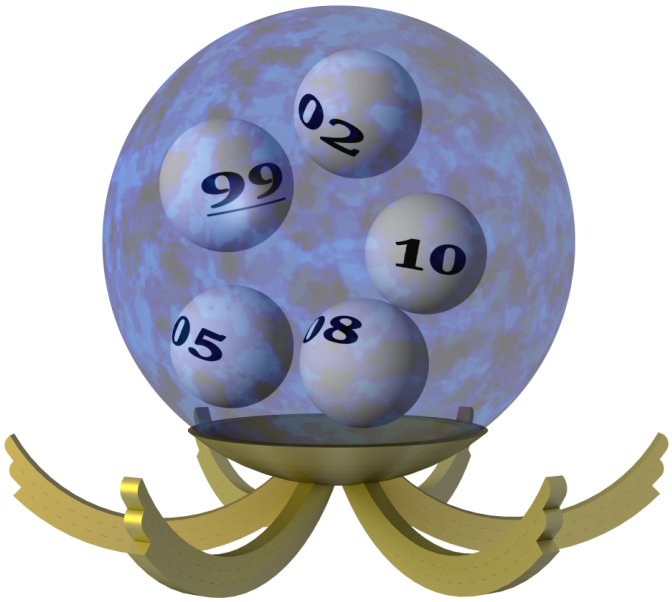
Lotto Sorcerer

V9

User's Guide



SATORI PUBLISHING

www.satoripublishing.com

Trademarks and Legal Notices

Satori Publishing
P.O. Box 8566
Michigan City, Indiana 46361-8566
U.S.A.
http://www.satoripublishing.com

Except where noted, screenshots in this manual are from the Mac OS X Snow Leopard (10.6) version of Lotto Sorcerer. The appearance of the screens from other operating systems will be reasonably similar to the views displayed here.

Table of Contents

Menu Directory

Conventions Used in this User's Guide

Menu Selection

Menus are referred to by using a greater-than symbol (">") between menu elements. For example, "Utilities > Database Utilities > Backup Database" means to choose the main menu item "Utilities", then, from the menu that appears, choose "Database Utilities", and then, from the submenu that appears, choose "Backup Database", as shown in Figure 1.

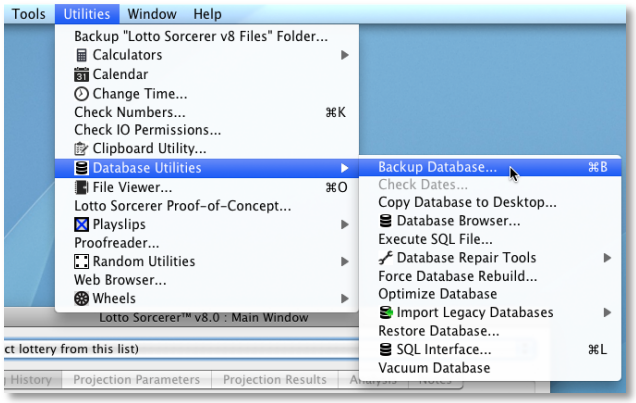


Figure 1.

Tab Selection

Tab selection is referred to in the same fashion as menu selection. For example, "Preferences > Auto" means to open the "Preferences" window, and then choose the "Auto" tab.

Ellipses in Menu Items

An ellipsis (...) at the end of a menu item indicates that an application needs additional user input to execute the item's command.

The "Main Window"

The "Main Window" refers to the primary window used by Lotto Sorcerer. The majority of activity takes place within this window, and closing this window terminates the program. Figure 2 shows screenshot of the Main Window:

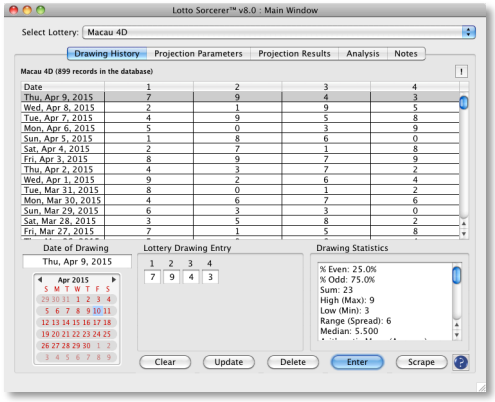


Figure 2.

Lotto Sorcerer End User License Agreement

This "End User License Agreement" ("Agreement"), dated July 1, 2015, supersedes all prior Agreements. Under this Agreement, Satori Publishing (the "Vendor") grants to the user (the "Licensee") a non-exclusive and non-transferable license ("License") to use Lotto Sorcerer (the "Software").

The English version of this Agreement supersedes all non-English versions, parts and parcels of the Agreement.

The Software includes the executable computer programs and any related printed, electronic and online documentation and any other files that may accompany the product.

Title, copyright, intellectual property rights and distribution rights of the Software remain exclusively with the Vendor. Intellectual property rights include the look and feel of the Software. This Agreement constitutes a license for use only and is not in any way a transfer of ownership rights to the Software.

The rights and obligations of this Agreement are personal rights granted to the Licensee only. The Licensee may not transfer or assign any of the rights or obligations granted under this Agreement to any other person or legal entity. The Licensee may not make available the Software for use by one or more third parties.

Failure to comply with any of the terms of this Agreement will be considered a material breach of this Agreement.

Part I: Your Agreement to this License

The Licensee should carefully and completely read this License before downloading, installing, using, distributing the Software, and before purchasing the registration fee for the Software. Unless you have a different license agreement signed by the Vendor, your use, distribution, or installation of the Software indicates your acceptance of the License. If agreement to this License is in violation of local, regional, national or international laws, you are prohibited from downloading, installing or using the Software. Your installation, use or purchase of the registration fee of the Software signifies your agreement to this License.

Part II: Evaluation Version Specifics

The terms and conditions of Part I, Part II and Part IV of this License describe the permitted use(s) of each Evaluation Copy of the Software.

Scope of License

This is not free software. Subject to the terms below, you are hereby licensed by the Vendor to use one copy of the Software, on one (1) computer or workstation, for evaluation purposes without charge for a period of 12 uses. If you use this software after the 12-use evaluation period a registration fee is required. For current pricing, see the Vendor's web site at http://www.satoripublishing.com/LS/ or write Satori Publishing, P.O. Box 8566, Michigan City, IN 46361-8566, USA. Payments via check must be in US dollars drawn on a US bank. Payments via money order must be made using a United States Postal Service money order or an international money order, and must be in US dollars. Checks and money orders should be sent to Satori Publishing, P.O. Box 8566, Michigan City, IN 46361-8566, USA.

See the Vendor's web site at http://www.satoripublishing.com/LS/ for information about secure online ordering. Online ordering is processed through PayPal, and can be transacted in other currencies besides US dollars.

Unregistered use of the Software after the 12-use evaluation period is in violation of U.S. and international copyright laws, including, but not limited to, the Digital Millennium Copyright Act of the United States.

You may, without making any payment to the Vendor:

a) give exact copies of this evaluation version of the Software personally to anyone, except for the purpose of extending their 12-use evaluation period;

b) distribute exact copies of this evaluation version of the Software, if done exclusively through electronic channels; and
c) make as many exact copies of this evaluation version of the Software as you wish, for purposes of distribution as described in (a) and (b) above.

You are not prohibited from charging, or requesting donations, for any copies of the evaluation version, however made, and from distributing such copies with other products of any kind, commercial or otherwise. However, the Vendor reserves the right to revoke the above distribution rights at any time, for any or no reason.

Part III: Licensed (Registered) Version Specifics

The terms and conditions of Part I, Part III and Part IV of this License describe the permitted use and user(s) of each Licensed (Registered) Copy of the Software.

For purposes of this License, if you have a valid single-copy license, you have the right to use a single "Licensed Copy" of the Software on one user account on one computer.

You agree to allow Lotto Sorcerer to transmit from time to time, via TCP/IP, information required to validate your registration code.

License Fee

The original purchase price paid by the Licensee will constitute the entire license fee and is the full consideration for this Agreement.

Scope of License

Each Licensed Copy of the Software may be used only by a single person on one user account on one computer. If there are multiple users on the computer or workstation, each user must have a separate license. This is not a concurrent use license. This is a "named-seats" license.

Part IV: General Terms and Conditions

The terms and conditions of Part IV of this License describe the permitted use of each Evaluation and/or Licensed (Registered) Copy of the Software.

All rights of any kind in the Software which are not expressly granted in this License are entirely and exclusively reserved to and by the Vendor.

You may not rent, lease, modify, translate, reverse engineer, decompile, disassemble, or create derivative works based on, the Software, nor permit anyone else to do so. Attempts to circumvent the registration process (including, but not limited to: counterfeiting the registration codes to the Software; using fraudulent means to acquire the registration codes to the Software) is software piracy and is a violation of international law, the Copyright Law and the Digital Millennium Copyright Act of 1998. Detected violations and the discovery will be reported to the appropriate law enforcement authorities; the violator will be subject to court costs, seizure of any property used in the act of the piracy, statutory and/or punitive damages to the Vendor of US$100,000 per infringed copy, plus fines of US$250,000 and/or prison terms for up to six years.

You may not make access to the Software available to others in connection with a service bureau, application service provider, or similar business, nor permit anyone else to do so.

Warranty Disclaimers and Liability Limitations

THE SOFTWARE, AND ANY AND ALL ACCOMPANYING SOFTWARE, FILES, DATA AND MATERIALS, ARE DISTRIBUTED AND PROVIDED "AS IS" AND WITH NO WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. YOU ACKNOWLEDGE THAT GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM, INCLUDING THE SOFTWARE, MUST BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE YOU RELY ON IT, AND YOU HEREBY ASSUME THE

ENTIRE RISK OF USING THE PROGRAM. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE.

THE VENDOR WILL NOT BE LIABLE FOR ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PRODUCTION, LOSS OF PROFITS, LOSS OF REVENUE, LOSS OF DATA, OR ANY OTHER BUSINESS OR ECONOMIC DISADVANTAGE SUFFERED BY THE LICENSEE ARISING OUT OF THE USE OR FAILURE TO USE THE SOFTWARE.

THE VENDOR DOES NOT WARRANT THAT USE OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. THE LICENSEE ACCEPTS THAT SOFTWARE IN GENERAL IS PRONE TO ERRORS AND FLAWS WITHIN AN ACCEPTABLE LEVEL AS DETERMINED IN THE INDUSTRY.

SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE SOFTWARE IS PROHIBITED FROM BEING INSTALLED, PURCHASED, USED OR DISTRIBUTED IN THOSE STATES OR JURISDICTIONS.

Historical lottery data included with the Software is included as a customer courtesy only. Although the Vendor endeavors in good faith to make this data complete and accurate, absolutely no warranty is implied as to its completeness and/or accuracy.

The word "guarantee", used in the Software, only has mathematical meaning and implies no legal liability.

Additionally, no refunds will be given after purchase of the Software. You must utilize the evaluation period to decide if the Software meets your needs and is compatible with your systems.

The absolute maximum liability of the Vendor will be limited exclusively to purchase price. In addition, in no event shall the Vendor, or its principals, owners, officers, employees, affiliates, contractors, subsidiaries, or parent organizations, be liable for any indirect, incidental, consequential, or punitive damages whatsoever relating to the use of the Software, or to your relationship with the Vendor.

In addition, in no event does the Vendor authorize you or anyone else to use the Software in applications or systems where the Software's failure to perform can reasonably be expected to result in a significant physical injury, or in loss of life. Any such use is entirely at your own risk, and you agree to hold the Vendor harmless from any and all claims or losses relating to such unauthorized use.

Warrants and Representations

The Vendor warrants and represents that it is the sole copyright holder of the Software. The Vendor warrants and represents that granting the license to use this Software is not in violation of any other agreement, copyright or applicable statute.

Acceptance

All terms, conditions and obligations of this Agreement will be deemed to be accepted by the Licensee ("Acceptance") on installation of the Software.

User Support

No user support or maintenance is provided as part of this Agreement.

Lottery Data Subscription Service

THE "LOTTERY DATA SUBSCRIPTION SERVICE" IS AN OPTIONAL ADD-ON SERVICE PROVIDED BY THE VENDOR, AND IS NOT INCLUDED WITH THE PURCHASE OF THE REGISTRATION FEE OF THE SOFTWARE.

Term

The term of this Agreement will begin on Acceptance and is perpetual.

Termination

This Agreement will be terminated and the License forfeited where the Licensee has failed to comply with any of the terms of this Agreement or is in breach of this Agreement. On termination of this Agreement for any reason, the Licensee will promptly destroy the Software or return the Software to the Vendor.

Force Majeure

The Vendor will be free of liability to the Licensee where the Vendor is prevented from executing its obligations under this Agreement in whole or in part due to Force Majeure, such as earthquake, typhoon, flood, fire, and war or any other unforeseen and uncontrollable event, natural or man-made, where the Vendor has taken any and all appropriate action to mitigate such an event.

Miscellaneous

This Agreement can only be modified in writing signed by both the Vendor and the Licensee.

This Agreement does not create or imply any relationship in agency or partnership between the Vendor and the Licensee.

Headings are inserted for the convenience of the parties only and are not to be considered when interpreting this Agreement. Words in the singular mean and include the plural and vice versa. Words in the masculine gender include the feminine gender and vice versa. Words in the neuter gender include the masculine gender and the feminine gender and vice versa.

If any term, covenant, condition or provision of this Agreement is held by a court of competent jurisdiction to be invalid, void or unenforceable, it is the parties' intent that such provision be reduced in scope by the court only to the extent deemed necessary by that court to render the provision reasonable and enforceable and the remainder of the provisions of this Agreement will in no way be affected, impaired or invalidated as a result.

This Agreement contains the entire agreement between the parties. All understandings have been included in this Agreement. Representations which may have been made by any party to this Agreement may in some way be inconsistent with this final written Agreement. All such statements are declared to be of no value in this Agreement. Only the written terms of this Agreement will bind the parties.

This Agreement and the terms and conditions contained in this Agreement apply to and are binding upon the Vendor's successors and assigns.

This License is the complete statement of the agreement between the parties on the subject matter, and merges and supersedes all other or prior understandings, purchase orders, agreements and arrangements. This License shall be governed by the laws of the State of Indiana, without regard to Indiana choice-of-law rules. Exclusive jurisdiction and venue for all matters relating to this License shall be in courts and fora located in the State of Indiana, county of La Porte, and you consent to such jurisdiction and venue. There are no third party beneficiaries of any promises, obligations or representations made by the Vendor herein. Any waiver by the Vendor of any violation of this License by you shall not constitute, nor contribute to, a waiver by the Vendor of any other or future violation of the same provision, or any other provision, of this License. This software is not to be used where the Software or any term of this License is void or prohibited by law.

System Requirements

Mac OS X

- Intel Macintosh (10.4 ["Tiger"] to 10.11 ["El Capitan"])

Windows

- Intel or AMD Athlon processor
- Windows XP, Windows Vista, Windows 7, Windows 8 (and 8.1), Windows 10

Program Overview

LOTTO SORCERER IS THE PREMIER, state-of-the-art, multi-threaded lottery number analysis and lottery prediction software. Originally based on the advanced statistical theories of Dr. W. Edwards Deming and Joseph M. Juran, it now couples their cutting-edge statistical analysis with predictive technology: fifth-generation artificial intelligence (neural network) algorithms, designed to detect subtle "patterns in chaos" to detect winning patterns and weighted influences in prior lottery draws, and then advises you, based on the best winning strategy.

The Basic Premise Behind Lotto Sorcerer's Number Suggestion Strategy

Critics and detractors say that "true random numbers cannot be predicted." And that is correct; no one can deny this. **But the basic premise behind Lotto Sorcerer's number suggestion generation strategy is that, because of their mechanical nature, *lottery drawings are close to, but not truly random.*** It is physics at work (specifically a function of Lagrangian mathematics), not randomness.

If there is a non-random influence at weight, it may be possible to detect a pattern to the non-random sequence. Lotto Sorcerer brings a wealth of tools to help find that pattern.

Why Lotto Sorcerer is the Vanguard of Lottery Software

Lotto Sorcerer is unique in that it looks for non-random patterns and influences. Even with lottery officials' attempts to make the drawings random, some weighted influence can alter the randomness. For example, does the weight of the ink on the balls have an effect? After all, the number "38" has over eight times the weight of ink than "1". Some balls have more ink than others, so there must be a weight variance. Are the balls of exactly the same thickness? Certainly not; plastic manufacturers generally cannot keep tolerances tighter than ±0.005" (±0.127 mm). Different thicknesses mean different weights. Although the weight differences are small, they still could (and probably do) effect whether some balls get picked more often than others.[1]

Some countries use wheels, instead of balls, to select the winning numbers. Are the wheels in perfect balance? Is the wheel spun with exactly the same torque? At the exact same starting position?

Is your lottery truly random? Or is there some weighted influence which slightly alters the odds? Only a neural network program, such as Lotto Sorcerer, which is designed to find patterns out of apparent chaos, can detect these influences. The end result is that you can maximize your hard-earned lottery-playing dollar.

1 The theory that differing weights of balls could effect the outcome of the lottery was, ironically, proved by criminals in a successful endeavor to alter the results of the Pennsylvania lottery in 1980: "The cheaters included key employees at a Pittsburgh TV station where drawings for Pennsylvania's Pick 3 game were held. A station art director, according to news reports from the time, injected a few grams of white latex paint into balls to be sucked into an air-powered machine. The cheaters weighed down all balls except those numbered with 4's and 6's, then bought combinations of those numbers. When 6-6-6 hit, they won $1.8 million." — Raleigh *News & Observer,* May 26, 2006.

Basic Operations

Quick Tutorial

Figure 3.

Although Lotto Sorcerer contains a wealth of tools and utilities, using the basic function of Lotto Sorcerer is an easy process, with just three steps:

1. Setup at least one lottery.
2. Enter numbers previously drawn (manually, through importing or online updating).
3. Have Lotto Sorcerer generate suggested numbers to play.

Step One of Three: Setup a lottery

You need to setup at least one lottery. Choose the menu item "Lottery Structure > Lottery Setup Wizard".

For more information on this subject, see Lottery Setup Wizard (page 42).

Step Two of Three: Bring the Lottery Drawing History Up-to-date

You will need to bring the lottery's database up-to-date with prior drawings so that Lotto Sorcerer can calculate a meaningful analysis and extrapolate recommended numbers to play. There are three ways to do this:

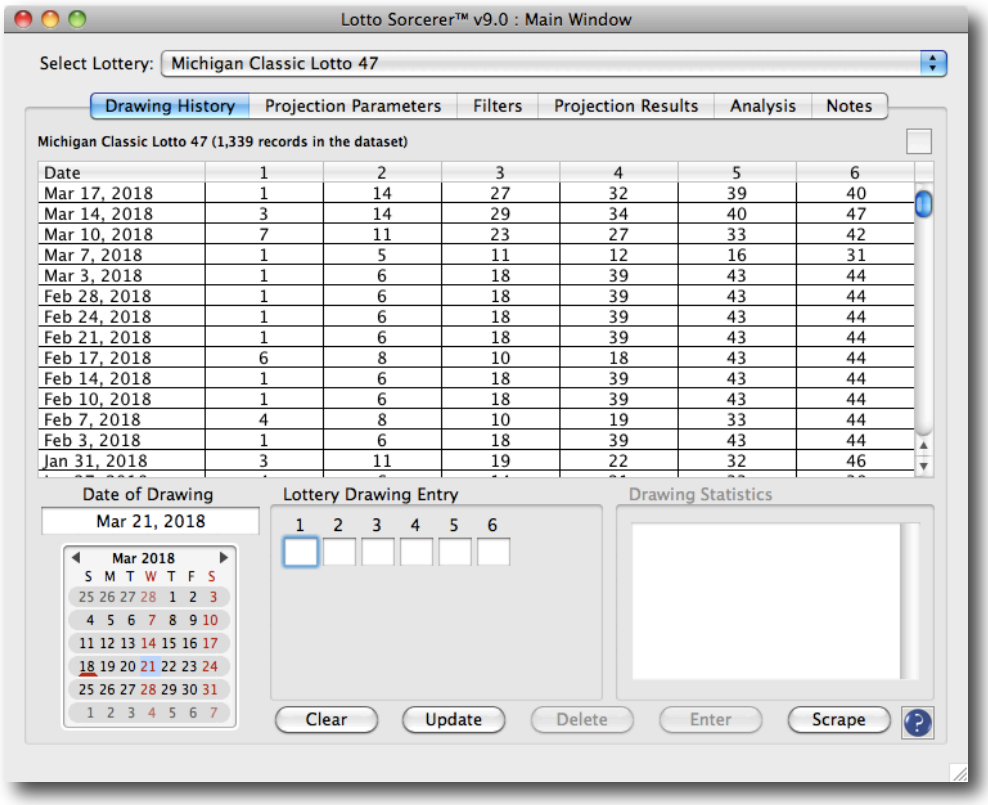
1. Entering Prior Draws via:
    a. Manual entry (page 20)
    b. Web scraping (page 306)
2. Updating by using the optional Lottery Data Subscription Service (page 92)
3. Importing past draws (page 61)

Step Three of Three: Have Lotto Sorcerer Generate Suggestions to Play

After you have entered a sufficient number of previous drawings into the database, you can have Lotto Sorcerer recommend numbers to play for the next upcoming drawing. To do this, from the Main Window, click on the Projection Parameters tab in the Main Window to set the parameters; then click the "Start" button in the Projection Results tab.

For more information on this subject, see Projection Parameters Tab (page 23).

Main Window



Date	1	2	3	4	5	6
Mar 17, 2018	1	14	27	32	39	40
Mar 14, 2018	3	14	29	34	40	47
Mar 10, 2018	7	11	23	27	33	42
Mar 7, 2018	1	5	11	12	16	31
Mar 3, 2018	1	6	18	39	43	44
Feb 28, 2018	1	6	18	39	43	44
Feb 24, 2018	1	6	18	39	43	44
Feb 21, 2018	1	6	18	39	43	44
Feb 17, 2018	6	8	10	18	43	44
Feb 14, 2018	1	6	18	39	43	44
Feb 10, 2018	1	6	18	39	43	44
Feb 7, 2018	4	8	10	19	33	44
Feb 3, 2018	1	6	18	39	43	44
Jan 31, 2018	3	11	19	22	32	46

Figure 4.

Overview

This is the primary window which is used in day-to-day operations within Lotto Sorcerer.

How to Invoke

The Main Window appears when Lotto Sorcerer is started, and closing the Main Window exits (quits) the program. If the Main Window is obscured by another window you can select it from Lotto Sorcerer's "Window" menu.

Basic Procedure

1. Select the lottery you want to work with in the "Select Lottery" dropdown menu
2. Enter the numbers drawn previously by using the "Drawing History" tab (or if you are using the optional Lottery Data Subscription Service, click the "Update" button)
3. Use the Projection Parameters tab to set generation options
4. Set Filters in the Filters tab
5. Use the Projection Results tab to generate suggestions

Window Controls

Select Lottery dropdown menu

Use this dropdown to select a lottery (that you have previously setup). If your lottery does not appear in this dropdown menu, set it up by using the Lottery Setup Wizard (page 42).

Drawing History tab

This tab is for entering, editing and deleting prior drawings into Lotto Sorcerer's database. For detailed information on this tab, see Drawing History Tab (page 20).

Projection Parameters tab

The Projection Parameters tab is used for selecting the analysis settings. For more information on this tab, see Projection Parameters (page 23).

Filters tab

Use this tab to set filters. Note that filters are for lotto-type and keno lotteries only... they are unavailable for "Pick type" lotteries. See page 27 for more information on the filters.

Projection Results tab

Use this tab to have Lotto Sorcerer analyze the data and suggest numbers to play. For more information on this tab, see Projection Results (page 34).

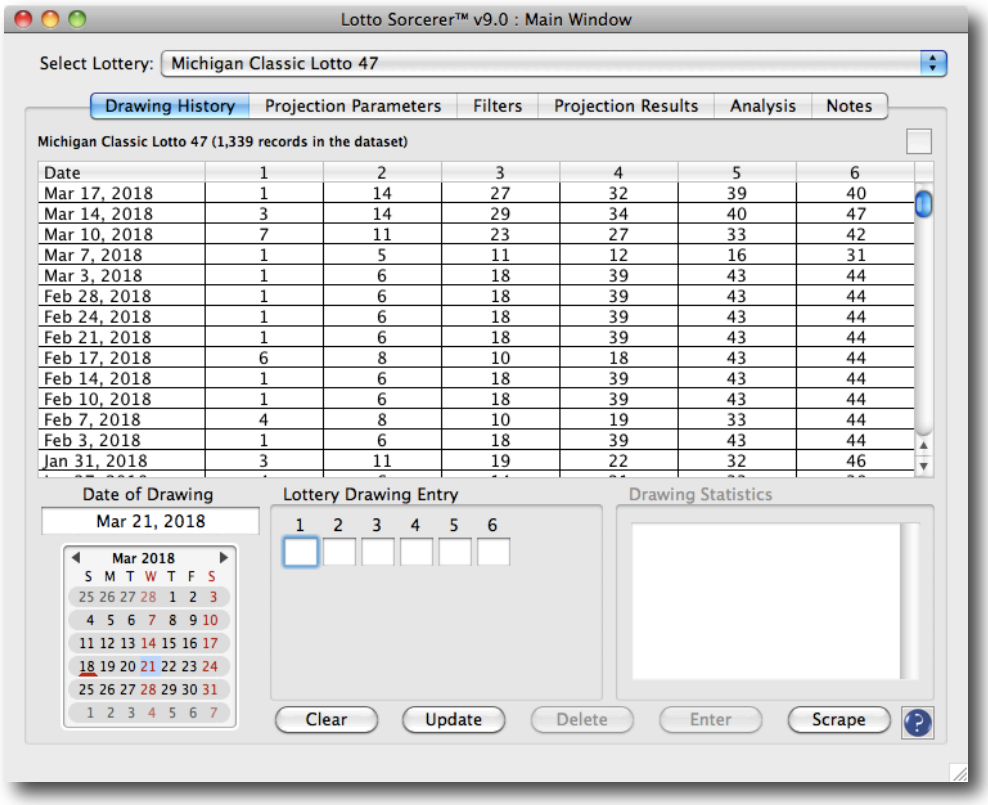
Analysis tab

The Analysis tab is used for viewing the in-depth suggestion analysis.

Notes tab

This tab lets you enter optional notes or a hyperlink for the lottery you are working with.

Drawing History Tab



Date	1	2	3	4	5	6
Mar 17, 2018	1	14	27	32	39	40
Mar 14, 2018	3	14	29	34	40	47
Mar 10, 2018	7	11	23	27	33	42
Mar 7, 2018	1	5	11	12	16	31
Mar 3, 2018	1	6	18	39	43	44
Feb 28, 2018	1	6	18	39	43	44
Feb 24, 2018	1	6	18	39	43	44
Feb 21, 2018	1	6	18	39	43	44
Feb 17, 2018	6	8	10	18	43	44
Feb 14, 2018	1	6	18	39	43	44
Feb 10, 2018	1	6	18	39	43	44
Feb 7, 2018	4	8	10	19	33	44
Feb 3, 2018	1	6	18	39	43	44
Jan 31, 2018	3	11	19	22	32	46

Figure 5.

Overview

This is the leftmost tab in the Main Window, and is used for entering, updating, editing or deleting the numbers that have been drawn in the lottery.

How to Invoke

After selecting a lottery from the "Select Lottery" dropdown, click the "Drawing History" tab in the Main Window.

Basic Procedure (Manual Data Entry)

1. Select the date of the drawing by clicking on the date in the Calendar
2. Enter the numbers drawn in the Lottery Drawing Entry text fields
3. Click the "Enter" button to enter it into the database

Basic Procedure (Editing)

You can also use this window for editing numbers that you have entered in error. To do this, just find the line in the grid at the top of the window that contains the data you want to correct, and click anywhere on its row; the data will appear in the bottom part of the window. Correct the data, then click the "Enter" button, or you can delete the entire drawing data by clicking the "Delete" button.

Basic Procedure (Online Updating)

Click the Update button in the Main Window. When doing so, Lotto Sorcerer will go online, and download the latest drawings for the selected lottery.

Online updating is only available for users who are subscribers to our optional Lotto Data Subscription Service, and for supported lotteries that were setup from the list of built-in lotteries in the Lottery Setup Wizard. For a list of current supported lotteries, use Lotto Sorcerer's menu item "Lottery Data > Lottery Data Subscription > Show Supported Lotteries".

Basic Procedure (Web Scraping)

If your lottery displays its drawings results online via a web page, you may be able to "scrape" this data into Lotto Sorcerer. For details, please see page 306.

Window Controls

Lottery grid

Located at the top of the window, this spreadsheet-style grid shows all of the numbers that you have entered for this lottery. Drawings are shown with the newest drawings at the top of the grid.

Date of Drawing field

Use this field shows the date of the drawing, whether you are entering a new drawing or editing an existing one. For Windows, this field uses the "Long" date setting on your computer (Control Panel > Region and Language > Format). For Mac OS X, this field uses the "Medium" date setting (System Preferences > Language & Text > Formats). You cannot enter or change the date in this field. To select a date, click on the desired date in the Calendar, located right below this field.

Lottery Drawing Entry fields

These are the numbered text boxes near the lower right part of the window. This is where you enter (or edit) past drawing numbers. Tip: for numbers less than ten, precede each number with a zero (for example, enter "7" as "07"), so that the text cursor automatically moves to the next field.

Calendar

Use the calendar to pick the day of the drawing. The vertical arrows to the right of the year let you increment and decrement the year; the horizontal arrows at the top left and top right of the calendar let you increment and decrement the month. Click on the specific day number to choose the day.

For details on using the Calendar control, please see page 304.

Clear button

Click this button to clear the buttons in the "Lottery Drawing Entry" text boxes.

Update button

If you are a subscriber to our optional Lottery Data Subscription Service, clicking this button will update your lottery. Please note that only supported built-in lotteries (that you selected with the Lottery Setup Wizard) can be updated in this fashion. If this button is not visible, then the lottery you are currently working with is not eligible for online updating.

Once updating is complete, this button becomes ghosted.

Delete button

Click this button to delete a record that already exists in the database. If it is ghosted, that is because you have not selected a record from the Lottery grid yet. Find the record in the grid, at the top of the window, and click on the row. The "Delete" button will become enabled.

Enter button

Click this to enter the data that you have typed into the database. You should then see it appear in the grid at the top of the window. If this button is disabled, that means that you have not filled out the date or all of the numbers drawn for that drawing.

Scrape button

This button lets you "scrape" a web page with drawing data to paste into the Lottery Drawing Entry boxes. Here is the specific procedure:

1. In the web page with the data you want to enter, click and drag through the data you want to enter.
2. With the data highlighted, copy the text data to your system clipboard. You can usually find this in your web browser's "Edit" menu.
3. In Lotto Sorcerer, click the Scrape button.

This will parse the data you selected and copied, and enter it into your Numbers Drawn boxes.

For more information, please see page 306.

Drawing Statistics Area

When you select a specific drawing in the Lottery grid, statistics about that drawing will appear in this window. Details of each statistic parameter are given in the Glossary (page 307).

Projection Parameters Tab

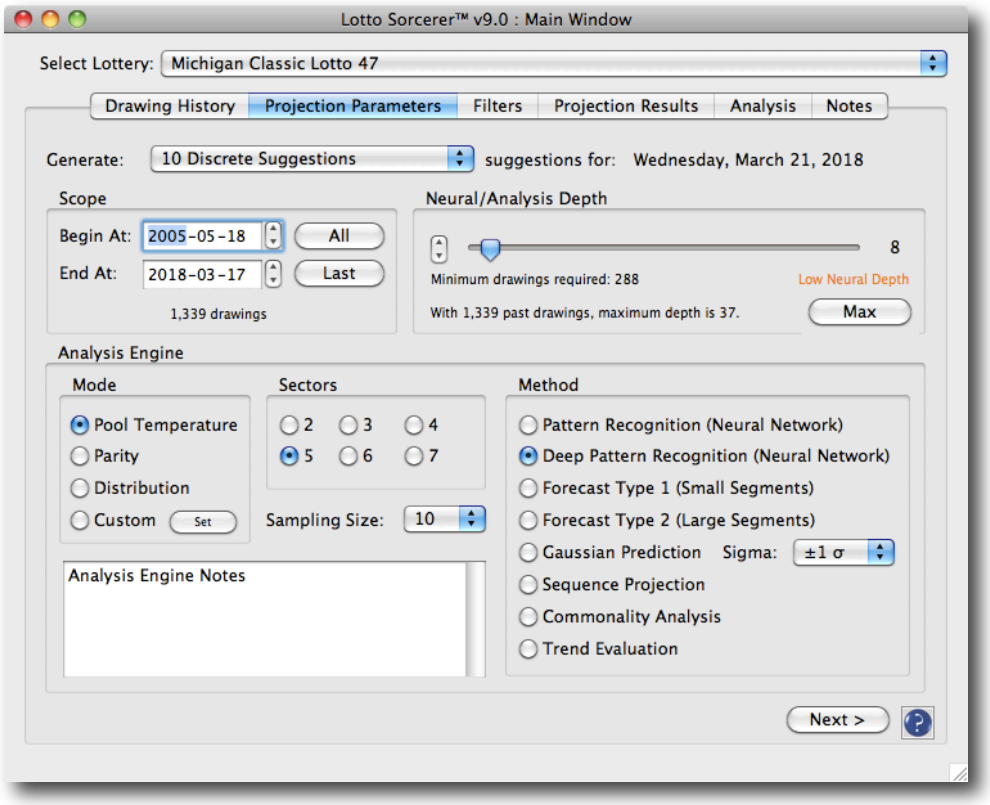


Figure 6.

Overview

This is the second tab in the Main window, and is used for setting the projection parameters used for the analysis.

How to Invoke

Click the "Projection Parameters" tab in the Main Window.

Basic Procedure

1. Select the number of lottery selections you want in the "Generate" dropdown menu
2. Set the Beginning and Ending Scope if you want something different from the default setting (optional)
3. Set the Neural/Analysis Depth if you want something different from the default (optional)
4. Choose the Neural/Analysis Mode
5. Select the Sectors you wish to use
6. Select the desired Sampling Size
7. Choose the Analysis method

Lotto Sorcerer v9 has a wealth of options available to you to "fine tune" your selection process. For help in choosing a selection strategy, please see "Choosing a Selection Generation Strategy" on page 301.

Window Controls

Generate dropdown menu

Use this dropdown to select the number of suggestions you want. All lotteries allow you to choose discrete (non-wheeled); most lotto-type lotteries also allow you to choose wheeled suggestions as well.

For discrete suggestions, choose from the dropdown menu the number of suggestions you want (from 1 to 25). If you want more than 25 suggestions, choose the "(other discrete suggestions)" selection from the dropdown menu. When you start the suggestion process, you will be asked for the quantity of suggestions you want (up to 999,999[2]).

If you want wheeled suggestions, choose the wheel you want from this dropdown menu. If there are no wheels showing up, make sure that you have the "Hide Wheels" unchecked in the Preferences window (see page 171). If there are still no wheels shown, then there are no wheels available for this lottery.

Scope

This control lets you select the starting and ending point for the analysis... that is, how far back in you want the starting analysis point to begin, and the ending analysis point.

It is generally recommended that you should start with the earliest drawing, so that Lotto Sorcerer can try to find longer patterns. However, there are some circumstances where you may want to choose a later date. One circumstance would be if your lottery changed its parameters at some point... for example, the lottery went from 1 to 48 balls to 1 to 52 balls. In this case, you may want to change the scope to the date where the parameters changed to the current settings. However, doing this will decrease the number of drawings available, and may lower the maximum Neural/Analysis Depth setting.

For generating actual suggestions to play in the lottery, it is strongly recommended that the ending Scope setting always be set to the last drawing in the dataset. This option to change the ending date is made available for experimentation purposes: if you set the ending Scope setting to the penultimate drawing, you can experiment with different settings and see if the suggestions generated match the final drawing in the dataset.

To change the Scope, use the date selector. For more information on using the date selector, see page 303. To quickly reset the beginning Scope to start with the first drawing, click the "All" button; to quickly set the ending Scope to end with the last drawing, click the "Last" button.

Neural/Analysis Depth Slider

Use this slider to select the starting neural/analysis depth. The larger the start depth, the more accurate the suggestions become, because Lotto Sorcerer will look for longer patterns; however, the greater the start depth, the longer the program will take to generate the suggestions. Also, the greater the start depth, more past numbers are required to run the analysis. Lotto Sorcerer will alert you if you do not have enough past drawings entered. In this case, either add more drawings to the database (if possible), or decrease the neural start depth.

☞ The evaluation version is limited to a maximum start depth of eight (8). The registered version can use the maximum number, 256 (if there are enough past drawings in the database).

Lotto Sorcerer will automatically set this control to the maximum setting, which is based on the Scope setting. If you are experimenting with the slider, and want to quickly set the slider to the maximum setting (for the number of drawings you have in your dataset), just click the "Max" button.

Neural/Analysis Mode dropdown menu

This dropdown menu lets you select the available Neural/Analysis mode you wish to use. There are currently four:

2 Note there are practical limitations to the number of suggestions possible, depending on the lottery. For example, for a Pick 3 type lottery (a lottery that draws three numbers between zero and nine), with five sectors, only eight suggestions are possible. Why? Because with five sectors, there are only two numbers per sector (10 digits divided by five is two). With only two numbers per position, and only three positions, the maximum is 2x2x2 = 8.

Pool Temperature
This mode works by selecting patterns by using "Hot numbers" (numbers drawn more often than average); "Cold numbers" (numbers drawn less often than average; and "Medium numbers" (numbers drawn close to average).

Parity
This mode looks for patterns in parity (even/odd).

Distribution
This mode looks for patterns based on the distribution of the drawn numbers. For example, for a lottery with numbers drawn from 1 to 48, "low numbers" would be "1 to 16", middle numbers would be "17 to 32" and "high numbers" would be "33 to 48".

Custom
This lets you run your own function (created by the Scripting Laboratory). Your script is expected to take complete control over the suggested generation process, from acquiring the drawing data to posting the suggested in the Projections Results tab. Note that when the script runs, it will not be run in its own thread.

Sectors
This is an important parameter (especially for lotteries which draw from small pools, as in Pick-type lotteries). The candidate pools should be of equal size as possible, otherwise undue weight is given to the largest pool.

In other words, *the sector size should be an even multiple for the numbers drawn.*

For example, suppose you are playing a "Pick 4" type lottery, which draws four numbers from zero to 9 (10 numbers). If you use our guideline, that "the sector size should be an even multiple for the numbers drawn", then you only have two options: 2 or 5 (because only these two numbers go into 10 equally).

What happens if you ignore this? What happens if you choose a sector size of "3", which does not go into 10 equally? The pools will not be equal size: two pools will have three numbers and one will have four. So the engine will favor the larger pool, giving you an erroneous result.

The rule of thumb is, "choose the largest sector size which is a factor of the number of numbers drawn". Of course, sometimes this is not possible (in the case of the "number of numbers drawn is a prime number"). In this case, just choose the largest sector available.

☞ In the case of the Parity mode, you are limited to two sectors, because of the nature of parity: there are only two choices (even and odd).

However, if you experimentation shows that a different sector size is yielding better results that what is recommended, use your experimentation size.

