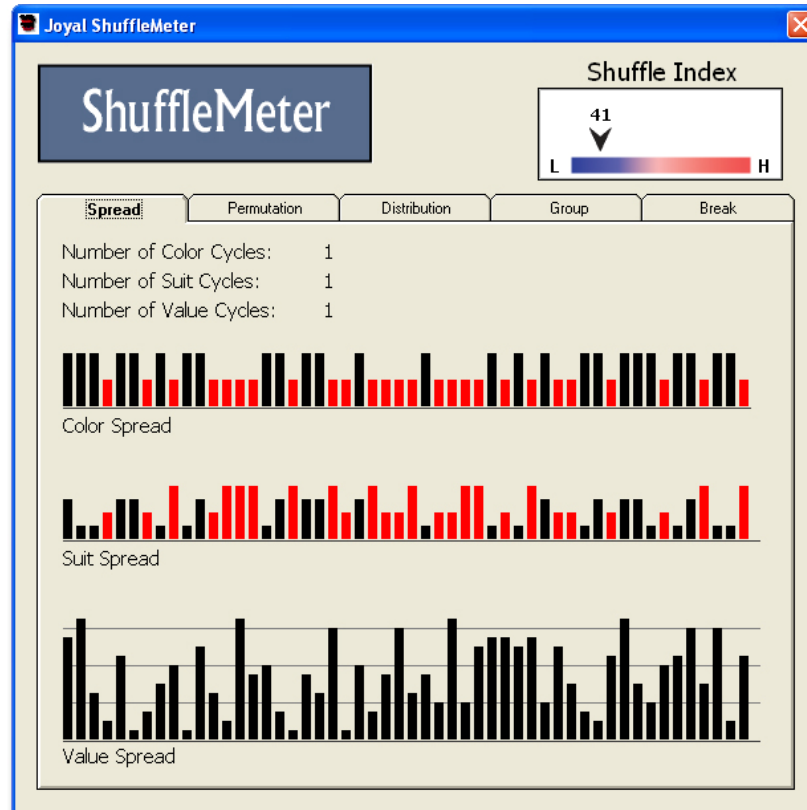


Figure 77

## Permutation

The *Permutation* tab (Figure 78) presents an analysis of the permutations of the thirteen consecutive four-card packets in the deck.

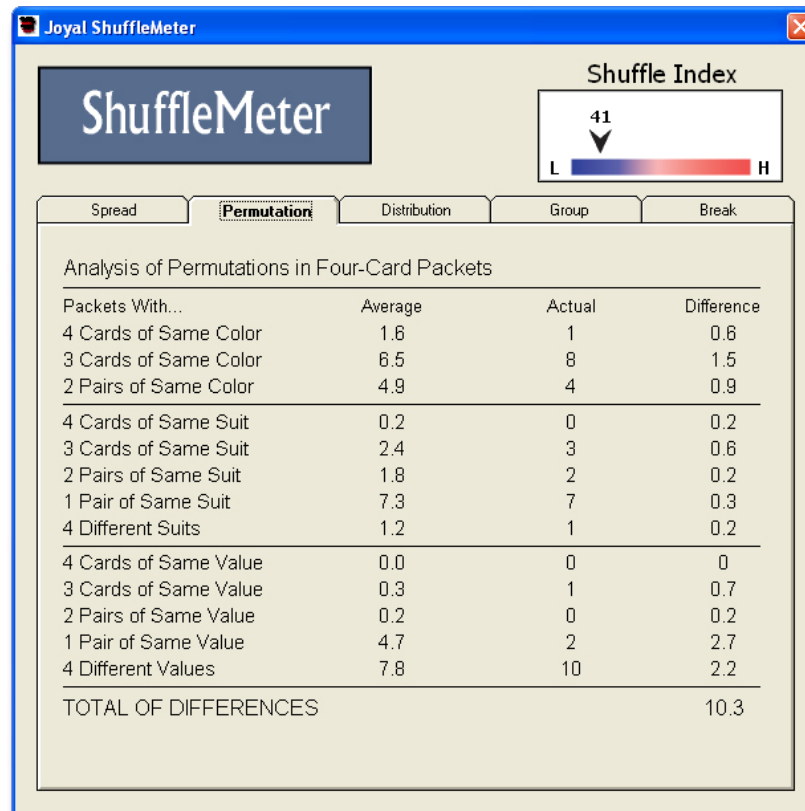


Figure 78

## Distribution

The *Distribution* tab (Figure 79) presents an analysis of color and suit distributions among the four quartiles of the deck.

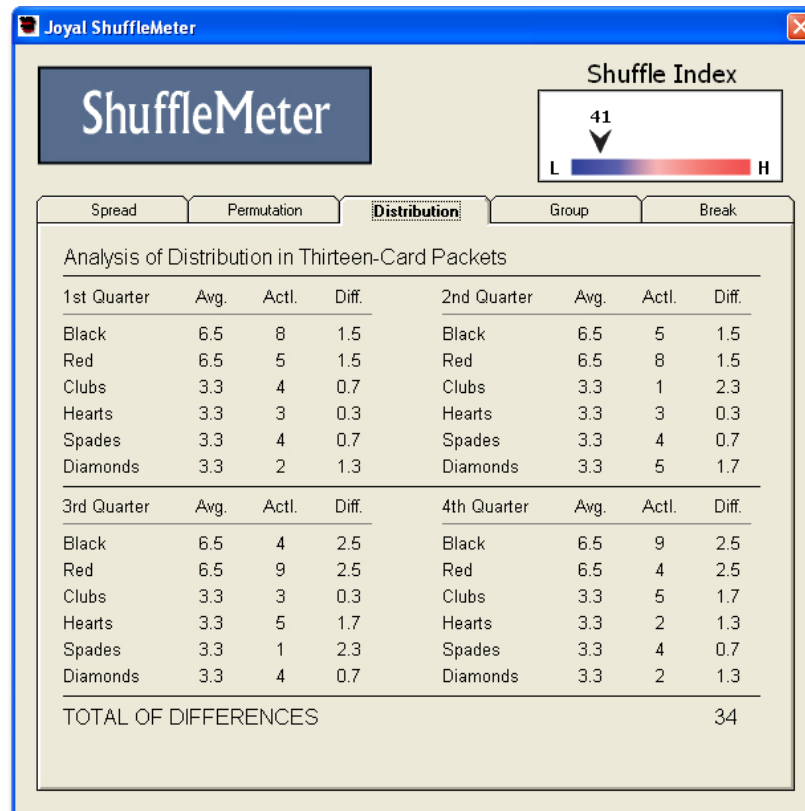


Figure 79

## Group

The *Group* tab (Figure 80) provides an analysis of the number of groups of colors, suits and values throughout the deck.

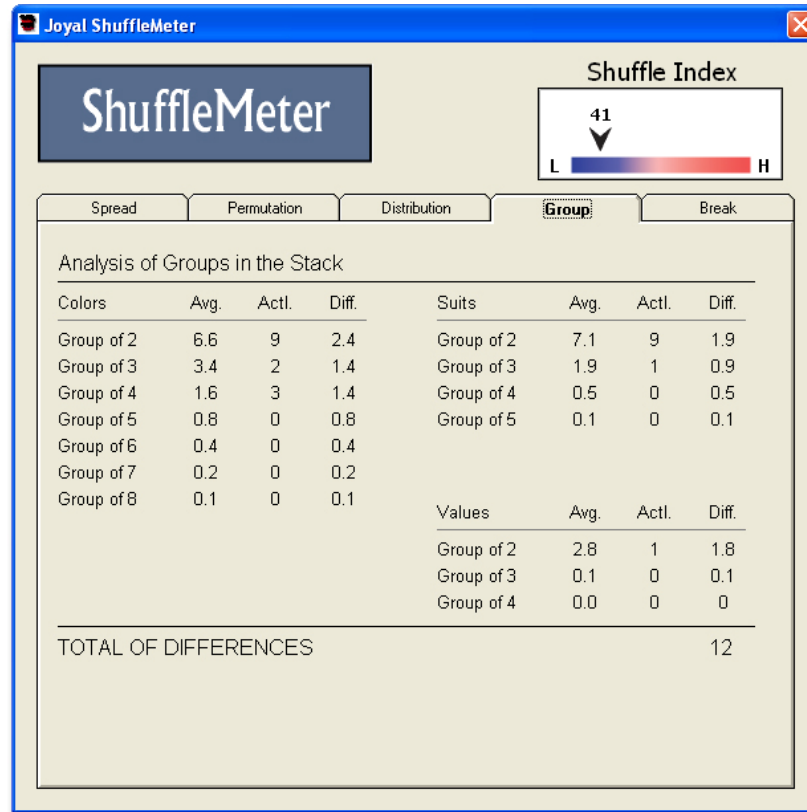


Figure 80



## Break

The Break tab (Figure 81) provides an analysis of the number of breaks in pattern of the colors, suits and values.

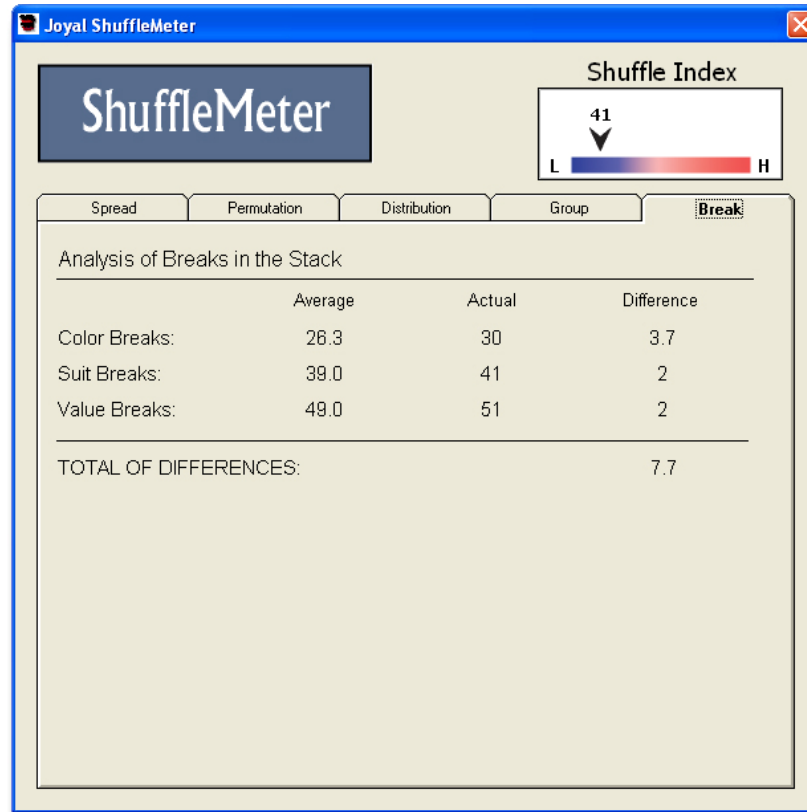
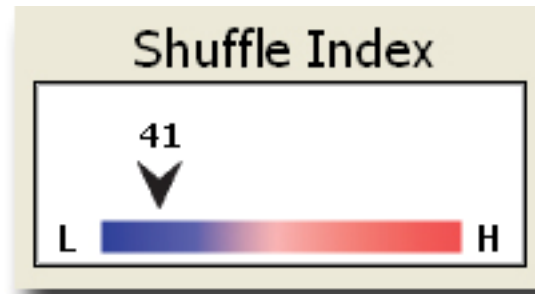


Figure 81

### ***Shuffle Index***

In the upper right hand corner of the window is the *Shuffle Index* as shown in Figure 82. The index is derived from the three component differences of the *Break* tab. The formula is:

$$\text{Shuffle Index} = 2 * \Delta \text{ Color Breaks} + 4 * \Delta \text{ Suit Breaks} + 13 * \Delta \text{ Value Breaks}$$



**Figure 82**

The values of a well shuffled deck range between 0 and 110. Any values above 110 indicate some level of regular patterns or sequences.

The *Shuffle Index* will be indicated by a floating arrow with the index value shown above it. There is an "L" at the left end of the bar indicating a *Low* degree of patterns and sequences in the deck. At the right end of the bar, there is an "H" indicating a *High* degree of patterns and sequences in the deck. The right edge of the bar is proportional to an index value of 250. It is possible to have an index reading greater than 250; in these cases, the arrow is shown off the right edge of the bar, with the correct index value shown above it.

## Reference: StackView Test

StackView includes a “quizzer” module called *StackView Test* that helps you test your knowledge of your favorite stack. It can test for *Card Values*, *Stack Values*, or a random mix. It can test you in the forward, reverse, current deck order, or random sequence. You can also specify a portion of the stack to test, either by *Stack Values* or *Position Values*. *StackView Test* will report the elapsed time of your test so that you can see how well you know your stack “cold.” You control the pace of the test by clicking special buttons on the screen. You can specify the *Test* and *Show* times so that *StackView Test* can run automatically. You can disable the timers and just run the test at your own pace. Finally, if you use mnemonics during the learning stages, you can enable the mnemonic hints that you established (page 144).

StackView also allows you to test your knowledge of the next or previous card in your stack. For example, what card comes after *Stack Value* 18? (Seven of Spades in Aronson) Or what *Stack Value* comes before the Three of Diamonds? (11 in Tamariz)

During the test, StackView will maintain a progress bar to show you how many cards are remaining in the current test.

The *StackView Test* approach does not require you to enter your answers to the questions. It is an honor based approach. As fast as you can think of the correct answer, you can move on to the next test item. You will be your own judge as to your true knowledge of the stack. This testing module is designed to help you with your speed and accuracy so that you can learn your stack “cold.”

You may also want to disable the *Deck* window to avoid inadvertent “peeking” at the deck during a test. This is up to you.

### ***Accessing StackView Test***

You access *StackView Test* from the *Tools* menu on the menu bar. When the *StackView Test* menu selection is checked, the *StackView Test* window is visible as shown in Figure 83. When the *StackView Test* menu selection is unchecked, the window is hidden. When the *StackView Test* window is closed, the *StackView Test* menu item will become unchecked.

### ***Deck Range***

The *StackView Test* module allows you to test however many cards you want to test in a testing session, and which ones, by specifying one of the *Deck Range* options.

Once the test begins, you may not change the *Deck Range* of the test. During a test, those options are disabled.

### **Full Deck**

You select this option if you would like to test the entire deck of cards.

### **Stack Value Range**

When you select the *Stack Value Range* option, you must enter the *Start* and *Finish* numbers for your desired portion of the deck. Since this is a range of *Stack Values*, the order of the current deck in the *Deck* window will be disregarded.

### **Current Deck Position Range**

When you select the *Current Deck Position Range* option, you must enter the *Start* and *Finish* numbers for your desired portion of the deck. Since this is a range of *Position Values*, the specific cards at those *Positions* of the current deck in the *Deck* window will be used for the test.

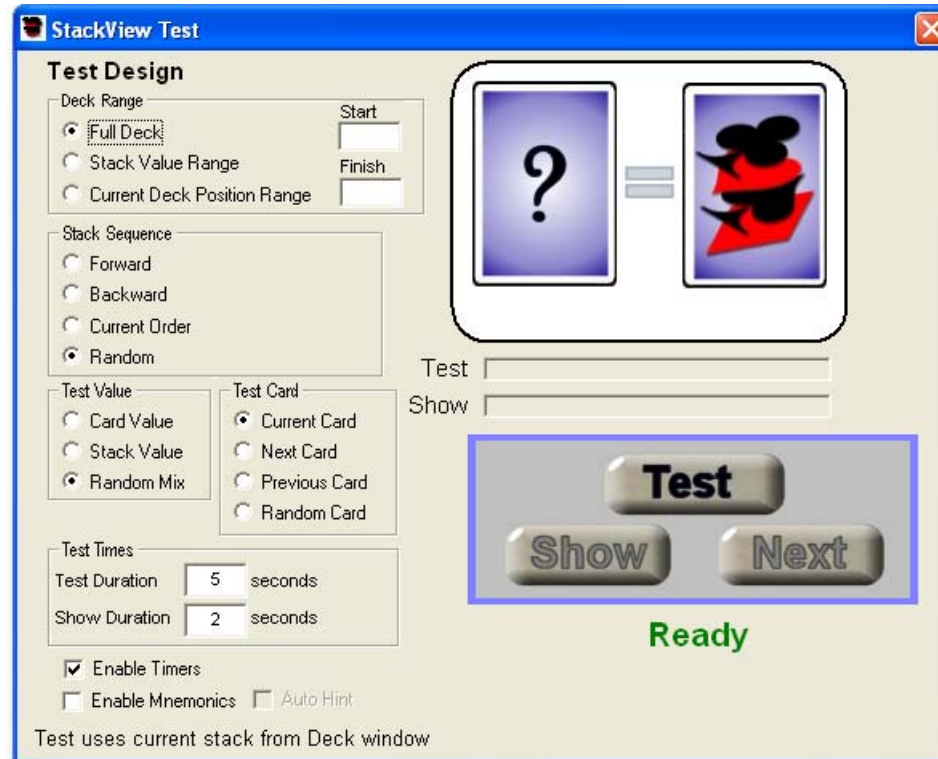


Figure 83

### ***Stack Sequence***

The *StackView Test* module will test you once for every card in the selected portion of the current active stack in the *Deck* window. You need to select which sequence will be presented to you with the *Stack Sequence* options.

Once the test begins, you may not change the *Stack Sequence* of the test. During a test, those options are disabled.

## **Forward**

When you select the *Forward* option, its behavior will depend on the *Deck Range* setting.

If you have selected the *Full Deck* or the *Stack Value Range*, as your range, the *Forward* option will cause the order of the cards to be presented to you in their original *Stack Value* sequence. The order of the current deck in the *Deck* window is disregarded.

If you have selected the *Current Deck Position Range*, the *Forward* option will cause the order of the cards to be presented to you in their current *Position Value* sequence.

## **Backward**

When you select the *Backward* option, its behavior will depend on the *Deck Range* setting.

If you have selected the *Full Deck* or the *Stack Value Range*, as your range, the *Backward* option will cause the order of the cards to be presented to you in the reverse order of the original *Stack Value* sequence. The order of the current deck in the *Deck* window is disregarded.

If you have selected the *Current Deck Position Range*, the *Backward* option will cause the order of the cards to be presented to you in the reverse order of the current *Position Value* sequence.

## **Current Order**

The *Current Order* button is only enabled when the *Full Deck* option for the *Deck Range* is selected. When this option is selected, the deck will be tested in the order as it appears in the *Deck* window. If you want to test the full deck in its shuffled order, use the *Current Deck Position Range*, with *Start* and *Finish* values of 1 and 52, and set the *Stack Sequence* to *Forward*.

If your *Deck Range* has a partial deck setting, and you want to have the test use the current order, you should use the *Forward* option.

### **Random**

When you select the *Random* option, the specified *Deck Range* will be presented to you randomly during the test. Each card will be shown only once during the test.

### **Test Value**

The *Test Value* option will determine what piece of information will be asked for during the test. You must select one of the options.

Once the test begins, you may change the *Test Value* of the test. During a test, those options are enabled.

### **Card Value**

When you select the *Card Value* option, you will be shown the *Stack Value* as the example shows in Figure 84. The *Stack Value* will always be shown on the left side of the card pair, and you need to identify the *Card Value* before you move on to the next card.

When the *Card Value* option is selected, a small blue arrow between the card pair is displayed pointing to the right, as highlighted in Figure 85. This indicates that *StackView Test* is asking for the *Card Value*.



Figure 84



Figure 85

### **Stack Value**

When you select the *Stack Value* option, you will be shown the *Card Value* as the example shows in Figure 86. The *Card Value* will always be shown on the right side of the card pair, and you need to identify the *Stack Value* before you move on to the next card.



When the *Stack Value* option is selected, a small blue arrow between the card pair is displayed pointing to the left, as highlighted in Figure 87. This indicates that *StackView Test* is asking for the *Stack Value*.