Sampling Size
This parameter effects only the Pool Temperature engine and/or the Limitation filters, which rely on calculating certain statistics, which, in turn, rely on "how far to go back".

This value is determined in units. A "unit" is based on the parameters of the lottery itself, where one unit is calculated as the number of drawings required for each number to be drawn at least once (if they were drawn equally). For example, for a lottery drawing five numbers between one and 35, the unit would be seven. So, in the case of this lottery, a Sampling Size of four would mean 28 past drawings from which the sampling is gathered.

Analysis Method

Choose the analysis method. There are ten choices:

1. Pattern Recognition (neural network)
2. Deep Pattern Recognition (neural network)
3. Forecast 1 (small segments)
4. Forecast 2 (large segments)
5. Gaussian Prediction (±1 σ)
6. Gaussian Prediction (±2 σ)
7. Gaussian Prediction (±3 σ)
8. Sequence Projection
9. Commonality Analysis
10. Trend Evaluation

Pattern Recognition uses a neural network to find a pattern in past drawings.

Deep Pattern Recognition also uses a neural network, but goes to a far greater depth. This is processor (and time) intensive, especially on older computers, and more so on "pick" type lotteries, because, in pick lotteries, each number is treated as a separate lottery.

Forecast 1 and **Forecast 2** do not use a neural network; it uses linear regression, and is identical to the algorithm used in the FORECAST function of Microsoft Excel. Forecast 1 uses small segments in its calculations, and Forecast 2 uses large segments.

Note that it is not uncommon for the results of both Forecast 1 and Forecast 2 to be identical, depending on the lottery, its drawing history, and other settings. Forecast 1 is more sensitive to short-range trends, and Forecast 2 is more sensitive to long-range trends.

The three **Gaussian Prediction** methods use the same technology that is in the Lottery Number Oracle function (Lotto Sorcerer menu item "Tools > Lottery Number Oracle"). This method has three settings, chosen from the dropdown menu to the right of the Gaussian Prediction selector, from which you can choose from one to three standard deviations (sigma). Note that it is not uncommon for the results of these three sigma settings to be identical, especially for lotteries that are very close to true random.

Sequence Projection looks at every possible pool combination, determines how often each specific pool combination occurs, and calculates which pool combination is most likely to be drawn in the next drawing.

Commonality Analysis chooses which pool combination has occurred more often than any other.

Trend Evaluation is similar to the Commonality Analysis method, except that far more statistical weight is given to more recent drawings.

Which analysis method should be used? Because all lotteries are different, only your experimentation can determine which to use.

Notes text box

This box lets you save short notes for this lottery's parameter settings. You are limited to 254 characters.

Filters Tab

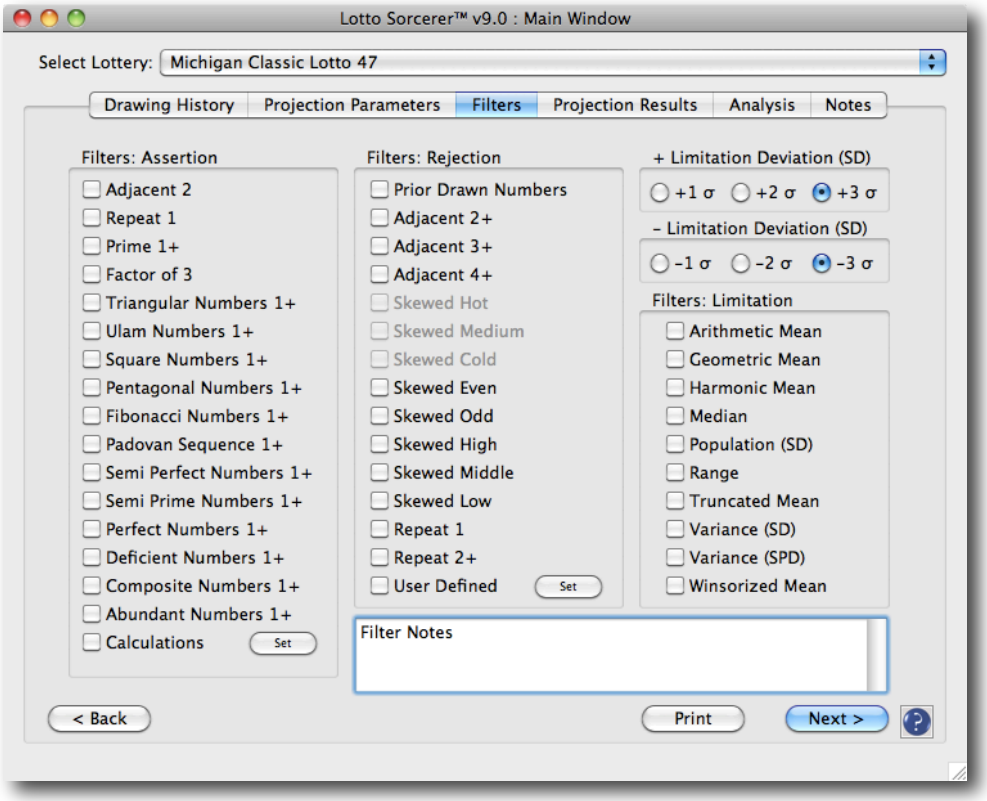


Figure 7.

Overview

This is the second tab in the Main window, and is used for setting the optional filters used for the analysis.

How to Invoke

Click the "Filters" tab in the Main Window.

Basic Procedure

1. Choose the Assertion filters (optional)
2. Choose the Rejection filters (optional)
3. Choose the Limitation Deviation (optional)
4. Choose the Limitation filters (optional)

Assertion Filters

Checking any of the Assertion Filters means that the attribute you choose must be in the suggestions produced. Here are the descriptions of the specific filters and their attributes:

Adjacent 2

This filter requires that each suggestion produced must have two, and only two numbers, that are "back-to-back". For example, the suggestion set of "02-03-08-19-31" would be allowed, because the "02" and the "03" are adjacent to each other.

Repeat 1

This filter requires that each suggestion produced contains one number, and only one number, that was chosen in the previous drawing. This does not consider bonus numbers.

Prime 1+
This filter requires that each suggestion produced contains at least one prime number. A prime number is a positive integer that has no positive divisors other than 1 and itself.

Factor of 3
This filter requires that each suggestion produced contains at least one number that is a factor of three.

Triangular Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "triangular number". A triangular number counts the objects that can form an equilateral triangle. The sequence of triangular numbers, up to 99, starting at the 0th triangular number, is: 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91.

Ulam Numbers 1+
This filter requires that each selection produced contains at least on number that is an "Ulam number". An Ulam number is a member of an integer sequence devised by and named after Stanislaw Ulam, who introduced it in 1964. The standard Ulam sequence (the (1, 2)-Ulam sequence) starts with $U_1 = 1$ and $U_2 = 2$. Then for $n > 2$, U_n is defined to be the smallest integer that is the sum of two distinct earlier terms in exactly one way. The sequence of Ulam numbers (up to 99) is: 1, 2, 3, 4, 6, 8, 11, 13, 16, 18, 26, 28, 36, 38, 47, 48, 53, 57, 62, 69, 72, 77, 82, 87, 97, 99.

Square Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "square number". A square number is an integer that is the square of an integer. The sequence of square numbers, up to 99, is: 1, 4, 9, 16, 25, 36, 49, 64, 81.

Pentagonal Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "pentagonal number". A pentagonal number is a figurate number that extends the concept of triangular and square numbers to the pentagon, but, unlike the first two, the patterns involved in the construction of pentagonal numbers are not rotationally symmetrical. The sequence of pentagonal numbers, up to 99, is: 1, 5, 12, 22, 35, 51, 70, 92.

Fibonacci Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "Fibonacci number". A Fibonacci number is characterized by the fact that every number after the first two is the sum of the two preceding ones. The sequence of Fibonacci numbers, up to 99, is: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.

Padovan Sequence 1+
This filter requires that each suggestion produced contains at least one number that is a member of the "Padovan sequence", named after Richard Padovan. The Padovan sequence, up to 99, is: 1, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86.

Semi Perfect Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "semi perfect number". A semi perfect number is a natural number n that is equal to the sum of all or some of its proper divisors. The sequence of semi perfect numbers, up to 99, is: 6, 12, 18, 20, 24, 28, 30, 36, 40, 42, 48, 54, 56, 60, 66, 72, 78, 80, 84, 88, 90, 96.

Semi Prime Numbers 1+
This filter requires that each suggestion produced contains at least one number that is a "semi prime number". A semi prime number is a natural number that is the product of two (not necessarily distinct) prime numbers. The sequence of semi prime numbers, up to 99, is: 4, 6, 9, 10, 14, 15, 21, 22, 25, 26, 33, 34, 35, 38, 39, 46, 49, 51, 55, 57, 58, 62, 65, 69, 74, 77, 82, 85, 86, 87, 91, 93, 94, 95.

Perfect Numbers 1+

This filter requires that each suggestion produced contains at least one number that is a "perfect number". A perfect number is a positive integer that is equal to the sum of its proper positive divisors. The sequence of perfect numbers up to 99, is: 2, 3, 4, 6, 8, 10, 12, 15, 18, 21, 24, 26, 32, 39, 42, 60, 65, 72, 84, 96.

Deficient Numbers 1+

This filter requires that each suggestion produced contains at least one number that is a "deficient number". A deficient number is a number in which the sum of all the divisors of the number $\sigma(n)<2$ is less than twice the value of the number *n*. The sequence of deficient numbers up to 99, is: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 29, 31, 32, 33, 34, 35, 37, 38, 39, 41, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 55, 57, 58, 59, 61, 62, 63, 64, 65, 67, 68, 69, 71, 73, 74, 75, 76, 77, 79, 81, 82, 83, 85, 86.

Composite Numbers 1+

This filter requires that each suggestion produced contains at least one number that is a "composite number". A composite number is a positive integer that can be formed by multiplying together two smaller positive integers. The sequence of composite numbers, up to 99, is: 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99.

Abundant Numbers 1+

This filter requires that each suggestion produced contains at least one number that is an "abundant number". An abundant number is a number for which the sum of its proper divisors is greater than the number itself. The sequence of abundant numbers, up to 99, is: 12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96.

Calculations

This filter is actually a collection of 10 separate assertion filters, each filter a calculated value. Clicking the "Set" button will invoke the Set Assertive Calculations window.

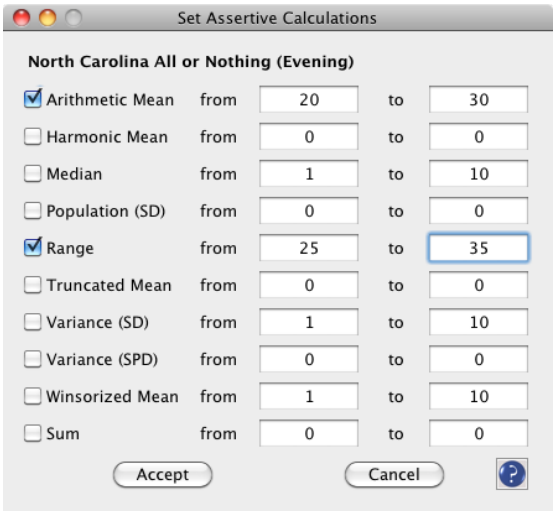


Figure 8.

To set any of the assertive calculation filters, check the desired filter and enter the appropriate values in the "from" and "to" filters. Also, the "Calculations" filter in the Assertion Filters section of the Projection Parameters in the Main Window must be checked.

When any of the Assertive Calculation filters is active, only those suggestions that meet the entered criteria will be allowed. For example, using the settings shown in Figure 8, only suggestions which have an arithmetic mean from 20 to 30 and only those suggestions which have a range[3] of 25 to 35 will be suggested.

These filters may appear to be similar to the Limitation Filters, but there are differences. First, the Assertion Calculation filters require certain values to be present, whereas the Limitation filters reject suggestions based on the selected criteria. Second, the Limitation Filters are selected as a multiple of Standard Deviation, whereas the Assertive Calculations use discrete values as the selection criteria.

Rejection Filters

These filters reject potential suggestions outright. These filters are available only for certain lotteries.

☞ Some rejection filters are unavailable based on the Neural/Analysis mode you choose. For example, if you choose the "Pool Temperature" mode, which selects numbers on their frequency, then the filters "hot", "cold" and "medium" are disabled, because this particular engine may actually suggest drawing numbers from one particular pool.

Prior Drawn Numbers

This filter rejects any suggestion that consists of numbers that were previously drawn. For example, if the numbers "02-03-08-19-31" is suggested, and all of those numbers were previously chosen, that suggestion is rejected.

Adjacent 2+

This filter rejects all suggestions with "back-to-back" numbers. For example, if the numbers "02-03-08-19-31" is suggested, it will be rejected, because of the "back-to-back" nature of "02-03-08-19-31".

Adjacent 3+

This filter rejects all suggestions with "back-to-back-to-back" numbers. For example, if the numbers "02-07-08-09-31" is suggested, it will be rejected, because of the "back-to-back-to-back" nature of "02-07-08-09-31".

Adjacent 4+

This filter rejects all suggestions with "back-to-back-to-back-to-back" numbers. For example, if the numbers "02-07-08-09-10" is suggested, it will be rejected, because of the "back-to-back-to-back-to-back" nature of "02-07-08-09-10".

Skewed Hot

This filter rejects suggestions where all numbers are from the "Hot" pool (i.e., numbers which are drawn more often than average).

Skewed Medium

This filter rejects suggestions where all numbers are from the "Medium" pool (i.e., numbers are neither hot nor cold).

Skewed Cold

This filter rejects suggestions where all numbers are from the "Cold" pool (i.e., numbers which are drawn less often than average).

Skewed Even

This filter rejects suggestions where all numbers are even.

Skewed Odd

This filter rejects suggestions where all numbers are odd.

---

3 It is important to note that "Range" refers to statistical range, i.e., the difference between the highest and lowest numbers of a set... it does *not* mean that suggested numbers will fall in a particular range of numbers.

Skewed High

This filter rejects suggestions where all numbers are from the "High" pool (i.e., numbers which are drawn from the high range of distribution).

Skewed Middle
This filter rejects suggestions where all numbers are from the "Middle" pool (i.e., numbers which are drawn neither from the high range nor low range of distribution).

Skewed Low
This filter rejects suggestions where all numbers are from the "Low" pool (i.e., numbers which are drawn from the low range of distribution).

Repeat 1
This filter rejects suggestions that contains a number that is in the previous drawing. This does not consider bonus numbers.

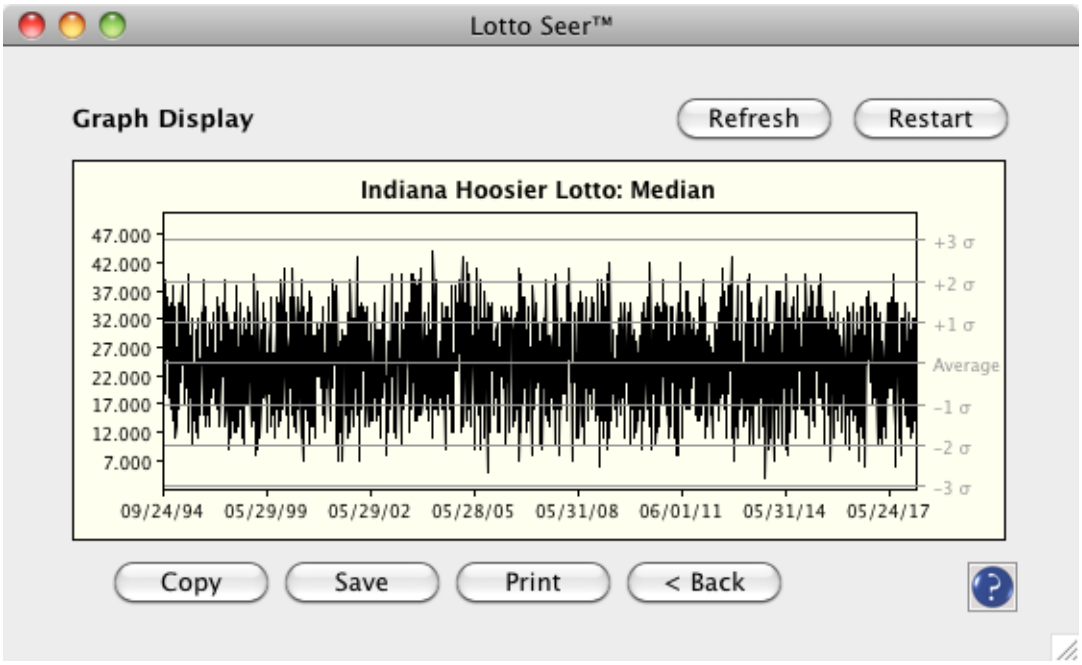
Repeat 2+
This filter rejects suggestions that contains two or more numbers that are in the previous drawing. This does not consider bonus numbers.

User Defined
The filter allows the user to enter a string of numbers which will be disallowed from the suggestions. Clicking the Set button will invoke the Set Rejectors window. In Set Rejectors window, enter the comma-delimited string of numbers to be rejected. For example, if the number set "4,10,11,23" is entered, any suggestion generated with *any* of those numbers will not be presented.

Limitation Deviation
This selection affects the range of the Limitation filters. The Limitation filters keep all suggestions within a certain statistical range, but what range? This control lets you select that range.

The range is defined as *standard deviations.* You can select from 1 standard deviation ("sigma" or "σ") to 3 standard deviations. You can set the "from" and the "to" values independently from each other, but not for each Limitation Filter.

Tip: if you use the Lotto Seer function (menu item "Tools > Lotto Seer"), you can view how often prior drawings exceed any of the standard deviations by viewing the line chart for that particular statistical function. For example, this shows the Median values:



As you can see, the right of the chart shows the six standard deviation lines, three above the Average (+1 σ, +2 σ, and +3 σ); and three below the Average (-1 σ, -2 σ, and -3 σ). The chart itself shows that no drawing has ever gone below -3 σ nor above +3 σ . So setting the Limitation Deviation radio buttons at -3 σ and +3 σ would ensure no suggestion will appear that would exceed those values.

Limitation Filters

These filters only pass suggestions that are within the statistical norm of prior drawings. The "within" is determined by the Limitation Deviation; the "statistical norm" is determined by the Sampling Size.

The Limitation Filters are purely statistical functions, and are described in the Glossary (page 307).

Notes text box

This box lets you save short notes for this lottery's filter settings. You are limited to 254 characters.

Print button

This prints out the Projection Parameters and Filters settings.

Next button

Click this to go to the "Projection Results" tab.

Note

When you click the "Start" button on the Projection Results page, Lotto Sorcerer memorizes most of the settings on this page. The next time you choose this lottery, the settings will return to the setting you have set here.

Projection Results Tab

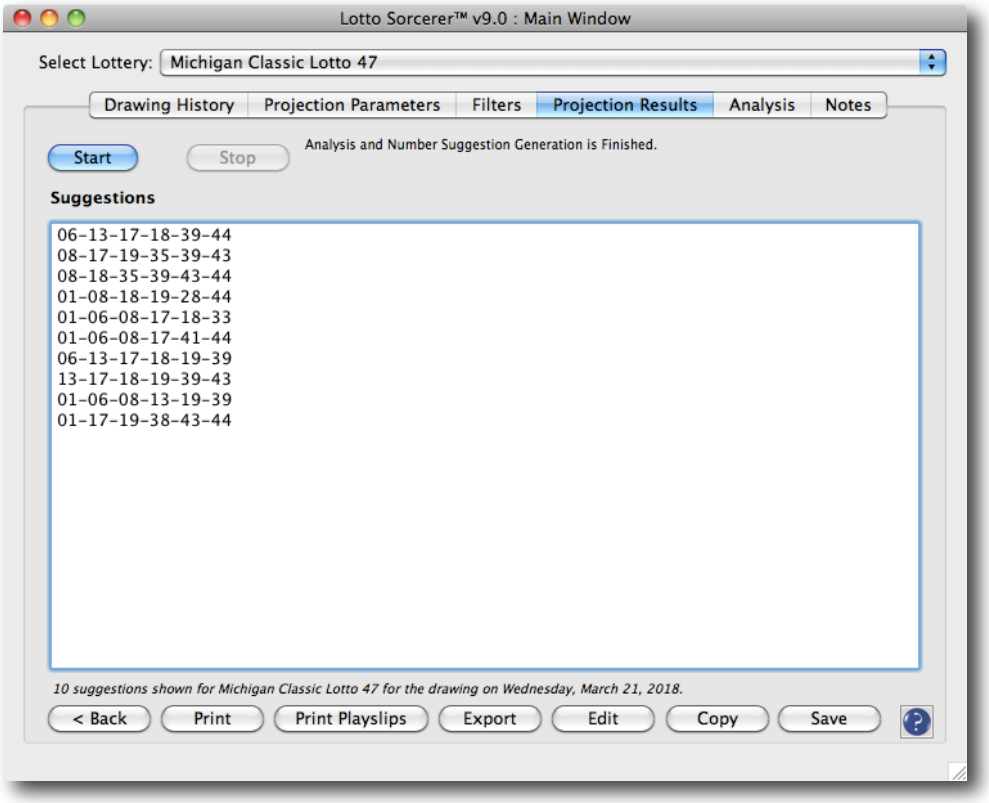


Figure 9.

Overview

This is the third tab in the Main window, and is used for generating and viewing or printing the results of the Analysis.

How to Invoke

Click the "Projection Results" tab in the Main Window.

Basic Procedure

- Click the Start button

Window Controls

Start button

This button starts the generation process. Once the process starts, this button is "ghosted" until the generation is finished (or unless the "Stop" button is pressed).

Stop button

This button stops the generation process. This button is available only if there is a generation is in process.

Suggestions box

When the suggestions are complete, they will be displayed here. When the suggestions are displayed, double-clicking on any suggestion invokes a window which shows statistics for that suggestion.

Back button

This button takes you back to the Projection Parameters page.

Print button

This button allows you to print the suggestions.

Print Playslips button

This button invokes the Print Playslip dialog box, allowing for the printing of the suggestions onto playslip(s).

Export button

This button exports the suggestions, *along with the statistics for each suggestion,* into a Microsoft Excel spreadsheet.

Edit button

Clicking this button invokes the Edit Suggestions dialog box, allowing you to edit the suggestions. This is described on page 39.

Copy button

This copies the suggestions to the System Clipboard.

Save button

Clicking this button invokes a standard file dialog box, allowing you to save the suggestions to a text file.

Analysis Tab

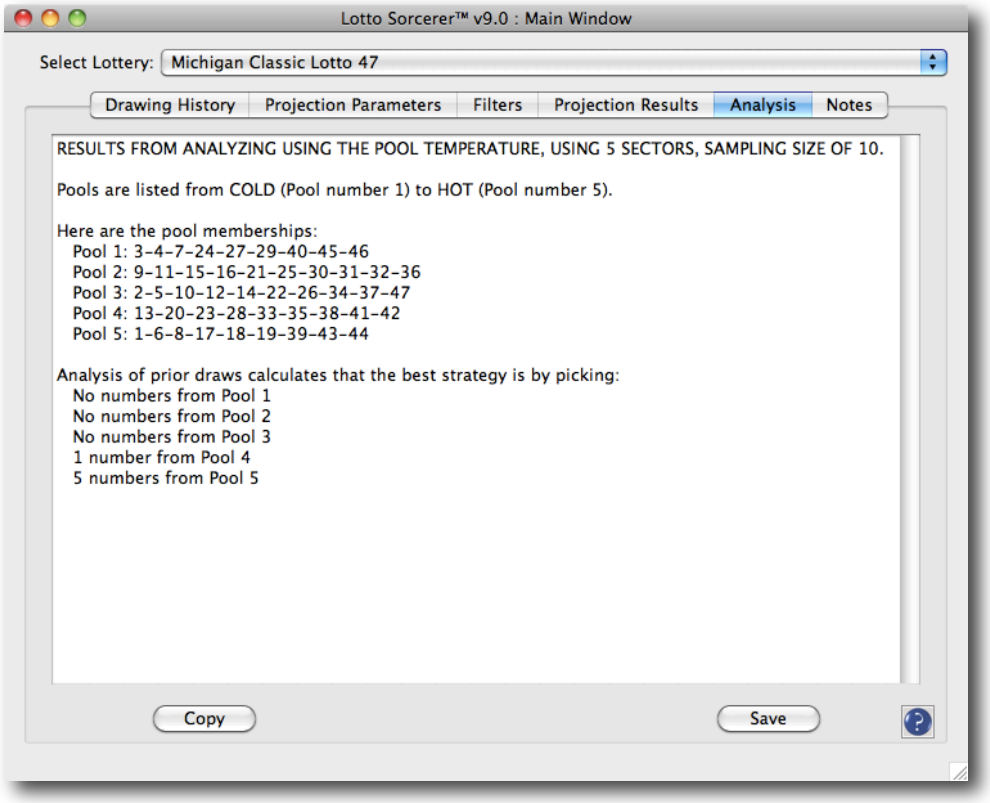


Figure 10.

Overview

This is the fourth tab in the Main window, and is used for displaying an in-depth analysis of the generate suggestions.

How to Invoke

Click the "Analysis" tab in the Main Window.

Window Controls

Analysis text box

This box holds the contents of the analysis, and is populated when the generation process is complete.

Copy button

This copies the analysis to the System Clipboard.

Save button

Use this for saving the analysis to a text file.

Notes Tab

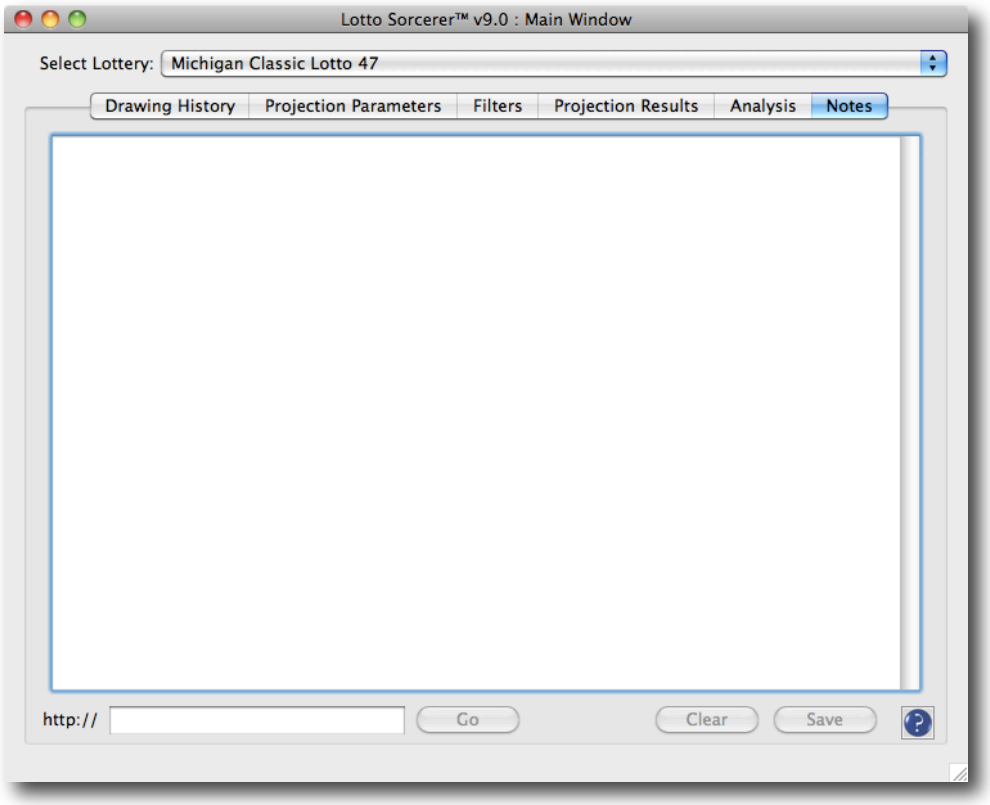


Figure 11.

Overview

This is the rightmost tab in the Main window, and is used for entering an optional note and a hyperlink for the current lottery.

How to Invoke

Click the "Notes" tab in the Main Window.

Window Controls

Notes text box

This is used for entering or editing your note. There is no practical limit as to how large your note can be.

Delete button

This is used to delete the note.

Save button

Use this for saving the note.

Hyperlink text box

Enter the URL (web address) for the web page you want to be able to view. Note that you should not enter the "http://" prefix.

Go button

Clicking this button will launch your default web browser and take you to the web page you entered in the Hyperlink box.

☞ This button becomes a "Save" button (for saving the hyperlink) whenever you edit text in the Hyperlink box.

Edit Suggestions

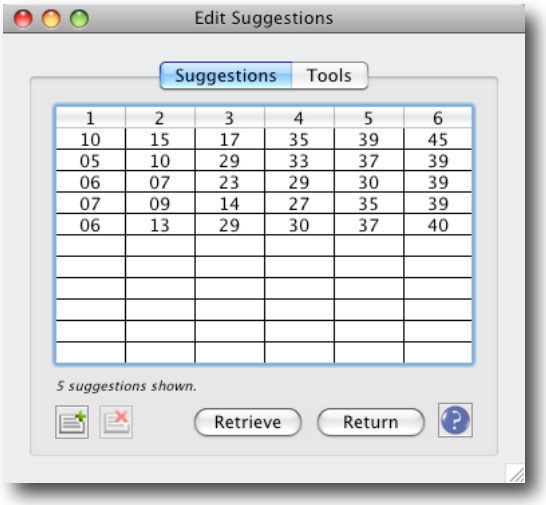


Figure 12.

Overview

This function was requested by users who, basically, wanted to add their own "lucky numbers" into Lotto Sorcerer's suggestions.

This function allows you to:

- Edit the suggestions created by Lotto Sorcerer.
- Delete any of the suggestions created by Lotto Sorcerer.
- Insert your own suggestions.

How to Invoke

Click the "Edit" button in the Projection Results tab in the Main Window. Please note that this button is enabled only when there are suggestions present.

Window Controls

Suggestions Tab: Suggestions list box

The Suggestions list box is like a spreadsheet. You can:

- Sort each column by clicking on the top row. The sorting alternates between descending and ascending by each subsequent click.
- Edit an individual cell by clicking on it.

Suggestions Tab: Add Suggestion icon

Clicking on this icon will append a suggestion to the end of the Suggestions list box. From here, you can edit the suggestion.

Suggestions Tab: Delete Suggestion icon

When you highlight a row in the Suggestions list box, this button will become enabled; clicking on this will delete that row.

Suggestions Tab: Retrieve button

This button will retrieve the suggestions from the Suggestions box from the Suggestions tab from the Main window.

Suggestions Tab: Return button

This button will push the suggestions that you edited back to the Suggestions box in the Suggestions tab in the Main Window.

Tools Tab: Search For dropdown menu

Use this control to perform a "Search and Replace" on the suggestions in the Suggestions list box. This control selects the number to be replaced.

Tools Tab: Replace With dropdown menu

This control will replace the items which you select in the previous control.

Tools Tab: Start button

This button starts the "Search and Replace" process.

Lottery Structure

Lottery Setup Wizard

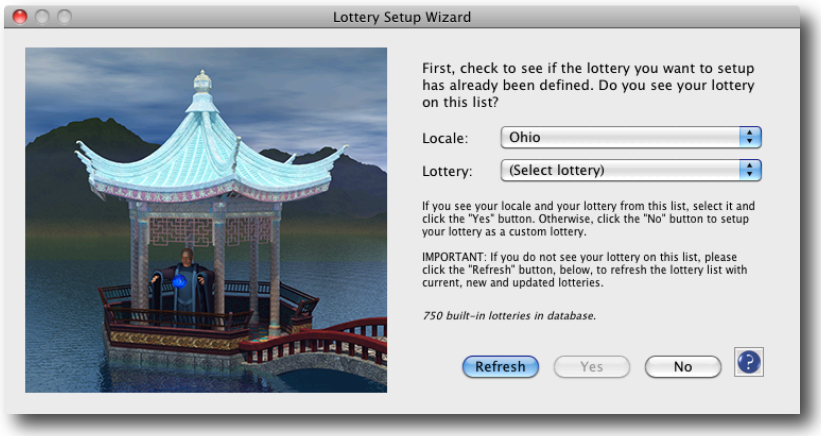


Figure 13.

Overview

This is used to setup a lottery in Lotto Sorcerer. Lotto Sorcerer comes with 500+ preset lotteries. If you can find the lottery you want to setup, and the settings are current and accurate, you can quickly set it up with only two mouse clicks. You can also "override" the settings if your lottery has changed. Or you can setup a new lottery which isn't on the list.

How to Invoke

Use the menu item "Lottery Structure > Lottery Setup Wizard".

Basic Procedure

1. Find the lottery you want to setup from the "Locale" and "Lottery" dropdown menu, or...
2. Create your own lottery. You will be taken through several pages, asking you questions about your lottery
3. Click the "Create" button to create the lottery

Notes

On the Summary page of this window, you will be shown the settings of the lottery you are setting up. Carefully review these settings, even if you set this lottery up as a preset, since the lottery could have made changes to the lottery's parameters. If any changes need to be made, just keep hitting the "Back" button until you have reached the page where the settings need to be changed.

If you can find your lottery in dropdown menu on the first page, it is strongly recommended that you use the "built-in" lottery. *Only built-in lotteries are eligible for the optional online updating service ("Lottery Data Subscription Service").* Please note that you are limited to 64 "built-in" lotteries at any one time.

The "Refresh" button will go online and automatically download any new or updated lottery setup settings that have been added or changed since the latest release of Lotto Sorcerer.

Edit Lottery Settings

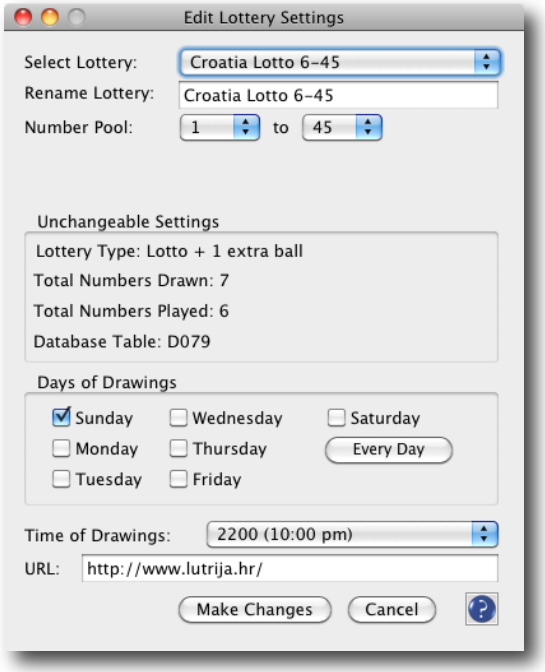


Figure 14.

Overview

This lets you edit some of the settings for a lottery you have already setup in Lotto Sorcerer.

How to Invoke

Use the menu item "Lottery Structure > Edit Lottery Settings".

Basic Procedure

1. Select the lottery that you want to edit in the "Select Lottery" dropdown menu
2. Change the name of the lottery, if desired
3. Make any required changes to the pool sizes
4. Select the days of week that the drawings of the lottery occurs
5. Select the time of the drawing
6. Click the "Make Changes" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to edit.

Rename Lottery text box

If you want to change the name of the lottery, enter the new name here. You are limited to 50 characters.

Number Pool dropdown menus

Use these dropdown menus to change the pool sizes for the lottery.

Unchangeable Settings box

This box lists settings of the lottery that cannot be changed. If these settings need to be changed, then the entire statistics of the lottery has been changed; so you must set it up as a separate, different lottery.

Days of Drawings check boxes

Select the days of the week for which this lottery has drawings.

Time of Drawings check boxes

Select the time of the drawings from this box. This value does not have to be exact. It is used only when using Virtual Lotteries, and is used to determine the order of drawings throughout the day.

URL text box

Enter the URL (website address) of the lottery. The URL must begin with either "`http://`" or "`https://`".

Make Changes button

Click this to make changes to this lottery.

Cancel button

Use this to close the current window and return to the Main window without making changes.

Notes

Virtual lotteries cannot be edited with this function, and will not appear in the "Select Lottery" dropdown. You cannot edit member lotteries of virtual lotteries either until you delete the parent virtual lottery.

Delete Lottery

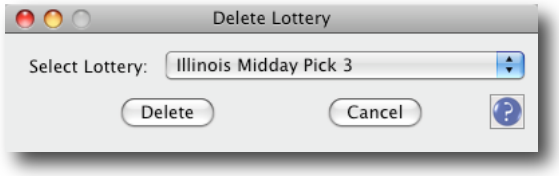


Figure 15.

Overview

This lets you delete a lottery from the Lotto Sorcerer database.

How to Invoke

Use the menu item "Lottery Structure > Delete Lottery".

Basic Procedure

1. Select the lottery that you want to delete in the "Select Lottery" dropdown menu
2. Click the "Delete" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to delete.

Delete button

Click this button to delete this lottery from the database.

Cancel button

Use this to close the current window and return to the Main window without deleting any lotteries.

Note

Virtual lotteries cannot be deleted with this function, and will not appear in the "Select Lottery" dropdown. You cannot edit member lotteries of virtual lotteries either until you delete the parent virtual lottery.

Virtual Lottery Setup Wizard

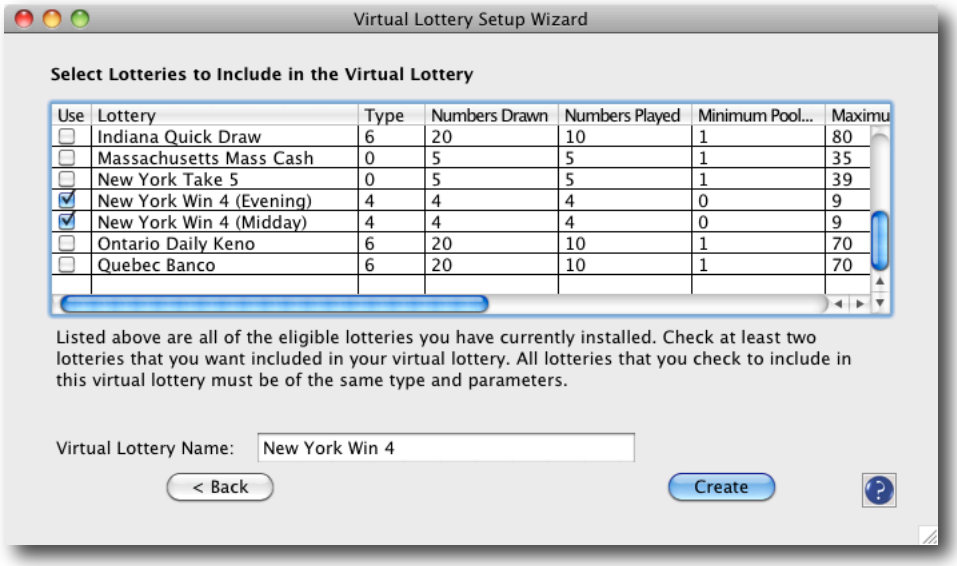


Figure 16.

Overview

In general, Lotto Sorcerer can only handle lotteries that play once per day. However, if you want to generate suggestions for a lottery that plays more than once a day, you can join the separate lotteries into one "virtual lottery".

Take, for example, Figure 15, which shows the "New York Win 4" midday and evening lotteries. These lotteries can be joined together as one virtual lottery for suggestion and analysis purposes.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Virtual Lottery Setup Wizard".

Basic Procedure

1. Select the lotteries that you will want to include in the virtual lottery
2. Give the virtual lottery a name
3. Click the "Create" button

Window Controls

Select Lottery to Include list

Check the lotteries you want to include in the virtual lottery. Only real lotteries that are not already members of a virtual lottery are listed.

Virtual lottery members *must be of the same type and parameters.* For example, you cannot create a virtual lottery by combining a Pick-type lottery with a keno-type lottery.

Virtual Lottery Name text box

You need to give the virtual lottery a meaningful name. Lotto Sorcerer will try to guess the name, based on your selections from the "Select Lottery" list.

Back button

Click this button to return to the home page of the Virtual Lottery Setup Wizard.

Create button

Use this button to create the virtual lottery.

Delete Virtual Lottery

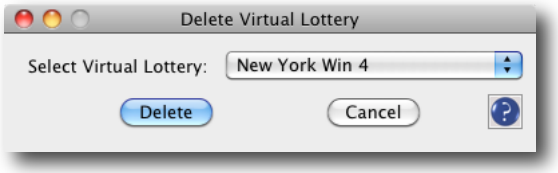


Figure 17.

Overview

This lets you delete a virtual lottery.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Delete Virtual Lottery".

Basic Procedure

1. Select the virtual lottery that you want to delete in the "Select Virtual Lottery" dropdown menu
2. Click the "Delete" button

Window Controls

Select Virtual Lottery dropdown

Use this dropdown to select the virtual lottery that you want to delete.

Delete button

Click this button to delete this virtual lottery. Member lotteries of the virtual lottery are not affected.

Cancel button

Use this to close the current window and return to the Main window without deleting any lotteries.

Show Virtual Lottery Children

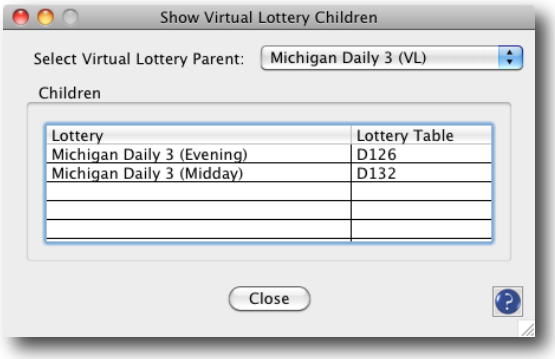


Figure 18.

Overview

This shows all children of a virtual lottery.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Virtual Lottery Utilities > Show Virtual Lottery Children".

Basic Procedure

- Select the virtual lottery in the "Select Virtual Lottery Parent" dropdown menu

Window Controls

Select Virtual Lottery Parent dropdown

Use this dropdown to select the virtual lottery parent.

Children list

This list shows the children of the virtual lottery parent that you selected. *If only one child is shown, then that would be an error condition,* because a virtual lottery ("parent") must consist of at least two lotteries ("children") . In that case, you should delete the parent virtual lottery.

Close button

Use this to close the current window.

Show Virtual Lottery Orphans

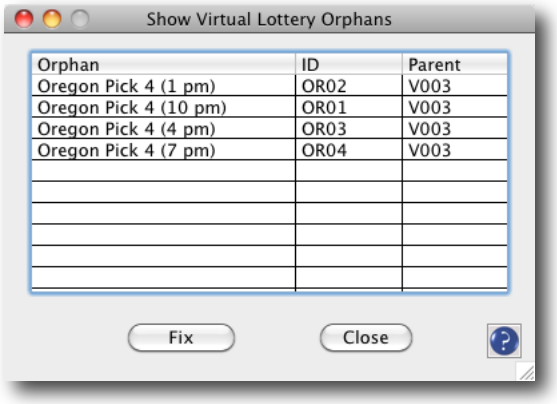


Figure 19.

Overview

This shows all virtual lottery orphans. A "virtual lottery orphan" is a lottery which has a parent virtual lottery listed, but that parent lottery does not exist. Ordinarily, this should never happen; but an erroneous SQL statement in the SQL Interface by the user could cause this situation.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Virtual Lottery Utilities > Show Virtual Lottery Orphans".

Window Controls

Orphan list

This list shows any and all virtual lottery orphans present in the database.

Fix button

Use this to remove the virtual lottery orphans. Note that this does not delete the lotteries, but only removes the marker which signifies these lotteries as children of a virtual lottery.

Close button

Use this to close the current window.

Show Virtual Lottery Parents

Figure 20.

Overview

This shows all parents of all virtual lottery children.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Virtual Lottery Utilities > Show Virtual Lottery Parents".

Window Controls

Child/Parent list

The left column shows the child virtual lottery, and the right column show the parent. Each parent has at least two children, and each child has only one parent.

Close button

Use this to close the current window.

Show Virtual Lottery Siblings

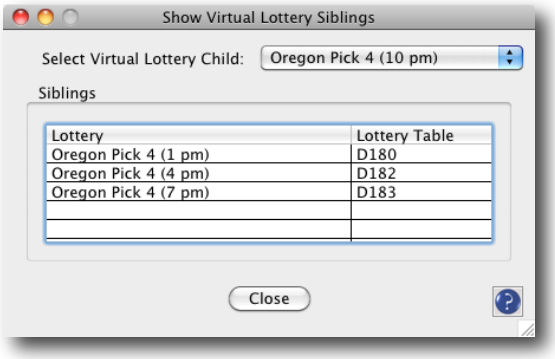


Figure 21.

Overview

This shows all siblings of a virtual lottery child.

How to Invoke

Use the menu item "Lottery Structure > Virtual Lotteries > Virtual Lottery Utilities > Show Virtual Lottery Siblings".

Basic Procedure

- Select the virtual lottery child in the "Select Virtual Lottery Child" dropdown menu

Window Controls

Select Virtual Lottery Child dropdown

Use this dropdown to select the virtual lottery child whose siblings you wish to see..

Siblings list

This list shows the siblings of the virtual lottery you selected. *If no siblings is shown, then that would be an error condition,* because each virtual lottery ("parent") must have at least two children. In that case, you should delete the parent virtual lottery.

Close button

Use this to close the current window.

Lottery Data

Clear Lottery

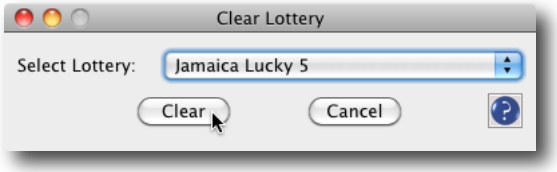


Figure 22.

Overview

This function will delete all drawings from the current selected lottery. Note that this function cannot be undone.

How to Invoke

Use the menu item "Lottery Data > Clear Lottery".

Basic Procedure

1. Choose the lottery you want to clear in the "Select Lottery" dropdown menu
2. Click the "OK" button to clear the lottery

Window Controls

Select Lottery dropdown menu

Use this dropdown to select the lottery that you want to clear.

Clear button

Click this to clear the data from the lottery's table.

Cancel button

This button will close the window without clearing any data.

Clear Virtual Lottery

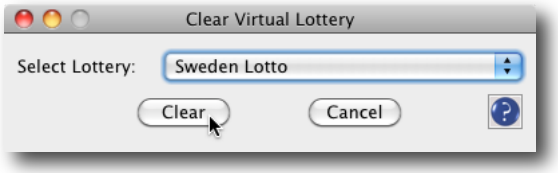


Figure 23.

Overview

This function will delete all drawings from the current selected virtual lottery. Note that this function cannot be undone. This function will not affect the member lotteries of this virtual lottery in any way.

How to Invoke

Use the menu item "Lottery Data > Clear Virtual Lottery...".

Basic Procedure

1. Choose the virtual lottery you want to clear in the "Select Lottery" dropdown menu
2. Click the "Clear" button to clear the virtual lottery

Window Controls

Select Lottery dropdown menu

Use this dropdown to select the virtual lottery that you want to clear.

Clear button

Click this to clear the data from the virtual lottery's table.

Cancel button

This button will close the window without clearing any data.

Force Virtual Lottery Refresh

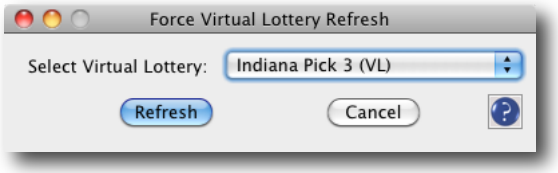


Figure 24.

Overview

This function will update the virtual lottery with all of the drawings from the member lotteries of the virtual lottery. Whenever you choose a virtual lottery in the Main Window, this is automatically accomplished. But this function is useful if you are working outside of the Main Window, and need to ensure that the virtual lottery is updated.

How to Invoke

Use the menu item "Lottery Data > Force Virtual Lottery Refresh...".

Basic Procedure

1. Choose the virtual lottery you want to update in the "Select Virtual Lottery" dropdown menu
2. Click the "Refresh" button to update the virtual lottery

Window Controls

Select Lottery dropdown menu

Use this dropdown to select the virtual lottery that you want to update.

Refresh button

Click this to update the virtual lottery.

Cancel button

This button will close the window.

Print Lottery Data Drawing History

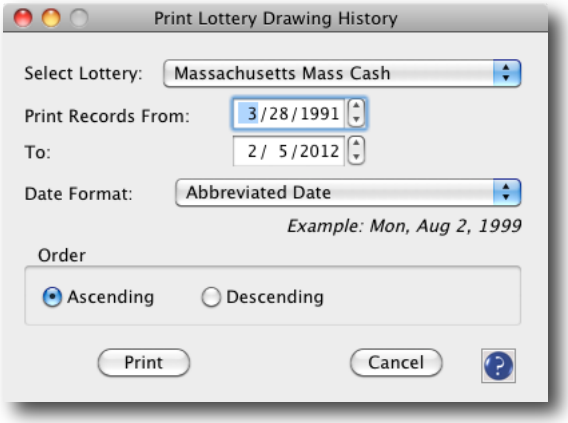


Figure 25.

Overview

This function will print the drawing history of the lottery of your choice.

How to Invoke

Use the menu item "Lottery Data > Print Lottery Drawing History".

Basic Procedure

1. Select the lottery you want print
2. Select the starting and ending dates
3. Choose the date format
4. Choose the ordering method
5. Click the Print button

Window Controls

Select Lottery dropdown menu

Use this dropdown to select the lottery that you want to print.

Print Records Starting Date dropdown menu

Select the starting date you want to use.

Print Records Ending Date dropdown menu

Select the ending date you want to use.

Date Format dropdown menu

Select the date format that you want to use in the printout. The date formats are defined by you system, and are user-configurable.

- For the Mac, choose "System Preferences > Language and Text > Formats"; click the "Customize..." button under Dates and redefine the date definitions.
- For Windows, choose "Control Panel > Clock, Language, and Region > Region and Language > Format" to redefine the date definitions.

Order radio buttons

Select whether you want the printout in ascending or descending format for the dates of the drawing.

Print button

Click this to start the printing process.

Cancel button

This button will close the window.

Prune Lottery

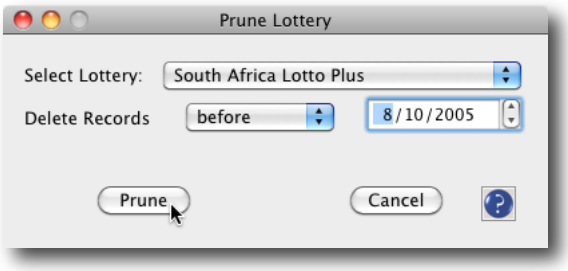


Figure 26.

Overview

This function will delete older drawings from the current selected lottery. Note that this function cannot be undone.

How to Invoke

Use the menu item "Lottery Data > Prune Lottery".

Basic Procedure

1. Select the lottery you want prune
2. Select the range of data you want to delete
3. Click the Prune button to begin the process

Window Controls

Select Lottery dropdown menu

Use this dropdown to select the lottery that you want to prune.

Before/after/between dropdown menu

Use this dropdown to select whether you want to delete records before, after or between the selected date(s).

Select Cutoff Date control(s)

Select the cutoff date.

Prune button

Click this to start the pruning process

Cancel button

This button will close the window.

Purge Lottery

Figure 27.

Overview

This function will delete older drawings from the current selected lottery where the drawings exceed the current parameters for that lottery. For example, if a lottery changed its drawing matrix from "1 to 59" to "1 to 55", this function will delete any drawings which contain numbers higher than what is allowed.

How to Invoke

Use the menu item "Lottery Data > Purge Lottery".

Basic Procedure

1. Select the lottery you want purge
2. Click the Purge button to purge the lottery

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to purge.

Purge button

Click this to start the purging process.

Cancel button

This button will close the window.

Import Lottery Data

Import Comma Separated Value (CSV) File

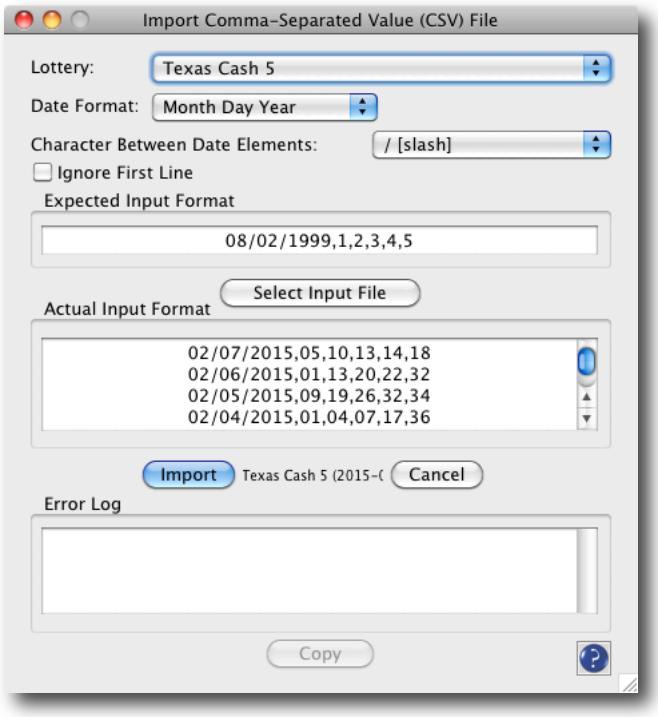


Figure 28.

Overview

This function lets you enter data from comma separated value files directly into Lotto Sorcerer. "Comma separated value files" are files where the data fields are separated by a comma. These files typically have an file extension of ".csv". Note that this expects the data to have the drawing date first, then the numbers drawn. If it is a bonus ball type lottery, it expects the final number(s) to be the bonus ball(s).

Any errors while importing will be displayed in the "Error Log" section at the bottom of this window.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import Comma Separated Value (CSV) File".

Basic Procedure

1. Set the input parameters of the comma separated value file
2. Select the input file
3. Click the "Import" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to import data to.

Date Format dropdown

This allows you to select how the drawing date format.

Character between Date Elements dropdown

This allows you to select the character that is between the date elements (month, day and year). The default is the dash ("-").

Ignore first line check box

Some files use the first line to describe the field layout. If your input file has this, check this box.

Expected Input Format box

Based on your previous selections, this box shows, using dummy data, how it expects the input file to look. Note that the tab character is represented as "[tab]".

Select Input File button

Clicking this button brings up a standard file selector. Choose the file you want to import.

Actual Input Format box

Based on the file you selected, this box shows an actual preview of the first few lines of that file. It is important that this box closely resembles what is in the "Expected Input Format" box.

Import button

Clicking this button imports the file you have chosen into Lotto Sorcerer's database.

Cancel button

Use this to close the current window and return to the Main window.

Import Delimited Text File

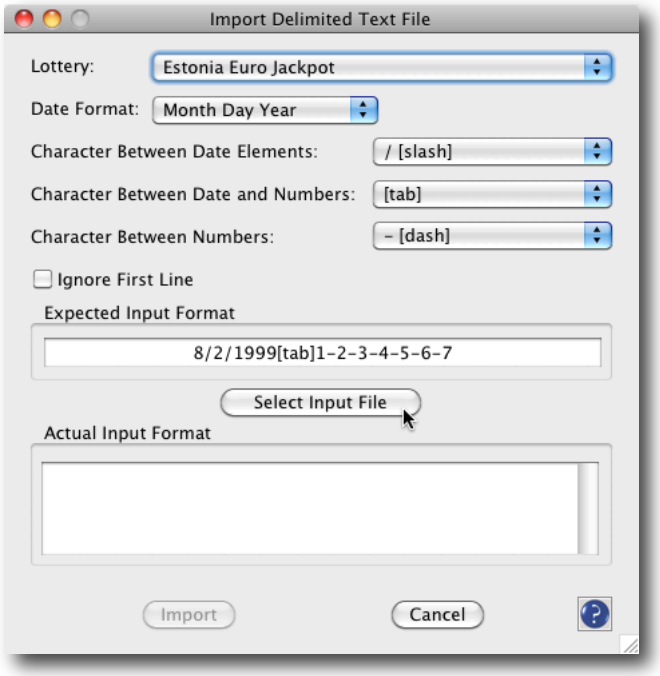


Figure 29.

Overview

This function lets you enter data from delimited text files directly into Lotto Sorcerer. "Delimited text files" are files where the data fields are separated by a specific character, such as a tab or comma. Note that this expects the data to have the drawing date first, then the numbers drawn. If it is a bonus ball type lottery, it expects the final number(s) to be the bonus ball(s).

The function can import a wide variety (over 52,000!) of different types of import files. *But the import file must perfectly match the import parameters you specify in order for this to work properly.*

Any errors while importing will be logged into the log file. Use the menu item "Utilities > File Viewer" and use the File Viewer to review.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import Delimited Text File".

Basic Procedure

1. Set the input parameters of the delimited text file
2. Select the input file
3. Click the "Import" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to import data to.

Date Format dropdown

This allows you to select how the drawing date format.

Character between Date Elements dropdown

This allows you to select the character that is between the date elements (month, day and year). The default is the slash ("/").

Character between Date and Numbers dropdowns

This allows you to select the character that is between the date and the first of the numbers drawn. The default is the tab character.

Character between Numbers dropdown

This allows you to select the character that is between the individual numbers drawn. The default is the dash ("-").

Ignore first line check box

Some delimited text files use the first line to describe the field layout. If your input file has this, check this box.

Expected Input Format box

Based on your previous selections, this box shows, using dummy data, how it expects the input file to look. Note that the tab character is represented as "[tab]".

Select Input File button

Clicking this button brings up a standard file selector. Choose the file you want to import.

Actual Input Format box

Based on the file you selected, this box shows an actual preview of the first few lines of that file. It is important that this box closely resembles what is in the "Expected Input Format" box.

Import button

Clicking this button imports the file you have chosen into Lotto Sorcerer's database.

Cancel button

Use this to close the current window and return to the Main window.

Import Fixed Width Text File

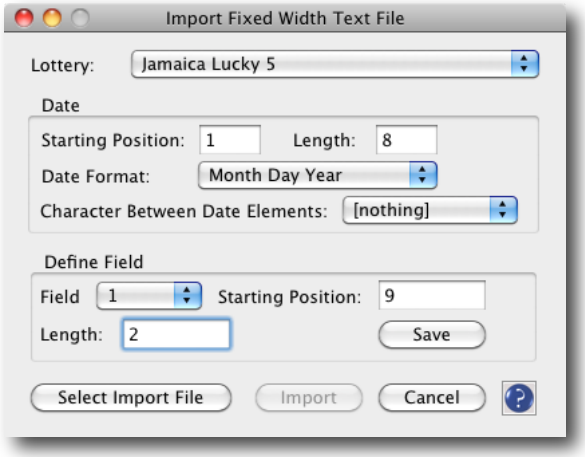


Figure 30.

Overview

This function lets you input data from space-padded, fixed width text files directly into Lotto Sorcerer. "Fixed width text files" are files where the data fields are at consistent positions and are at consistent lengths.

Any errors while importing will be logged into the log file. Use the menu item "Utilities > File Viewer" and use the File Viewer to review.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import Fixed Width Text File".

Basic Procedure

1. Select the lottery
2. Set the field parameters of the input file
3. Select the input file
4. Click the Import button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to import data to.

Date Field controls

Select the starting position and length of the date field, the date format, and the character between the date elements.

In the following example, we see the date field is the first eight characters in the line (the first two characters are the month, the third and fourth characters are the day of the month, and the fifth through eighth characters are the year. So the starting position is "1" and the length is "8".

```
01/18/20151511172230
01/11/20151508121720
01/04/20151504142228
12/28/20141403072634
```

Number Field controls

Select the starting position and length of the first number field, and click the "Save" button to fix that position into memory.

The following example shows a lottery that draws five numbers (each number having two digits). So the setting for Field #1 would have a starting position of "9", with a length of "2". Field #2 has a starting position of "11", with a length of "2", and so on.

```
01/18/2015151117223O
01/11/2015150812172O
01/04/2015150414222B
12/28/2014140307263B
```

After entering the starting position and length for a field, click the "Save" button to save that field's parameters in memory. Continue with every field for that lottery.

Select Input File button

Clicking this button brings up a standard file selector. Choose the file you want to import.

Import button

Clicking this button imports the file you have chosen into Lotto Sorcerer's database.

Cancel button

Use this to close the current window and return to the Main window.

Import Tab Delimited File

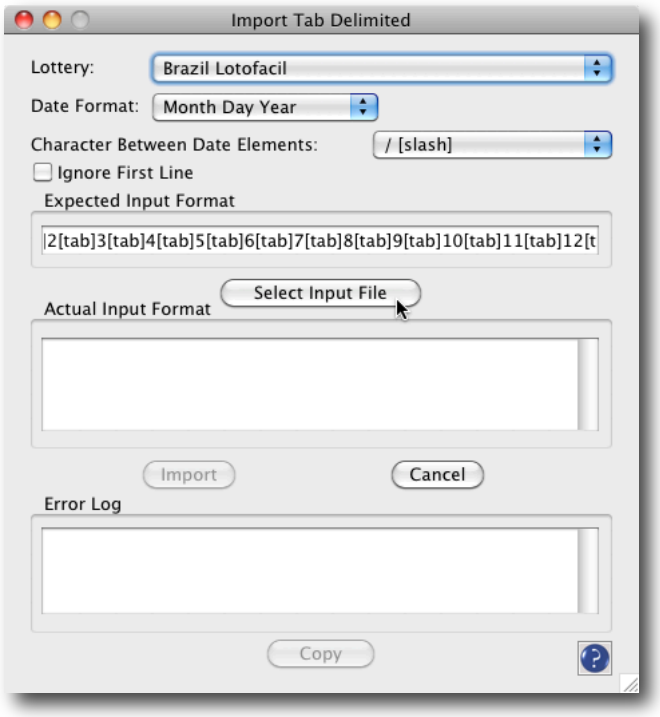


Figure 31.

Overview

This function lets you enter data from tab delimited files directly into Lotto Sorcerer. "Tab delimited files" are files where the data fields are separated by a tab. These files typically have an file extension of ".txt" or ".tab". Note that this expects the data to have the drawing date first, then the numbers drawn. If it is a bonus ball type lottery, it expects the final number(s) to be the bonus ball(s).

Any errors while importing will be displayed in the "Error Log" section at the bottom of this window.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import Tab Delimited File".

Basic Procedure

1. Set the input parameters of the tab delimited text file
2. Select the input file
3. Click the "Import" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to import data to.

Date Format dropdown

This allows you to select how the drawing date format.

Character between Date Elements dropdown

This allows you to select the character that is between the date elements (month, day and year). The default is the slash ("/").

Ignore first line check box

Some delimited text files use the first line to describe the field layout. If your input file has this, check this box.

Expected Input Format box

Based on your previous selections, this box shows, using dummy data, how it expects the input file to look. Note that the tab character is represented as "[tab]".

Select Input File button

Clicking this button brings up a standard file selector. Choose the file you want to import.

Actual Input Format box

Based on the file you selected, this box shows an actual preview of the first few lines of that file. It is important that this box closely resembles what is in the "Expected Input Format" box.

Import button

Clicking this button imports the file you have chosen into Lotto Sorcerer's database.

Cancel button

Use this to close the current window and return to the Main window.

Import File Inspector

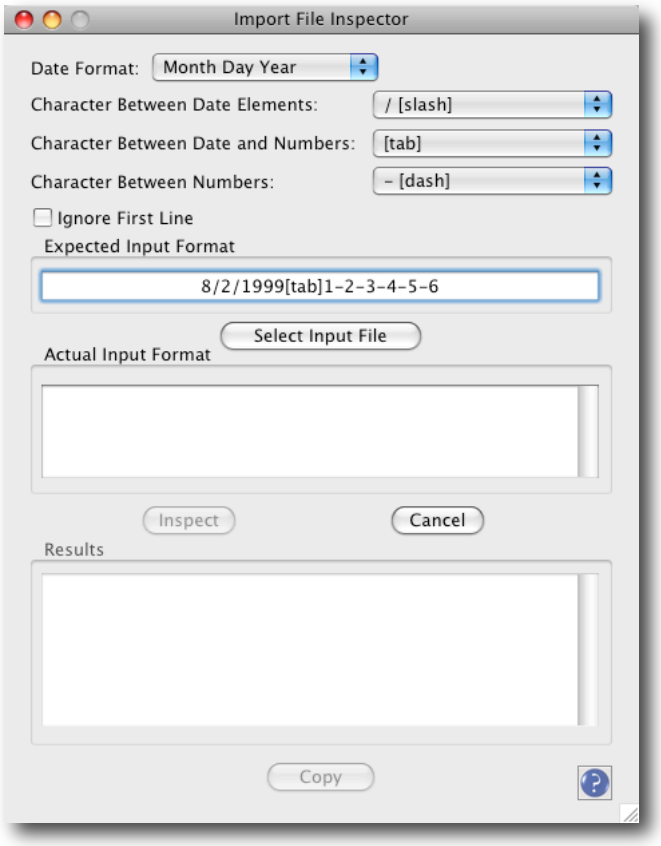


Figure 32.

Overview

The Import File Inspector is a useful tool that allows you to inspect a file that is a candidate for importation.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import File Utilities > Import File Inspector".

Basic Procedure

1. Set the input parameters of the comma separated value file
2. Select the input file
3. Click the "Inspect" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to import data to.

Date Format dropdown

This allows you to select how the drawing date format.

Character between Date Elements dropdown

This allows you to select the character that is between the date elements (month, day and year). The default is the dash ("-").

Date Prefixer

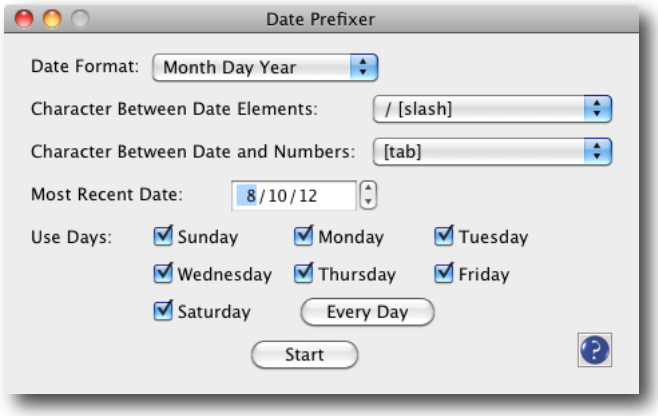


Figure 33.

Overview

This function will prefix dates to a text file containing only drawing data. For example, you may have past drawing data, but the dates are missing. This routine lets you enter the last date of the drawings, and will insert the draw dates.

☞ The specific draw dates are not important, *only the chronological sequence of drawings.*

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import File Utilities > Date Prefixer".

Basic Procedure

1. Select the date parameters
2. Set the last date of the drawings in the Most Recent Date control
3. Click the "Start" button

Window Controls

Date Format dropdown

This allows you to select how the drawing date format. The default is the month, followed by the day, followed by the year.

Character between Date Elements dropdown

This allows you to select the character that is between the date elements (month, day and year). The default is the slash ("/").

Character between Date and Numbers dropdowns

This allows you to select the character that is between the date and the first of the numbers drawn. The default is the tab character.

Most Recent Date calendar control

Select the date for the last drawing in the data (that is, the last entry).

Use Days checkboxes

Check the days on which the lottery holds drawings.

Every Day button

This selects every selection of the Use Days checkboxes.

Start button

When you click this button, you will be asked to select the input file and the output file.

DBF to TXT Converter

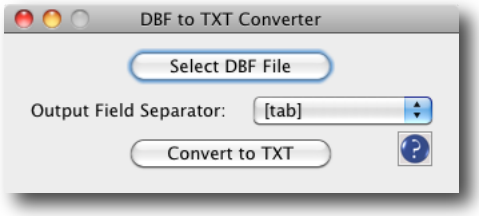


Figure 34.

Overview

This function converts .dbf (dBase™) data files to .txt (text) files.

☞ Not all types of .dbf files are supported.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import File Utilities > DBF to TXT Converter".

Basic Procedure

1. Select the .dbf input file
2. Set the output field separator
3. Click the "Convert to TXT" button

Window Controls

Select DBF File button

This invokes a standard file selector. Choose the .dbf file you want to convert.

Output Field Separator dropdown

This allows you to select the character that is between the fields in the output file. The default is the tab character.

Convert to TXT button

This invokes a standard file selector. Select the name and location of the output file.

Field Stripper

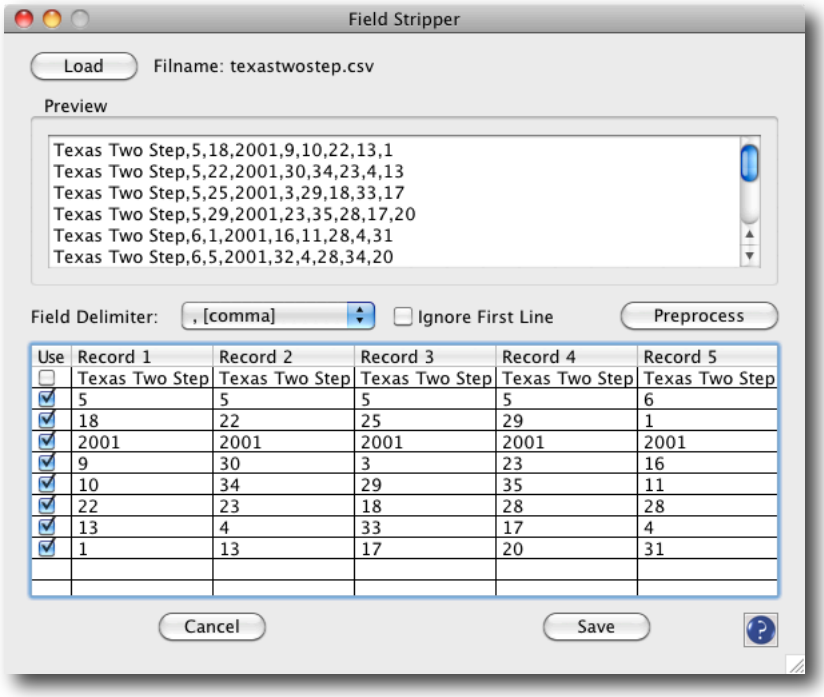


Figure 35.

Overview

Lotto Sorcerer's import file utilities insist that the input file contains *only* the date of the drawing and the numbers drawn. Yet many data files provided by lotteries contain extraneous fields (for example, the name of the lottery [as shown above], the drawing number, the jackpot value, the number of winners, etc.). This utility, *Field Stripper,* helps remove the unnecessary fields. You can then use the modified file, with the unneeded fields removed, in any of Lotto Sorcerer's import data file utilities.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import File Utilities > Field Stripper".

Basic Procedure

1. Load the input file
2. Select the field delimiter
3. Click the "Preprocess" button
4. Check the fields that you want included in the output file
5. Click the "Save" button

Window Controls

Load button

Use this button to save the file that you want to load. After selecting the file, the first few lines will be shown in the Preview area.

Field Delimiter dropdown menu

This utility needs to know what character is used to delineate the different fields. This control allows you to select the delimiter. The default is the comma (",") for comma-separated value (CSV) files.

Ignore First Line checkbox

If the first line in the input file contains field descriptors instead of data, check this box.

Preprocess button

Click this button to populate the fieldmap list at the bottom half of the window. This area will show the first few records, with the fields separated.

Fieldmap List

Check the fields you wish to have in the output file. For fields you want stripped, leave the checkbox blank. In the example shown above, the first field, which contains the name of the lottery, is unchecked and will not be in the output file.

Save button

When you click this button, a standard file selector will appear, allowing you to choose the name and location of the output file.

Data Source Editor

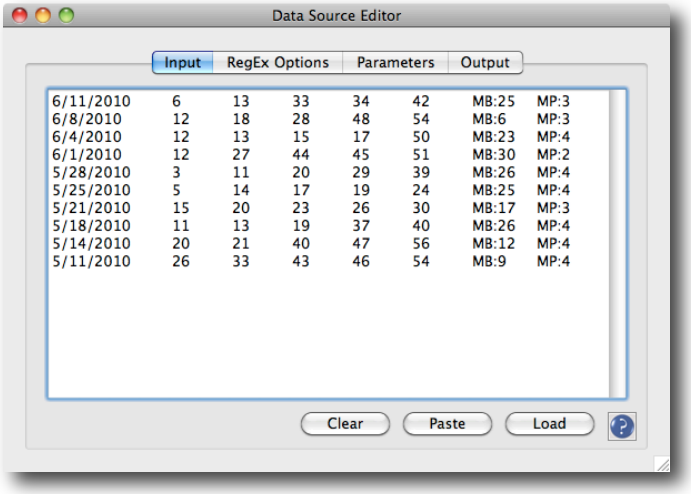


Figure 36.

Overview

The Data Source Editor is a powerful tool for manipulating data for importing into Lotto Sorcerer. Al-though Lotto Sorcerer's importing functions can handle a wide variety of input formats, still, most of the files from lottery websites may still require some manipulation.

Invisible Encoded (Control) Character Handling

This Data Source Editor has tools that you will not be able to find on standard text editors. For example, you can search for and replace invisible encoded characters, such as the tab character and the line feed character.

Regular Expressions

Even more impressive, you can perform Regular Expression searches. Think of Regular Expressions as a powerful form of wildcards. Regular search and replace requires you to type in exactly what you are searching for (for example, the date "1999/08/02"). But what if you wanted to search for *all* dates, not just a specific date? This is where Regular Expressions come in. It is far beyond the scope of this User's Guide to be a tutorial on Regular Expressions. If you are interested, there are many good books available on this powerful tool.

Multiple Pass

This Data Source Editor lets you easily copy the output back to the input, in the likely event that you need to do multiple search and replace procedures to the data source "just right".

How to Invoke

Use the menu item " Lottery Data > Import Lottery Data > Import File Utilities > Data Source Editor".

Basic Procedure

1. Enter, paste, load or "drag and drop" the text you want to parse into the Input box
2. If you are using Regular Expressions, choose the RegEx options in the RegEx Options tab
3. Choose either Regular Expressions or standard Search-and-Replace in the Parameters tab, then click the Process button
4. Send the Output back to the Input, if necessary, or save or copy the results in the Output tab.

Window Controls

Input Tab

Input text box

Enter, paste or load the text you want to parse in this box.

Clear button

This button clears the Input text box.

Paste button

This button pastes any text in the System Clipboard into the Input text box.

Load button

This button opens up a standard file selector, letting you choose a text file for the Input text box.

RegEx Options Tab

Case Sensitive check box

Check this box to treat uppercase and lowercase letters as different characters.

Dot Matches All check box

Check this box to make the dot character match all characters, including line break characters.

Greedy check box

Uncheck this box if you want quantifiers to be lazy, effectively making .* the same as .*?.

Match Empty Strings check box

Uncheck this box if you want to skip zero-length matches.

Replace All Matches check box

Check this box if you want the engine to search-and-replace all regex matches in the subject string rather than just the first one.

String Begins = Line Begins check box

Uncheck this box if you do not want the start of the string to be considered the start of the line. This can be useful if you're processing a large chunk of data as several separate strings, where only the first string should be considered as starting the (conceptual) overall string.

String Ends = Line Ends check box

Uncheck this box if the string you are passing to the Search method is not really the end of the whole chunk of data you're processing.

Treat Target as One Line check box

Check this box to make the caret-and-dollar match at the start and the end of the string only. By default, they will also match after and before embedded line breaks.

Line End pop up menu

Select the line ending character used in the text you are parsing.

Parameters Tab

Regular Expression/Standard Search-and-Replace radio button

Choose the method you want to use.

Encoded Control Characters Checkbox

If you are using the Standard Search-and-Replace mode, and if you want to search for and/or replace encoded control characters, check this box.

If this option is checked, you can search for seven control characters. Just prefix the appropriate control character with a backslash ("\"). Note that these encodings are case sensitive.

Enter this	To search for/replace	ASCII code
\r	Carriage Return	13
\n	Line feed	10
\t	Tab	9
\f	Form feed	12
\a	Bell	7
	Backspace	8
\e	Escape	27
\\	Backslash	92

Search for/RegEx text box

Enter or paste the regular expression or search term here.

Replace With text box

If you chose "Standard Search and Replace", enter or paste the regular expression or search term here.

Process button

Click this button to start the parsing process.

Output Tab

Output text box

This contains the results of the parsing process.

Send to Input button

This sends the output (results) back to the input box so that you can further parse the text.

Clear button

This button clears the Output box.

Copy button

This button copies the output to the System Clipboard.

Save button

This opens up a standard file selector, allowing you to save the output (results) to a file on your computer.

Space Remover

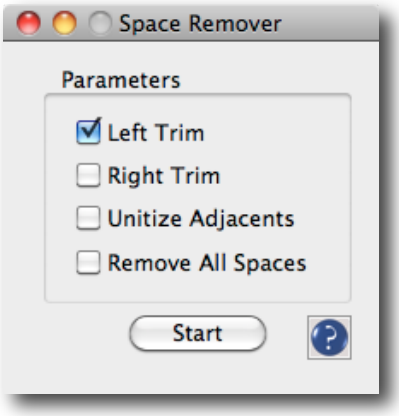


Figure 37.

Overview

Use this function to manipulate space characters (ASCII 32) in a data source file.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Import File Utilities > Space Remover".

Basic Procedure

1. Choose the options you want
2. Click the "Start" button

Window Controls

Left Trim checkbox

Choosing this will remove all spaces from the front of each line.

Right Trim checkbox

Choosing this will remove all spaces from the end of each line.

Unitize Adjacents checkbox

This will change all instances of two or more adjacent spaces into one space.

Remove All Spaces checkbox

This removes all spaces from the file.

Start button

When you click this button, you will be prompted to select and input and output file using standard file dialog boxes.

Purchase Lottery Data for Importing

Overview

This function takes you (if you have an internet connection) to our webpage where you can purchase datasets of prior lottery drawings. Although many lotteries allow you to download prior drawings for free, we add value to this data by making it easy to import into Lotto Sorcerer.

How to Invoke

Use the menu item "Lottery Data > Import Lottery Data > Purchase Lottery Data for Importing".