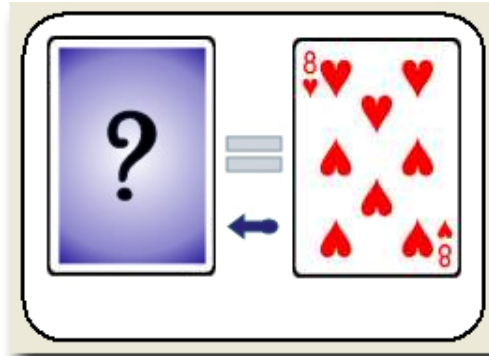


Figure 86

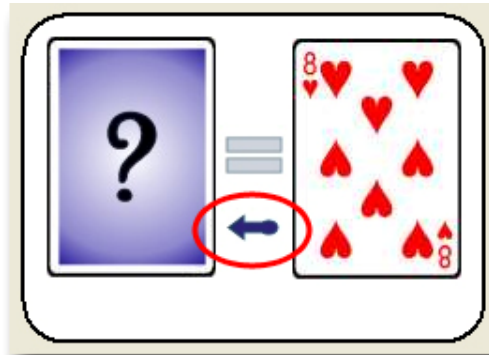


Figure 87

### **Random Mix**

When you select the *Random Mix* option, you will be shown the either the *Card Value* or the *Stack Value* randomly throughout the test. When the test is asking for the *Stack Value*, the *Card Value* will always be shown on the right side of the card pair, and you need to identify the *Stack Value* before you move on to the next card. Similarly, when the test is asking for the *Card Value*, the *Stack Value* will always be shown on the left side of the card pair, and you need to identify the *Card Value* before you move on to the next card. The test card will change randomly throughout the test, unless you change the *Test Value* option selection.

When the *Random Mix* option is selected and the test has begun, a small blue arrow between the card pair is displayed pointing to either the left or right side, depending on the random *Test Value*. When the arrow points to the left as shown in Figure 87, you are shown the *Card Value* on the right, and are asked for the *Stack Value* on the left. Likewise, when the arrow points to the right as shown in Figure 85, you are shown the *Stack Value* on the left, and are asked for the *Card Value* on the right.

### **Test Card**

You are able to specify which relative card is tested for, with respect to the shown card or *Stack Value*.

Once the test begins, you may change the *Test Card* of the test. During a test, those options are enabled.

### **Current Card**

If you select the *Current Card* option, an "equals" sign will be shown between the cards as shown in Figure 88, and you will be tested for the same card information. For example, with the Aronson Stack, if the test shows you a *Stack Value* of 24 with an "equals" sign, you are being tested for the *Current Card*

value of *Three of Diamonds*. Likewise, if the test shows you a *Jack of Clubs* and an "equals" sign, you are being tested for the *Current Card Stack Value* of 33.



Figure 88

### Next Card

If you select the *Next Card* option, a "plus" sign will be shown between the cards as shown in Figure 89, and you will be tested for the *Next Card* information to the one you are being shown. For example, with the Aronson Stack, if the test shows you a *Stack Value* of 24 with a "plus" sign, you are being tested for the next *Card Value* of the stack, which is the *Seven of Hearts*. Likewise, if the test shows you a *Jack of Clubs* and a "plus" sign, you are being tested for the next card's *Stack Value* of 34.

If the test card shown is the last card in your stack, you will be tested for the next cyclical value, which is the top card of your stack (*Stack Value* = 1).

### Previous Card

If you select the *Previous Card* option, a "minus" sign will be shown between the cards as shown in Figure 90, and you will be tested for the *Previous Card* information to the one you are being shown. For example, with the Aronson

Stack, if the test shows you a *Stack Value* of 24 with a "minus" sign, you are being tested for the previous *Card Value* of the stack, which is the *Eight of Spades*. Likewise, if the test shows you a *Jack of Clubs* and a "minus" sign, you are being tested for the previous card's *Stack Value* of 32.

If the test card shown is the first card in your stack, you will be tested for the previous cyclical value, which is the bottom card of your stack (*Stack Value* = 52).

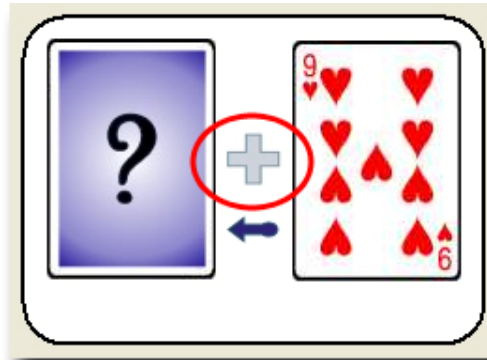


Figure 89



Figure 90

### **Random Card**

If you select the *Random Card* option, StackView will randomly select from among the *Current Card*, *Next Card*, and *Previous Card* options throughout the test. The “equals,” “plus,” and “minus” signs will be updated automatically (even though the “minus” sign shows when you initially select this option).

### **Additional Note:**

If you select *Current Deck Position Range* or *Stack Value Range* as your *Deck Range* setting, StackView will use that range of cards to test you with. However, the answers for *Next Card* and *Previous Card* will extend beyond your range by one card. For example, if your partial range is from *Start* value of 5 to *Finish* value of 10, and you selected *Next Card*, when you are tested for card 10, the answer will be the information about card 11 (not card 5). Likewise, cyclicity (if your range includes the first or last card of the deck) does not apply to the range, but rather to the entire stack.

### ***Timers***

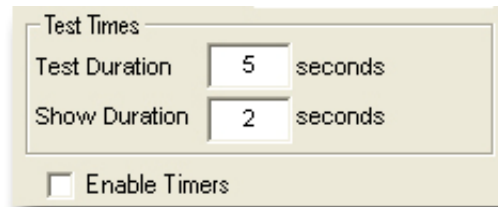
You can set the amount of time that you are tested on a card before you are shown the correct answer. You can also set the amount of time the answer is shown to you before you are automatically tested on the next card. You may also chose to not use the timers, and have the next test card be prompted by you.

Once the test begins, you may not change the *Timer* settings.

### **Durations**

You enter the number of seconds you want for the timers in the *Test Duration* and *Show Duration* text input fields, as shown in Figure 91. You may enter any time between 0.1 seconds and 60 seconds. You may use decimal times such as 2.5 seconds. Any time entry will round to the nearest tenth of a second. For

example, if you enter a *Test Duration* of 3.44 seconds, StackView will round it to 3.4 seconds.



Test Times

Test Duration 5 seconds


Show Duration 2 seconds

Enable Timers

**Figure 91**

The *Test Duration* is the amount of time that the test question is shown before the answer is automatically shown to you. The *Show Duration* is the amount of time that the answer will be displayed before StackView automatically moves on to the next test question.

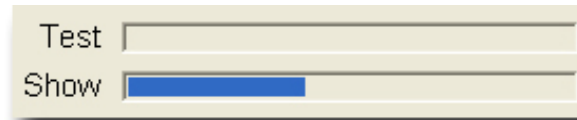
During the test, two progress bars will activate beneath the pair of testing cards. The top bar corresponds to the *Test Duration*, and it will move from the left to the right as the test timer elapses, as shown in Figure 92. During this time, the test question is active. The bottom bar corresponds to the *Show Duration*, and it will also move from the left to the right as the show timer elapses, as shown in Figure 93. During this time, the answer is being shown.



Test [Progress Bar]

Show [Progress Bar]

**Figure 92**



**Figure 93**

As the timers cycle, the progress bars will indicate how much testing time has elapsed for each part of the test.

### **Enable Timers**

The *Enable Timers* checkbox is checked as the default setting. If you uncheck the box, the progress bars will disappear, and the timer inputs will be disabled. In addition, the test will not have any timers operating. You must press the *Next* button to advance the test to the next card. Finally, with the *Enable Timers* box unchecked, there will not be an elapsed time reported at the end of the test.

### ***Mnemonics***

*StackView Test* incorporates the *Custom Mnemonic* settings as aids in learning your stack. For more information on working with the *Custom Mnemonic* settings, see page 144.

Once the test begins, you may change the mnemonic settings of the test. During a test, those options are enabled.

### **Enable Mnemonics**

When the *Enable Mnemonics* box is checked, the mnemonic aids are activated. A new button labeled "Hint" will appear in the control button area as shown in Figure 94 (unless the *Auto Hint* option is also selected).

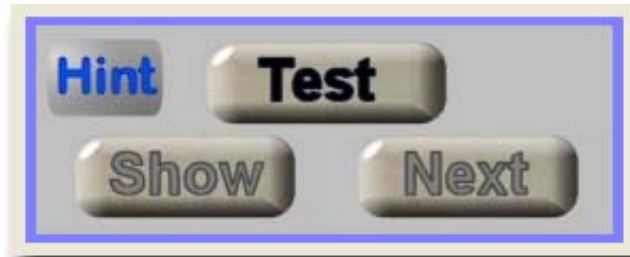


Figure 94

When mnemonics are enabled, and the *Auto Hint* feature is off, you will be shown the mnemonic of the card that is being shown in the test question. For example, Figure 95 shows the *Stack Value* of 23 with its Aronson mnemonic aid of "gNoMe." This mnemonic aid will help trigger the answer for you. When the answer is being shown, as in Figure 96, the answer card's mnemonic aid is also shown.



Figure 95





Figure 96

### Auto Hint

When the *Auto Hint* option is selected, you will be shown both parts of the mnemonic pair during the test portion of the question. The *Hint Button* will be removed from the control button area.

If the *Test Card* is activated as the *Next Card* or *Previous Card* setting, the mnemonic aid for the answer card is set correctly to the *Next* or *Previous* card. For example, Figure 97 shows the hint for the *Next Card's Stack Value* starting from the *Five of Hearts*. In this case, the next card in the Aronson Stack is the *Two of Diamonds*, and its *Stack Value* mnemonic is "ToMb."

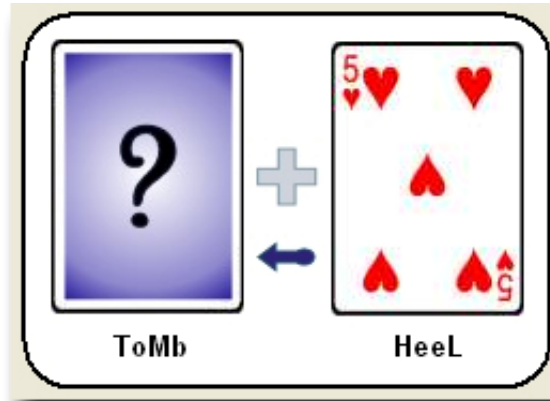


Figure 97

### ***Control Buttons***

Beneath the pair of test cards are the *Control Buttons*. There are three main buttons and an extra *Hint* button that appears when the mnemonic aid is enabled. The buttons will dynamically change their appearance and function based on where you are in the testing sequence. When StackView is not testing, there will be a green "*Ready*" indicator below the control buttons as shown in Figure 98. When testing is in progress, the indicator below the buttons will read: "*Testing*" in red, as shown in Figure 99.

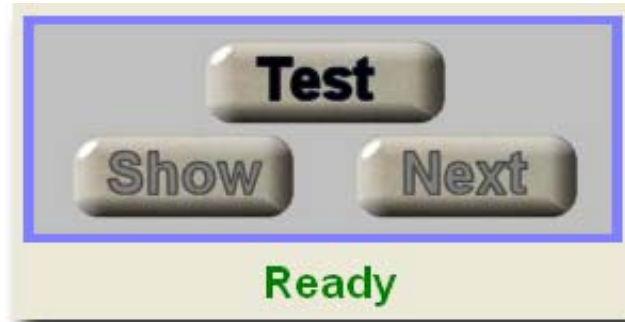


Figure 98



Figure 99

### Test

The top center button is the *Test/Stop* control. It is used to start and stop the test. When you first press the *Test* button to start the testing sequence, it will toggle to display "*Stop*," and the status indicator will read: "*Testing*." The lower right button will read: "*Start*," and the lower left button will indicate "*Show*" with the blue arrow overlay. Figure 100 shows what the buttons look like after you

press the *Test* button. At this stage, the test timer has not started counting yet. It will begin timing only after you press one of the two lower buttons.

Once the test begins, the current question will be displayed only as long as the *Test Duration* setting permits (if the *Enable Timers* box is checked – page 165). If you do not press a button during the *Test Duration* opportunity, the answer to the question will be displayed only as long as the *Show Duration* setting permits. If you do not press a button during the *Show Duration* opportunity, the next question will be shown. This cycle will continue throughout the rest of the defined test.

If you press the *Stop* button at any time during the test, the whole test is disabled, and no elapsed time is reported. (If you want the test to complete with a reported elapsed time, you must end the test correctly by pressing one of the lower buttons when the *Finish* button is showing, or by allowing the final *Show Duration* timer to finish.)

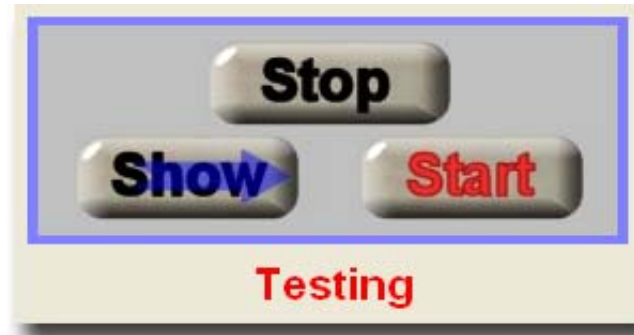


Figure 100

## **Next**

The lower right button is the *Next* control. It is used to advance the test to the next card without showing the answer to the current question (unless the answer is already showing).

The *Next* button will read: "*Start*" at the beginning of the test, as shown in Figure 100. Pressing it in this mode begins the test and the test timers. At the end of the test on the last card, the button will read: "*Finish*" as shown in Figure 101. Pressing it in this mode will end the test, stop the timers, and report the elapsed time (if the *Enable Timers* box is checked).

Throughout the rest of the test, the button will display: "*Next*." Pressing the *Next* button will advance the test to the next card question, and it will restart the timers as well as the progress bars. If you press the *Next* button during a *Test Duration* opportunity, StackView will advance to the next question without showing you the answer to the current question. If you press the *Next* button during a *Show Duration* opportunity, the test will automatically stop showing you the current answer, and will advance to the next card question. It will also reset the timers and the progress bars.

When you are very comfortable with your stack, you will find the *Next* button useful to quickly move through the stack since you already know the answer. This process helps you with speed.

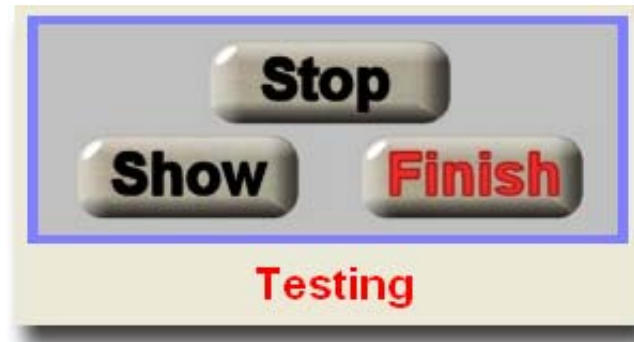


Figure 101

### Show

The lower left button is the *Show/Next* control. It is used to show the answer to the current question, and also to advance the test to the next card. When the button just reads *Show*, pressing it will display the answer to the current question. When the *Show* button has a blue arrow overlaid on it, pressing the button will advance the test to the next card. The blue arrow points to the button to the right (*Next*), which indicates that the *Show* button has taken on the functionality of the *Next* button.

The *Show* button has two modes with separate functions. Depending on the current state of the test, the button will be in one of the two modes.

One of the modes reads: "*Show*" with a blue arrow overlaid on the button. In this mode, the *Show* button behaves exactly like the *Next* button. It will advance the test to the next card question as described in the preceding section. When it is pressed in this mode, it will toggle to the other mode (*Show* with no overlaid arrow).

When the *Show* button is in its regular mode (no overlaid arrow), pressing it will stop the current *Test Duration* timer, and the answer will immediately be displayed. It will also turn on the *Show Duration* timer.

At the beginning and end of the test, the *Show* button will have the blue arrow overlay. It can be used to start and finish the test (just like the *Next* button).

When you are still working on knowing your stack “cold,” you will find the *Show* button useful to pace yourself and learn your stack better. You can keep your mouse pointer over the *Show* button and focus your attention on the test cards. The one button will advance you through all steps of the test procedure.

### Hint

The *Hint* button, as shown in Figure 102, is only visible if the *Enable Mnemonics* box is checked and the *Auto Hint* box is not checked (page 144). The *Hint* button is only active during the *Test Duration* opportunity. When you press the *Hint* button, the mnemonic aid for the answer card is immediately displayed beneath the unanswered card, as shown in Figure 103.

When you are in *Auto Hint* mode, the *Hint* button does not appear.

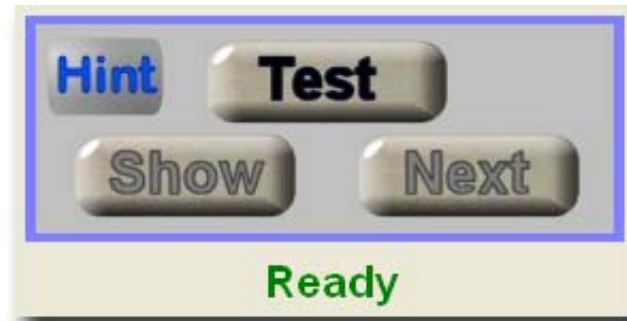


Figure 102

### **Cards Remaining Progress Bar**

During an active test, there is a vertical progress bar to the right of the test cards that indicated how many cards remain in the test, as shown in Figure 103.

### ***Test Results***

When a timed test has completed, StackView will report the total elapsed time for the test. The elapsed time will include both the *Test Duration* and the *Show Duration* times. The report is presented beneath the *Control Buttons* as shown in Figure 104.

Assuming you answered all of the questions correctly, a time of 50 seconds for a full deck with all random settings is excellent.



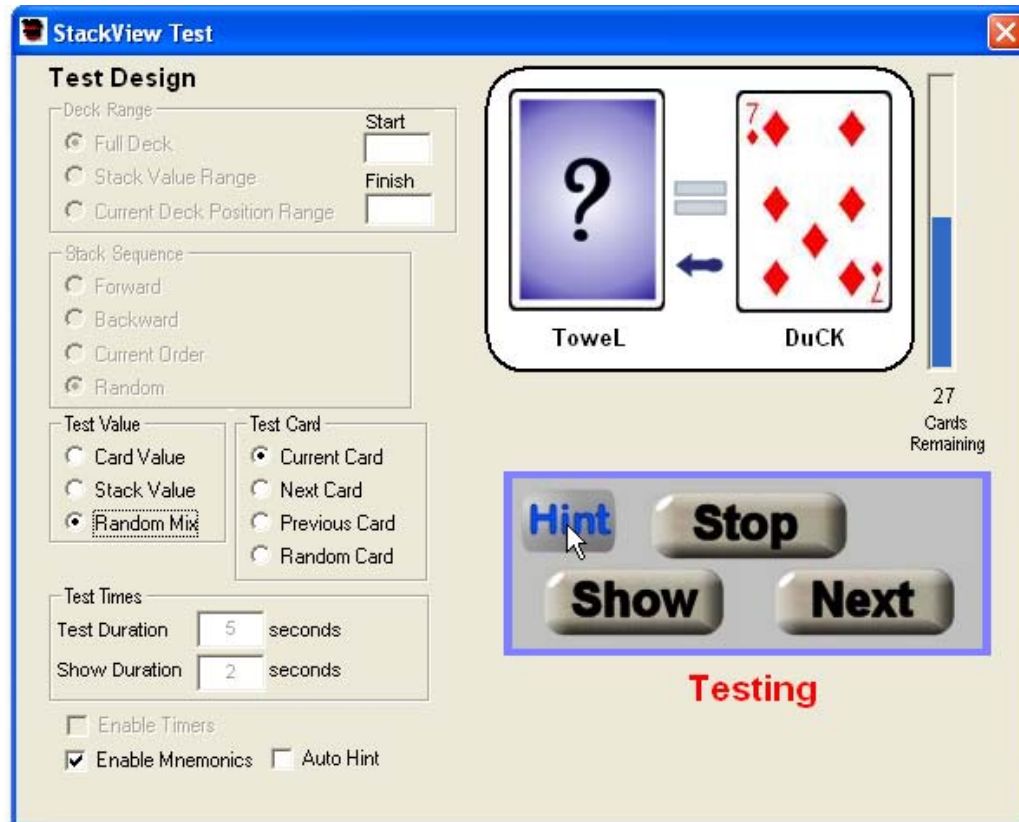


Figure 103

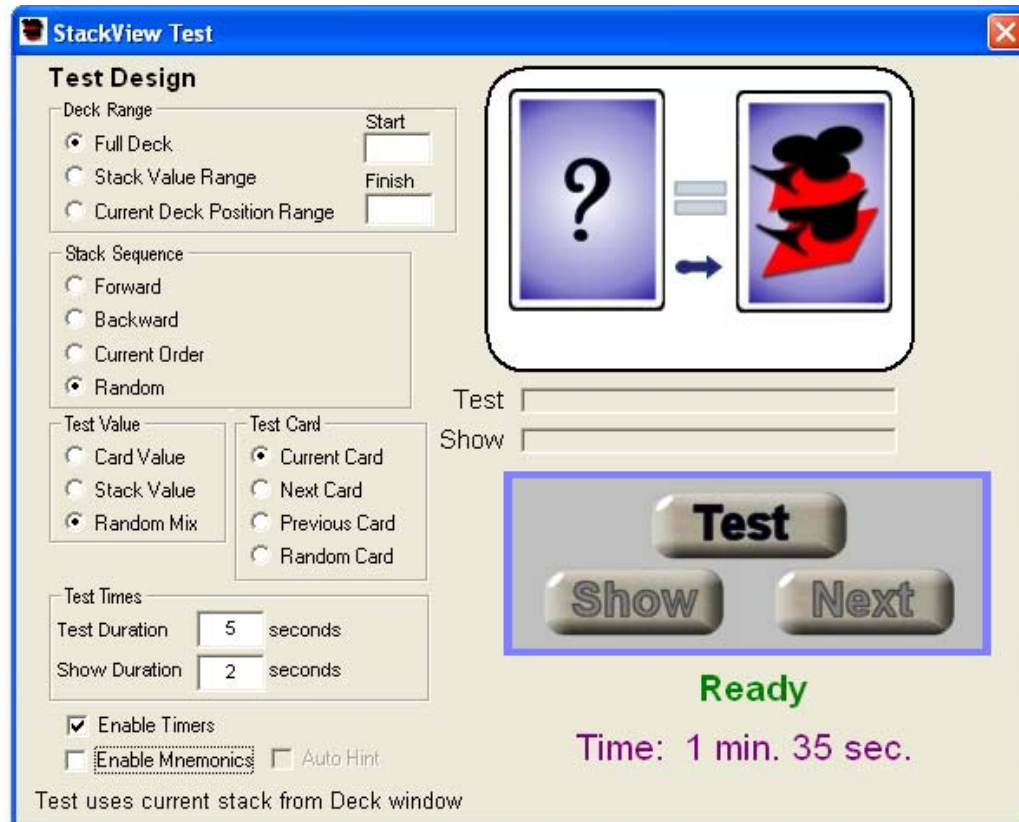


Figure 104

## **Reference: StackView Advanced Test**

As the name implies, StackView Advanced Test is for advanced users who already know their stack “cold.” It is designed to help you practice the “any card to any number” methodology with a great degree of control of how the test is conducted.