Basic Procedure

US Datamines will email the lottery datasets to you. Upon receipt of these files, the basic procedure is:

1. Save the attachments to your hard drive
2. Use Lotto Sorcerer's menu item Lottery Data > Import Lottery Data > Delimited Text File
3. Make sure the settings in the Import Delimited Text File match the import file
4. Choose the import file (you saved in step 1) by clicking the "Select Import File" button in the Import Delimited Text File window
5. Click the Import button

For details on using the Import Delimited Text File window, see Import Delimited Text File (page 64).

Export Lottery Data

Export as CSV

Figure 38.

Overview

This function lets you export data from Lotto Sorcerer into a comma separated value (CSV) file, where all fields are separated by a comma. These files typically have a file extension of ".csv". This is useful for backing up data, importing the data into an SQL database, as well as sharing data with another user of Lotto Sorcerer. The file you export can be easily imported back into the program.

How to Invoke

Use the menu item "Lottery Data > Export Lottery Data > Export as Comma Separated Value (CSV) File".

Basic Procedure

1. Select the lottery you want to export
2. Choose the export parameters
3. Click the Export button

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Date Format dropdown

Choose the exact date format you want to use.

Character Between Date Elements dropdown

Choose the character that separates the year, month and day parts in the date field.

End-of-Line Terminator dropdown

Different operating systems use different characters to mark the end of line. Choose the appropriate one.

Export button

Clicking this button starts the export process.

Cancel button

Use this to close the current window and return to the Main window.

Export as Delimited Text File

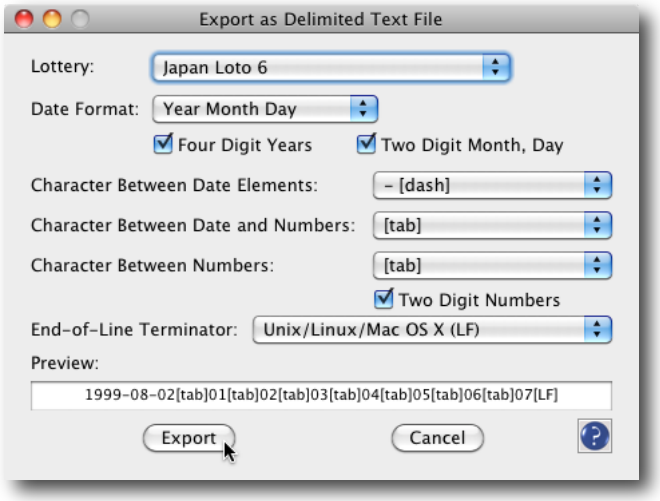


Figure 39.

Overview

This function lets you export data from Lotto Sorcerer into a delimited text file. This is useful for backing up data, importing the data into an SQL database, as well as sharing data with another user of Lotto Sorcerer. The file you export can be easily imported back into the program. This function will let you export your data in well over 620,000 different formats!

How to Invoke

Use the menu item "Lottery Data > Export Lottery Data > Export as Delimited Text File".

Basic Procedure

1. Select the lottery you want to export
2. Choose the export parameters
3. Click the Export button

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Date Format dropdown

Choose the exact date format you want to use.

Four Digit Years checkbox

If checked, the data will be exported as four-digit years (for example, "2005"); if unchecked only the last two digits are exported (for example, "2005" will be exported as "05"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Two Digit Years Month, Day

If checked, the data will be exported as two-digit months and days (for example, "01/09/2013"); if unchecked, these values will have only one digit (for example, "1/9/2013"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Character Between Date Elements dropdown

Choose the character that separates the year, month and day parts in the date field.

Character Between Date and Numbers dropdown

Choose the character that separates the date field and the number fields.

Character Between Numbers dropdown

Choose the character that separates the different number fields.

Two Digit Numbers

If checked, single digit numbers will be zero padded. For example, "1-2-5-16-19" will be exported as "01-02-05-16-19". If you chose "[nothing] as the character between numbers, this will be checked automatically.

End-of-Line Terminator dropdown

Different operating systems use different characters to mark the end of line. Choose the appropriate one.

Export button

Clicking this button starts the export process.

Cancel button

Use this to close the current window and return to the Main window.

Export as SQL File

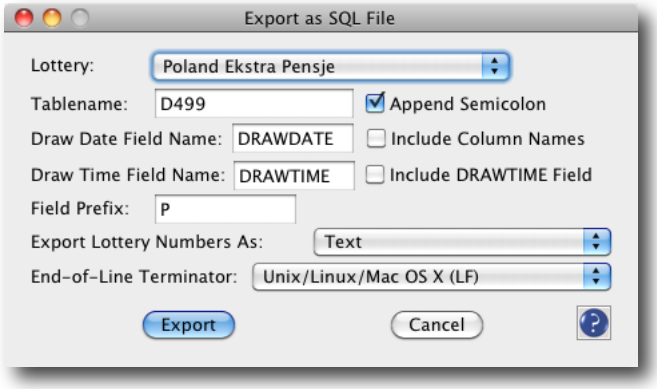


Figure 40.

Overview

This function lets you export data from Lotto Sorcerer into a SQL formatted text file. This is intended for importing the data into an SQL-92 compliant database, as well as sharing data with another user of Lotto Sorcerer. You can also use this to export data back into Lotto Sorcerer.

How to Invoke

Use the menu item "Lottery Data > Export Lottery Data > Export as SQL File".

Basic Procedure

1. Select the lottery you want to export
2. Enter the tablename to be used in the SQL file's "insert into..." clause
3. Enter the date field name
4. Enter the time field name
5. Choose optional parameters
6. Enter the field prefix to be used
7. Choose the End-of-Line terminator character(s) to use

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Tablename text box

Enter the tablename that will be used in the SQL file (in the "insert into..." clause).

Append Semicolon check box

If your database expects a semicolon after each SQL statement (most do), check this box.

Draw Date field name text box

If you are using the "Include Column Names" option, enter the field name for the date field.

Draw Time field name text box

If you are using the "Include Column Names" and the "Include DRAWTIME Field" options, enter the field name for the time field.

Field Prefix text box

Enter the field prefix to be used for the fields containing the numbers drawn. Lotto Sorcerer will automatically append field numbers (from 1 to the number of drawing fields).

Include Column Names checkbox

If you wish to use the field names in the "insert into..." clause, check this box.

End-of-Line Terminator dropdown

Different operating systems use different characters to mark the end of line. Choose the appropriate one.

Export button

Clicking this button starts the export process.

Cancel button

Use this to close the current window and return to the Main window.

Export as Tab Delimited File

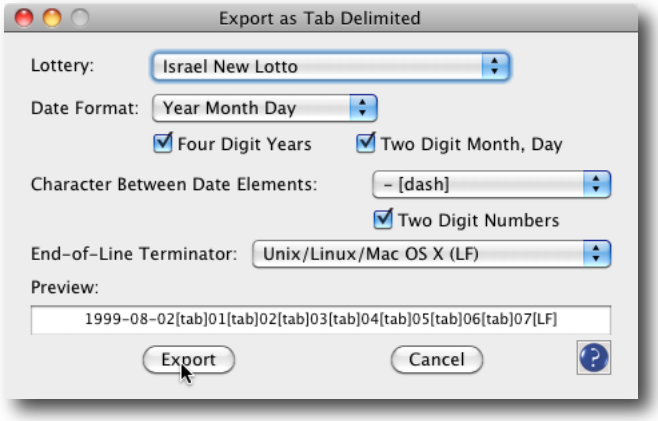


Figure 41.

Overview

This function lets you export data from Lotto Sorcerer into a tab delimited text file. This is useful for backing up data, importing the data into an SQL database, as well as sharing data with another user of Lotto Sorcerer. The file you export can be easily imported back into the program.

How to Invoke

Use the menu item "Lottery Data > Export Lottery Data > Export as Tab Delimited File".

Basic Procedure

1. Select the lottery you want to export
2. Choose the export parameters
3. Click the Export button

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Date Format dropdown

Choose the exact date format you want to use.

Four Digit Years checkbox

If checked, the data will be exported as four-digit years (for example, "2005"); if unchecked only the last two digits are exported (for example, "2005" will be exported as "05"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Two Digit Years Month, Day

If checked, the data will be exported as two-digit months and days (for example, "01/09/2013"); if unchecked, these values will have only one digit (for example, "1/9/2013"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Character Between Date Elements dropdown

Choose the character that separates the year, month and day parts in the date field.

Two Digit Numbers

If checked, single digit numbers will be zero padded. For example, "1-2-5-16-19" will be exported as "01-02-05-16-19".

End-of-Line Terminator dropdown

Different operating systems use different characters to mark the end of line. Choose the appropriate one.

Export as Microsoft Excel Spreadsheet

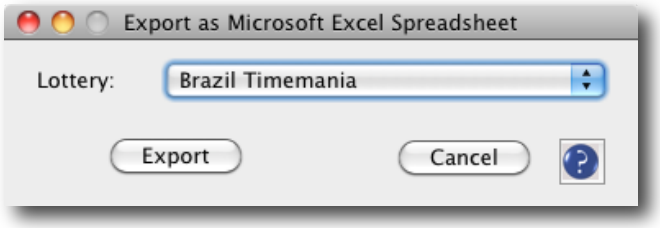


Figure 42.

Overview

This function lets you export data from Lotto Sorcerer into a Microsoft Excel spreadsheet. This uses default settings from Preferences (Miscellany tab).

How to Invoke

Use the menu item "Lottery Data > Export Lottery Data > Export as Microsoft Excel Spreadsheet".

Basic Procedure

1. Select the lottery you want to export
2. Click the Export button

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Export button

Click this button to begin the export process.

Subscriptions

Subscription Overview

If you setup a built-in lottery using the Lottery Setup Wizard, you may be able to keep your lottery database up-to-date with minimal (or even no) effort on your part by using our optional Lottery Database Subscription Service.

Requirements for Online Updating

- You must have an active subscription
- You must have unencumbered Internet access
- You must have an active PayPal account

How to Subscribe

Use menu item "Lottery Data > Lottery Data Subscription > Start Subscription". You will be taken to a website to sign up. New subscribers are eligible for a free, two-week trial subscription, so you can try it to make sure that it works. You can cancel anytime, and if you cancel before the two-week trial period is up, you will not be charged. Subscription costs are only a few dollars per month.

How to Cancel Your Subscription

Just log into your PayPal account, find the original transaction where you signed up, and click the "Cancel" button.

How it Works

When you click the "Update Data" button in the "Drawing History" tab of the Main Window, Lotto Sorcerer will download data for the lottery you are working with, from the date of the last drawing in your current database up to the last drawing we have on our servers.

Note

The only lotteries that can be set up are Lotto Sorcerer's built-in lotteries. These are the lotteries listed in the dropdown menu of the first page of the Lottery Setup Wizard.

Cancelling a Subscription

In order to cancel your subscription, you will need to:

1. Login to your PayPal account.
2. Find the transaction where you signed up for the subscription (you should be able to find this in the History tab on PayPal).
3. Click the Cancel button.

Note

It is important that you find the transaction where you signed up, not where you made a payment towards the subscription. If you do the latter, you will not see a "Cancel" button. However, there will be a link on that webpage to the original subscription.

Check Network Status

Figure 43.

Overview

This lets you do a quick check of the remote network status.

How to Invoke

Use the menu item "Lottery Data > Lottery Data Subscription > Check Network Status..."

Basic Procedure

- Click the "Check" button

Window Controls

Check button

Click this button to check the network status.

Cancel button

Use this to close the current window.

Check Subscription Status

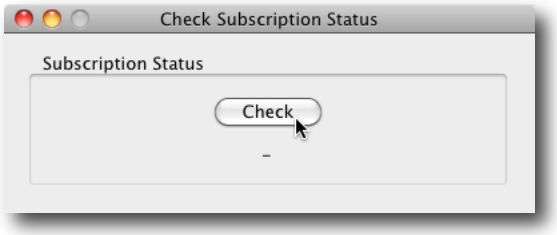


Figure 44.

Overview

This lets you check the subscription status for the optional Lottery Data Subscription Service. Once you click the "Check" button, your subscription status will show in this window, as well at the serial number to which this subscription is tied.

There are four states to your Lottery Data Subscription Status:

NEVER SUBSCRIBED
ACTIVE
CANCELLED - subscription will still work until it switches over to "EXPIRED" at the end of the term.
EXPIRED

How to Invoke

Use the menu item "Lottery Data > Lottery Data Subscription > Check Subscription Status..."

Basic Procedure

- Click the "Check" button

Window Controls

Check button

Click this button to check the subscription status.

Cancel button

Use this to close the current window.

Get Subscription ID

Figure 45.

Overview

If you are a subscriber to the optional Lottery Data Subscription Service, this function lets you retrieved the Subscription ID that is tied to your installation of Lotto Sorcerer.

Please note that this will not work if Lotto Sorcerer version 9 is using the subscription for your Lotto Sorcerer version 8 installation.

How to Invoke

Use the menu item "Lottery Data > Lottery Data Subscription > Get Subscription ID..."

Basic Procedure

- Click the "Get" button

Window Controls

Get button

Click this button to retrieve the Subscription ID.

Copy button

If the Subscription ID is found, you can copy it to the System Clipboard by clicking this button.

Subscription Troubleshooter

This is an online utility, which will help to quickly track down issues you may be having in using the Lottery Data Subscription Service.

How to Invoke

To go to the Subscription Troubleshooter, use menu item "Lottery Data > Lottery Data Subscription > Subscription Troubleshooter".

Tools

Lotto Augur

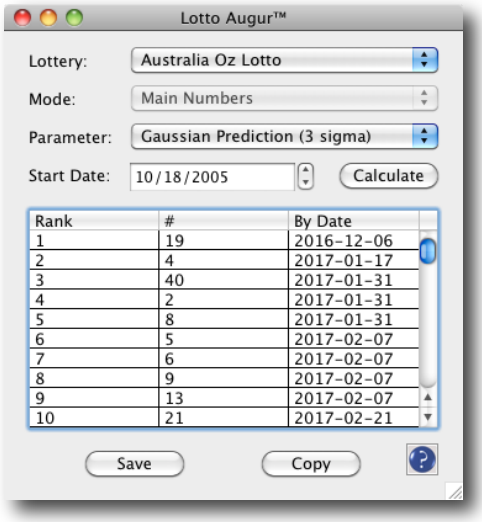


Figure 46.

Overview

This function lets you check for various statistics to the selected lottery. As the name of the tool implies, this works only with lotto-type lotteries.

How to Invoke

Use the menu item "Tools > Lotto Augur...".

Window Controls

Select Lottery dropdown menu

Choose the lottery with which you wish to work.

Mode dropdown menu

If your lottery is a bonus-type lottery, you can choose from analyzing either the main numbers or the bonus number(s). If you lottery is not a bonus-type lottery, this menu will be ghosted and unavailable.

Parameter dropdown menu

Depending on the lottery, you have up to twelve parameters from which to choose:

1. **Numbers Frequency**: this shows each number, how many times it has been drawn (in the "Count") column, and when the last time this number, in draws, has been drawn (in the "Age") column.
2. **Most Common Numbers**: this lists, in descending order, the most common numbers that have been draw, how many times the number has been drawn, and the last time this number has been drawn.
3. **Least Common Numbers**: this lists, in ascending order, the least common numbers that have been draw, how many times the number has been drawn, and the last time this number has been drawn.
4. **Most Overdue Numbers**: this lists, in descending order the numbers that are the most overdue to be drawn, along with then number of times the numbers has been drawn, and the last time the number has been drawn.
5. **Most Common Pairs**: this lists the most common number pairs found in the drawings, followed by the number of times the pair have appeared. Because this counts occurrences for every single two-number combination, it can take quite a while. This function is not available for the bonus portion of bonus-type lotteries which pick only one bonus number.

6. **Most Common Consecutive Pairs**: this lists the most common consecutive ("back-to-back") number pairs found in the drawings, followed by the number of times the pair have appeared. "Back-to-back" means two sequential numbers (in numerical order). This function is not available for the bonus portion of bonus-type lotteries which pick only one bonus number.
7. **Most Common Triplets**: this lists the most common number triplets found in the drawings, followed by the number of times the triplets have appeared. Because this counts occurrences for every single three-number combination, it can take quite a while. This function is not available for the bonus portion of bonus-type lotteries.
8. **Most Common Consecutive Triplets**: this lists the most common consecutive ("back-to-back-to-back") number triplets found in the drawings, followed by the number of times the triplets have appeared. "Back-to-back-to-back" means three sequential numbers (in numerical order). .This function is not available for the bonus portion of bonus-type lotteries.
9. **Odd versus Evens**: this shows all possible even-odd pairings, and how many times the pairings have occurred.
10. **Gaussian Prediction (1 sigma)**: this parameters ranks, in order, which number is expected by the earliest date, calculated as 1 sigma.
11. **Gaussian Prediction (2 sigma)**: this parameters ranks, in order, which number is expected by the earliest date, calculated as 2 sigma.
12. **Gaussian Prediction (3 sigma)**: this parameters ranks, in order, which number is expected by the earliest date, calculated as 3 sigma.

State Date selector

Choose the starting date here. When you select a lottery in the Lottery dropdown menu, this date selector will default to the oldest (first) drawing.

Calculate button

This starts the calculation process.

Save button

Clicking this button will invoke a file selector, allowing you to save the results to a text file.

Copy button

This copies the results to the System Clipboard.

Pick Lottery Augur

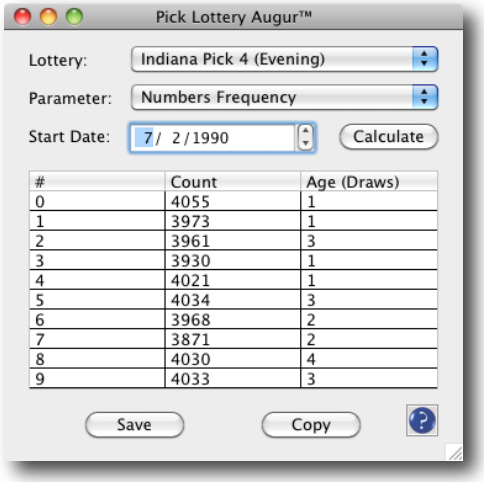


Figure 47.

Overview

This function lets you check for various statistics to the selected lottery. As the name of the tool implies, this works only with "pick lottery" type lotteries.

How to Invoke

Use the menu item "Tools > Pick Lottery Augur...".

Window Controls

Select Lottery dropdown menu

Choose the lottery with which you wish to work.

Parameter dropdown menu

You have nine parameters from which to choose:

1. **Numbers Frequency**: this shows each number, how many times it has been drawn (in the "Count") column, and when the last time this number, in draws, has been drawn (in the "Age") column.
2. **Most Common Numbers**: this lists, in descending order, the most common numbers that have been draw, how many times the number has been drawn, and the last time this number has been drawn.
3. **Least Common Numbers**: this lists, in ascending order, the least common numbers that have been draw, how many times the number has been drawn, and the last time this number has been drawn.
4. **Most Overdue Numbers**: this lists, in descending order the numbers that are the most overdue to be drawn, along with then number of times the numbers has been drawn, and the last time the number has been drawn.
5. **Most Common Pairs**: this lists the most common number pairs found in the drawings, followed by the number of times the pair have appeared. This parameter is not available for lotteries which draw only one bonus number.
6. **Most Common Consecutive Pairs**: this lists the most common consecutive ("back-to-back") number pairs found in the drawings, followed by the number of times the pair have appeared.
7. **Most Common Triplets**: this lists the most common number triplets found in the drawings, followed by the number of times the triplets have appeared.

8. **Most Common Consecutive Triplets**: this lists the most common consecutive ("back-to-back-to-back") number triplets found in the drawings, followed by the number of times the triplets have appeared.
9. **Odd versus Evens**: this shows all possible even-odd pairings, and how many times the pairings have occurred.

State Date selector

Choose the starting date here. When you select a lottery in the Lottery dropdown menu, this date selector will default to the oldest (first) drawing.

Calculate button

This starts the calculation process.

Save button

Clicking this button will invoke a file selector, allowing you to save the results to a text file.

Copy button

This copies the results to the System Clipboard.

Lottery Number Oracle

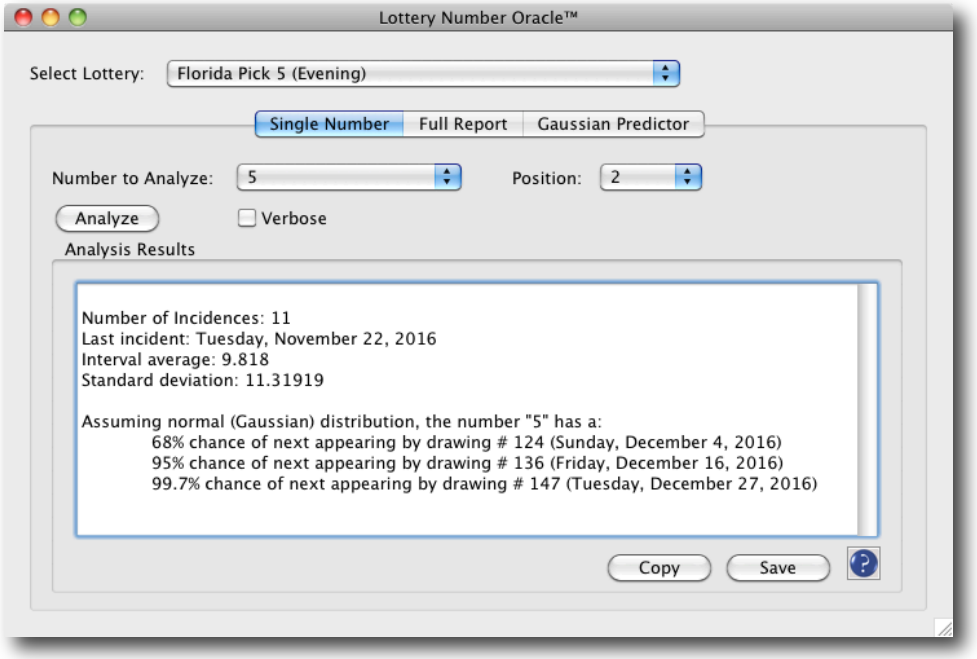


Figure 48.

Overview

Although the neural engines of Lotto Sorcerer look for non-random patterns to lottery draws, *Lottery Number Oracle* is designed to work with lotteries that are truly random.

This tool has two modes. The first, "Single Number", analyzes a single number; the second, "Full Report" analyzes all of the numbers; and the third, "Gaussian Predictor" sorts numbers by the best Gaussian (normal distribution) prediction by-date.

How to Invoke

Use the menu item "Tools > Lottery Number Oracle...".

Window Controls

Select Lottery dropdown menu

Select the lottery you want to analyze.

Single Number Mode

Number to Analyze dropdown menu

Select the number you want to analyze. You can analyze any number from the current number pool. If the lottery you chose has bonus ball(s), then these numbers will appear at the bottom of the dropdown menu.

Position dropdown menu

This dropdown menu is active only for "pick type" lotteries. For these lotteries, select the number position. Positioning is reckoned from left-to-right, so choosing "3" from a Pick 4 type lottery will analyze the third number from the left.

Verbose checkbox
If checked, the analysis will list every instance of where the number you are analyzing has appeared.

Analyze button
Clicking this button starts the analysis process.

Analysis Results text box
Clicking this button starts the analysis process.

Copy button
Clicking this button copies the Analysis results to the System Clipboard.

Save button
Clicking this button saves the Analysis results to a text file.

Full Report Mode

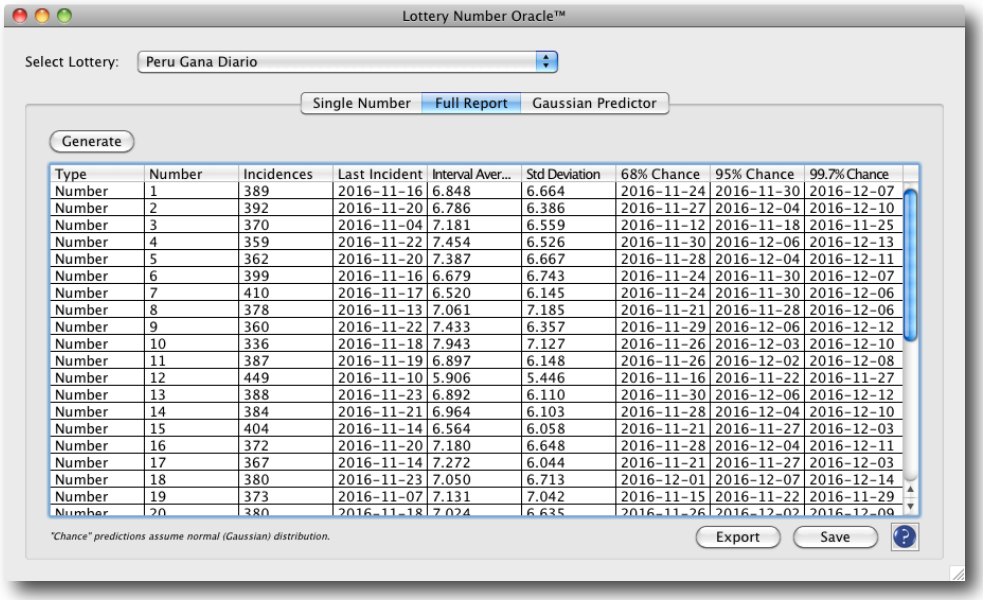


Figure 49.

Generate button
Clicking this button starts the report generation process.

Report list box
This list box shows the content of the entire report. You can resize columns by dragging the header borders. You can also sort any column by clicking on the column's header. Subsequent clicking alternates between ascending and descending order.

Here is a description of each column:

1. Type: the type of number (whether it is a bonus number or not; number position [pick type lottery]; etc.)
2. Number: the number being analyzed.
3. Incidences: how many times the number has been drawn.
4. Last Incident: the last time the number has been drawn.
5. Interval Average: the average interval, *in number of draws,* for that number.
6. Standard Deviation: the standard deviation for this number's interval.
7. 68% Chance: the last date of which this number has a 68% chance of being drawn*.
8. 95% Chance: the last date of which this number has a 95% chance of being drawn*.

9. 99.7% Chance: the last date of which this number has a 99.7% chance of being drawn*.

*assuming normal (Gaussian) distribution

Export button
Clicking this button exports the report as a Microsoft Excel file.

Save button
Clicking this button exports the report as a Microsoft Excel file.

Gaussian Predictor Mode

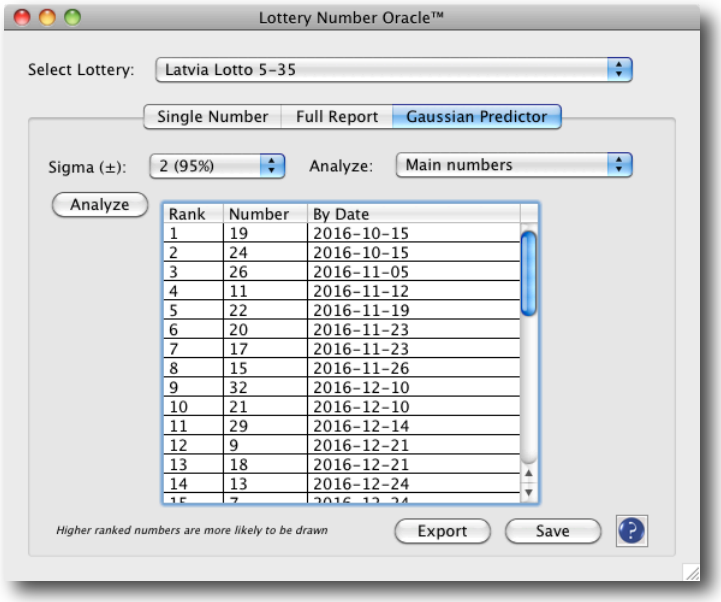


Figure 50.

The *Gaussian Predictor* mode show the predicted (by-date) values for the numbers in the lottery. Values shown are sorted so that the earliest dates appear at the top of the report. You can select the sigma value and, if applicable, which numbers to display. For example, if your lottery has a bonus number, drawn from a separate pool, you can select the main (non-bonus) numbers or the bonus numbers. If your lottery is a "pick-type" lottery, you can select which number position to analyze.

Sigma dropdown menu
Choose between 1 sigma or 3 sigma. 1 sigma show the predicted by-date values based on 1 standard deviation (68%); 2 sigma shows values based on 2 standard deviations (95%); and 3 sigma shows values based on 3 standard deviations (99.7%).

Analyze dropdown menu
If your lottery has a bonus number, drawn from a separate pool, you can select the main (non-bonus) numbers or the bonus numbers. If your lottery is a "pick-type" lottery, you can select which number position to analyze.

Analyze button
This starts the analysis process.

Report list box

This list box shows analysis results. You can resize columns by dragging the header borders.

Here is a description of each column:

1. Rank: results are ordered by date, so number 1 will have the earliest date.
2. Number: the number being analyzed.
3. By Date: the date shown represents the threshold date in year-month-day format. For example, "2016-11-10" means that there is a 68% chance of the number shown in the "Number" column will be drawn by that date (if you have chosen a value of "1" in the Sigma dropdown menu).

Export button

Clicking this button exports the report as a Microsoft Excel file.

Save button

Clicking this button exports the report as a Microsoft Excel file.

Pick Lottery Frequency Distribution

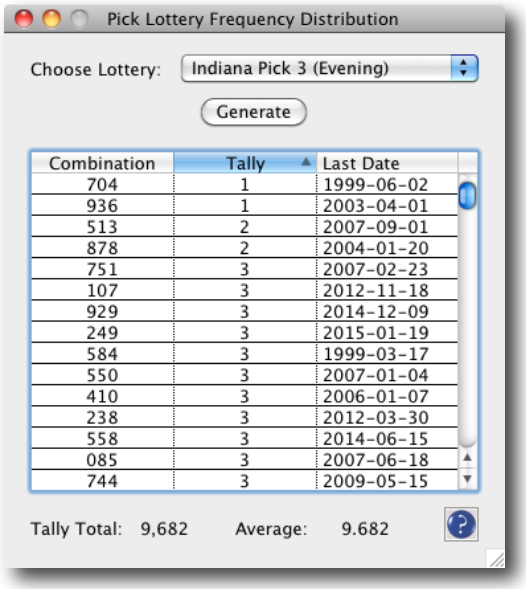


Figure 51.

Overview

This function shows the frequency distribution for pick-type lotteries. It will show how many times each combination has been drawn, as well as the last date of that particular drawing has occurred.

How to Invoke

Use the menu item "Tools > Pick Lottery Frequency Distribution...".

Window Controls

Choose Lottery dropdown menu

Select the pick-type lottery you want to analyze. Please note that only Pick 1, Pick 2, Pick 3, Pick 4 and Pick 5 lotteries that you have already set up can be selected.

Generate button

Click this button to start the analysis.

Combination/Tally/Last Date list box

This list box has three columns: the left column show the combination, the center column shows the number of times that combination has been drawn and the right column shows that last date that particular combination was drawn. Click on either header to sort the list, either in ascending or descending order.

Lotto Seer

Figure 52.

Overview

Lotto Seer allows you to view and print charts and graphs of your lottery's data.

How to Invoke

Use the menu item "Tools > Lotto Seer".

Basic Procedure

1. Select the lottery you want to see the data for.
2. Use the "Back" and "Next" buttons to switch from page to page.

Data Parameters Page

First, use the dropdown menu to choose the lottery you want to work with. After you do this, the Start Date and End Dates dropdowns will contain the first and last date in the drawing database, respectively. If you want to view a different date range, use the dropdowns to make your choices.

Clicking the "First" button will populate the Start Date with the first date of the database for that lottery, and clicking the "Last" button will populate the End Date dropdowns.

If the lottery you are analyzing has one or more bonus balls, you will see a dropdown menu appear at the bottom of this page. On this dropdown, choose whether you want to analyze the main (non-bonus) numbers or the bonus number(s). Please note that if you want to analyze only the bonus number(s), on the Frequency Distribution bar chart will be available.

Frequency Distribution Page

This page shows a chart with the frequency distribution for each number drawn. The first column show the number, the second column shows the frequency (i.e., number of times that number has been drawn), and the third column shows the percentage of time the number has been drawn.

The Copy button copies the Frequency Distribution chart to your System Clipboard; holding down the shift key while clicking the Copy button copies the Summary chart to the clipboard. The Save button saves the Frequency Distribution chart to a text file; holding down the shift key while clicking saves the Summary chart to a text file. In each case, the contents will have a tab character between each column's value.

Data Display Page

This chart shows the calculated values for each date within your range:

- Sum (total of all of the numbers drawn)
- High (highest number drawn)
- Low (lowest number drawn)
- Range (number spread of the numbers drawn)
- Odd (number of odd numbers drawn)
- Median
- Arithmetic Mean ("average")
- Harmonic Mean
- Truncated Mean
- Winsorized Mean
- Standard Deviation
- Variance (Standard Deviation)
- Population (Standard Deviation)
- Variance (Population Standard Deviation)

The "Export", "Copy" and "Save" buttons allow you to save the contents of this chart as a Microsoft Excel spreadsheet, to the system clipboard or to a text file, respectively.

Summary Display

This page contains a summary of the calculated values from the Data Display Page. The left-hand column contains the values, and the remaining columns contain the summaries for those values. For example, the second column ("Sum") and the fourth row ("Range") gives the sum of the Range column from the Data Display Page.

Graph Setup Display

This page lets you configure which graph you want to see, as well as colors and features.

Choose the graph you want to see from the dropdown menu at the top of the screen. The Frequency Distribution graph is a bar graph; the remainder are line graphs.

The Colors dropdown menus let you choose the different colors for the graph:

- The Border color is the color for the outer border of the graph as well as the outline of the bars.
- The Bars color is for the main color of the bars in the bar graph.
- The Datum color is for the color of the data lines in the line graphs.
- The Background color is the background color of the entire graph.
- The Text color is for the color of the text.
- The Other color is the color of the other data lines (average and standard deviation).

The Display Section will display optional lines on the chart (average and standard deviation). Please note that the Standard Deviation lines will appear only if they are far enough apart from the Average line to be visible.

The "Default" button will set the Colors and Display settings back to the factory defaults. The "Save" button will save the Color and Display settings.

Graph Display

This shows the graph of the data you chose. The "Copy" button will copy the graph image to the system clipboard, the "Save" button will let you save the graph, and the "Print" button will printout the graph. The Restart button will take you back to the beginning. Use the Refresh button if you resized the *Lotto Seer* window, or if the graph appears corrupted due to interface artifacts..

Tips:

- For each of the three charts, you can resize each column by clicking and dragging the borders of the top rows.
- For each of the three charts, so can sort each column by clicking the top row. Each time you click, the column will alternate between ascending sorting and descending sorting.
- You can resize the window if the graph appears to "packed" with data.

Utilities

Database Utilities

Backup Database

Overview

This function will backup the current database into a text file. You can restore your database by using the menu item "Utilities > Database Utilities > Restore Database". It is strongly recommend that you backup your database frequently. Note that you are responsible for moving the file created by this function to a location specified by your organization's Disaster Recovery Policy.

How to Invoke

Use the menu item "Utilities > Database Utilities > Backup Database".

Basic Procedure

A file selector will appear; choose the location where you want the backup file saved. The default location is in the folder "Backup Files", located in the "Lotto Sorcerer v9 Files folder", which, in turn, is located in your Documents folder.

Copy Database to Desktop

Overview

This places a copy of your Lotto Sorcerer database onto your Desktop.

Why is this function needed?

If you ever need to contact the Lotto Sorcerer support team because of an issue with Lotto Sorcerer, they will often request that you email them a copy of your database (so that they can duplicate the issue). Some users have difficulty locating this database. By using this function, the database is copied to your Desktop, so that you can easily find it.

☞ The original database is still in its location; only a copy is placed on your Desktop.

No personal information is in the database. Only your lottery settings, structure and lottery data (past drawings).

How to Invoke

Use the menu item "Utilities > Database Utilities > Copy Database to Desktop".

Execute SQL File

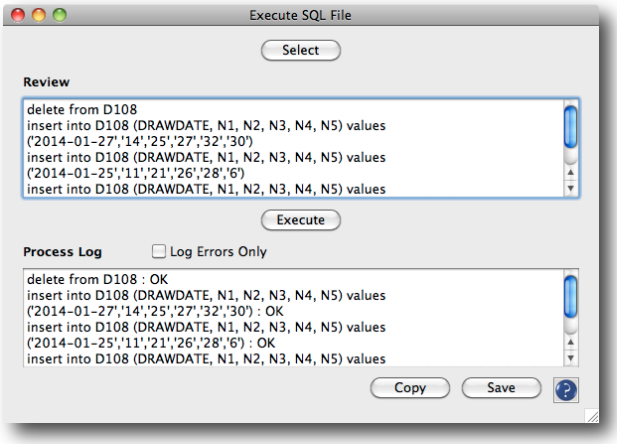


Figure 53.

Overview

This function executes a text file consisting of one or more SQL Statements..

☞ Any lines beginning with two slashes ("//") is treated as a comment line.

How to Invoke

Use the menu item "Utilities > Database Utilities > Execute SQL File...".

Basic Procedure

- Select the SQL file to execute by using the "Select" button.
- Important! Review the SQL file carefully in the Review box.
- Click the "Execute" button to execute the SQL file.

Window Controls

Select button

Clicking this button brings up the standard file selector. Choose the SQL file that you wish to execute. When you select the SQL file, the contents will be shown in the Review text box.

Review text box

This shows the contents of the SQL file that you selected. It is highly recommended that you review each and every line in the SQL file, since it can drastically effect the database.

Execute button

Clicking this button executes the SQL file. Clicking this button brings up the standard file selector. Choose the SQL file that you wish to execute.

Log Errors Only checkbox

If checked, only errors will show up in the "Process Log" text box. Otherwise, every line in original SQL file will appear, with the results.

Copy button

This copies the contents of the Process Log to the System Clipboard.

Save button

This saves the contents of the Process Log as a text file.

Force Database Rebuild

Overview

This is an emergency function in the event that Lotto Sorcerer's internal database has become corrupted (usually due to a hard disk hardware fault).

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Repair Tools > Force Database Rebuild".

Important!

Please note that this function does not rebuild data on the database; it re-creates a new, blank database.

Check Dates

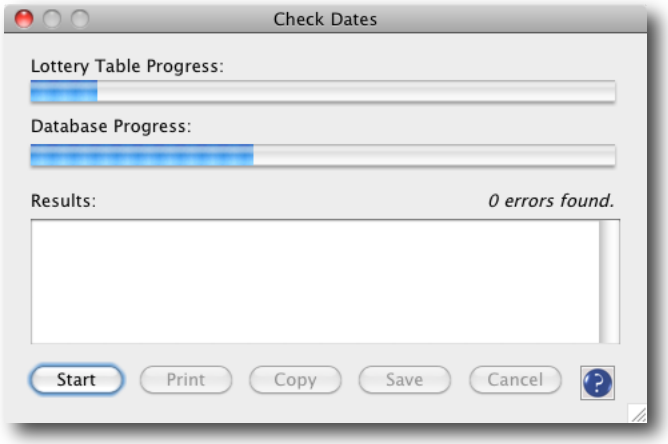


Figure 54.

Overview

This function will check the database for invalid dates. Invalid dates in the database can cause a host of strange behaviors. Note that it does not check the database for incorrect data.

How to Invoke

Use the menu item "Utilities > Database Utilities > Check Dates".

Window Controls

Start button

This starts the checking process.

Print button

This prints the report.

Copy button

This copies the report to the system Clipboard.

Save button

This saves the report as a text file.

Cancel button

This closes the window.

Database Browser

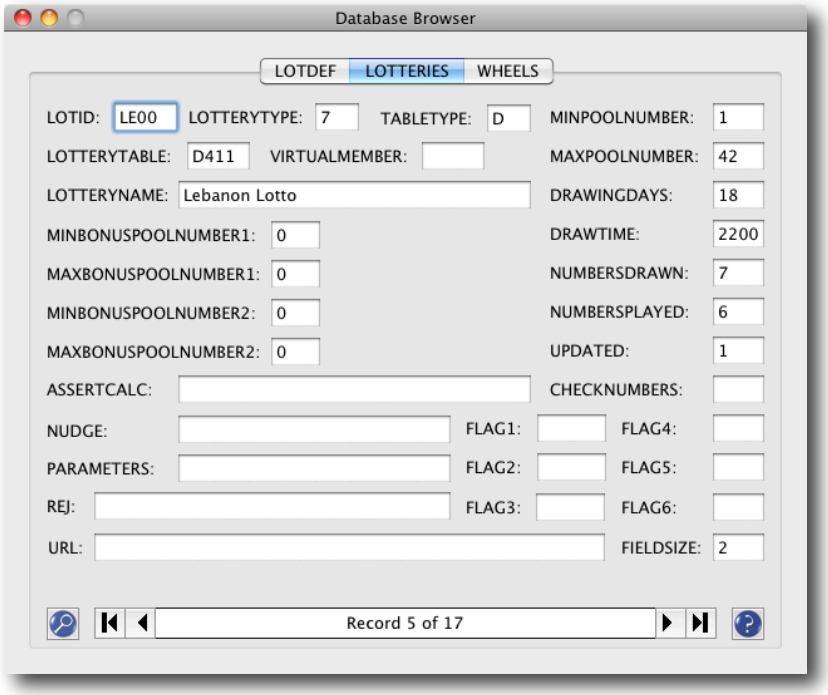


Figure 55.

Overview

This utility lets you browse (read-only) the three primary tables in Lotto Sorcerer:

1. LOTDEF - data containing all built-in lotteries in Lotto Sorcerer
2. LOTTERIES - data containing all of the lotteries that you have setup
3. WHEELS - data containing all of the wheels within Lotto Sorcerer

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Browser".

Window Controls

Data control

This lets you browse, one record at a time. Use the leftmost arrow to go to the first record; the next arrow to go to the previous record; the third arrow (second from the right) lets you go to the following record; and the rightmost arrow lets you go to the last record.

Search Tools

Clicking the icon at the bottom left of the Data Browser window opens up a search window. In this window, you can narrow down your view by selecting search terms for a particular field, sort by a particular field, and sort directions.

Import v6 Database

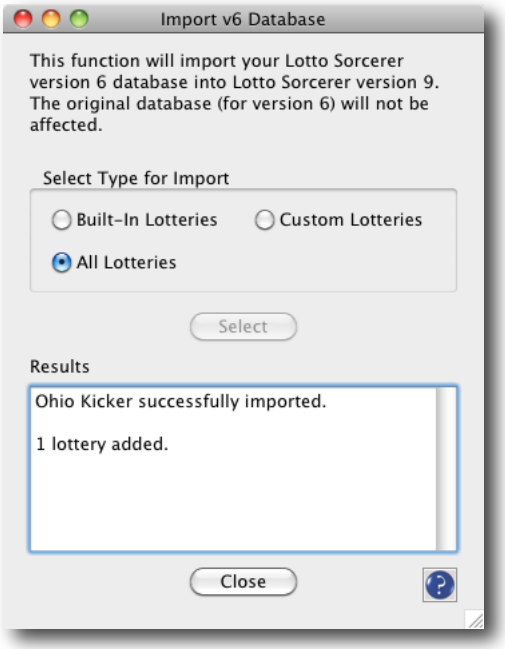


Figure 56.

Overview

This function lets you import the custom lotteries from your Lotto Sorcerer v6 installation directly into Lotto Sorcerer v9.

Please note that this function will work only on a new, empty database on Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Database Utilities > Import Legacy Databases > Import v6 Database".

Window Controls

Select Type for Import radio buttons

Use this to select the types of lotteries to import from the v6 database: built-in lotteries, custom lotteries, or all lotteries.

Select button

Clicking this button opens up a standard file selector. If Lotto Sorcerer v6 is installed on the same computer and user folder as Lotto Sorcerer v9, the file selector will "land" on the v6 database file by default. Selecting it will immediately launch the importation process.

Import v7 Database

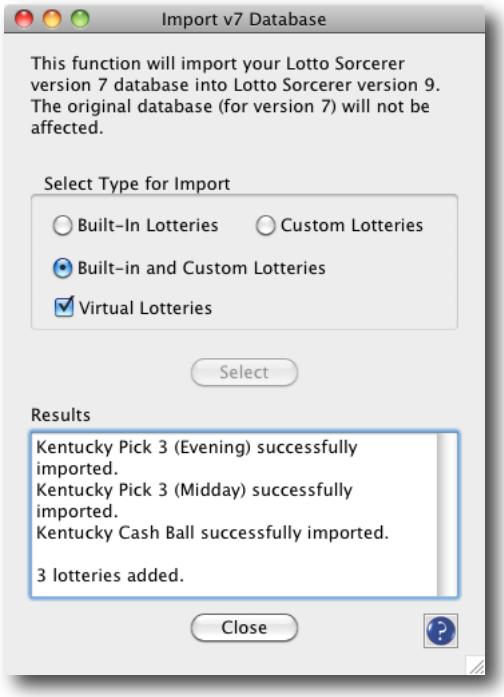


Figure 57.

Overview

This function lets you import the custom lotteries from your Lotto Sorcerer v7 installation directly into Lotto Sorcerer v9.

Please note that this function will work only on a new, empty database on Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Database Utilities > Import Legacy Databases > Import v7 Database".

Window Controls

Select Type for Import radio buttons

Use this to select the types of lotteries to import from the v7 database: built-in lotteries, custom lotteries, or both built-in and custom lotteries.

Virtual Lotteries checkbox

If you have virtual lotteries setup in your v7 database, and you want to import them into v9, check this checkbox.

Select button

Clicking this button opens up a standard file selector. If Lotto Sorcerer v7 is installed on the same computer and user folder as Lotto Sorcerer v9, the file selector will "land" on the v7 database file by default. Selecting it will immediately launch the importation process.

Import v8 Database

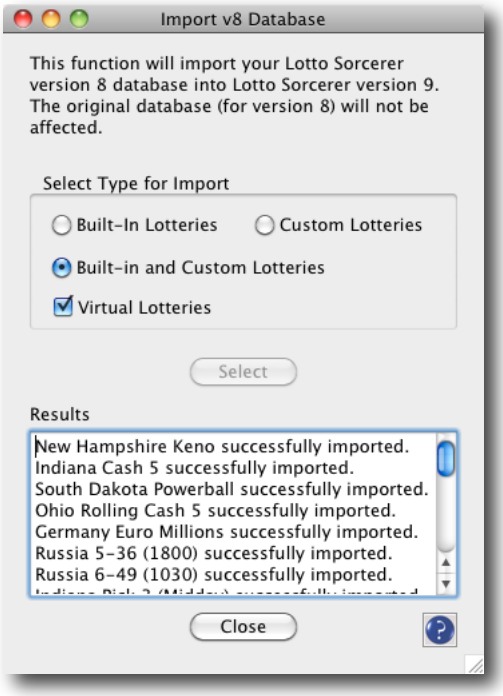


Figure 58.

Overview

This function lets you import the custom lotteries from your Lotto Sorcerer v8 installation directly into Lotto Sorcerer v9.

Please note that this function will work only on a new, empty database on Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Database Utilities > Import Legacy Databases > Import v8 Database".

Window Controls

Select Type for Import radio buttons

Use this to select the types of lotteries to import from the v8 database: built-in lotteries, custom lotteries, or both built-in and custom lotteries.

Virtual Lotteries checkbox

If you have virtual lotteries setup in your v8 database, and you want to import them into v9, check this checkbox.

Select button

Clicking this button opens up a standard file selector. If Lotto Sorcerer v8 is installed on the same computer and user folder as Lotto Sorcerer v9, the file selector will "land" on the v8 database file by default. Selecting it will immediately launch the importation process.

Lottery Extractor

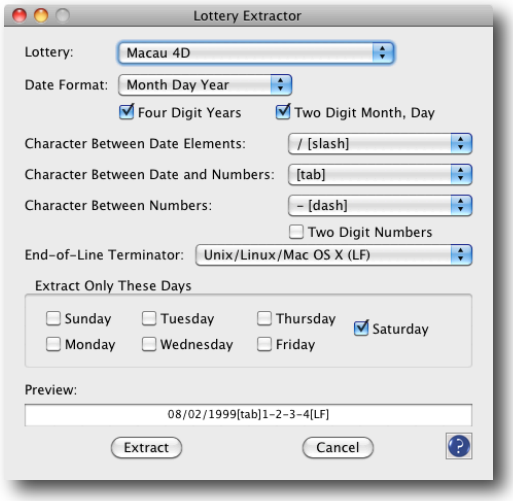


Figure 59.

Overview

This function lets you extract data (by specific day or days) from Lotto Sorcerer into a delimited text file. The file you export can be easily imported back into the program. For example, if you have a lottery that holds drawings every day, but you want to extract the drawings only from Saturday, use this function.

Please note that unregistered versions of Lotto Sorcerer are limited to 50 source records for exporting.

How to Invoke

Use the menu item "Utilities >Database Utilities > Lottery Extractor".

Basic Procedure

1. Select the lottery you want to export
2. Choose the export parameters
3. Choose the day(s) you want
4. Click the Extract button

Window Controls

Select Lottery dropdown

Choose the lottery you want to export.

Date Format dropdown

Choose the exact date format you want to use.

Four Digit Years checkbox

If checked, the data will be exported as four-digit years (for example, "2005"); if unchecked only the last two digits are exported (for example, "2005" will be exported as "05"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Two Digit Years Month, Day

If checked, the data will be exported as two-digit months and days (for example, "01/09/2013"); if unchecked, these values will have only one digit (for example, "1/9/2013"). Note that if you choose "[nothing]" as the character between Date Elements, this checkbox will be checked automatically.

Character Between Date Elements dropdown
Choose the character that separates the year, month and day parts in the date field.

Character Between Date and Numbers dropdown
Choose the character that separates the date field and the number fields.

Character Between Numbers dropdown
Choose the character that separates the different number fields.

Two Digit Numbers
If checked, single digit numbers will be zero padded. For example, "1-2-5-16-19" will be exported as "01-02-05-16-19". If you chose "[nothing] as the character between numbers, this will be checked automatically.

End-of-Line Terminator dropdown
Different operating systems use different characters to mark the end of line. Choose the appropriate one.

Extract button
Clicking this button starts the extracting process.

Extract Only These Days checkboxes
Check the day (or days) you wish to limit the exported file to.

Cancel button
Use this to close the current window and return to the Main window.

Optimize Database

Overview

This function will optimize the database, putting all records in descending chronological order. This has the effect of making Lotto Sorcerer faster during day-to-day use.

How to Invoke

Use the menu item "Utilities > Database Utilities > Optimize Database".

Rebuild Lottery Definitions

Overview

This function will reset the lottery definitions ("LOTDEF") table to the factory defaults.

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Repair Tools > Rebuild Lottery Definitions...".

Reinforce Table Integrity

Overview

This function ensures that certain fields in the LOTTERIES table are mapped properly. The symptom of a table having problems of this nature would be if certain lotteries are not appearing in some of the "Select Lottery" dropdown menus.

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Repair Tools > Reinforce Table Integrity".

Basic Procedure

No dialog box will appear. The procedure will occur whenever you select this menu item. It never hurts to run this function.

Remove Orphans

Overview

An "orphan" is a record in the LOTTERIES table in the database that points to a non-existing table. An orphan record can be created due to:

- A computer crash or power outage at the time of the creation of the lottery
- A miswritten SQL command issued in the SQL Interface
- A hard drive glitch

A symptom of an orphan would be a error message being displayed when selecting a lottery in the Main Window. This function will search for an automatically remove all orphans.

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Repair Tools > Remove Orphans".

SQL Command Line Interface

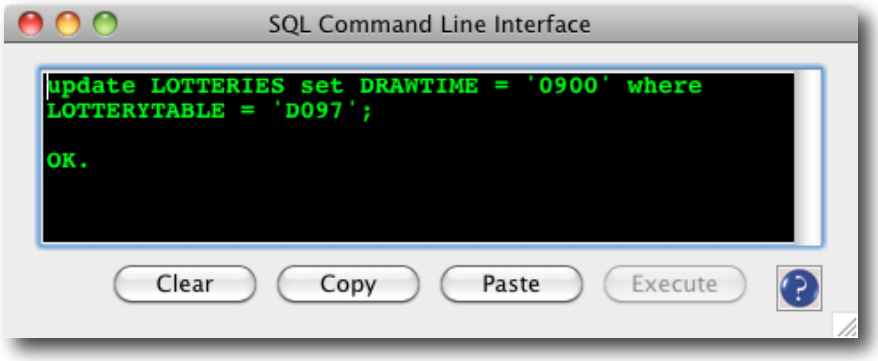


Figure 60.

Overview

Important! This function is intended only for power users who know SQL (Structured Query Language). Because you have full control over the database using this interface, it is easy to inadvertently corrupt the database. You cannot "undo" any action.

Lotto Sorcerer's SQL database is a SQLite database; see the SQLite web site (at www.sqlite.org) for full information and syntax. This database is an ACID-compliant embedded relational database management system adhering to most of SQL-92 standards.

Note

Only one command or query can be entered at one time. After the command or query is executed, the Execute button will become ghosted. Clicking the Clear button will clear the command box and re-enable the Execute button.

How to Invoke

Use the menu item "Utilities > Database Utilities > SQL Command Line Interface".

Basic Procedure

1. Enter the SQL command or query in the command box
2. Click the "Execute" button

Window Controls

Clear button

This clears the command box.

Copy button

This copies the contents of the command box into the System Clipboard.

Paste button

This pastes the System Clipboard into the command box.

Execute button

This will execute the SQL command or SQL query that is shown in the command box.

SQL Interface

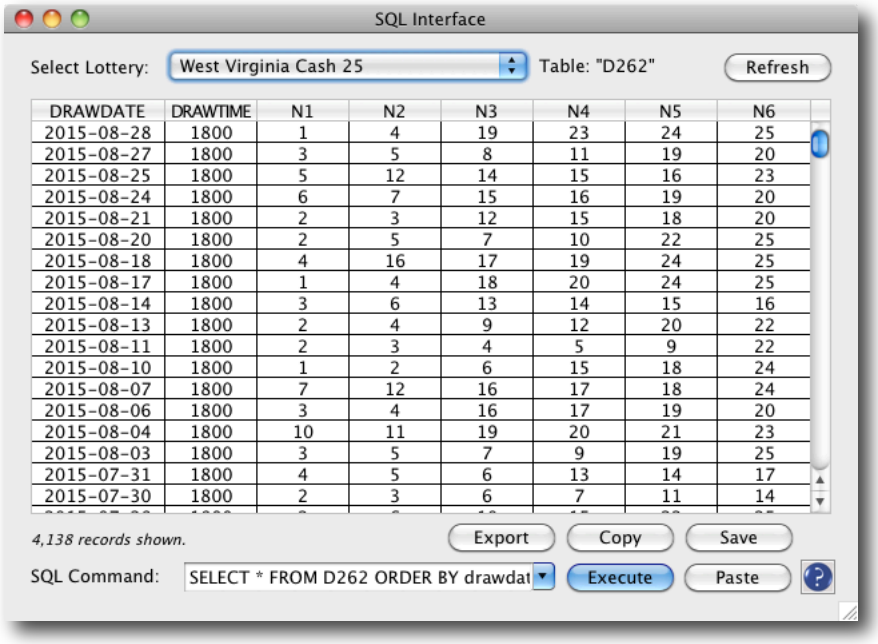


Figure 61.

Overview

Important! This function is intended only for power users who know SQL (Structured Query Language). Because you have full control over the database using this interface, it is easy to inadvertently corrupt the database. You cannot "undo" any action.

Lotto Sorcerer's SQL database is a SQLite database; see the SQLite web site (at www.sqlite.org) for full information and syntax. This database is an ACID-compliant embedded relational database management system adhering to most of SQL-92 standards.

How to Invoke

Use the menu item "Utilities > Database Utilities > SQL Interface".

Basic Procedure

3. Select the lottery you want to work with from the top dropdown menu
4. Enter valid SQL statements in the text box at the bottom
5. Click the "Execute" button

Window Controls

Select Lottery dropdown

Use this dropdown to select the lottery that you want to work with.

Refresh button

If you executed a SQL command that changed the data, clicking this button will refresh the view of the table.

SQL Command combo box

Enter the SQL command or statement you want to execute in this box.

Export button
This will export the results as a Microsoft Excel spreadsheet.

Copy button
This will copy the results to the System Clipboard.

Save button
This will save the results to a text file.

Execute button
This will execute the SQL command that you typed in the SQL Command combo box.

Paste button
This will copy the text in your System Clipboard into the SQL Command combo box.

Notes
When you select the lottery in the top dropdown, the name of the table will be shown at the top-right of the window.

Successfully executed SQL statements will be "remembered" and displayed in the SQL Command text box as long as you have Lotto Sorcerer running. Use the dropdown to recall these statements.

Vacuum Database

Overview

When an lottery is deleted from the database, it leaves behind empty space. This makes the database file larger than it needs to be, but can speed up inserts. In time, inserts and deletes can leave the database file structure fragmented, which slows down disk access to the database contents. Vacuuming the database cleans the main database by copying its contents to a temporary database file and reloading the original database file from the copy. This eliminates free pages, aligns table data to be contiguous, and otherwise cleans up the database file structure.

How to Invoke

Use the menu item "Utilities > Database Utilities > Vacuum Database".

Restore Database

Overview

This function will restore the database from the text file created from the "Backup Database" function.

How to Invoke

Use the menu item "Utilities > Database Utilities > Restore Database".

Basic Procedure

A file selector will appear; choose the file that was created by the "Backup Database" function (see page 113).

Zap Gremlins

Overview

This function remove "gremlins" (unwanted characters) from the DRAWDATE field in all lottery tables (except Virtual Lotteries).

How to Invoke

Use the menu item "Utilities > Database Utilities > Database Repair Tools > Zap Gremlins".

Basic Procedure

No dialog box will appear. The procedure will occur whenever you select this menu item. It never hurts to run this function.

Note

Gremlins can appear during an errant import process. A symptom of gremlins would be odd dates showing in the data.

Random Utilities

Data Padder

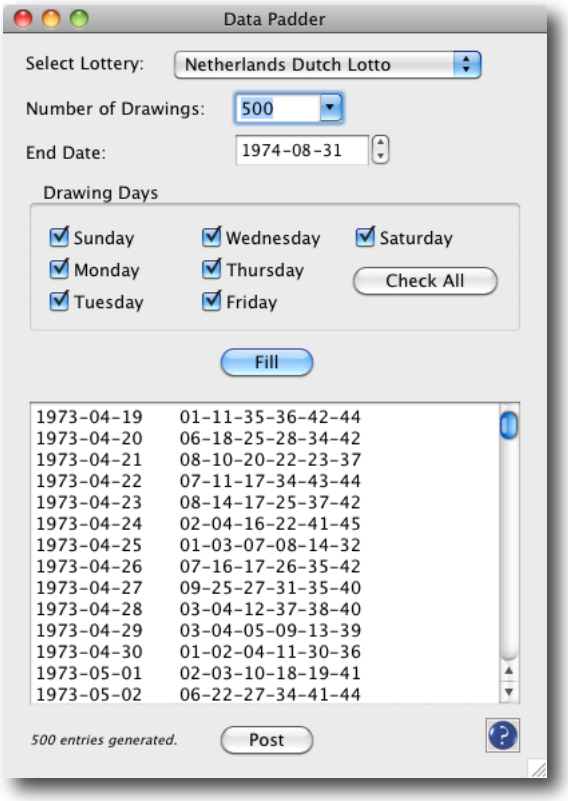


Figure 62.

Overview

This purpose of this function fills a lottery will statistically neutral random drawings. This can be useful if you have a lottery with too few drawings to even run at neural level depth of 1. You can add enough of these "drawings" to reach the threshold required to run at neural level 1. Accuracy will be compromised, of course, because these are not actually drawings; fortunately, Lotto Sorcerer gives more statistical weight to more recent drawings, which, in this case, will be actual drawings.

Once the real drawings are numerous enough to run Lotto Sorcerer at the neural depth of 1, the spurious drawings should be deleted using the Prune Lottery function (see page 59).

How to Invoke

Use the menu item "Utilities > Random Utilities > Data Padder".

Basic Procedure

1. Select the lottery you want to work with
2. Choose the number of "drawings" you want added
3. Choose the End Date for the data you want to insert into the lottery
4. Choose the days of the week for the drawing data
5. Click the Fill button; preview data will be shown
6. Click the Post button

Window Controls

Select Lottery dropdown

Choose the desired lottery.

Number of Random Numbers combo box

Choose or enter the number of "drawings" to add.

End Date date selector

Choose the ending date. It is important that the date selected be before the first actual drawing in the database for this lottery. For your convenience, when a lottery is selected, this selector is set automatically with the proper value.

Drawing Days checkboxes

Choose the drawing days for this lottery. This will automatically be selected by the lottery's settings, but you can override them.

Fill button

This starts the generation process. The results will appear in the box immediately below this button. A count of the number of "drawings" generated will be shown to the left of the "Post" button.

Post button

This posts the data directly into the database. Please note that existing data will not be overwritten.

Generate Random Numbers

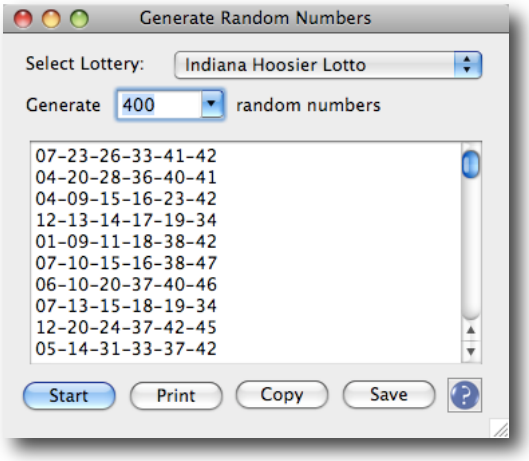


Figure 63.

Overview

This function lets you create random numbers for a lottery. This function was created for users who did not trust that the lottery's "Quick Pick" or "Easy Pick" results were truly random, or who prefer to preview numbers, even random numbers, before playing them.

How to Invoke

Use the menu item "Utilities > Random Utilities > Generate Random Numbers".

Basic Procedure

7. Select the lottery you want to generate random numbers for
8. Select or enter the number of random numbers you want
9. Click the Start button

Window Controls

Select Lottery dropdown

Choose the lottery you want to generate random numbers for.

Number of Random Numbers combo box

Choose the number of random numbers you want generated.

Start button

This button starts the generation process.

Print button

This prints the results.

Copy button

Clicking the "Copy" button saves the results to the system clipboard.

Save button

This saves the results to a text file.

Generate Seeded Random Numbers

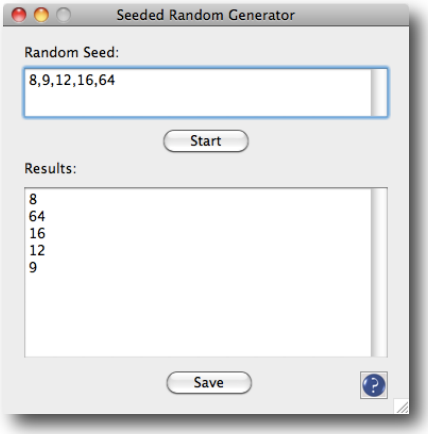


Figure 64.

Overview

This function lets you create random numbers based on your own seed. This function was suggested by users.

How to Invoke

Use the menu item "Utilities > Random Utilities > Generate Seeded Random Numbers".

Basic Procedure

1. Enter the seeds for the random numbers, separated by commas. Spaces will be ignored.
2. Click the Start button

Window Controls

Random Seed text box

Enter the seeds for the random numbers, separated by commas. Spaces will be ignored.

Start button

This button starts the generation process.

Results text box

This contains the results of the random number generation process.

Save button

This saves the results to a text file.

Scrambler

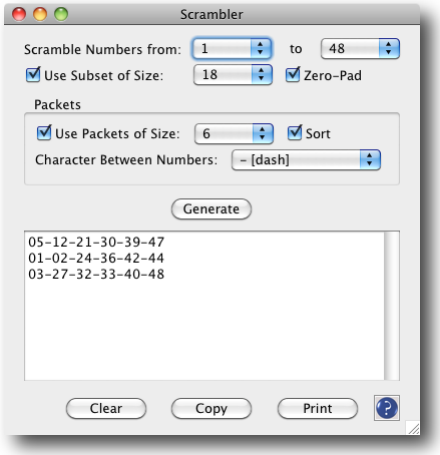


Figure 65.

Overview

Several users requested this function. It simply generates random numbers in the range you wish.

How to Invoke

Use the menu item "Utilities > Random Utilities > Scrambler...".

Basic Procedure

1. Select the lower and upper limits for the number range you want
2. Choose options, if desired
3. Click the Generate button

Window Controls

Lower Number dropdown

Choose the lottery you want to generate random numbers for (0-1).

Upper Number dropdown

Choose the number of random numbers you want generated (1-99).

Use Subset of Size checkbox and dropdown

If you do not want the entire set of numbers (from the Lower Number dropdown to the Upper Number dropdown), check this box and choose the subset size. For example, if your lottery draws six numbers from 1 to 48, but you want three games, you would choose 18 (6 x 3 = 18).

Zero-Pad checkbox

Any number generated that is nine and below will have a zero in front (for example, "3" becomes "03").

Use Packets of Size checkbox and dropdown

If checked, this will group the random numbers in packets the size you selected in the dropdown.

Sort checkbox

f checked, this sort the packets in numerical order.

Generate button

This button starts the generation process.

Clear button

This clears the results box.

Copy button

Clicking the "Copy" button saves the results to the system clipboard.

Print button

This prints the results.

Calculators

Boolean Calculator

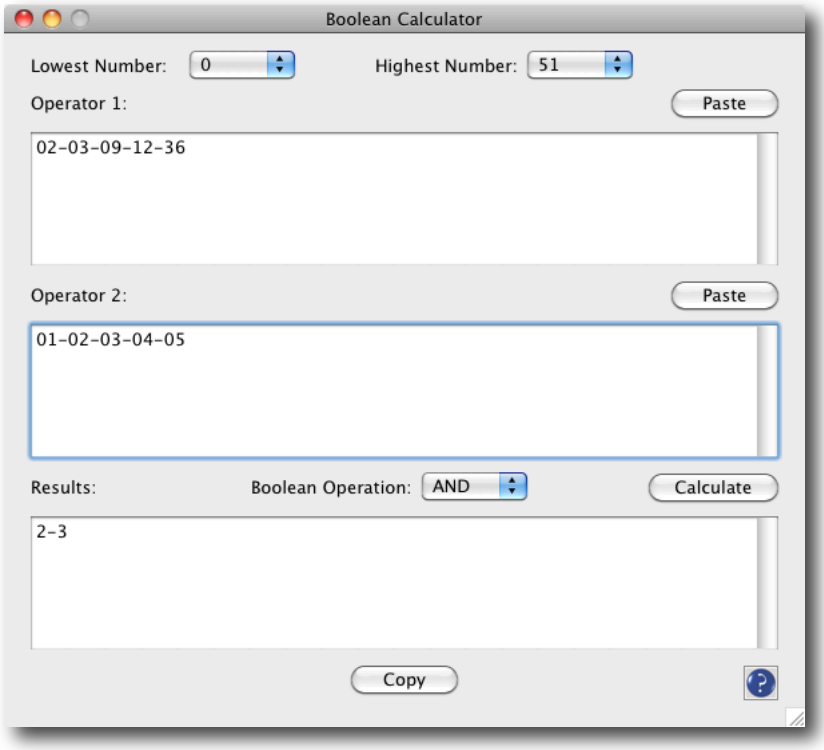


Figure 66.

Overview

Although boolean operations are generally associated with binary operations, this calculator is targeted towards decimal operations.

How to Invoke

Use the menu item "Utilities > Calculators > Boolean Calculator...".

Basic Procedure

1. Select the lowest and highest numbers of the operator in the "Lowest Number" and "Highest Numbers" drop down menus
2. Enter the first operator in the "Operator 1" text box. The delimiter for this operator can be any non-numeric character.
3. Enter the second operator in the "Operator 2" text box. The delimiter for this operator can be any non-numeric character
4. Choose the boolean operation in the "Operation" drop down menu
5. Click the "Calculate" button

Window Controls

Lowest Number drop down menu

Select the lowest number of the operation. Note that this effects only NOR and NAND operations.

Highest Number drop down menu

Select the highest number of the operation. Again, this effects only NOR and NAND operations.

Boolean Operation drop down menu

Choose the desired Boolean operation:

- AND
- NAND (not AND)
- NOR (not OR)
- OR
- XOR (eXclusive OR)

Calculate button

Click to display the calculations.

Bitwise Day Calculator

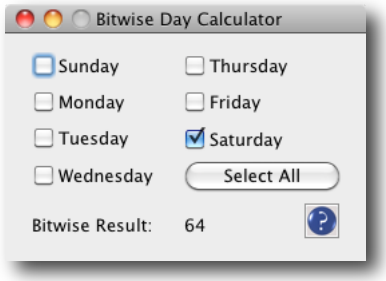


Figure 67.

Overview

This function calculates the bitwise value for days. This value is stored in Lotto Sorcerer's internal database.

How to Invoke

Use the menu item "Utilities > Calculators > Bitwise Day Calculator...".

Basic Procedure

1. Select the days by checking their checkboxes.
2. The bitwise day value is shown at the bottom of the window.

Window Controls

Day of Week checkboxes

Select the appropriate days you want by checking the days' checkboxes.

Select All / Select None button

This button alternates between "Select All" and "Select None" to check all of the boxes or uncheck all of the boxes, respectively.

Lottery Odds Calculator

Figure 68.

Overview

This function calculates the standard odds of standard lotto and pick-type lottery games. This assumes that the lottery is choosing truly random numbers.

How to Invoke

Use the menu item "Utilities > Calculators > Lotto Odds Calculator...".

Basic Procedure

1. Select the parameters of the lottery with the dropdown menus
2. Click the "Calculate" button

Window Controls

Pool Size dropdown

Select the size of the lottery. For example, for a lottery that draws from 1 to 49, choose "49".

Numbers Drawn dropdown

Select how many numbers are drawn. For example, if the lottery draws six numbers from 1 to 49, choose "6".

Numbers Match dropdown

Select how many numbers that match for the odds you want to see. For example, if the lottery draws six numbers, but you want to see what the odds are of winning four (of the six) numbers, choose "4". You can even see the odds that you will not match any drawn numbers at all, by choosing zero ("0").

Calculate button

Click to display the calculations.

Permutations Calculator

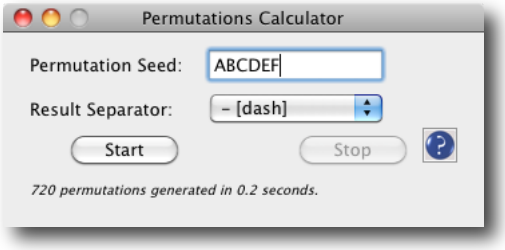


Figure 69.

Overview

This function will generate the permutations of numbers or letters, up to ten characters long.

How to Invoke

Use the menu item "Utilities > Calculators > Permutations Calculator...".

Basic Procedure

1. Enter the permutation seed
2. Choose the separator for the results
3. Click the Start button

Window Controls

Enter Permutation Seed text box

Enter the permutation seed, either numbers or letters. Do not separate the characters; all input will be used. You are limited to ten characters.

Result Separator dropdown

Choose the separator character you want used in the results.

Start button

The Permutation Calculator saves the results to a text file. When you click the Start button, you will be asked the name and location of the file. This calculator will default to a filename as the permutation seed (with a ".txt" extension) and a default location of the "Permutations" folder in your "Lotto Sorcerer v9 Files" folder (which, in turn, is in your "Documents" folder).

Stop button

You can stop the generation process at any time by clicking this button.

Date Calculator

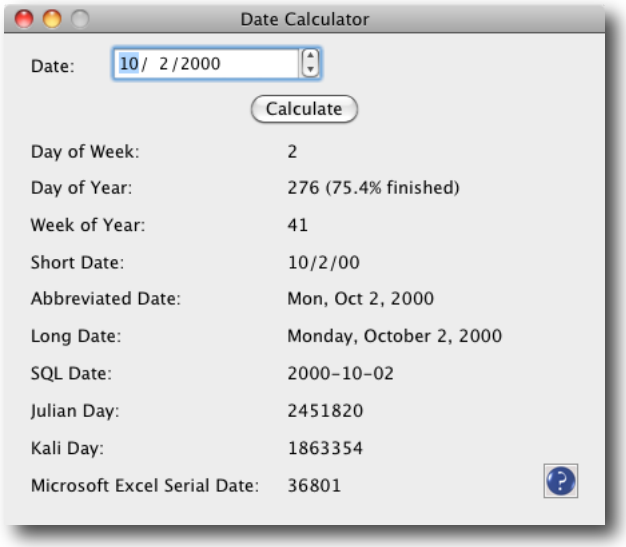


Figure 70.

Overview

This function calculates different versions of a given date.

How to Invoke

Use the menu item "Utilities > Calculators > Date Calculator...".

Basic Procedure

1. Choose the date you wish converted
2. Click the Calculate button

Window Controls

Enter Date control

Enter the date you want calculated in the Date control.

Calculate button

This calculates and converts the different date formats:

1. Day of Week: reckoned as Sunday = 1, Saturday = 7
2. Day of Year: reckoned with January 1st being the first day of the year
3. Week of Year: reckoned with the week on which January 1st falls as being the first week of the year
4. Short Date: this is based on your locale and formatting (using both Windows and Macintosh's "Short" setting)
5. Abbreviated Date: this is based on your locale and formatting (using Mac OS X's "Medium" date setting and an abbreviated version of Windows' "Long" setting).
6. Long Date: this is based on your locale and formatting (using Macintosh's "Full" date setting and an abbreviated version of Windows' "Long" setting).
7. SQL date: this displays in YYYY-MM-DD format.
8. Julian Day: this displays the number of days since the beginning of the Julian Period (November 24, 4714 BC in the Gregorian Calendar).
9. Kali Day Number: this displays the number of days since the beginning of Kali Yuga (January 14, 3102 BC in the Gregorian Calendar).

10. Excel Serial Date: this displays the number used by Microsoft's Excel spreadsheet for storing dates.

Notes

On Windows, the Regional and Language Options panel determines how dates are formatted. On Macintosh, the date formats are specified in the Formats panel in the International control panel.

Combinations Calculator

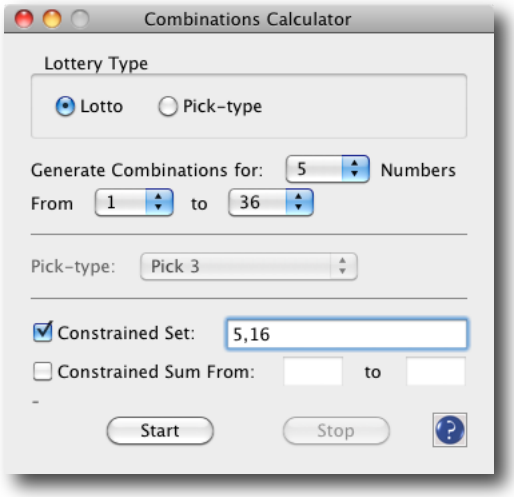


Figure 71.

Overview

This function will generate the combinations of numbers for a standard lotto-type lottery as well as pick-type lotteries.

How to Invoke

Use the menu item "Utilities > Calculators > Combinations Calculator...".

Basic Procedure

1. Select the type of lottery (lotto or pick-type)
2. If a standard lottery, select the number pool and boundaries of the lottery pool; if a pick-type lottery, choose the numbers drawn
3. If you want to constrain the combinations, check the "Constrain" checkbox and enter the constrainers in the text box, separated by commas
4. Click the Start button

Window Controls

Lottery type radio buttons

Choose the type of lottery here. "Pick-type" lotteries are lotteries which draw multiple numbers between 0 and 9.

Generate Combinations dropdown

Select the numbers drawn, from 2 to 10. For example, for a lottery that draws six numbers from one to 53, choose "6". If you chose "Pick-type" for the lottery type, then this control, and the next two controls, will be ghosted.

"From" number dropdown

Select the lowest number from the lottery pool. For example, for a lottery that draws six numbers from one to 53, choose "1".

"To" number dropdown

Select the highest number from the lottery pool. For example, for a lottery that draws six numbers from one to 53, choose "53".

Pick-type dropdown menu

Choose the pick-type lottery, from "Pick 2" to "Pick 8". If you chose "Lotto" for the lottery type, then this control will be ghosted.

Constrain checkbox and text box

If you want to limit the results to a number or set of number, check this box and enter the constrainers in the text box, separated by commas. In the example shown in Figure 61, only combinations that contain the numbers 5 and 16 will be generated.

Constrain Sum checkbox and text boxes

If you want to limit the results to a range of sums, check this box and enter the minimum and maximum values will be generated. For example, if you enter "81" in the "From" box and "108" in the "To" box, only numbers that add up to between 81 and 108 will be generated.

Start button

The Combinations Calculator saves the results to a text file. When you click the Start button, you will be asked the name and location of the file. This calculator will default to a filename as the combination parameters (with a ".txt" extension) and a default location of the "Combinations" folder in your "Lotto Sorcerer v9 Files" folder (which, in turn, is in your "Documents" folder).

Stop button

You can stop the generation process at any time by clicking this button.

Other Utilities

Backup "Lotto Sorcerer v9 Files" Folder

Figure 72.

Overview

This function backs up your entire "Lotto Sorcerer v9 Files" folder to your Desktop. This essential folder is located in your Documents folder. By default, Lotto Sorcerer stores several important files in this location, including your lottery drawing database, modified and added wheels, suggestions, logs and so on.

If you ever need to restore this folder, simply quit Lotto Sorcerer and copy this folder back to your Documents folder.

How to Invoke

Use the menu item "Utilities > Backup 'Lotto Sorcerer v9 Files Folder'".

Basic Procedure

1. Click the Start button.

Window Controls

Start Button

Click this button to start the backup process. When finished, the backup folder will be located on your Desktop.

Cancel Button

Click this to cancel and close the window.