While the regular StackView Test module allows you to specify the number of cards that will be shown in the deck, StackView Advanced Test always tests you with 30 cards. At the end of the timed test, you will be given a performance report.

### ***Accessing StackView Advanced Test***

You access *StackView Advanced Test* from the *Tools* menu on the menu bar. When the *StackView Advanced Test* menu selection is checked, the *StackView Advanced Test* window is visible. When the *StackView Advanced Test* menu selection is unchecked, the window is hidden. When the *StackView Advanced Test* window is closed, the *StackView Advanced Test* menu item will become unchecked.

### ***Deck Order***

You may specify the deck order for the test by selecting one of the options. You also have the control of whether the deck is randomly cut before the next test cards are shown.

### **Starting Deck Order**

When the *Starting Deck Order* option is selected, the deck will begin in its original stack order for the first set of test cards. The order of the current deck in the *Deck* window will be ignored.

You may also place a check in the *Random Cut* checkbox. Doing so will cut the deck randomly before each new set of test cards. If you do not place a check there, all of the test cards will be chosen with the deck in *Starting Deck Order*.

### **Current Deck Order**

When the *Current Deck Order* option is selected, the deck will begin in its current order from the *Deck* window for the first set of test cards.

You may also place a check in the *Random Cut* checkbox. Doing so will cut the deck randomly before each new set of test cards. If you do not place a check there, all of the test cards will be chosen with the deck in *Current Deck Order*.

### **Known Card**

For the *StackView Advanced Test*, it is presumed that you know the identity of the Top or Bottom card of the deck. You must select one of the two options indicating your knowledge.

### **Top Card**

If you select the *Top Card* option, the *Top Card* of the deck will be shown to you during the test. The image of the *Top Card* will appear in the leftmost card picture. The label above the card will read "*Top Card*."

### **Bottom Card**

If you select the *Bottom Card* option, the *Bottom Card* of the deck will be shown to you during the test. The image of the *Bottom Card* will appear in the leftmost card picture. The label above the card will read "*Bottom Card*."

### **Desired Card**

The *Desired Card* is the target card in the "any card to any position" challenge. During the test, the image of the test card will appear in the second card position. You have two ways of specifying the *Desired Card*.

### **Random Card**

When you select the *Random Card* option, the *Desired Card* is randomly chosen for each set of test cards.

### **Specified Stack Value**

When you select the *Specified Stack Value* option, the *Desired Card* is the same card for each set of test cards. It is determined by the *Stack Value* that you enter in the text field provided.

### ***Desired Position***

The *Desired Position* is the target position in the “any card to any position” challenge. During the test, the image of the target position value will appear in the third card position. You have two ways of specifying the *Desired Position*.

### **Random Position**

When you select the *Random Position* option, the *Desired Position* is randomly chosen for each set of test cards.

### **Specified Position Value**

When you select the *Specified Position Value* option, the *Desired Position* is the same *Position Value* for each set of test cards. It is determined by the *Position Value* that you enter in the text field provided.

### ***Test Questions***

Given a set of test cards presented, there are three logical test questions that can be asked in the “any card to any position” challenge. You must select one of the question types for the *StackView Advanced Test*.

Once a *StackView Advanced Test* session is active, you may change the *Test Question* that will be prompted by selecting a different option.

**Cards to Cut**

When this question is selected, you must identify how many cards must be cut from the top to the bottom of the deck, so that the *Desired Card* becomes positioned to the *Desired Position* (with the *Known Card* starting in the appropriate position before the cut).

**New Top Card Stack Value**

When this question is selected, you must identify the *Stack Value* of the *Top Card* after the appropriate number of cards has been cut from the top to the bottom of the deck, so that the *Desired Card* becomes positioned to the *Desired Position* (with the *Known Card* starting in the appropriate position before the cut).

**New Bottom Card Stack Value**

When this question is selected, you must identify the *Stack Value* of the *Bottom Card* after the appropriate number of cards has been cut from the top to the bottom of the deck, so that the *Desired Card* becomes positioned to the *Desired Position* (with the *Known Card* starting in the appropriate position before the cut).

***Answer Input***

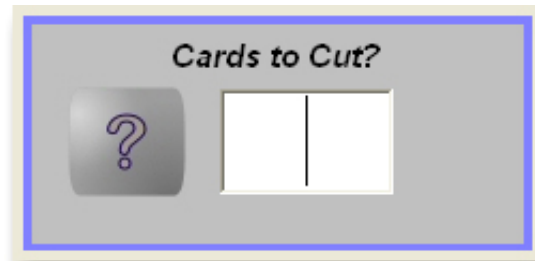
During the *StackView Advanced Test*, you must enter the correct answer into the *Answer Input* box.

**Answer Entry**

When a new set of test cards is being shown, the *Answer Input* box has a blank field with a flashing cursor and a grayed "question mark" button as shown in Figure 105.

Based on your selected *Test Question*, it will appear above the input field.

After you enter your numeric answer to the question, you must press the "Enter" key (or "Return" key).



**Figure 105**

### **Answer Response**

If your answer was correct, the gray "question mark" button will turn green and display the correctly entered number. For example, Figure 106 shows the *Response* to a correct answer given.

If your answer was incorrect, the gray "question mark" button will turn red and display the correct answer that should have been entered. For example, Figure 107 shows the *Response* to an incorrect answer given.

After the *Answer Response* is displayed, you may press the "Enter" key to immediately move on to the next set of test cards. This will help you develop speed with the advanced test.

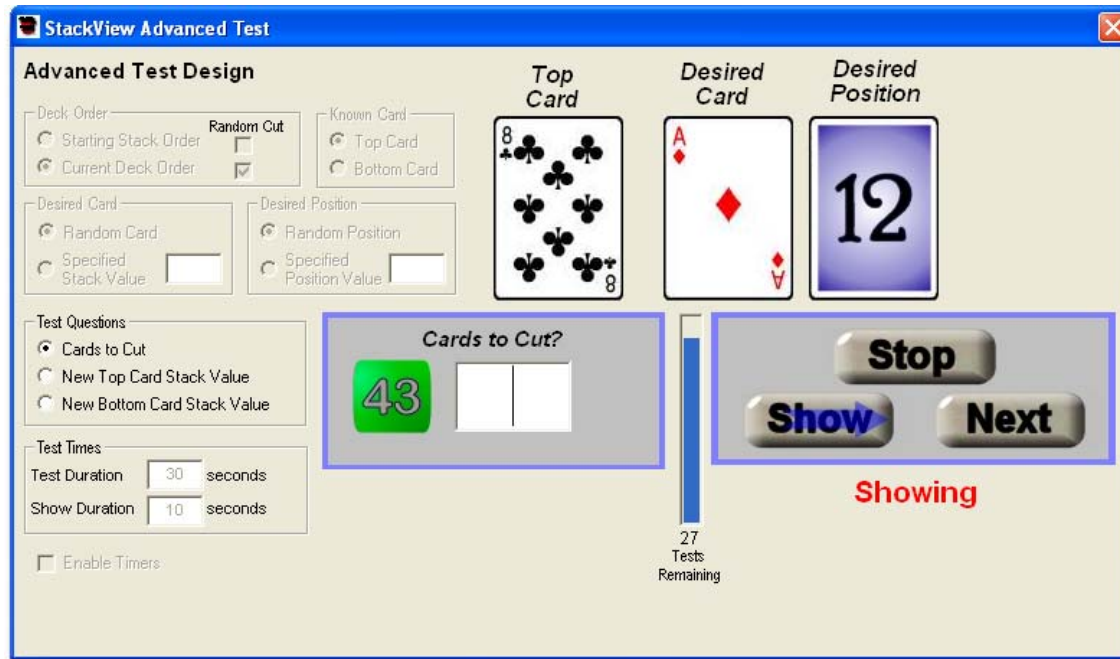


Figure 106



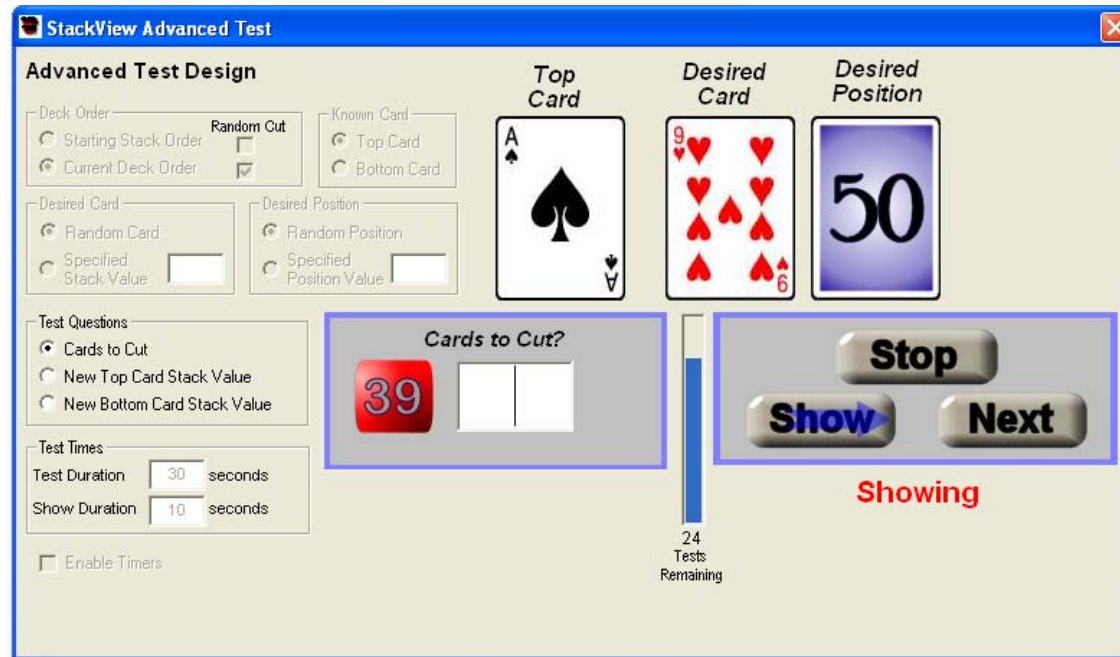


Figure 107

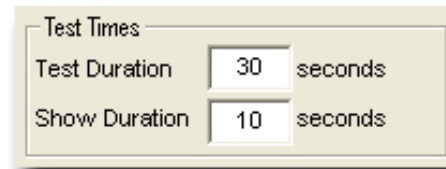
### ***Timers***

You can set the amount of time that you are tested on a set of test cards before you are shown the correct answer. You can also set the amount of time the answer is shown to you before you are automatically tested on the next set of test cards. You may also chose to not use the timers, and have the next set of test cards be prompted by you.

Once the test begins, you may not change the *Timer* settings.

## Durations

You enter the number of seconds you want for the timers in the *Test Duration* and *Show Duration* text input fields, as shown in Figure 108. You may enter any time between 0.1 seconds and 60 seconds. You may use decimal times such as 2.5 seconds. Any time entry will round to the nearest tenth of a second. For example, if you enter a *Test Duration* of 3.44 seconds, StackView will round it to 3.4 seconds.



The image shows a dialog box titled "Test Times" with a light beige background and a thin border. Inside the dialog, there are two rows of controls. The first row is labeled "Test Duration" on the left, followed by a small rectangular input field containing the number "30", and the word "seconds" to the right. The second row is labeled "Show Duration" on the left, followed by a similar input field containing the number "10", and the word "seconds" to the right.

**Figure 108**

The *Test Duration* is the amount of time that the test question is shown before the answer is automatically shown to you. The *Show Duration* is the amount of time that the answer will be displayed before StackView automatically moves on to the next test question.

During the test, two progress bars will activate beneath the pair of testing cards. The top bar corresponds to the *Test Duration*, and it will move from the left to the right as the test timer elapses, as shown in Figure 109. During this time, the test question is active. The bottom bar corresponds to the *Show Duration*, and it will also move from the left to the right as the show timer elapses, as shown in Figure 110. During this time, the answer is being shown.

As the timers cycle, the progress bars will indicate how much testing time has elapsed for each part of the test.

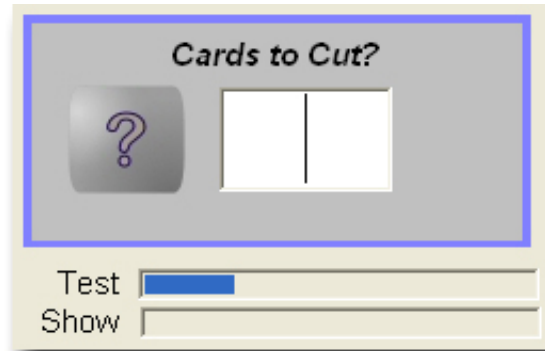


Figure 109

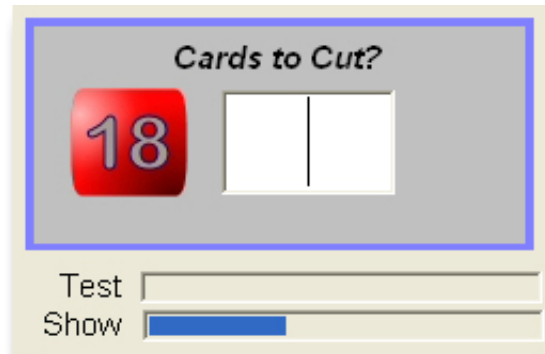


Figure 110

### Enable Timers

The *Enable Timers* checkbox is checked as the default setting. If you uncheck the box, the progress bars will disappear, and the timer inputs will be disabled. In addition, the test will not have any timers operating. You must press the *Next* button to advance the test to the next card. Finally, with the *Enable Timers* box unchecked, there will not be an elapsed time reported at the end of the test.

### ***Control Buttons***

Beneath the set of test cards are the *Control Buttons*. There are three buttons that are used to control the test flow. The buttons will dynamically change their appearance and function based on where you are in the testing sequence. When StackView is not testing, there will be a green "*Ready*" indicator below the control buttons as shown in Figure 111. When testing is in progress, the indicator below the buttons will read: "*Testing*" in red, as shown in Figure 112.

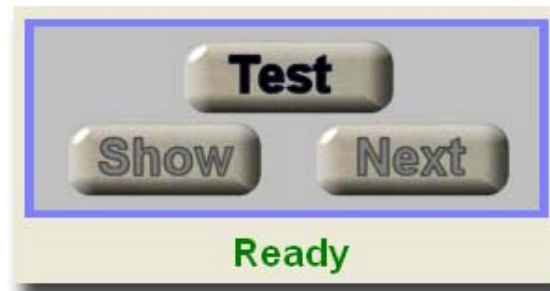


Figure 111

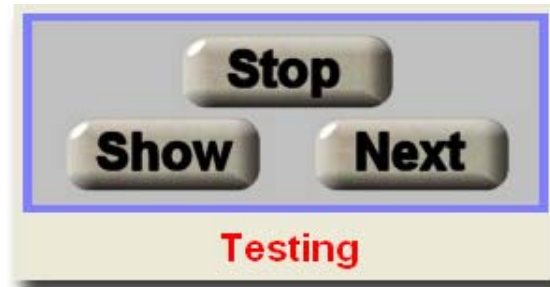


Figure 112

## Test

The top center button is the *Test/Stop* control. It is used to start and stop the test. When you first press the *Test* button to start the testing sequence, it will toggle to display "*Stop*," and the status indicator will read: "*Testing*." The lower right button will read: "*Start*," and the lower left button will indicate "*Show*" with the blue arrow overlay. Figure 113 shows what the buttons look like. At this stage, the test timer has not started counting yet. It will begin timing only after you press one of the two lower buttons.

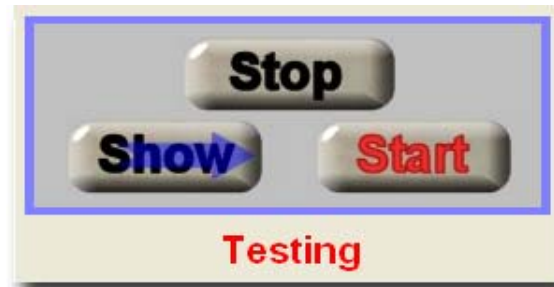


Figure 113

Once the test begins, the current question will be displayed only as long as the *Test Duration* setting permits (if the *Enable Timers* box is checked – page 183). If you do not press a button during the *Test Duration* opportunity, the answer to the question will be displayed only as long as the *Show Duration* setting permits. If you do not press a button during the *Show Duration* opportunity, the next question will be shown. This cycle will continue throughout the rest of the defined test.

If you press the *Stop* button at any time during the test, the whole test is disabled, and no elapsed time is reported. (If you want the test to complete with a reported elapsed time, you must end the test correctly by pressing one of

the lower buttons when the *Finish* button is showing, or by allowing the final *Show Duration* timer to finish.)

### Next

The lower right button is the *Next* control. It is used to advance the test to the next set of test cards without showing the answer to the current question (unless the answer is already showing).

The *Next* button will read: "*Start*" at the beginning of the test, as shown in Figure 113. Pressing it in this mode begins the test and the test timers. At the end of the test on the last card, the button will read: "*Finish*" as shown in Figure 114. Pressing it in this mode will end the test, stop the timers, and report the elapsed time (if the *Enable Timers* box is checked).

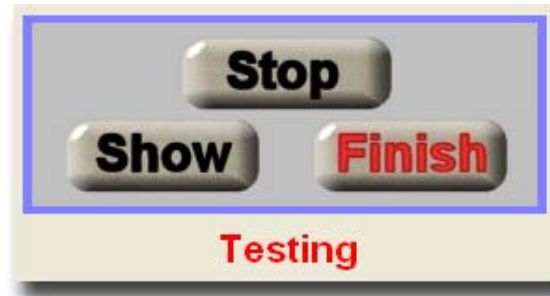


Figure 114

Throughout the rest of the test, the button will display: "*Next*." Pressing the *Next* button will advance the test to the next card question, and it will restart the timers as well as the progress bars. If you press the *Next* button during a *Test Duration* opportunity, StackView will advance to the next question without showing you the answer to the current question. If you press the *Next* button during a *Show Duration* opportunity, the test will automatically stop showing you

the current answer, and will advance to the next card question. It will also reset the timers and the progress bars.

Since *StackView Advanced Test* expects you to enter the correct answers, when you press the *Next* button, you will be attributed with a wrong answer.