Check Numbers

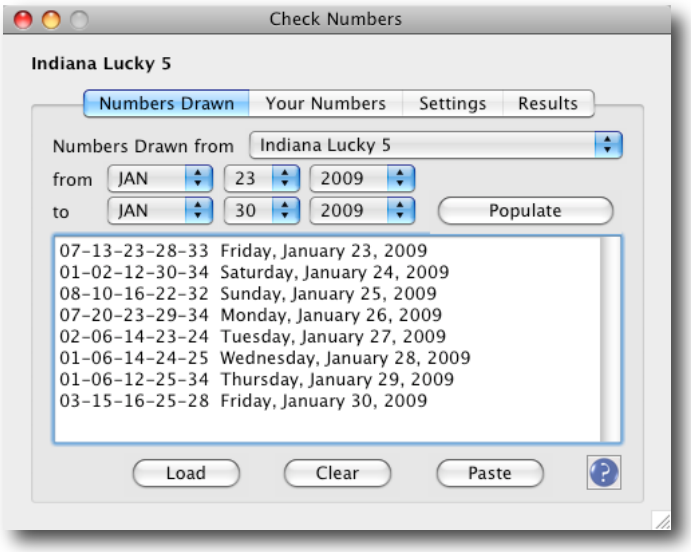


Figure 73.

Overview

This is used to check to see if your numbers have won. Note that the numbers are checked with the built-in database. It is important to verify winning numbers with your lottery office. Also, please note that some lotteries have subtle rules regarding what is truly a "win"... Lotto Sorcerer uses generalized rules, and any wins listed should be viewed as potential winners only (until you have verified the numbers with your lottery's officials).

How to Invoke

Use the menu item "Utilities > Check Numbers".

Basic Procedure

1. Enter or select the numbers that were drawn in the "Numbers Drawn" tab.
2. Enter or select the numbers you played in the "Your Numbers" tab.
3. Choose appropriate settings in the "Settings" tab.
4. Use the "Results" tab to check for any wins.

Window Controls

Numbers Drawn Tab

In this box, enter the numbers drawn that you want to check your numbers against. You can populate this box from the database by selecting the beginning and ending numbers, then clicking the "Populate" button. You can also type in the numbers, paste numbers from the system clipboard (by clicking the "Paste" button), or open a text file containing the numbers by clicking the "Load" button.

When entering numbers, there are some rules which you must adhere by:

- Use any non-numeric character to separate the numbers in a drawing.
- Anything after the final number is ignored.
- If the lottery is a bonus ball type lottery, the bonus ball(s) must be last in the series.

Your Numbers Tab

In this box, enter the numbers drawn that you that you played. You can type in the numbers, paste numbers from the system clipboard (by clicking the "Paste" button), or open a text file containing the numbers by clicking the "Load" button.

The separator between the numbers can be any non-numeric character.

Settings Tab

In this tab, you can choose what is defined as a "winning" number.

Important! Note that, for bonus-ball type lotteries, the matching of any bonus ball is always considered to be a winner. For lotteries with "extra" or supplemental numbers, the supplemental number is always counted towards matching numbers. Because these rules vary among different lotteries, these rules may not apply to your lottery, and may give a "false positive" result.

Pick 3 and Pick 4 numbers have other winning options, which you can choose from. Note that not all lotteries recognize all of these as winning matches:

- Straight: all numbers match, in order. Example: "4719" drawn and "4719" played would win "straight".
- Box: any combination wins. Example: "4719" drawn and "1974" played wins "boxed".
- Front Pair: the first two numbers drawn match the first two numbers played. Example: "4719" drawn and "4732" played would win "front pair".
- Back Pair: the last two numbers drawn match the last two numbers played. Example: "4719" drawn and "5219" played would win "back pair".
- Split Pair: the first and last numbers drawn match the first and last numbers played. Example: "4719" drawn and "4569" would win "split pair."
- Exact 1: any one number matches in the exact position. Example: "4719" drawn and "5213" matches "Exact 1".
- Exact 2: any two numbers match in the exact positions. Example: "4719" drawn and "4213" matches "Exact 2".
- Exact 3: any three numbers match in the exact positions. Example: "4719" drawn and "4219" matches "Exact 3". This is for Pick 4 type lotteries only.
- First 3, any order: the first three numbers in the drawn number set match the first three numbers in the played number set, in any order. This is for Pick 4 type lotteries only. Example: if "6108" is drawn and "1067" is played, this is a winner.
- Last 3, any order: the last three numbers in the drawn number set match the last three numbers in the played number set, in any order. This is for Pick 4 type lotteries only. Example: if "6108" is drawn and "7801" is played, this is a winner.

Results Tab

Click the "Check" button for an analysis. When complete, you can save the results to a text file (by clicking the "Save" button), print the results, or copy the results to the system clipboard.

Change Time

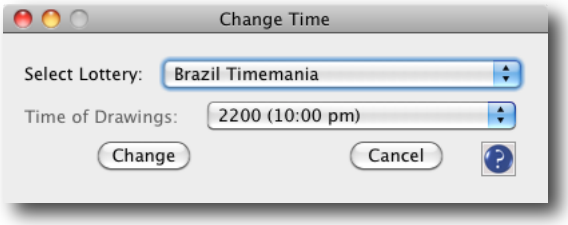


Figure 74.

Overview

This function will easily change the drawing time setting of a lottery as well as the drawing time for existing draws for that lottery.

The drawing time is only important for virtual lotteries, because all members of a virtual lottery must have a different drawing time from each other. The exact drawing time is not important; only the relative times: i.e., the earlier drawing must have an earlier time than the later drawing. The actual drawing times are not important.

How to Invoke

Use the menu item "Utilities > Change Time".

Basic Procedure

1. Select the lottery
2. Select the new time

Window Controls

Select Lottery dropdown menu

Select the lottery you want to change the drawing time settings.

Time of Drawing dropdown menu

Select the time of the drawing.

Change button

This changes the drawing times for the lottery.

Cancel button

This closes the window.

Clipboard Utility

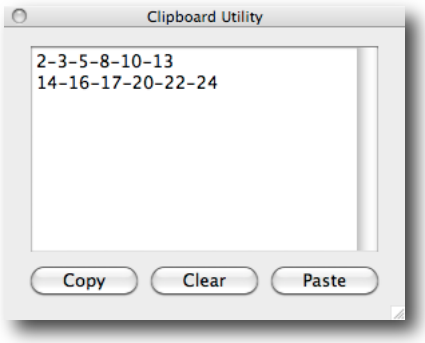


Figure 75.

Overview

This function gives you access to the System Clipboard (for text clippings only).

How to Invoke

Use the menu item "Utilities > Clipboard Utility...".

Window Controls

Copy button

This copies the contents of the text box to the System Clipboard.

Clear button

This clears the text box. It has no effect on the System Clipboard.

Paste button

This pastes the text clipping of the System Clipboard to the text box.

Check IO Permissions

Figure 76.

Overview

This function checks whether Lotto Sorcerer has the IO (Input/Output) permissions set properly. Symptoms of the permissions set wrong would be unexpected behavior, such as Lotto Sorcerer not being able to "remember" its preferences, or lottery data that was entered into the database is not there.

Causes of wrong permissions are varied: an improper shut down; running Lotto Sorcerer with less than an Administrator's access; a third-party installer incorrectly setting permissions; accessing software that was installed while logged in as another user.

This function responds with either an "All permissions seem to be set properly" message or a specific error message.

How to Invoke

Use the menu item "Utilities > Check IO Permissions...".

Window Controls

Start button

This starts the check process.

Cancel button

This cancels the operation and closes the window.

Lotto Sorcerer Proof-of-Concept

Figure 77.

Overview

Lotto Sorcerer Proof-of-Concept is in the form of a "game" which will detect hidden patterns to your clicks, and predicts which button you will click next. This program uses an extremely simplified version of the same neural network algorithm that Lotto Sorcerer uses.

How to Invoke

Use the menu item "Utilities > Lotto Sorcerer Proof-of-Concept".

Basic Procedure

1. Move the window so that you cannot see the Prediction Side (so that you will not be unduly influenced by the predictions).
2. Click the "Button 1" and "Button 2" buttons in random order.
3. View the results.

File Viewer

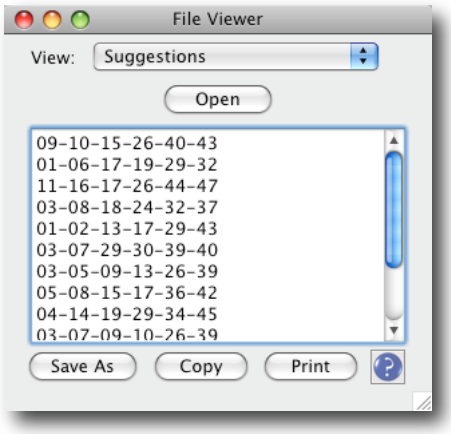


Figure 78.

Overview

This utility lets you view or print log files, suggestions and reports that Lotto Sorcerer generates.

How to Invoke

Use the menu item "Utilities > File Viewer".

Basic Procedure

1. Select the category of document you want to view (Logs, Suggestions, Reports or Other)
2. Click the "Open" button to open the document

Window Controls

View dropdown menu

Documents created by Lotto Sorcerer are stored in specific folders within the Lotto Sorcerer v9 Files folder, which is located in your Documents Folder:

- Logs are stored in the Logs folder.
- Suggestions are stored in the Suggestions folder.
- Reports are stored in the Reports folder.

By choosing the type of document you want to see, the Open button will take you to the appropriate folder.

Open button

This brings up a standard file selector. Choose the appropriate file.

Save As button

This will allow you to save the document at another location or filename.

Copy button

This will copy the displayed document to your computer's system clipboard.

Print button

This will print the displayed document.

☞ You can also "drag and drop" a text file into File Viewer to open.

Proofreader

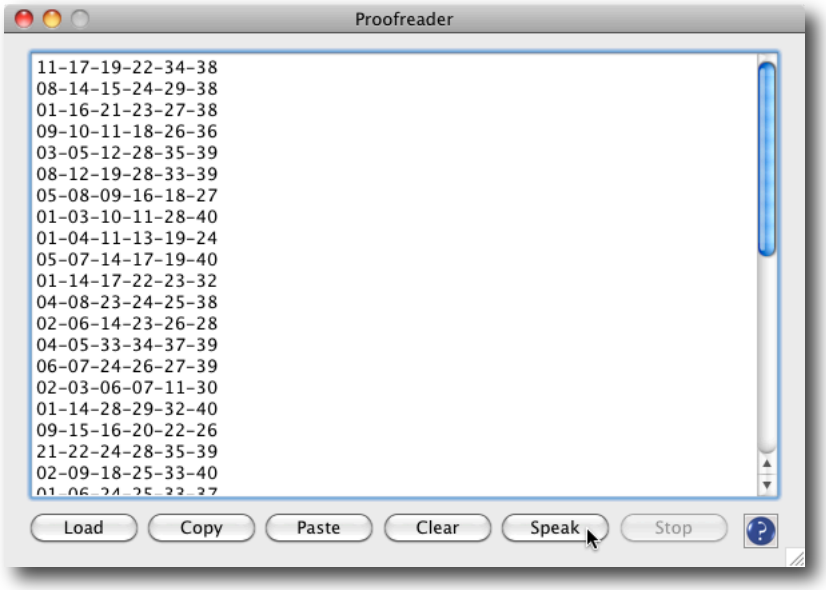


Figure 79.

Overview

This utility helps you proofread lists of data, especially from a printout. It works by actually speaking the text, so that you can follow along, checking the data.

How to Invoke

Use the menu item "Utilities > Proofreader..."

Basic Procedure

1. Load, drag-and-drop or paste the text data into the text box
2. Click the "Speak" button

Window Controls

Load button

Clicking this button opens up a file selector. Select the file you want to proofread.

Copy button

This will copy the displayed document to your computer's system clipboard.

Paste button

This brings up a standard file selector. Choose the appropriate file.

Speak button

Clicking this speaks whatever is in the text box.

Stop button

Clicking this stops the text-to-speech process.

Note

This function passes the text to your operating system's voice-to-speech processor. It will work only if your operating system can handle voice-to-speech commands. Mac OS X has this built-in; Windows may or may not have this (built-in) capability. Linux absolutely requires third-party utilities to accomplish this.

Scripting Laboratory

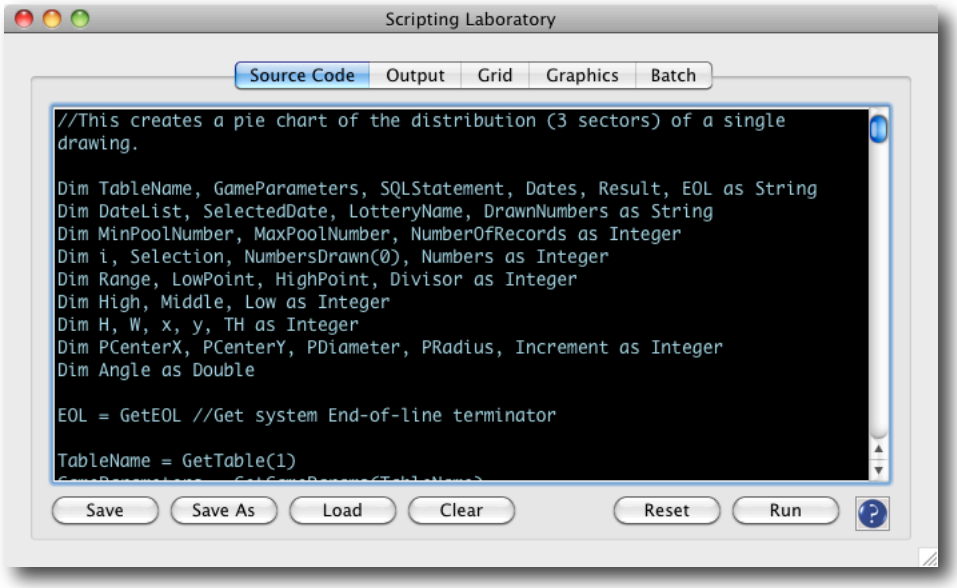


Figure 80.

Overview

This powerful utility lets you write scripts within Lotto Sorcerer. For a detailed explanation of what you can do with scripts, please see Appendix A: LS Script Introduction (page 221). For tutorials, see Appendix B: LS Script Tutorials (page 223). And for the Programmer's Reference Guide, see Appendix C: LS Script Programmer's Reference Guide (page 241).

How to Invoke

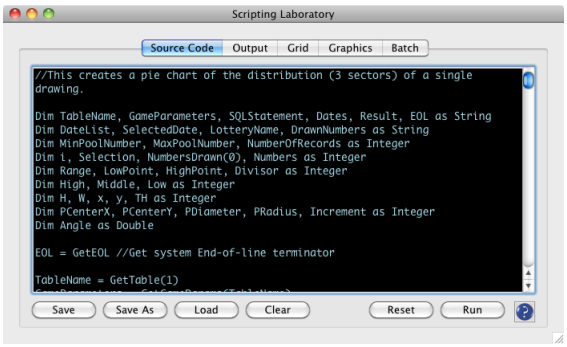
Use the menu item "Utilities > Scripting Laboratory..."

Basic Procedure

1. Enter or load a script into the Source Code window.
2. Click the "Run" button to execute the script.

Window Controls

Source Code Tab



Source Code text area
This is where you can type or edit the script's source code.

Save button
Clicking this button opens up a file selector, allowing you to save your script. It is strongly recommended that you save your scripts with a ".bas" extension.

Save As button
Clicking this button opens up a file selector, allowing you to save your script under a different name. Again, it is strongly recommended that you save your scripts with a ".bas" extension.

Load button
Use this to load an existing script.

Clear button
This will clear the source code text area

Reset button
This clears all variables from memory, clears the Source Code text area, the Output text field on tab 2 of the Scripting Laboratory, and the Grid on tab 3 of the Scripting Laboratory.

Run button
Clicking this executes the script.

Saving, Loading and Running Encrypted Scripts
With the Scripting Laboratory, you can save, load and run two different types of encrypted scripts:

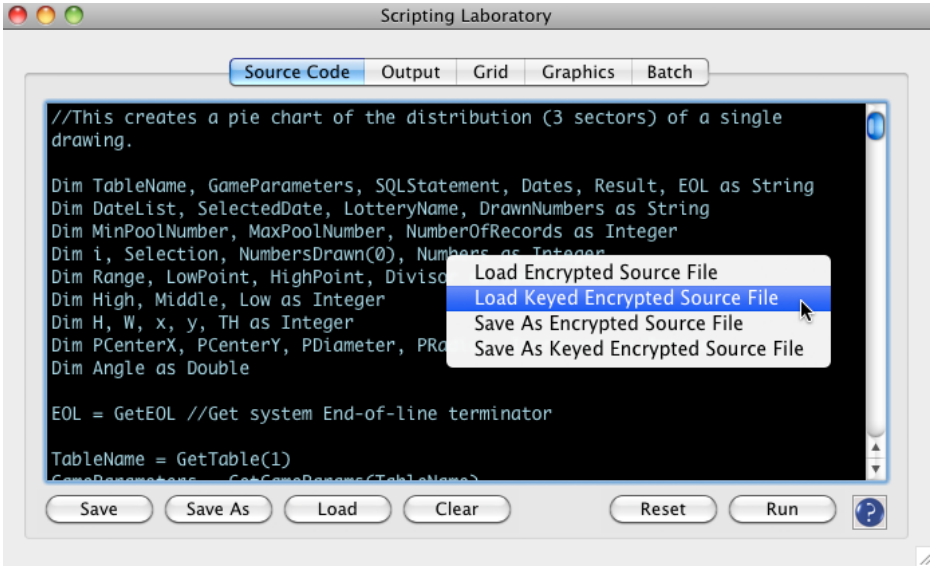
1. Encrypted Scripts
2. Keyed Encrypted Scripts

The first type, "Encrypted Scripts" can be run by anyone with Lotto Sorcerer's Scripting Laboratory. However, anyone loading this type of encrypted script cannot view the source code.

The second type, "Keyed Encrypted Scripts" can only be run by someone who knows the password. **No feedback is given to the user if the person enters the wrong password; the script will simply not work, and will present error messages.**

Important: in both types of scripts, the source code is never visible and cannot be made visible. If you save a script as an encrypted file (either type), you should save an unencrypted version of the script for your own use.

To save or load encrypted scripts, right-click (Windows) or control-click (Mac) in the Source Code text field. You will be presented with a contextual (popup) menu, with four choices:



Load Encrypted Source File

You will be presented with a file selector. Choose the encrypted source file. The source code will not be displayed, but you will be able to run it.

Load Keyed Encrypted Source File

You will be presented with a file selector. Choose the encrypted source file. You will then be prompted for the password (4 digits). There is no feedback if you entered the wrong password. The symptom of entering a wrong password is that the script will not run.

If you enter the password correct, the source code will not be displayed, but you will be able to run it.

Save As Encrypted Source File

This opens up a file selector, allowing you to save the source code as an encrypted file. It is strongly recommended that you use the ".ebas" file extension.

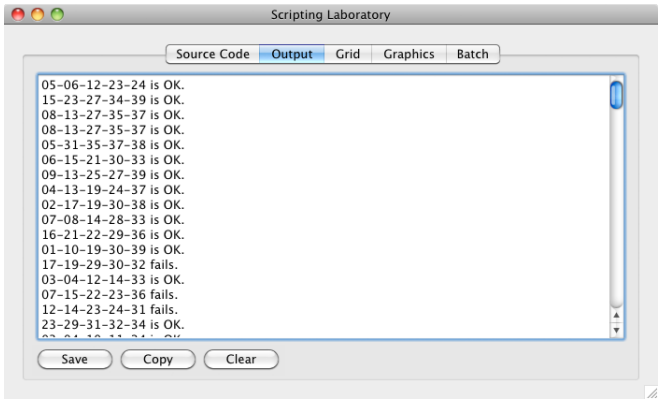
Save As Keyed Encrypted Source File

This opens up a file selector, allowing you to save the source code as an encrypted file. You will be prompted for a password. Enter carefully, there is no verification.

It is strongly recommended that you use the ".ebas" file extension.

Important!
The encryption method used is to be considered low-level encryption. It will keep most prying eyes away, but it is not unbreakable code.

Output Tab



This tab is for displaying text results from the script, using the `PRINT` keyword within the script. Be sure to review the documentation for this command in the Programmer's Reference Guide on page 264.

Once you have data in the Output tab, you have three controls available:

Save Button

This opens up a standard file selector, allowing you to save the contents of the Output text area to a text file.

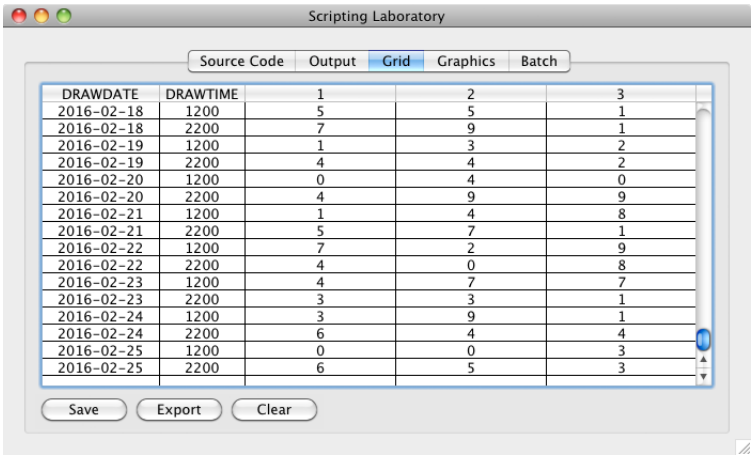
Copy Button

This copies the contents of the Output text area to the System Clipboard.

Clear Button

This clears the Output text area.

Grid Tab



This is another way to output data from your script. You can control the number of columns, names of the columns, the widths of the columns and the alignment of the columns. Methods are described in the Programmer's Reference Guide (page 241) and shown in the third example in the LS Script Tutorial (page 223).

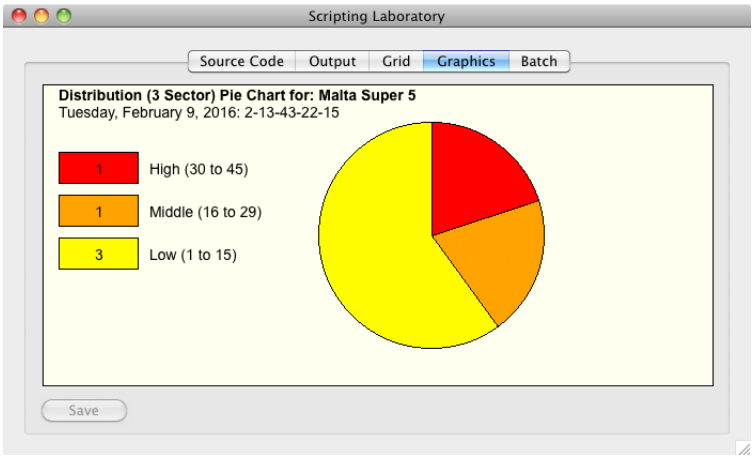
After the grid is populated with data from your script, you have three controls:

Save Button
This invokes a file selector, allowing you to save the grid as a tab-delimited text file.

Export Button
This invokes a file selector, allowing you to save the grid as a Microsoft Excel spreadsheet.

Clear Button
This clears the grid.

Graphics Tab

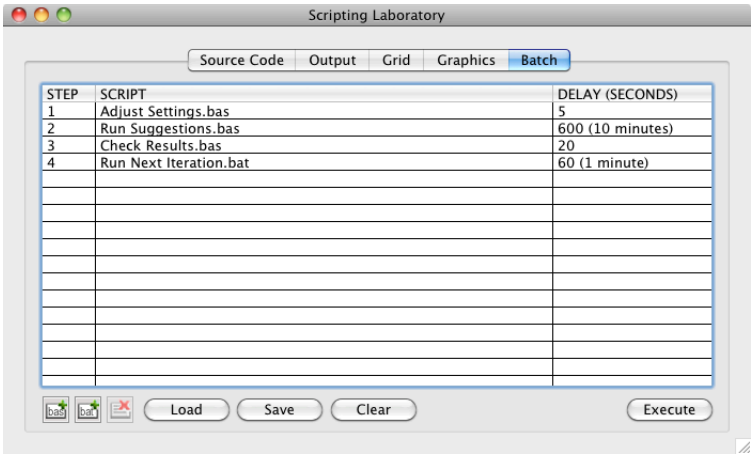


The Graphics tab lets you create your own column, pie, line, area, xy (scatter), bar charts and much more. Methods are described in the Programmer's Reference Guide (page 241) and shown in the fourth example in the LS Script Tutorial.

After the grid is populated with data from your script, you have one control available:

Save Button
This invokes a file selector, allowing you to save the canvas in several different image formats.

Batch Tab



The Batch tab lets you run multiple scripts, sequentially, in a process known as *batch processing.* Furthermore, it give you the ability to run dynamic batches... i.e., batches where you can alter the batch *while the batch is running* (by using the `AppendBatchLine` command.

Not only can you run scripts you specify, you can also invoke another batch file. However, if you do so, *the batch file must be the last item on the list,* because, once this line is invoked, the batch list is cleared and populated with the contents of the batch you invoked. The above screenshot shows three scripts (with the ".bas" extension), followed by one batch file (with the ".bat" extension).

What can you do with batch processing? One possibility would be automating the process where Lotto Sorcerer can run suggestions with different settings, and checking to see which setting yields the best result (most number of wins). To do this manually, you would have to adjust the settings in the Projection Parameters tab in the main window; run the suggestions; check the suggestions to see how many suggestions resulted in wins; go back and adjust the parameters again, run the suggestions again, and so forth. But with batch processing, this process can be completely automated. The entire process would take hours, or even days to run, but it would be able to be run unattended.

There are a few important points to remember when doing batch processing:

Any scripts used must not have any interactive elements in them, where the script stops, waiting for user input. Examples of "interactive elements" would be:

- A *message box,* which waits until the user clicks a button
- A *dialog box* with a popup menu, which waits until the chooses an item
- A *file selector,* which waits until the user chooses a file. If any of your scripts require the ability to read and/or write files, do not use the `GetFile` or `SaveFile` functions, which open up a file selector; use `ReadFile` or `WriteFile` instead, which reads and writes files programmatically, without user interaction.

The delay setting is absolutely critical. The delay is the third column in the Batch tab. This delay does not refer to the delay between scripts, but the delay that the batch processor will wait until launching the next script. If you have a script that takes a long time (for example, suggestion generation), you must make sure that the delay is long enough so that the script can finish.

Scripts and batches must be located in the default location. Scripts that are invoked within the batch file must be located in the "Scripts" folder of the "Lotto Sorcerer v9 Files" folder (which is located in your "Documents" folder. Batch files invoked from within the batch must be located in the "Batch Files" folder.

Only standard (non-encrypted) scripts can be used in a batch.

Add Script Button

This button is at the far bottom-left corner of the window. Click this, and a file selector will open. Choose the script to add to the list.

Add Batch Button

This button located at the bottom of the window, second from the left. Click this, and a file selector will open. Choose the batch to add to the list. Only one batch file can exist in the batch list, and it must be the last item on the list.

Delete Item Button

This button located at the bottom of the window, third from the left. Choose an item in the batch list that you want to delete, then click this button to delete the item from the list.

Load Button

This button loads in a previously saved batch list.

Save Button

This button lets you save the current file list as a batch file.

Clear Button

This button clears the batch file list.

Execute Button

This button executes the batch. The progress of the batch will be shown to the left of the Execute button. To prematurely quit the running batch, just close the Scripting Laboratory window.

Preferences

Preferences - Interface

Figure 81.

Overview

This lets you set interface settings for Lotto Sorcerer.

How to Invoke

Mac OS X

Use the menu item "Lotto Sorcerer v9 > Lotto Sorcerer Preferences", then choose the "Interface" tab.

Windows

Use the menu item "File > Lotto Sorcerer Preferences", then choose the "Interface" tab.

Basic Procedure

1. Select the settings you want
2. Click the OK button

Window Controls

Hide Wheels checkbox

If checked, wheels will not appear in the Generate Suggestions dropdown menu in the Suggestions tab in the Main Window.

Sort Data Entry

If checked this will automatically sort numbers you enter into the database (where appropriate). This effects only standard and keno lotteries and lotteries with bonus or extra numbers (although the bonus or extra numbers will not be sorted).

Remember Window Size/Position

If checked, the main window of Lotto Sorcerer will remember the last size and position (on the screen) which it was last set to.

Hide Minor Informative Alerts

Checking this will prevent minor informative alerts (for example, "the information has been copied to your system clipboard") from appearing.

Show Internet Alerts

If this is checked, a message box will remind you to connect to the Internet before continuing. If you have an "always on" Internet connection (DSL, cable, etc.), you should uncheck this box.

Remote Database

This preference lets you choose which remote database to use. The default is "Primary". If the mirrored remote database is unavailable, a second choice, "Mirror", can be chosen.

The remote database is used by Lotto Sorcerer for updating drawings; checking for program updates; checking subscription status; downloading prior drawings; and installing new lotteries via the Lottery Setup Wizard. If any of these services encounter a problem they will automatically switch to the alternate database.

Default File Location

This dropdown menu selector lets you choose where the Open and Save file dialogs will default to. You have your choice between your Desktop, your Documents Folder or any custom location of your choice. For maximum functionality, we strongly recommend you keep it on the default setting ("Documents Folder").

Preferences - Analysis

Figure 82.

Overview

This lets you set the analysis settings for Lotto Sorcerer.

How to Invoke

Mac OS X

Use the menu item "Lotto Sorcerer > Lotto Sorcerer Preferences", then choose the "Analysis" tab.

Windows

Use the menu item "File > Lotto Sorcerer Preferences", then choose the "Analysis" tab.

Basic Procedure

1. Select the settings you want
2. Click the OK button

Window Controls

Ring on Suggestion Completion

Checking this causes your system bell/beep to sound when the suggestion process is completed.

Separator dropdown menu

Choose the separator between numbers when suggested numbers are displayed.

Precision dropdown menu

Use this dropdown menu to choose how many decimal places to display in most reports.

Thread Priority

You can set the priority of the suggestion generation thread here. The higher the thread priority, the faster Lotto Sorcerer will run when generating suggestions, and the less processing power will be available to other applications on your computer. Conversely, the lower the priority, the slower Lotto Sorcerer will be when generating suggestions (while giving more processor power to your other applications).

Interface Updating

You can set the priority of the interface updating during the suggestion generation process here. The slower the interface updating, the faster Lotto Sorcerer will run when generating suggestions. Conversely, the faster the interface updating, the slower Lotto Sorcerer will be when generating suggestions.

Logging dropdown menu

This preference gives you three choices for logging during the suggestion generation process:

1. None (no logging)
2. Normal (logging)
3. Verbose (logging)

Choosing the "verbose" setting causes Lotto Sorcerer to write a far more verbose log while generating suggestions. This will slow down the generation process, and some lotteries can write log files as long as 100 megabytes. Logs are saved in the "Logs" folder of the "Lotto Sorcerer v9 Files" folder, located in your Documents folder.

Preferences - Auto

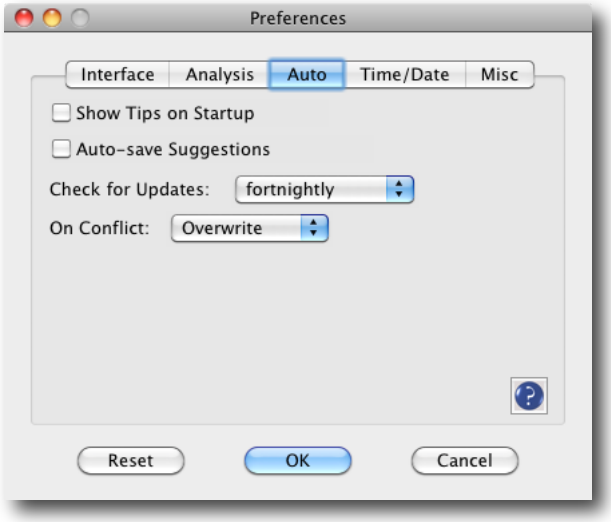


Figure 83.

Overview

This lets you set the automation settings for Lotto Sorcerer.

How to Invoke

Mac OS X

Use the menu item "Lotto Sorcerer > Lotto Sorcerer Preferences", then choose the "Auto" tab.

Windows

Use the menu item "File > Lotto Sorcerer Preferences", then choose the "Auto" tab.

Basic Procedure

1. Select the settings you want
2. Click the OK button

Window Controls

Show Tips on Startup checkbox

If checked, the "Tip of the Day" window will appear at startup. Even if unchecked, the Tips window can be shown by using the menu item "Help > Show Tips".

Auto-save Suggestions checkbox

If checked, when suggestions are generated, a file will be saved of those suggestions. Suggestions are saved in the "Suggestions" folder, of the "Lotto Sorcerer v9 Files" folder, located in your Documents folder.

Check for Updates dropdown menu

Choose how often you want Lotto Sorcerer to check for updates:

- Daily
- Weekly
- Fortnightly
- Monthly

- Quarterly

You can also check manually by using the menu item "Help > Check for Updates".

On Conflict dropdown menu

This dropdown menu gives you a choice of "Append" or "Overwrite" for when a report filename conflicts with an existing filename. If you choose "Append", the report will be appended with the existing file; if you choose "Overwrite", the report will replace the existing report with the new report.

Preferences – Time/Date

Figure 84.

Overview

This lets you set the various time and date settings within Lotto Sorcerer.

How to Invoke

Mac OS X

Use the menu item "Lotto Sorcerer > Lotto Sorcerer Preferences", then choose the "Time/Date" tab.

Windows

Use the menu item "File > Lotto Sorcerer Preferences", then choose the "Time/Date" tab.

Basic Procedure

1. Select the settings you want
2. Click the OK button

Window Controls

Show Extraneous Days checkbox

If checked, calendars will show days from the preceding and following months as ghosted values; otherwise, these days do not show up in the calendar.

First Weekday dropdown menu

This lets you set the first day of the week in the calendar.

Interface Format dropdown

This setting will determine the default way the date is displayed in the interface for the following: Check Numbers; Lotto Seer; Print Lottery Drawing History; Prune Lottery; and the Date Calculator functions. The Drawing History date format is from your computer's date settings. For Windows, this field uses the "Long" date setting on your computer (Control Panel > Region and Language > Format). For Mac OS X, this field uses the "Medium" date setting (System Preferences > Language & Text > Formats).

Port Format dropdown menu

This setting will determine the default date format for importing and exporting functions.

Time Format dropdown menu

This setting will determine the way you want time displayed.

Display Format dropdown menu

This setting will determine the way you want dates displayed. The date definitions are dependent upon your system's preferences, which can be changed.

Preferences – Miscellany

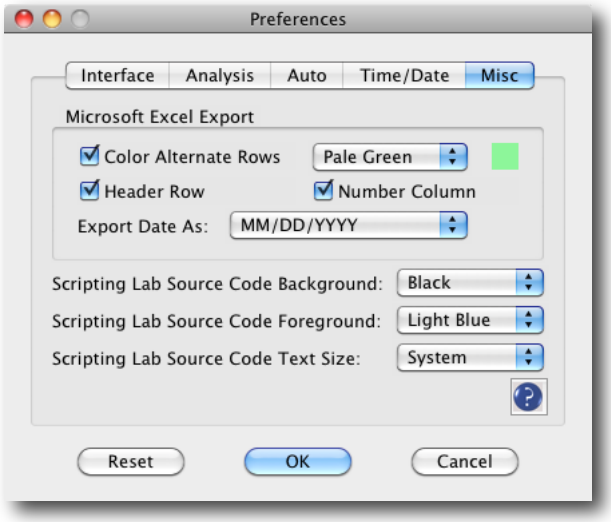


Figure 85.

Overview

This page is for miscellaneous settings.

How to Invoke

Mac OS X

Use the menu item "Lotto Sorcerer > Lotto Sorcerer Preferences", then choose the "Misc" tab.

Windows

Use the menu item "File > Lotto Sorcerer Preferences", then choose the "Misc" tab.

Basic Procedure

1. Select the settings you want
2. Click the OK button

Window Controls

Microsoft Excel Export – Color Alternate Rows checkbox and color selector

If checked, alternate rows in the spreadsheet will be colored chosen in the color selector.

Microsoft Excel Export – Header Row checkbox

If checked, the first row in the spreadsheet will be the title of the columns.

Microsoft Excel Export – Number Column checkbox

If checked, the first column in the spreadsheet will be line number of each row.

Microsoft Excel Export – Export Date As dropdown menu

Choose the desired date format.

Scripting Lab Source Code Background dropdown menu

Choose the desired color for the background of the Source Code tab in the Scripting Laboratory. The default value is "Black".

Scripting Lab Source Code Foreground dropdown menu

Choose the desired color for the foreground (text color) of the Source Code tab in the Scripting Laboratory. The default value is "Light Blue".

Scripting Lab Source Code Text Size dropdown menu

Choose the desired text size for the Source Code tab in the Scripting Laboratory. The default value is "System".

Wheels

Wheels Overview

Wheels are a powerful betting technique that many users have asked for. Lotto Sorcerer has 7,000 built-in optimized wheels (all from the La Jolla Covering Repository) that will work with most lotto-type lotteries.

A lottery wheel (also called a covering) is a pattern or template that offers a certain guarantee based on certain conditions. A lottery wheel is a subset of all possible combinations the lottery can produce.

An Example

It is best to illustrate this by using an actual example; in this case, we will use Indiana's Lucky 5 lottery. This lottery draws 5 numbers from a pool of 36 numbers. This lottery 376,992 combinations, and cost 50¢ per play. We decide to use the wheel "230 wheeled suggestions (4 numbers match if 4/14 numbers drawn)". At 50¢ a play, this would cost $115.

For this wheel, a truncated pool is used: instead of 36 numbers in the pool, the pool is now only 14 numbers. *If four of the five numbers are drawn from this truncated pool of 14 numbers, and we played all 230 wheeled suggestions, we are guaranteed at least one win of 4 numbers matching, a $200 prize.* It is certainly possible that we may have more than one set of 4-number-matching winners, and it is quite probable we will have several 3-number-matching winners, but we are guaranteed to have at least one 4-number-matching winner.

How to "Read" a Wheel Description

Wheels are described as "*n* wheeled suggestions (*t* numbers match if *x*/*z* numbers drawn)"

where

n = number of wheeled suggestions generated;
t = guaranteed number of matching numbers;
x = numbers drawn from the truncated pool;
v = truncated pool size

Although not part of the wheel description, *per se,* the variable *k* is used in the different wheel utilities to refer to the numbers drawn for that particular lottery. For example, if you have a lottery that draws 6 numbers from 1 to 48, *k* = 6.

Usage

To use a wheel, select it from the "Generate" dropdown menu in the Projections Parameters tab of the Main window. Anything that is not a "discrete" suggestion is a wheel.

You can edit the existing wheels using the Wheel Editor (menu item "Utilities > Wheels > Wheel Editor"), or create your own by using the Wheel Creator (menu item "Utilities > Wheels > Wheel Creator").

Lotto Sorcerer also has a "Lotto Wheeler" which lets you generate your own wheels, completely independent from Lotto Sorcerer's own suggestions. To invoke, use menu item "Utilities > Wheels > Lotto Wheeler".

Notes

If you wish to play a wheel in its entirety, *all filters must be turned off.* Why? Because Lotto Sorcerer will create the wheeled suggestions, and then apply the filters.

☞ The word "guarantee", used in the program, only has mathematical meaning and implies no legal liability.

Wheel Creator

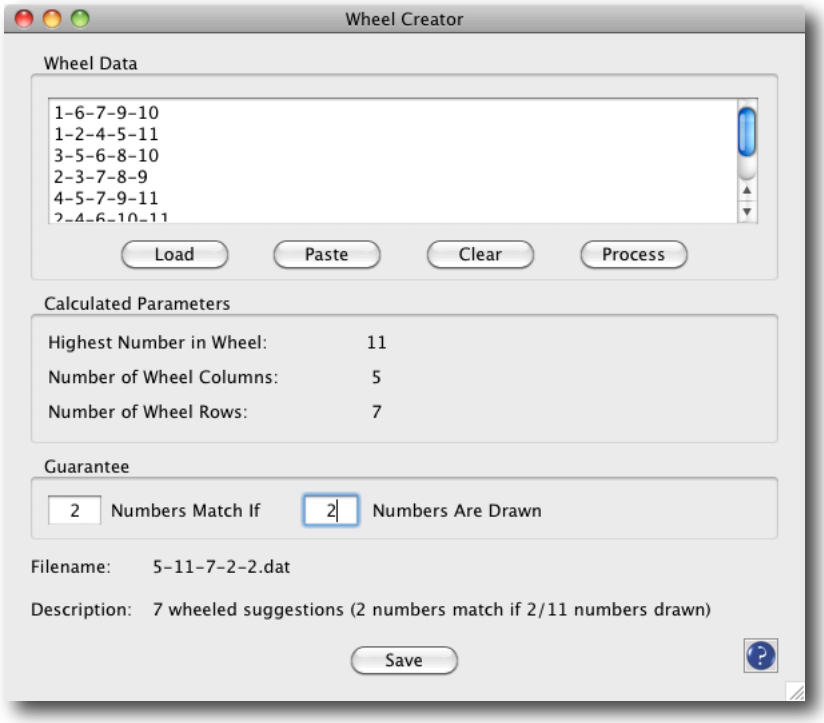


Figure 86.

Overview

This is used to create your own wheel.

How to Invoke

Use the menu item "Utilities > Wheels > Wheel Creator".

Basic Procedure

1. Enter, Load or paste the wheel data in the "Wheel Data" text box
2. Click the Process button to determine the "Calculated Parameters"
3. Enter the guarantee values in the Guarantee section
4. Click the "Save" button to save the wheel

Notes

It is imperative that you adhere to certain rules if you want to modify a wheel, otherwise the adage "Garbage in, garbage out" will apply:

Data entered must be two digit numbers, from 1 to 99. Numbers less than ten must be preceded by a zero (so "8" becomes "08".

Wheel numbers must be consecutive; if it is 12 number wheel, the wheel must contain entries between "01" and "12", inclusive.

Numbers must be separated by a dash ("-").

☞ The word "guarantee", used in the program, only has mathematical meaning and implies no legal liability.

Wheel Editor

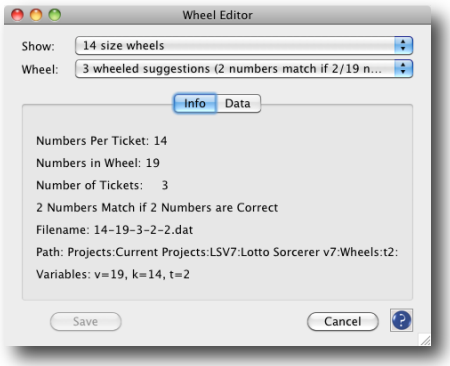


Figure 87.

Overview

This is used to edit or alter an existing wheel.

How to Invoke

Use the menu item "Utilities > Wheels > Wheel Editor".

Basic Procedure

1. Choose the wheel you want to work with in the "Wheel" dropdown
2. Make any changes you want in the spreadsheet-like grid
3. Click the "Save" button when finished

Notes

It is imperative that you adhere to certain rules if you want to modify a wheel, otherwise the adage "garbage in, garbage out" will apply:

- Data entered must be two digit numbers, from 1 to 99. Numbers less than ten must be preceded by a zero (so "8" becomes "08".
- Wheel numbers must be consecutive; if it is 12 number wheel, the wheel must contain entries between "01" and "12", inclusive.

Lotto Wheeler

Figure 88.

Overview

Lotto Wheeler lets you generate your own wheels, completely independent from Lotto Sorcerer's suggestions.

How to Invoke

Use menu item "Utilities > Wheels > Lotto Wheeler".

Basic Procedure

1. Choose the wheel you want to work with in the "Choose Wheel" dropdown menu
2. Enter the seeds in the text box in the Parameters tab
3. Click the Fill button
4. When finished, the generated wheel will be visible in the Results tab

Window Controls

Parameters Tab: Choose Wheel dropdown menu

Use this control to select a wheel. If you do not find a wheel you want, you can create your own wheel (or edit an existing wheel) use menu item Utilities > Wheel Creator or Utilities > Wheel Editor, respectively.

Parameters Tab: Fill button

This button starts the generation of the wheel.

Parameters Tab: Clear button

This button clears the Seed boxes.

Results Tab: Copy button

This copies the generated wheel to the System Clipboard.

Results Tab: Save button

This button saves the wheel to a text file.

Results Tab: Print button

This prints the wheel to your printer.

Wheel Conjuror

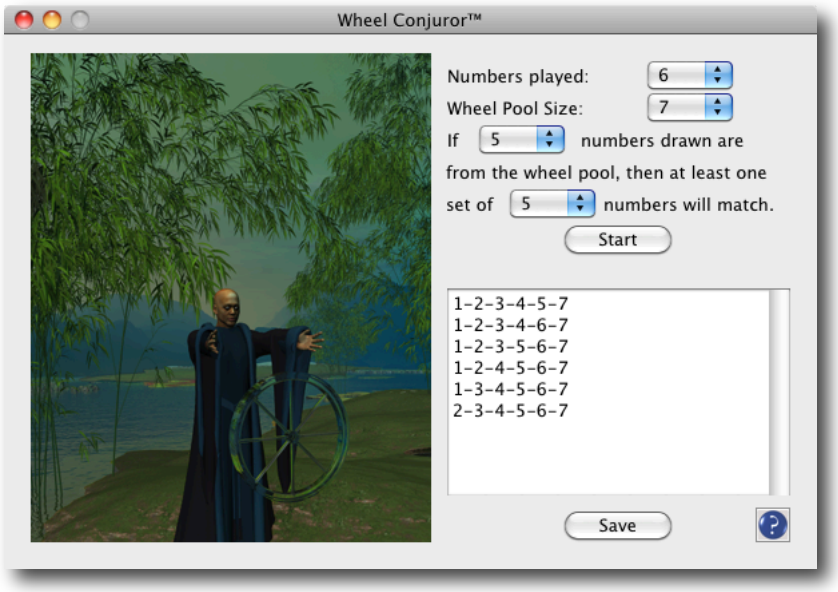


Figure 89.

Overview

Wheel Conjuror gives you the ability to easily create your own wheels. You can create wheels for lotteries from 4 to 6 numbers, with the wheel pools from 5 to 24 numbers.

How to Invoke

Use menu item "Utilities > Wheels > Wheel Conjuror".

Basic Procedure

Enter the wheel parameters in the appropriate dropdown menus.

1. In the "Numbers Played" dropdown, enter the number of numbers you play in your lottery. For example, if your lottery plays 6 numbers out of 48, choose "6". *Note:* do not count bonus ball(s)
2. Choose the wheel pool size in the "Wheel Pool Size" dropdown. The wheel pool size must always be less than the lottery pool size, of course
3. Choose the guarantee parameters in the two remaining dropdowns
4. Click the "Start" button to begin the wheel generation process. When the generation is complete, the wheel will be shown in the box at the bottom of the window
5. Click the "Save" button to save the completed wheel to Lotto Sorcerer. The wheel will now be found as a choice in the Main Window (in the "Generate" dropdown in the Suggestions Tab)

Notes

- Large wheel generations can take a long time. If you need to cancel the process, just close the Wheel Conjuror window
- Although Wheel Conjuror will optimize the wheel, please note that the wheels will not be maximally optimized; fully optimized wheels are an intensive process, and is actually an ongoing study as a field within statistics

Wheel Importer

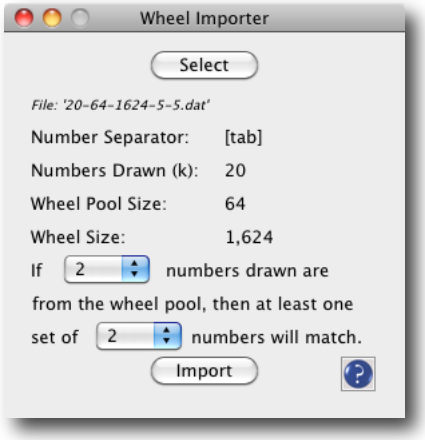


Figure 90.

Overview

Wheel Importer gives you the ability to easily import wheels from a text file.

How to Invoke

Use menu item "Utilities > Wheels > Wheel Importer".

Basic Procedure

1. Select the wheel file to import. Lotto Sorcerer will automatically detect the number delimiter.
2. Select the wheel guarantee parameters.
3. Click the "Import" button.

☞ The word "guarantee", used in the program, only has mathematical meaning and implies no legal liability.

Wheel Explorer

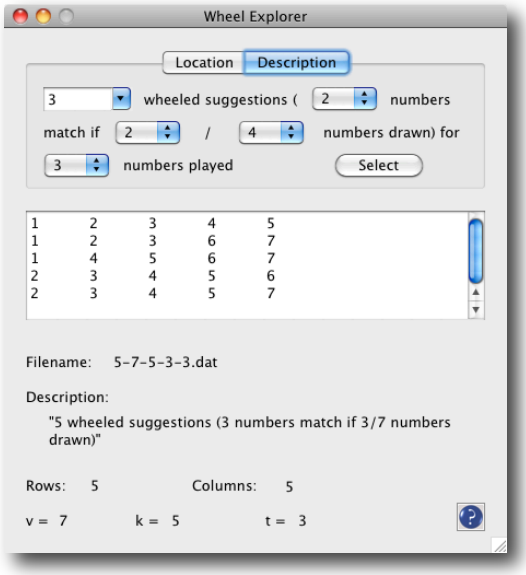


Figure 91.

Overview

This function lets quickly view a wheel and its settings.

How to Invoke

Use menu item "Utilities > Wheels > Wheel Explorer".

Basic Procedure

There are three ways of loading in a wheel:

1. Dragging and dropping a wheel in the wheel text box in the window.
2. Using the Location tab, choose the location (Documents or Program), then click the "Select" button. A standard file selector will appear, from which you can select the wheel.
3. Using the Description tab, choose the parameters of the wheel, and click the Select button.

Wheel Exporter

Figure 92.

Overview

Wheel Exporter gives you the ability to export any of Lotto Sorcerer's 7,000 wheels to an external text file.

How to Invoke

Use menu item "Utilities > Wheels > Wheel Exporter".

Basic Procedure

Enter the wheel parameters in the appropriate dropdown menus.

1. Choose the wheel that you want to export in the dropdown menu at the top of the window.
2. Choose the Number Separator that you want.
3. Choose the End-of-Line Terminator that you want to use.
4. Click the Export button, and choose the file that you want to export the wheel to.

Rebuild Wheel Table

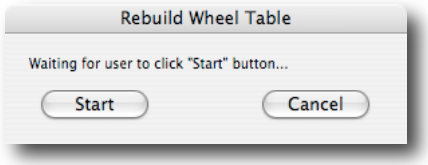


Figure 93.

Overview

This function allows you to rebuild the Wheels table. Generally, you should never have to use this function, but if you have manually added or deleted wheels from Lotto Sorcerer's folders, outside of Lotto Sorcerer, you should run this. That being said, it never hurts to run this function.

How to Invoke

Use menu item "Utilities > Wheels > Rebuild Wheels Table".

Basic Procedure

Click the Start Button.

Verify Wheel

Overview

This function checks your wheel for invalid parameters:

- Invalid sequencing of values
- Values larger than that allowed
- Wheels with incorrect number of lines
- Wheels with missing values

How to Invoke

Use menu item "Utilities > Wheels > Verify Wheel".

Basic Procedure

A standard file selector will open. Choose the wheel you want to verify.

Verify Wheel Table

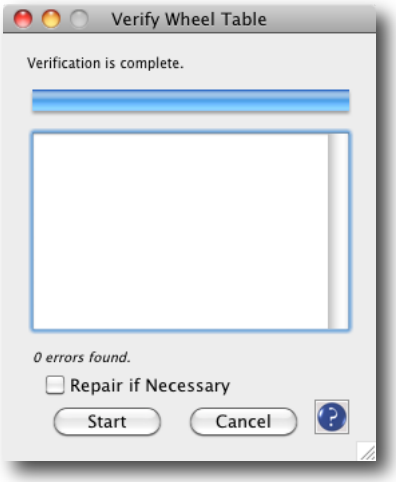


Figure 94.

Overview

This function checks the entire wheel table on the Lotto Sorcerer system (built-in as well as custom). It never hurts to run this function.

How to Invoke

Use menu item "Utilities > Wheels > Verify Wheel Table".

Basic Procedure

Click the Start Button.

Window Controls

Repair if Necessary checkbox

If this box is checked and if an invalid wheel is found, that wheel will be deleted from the wheel table.

☞ This repairs the table itself (by deleting an invalid wheel); it does not repair the wheel itself.

Start button

This button starts the verification process.

Cancel button

This button closes the window.

Playslips

Playslips Overview

Lotto Sorcerer can print directly on playslips, if, and only if:

- Your printer can handle the paper the playslips are printed on:
- Your printer (and its printer driver) can print close enough to the edge of the playslip

Lotto Sorcerer comes with definition files for over well over 300 different lotteries. But the playslips will probably need "tweaking" on your part (using the Playslip Setup Wizard) to get them to print properly with your operating system, printer and version of your printer driver.

To print on the playslip, click on the "Print Playslips" button on the "Suggestions" tab in the Main Window, after the suggestions have been generated.

Not only can you print the numbers you want to play, but also "special marks". An example of special marks would be the marks some playslips have for marking the evening or midday draw.

To keep you from having to setup the coordinates for every single mark on the playslip, Lotto Sorcerer asks for minimal data, then calculates the marks for every single number. This entails having a symmetrical playslip. But Lotto Sorcerer also allows you to override these marks, in the event that you have an asymmetrical playslip.

Playslip Setup Wizard

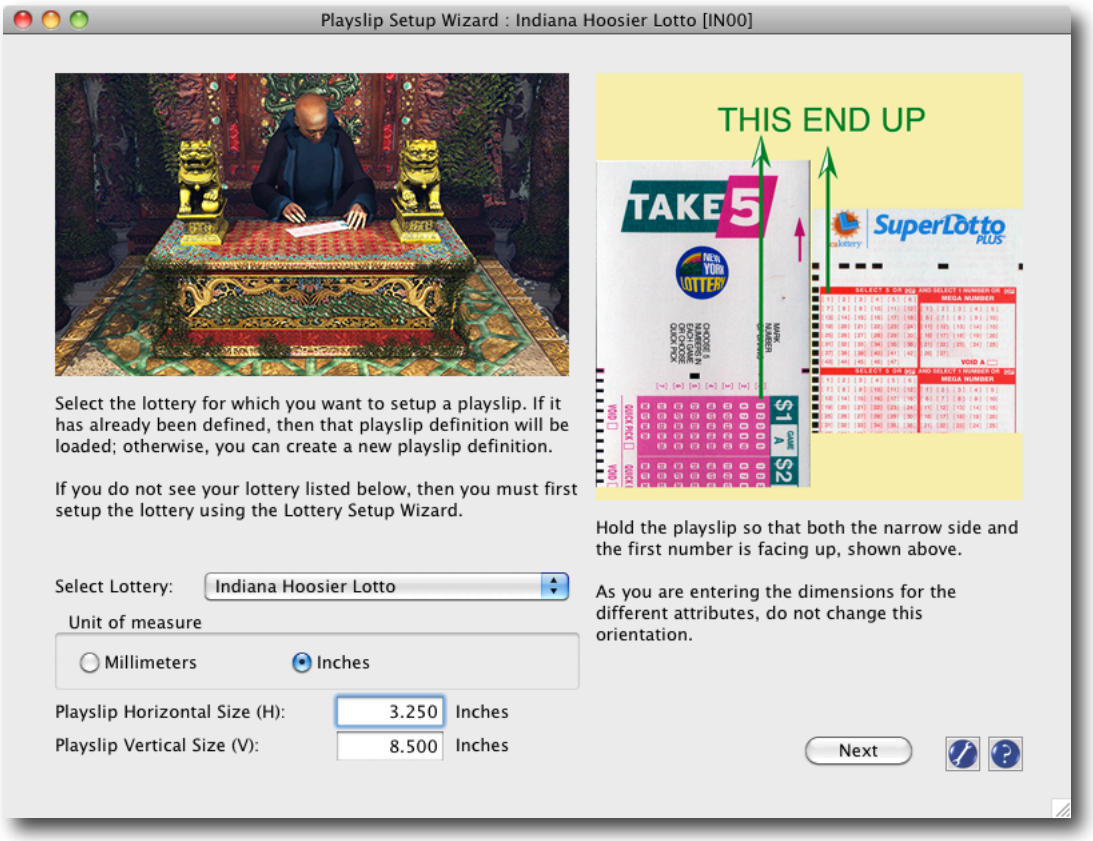


Figure 95.

The Playslip Setup Wizard can make it easy to create or edit lottery playslip definition files.

How to Invoke

Use the menu item "Utilities > Playslips > Playslip Setup Wizard".

Basic Procedure

1. Select the lottery that you want to work with in the "Select Lottery" dropdown menu on the first page of the Wizard.
2. Enter playslip settings shown on each page that you are shown.
3. Preview, test print and save the settings on the last page of the Wizard.

Notes

- When you select the lottery you want to work with on the first page of the Wizard, the Wizard will load in the current settings for that playslip, if settings exist; if not, a new playslip definition file will be created.
- All dimensions must be entered in decimal format. For this reason, you may find it easier to work in millimeters (instead of inches).
- On the final page of the Wizard, the Wizard will show you a preview of the playslip before you can test print or save it. This preview should closely resemble your playslip with all boxes marked.

☞ Each page of the Playslip Setup Wizard has a help file for that page. Consult those help files by clicking on the Help icon at the bottom right of each page.

Page 1: Playslip Properties

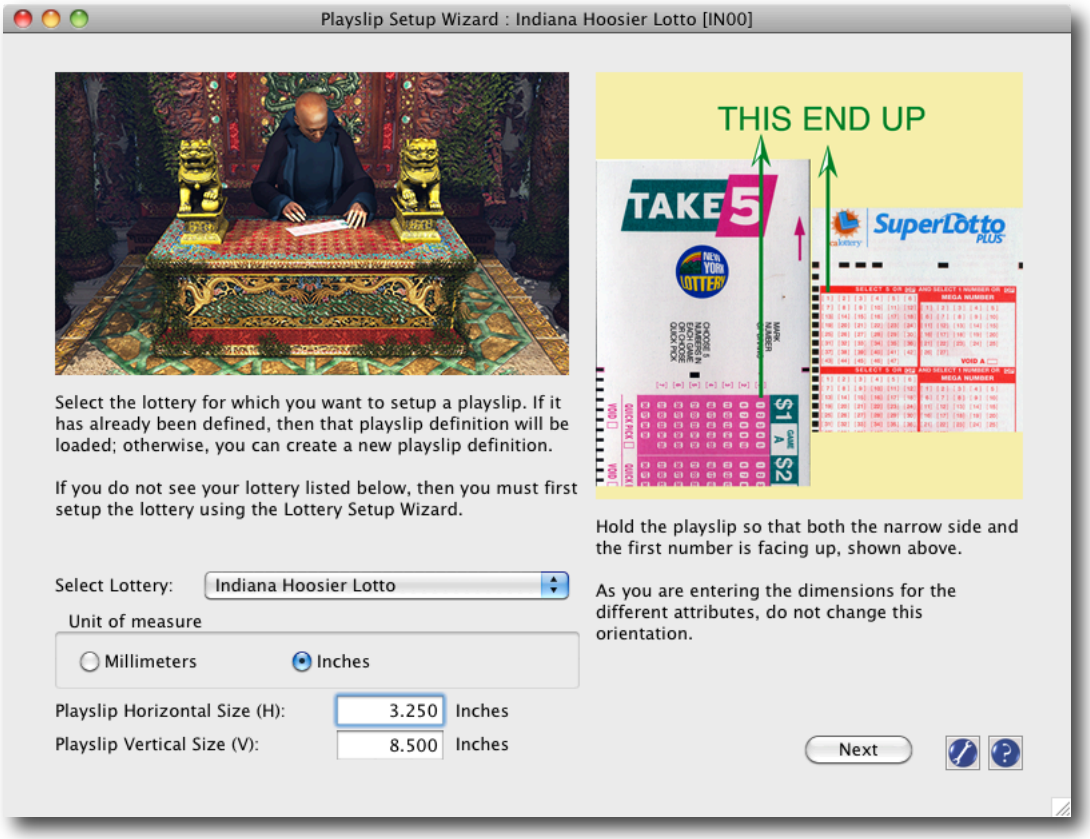


Figure 96.

Overview

Use this "page" to set the playslip's dimensions and unit of measure.

Basic Procedure

1. Hold the playslip so that the narrow side and the first number of the first game are facing up.
2. Select the lottery from the "Select Lottery" dropdown menu. If you do not see your lottery listed, you must set it up in the Lottery Setup Wizard.
3. Choose the units of measure you want to use.
4. Enter the horizontal width and vertical height.
5. Click the "Next" button.

Page 2: Board Properties

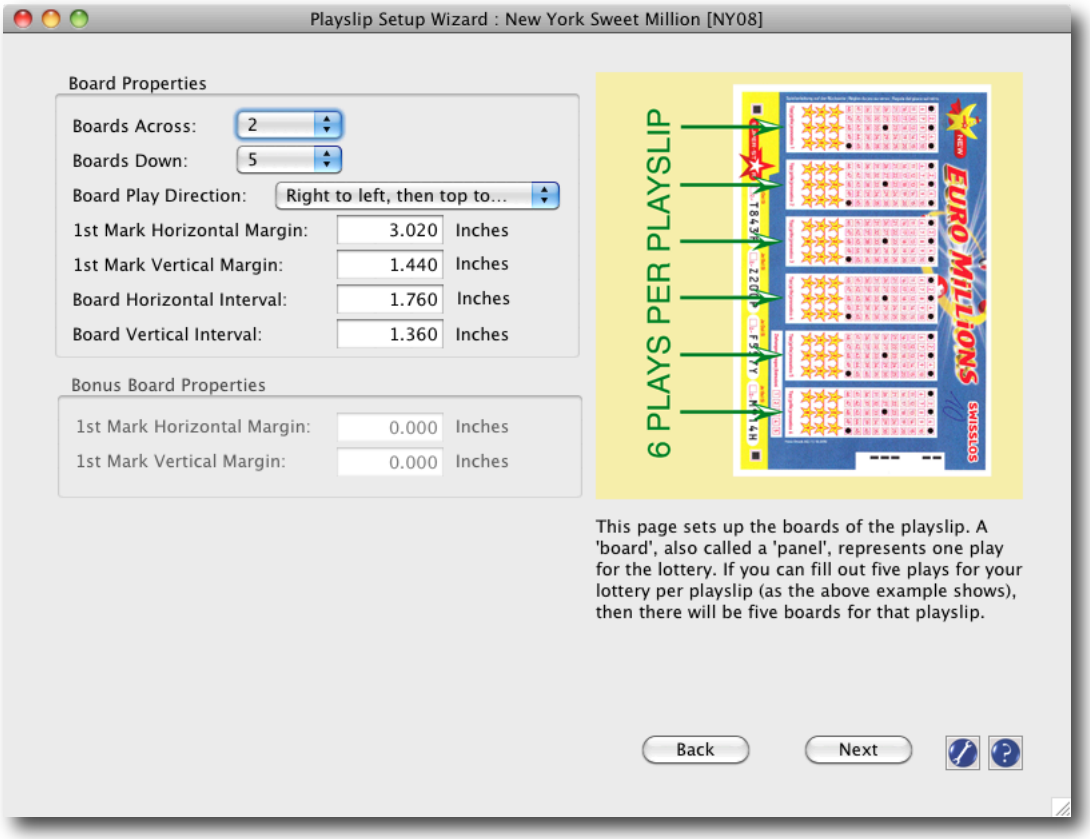


Figure 97.

Overview

A "board", or "panel" is the collection of marks that make up one game.

Basic Procedure

1. Enter the number of boards across and boards down in the first two dropdown menus.
2. Select the direction of play for the boards. Some playslips insist that the second game to be marked is to the left of the first game; others, the second game is the one right below the first.
3. Enter the horizontal and vertical margins of the first mark. A "margin" is the distance from the mark to the edge of the playslip.
4. Enter how far apart the boards are apart, horizontally and vertically.
5. If this playslip is for a bonus-type lottery, enter the margins for the first bonus mark of the first game.*
6. When finished with this page, click the Next button.

*if you wish to prevent the printing of bonus numbers, set the "1st Mark Horizontal Margin" in the "Bonus Board Properties" setting in the Playslip Setup Wizard to zero ("0").

Note

Some controls become "ghosted", or greyed-out, if your prior selection renders that value unnecessary. For example, if your playslip has only one column of boards, the "horizontal interval" between boards is irrelevant.

Page 3: Mark Properties

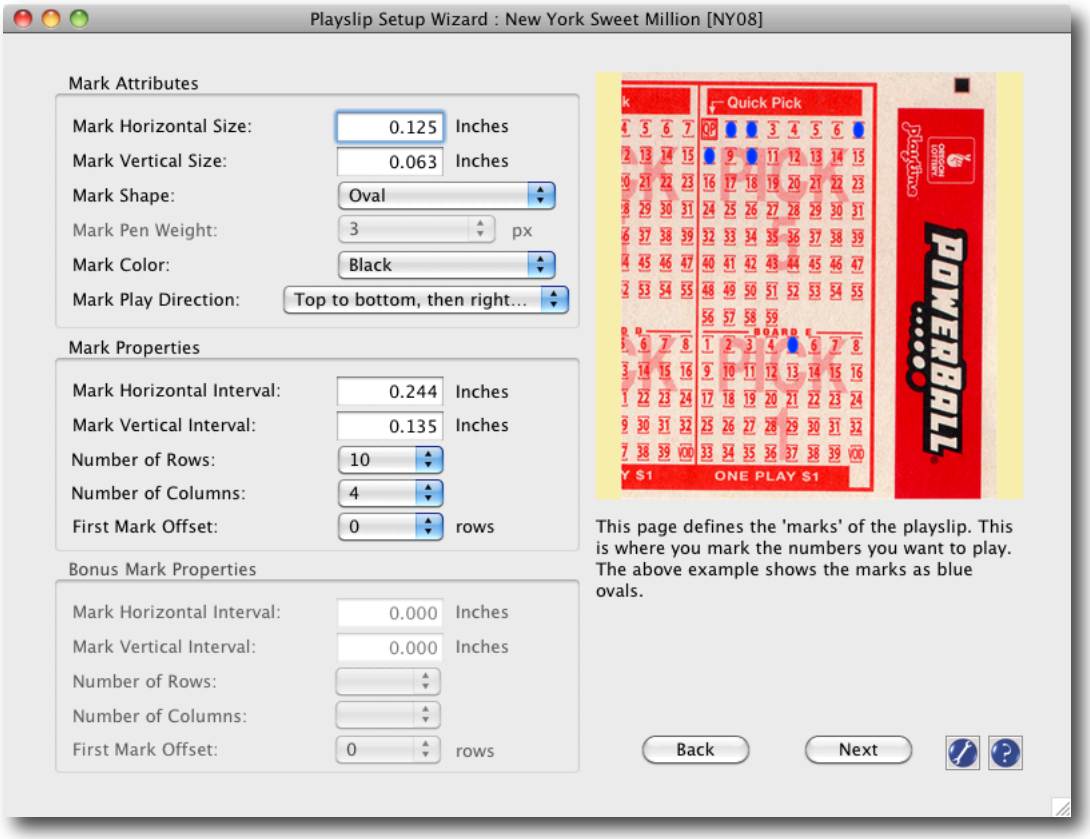


Figure 98.

Overview

A "mark" is the tally you draw (usually a block or oval) on a number space to signify that you want to play that number.

Basic Procedure

1. Enter the horizontal width and vertical height of the mark itself.
2. Select the shape of the mark (either oval/circle, block, "X", checkmark, or a horizontal or vertical line).
3. If you chose "X" or a checkmark or line for the shape of the mark, select the pen weight (thickness) of the X or line.
4. Select the mark play direction.
5. Enter the horizontal and vertical interval, or distance, between marks.
6. Enter the number of rows and columns of marks per board.
7. If the first mark is offset from the other marks, enter the offset.
8. If your lottery is a bonus-type lottery, enter the interval, offset and number of rows and columns in the "Bonus Mark Properties" section.
9. When finished with this page, click the Next button.

Page 4: Special Marks

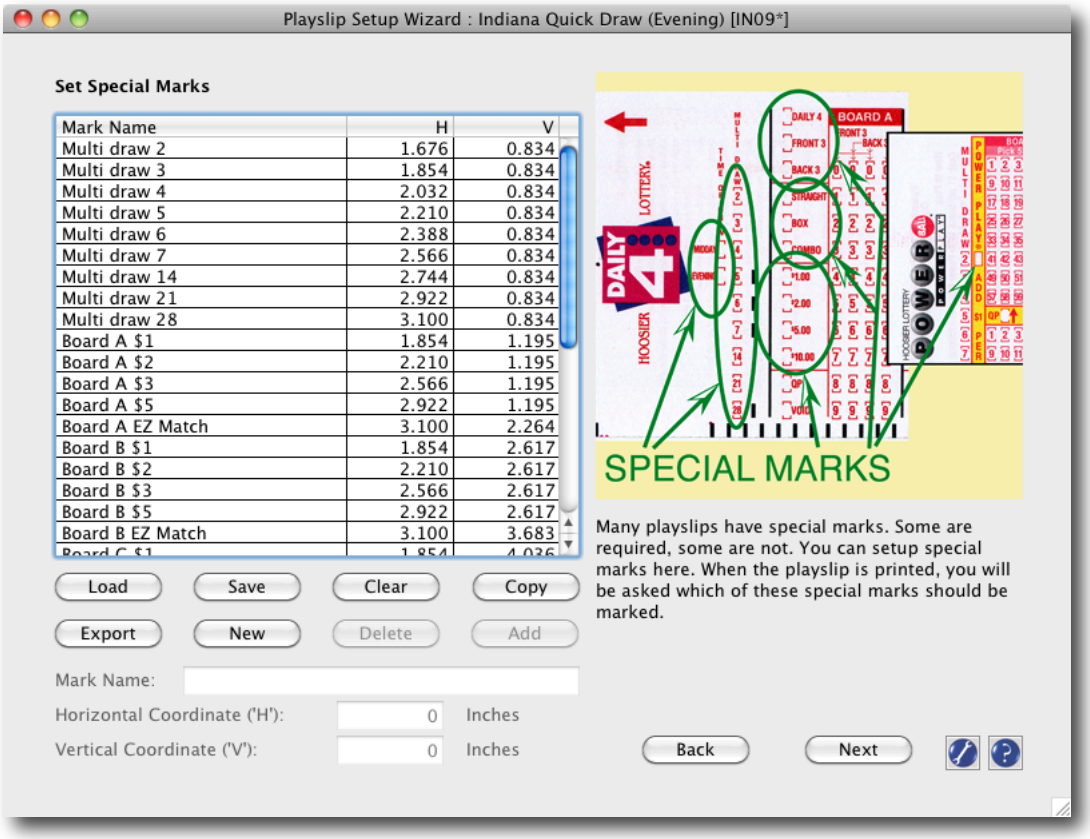


Figure 99.

Overview

Many playslips have "special marks", that is, marks for other than the numbers you want to play. Some marks are required. For example, you may be expected whether you want to play the evening or midday version of the game. Others may be optional; for example, on some playslips you can select a box for a separate bet.

Basic Procedure

1. Click the "New" button to setup a special mark.
2. In the Mark Name box, enter a descriptive name for the mark. For example, "Evening draw" or "Power Play".
3. Enter the horizontal and vertical coordinates for the mark; that is, the distance from the center of the mark to the left edge and top edges of the playslip, respectively.
4. Click the "Add" button to add it to the Special Marks list at the top of the window.
5. When finished, click the "Next" button.

Editing a Value on the List

1. Click on the mark that you want to change; it will populate the boxes at the bottom of the window.
2. Make your changes.
3. Click the "Change" button to push the value back into the list.

Deleting a Value on the List

1. Click on the mark that you want to change; it will populate the boxes at the bottom of the window.
2. Click the "Delete" button to push the value back into the list.

Saving the List

If you have a lot of values to deal with, and if you are skilled at working with a spreadsheet, you can export the list. Then, import the list into your spreadsheet. To do this, just click the "Save" button. It will save the list to a tab-delimited file.

Loading a List

Once you have done working with your list in your spreadsheet, export it from your spreadsheet as a three-field tab-delimited text file. Then, load that file into the Playslip Setup Wizard by clicking the "Load" button.

Exporting the List

You can also export the list directly into a Microsoft Excel spreadsheet. To do this, just click the "Export" button.

Page 5: Overrides

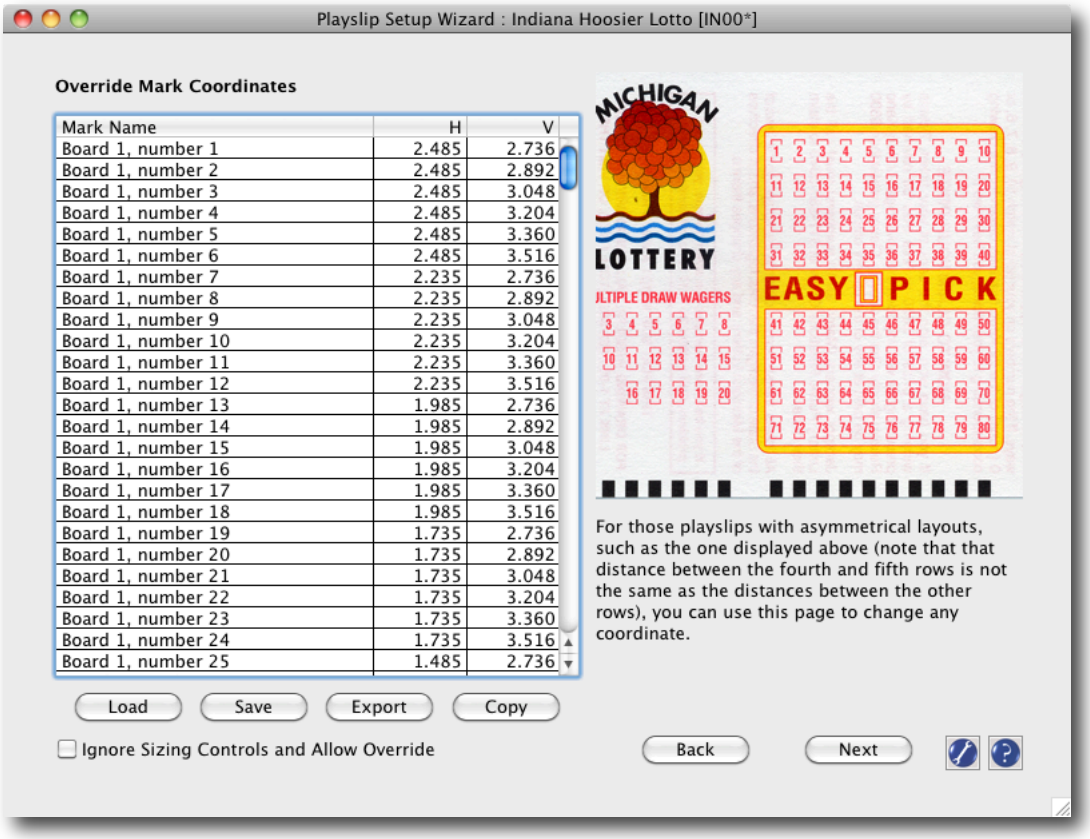


Figure 100.

Overview

If your lottery plays number 1 through 48, with 5 boards (games) per playslip, there would be 240 coordinates to enter! But you do not have to enter all 240, because the values you entered earlier (distance between marks, distance between boards, distances from the playslip's edge, etc.) allows Lotto Sorcerer to calculate each coordinate. But this assumes that the playslip is symmetrical (that is, the distance between all of the marks and the boards are the same).

But sometimes the playslip is asymmetrical. This page, "Override Mark Coordinates" allows you to modify the calculated value of each mark, if you wish.

Important! If you want to make changes in this fashion, you must check the "Ignore Sizing Controls and Allow Override" checkbox. If you do not do this, any changes you make in the first four pages (or even just loading in the playslip) will overwrite any changes you made in this page.

If you do **not** want to make any overrides, you must not check the "Ignore Sizing Controls and Allow Override" checkbox. If you do, any changes you make in the first four pages will not do anything.

Basic Procedure

1. Double-click on the value(s) to change in the list.
2. Check the "Ignore Sizing Controls and Allow Override".
3. When finished, click the "Next" button.

Editing a Value on the List

1. Just double-click on the value you want to change in the list.

2. Type your changes.

Saving the List

If you have a lot of values to deal with, and if you are skilled at working with a spreadsheet, you can save the list as a tab-delimited text file. Then, load the list into your spreadsheet. To do this, just click the "Save" button. It will export the list to a four-field tab-delimited file.

When working on the spreadsheet, be sure to modify *only* that last two fields (the horizontal and vertical measurement fields). Do not modify the first two fields.

Exporting the List

By clicking the "Export" button, you can save this list directly as a Microsoft Excel spreadsheet.

Loading a List

Once you have done working with your list in your spreadsheet, export it from your spreadsheet as a four-field tab-delimited text file. Then, import that file into the Playslip Setup Wizard by clicking the "Load" button.