### **Show**

The lower left button is the *Show/Next* control. It is used to show the answer to the current question, and also to advance the test to the next set of test cards. When the button just reads *Show*, pressing it will display the answer to the current question. When the *Show* button has a blue arrow overlaid on it, pressing the button will advance the test to the next card. The blue arrow points to the button to the right (*Next*), which indicates that the *Show* button has taken on the functionality of the *Next* button.

The *Show* button has two modes with separate functions. Depending on the current state of the test, the button will be in one of the two modes.

One of the modes reads: "*Show*" with a blue arrow overlaid on the button. In this mode, the *Show* button behaves exactly like the *Next* button. It will advance the test to the next card question as described in the preceding section. When it is pressed in this mode, it will toggle to the other mode (*Show* with no overlaid arrow).

When the *Show* button is in its regular mode (no overlaid arrow), pressing it will stop the current *Test Duration* timer, and the answer will immediately be displayed. It will also turn on the *Show Duration* timer.

At the beginning and end of the test, the *Show* button will have the blue arrow overlay. It can be used to start and finish the test (just like the *Next* button).

When you are still working on knowing your stack "cold," you will find the *Show* button useful to pace yourself and learn your stack better. You can keep your

mouse pointer over the *Show* button and focus your attention on the test cards. The one button will advance you through all steps of the test procedure. However, since *StackView Advanced Test* expects you to enter the correct answers, when you press the *Show* button, you will be attributed with a wrong answer, and the correct answer will be shown in a red button by the *Answer Input* field.

During the *StackView Advanced Test*, pressing the "Enter" key is equivalent to clicking the *Show* button.

### **Cards Remaining Progress Bar**

During an active test, there is a vertical progress bar to the left of the Control Buttons that indicates how many cards remain in the test, as shown in Figure 115.



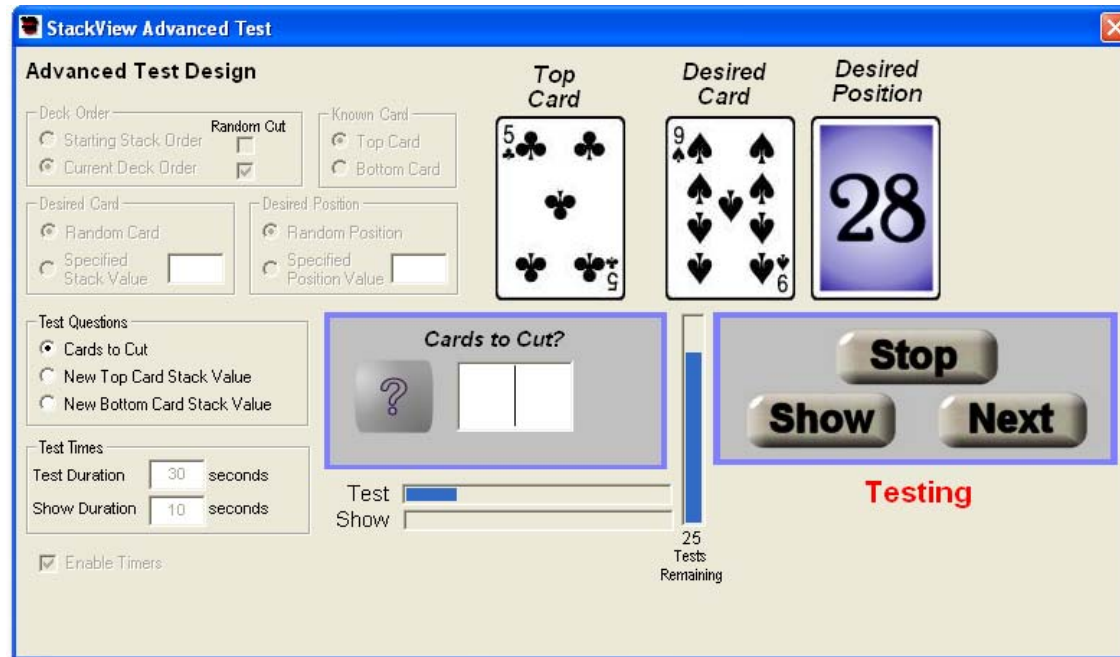


Figure 115

### ***Test Results***

When a timed test has completed, StackView will report the test results. total elapsed time for the test. The elapsed time will include both the *Test Duration* and the *Show Duration* times. The report is presented beneath the *Control Buttons* as shown in Figure 116.

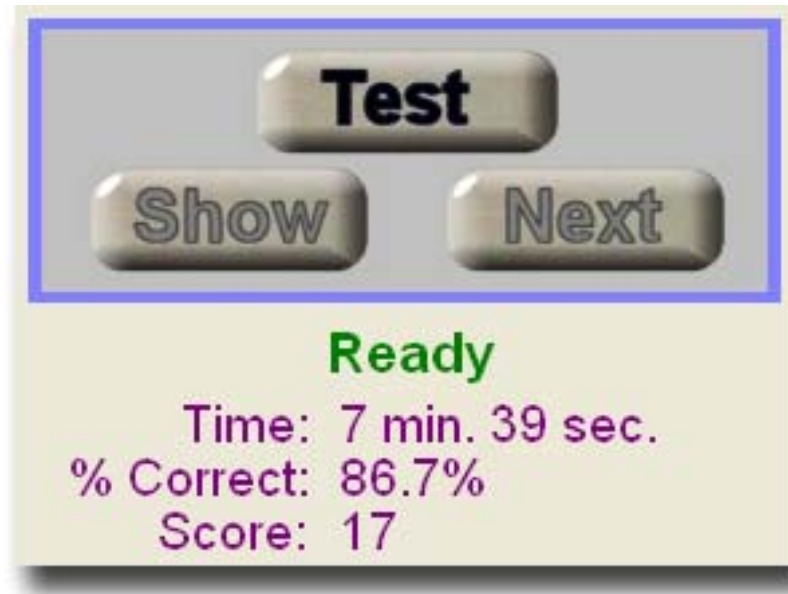


Figure 116

**Time**

StackView will report the total elapsed time for the test. The elapsed time will include both the *Test Duration* and the *Show Duration* times.

**Percent Correct**

StackView will report the percentage of questions that were answered correctly.

**Score**

The final score is a composite of the time and percentage answered correctly. The formula is:

Score = (# Correct \* 3 / seconds) \* 100

A score of 100 is excellent.

## Reference: StackView Search

*StackView Search* is a special tool which allows you to search for a sequence of manipulations that will get a deck from one known order to another known order. This is a very computationally intensive process, so some search sessions may take a long time to find a solution. However, it does provide the student of stacked deck magic a new tool for designing routines and exploring possibilities. The search feature allows you to limit the search space to tightly defined manipulation parameters, as well as allowing you to trap the results during a search if a close match is found.

With *StackView Search*, you set the starting deck and the target deck from saved deck files. You also specify whether the entire target deck must match, or only a specific range of card positions. You must also specify the maximum number of moves to try. Finally, you must identify the complete or specific set of manipulations to try per move.

As you identify the search parameters, the total possible search time is updated. In the most extreme condition, if you select all possible manipulations to try (16,071) and the maximum number of moves (26), you will see that the estimated time will be quite large (1.8 E +95 Millennia). This is not a joke. Doing a complete search of 26 possible moves with 16,071 possible manipulations per move, results in a very large number of combinations to search through. Due to this size problem, StackView allows you to specify fewer moves, and specific ranges of manipulations to try. This results in shorter possible search times.

During a search, you can continue to use all other parts of StackView, and it will not affect the search task.

When a match is found, the successful sequence of moves is recorded in the listbox that can then be transferred over to the StackView Sessions window for testing and saving.

### ***Deck Settings***

Before a *Search* can begin, you must specify several parameters. You must identify: a *Start Deck*, a *Target Deck*, the *Maximum Moves* to try, and the specific *Manipulations* to include in the *Search*.

#### **Start Deck**

Press the *Set Start Deck* button to select a saved deck file ( *.svf*) from your computer. This saved deck order will be the deck order that *StackView Search* will start from.

#### **Target Deck**

Press the *Set Target Deck* button to select a saved deck file ( *.svf*) from your computer. This saved deck order will be the deck order will be the deck order that *StackView Search* will try to match as the target.

#### **Deck Match Setting**

You must also specify whether the whole or partial *Target Deck* should be matched in the search. If you specify *Whole Deck Match* by selecting its option, the search process will try to match the entire *Target Deck order*. If you specify *Partial Deck Match* by selecting its option, you will be prompted to specify the *Starting* and *Ending* positions of the partial range. With the *Partial Deck Match* option selected, the search process will only try to match that partial deck range of the *Target Deck*.

#### **Threshold Trap**

You can also specify a *Trap* event by setting a threshold number of cards to match in the search progress. For example, even though you may want a whole

deck match, you may be interested in seeing a partial solution of 45 cards or more. You would then be able to use your own judgment to figure out the rest versus continuing a long search. In other words, for you, 45 cards may be “close enough.” If a *Threshold Trap* does occur, StackView will pause the search and show you the manipulation sequence. You can choose to stop or to resume the search.

When you press either of the *Target Deck* option buttons, a dialog box will open. If you check the *Threshold Trap* checkbox, you must also specify how many cards in the search range should match for the *Trap Event* to occur. The value you enter must be less than the number of cards being searched for in the range. For example, if you are doing a whole deck match, the *Threshold* value must be less than 52 cards. As another example, if you are doing a partial deck match with a range from 5 to 10, the *Threshold* value must be less than 6 cards.

When you specify a *Threshold Trap*, you must also specify whether you want the *Search* to be interrupted when a *Trap Event* occurs, or if you want the *Trap Events* to be saved to a special *Trap* file ( *.svt*).

When a *Trap Event* does occur with the interrupt setting, the *Search* will be interrupted, and the sequence of *Moves* that created the *Trap* condition will be listed in the listbox. Above the listbox will be the label *Threshold: X*, where *X* indicates the number of card matches that occurred. If the *Save Traps to File* setting is set, the *Threshold: X* label will be listed in the saved text file along with the accompanying sequence of *Moves*.

The purpose of the *Threshold Trap* feature is to incorporate some human judgment into the *Search* process. For example, even though you may want a whole deck match, you may be interested in seeing a partial solution of 45 cards or more. You would then be able to use your own judgment to figure out the rest versus continuing a potentially long search. In other words, for you 45

cards may be “close enough.” If a *Threshold Trap* does occur, StackView will show or save the results for you to study further.

There are two important things to remember when using the *Trap* feature. First, when a *Trap* does occur, StackView will suspend the search, and it will wait for your further input. So, if you want the search to run unattended, you must disable the *Trap* feature, or select the setting to save the *Threshold Traps* to a file. Second, with the *Trap* feature enabled, the search time will be slightly longer. There will be an approximate single digit percentage increase in the total search time.

### ***Manipulations and Moves***

You must specify the nature of the *Search* space by identifying the *Moves* and *Manipulations*.

#### **Maximum Moves**

Enter the value for the maximum number of *Moves* to try in the *Search*. The largest allowable value for this parameter is 26. The larger the value, the longer the possible search time.

#### **Identify Manipulations to Include in Search**

Each *Manipulation* that you want to include in the *Search* must be selected by checking one of its checkboxes. The left checkbox will include *All* possible variations of the *Manipulation*. The right checkbox allows you to narrow the *Specific* possible variations of the *Manipulation*.

You may clear all of the *Manipulation* checks by pressing the *Uncheck All* button.

#### ***All***

When you check the left box for a *Manipulation*, every possible variation will be tried. For example, if you select *All* for *Cut Deck Precise*, you will see the indicated value for *Manipulations per Move* increase by 51 *Manipulations*.

StackView will then include all 51 variations of *Cut Deck Precise* in the *Search* process. (A value of 52 is not included since that *Manipulation* does nothing to the deck, so StackView will not waste a processing cycle to test it.)

### ***Specific***

When you check the right box for a *Manipulation*, you will be prompted with range parameters to limit the number of *Manipulations*.

### ***Control Buttons***

#### **Starting the Search**

Once you have entered your search parameters, press the *Start* button to start the search process. After the search process begins, the *Start* button will turn in to a *Stop* button.

You may also first open a saved *Search File* ( *.svh* format, page 205) before pressing the *Start* button. When you open a saved *Search File* and press the *Start* button, the *Continue Search* button will appear for you to resume the search, or press *Start* to begin the search from the beginning.

#### **Interrupting a Search**

You may interrupt the search progress by pressing the *Stop* button at any time. The *Stop* button will become the *Start* button again, and a new button will appear labeled *Continue Search*. When you press the *Continue Search* button, the search process will resume from the point that it left off when you pressed stop. If you press *Start* while the *Continue Search* button is displayed, the current search progress will be cleared, and the search will start again from the beginning.

You can save your interim search results in a special *Search File* format. From the *File* menu, select the *Save Search As...* option to open up a dialog box. For more information see page 205.



If a Threshold Trap occurs, the search progress will be interrupted. See more information on page 197.

### ***Search Progress Indication***

During the search, there will be continuous updating of the expected time for completion.

### **Estimated Possible Search Time**

Based on the number of *Manipulations per Move* selected and the *Maximum Moves* indicated, there will be an *Estimated Possible Search Time* calculated and indicated.

### **Interim Results**

As the search is under way, a progress bar will continue to scroll across the screen indicating that the search is active. The first time the progress bar scans the screen will possibly be slower than subsequent scans. This is due to a speed calibration that takes place to adjust the *Estimated Possible Search Time* based on the speed of your computer. The estimated time will continue to recalibrate after each scan of the progress bar. If there are other activities that are taxing your processor, you may see some fluctuation in the estimated time.

After each scan of the progress bar, the list box will display the last move sequence that was tried. In addition, the *Elapsed Time* is updated and displayed, as well as the search progress. The search progress is shown in the long, narrow rectangle beneath the *Elapsed Time*. The format of the search progress is  $x[p, q, r, \dots]$ , where  $x$  indicates at which "move" the search is currently at. StackView tries all smaller combinations of moves first, and then searches upwards. For example, if you selected 7 as your *Maximum Moves* parameter, and several manipulations that result in 25 *Manipulations per Move*, StackView will first try all versions of 1 move (25 manipulations will be tried). If no match is found, then StackView will try all combinations (625 combinations) of 2 moves.

Then 3 moves (15,625 combinations), then 4 moves (390,625 combinations), and so on up to 7 moves (6,103,515,625 combinations). So, to continue this example, if the progress indicator shows 6[23,4,12,25,2,19], this means that the search has just tried a 6 move combination. The remaining numbers correspond to the manipulation identifiers as defined dynamically by your manipulation selections. The listbox shows the specific manipulations that are indicated by the values on the progress status line.

### ***Transfer List to Sessions***

Once a match is found, the successful sequence will be displayed in the listbox. You may transfer that sequence to the *Sessions* window by pressing the *Transfer List to Sessions* button.

## Reference: Menus

The StackView menus allow you to access and control all of the features in the program.

### *File*

The File menu selections allow you to save and open the various special StackView file structures. Each file structure has an unique file extension and is described more fully in Appendix 4 on page 269.

### **Open Deck...**

The *Open Deck* menu selection allows you to open a previously saved StackView deck file. StackView deck files have a `.svf` extension.

If the file was saved with StackView version 5.0 or later, the saved *Back Design* (see page 147) will also be restored when opened. If the file was saved with StackView versions earlier than version 5.0, the *Back Design* will be set to a default *Red Bicycle Rider* back.

### **Save Deck As...**

The *Save Deck As...* menu selection allows you to save the current deck (from the *Deck* window) to a StackView file format (`.svf`). It is important to note that both the sequence and the stack values of the cards are saved in the deck file.

For example, let's say you load in the *Nikola* stack using the *Set Stack* button. The cards will show in the *Deck* window, and the order of the *Stack Values* is sequential (1, 2, 3, 4, ... 50, 51, 52). Perform two *Reverse Out Faro* shuffles, and the new deck order will show the *Stack Values* as (1, 5, 9, 13, ... 44, 48, 52). However, if you save the current stack as `myNikola.svf`, when you open it later, the cards will be in the same sequence as when you saved them. The *Stack Values* will still read (1, 5, 9, 13, ... 44, 48, 52).

If you had wanted the new sequence (6D, 5S, 3C, 4D, 2D, ...) to match up with a fresh *Stack Value* order (1, 2, 3, 4, 5, ...) you would need to use the *Custom Deck* tool as described on page 234.

### **Save Deck As Default**

When a stack is saved as the default file, it will be the one that is opened automatically when the program starts up. The *Save Deck As Default* menu selection will always name the file as `stackview.svf`, and the file will always be saved in the same directory as the StackView application. (You can also use the *Save Deck As...* menu selection to save your stack with the `stackview.svf` filename, and save it to the correct directory instead. However, the *Save Deck As Default* option is easier.)

If the `stackview.svf` file is missing, the program will then open the *New Deck (Bicycle)* stack as the secondary default starting stack.

### **Open Session...**

The *Open Session...* menu selection allows you to open a previously saved *Session* file. StackView *Session* files have a `.svs` extension.

If there are existing *Session Events* in the *Sessions* listbox, they will be overwritten.

In StackView version 5.0, several of the command names have changed. As a result, some *Session* files from earlier versions will not run in version 5.0. (For example, the phrase "reversed" from earlier versions is now referred to as "inverse.")

### **Save Session As...**

The *Save Session As...* menu selection allows you to save the current *Session* (from the *StackView Control* window, *Sessions* tab) to a *Session* file format (`.svs`).

You do not have to stop recording your *Session* before you save it to a *Session* file.

### **Open Search...**

The *Open Search...* menu selection allows you to open a previously saved *Search* file. StackView *Search* files have a `.svh` extension.

If the *Search* file contains only parameter settings, and the *Search* process was not initiated, pressing the *Start* button will initiate the *Search*.

If the *Search* file contains an interrupted *Search* process, you will see both a *Start* button, and a *Continue Search* button. If you press the *Start* button, the *Search* will re-initiate and start from the beginning of the *Search* space. If you press the *Continue Search* button, the *Search* process will resume searching from where it left off.

If a *Search* file contains a completed *Search* and it had found a successful match, the label above the listbox will read `Match Found!!!` You will also see just the *Start* button. Pressing it will start the *Search* from the beginning again.

If a *Search* file contains a completed *Search* and it had exhausted all combinations and no found a successful match, the label above the listbox will read `No Match Found` You will also see just the *Start* button. Pressing it will start the *Search* from the beginning again.

StackView version 5.0 uses a revised `.svh` file structure. You can not open *Search* files saved with earlier versions of StackView.

### **Save Search As...**

The *Save Search As...* menu selection allows you to save the current *Search* activity (from the StackView *Search* window) to a *Search* file format (`.svh`).

The current status of the *Search* progress is saved so that when you open the file later, you can continue from where you left off.

### **Open Mnemonic...**

The *Open Mnemonic...* menu selection allows you to open a previously saved *Mnemonic* file. StackView *Mnemonic* files have a `.svm` extension.

The Mnemonic window will open with the mnemonic setting from the saved file.

### **Save Mnemonic As...**

The *Save Mnemonic As...* menu selection allows you to save the current *Mnemonic* set (from the *Mnemonics* window) to a *Mnemonic* file format (`.svm`).

### **Save Mnemonic As Default**

When a mnemonic file is saved as the default file, it will be the one that is loaded automatically when the program starts up. The *Save Mnemonic As Default* menu selection will always name the file as `stackview.svm`, and the file will always be saved in the same directory as the StackView application. (You can also use the *Save Mnemonic As...* menu selection to save your stack with the `stackview.svm` filename, and save it to the correct directory instead. However, the *Save Mnemonic As Default* option is easier.)

If the `stackview.svm` file is missing, the program will then load the *Aronson* mnemonic set as the secondary default.

### **Print...**

When you select the *Print...* menu item, the visual contents of the active window will be sent to your printer. A print dialog box will appear so that you can edit your printer settings.

## **Exit**

When you select the Exit menu item, you are requesting that the StackView program shuts down. You will always receive a confirmation dialog box before the software exits to prevent unsaved Search, Session and Mnemonic activities from being accidentally lost.

## ***View***

The *View* menu selections allow you to see specific windows and files. The *Deck* and *Controls* menu items will toggle the visibility of those two StackView windows. The rest of the *View* menu items will open specific StackView file types for your viewing in a pop-up window.

## **Deck**

The *Deck* menu item will toggle the visibility of the *Deck* window. When the *Deck* window is visible, there is a checkmark beside the *Deck* menu item. When the *Deck* window is not visible, there is no checkmark beside the *Deck* menu item. For more information on the *Deck* window, see page 24.

## **Controls**

The *Controls* menu item will toggle the visibility of the *StackView Control* window. When the *StackView Control* window is visible, there is a checkmark beside the *Controls* menu item. When the *StackView Control* window is not visible, there is no checkmark beside the *Controls* menu item. For more information on the *StackView Control* window, see page 32.

When you toggle the *StackView Control* window to the invisible state, any *Session* activities you have recorded will still remain in memory. When you toggle the visibility back, the *Session Events* will still be retained. However, if you close the *StackView Control* window by pressing the "X" in the upper right hand corner of the window, you will get prompted if you have any unsaved *Session* activities. If you close the *StackView Control* window without saving the

*Session Events*, you will lose them. For more information on Sessions, see page 80.

### **Deck File**

When you select the *Deck File* menu item, you will be prompted to open a saved StackView *Deck* file. These files have an extension of `.svf`. The file will be opened in a read-only text viewer. For more information on *Deck* file structure, see page 269.

### **Session File**

When you select the *Session File* menu item, you will be prompted to open a saved StackView *Session* file. These files have an extension of `.svs`. The file will be opened in a read-only text viewer. For more information on *Session* file structure, see page 270.

### **Mnemonic File**

When you select the *Mnemonic File* menu item, you will be prompted to open a saved StackView *Mnemonic* file. These files have an extension of `.svm`. The file will be opened in a read-only text viewer. For more information on *Mnemonic* file structure, see page 270.

### **Search File**

When you select the *Search File* menu item, you will be prompted to open a saved StackView *Search* file. These files have an extension of `.svh`. The file will be opened in a read-only text viewer. For more information on *Search* file structure, see page 270.

### **Threshold Trap File**

When you select the *Threshold Trap File* menu item, you will be prompted to open a saved StackView *Threshold Trap* file. These files have an extension



of `.svt`. The file will be opened in a read-only text viewer. For more information on *Threshold Trap* file structure, see page 271.

### ***Tools***

The *Tools* menu selections provide you with special features in StackView. Each menu item opens an unique window for its specialized functionality.

When a special Tools window is open, there will be a checkmark next to that menu item. If you select that menu item while it is checked, the corresponding window will close.

### **Custom Deck**

The *Custom Deck* menu item opens the interactive *Custom Deck* module. In this window you can customize your own stacks with complete control of the *Stack Values* and *Stack Positions* of each card. For more information, see page 133.

### **Custom Mnemonics**

The *Custom Mnemonics* menu item opens the *Mnemonics* module. In this window you can specify your own mnemonic scheme for your stack. Your scheme is then incorporated in the *StackView Test* module. For more information on the *Custom Mnemonics* module, see page 144.

### **Joyal ShuffleMeter**

The *Joyal ShuffleMeter* menu item opens the *Joyal ShuffleMeter* stack analysis tool. For more information on the *Joyal ShuffleMeter*, see page 148.

### **Piles Control**

The *Piles Control* menu item opens the full featured *Piles Control* module. With *Piles Control*, you have significant control over subsets of the deck, called *Piles*. For more information on the *Piles Control* module, see page 88.

### **Set Back Design**

The *Set Back Design* menu item opens the *Back Design* module. With this module, you can specify the design and the color of the back of the cards for your stack. For more information on the *Back Design* module, see page 147.

### **StackView Test**

The *StackView Test* menu item opens the full featured *StackView Test* module. With this module you can test your knowledge of your stack. For more information on the *StackView Test* module, see page 155.

### **StackView Advanced Test**

The *StackView Advanced Test* menu item opens the *StackView Advanced Test* module. This module provides a testing structure for expert stack users. For more information on the *StackView Advanced Test* module, see page 179.

### **StackView Search**

The *StackView Search* menu item opens the *StackView Search* module. The *StackView Search* module allows you to explore the combinatorics of manipulations to achieve a target stack sequence. For more information on the *StackView Search* module, see page 196.

### ***Window***

The *Window* menu items allow you to control the windows that may be open in your workspace.

### **Cascade**

The *Cascade* menu item creates a tight waterfall effect with all of the open windows in the application. This is a convenient way to organize your application space.

### **Open Window Selection**

Each open window's title will appear as a *Window* menu item. The active window's menu item will have a checkmark beside it. If you want to active any of the open windows to bring it to the top, just select that window's menu item from the *Window* menu selections.

### ***Help***

#### **Open User Guide**

The *Open User Guide* menu item will open this document with Adobe Acrobat Reader. You must have the Adobe Acrobat Reader installed on your computer. If you do not have it, you may download it for free from [www.adobe.com](http://www.adobe.com).

The StackView User Guide.pdf file must be located in the StackView application directory. (The default location is C:\Program Files\StackView.)

#### **About**

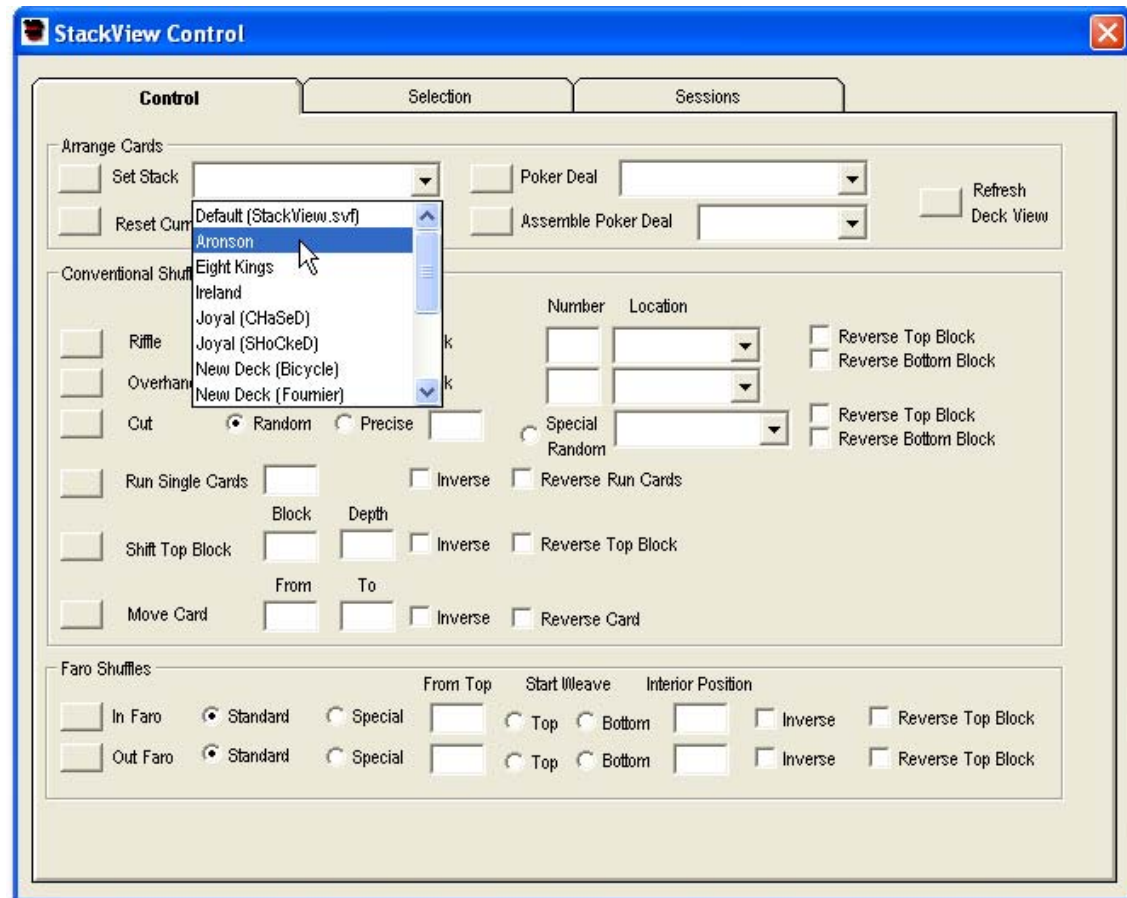
The *About* menu option provides some basic information about the StackView application.

## Appendix 1: Sample Exercises

### *Specifying the Order of a Stack*

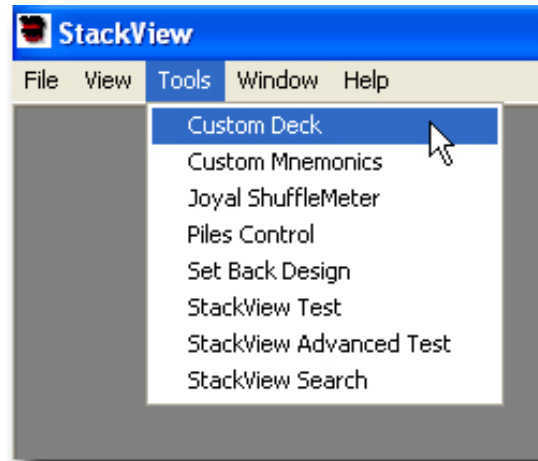
For this first exercise, we will explore how to take a given stack and set a new specific deck order without losing the original *Stack Values* of each card. For example, we will reorder the Aronson Stack into the Eight-Kings order, but still retain the Stack Values so that we can save the deck in the new order.

1. From the *Control* tab in the *StackView Control* window, select the *Aronson Stack* from the *Set Stack* listbox as shown in Figure 117, and then press the *Set Stack* button.



**Figure 117**

2. Open the *Custom Deck* window by selecting the Custom Deck choice on the Tools menu as shown in Figure 118.



**Figure 118**

3. Press the *Import Current Deck from StackView* button. This will place the Aronson Stack into the *Custom Deck* workspace.
4. Now press the *Reset: Retain* button. This will move the card indexes below the dividing line. You will notice that the *Stack Values* of each index also move below the line.
5. Next, you click and drag each card index into the correct new position above the line as shown in Figure 119 and Figure 120. You place each index in to the new order that you want. The *Stack Values* will move with each card.

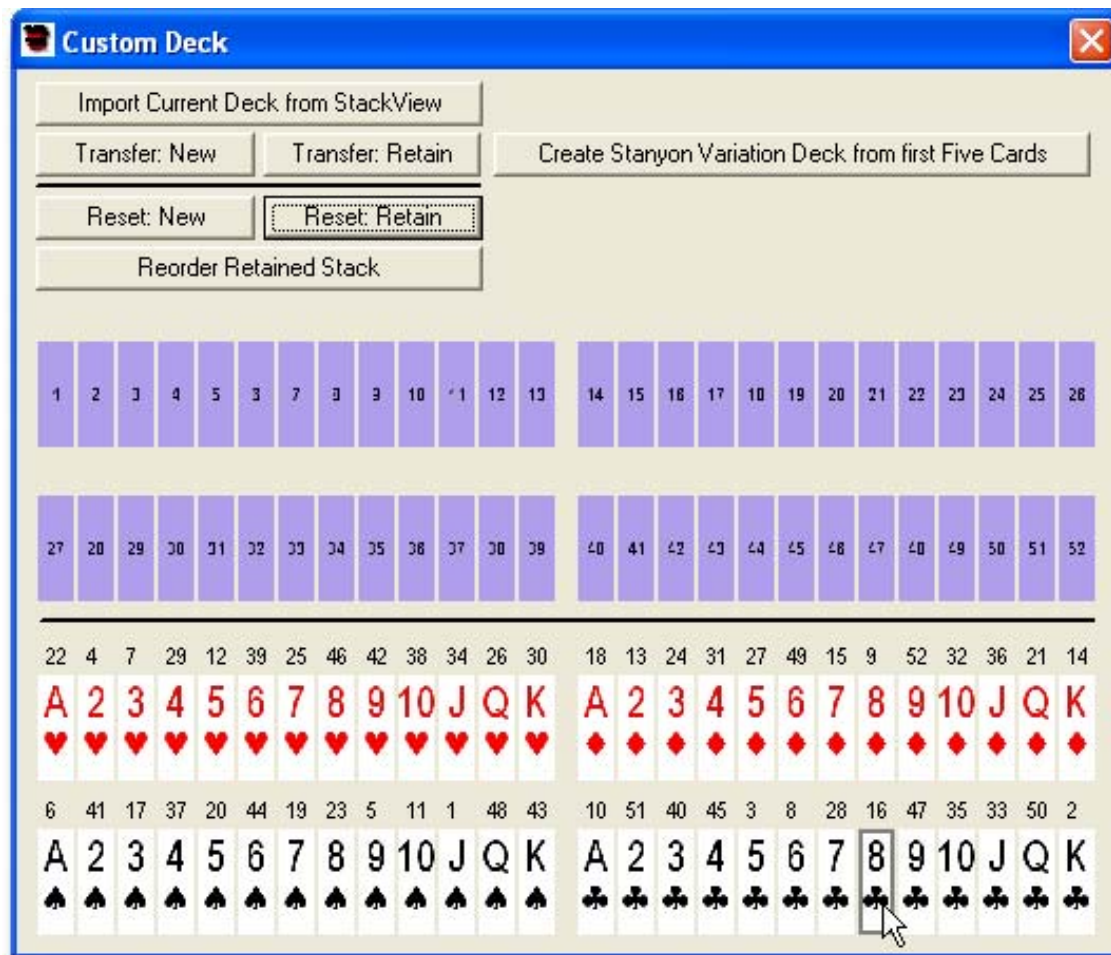


Figure 119

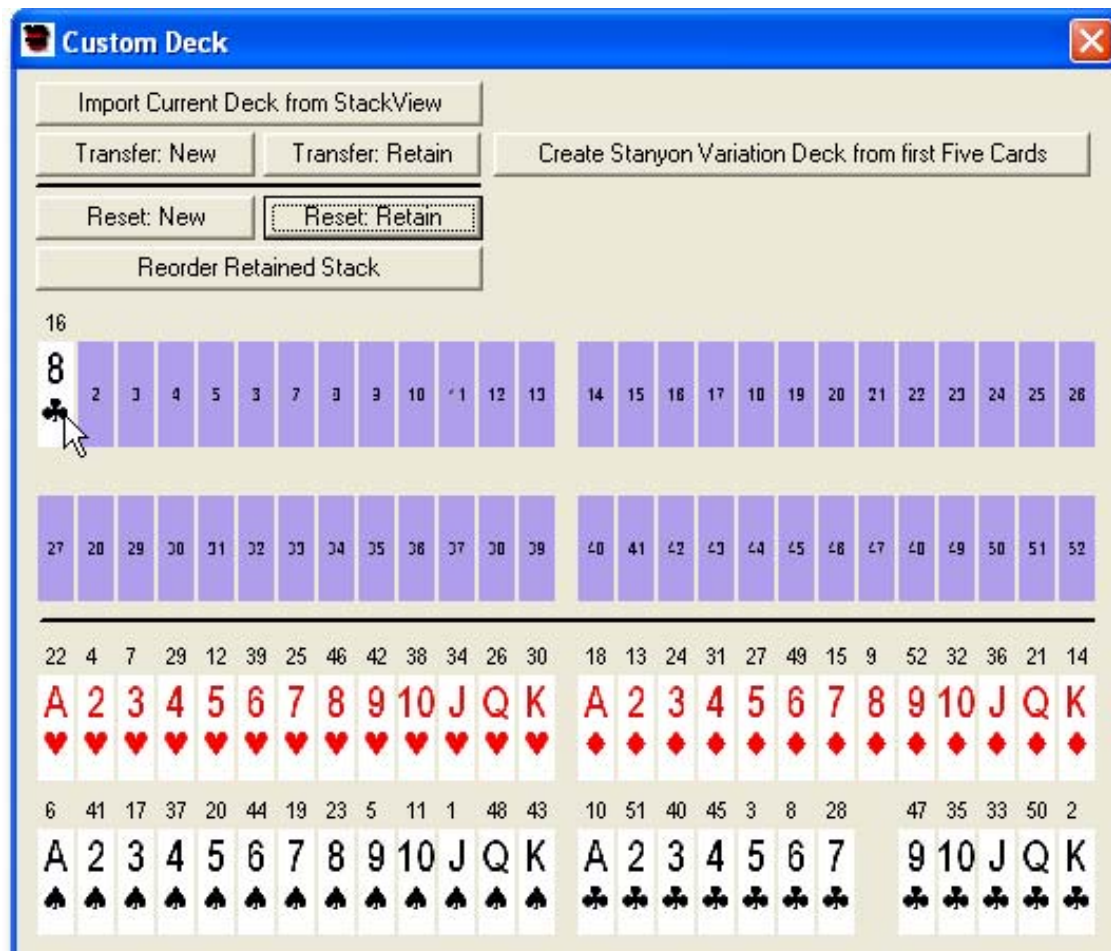


Figure 120



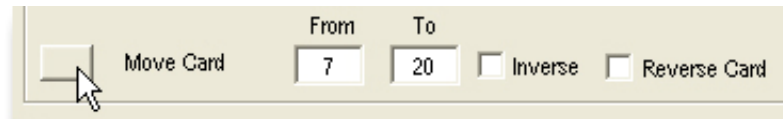
6. After all of the card indexes are above the line, press the *Transfer: Retain* button. This will place the new deck order and retained *Stack Values* into the *Deck* window.
7. Save the deck with the *Save As...* selection from the *File* menu.

You now have the Aronson Stack reordered to the Eight Kings sequence.

### ***Any Card to Any Number with Faros***

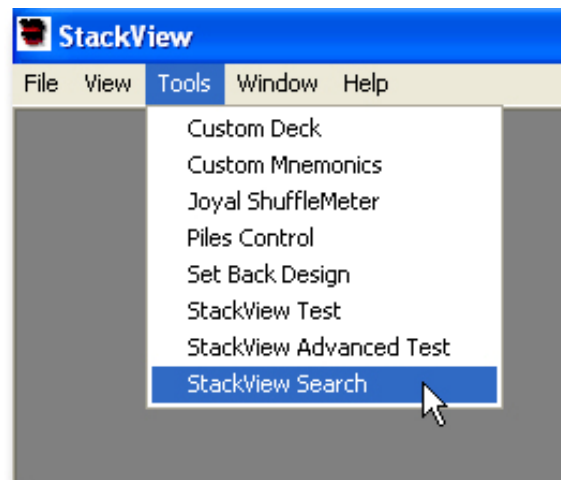
A classic exercise in card magic is to establish the correct sequence of Out Faros and In Faros to get a specific card from a starting position to a target position in the deck. For example, what is the Faro sequence to get the card in the 7<sup>th</sup> position to the 20<sup>th</sup> position in the deck?

1. Using the Search tool, this is a straightforward task. We will need to establish the correct starting target decks.
2. Start with the deck in any known sequence. In this example, we will start with the *Aronson Stack*. From the *Control* tab in the *StackView Control* window, select the *Aronson Stack* from the *Set Stack* listbox as shown in Figure 117, and then press the *Set Stack* button. Save the deck with the *Save As...* selection from the *File* menu, and name the saved file as `Start.svf`.
3. We now need to establish the target deck. The target deck will need to have the Three of Hearts (the 7<sup>th</sup> card in the *Aronson Stack*) repositioned to the 20<sup>th</sup> position. The easiest way to do this is with the *Move Event* from the *Control* tab in the *StackView Control* window. Set the *From* parameter to 7 and the *To* parameter to 20. Press the *Move Card* button as shown in Figure 121. Next, save the target deck with the *Save As...* selection from the *File* menu, and name the saved file as `Target.svf`.



**Figure 121**

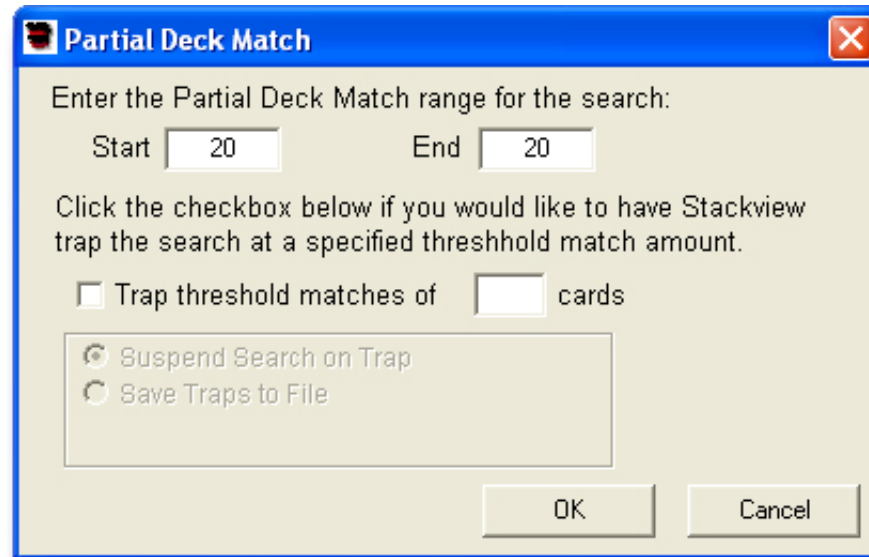
4. Open the StackView Search module from the Tools menu as shown in Figure 122.



**Figure 122**

5. Set the *Search* parameters for the task at hand. Set the start deck to `Start.svf`, which was saved in step 2. Set the target deck to `Target.svf`, which was saved in step 3. Select the *Partial Deck Match* option, and set both the *Start* and *End* range parameters to 20, as shown

in Figure 123. By setting the range to just this one position, all other card displacements can be ignored since we are interested in only the 20<sup>th</sup> position being matched.



**Figure 123**

6. Set the Maximum Moves parameter to 20. This will allow a generous sequence of moves for the search process.
7. Finally select the Out Faro and In Faro manipulations. Your StackView Search module should look like the one shown in Figure 124.

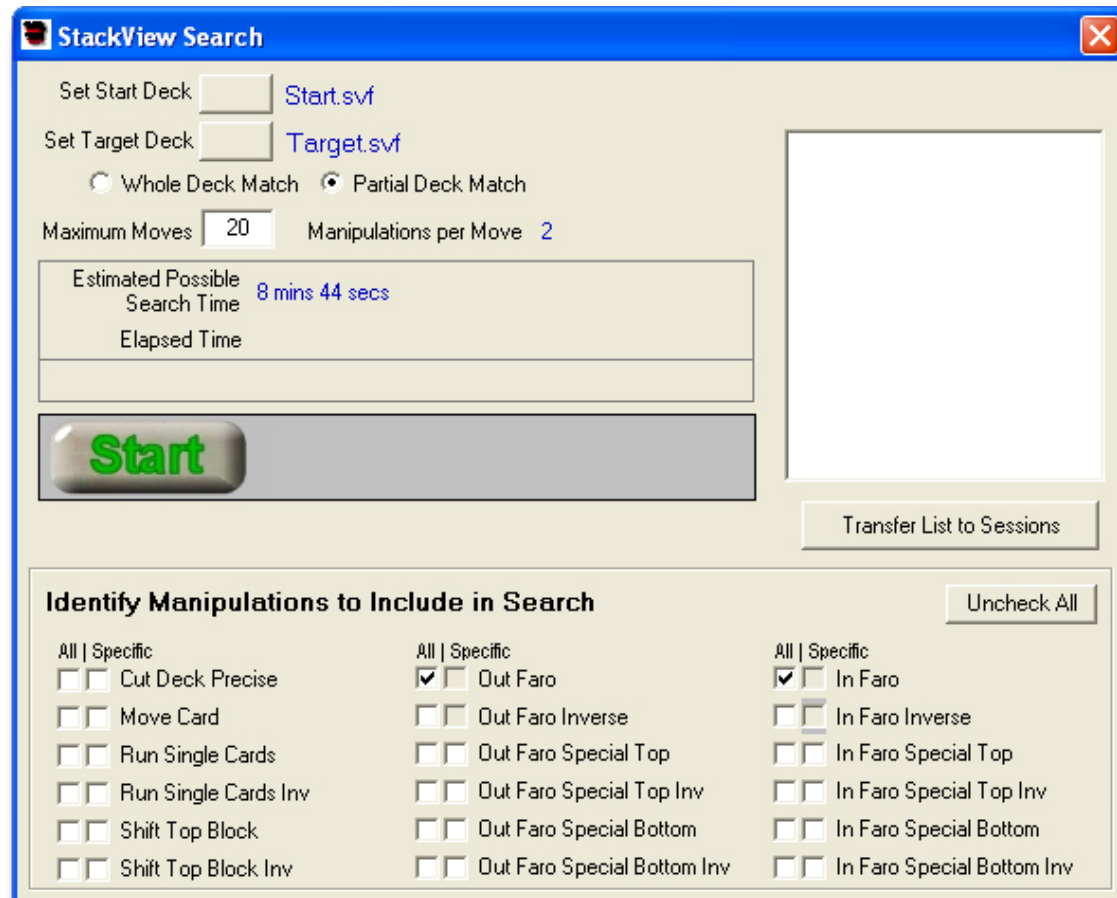


Figure 124

8. Press the Start button, and the solution will display in the listbox as shown in Figure 125. You may now transfer the Event list sequence to the

Sessions module by pressing the Transfer List to Sessions button. When you now run that sequence of Events, the 7<sup>th</sup> card will move to the 20<sup>th</sup> position.



Figure 125

### ***Working with Sessions***

For the example exploring *Sessions*, we will set up a practice session to locate a selected card after a specific selection and shuffle sequence. The deck will start in the Aronson Stack order. The spectator will randomly cut some cards to the table. The next card from the deck becomes the selected card, which is placed on the cut portion. The balance of the cards are placed on the selected card. The spectator then gives the deck a legitimate riffle shuffle. You are able to look through the cards and identify the selection.

We will use the *Sessions* tool to set up a practice sequence. Select the *Sessions* tab on the *StackView Control* window as shown in Figure 126.



Figure 126

1. Press the *Record* button. (Once you press it, the button changes to a *Stop* button.) When StackView is in a recording mode, all recordable *Event* buttons turn red.
2. From the *Control* tab in the *StackView Control* window, select the *Aronson Stack* from the *Set Stack* listbox as shown in Figure 127, and then press the *Set Stack* button. If you go back to the *Sessions* tab, you will see that you have recorded the first *Event* of the *Session* as shown in Figure 128.

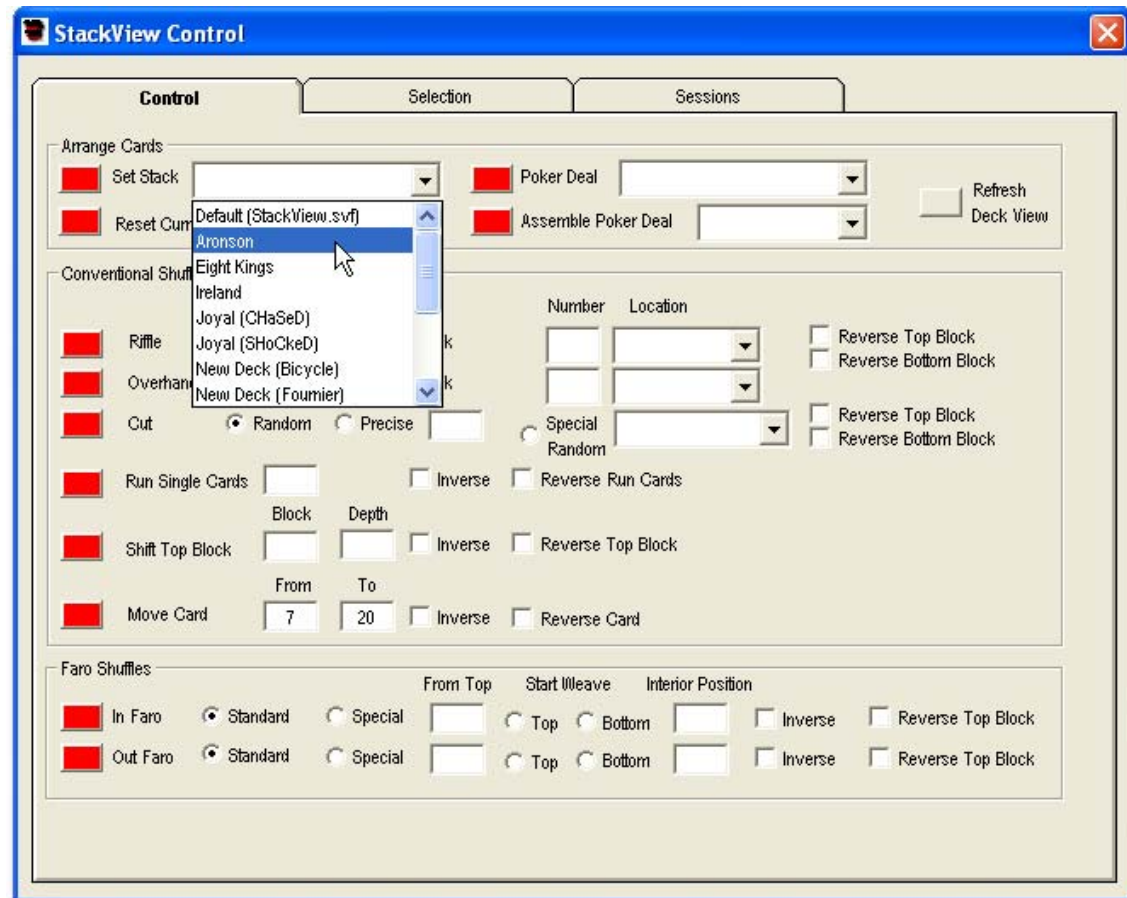


Figure 127



3. Select the *Selections* tab on the *StackView Control* window. In the *Free Choice with Special Handling* section, select the *Cut – Select Next 1* option from the listbox as shown in Figure 129. Next, press the *Select* button which is beside the listbox.



Figure 128

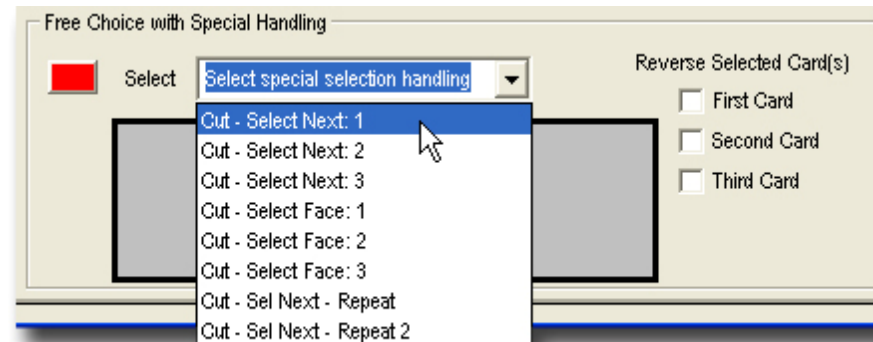


Figure 129

4. Select the *Control* tab on the *StackView Control* window.
5. With the *Random* parameter selected, press the *Riffle* shuffle button as shown in Figure 130.



**Figure 130**

6. Your *Deck* window should look something like Figure 131. In order to practice this routine, you will most likely not want the selected card to be highlighted. Chose the *Selections* tab of the *StackView Control* window. Uncheck the *Highlight Selections* box as shown in Figure 132.
7. Next, select the *Session* tab on the *StackView Control* window. Press the *Stop* button to complete your recording session. Your *Session* window should look like Figure 133.
8. From the *File* menu, select the *Save Session As...* menu item and enter a filename to save your session.
9. Now, to practice, simply press the *Play ALL Current Session* button, and the *Events* will all play in the correct sequence so that you can study the resulting deck order.

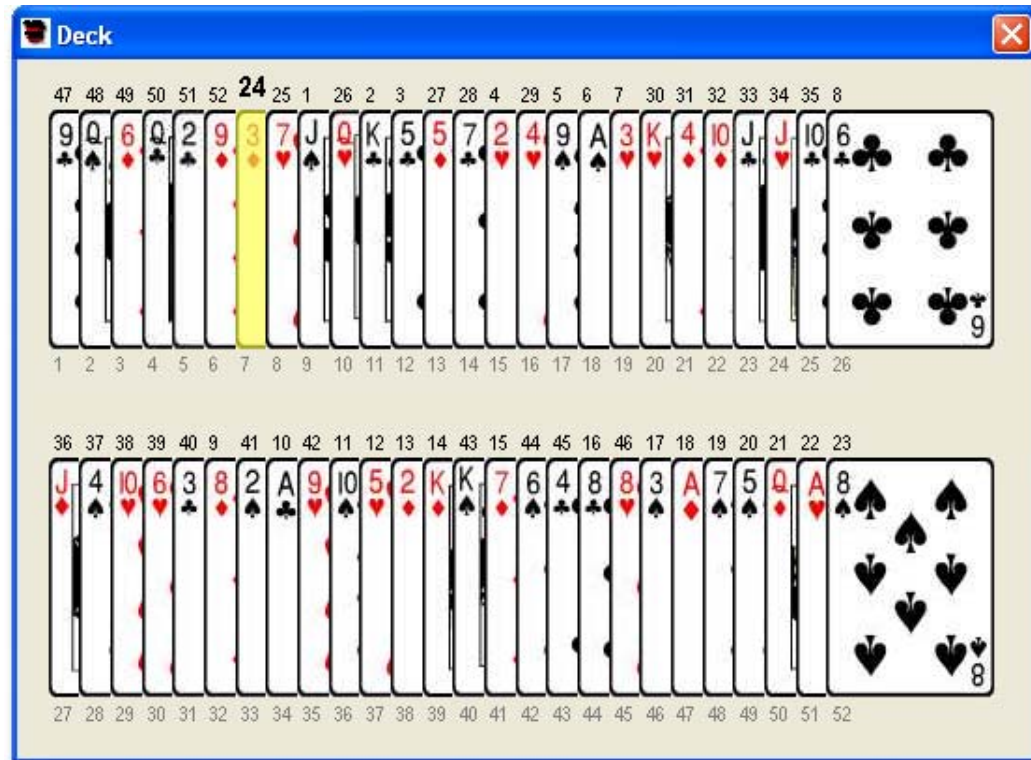


Figure 131

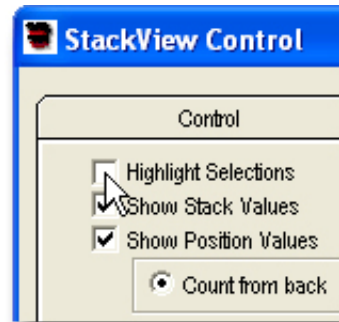


Figure 132



Figure 133

## **Appendix 2: Frequently Asked Questions**

### ***Who do I contact if I have specific questions?***

You can contact Nick Pudar at [nick@stackview.com](mailto:nick@stackview.com) with any questions, ideas, complaints, recommendations, etc. Visit [www.stackview.com](http://www.stackview.com) for the latest information and versions of the software.

If you would like to be added to a mailing list to be informed of updated releases of StackView, send an email requesting to be added to the list to [nick@stackview.com](mailto:nick@stackview.com). The email will not be used for any other purpose other than informing you of a software update.

### ***The program crashed. What now?***

While I have made every effort to anticipate all error conditions and provide a soft landing, I do realize that I may have overlooked some situations. If StackView does crash on you, please let me know by sending me an email to [nick@stackview.com](mailto:nick@stackview.com). If you can also provide any details about your computer, and what you were trying to do before the crash, it would be most appreciated. If the crash event is repeatable, that is even more helpful in the debugging process.

### ***I think I found a bug in the software. The results don't seem right. What do I do?***

If you believe the software is not behaving correctly, I would appreciate hearing about it. I have made every effort to fully test all aspects of the software simulations. However, I do recognize that there are a lot of complex situations that I may not have tested. If you do find something that does not seem right, please send me the details to [nick@stackview.com](mailto:nick@stackview.com).

### ***How do I learn more about stacked deck magic?***

There are many publications written about stacked deck magic. An extremely well documented internet site, [www.joyalstack.com](http://www.joyalstack.com) by Martin Joyal provides a wealth of material and references for the serious student of stacked deck magic.

### ***In the Custom Deck window, why don't the cards drag-and-drop correctly sometimes?***

When you are dragging a custom card index, it will be dropped onto the space that is precisely designated by the tip of the mouse arrow. The tip of the arrow must be on one of the gray card slots to be dropped correctly. If the tip of the arrow happens to be between card slots, it will default back to its starting position below the dividing line. For more detailed information about the *Custom Deck* interface, see page 133.

### ***Why is everything so big on my screen?***

If the windows appear too large for your monitor, your screen resolution may be set too small. For me personally, the optimal resolution is 1024 x 768. However, depending on your computer hardware, you may find your own preferred screen resolution.

To adjust the screen resolution go to your *Control Panel* in *Windows*, and adjust your *Display Settings*. Another way is to right-click on your desktop, and select *Properties*. On the *Settings* tab, there is a slide bar to adjust the *Screen Area*.

### ***When will an Apple Mac or a Pocket PC version be written?***

I have no plans to develop an Apple version of StackView. However, if anyone is SERIOUSLY contemplating taking my source code and converting it to the Apple O/S, please contact me at [nick@stackview.com](mailto:nick@stackview.com).

I have been told that Windows emulators should be able to run StackView on a Mac.

Likewise, I have no plans on developing a PocketPC version of StackView. I can't imagine what it look like on such a small screen.

Please keep in mind that I wrote this while I was learning the programming language (Visual Basic 6.0) in a "just-in-time" mode. The result is a lot of hard-wired and inelegant code. Also there is NO documentation to explain what is going on in the code. (This fact is troublesome at time when I try to figure out how the software is working!)

### ***How do I create my own stacks?***

In order to create your own stack, here are the essential steps:

1. Select the *Custom Deck* module from the *Tools* menu.
2. Drag-and-drop the card indexes from below the dividing line to the open card slots above the line until all 52 cards are positioned in your stack order.
3. Press the *Transfer: New* button. This will transfer the new stack order to the *Deck* window. (You may need to make the *Deck* window visible by selecting it from the bottom of the *Windows* menu.)
4. Save your stack by selecting *Save Deck As...* from the *File* menu on the menu bar. You will be prompted for a filename and location to save the stack to. The file extension for saved stacks is .SVF (StackView Format).
5. You now have your own custom stack.

For more information on custom decks, see page 133.

### ***How do I make a specific stack be the default stack that StackView uses when it loads?***

If you would like to set any particular stack as the default stack to load when the program starts, you need to save the stack as the default stack. The default

stack must be named `stackview.svf` and it must be saved in the same directory as the StackView program. There are two ways to do this:

1. The easiest way to create a default file is to select *Save Deck As Default* from the *File* menu in the menu bar. This will create the `stackview.svf` file using the current deck and order from the *Deck* window. It will save the file to the correct directory automatically. When you open the software later, it will automatically load the correct deck in the same order as when it was saved as the default.
2. The other way to create a default stack is to select *Save Deck As...* selection from the *File* menu on the menu bar. When prompted for the filename, enter the name `stackview` and make sure to save it in the same directory as the StackView program.

For example, let's say that you want the Aronson Stack to always load when StackView opens. Follow these instructions:

1. Open StackView
2. Select Aronson from the *Set Stack* drop down box (*StackView Control* window, *Control* tab).
3. Press the *Set Stack* button to load the Aronson Stack.
4. Select *Save Deck As Default* from the *File* menu on the menu bar.

You have finished – the next time you start up StackView, the Aronson Stack will automatically load for you.

### ***One (or more) of the windows is missing from the screen. Where are they?***

There are four possible reasons why one or more of the windows are missing on your screen:



1. One or several of the windows have been inadvertently closed during operation of the program. In this case, select the *View* menu from the menu bar and/or the *Tools* menu, and then select the window that appears to be missing.
2. One or several of the windows may have been un-checked from the *View* and/or *Tools* menu on the menu bar. In this case, select the *View* or *Tools* menu from the menu bar, and then select the window that appears to be missing to re-check the selection.
3. Based on your screen resolution, one of the windows may still be open, but out of range of your visibility. In this case, from the *Window* menu on the menu bar, select *Cascade*. This will return the windows to the visible range. If you are still missing windows, see items 1 and 2 above.
4. Finally, one of the windows may be behind another window. Cascading will help this condition, but you can also directly select that window from the *Window* menu on the menu bar. In this case, it will bring that window to the front of the stack of windows, and make it the active window.

### ***What are all the special file formats that StackView uses?***

StackView uses five special file formats to save different elements of the program interaction. For specific information about each of the file formats, see Appendix 4 on page 269.

- *StackView Deck Format* (.svf) files store the *Stack Values* and the order of a deck of cards. The back design and color of the deck is also saved in the file.
- *Session Format* (.svs) files store the *Event* sequences of your *Session* so that they can be accessed later and played.
- *Mnemonic Format* (.svm) files store your custom mnemonics assignments.

- *Search Format* (.svh) files store all of the interim parameters of an interrupted *Search* activity. The parameters can be opened later to resume the *Search*.
- *Threshold Trap Format* (.svt) files store the interim *Search Threshold Trap* results.

### ***What is a Threshold Trap?***

The purpose of the *Threshold Trap* feature is to incorporate some human judgment into the search process in order to minimize the total search time. Some searches can take a very long time. You may be interested to know when a search sequence gets “close” to the desired conclusion. You get to assign how many cards are “close enough” for your purposes so that you can add in some of your own judgment to complete the task. For detailed information in using the *Threshold Trap* settings, see page 197.

### ***How do I reset/re-assign the stack values without disturbing the current deck order?***

After manipulating the deck, you may want to establish that deck as your stack, and you want the *Stack Values* to reset themselves (1, 2, 3, ... ..50, 51, 52). You can do this by performing the following sequence:

1. When the deck is in the sequence you want, open the *Custom Deck* module from the *Tools* menu.
2. Press the *Import Current Deck from StackView* button.
3. Press the *Transfer: New* button.

Now the *Stack Values* have been reset (1, 2, 3, ... ..50, 51, 52).

### ***Is StackView Search useful?***

The *Search* feature is useful for targeted exploration. Given the vast number of possible combinations of manipulations, the search time can be essentially infinite relative to our lifetimes. This is true for an open-ended unconstrained search. Computing power is nowhere near where it needs to be for these needs. However, with a reasonably constrained problem, *StackView Search* can be very valuable in identifying the shortest path to a desired outcome.

At one point, I (very briefly) thought about setting up a SETI-like website that would let any interested magicians share their unused computer processing time. The website would parse out portions of the search space to find out how to get to the Aronson Stack from a New Deck (Bicycle) order. However, a quick back-of-the-envelope calculation suggested these efforts would be futile, as it would still take vast millennia even if every human alive had a computer dedicated to a portion of the search. Oh well, at least we have the comfort knowing that when parallel processing quantum computers (people are actually working on this!) become available, *StackView Search* will take advantage of the speed.

So, to make *StackView Search* practical today, do the following:

1. Focus on very specific desired outcomes (partial ranges).
2. Only include manipulations that you would actually do. While StackView can include a *OutFaroSpecialBottom(12, 9)* in the search, I'm fairly confident that you wouldn't want to have to actually do it in a routine.

### ***Are the StackView Test timers accurate?***

Yes, the timer is accurate. It correctly reports the elapsed time from when you first press *Start* until you press *Finish*. However, someone may notice that if you set the *Test* and *Show* times, and then let the test run out on its own (for a full deck), the elapsed time will be slightly longer than it should be. The actual reported test time will accurately reflect the total elapsed time. However, the

apparent extra time reflects the additional processing the software is doing that is not part of the *Test* or *Show* times. This extra overhead time is a small fraction of a second per card in the deck.

If you are actually testing yourself, this minor timing issue will in no way interfere with your real test times.

### ***Why are there "Inverse" parameters on many of the Events?***

The inverse commands help you see what the order of the deck was before an *Event* occurs. The inverse option is available on many *Events* throughout StackView.

In earlier versions of StackView the inverse feature was referred to as "reverse." However, in Version 5, the ability to reverse (turn over) cards in the deck required the shift in nomenclature.

### **Example 1**

You want to see what the order of the deck is before an *OutFaro* shuffle. You simply do an *InverseOutFaro* shuffle by checking the Inverse box of the *OutFaro* parameters.

### **Example 2**

In a more detailed example, suppose you wanted to construct a series of controlled shuffles that included single card runs and a couple of Out Faro shuffles. At the end of the shuffle sequence, you want the deck to finish in the *Joyal (CHaSeD)* order. The desired shuffling sequence is:

1. Run 5 single cards, throw rest of deck on top
2. Run 4 single cards, throw rest of deck on top
3. Do two Out Faro shuffles
4. Run 3 single cards, throw rest of deck on top

You would like to know what the starting deck order should be so that after the above sequence of shuffles, the deck is in the Joyal Stack order.

To do this, you must do the sequence backwards. Perform the following:

1. Set the deck in *Joyal (CHaSeD)* order
2. Press the *Run Single Cards* button, with 3 in the input box and the *Inverse* box checked
3. Press the *Reverse Out Faro* button (*Standard* option selected) two times
4. Press the *Run Single Cards* button, with 4 in the input box and the *Inverse* box checked
5. Press the *Run Single Cards* button, with 5 in the input box and the *Inverse* box checked

The deck is now in the correct order so that after you perform the intended sequence of shuffles, the deck will finish in the desired Joyal Stack order.

### **Example 3**

You would like to know what position in the deck a card must be so that after three In Faro shuffles, the card will land in the 37<sup>th</sup> position.

To illustrate how to answer this question, set your deck to the *New Deck (Bicycle)* order. Note that the 37<sup>th</sup> card is the Three of Diamonds. Now do three *Inverse In Faros* (*Standard* option selected). Note that the Three of Diamonds is now in the 51<sup>st</sup> position. Now you know that after three In Faro shuffles, the 51<sup>st</sup> card will relocate to the 37<sup>th</sup> position.

### ***Are there version incompatibilities?***

In Version 5 of StackView, there are incompatibilities in the *Deck Format* files (.svf) and Session Format files (.svs).

For the *Deck* files, Version 5 added an additional parameter to designate the desired back design and color. Version 3 and later of StackView *Deck* files will open in Version 5. However, Version 5 *Deck* files will not open in any earlier version.

For the *Session* file, there is no compatibility (forwards or backwards) between Version 5 and earlier versions. If you have *Session* files from earlier versions, you will have to re-record them, or edit them using the detailed *Session Command* details found in the next Appendix.

## Appendix 3: Session Commands

This section presents the detailed command structure that is expected in the Sessions module. When you record your Sessions, each Event and its specified parameters are properly represented in the command structure. The parameter codes are not always intuitively obvious; this Appendix can be used to interpret the details of the Session file. You can also use these details to construct your own Session file as an ascii text file if you wish.

The parameters will typically have constraints. In most cases, these are obvious, such as the number of cards in the deck or in any given *Pile*. However, there may be some subtle constraints that will get reported by StackView when you try to run the *Session*.

In this section, each *Event* is represented by a name such as:

**EventName(p1, p2,... pN)**, where pX indicates the parameters for the *Event*.

The table that appears beneath the Event name provides all of the parameter options.

### AssemblePokerDeal(p1)

<b>p1</b>	Text	Specifies the re-assembly order of Poker deals
	Backwards	Re-assembles the hands from right to left
	Forwards	Re-assembles the hands from left to right
	Unwind	Unwinds the Poker deal so that the cards are in the same order as they were before the deal

### AustralianDeal(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11)

<b>p1</b>	Text	Specifies whether to start the deal Down or Under, and also which Pile to apply the deal to
	Dx	Start the Deal with a Down card with Pile x
	Ux	Start the Deal with an Under card with Pile x
<b>p2</b>	Numeric	Specifies the exact number of cards for the Down deal
	x	$0 < x \leq$ number of cards in Pile
<b>p3</b>	Numeric	Specifies the exact number of cards for the Under deal
	x	$0 < x \leq$ number of cards in Pile
<b>p4</b>	Numeric	Specifies the limit of random cards for the Down deal
	x	$0 < x \leq$ number of cards in Pile
<b>p5</b>	Numeric	Specifies the limit of random cards for the Under deal
	x	$0 < x \leq$ number of cards in Pile
<b>p6</b>	Text or Numeric	Specifies either Selected card reversal or All Down cards
	S	Specifies that only the Selected card is reversed
	0	No Down cards are reversed
	1	All Down cards are reversed
<b>p7</b>	Numeric	Specifies whether all Under cards are reversed
	0	No Under cards are reversed
	1	All Under cards are reversed
<b>p8</b>	Numeric	Specifies whether random Down cards are reversed
	0	No random Down cards are reversed
	1	Random Down cards are reversed
<b>p9</b>	Numeric	Specifies whether random Under cards are reversed
	0	No random Under cards are reversed
	1	Random Under cards are reversed
<b>p10</b>	Text or Numeric	Specifies which card is Selected
	F	Final Down card is the Selected one
	R	Random Down card is the Selected one
	0	No card is Selected
<b>p11</b>	Numeric	Specifies if the Australian Deal is regular or Inverse
	0	Regular deal
	1	Inverse deal



### CombinePiles(p1, p2)

<b>p1</b>	Text and Numeric	Specifies Primary Pile conditions
	x	0 < x <= Number of Piles Pile x is on bottom of combined Pile and not reversed
	xR	0 < x <= Number of Piles Pile x is on bottom of combined Pile and is reversed
	Tx	0 < x <= Number of Piles Pile x is on top of combined Pile and not reversed
	TxR	0 < x <= Number of Piles Pile x is on top of combined Pile and is reversed
<b>p2</b>	Text and Numeric	Specifies Secondary Pile conditions
	x	0 < x <= Number of Piles Pile x is on bottom of combined Pile and not reversed
	xR	0 < x <= Number of Piles Pile x is on bottom of combined Pile and is reversed
	Tx	0 < x <= Number of Piles Pile x is on top of combined Pile and not reversed
	TxR	0 < x <= Number of Piles Pile x is on top of combined Pile and is reversed

### CreatePilesCut(p1, p2, p3, p4, p5, p6, p7, p8, p9)

<b>p1</b>	Numeric	Specifies the number of Piles to cut
	X	$1 \leq x \leq 8$
<b>p2</b>	Text and Numeric	Specifies nature of Pile 1
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p3</b>	Text and Numeric	Specifies nature of Pile 2
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p4</b>	Text and Numeric	Specifies nature of Pile 3
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p5</b>	Text and Numeric	Specifies nature of Pile 4
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p6</b>	Text and Numeric	Specifies nature of Pile 5
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p7</b>	Text and Numeric	Specifies nature of Pile 6
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used

<b>p8</b>	Text and Numeric	Specifies nature of Pile 7
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used
<b>p9</b>	Text and Numeric	Specifies nature of Pile 8
	R	Purely Random Pile
	Rx	Random cut of approximately x cards
	Sx	Specified cut of exactly x cards
	X	Pile is not used

### **CreatePilesDealAlternatingRandom(p1, p2)**

<b>p1</b>	Numeric	Specifies the number of Piles to deal
	x	$0 < x \leq 8$ , ( $\leq 7$ if not complete deck)
<b>p2</b>	Numeric	Specifies number of cards to deal
	x	$0 < x \leq 52$

### **CreatePilesDealAlternatingRegular(p1, p2)**

<b>p1</b>	Numeric	Specifies the number of Piles to deal
	x	$0 < x \leq 8$ , ( $\leq 7$ if not complete deck)
<b>p2</b>	Numeric	Specifies number of cards to deal
	x	$0 < x \leq 52$

### CreatePilesDealComplete(p1, p2, p3, p4, p5, p6, p7, p8, p9)

<b>p1</b>	Numeric	Specifies the number of Piles to deal
	X	$1 \leq x \leq 8$
<b>p2</b>	Text and Numeric	Specifies nature of Pile 1
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p3</b>	Text and Numeric	Specifies nature of Pile 2
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p4</b>	Text and Numeric	Specifies nature of Pile 3
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p5</b>	Text and Numeric	Specifies nature of Pile 4
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p6</b>	Text and Numeric	Specifies nature of Pile 5
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p7</b>	Text and Numeric	Specifies nature of Pile 6
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used

<b>p8</b>	Text and Numeric	Specifies nature of Pile 7
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used
<b>p9</b>	Text and Numeric	Specifies nature of Pile 8
	R	Purely Random Pile
	Rx	Random deal of approximately x cards
	Sx	Specified deal of exactly x cards
	X	Pile is not used

### **CreatePilesDealCompleteRandom(p1, p2)**

<b>p1</b>	Numeric	Specifies the number of Piles to deal
	x	$0 < x \leq 8$ , ( $\leq 7$ if not complete deck)
<b>p2</b>	Numeric	Specifies number of cards to deal
	x	$0 < x \leq 52$

### CutPiles(p1, p2, p3, p4)

<b>p1</b>	Text and Numeric	Specifies Cut Pile
	Px	Primary Pile x is cut $0 < x \leq 8$
	R	Random Pile is cut
<b>p2</b>	Text and Numericx	Specifies how much is cut
	R	Random cut depth
	C	Complete Pile is lifted during cut
	Sx	Specified number of x cards are cut
<b>p3</b>	Text and Numeric	Specifies placement of cut Pile
	P	Primary Pile
	E	Equivalent (same) Pile that was cut from (this is used when the cut Pile was Random)
	Sx	Secondary Pile x $0 < x \leq 8$
	R	Random Any Pile
	M	Top Same Pile
	D	Random Not Same (Different)
	Nx	New Pile specified (x) $0 < x \leq 8$
	L	New Pile Random
<b>p4</b>	Text	Specifies if cut portion is Reversed
	R	Cut portion is reversed
	X	Cut portion is not reversed

### CutDeckPrecise(p1, p2)

<b>p1</b>	Numeric	Specifies the number of cards to cut
	x	$0 < x \leq 52$
<b>p2</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### CutDeckRandom(p1)

<b>p1</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### CutSpecialRandom(p1, p2)

<b>p1</b>	Text	Specifies the number of cards to cut
	Quarter	Random cut, approximately Quarter cut
	Third	Random cut, approximately Third cut
	Half	Random cut, approximately Half cut
	Two Thirds	Random cut, approximately Two Thirds cut
	Three Quarters	Random cut, approximately Three Quarters cut
	Shallow	Random Shallow cut
	Deep	Random Deep cut
<b>p2</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### ElmsleyCount(p1)

<b>p1</b>	Numeric	Specifies the number of Piles to do Elmsley Count on
	x	$0 < x \leq 8$

### ForceCard(p1, p2)

<b>p1</b>	Numeric	Specifies the position of the Force card
	x	$0 < x \leq 52$
<b>p2</b>	Text	Specifies if Forced card is reversed
	X	Force card is not reversed
	R	Force card is reversed

The *ForceCard Event* must be followed by the *ReturnCard Event*.

### FreeChoiceSpreadSelect(p1, p2)

<b>p1</b>	Text	Specifies the position of the Force card
	"Any Card"	Totally random card is selected
	Top Third	Free selection from Top Third of deck
	Middle Third	Free selection from Middle Third of deck
	Bottom Third	Free selection from Bottom Third of deck
<b>p2</b>	Text	Specifies if Selected card is reversed
	X	Selected card is not reversed
	R	Selected card is reversed

The *FreeChoiceSpreadSelect Event* must be followed by the *ReturnCard Event*.

### InFaro

(no parameters with this Event)

### InFaroReverse

(no parameters with this Event)



### **InFaroSpecialBottom(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### **InFaroSpecialBottomReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### **InFaroSpecialTop(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### **InFaroSpecialTopReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### **InverseElmsleyCount(p1)**

<b>p1</b>	Numeric	Specifies the number of Piles to do Elmsley Count on
	x	$0 < x \leq 8$

### **InverseInFaro**

(no parameters with this Event)

### **InverseInFaroReverse**

(no parameters with this Event)

### **InverseInFaroSpecialBottom(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### **InverseInFaroSpecialBottomReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### **InverseInFaroSpecialTop(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### **InverseInFaroSpecialTopReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### **InverseJordanCount(p1)**

<b>p1</b>	Numeric	Specifies the number of Piles to do Jordan Count on
	x	$0 < x \leq 8$

### **InverseMoveCard(p1, p2)**

<b>p1</b>	Numeric	Specifies position of card to Move from
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies position to Move card to
	x	$0 < x \leq 52$

### **InverseMoveCardReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies position of card to Move from
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies position to Move card to
	x	$0 < x \leq 52$

### **InverseOutFaro**

(no parameters with this Event)

### InverseOutFaroReverse

(no parameters with this Event)

### InverseOutFaroSpecialBottom(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### InverseOutFaroSpecialBottomReverse(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### InverseOutFaroSpecialTop(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### InverseOutFaroSpecialTopReverse(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### **InverseRunSingleCards(p1)**

<b>p1</b>	Numeric	Specifies number of cards to Run
	x	$0 < x \leq 52$

### **InverseRunSingleCardsReverse(p1)**

<b>p1</b>	Numeric	Specifies number of cards to Run
	x	$0 < x \leq 52$

### **InverseShiftTopBlock(p1, p2)**

<b>p1</b>	Numeric	Specifies top block size
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies depth for insertion of block
	x	$0 < x \leq 52$

### **InverseShiftTopBlockReverse(p1, p2)**

<b>p1</b>	Numeric	Specifies top block size
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies depth for insertion of block
	x	$0 < x \leq 52$

### **JordanCount(p1)**

<b>p1</b>	Numeric	Specifies the number of Piles to do Jordan Count on
	x	$0 < x \leq 8$

### **Macro(p1)**

<b>p1</b>	Text	Specifies the Session file to run as a macro
	filename	filename.svs

### MoveCard(p1, p2)

<b>p1</b>	Numeric	Specifies position of card to Move from
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies position to Move card to
	x	$0 < x \leq 52$

### MoveCardReverse(p1, p2)

<b>p1</b>	Numeric	Specifies position of card to Move from
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies position to Move card to
	x	$0 < x \leq 52$

### OHShuffle

(no parameters with this Event)

### OHShuffleBottom(p1)

<b>p1</b>	Numeric	Specifies the number of cards to protect on bottom
	x	$0 < x \leq 52$

### OHShuffleTop(p1)

<b>p1</b>	Numeric	Specifies the number of cards to protect on top
	x	$0 < x \leq 52$

### OutFaro

(no parameters with this Event)

### OutFaroReverse

(no parameters with this Event)

### OutFaroSpecialBottom(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### OutFaroSpecialBottomReverse(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from bottom for weave
	x	$0 < x \leq 52$

### OutFaroSpecialTop(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### OutFaroSpecialTopReverse(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards cut from top
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies interior position from top for weave
	x	$0 < x \leq 52$

### PokerDeal(p1)

<b>p1</b>	Numeric	Specifies the number of Poker hands to deal
	x	$2 \leq x \leq 10$

### PokerDiscard(p1)

<b>p1</b>	Text or Numeric	Specifies the card to discard
	x	$0 < x \leq 50$ (specific position is discarded)
	VS	V = A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K (value) S = C, H, S, D (suit) (specific card is discarded)

### ResetCurrentDeck

(no parameters with this Event)

### ReturnCard(p1)

<b>p1</b>	Text or Numeric	Specifies the place to return card
	x	$0 < x \leq 52$
	"Original Position"	Returns card to original position (not with Force)
	"Anywhere"	Returns card randomly
	Top Third	Returns card randomly in Top Third
	Middle Third	Returns card randomly in Middle Third
	Bottom Third	Returns card randomly in Bottom Third



### ReverseCard(p1)

<b>p1</b>	Text or Numeric	Specifies the card to reverse
	x	$0 < x \leq 52$ (specific position is reversed)
	VS	V = A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K (value) S = C, H, S, D (suit) (specific card is reversed)

### RiffleShuffle(p1)

<b>p1</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### RiffleShuffleBottom(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards to protect
	x	$0 < x \leq 52$
<b>p2</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### RiffleShufflePile(p1, p2, p3, p4)

<b>p1</b>	Text and Numeric	Specifies Primary Pile
	x	Pile number
	Tx	Protect Top of Pile x
	Bx	Protect Bottom of Pile x
	xR	Pile number, reverse Pile
	TxR	Protect Top of Pile x, reverse Pile
	BxR	Protect Bottom of Pile x, reverse Pile
<b>p2</b>	Text and Numeric	Specifies Secondary Pile
	x	Pile number
	Tx	Protect Top of Pile x
	Bx	Protect Bottom of Pile x
	xR	Pile number, reverse Pile
	TxR	Protect Top of Pile x, reverse Pile
	BxR	Protect Bottom of Pile x, reverse Pile
<b>p3</b>	Numeric	Number of cards to Protect
	x	$0 < x \leq 52$
<b>p4</b>	Text	Specifies if Gilbreath View is shown
	X	No Gilbreath View is shown
	G	Gilbreath View is shown

### RiffleShuffleTop(p1, p2)

<b>p1</b>	Numeric	Specifies number of cards to protect
	x	$0 < x \leq 52$
<b>p2</b>	Text	Specified if either block is reversed
	X	No block is reversed
	T	Original Top block is reversed
	B	Original Bottom block is reversed

### RunSingleCards(p1)

<b>p1</b>	Numeric	Specifies number of cards to Run
	x	$0 < x \leq 52$

### RunSingleCardsReverse(p1)

<b>p1</b>	Numeric	Specifies number of cards to Run
	x	$0 < x \leq 52$

### SelectCardsCutSelectFace1(p1)

<b>p1</b>	Text	Specifies whether the selected card is reversed
	X	Selected card is not reversed
	R	Selected card is reversed

### SelectCardsCutSelectFace2(p1, p2)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed

### SelectCardsCutSelectFace3(p1, p2, p3)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed
<b>p3</b>	Text	Specifies whether the third selected card is reversed
	X	Third selected card is not reversed
	R	Third selected card is reversed

### SelectCardsCutSelectNext1(p1)

<b>p1</b>	Text	Specifies whether the selected card is reversed
	X	Selected card is not reversed
	R	Selected card is reversed

### SelectCardsCutSelectNext2(p1, p2)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed

### SelectCardsCutSelectNext3(p1, p2, p3)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed
<b>p3</b>	Text	Specifies whether the third selected card is reversed
	X	Third selected card is not reversed
	R	Third selected card is reversed

### SelectCardsCutSelectNextRepeat(p1, p2)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed

### SelectCardsCutSelectNextRepeat2(p1, p2, p3)

<b>p1</b>	Text	Specifies whether the first selected card is reversed
	X	First selected card is not reversed
	R	First selected card is reversed
<b>p2</b>	Text	Specifies whether the second selected card is reversed
	X	Second selected card is not reversed
	R	Second selected card is reversed
<b>p3</b>	Text	Specifies whether the third selected card is reversed
	X	Third selected card is not reversed
	R	Third selected card is reversed

### SelectReturn(p1, p2, p3, p4, p5)

<b>p1</b>	Text and Numeric	Specifies Selection Pile
	Px	Primary Pile x used for selected card $0 < x \leq 8$
	R	Random Pile used for selected card
<b>p2</b>	Text and Numericx	Specifies which card is selected and whether it is reversed
	T	Top card is selected and not reversed
	B	Bottom card is selected and not reversed
	R	Random card is selected and not reversed
	Sx	Specified position in pile is selected and not reversed $0 < x \leq \text{Number of cards in Pile}$
	TR	Top card is selected and is reversed
	BR	Bottom card is selected and is reversed
	RR	Random card is selected and is reversed
	SxR	Specified position in pile is selected and is reversed $0 < x \leq \text{Number of cards in Pile}$
<b>p3</b>	Text and Numeric	Specifies Return Pile
	Sx	Secondary Pile x used for return Pile $0 < x \leq 8$
	E	Equivalent (same) Pile that was selected from (this is used when the selected Pile was Random)
	Px	Primary Pile x $0 < x \leq 8$
	R	Random Any Pile
	D	Random Not Same (Different)
	Nx	New Pile specified (x) $0 < x \leq 8$
	L	New Pile Random
<b>p4</b>	Text and Numeric	Specifies return position in return Pile
	N	Creates new Pile of single card
	T	Returns to top card position
	B	Returns to Bottom position
	E	Equivalent (same) position as from selected Pile

	R	Random position in return Pile
	Sx	Specified position in return Pile $0 < x \leq \text{Number of cards in Pile}$
<b>p5</b>	Text	Specifies if card is a selection or only moved
	S	Specified card is a Selection
	M	Specified card is only Moved

### SetStack(p1)

<b>p1</b>	Text	Specifies which standard stack to load
	"Default"	stackview.svf is loaded
	"Aronson"	Aronson Stack is loaded
	"Ireland"	Ireland Stack is loaded
	"Eight Kings"	Eight Kings Stack is loaded
	"Joyal (CHaSeD)"	Joyal (CHaSeD) Stack is loaded
	"Joyal (SHoCkeD)"	Joyal (SHoCkeD) Stack is loaded
	"New Deck (Bicycle)"	New Deck (Bicycle) deck is loaded
	"New Deck (Fournier)"	New Deck (Fournier) deck is loaded
	"Nikola"	Nikola Stack is loaded
	"Osterlind"	Osterlind Breakthrough System is loaded
	"Si Stebbins (3)"	Si Stebbins (3) Stack is loaded
	"Si Stebbins (4)"	Si Stebbins (4) Stack is loaded
	"Stanyon"	Stanyon Stack is loaded
"Tamariz"	Tamariz Stack is loaded	

### ShiftTopBlock(p1, p2)

<b>p1</b>	Numeric	Specifies top block size
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies depth for insertion of block
	x	$0 < x \leq 52$

### ShiftTopBlockReverse(p1, p2)

<b>p1</b>	Numeric	Specifies top block size
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies depth for insertion of block
	x	$0 < x \leq 52$

### SwapCardsRandom(p1)

<b>p1</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### SwapDifferentColors(p1)

<b>p1</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections



### SwapDifferentSuits(p1, p2, p3)

<b>p1</b>	Text	Specifies suit of first swapped card
	X	Suit is Randomly selected
	C	Suit is Clubs
	H	Suit is Hearts
	S	Suit is Spades
	D	Suit is Diamonds
<b>p2</b>	Text	Specifies suit of second swapped card
	X	Suit is Randomly selected
	C	Suit is Clubs
	H	Suit is Hearts
	S	Suit is Spades
	D	Suit is Diamonds
<b>p3</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### SwapPiles(p1, p2)

<b>p1</b>	Text	Specifies first swap Pile
	Px	Primary Pile x, no reversal of Pile $0 < x \leq \text{number of piles}$
	R	Random, no reversal of Pile
	S	Pile includes Selected card, no reversal of Pile
	N	Pile does Not include selected card, no reversal of Pile
	PxR	Secondary Pile x, Pile is reversed $0 < x \leq \text{number of piles}$
	RR	Random, Pile is reversed
	SR	Pile includes Selected card, Pile is reversed
	NR	Pile does Not include selected card, Pile is reversed
<b>p2</b>	Text	Primary Pile x, no reversal of Pile $0 < x \leq \text{number of piles}$
	Px	Random, no reversal of Pile
	R	Pile includes Selected card, no reversal of Pile
	S	Pile does Not include selected card, no reversal of Pile
	N	Secondary Pile x, Pile is reversed $0 < x \leq \text{number of piles}$
	PxR	Random, Pile is reversed
	RR	Pile includes Selected card, Pile is reversed
	SR	Pile does Not include selected card, Pile is reversed
	NR	Primary Pile x, no reversal of Pile $0 < x \leq \text{number of piles}$

### SwapSameColor(p1, p2)

<b>p1</b>	Text	Specifies color of swapped cards
	X	Color is Randomly selected
	R	Color is Red
	B	Color is Black
<b>p2</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### SwapSameSuit(p1, p2)

<b>p1</b>	Text	Specifies suit of swapped cards
	X	Suit is Randomly selected
	C	Suit is Clubs
	H	Suit is Hearts
	S	Suit is Spades
	D	Suit is Diamonds
<b>p2</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### SwapSpecifiedCards(p1, p2, p3)

<b>p1</b>	Numeric	Specifies first swap card Stack Value
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies second swap card Stack Value
	x	$0 < x \leq 52$
<b>p3</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### SwapSpecifiedPositions(p1, p2, p3)

<b>p1</b>	Numeric	Specifies first swap card Position Value
	x	$0 < x \leq 52$
<b>p2</b>	Numeric	Specifies second swap card Position Value
	x	$0 < x \leq 52$
<b>p3</b>	Numeric	Specifies selection state
	0	Swapped cards are selections
	1	Swapped cards are not selections

### TurnOver(p1)

<b>p1</b>	Numeric	Specifies Pile to Turn Over
	x	$0 < x \leq \text{number of Piles}$

## Appendix 4: StackView File Structures

Within StackView, there are several aspects of your work that can be saved to a file for later retrieval. You may view (not edit) these files using the *View* menu as described on page 211.

### ***Deck File***

The *Deck File* is saved as a `.svf` extension. When you save a stack with a filename from within StackView, it is automatically given a `.svf` extension.

The SVF files are simply text files with a special extension. The special extension makes it more convenient for you to maneuver through file directories when you are saving or opening StackView files. Each Version 5.0 `.svf` file contains 105 lines. The first line of text corresponds to the *Stack Value* of the first card (top card) of the deck. The second line of text corresponds to the name of the card (such as JS for the Jack of Spades). The third line is the *Stack Value* of the second card in the deck. The fourth line is the name of the second card. And so on. This continues for each card in the deck. That gets you to the 104<sup>th</sup> line. The 105<sup>th</sup> line indicates the *Back Design* that is saved with the deck. Earlier versions of `.svf` files did not have the 105<sup>th</sup> line for the *Back Design*. Version 3 and later of StackView *Deck* files will open in Version 5. However, Version 5 *Deck* files will not open in any earlier version.

While it is recommended that you use the StackView program to create stacks and save the files, you can use any text editor to create the files and save them with a `.svf` extension. You must keep in mind that there is no error checking with regard to StackView files. It is assumed that all files have been created correctly.

Also remember that the file `stackview.svf` is reserved for the default stack to load into the program when it starts up. This file must be in the same directory

as the StackView program. The *Save Deck As Default* menu selection automatically save the current stack to the right filename and directory location as described on page 204.

### ***Session File***

The *Session File* is saved as a `.svs` extension. When you save a Session from within StackView, it is automatically given a `.svs` extension.

The SVF files are simply text files with a special extension. The special extension makes it more convenient for you to maneuver through file directories when you are saving or opening StackView files. Each SVS file contains an arbitrary number of lines of text, one line for each *Event* from your *Session*. (Card Selection *Events* have two lines that are kept in pairs – the 1<sup>st</sup> line of the pair is for the *Selection* portion of the *Event*, and the 2<sup>nd</sup> line of the pair is for the *Return* portion of the *Event*.) The first line of the *Session* file corresponds to the first *Event* of the *Session*, and the last line corresponds to the last *Event* of the *Session*.

Each line contains the text for each *Event* of the *Session*. The text is case sensitive. While it is strongly recommended that you use the StackView program to create *Sessions* and save the files, you can use any text editor to create the files and save them with `.svs` extensions. There is error checking within StackView when it reads *Session* files. If you chose to create your own *Session* files manually with a text editor, the complete set of allowable *Events* is found in Appendix 4 (page 239).

### ***Mnemonic File***

The *Mnemonic File* is saved as a `.svm` extension. When you save a *Mnemonic* set with a filename from within StackView, it is automatically given a `.svm` extension.

The SVM files are simply text files with a special extension. The special extension makes it more convenient for you to maneuver through file directories when you are saving or opening StackView files. Each `.svm` file contains 104 lines of text. The first 52 lines of text contain the *Mnemonics* for the card names, and the next 52 lines of text contain the *Mnemonics* for the *Stack Values*.

### ***Search File***

The *Search File* is saved as a `.svh` extension. When you save a *Search* state with a filename from within StackView, it is automatically given a `.svh` extension.

The SVH files are simply text files with a special extension. The special extension makes it more convenient for you to maneuver through file directories when you are saving or opening StackView files. Each SVH file contains all of the parameters necessary to resume a Search from wherever you left off. It is not advised to manually edit these files, as it will surely cause error conditions when they are read in by StackView.

### ***Threshold Trap File***

The *Threshold Trap File* is saved as a `.svt` extension. When you save a *Threshold Trap* setting with a filename from within StackView, it is automatically given a `.svt` extension.

The SVT files are simply text files with a special extension. The special extension makes it more convenient for you to maneuver through file directories when you are saving or opening StackView files. Each SVT file contains all of manipulation combinations that resulted in a trap that met your threshold conditions. For more information on *Threshold Trap* files, see page 197

# Index

---

## A

Advanced Test · 179  
Any Card to Any Number with Faros · 217  
Apple Mac · 230  
Aronson Stack · 19, 33  
Aronson, Simon · 18  
Assemble Poker Deal · 36  
**AssemblePokerDeal** · 239  
Australian Deal · 115  
**AustralianDeal** · 240

## B

Back Design · 147  
Bauer, Ron · 18  
Beam, Steve · 16  
Berger, Eugene · 18  
Bottom card · 18

## C

Clear ALL Session Events · 86  
Close, Michael · 17  
**CombinePiles** · 241  
Control Tab · 32  
    Arrange Cards · 32  
    Conventional Shuffles · 37  
    Faro Shuffles · 53  
    Poker Deal · 33  
    Reset Current Deck Order · 33  
Control Window · 32  
Create a Stanyon Variation Deck · 138  
Create Piles · 88

Cut Cards · 89  
Deal Cards · 93  
**CreatePilesCut** · 242  
**CreatePilesDealAlternatingRandom** · 243  
**CreatePilesDealAlternatingRegular** · 243  
**CreatePilesDealComplete** · 244  
**CreatePilesDealCompleteRandom** · 245  
Creating Custom Mnemonics · 144  
Credits · 16  
Custom Deck · 133  
    Control Buttons · 136  
    User Interface · 133  
Custom Mnemonics · 144  
Cut · 42  
Cut Piles · 112  
Cut Select Next · 71  
Cut Select Next Face · 73  
Cut Select Next Repeat · 75  
**CutDeckPrecise** · 246  
**CutDeckRandom** · 247  
**CutPiles** · 246  
**CutSpecialRandom** · 247

## D

Dale, Dick · 18  
Deck · 24  
    Deck Views · 24  
    Point of View · 24  
Deck File · 269  
Default Mnemonic File · 144  
Delete Current Event · 86  
Double-clicking · 20, 31



---

## *E*

Eight Kings Stack · 33  
Elmsley Count · 129  
**ElmsleyCount** · 247  
Event · 12

---

## *F*

File · 203  
  Exit · 207  
  Open Deck... · 203  
  Open Mnemonic... · 206  
  Open Search... · 205  
  Open Session... · 204  
  Print... · 206  
  Save Deck As Default · 204  
  Save Deck As... · 203  
  Save Mnemonic As Default · 206  
  Save Mnemonic As... · 206  
  Save Search As... · 205  
  Save Session As... · 204  
File Structures · 269  
Force Selection · 68  
**ForceCard** · 248  
Free Choice From Spread · 69  
Free Choice with Special Handling · 71  
**FreeChoiceSpreadSelect** · 248  
Frequently Asked Questions · 229

---

## *G*

Gilbreath Principle · 109  
Gilbreath View · 111

---

## *H*

Help · 211  
Hover mouse · 22, 31

---

## *I*

Import Current Deck from StackView · 136  
Importing a Deck · 142  
InFaro · 54, 248  
**InFaroReverse** · 248  
**InFaroSpecialBottom** · 249  
**InFaroSpecialBottomReverse** · 249  
**InFaroSpecialTop** · 249  
**InFaroSpecialTopReverse** · 249  
Insert Session File as Macro · 86  
Interrupting a Search · 200  
Inverse vs. Reverse · 21  
**InverseElmsleyCount** · 249  
**InverseInFaro** · 250  
**InverseInFaroReverse** · 250  
**InverseInFaroSpecialBottom** · 250  
**InverseInFaroSpecialBottomReverse** · 250  
**InverseInFaroSpecialTop** · 250  
**InverseInFaroSpecialTopReverse** · 251  
**InverseJordanCount** · 251  
**InverseMoveCard** · 251  
**InverseMoveCardReverse** · 251  
**InverseOutFaro** · 251  
**InverseOutFaroReverse** · 252  
**InverseOutFaroSpecialBottom** · 252  
**InverseOutFaroSpecialBottomReverse** · 252  
**InverseOutFaroSpecialTop** · 252  
**InverseOutFaroSpecialTopReverse** · 252  
**InverseRunSingleCards** · 253  
**InverseRunSingleCardsReverse** · 253  
**InverseShiftTopBlock** · 253  
**InverseShiftTopBlockReverse** · 253  
Ireland Stack · 33

---

## *J*

Jordan Count · 130  
**JordanCount** · 253  
Joyal Shuffle Meter  
  Break · 153  
  Distribution · 151  
  Group · 152  
  Permutation · 150  
  Shuffle Index · 154  
  Spread · 149  
Joyal ShuffleMeter · 148  
Joyal Stack · 33  
Joyal, Martin · 18

---

## *L*

Loomis, Dennis · 18

---

## *M*

**Macro** · 253  
Manipulations and Moves · 199  
Menus · 203  
Minch, Stephen · 18  
Mnemonic File · 271  
Mnemonics · 144  
Move Card · 50  
Move Event Down · 86  
Move Event Up · 86  
**MoveCard** · 254  
**MoveCardReverse** · 254

---

## *N*

New features · 20

Nikola Stack · 33

---

## *O*

**OHShuffle** · 254  
**OHShuffleBottom** · 254  
**OHShuffleTop** · 254  
Osterlind Stack · 33  
Osterling, Richard · 18  
OutFaro · 56, 254  
**OutFaroReverse** · 255  
**OutFaroSpecialBottom** · 255  
**OutFaroSpecialBottomReverse** · 255  
**OutFaroSpecialTop** · 255  
**OutFaroSpecialTopReverse** · 255  
Overhand · 41

---

## *P*

Partial Faro · 57  
Pile Control Matrix · 102  
Piles  
  Australian Deal · 115  
  Cut Piles · 112  
  Riffle Shuffle · 109  
  Select/Return · 105  
  Swap Piles · 122  
Piles Control · 88  
Piles Viewpoint · 98  
Play ALL Current Session · 85  
Play Current Event · 85  
Pocket PC · 230  
Poker Deal · 33  
  Assemble Poker Deal · 36  
*Poker Deal Discard* · 35  
**PokerDeal** · 256  
**PokerDiscard** · 256  
Position Value · 13, 30

---

## R

Recordable Events · 81  
Recording Sessions · 81  
Refresh Deck View · 37  
Reorder Retained Stack · 138  
Reset: New · 137  
Reset: Retain · 138  
**ResetCurrentDeck** · 256  
Retained vs. New Stack Values · 135  
**ReturnCard** · 256  
Reverse vs. Inverse · 21  
**ReverseCard** · 257  
Reversing Cards · 31  
Riffle · 37  
Riffle Shuffle · 109  
**RiffleShuffle** · 257  
**RiffleShuffleBottom** · 257  
**RiffleShufflePile** · 258  
**RiffleShuffleTop** · 258  
Riobóo, Ramón · 18  
Run Single Cards · 44  
**RunSingleCards** · 259  
**RunSingleCardsReverse** · 259

---

## S

Sample Exercises · 212  
Saving a Mnemonic File · 145  
Saving Your Created Stack · 143  
Search File · 271  
Select / Return · 105  
**SelectCardsCutSelectFace1** · 259  
**SelectCardsCutSelectFace2** · 259  
**SelectCardsCutSelectFace3** · 260  
**SelectCardsCutSelectNext1** · 260  
**SelectCardsCutSelectNext2** · 260  
**SelectCardsCutSelectNext3** · 261  
**SelectCardsCutSelectNextRepeat** · 261

**SelectCardsCutSelectNextRepeat2** · 261  
Selected Cards · 30  
Selection Tab · 63  
    Deck Visual Indicators · 64  
    Force Selection · 68  
    Free Choice From Spread · 69  
    Free Choice with Special Handling · 71  
    Swap Cards Selection · 77  
**SelectReturn** · 262  
Session Commands · 239  
Session File · 270  
Sessions Tab · 80  
Set Stack · 32  
**SetStack** · 263  
Shift Top Block · 47  
**ShiftTopBlock** · 263  
**ShiftTopBlockReverse** · 264  
Shuffle Index · 154  
ShuffleMeter · 148  
Si Stebbins Stack · 33  
Sisti, Jim · 18  
Special Faros · 57  
Specifying the Order of a Stack · 212  
Stack Value · 13, 29  
StackView · 11  
    Structure · 12  
StackView Advanced Test · 179  
StackView Search · 196  
StackView Test · 155  
Stanyon Stack · 33  
Stanyon Variation Deck · 138  
Swap Cards Selection · 77  
Swap Piles · 122  
**SwapCardsRandom** · 264  
**SwapDifferentColors** · 264  
**SwapDifferentSuits** · 265  
**SwapPiles** · 266  
**SwapSameColor** · 267  
**SwapSameSuit** · 267  
**SwapSpecifiedCards** · 267  
**SwapSpecifiedPositions** · 268

---

## *T*

Tamariz Stack · 33  
Tamariz, Juan · 18  
Test · 155  
Threshold Trap · 197  
Threshold Trap File · 271  
Tools · 209

- Custom Deck · 209
- Custom Mnemonic · 209
- Joyal Shuffle Meter · 209
- Piles Control · 209
- Set back Design · 210
- StackView Advanced Test · 210
- StackView Search · 210
- StackView Test · 210

*Top* card · 18  
Transfer List to Sessions · 202  
Transfer: New · 137  
Transfer: Retain · 137  
Transferring a Deck · 143  
Turn Over Pile · 131  
**TurnOver** · 268

---

## *U*

User Guide · 211  
User Interface · 13

---

## *V*

Version Incompatibilities · 237  
View · 207

- Controls · 207
- Deck · 207
- Deck File · 208
- Mnemonic File · 208
- Search File · 208
- Session File · 208
- Threshold Trap File · 208

---

## *W*

Window · 210

- Cascade · 210
- Open Window Selection · 211

Working with Sessions · 221