Page 6: Test and Finish

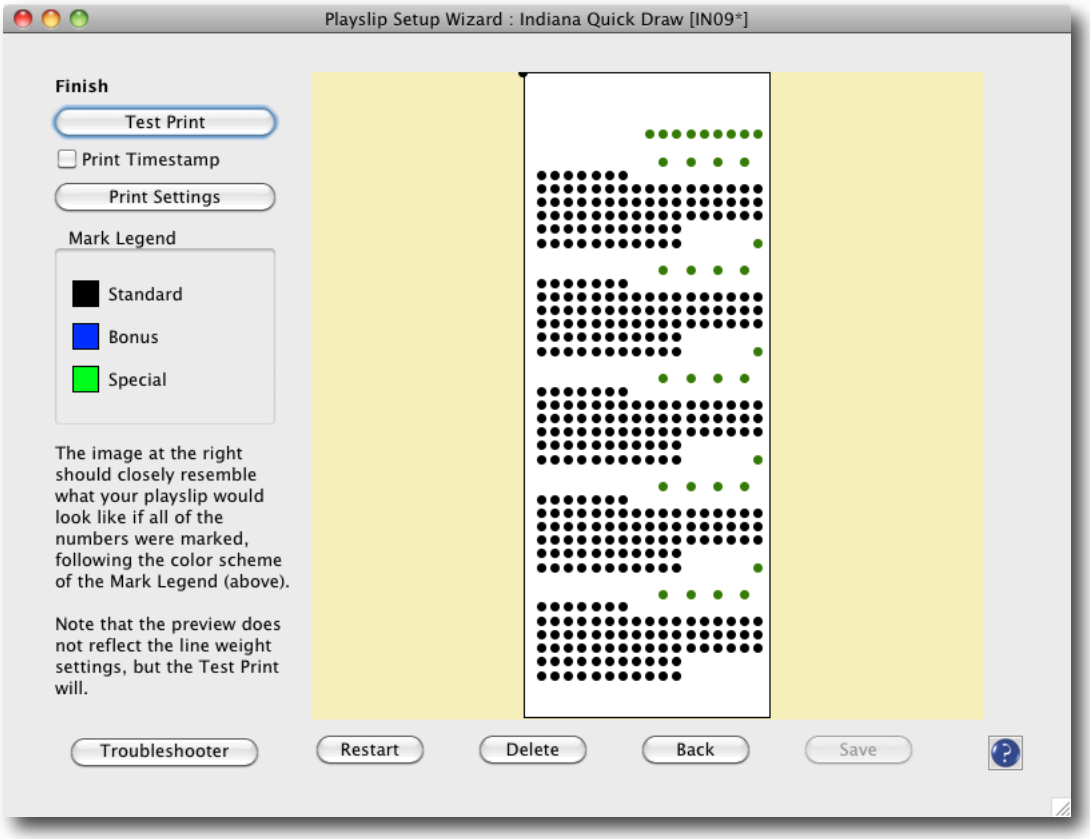


Figure 101.

Overview

This page lets you preview your settings by displaying on the screen every mark of the playslip marked.

Basic Procedure

1. Click "Test Print" to print what you see to a printer.
2. If everything is "OK", click the "Save" button; or click the "Back" button to make changes.

Window Controls

Test Print button

This prints a playslip with all of the marks of the playslip marked to your printer.

Print Timestamp checkbox

This prints the current date and time on the printout.

Print Settings button

Instead of printing the marks, this will print out the playslip settings of this playslip.

Troubleshooter button

Clicking this button will take you to our Playslip Troubleshooter webpage.

Restart button

This goes back to the beginning (Page 1) with a blank slate.

Delete button

This will delete the playslip file altogether.

Back button

This goes back to the prior page.

Save button

This saves the playslip file.

Notes

- The test display does not take into account the pen thickness settings from page 3. The printed version (from clicking the "Test Print" button) does reflect these settings.
- The test display does not take into account the mark color settings from page 3. Instead, it follows the color scheme of the Mark Legend section. The printed version (from clicking the "Test Print" button) does reflect these settings.

Calibrate Printer

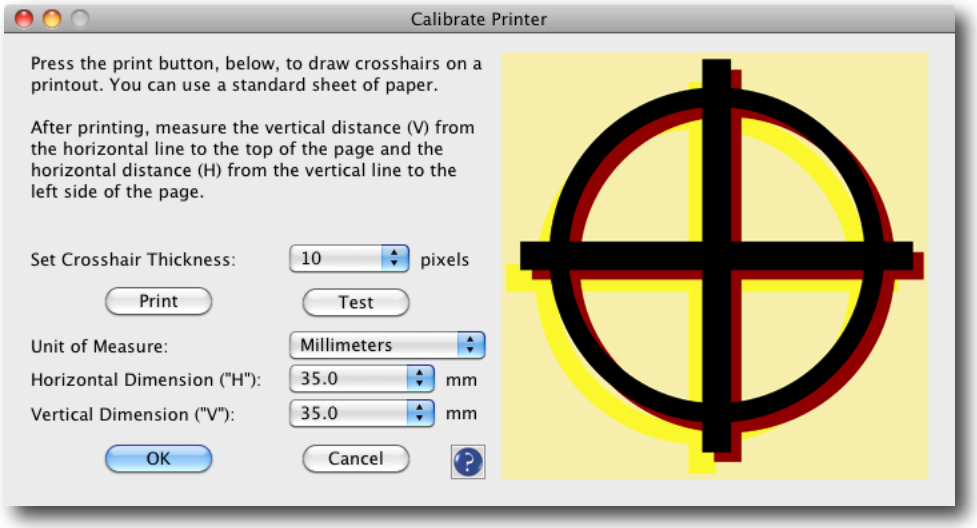


Figure 102.

Overview

This function will print out a crosshair on a sheet of paper. By entering the horizontal and vertical dimensions of this crosshair, Lotto Sorcerer can compensate for the accuracy of your printer.

How to Invoke

Use the menu item "Utilities > Playslips > Calibrate Printer".

Basic Procedure

1. Click the "Print" button.
2. Measure and submit the true horizontal and vertical dimensions of the crosshairs.
3. Click the "Test" button to verify.
4. Click the "OK" button to finalize.

Note

You do not need to use an actual playslip for this calibration procedure; you can use any paper that is compatible with your printer.

Import v6 Playslip Settings

Figure 103.

Overview

This lets you import your Lotto Sorcerer v6 playslip file into Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Playslips > Import Playslip Settings > Import v6 Playslip Settings".

Basic Procedure

1. Select the Lotto Sorcerer v6 Playslip file by clicking the "Select" button
2. Select the Lotto Sorcerer v9 lottery for which this playslip is intended by using the "Select v9 Lottery" dropdown
3. Click the "Import" button

Window Controls

Select button

Click this button to select the Lotto Sorcerer v6 Playslip file you want to import.

Select v9 Lottery dropdown

Choose the v9 lottery for which this playslip is intended. This lottery must already have been setup. If it has not been setup, use the Lottery Setup Wizard to do so.

Import button

This will import and translate those settings into a Lotto Sorcerer v9 Playslip file.

Note

There are additional settings that v9 needs that is not in the v6 file. You will need to go to the Playslip Setup Wizard to insert those settings.

Import v7 Playslip Settings

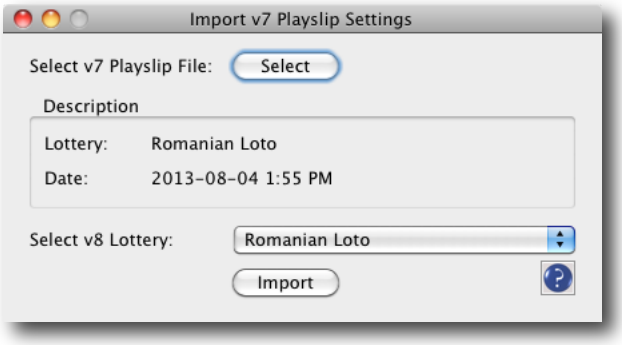


Figure 104.

Overview

This lets you import your Lotto Sorcerer v7 playslip file into Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Playslips > Import Playslip Settings > Import v7 Playslip Settings".

Basic Procedure

1. Select the Lotto Sorcerer v7 Playslip file by clicking the "Select" button
2. Select the Lotto Sorcerer v9 lottery for which this playslip is intended by using the "Select v9 Lottery" dropdown
3. Click the "Import" button

Window Controls

Select button

Click this button to select the Lotto Sorcerer v7 Playslip file you want to import.

Select v9 Lottery dropdown

Choose the v9 lottery for which this playslip is intended. This lottery must already have been setup. If it has not been setup, use the Lottery Setup Wizard to do so.

Import button

This will import and translate those settings into a Lotto Sorcerer v9 Playslip file.

Import v8 Playslip Settings

Figure 105.

Overview

This lets you import your Lotto Sorcerer v8 playslip file into Lotto Sorcerer v9.

How to Invoke

Use the menu item "Utilities > Playslips > Import Playslip Settings > Import v8 Playslip Settings".

Basic Procedure

1. Select the Lotto Sorcerer v8 Playslip file by clicking the "Select" button
2. Select the Lotto Sorcerer v9 lottery for which this playslip is intended by using the "Select v9 Lottery" dropdown
3. Click the "Import" button

Window Controls

Select button

Click this button to select the Lotto Sorcerer v8 Playslip file you want to import.

Select v9 Lottery dropdown

Choose the v9 lottery for which this playslip is intended. This lottery must already have been setup. If it has not been setup, use the Lottery Setup Wizard to do so.

Import button

This will import and translate those settings into a Lotto Sorcerer v9 Playslip file.

Import v9 Playslip Settings

Overview

This lets you import an external Lotto Sorcerer v9 playslip file into Lotto Sorcerer v9. For example, if a friend or a fellow member of a local lottery club sets up a playslip file for you (and emails it to you), you can import that playslip file using this utility.

How to Invoke

Use the menu item "Utilities > Playslips > Import Playslip Settings > Import v9 Playslip Settings".

Basic Procedure

- Select the Lotto Sorcerer v9 Playslip file by choosing it in the standard file selector that appears.

Export Playslip Setting

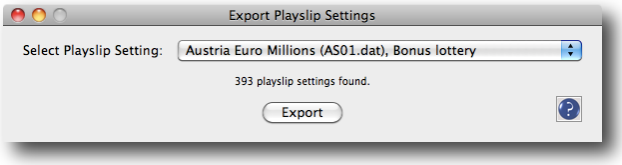


Figure 106.

Overview

This lets you export any Lotto Sorcerer playslip.

How to Invoke

Use the menu item "Utilities > Playslips > Export Playslip Setting".

Basic Procedure

1. Select the playslip file you want to export
2. Click the "Export" button

Window Controls

Select Playslip dropdown menu

Choose the playslip you want to export.

Export button

When you click this button, a standard file selector will open; choose the location where you want to save the playslip to.

Test Printer

Overview

This function will print out a box on a piece of paper, as close to the edge of the paper as your printer and printer driver will let it. *This test is important, because if the playslip you want to print to has boxes closer to the edge than your printer can print,* ***then your printer will never be able to print to that playslip, no matter how much you adjust the settings.***

Playslip Troubleshooter

This is an online utility which may help to quickly track down (and fix) issues you may be having in setting up a playslip.

How to Invoke

To go to the Playslip Troubleshooter, use menu item "Utilities > Playslips > Playslip Troubleshooter".

Lotto Scribe

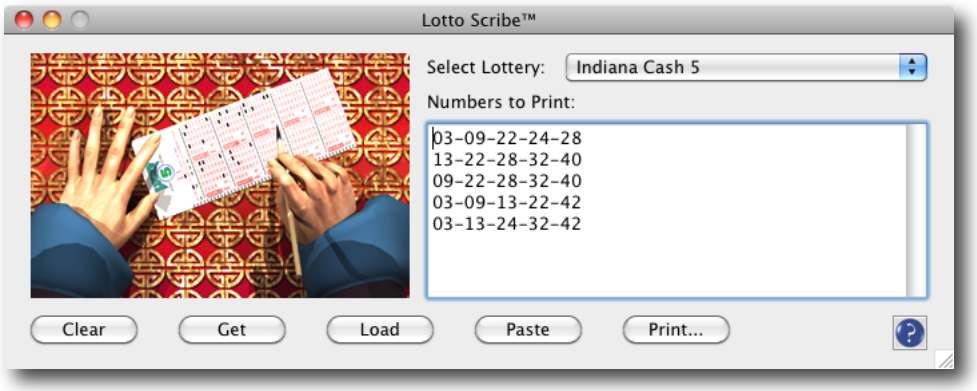


Figure 107.

Overview

Lotto Scribe lets you print your own numbers onto lottery playslips

How to Invoke

Use the menu item "Utilities > Playslips > Lotto Scribe".

Basic Procedure

1. Select the lottery that you want to work within the "Select Lottery" dropdown menu
2. Enter the numbers you want to play in the text box in the center of the window
3. Click the "Print" button

Window Controls

Select Lottery dropdown menu

Use this dropdown to select a lottery.

Clear button

This clears the text window.

Get button

This retrieves the suggestions from the Main Window > Suggestions tab of Lotto Sorcerer. Please note that this button is enabled only when 1) the lottery in the Main Window of Lotto Sorcerer matches the lottery you have selected in Lotto Scribe; and 2) There are suggestions in the Suggestions tab of the Main Window of Lotto Sorcerer.

Load button

This opens up a file selector, where you select a text file containing the numbers you want to play.

Paste button

This copies text from the System Clipboard into the text box. Note that this will work only if there is text data in the Clipboard.

Print button

This prints the numbers in the text box to your playslip(s).

Notes

When entering data into the text box, Lotto Scribe expects the numbers to be separated by a non-numeric character (the dash ["-"] is recommended).
Bonus number(s) should be the final number(s) in the series.
The playslip(s) you want to print to must have been previously setup in the Playslip Setup Wizard.

Registration

Registration Overview

Lotto Sorcerer is distributed as trial software, which, basically means you get to try the software, for free, for 12 uses. After the 12 uses, the software becomes disabled. If you like it and want to continue to use it, you must register it.

To register Lotto Sorcerer you will need to purchase the license for Lotto Sorcerer from Satori Publishing. Purchasing details and current pricing can be found on Lotto Sorcerer's website (www.satoripublishing.com/LS/).

When you have paid for a license, Satori Publishing will send you a name and registration code. Entering these items into the appropriate places in the Enter Registration Code window, then click the "OK" button.

If you entered all of the information correctly, the evaluation version will be transformed into the full, registered version.

Enter Registration Code

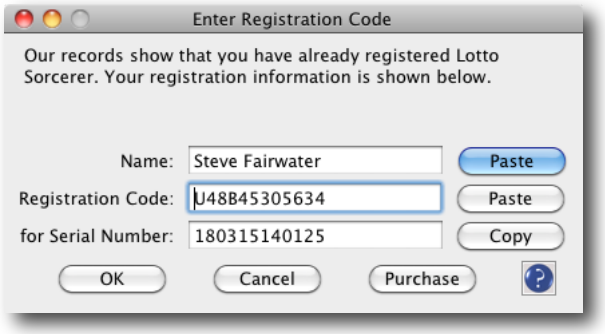


Figure 108.

Overview

This is the window you use to enter your name and registration code (provided to you by Satori Publishing).

If you entered all of the information correctly, the evaluation version will be transformed into the full, registered version.

How to Invoke

Use the menu item "File > Registration > Enter Registration Code".

Basic Procedure

1. Enter your name in the "Name" box
2. Enter your registration code in the "Registration Code" box
3. Click the "OK" button

Window Controls

OK button

Click this after entering your registration information. If you entered everything correctly, your evaluation version of Lotto Sorcerer will be transformed into the full, registered version.

Cancel button

Click this button to close the window.

Purchase Codes button

Clicking this button will take you to the Lotto Sorcerer website (if you have an active Internet connection).

Note

The serial number of your copy of Lotto Sorcerer is fixed, and cannot be changed manually. Reinstalling your operating system can cause the serial number to change. In this case, if your copy of Lotto Sorcerer is registered, you will need to have your old registration code retired and a replacement registration code generated. This can be accomplished by visiting this webpage:

http://www.satoripublishing.com/LS/v9_regcode_replace.php

Registration Troubleshooting

If, after entering your codes, Lotto Sorcerer tells you your codes are invalid, try these steps:

1. Are you entering codes for version 9? Codes for any other version (versions 8 and under) will not work with Lotto Sorcerer version 8.

2. Are you entering the codes for your installation's serial number? Your serial number is unique, and the registration codes are based on this serial number.

3. Are you entering codes for your platform? Codes for the Windows version will not work with the Mac OS X version (and vice versa).

4. Are you entering your name EXACTLY as we sent it to you? If we sent your name as "John R. Doe", do not enter "John R Doe", "JOHN R. DOE", "J. R. Doe", "john r. doe", "John Doe" or any other variation.

5. Are you entering the registration code EXACTLY as we sent it to you? The registration code always starts with any capital letter; the remainder is either numbers from "0" to "9" or capital letters from "A" to "F".

6. Only standard Roman characters are used. Non-Roman characters are mapped to the nearest Roman character. For example, "Müller Brøderbund" will be mapped to "Muller Broderbund".

If You Have Not Received Your Registration Codes...

We send out the registration codes within 24 hours of the receipt of payment... although most codes are sent out within a couple of minutes. We have nothing to gain by withholding your registration codes, since a large part of our business is from satisfied customers who continually purchase upgrades.

If you do not receive your codes after 24 hours, follow these guidelines:

1: Check your spam folder

The word "Lotto" in our email seems to trigger a lot of false positives with spam filters (and especially with Google mail and Yahoo mail). So check your spam, or junk mail folder. It is always a smart practice to add "sales@satoripublishing.com" on your white list of trusted email senders.

2: Verify that your PayPal address is your current email address

This is actually very common: people have registered with PayPal with an older email address. We send the codes to your PayPal address. Tip: PayPal will send you an email receipt within seconds of your purchase. If you have not received this email receipt from PayPal, this is probably the cause of the problem (because PayPal is also using your old email address).[4]

3: Verify that your email box is not full

This too, is common. If your email box is full, incoming email will not be delivered.

4: Contact us

Please contact us at support@satoripublishing.com. Include your phone number in the email! If you are having trouble receiving our emails, it is pointless sending you additional emails. We will contact you by phone.

4 This, of course, applies only if you purchased by PayPal. If you purchased by an alternative method, please ensure your email address is correct.

Appendices

Appendix A: LS Script Introduction

LS Script (short for "Lotto Sorcerer Script") is a powerful tool that lets you add your own functionality to Lotto Sorcerer.

One reason it was developed was because of so many requests for added features within Lotto Sorcerer. Many of these requests, although useful to the particular user requesting the feature, are esoteric. For example, there are many requests for adding additional filters to the Projection Parameter tab in the Main Window. But Lotto Sorcerer is running out of "real estate" space. Where do we put additional checkboxes to accommodate additional filters? We could make the Main Window larger, but that would make Lotto Sorcerer unusable to users with smaller monitors. Having an open-ended scripting system, like LS Script, solves this. Users can now easily create their own scripts to add these features themselves.

"What can I do with LS Script?"

You can do anything that is allowed within the parameters of the framework. Although LS Script is basically "sandboxed", there are portals that allow the script to communicate with the "outside world" (i.e., outside of Lotto Sorcerer's *Script Laboratory.*

Although LS Script was written to work with Lotto Sorcerer, you can actually use it as an independent programming environment, using it for applications that have nothing to do with Lotto Sorcerer or even lotteries. For example, you could download Qbasic source code from an astronomy website, and, with some minor modifications, adapt it to work under LS Script (provided, of course, that the program works under the framework's parameters). There is a plethora of websites that contain BASIC source code from different dialects of BASIC (e.g., Visual Basic, QuickBASIC, Amiga BASIC, and over 100 other dialects) that can be modified to work with LS Script.

"What are some real world examples?"

Here are four examples. The scripts which fulfill these three requests are provided with this installation of Lotto Sorcerer (under the "Scripts" folder, located in the "Lotto Sorcerer v9 Files" folder, which is located in your Document folder). Furthermore, the first two scripts are dissected in the Tutorial section of this document (Appendix B).

1. One user requested a feature where he could use the Fibonacci number sequence as an Acceptance Filter. He wanted to be able to select the number of Fibonacci numbers required in a suggestion.
2. Still another user wanted to be able to extract the drawings for a specific day from a lottery that draws numbers on multiple days.
3. Another user wanted to be able to generate all combinations of a set of numbers.
4. Another example is being able to extract the numbers drawn directly from a website, and pushing that data directly into the database.

"How difficult is it to learn LS Script?"

LS Script is a strongly typed subset of BASIC (Beginner's All-purpose Symbolic Instruction Code). As the name implies, BASIC is for beginners, and is considered the easiest programming language to learn. LS Script is actually a subset of REALbasic (now called XOJO). Anyone familiar with REALBasic, Microsoft's Visual Basic (including Visual Basic for Applications [part of the Microsoft Office suite]), QuickBASIC or QBasic should be able to pick up on LS Script quite easily.

BASIC is considered to be among the easiest of all computer languages to learn. Many professional programmers started with BASIC, which was included on the Commodore 64, Amiga and MS-DOS personal computers.

There are many resources available on those languages, both online as well as in books. Online, you will find a wealth of tutorials and free source code.

Whether you are new to programming, or if you are an experienced programmer, we urge you to go through the tutorial (in Appendix B). There are always nuances between the different dialects of BASIC, and going through the tutorial will familiarize you with the subtleties of LS Script.

"What can I *not* do with LS Script?"

1. You cannot create new forms (windows) within Lotto Sorcerer, nor can you modify existing forms.
2. You cannot access the hard drive except by using the built-in portals "GetFile" and "SaveFile", which lets you retrieve and save text files.

Important Points

- LS Script does not use line numbers (as in early forms of BASIC).
- LS Script **requires that all variables be declared before use.**
- Variable names are not case sensitive.
- Variable names must start with a letter.
- Arrays are zero-based.
- Functions and subroutines are placed after the main code.
- All variable are local to the routine in which they reside; there are no global variables.
- Any source code that your create is your property, and you can do with as you see fit. You can give away your source code or even sell it to others.

Tips and Tricks

- Although indentation of the source code is not required, it is strongly recommended to improve readability. You can use either the space or tab character for indentation.
- Commenting of source code is also strongly recommended. You can insert a comment by the use of a single apostrophe or two slash marks or the REM keyword.

Appendix B: LS Script Tutorial

This section contains a few tutorials you can go through to learn and familiarize yourself with LS Script. You do not need to type in the code listings; they are provided in the Lotto Sorcerer installation: use the "Load" button in the Scripting Laboratory (Lotto Sorcerer menu item "Utilities > Scripting Laboratory").

It is strongly recommend that you do these tutorials in order, since each subsequent one assumes increasing familiarity with this scripting technology.

Example 1: Creating a Custom Filter

A user once requested an assertion filter be added to Lotto Sorcerer, which would pass suggestions that contain at least one Fibonacci number. A Fibonacci number is a pattern of numbers where each number is the sum of the two preceding numbers, starting with 0 and 1.

We declined to add the filter, since Lotto Sorcerer has run out of "real estate" space to add this feature. But this is an excellent opportunity for LS Script.

What this script does is pull the suggestions that were generated by Lotto Sorcerer (in the "Projection Results" tab of the Main Window; checks each suggestion to see if it contains at least one Fibonacci number (and, if so, puts it back in the suggestions list).

First, we define the variable the script will use. Normally, you will define variables as you write the code, but the variables must always be declared first. Here are the first four lines of the script:

```
Dim EOL, Suggestions, Fibonacci, l, Candidate, o as String
Dim NumberOfSuggestions, NumbersPlayed, TestNumber as Integer
Dim i, j, k, OKSuggestions as Integer
Dim OK as Boolean
```

Since we do not know what type of computer the user will be using, we define the system's End-of-line string by calling the `GetEOL()` function:

```
EOL = GetEOL()
```

Next, we must define the Fibonacci sequence, or, at least, the first unique numbers in the Fibonacci sequence, up to 99. Why a maximum number of "99"? Because that its the highest number Lotto Sorcerer handles in lotteries. We define the number sequences as a comma-delimited string:

```
Fibonacci = "0,1,2,3,5,8,13,21,34,55,89"
```

Next, we pull in the suggestions from the "Projections Results" tab in the Main Window with the `GetSuggestions()` function:

```
Suggestions = GetSuggestions()
```

Now, we need to find out how many suggestions there are. We know that Suggestions uses the system's End-of-line terminator as a separator, so we can use the `CountFields` function to determine the number of suggestions:

```
NumberOfSuggestions = CountFields(Suggestions, EOL) - 1
```

Why did we subtract "1" from the number returned by `CountFields`? Because the `GetSuggestions()` function adds an end-of-line string at the end of the suggestions.

We need to make sure there are suggestions to analyze (in case somebody runs this script without having generated any suggestions first. So we add these lines:

```
If NumberOfSuggestions > 0 then
      ...
Else
      Msg "There are no suggestions to analyze!"
End If
```

Note that the last three lines are at the end of the script. The ellipsis ("...") represents the remainder of the program.

Assuming that there are suggestions to analyze, we need to calculate the numbers played:

```
      NumbersPlayed = CountFields(StripExtra(nThField(Suggestions, EOL, 1), "-
"),"-")
```

Here we see nested functions. The innermost function, `nThField(Suggestions, EOL, 1)`, retrieves the first suggestion. The next function, `StripExtra(..., "-")` converts the suggestion to make sure it uses the dash ("-") as the delimiter. Why do we do this? Because the user could be using a different delimiter. Also, bonus balls are shown in the suggestions as "BB:", and we need a consistent delimiter between each number. These two functions leave us with the numbers in the suggestion, dash-delimited. The outermost function `CountFields(...,"-")` counts the numbers in the suggestion.

We now have all of the variables we need. Now we reach the heart of the script:

We start with a For...Next counter, going through the suggestions, one-by-one:

```
      for i = 1 to NumberOfSuggestions
```

We call each potential suggestion, "Candidate":

```
      Candidate = StripExtra(nThField(Suggestions, EOL, i), "-")
```

For each pass through the suggestions, we set the boolean variable, "OK", to "false", to use as a flag:

```
      OK = False
```

Now, set up another For...Next counter, and we go through each number in the suggestion, calling it "TestNumber":

```
            for j = 1 to NumbersPlayed
                  TestNumber = Val(nThField(Candidate, "-", j))
```

And for each "TestNumber", we set up another counter to go through each Fibonacci number:

```
                  for k = 1 to CountFields(Fibonacci, ",")
```

If the "TestNumber" matches at least one Fibonacci number, we set the OK flag to "True", increment the number of suggestions that have passed the test, and write the suggestion to the "o" output string variable. Also, we append a message to a log file:

```
                  if TestNumber = Val(nThField(Fibonacci, ",", k)) then
                        //One number matches, suggestion is acceptable
                        OK = True
                        OKSuggestions = OKSuggestions + 1
                        o = o + Candidate + EOL
                        l = l + Candidate + " is OK." + EOL
                  end if
```

Note that the line which starts with a double slash ("//") is a comment line. The script ignores the line when it is run. Also, not that we use the `Val` function to convert the numbers, which are stored as strings, to a numeric variable when we compare the values. Why do we do this? Because the string values "5" and "05" are not equal, although they are equivalent numerically.

We then close the "j" and "k" for...next loops with the "next" keyword:

```
                next
            next
```

Next, we make use of the "OK" flag we set earlier. If "OK" is false, that is, if the particular suggestion does not have any Fibonacci number in it, we log that as well:

```
            If OK = False then
                  //No Fibonacci numbers in suggestion; log it
                  l = l + Candidate + " fails." + EOL
            End if
```

We now close the "i" for...next loop.

```
      Next
```

If any of the suggestions passed, we place them back in the Projection Results, using the `PostSuggestions` function:

```
      if OKSuggestions > 0 then
            //Put approved suggestions back
            PostSuggestions(o)
      else
            Msg "No suggestions contain any Fibonacci numbers."
      end if
```

Next, we write the log to the Script Laboratory's "Output" tab, using the `Print` statement:

```
      Print l
```

What is the "l"? It is the string variable we kept the log output in.

Finally we call the ShowTab() function to make the Output tab visible:

```
      ShowTab(2)
```

Example 1, Full Listing

```
Dim EOL, Suggestions, Fibonacci, l, Candidate, o as String
Dim NumberOfSuggestions, NumbersPlayed, TestNumber as Integer
Dim i, j, k, OKSuggestions as Integer
Dim OK as Boolean

EOL = GetEOL()
Fibonacci = "0,1,2,3,5,8,13,21,34,55,89"
Suggestions = GetSuggestions()
NumberOfSuggestions = CountFields(Suggestions, EOL) - 1
If NumberOfSuggestions > 0 then
      NumbersPlayed = CountFields(StripExtra(nThField(Suggestions, EOL, 1), "-
"),"-")
```

```
      for i = 1 to NumberOfSuggestions
            Candidate = StripExtra(nThField(Suggestions, EOL, i), "-")
            OK = False
            for j = 1 to NumbersPlayed
                  TestNumber = Val(nThField(Candidate, "-", j))
                  for k = 1 to CountFields(Fibonacci, ",")
                        if TestNumber = Val(nThField(Fibonacci, ",", k)) then
                              //One number matches, suggestion is acceptable
                              OK = True
                              OKSuggestions = OKSuggestions + 1
                              o = o + Candidate + EOL
                              l = l + Candidate + " is OK." + EOL
                        end if
                  next
            next
            If OK = False then
                  //No Fibonacci numbers in suggestion; log it
                  l = l + Candidate + " fails." + EOL
            End if
      Next

      if OKSuggestions > 0 then
            //Put approved suggestions back
            PostSuggestions(o)
      else
            Msg "No suggestions contain any Fibonacci numbers."
      end if

Else
      Msg "There are no suggestions to analyze!"
End If

Print l
ShowTab(2)
```

Example 2: Lottery Extractor

Another customer wanted a feature where he would be able to extract the drawing data for a certain day of week. For example, if a lottery drew two days a week, like Wednesday and Saturday, he wanted to be able to extract the drawings from Saturdays and create a custom lottery from that.

Here is a script that will accomplish that.

First comes the variables that will be used:

```
Dim EOL, TableName, wd, WeekDays(7) as String
Dim SQLStatement, DrawingData, DrawingRecord, Output as String
Dim i, Day, NumberOfRecords, Count as Integer
```

Then we set up some constants:

```
EOL = GetEOL //Get system end-of-line terminator
wd = "Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday"
```

Next, we build a string array for the names of the days of the week:

```
//Build WeekDays() array
For i = 1 to 7
      WeekDays(i) = nThField(wd, ",", i)
Next
```

We then have the user choose the lottery he wants to work with. This function pops up a small window, allowing the user to choose a lottery (from the list of built-in lotteries) to work from:

```
//Have user choose lottery
TableName = GetTable(3)
```

Note that we are not allowing any virtual lotteries to be chosen, because that is more involved, and will be covered in the next tutorial.

Now we need to user to choose the day of the week that he wants to extract data for. This function pops up another window, allowing him to choose the day of the week:

```
//Have user select day
Day = GetDropDown("Weekday Selector", "Select Day", "Select", wd)
```

The variable `Day` now contains a number between 1 and 7.

Next, we retrieve all data for the selected lottery from the database by using a SQL statement:

```
//Get data from database
DrawingData = SQLSelect("select * from " + Tablename + " order by DRAWDATE")
```

The variable `DrawingData` now contains the data, with each record separated by an end-of-line character.

Next is the heart of the program. We go through `DrawingData`, one record at a time. If a record is drawn on the day the user wants, it is added to a string variable called `Output` and the `Count` variable is incremented:

```
//Calculate Number of records
NumberOfRecords = CountFields(DrawingData, EOL) - 1

//Go through the records
for i = 1 to NumberOfRecords
      DrawingRecord = nThField(DrawingData, EOL, i)
      if CheckDay(DrawingRecord, Day) = True then
            Output = Output + DrawingRecord + EOL
            Count = Count + 1
      end if
next
```

Take notice of the function `CheckDay`. This function is not listed in the LS Script Programmer's Reference Guide. Why not? Because it is a custom function. LS Script allows you to create your own functions, and this is one of them. Functions always come at the end of the script, and that will be covered later.

Finally, we output the results. If there are no drawings on the day the user wanted, a message box pops up, advising him of this. Otherwise, a standard file selector appears, allowing the user to select the filename and location of the data:

```
If Count = 0 then
      Msg "There are no drawings drawn on " + Weekdays(Day) + "."
```

```
else
      //Output results
      SaveFile(Output)
      Msg "File has been saved."
end if
```

That is the end of the main part of the script. What follows is the custom function, `CheckDay`. This function expects the database record be passed along with the desired day of the week (1 through 7). This function returns with either the Boolean value TRUE or FALSE. "TRUE" meaning, "yes, the drawing is drawn on the desired day" and "FALSE" meaning "no, the drawing is not drawn on the desired date."

```
Function CheckDay(DataLine as String, DesiredDay as Integer) as Boolean
      //This function checks to see if the drawing occurs on the desired day of
week
      Dim DrawingDate as String
      DrawingDate = nThField(DataLine, Chr(9), 1)
      If GetDayOfWeek(DrawingDate) = DesiredDay then
            Return True
      Else
            Return False
      End If
End Function
```

Example 2, Full Listing

```
Dim EOL, TableName, wd, WeekDays(7) as String
Dim SQLStatement, DrawingData, DrawingRecord, Output as String
Dim i, Day, NumberOfRecords, Count as Integer

EOL = GetEOL //Get system end-of-line terminator
wd = "Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday"

//Build WeekDays() array
For i = 1 to 7
      WeekDays(i) = nThField(wd, ",", i)
Next

//Have user choose lottery
TableName = GetTable(3)

//Have user select day
Day = GetDropDown("Weekday Selector", "Select Day", "Select", wd)

//Get data from database
DrawingData = SQLSelect("select * from " + Tablename + " order by DRAWDATE")

//Calculate Number of records
NumberOfRecords = CountFields(DrawingData, EOL) - 1

//Go through the records
for i = 1 to NumberOfRecords
      DrawingRecord = nThField(DrawingData, EOL, i)
      if CheckDay(DrawingRecord, Day) = True then
            Output = Output + DrawingRecord + EOL
            Count = Count + 1
```

```
      end if
next

If Count = 0 then
      Msg "There are no drawings drawn on " + Weekdays(Day) + "."
else
      //Output results
      SaveFile(Output)
      Msg "File has been saved."
end if

Function CheckDay(DataLine as String, DesiredDay as Integer) as Boolean
      //This function checks to see if the drawing occurs on the desired day of
week
      Dim DrawingDate as String
      DrawingDate = nThField(DataLine, Chr(9), 1)
      If GetDayOfWeek(DrawingDate) = DesiredDay then
            Return True
      Else
            Return False
      End If
End Function
```

Example 3: Virtual Lotteries

Because virtual lotteries do not exist as discrete tables within Lotto Sorcerer, they must be built dynamically. This example shows how to do just that. It also shows how to output the Scripting Laboratory's grid.

In this tutorial, the user will select a virtual lottery, and the sorted contents of that virtual lottery will be output to the grid. Of course, it is expected that a virtual lottery is already setup.

The routine for processing a virtual is presented, at the end of the script, as a portable function. You can add this function to any of your scripts.

At the beginning of this script, we set up the dimension variables and acquire the system's end-of-line character:

```
Dim EOL, Tablename, VirtualLotteryData, DataLine as String
Dim NumberOfRecords, NumberOfColumns as Integer
Dim i, j as Integer

EOL = GetEOL //Get system end-of-line terminator
```

Next, we prompt the user for the virtual lottery to use:

```
Tablename = GetTable(4)
```

We make sure the user did not cancel the virtual lottery selection process. We do this by making sure the `Tablename` variable is not empty:

```
If Tablename <> "" then
```

Next, we call the `GetVirtualData()` function (located at the end of the script. Upon calling this, the variable `VirtualLotteryData` is now filled with the combined records of all members of the virtual lottery:

```
VirtualLotteryData = GetVirtualData(Tablename)
```

Because the lottery data is always presented as records separated by end-of-line characters, we can calculate the number of records by using the `CountFields` function:

```
NumberOfRecords = CountFields(VirtualLotteryData, EOL) - 1
```
We subtract one from the value, because lottery data is always terminated by an end-of-line character.

Next, we prepare the grid (on the "Grid" tab of the Scripting Laboratory). First, we clear it (in case it is already holding something):

GridClearAll

We set up the number of columns in the grid. We do this by counting the number of fields in the first record of the lottery data, and then we call the `GridSetColNumber()` function. We can calculate the number of fields because we know that lottery data is always presented with an ASCII 9 character between each field.

```
      NumberOfColumns = CountFields(nThField(VirtualLotteryData, EOL, 1),
Chr(9))
      GridSetColNumber(NumberOfColumns)
```

Next, we set the headings. We want the first column to be called "DRAWDATE", the second column to be called "DRAWTIME", and all subsequent columns to be the number:

```
      GridSetHeadings(0, "DRAWDATE")
      GridSetHeadings(1, "DRAWTIME")
      For i = 2 to NumberOfColumns - 1
            GridSetHeadings(i, Str(i - 1))
      Next
```

Next, we set the widths of the first two columns. All subsequent columns' widths will be divided equally amongst the remaining space:

```
GridColWidths("110,80")
```

The last step in defining the grid is making everything center-aligned:

```
      For i = 0 to NumberOfColumns - 1
            GridColAlignment(i, 2) //Center alignment
      Next
```

Next, we fill the grid with the data from the variable `VirtualLotteryData`. We do this by going through the data, line by line. The `GridAddRow()` function adds a new row to the grid and populates the first column. To populate the remaining columns in that row, we go through the record's fields, one by one, using the `GridPostCell()` function:

```
      For i = 1 to NumberOfRecords
            DataLine = nThField(VirtualLotteryData, EOL, i)
            GridAddRow(nThField(DataLine, Chr(9), 1))
            for j = 2 to NumberOfColumns
                  GridPostCell(i - 1, j - 1, nthField(DataLine, Chr(9), j))
            next
      Next
```

Because a virtual lottery can be huge, we ring the computer's "bell" when we are finished:

```
Ring
```

And force the third tab, "Grid", of the Scripting Laboratory forward into view:

ShowTab(3)
What follows is the function which combines and sorts the data of the virtual lottery's member lotteries. It does this by finding out which lotteries are the members to this virtual lottery:

```
      SQLStatement = "select LOTTERYTABLE from LOTTERIES where VIRTUALMEMBER =
'" + Table + "'"
      SQLResults = SQLSelect(SQLStatement)
```

And then building up a SQL "union" statement...

```
      NumberOfRecords = CountFields(SQLResults, EOL) - 1
      SQLStatement = ""
      For i = 1 to NumberOfRecords
            SQLStatement = SQLStatement + "select * from " + nTh-
Field(SQLResults, EOL, i)
            if i < NumberOfRecords then
                  SQLStatement = SQLStatement + " union "
            end if
      Next
```

...and then appending a SQL sort statement:

```
SQLStatement = SQLStatement + " order by DRAWDATE, DRAWTIME"
```

And then executing that SQLStatement...

```
SQLResults = SQLSelect(SQLStatement)
```

...and then passing that data back to the main script:

```
      Return SQLResults
```

Example 3, Full Listing

```
//This script is an example of working with a virtual lottery.

Dim EOL, Tablename, VirtualLotteryData, DataLine as String
Dim NumberOfRecords, NumberOfColumns as Integer
Dim i, j as Integer

EOL = GetEOL //Get system end-of-line terminator

//Have user choose Virtual Lottery
Tablename = GetTable(4)

If Tablename <> "" then

      VirtualLotteryData = GetVirtualData(Tablename)
      NumberOfRecords = CountFields(VirtualLotteryData, EOL) - 1

      //Setup Grid
      GridClearAll
```

```
      NumberOfColumns = CountFields(nThField(VirtualLotteryData, EOL, 1),
Chr(9))
      GridSetColNumber(NumberOfColumns)
      GridSetHeadings(0, "DRAWDATE")
      GridSetHeadings(1, "DRAWTIME")
      For i = 2 to NumberOfColumns - 1
            GridSetHeadings(i, Str(i - 1))
      Next
      GridColWidths("110,80")
      For i = 0 to NumberOfColumns - 1
            GridColAlignment(i, 2) //Center alignment
      Next

      //FILL GRID
      For i = 1 to NumberOfRecords
            DataLine = nThField(VirtualLotteryData, EOL, i)
            GridAddRow(nThField(DataLine, Chr(9), 1))
            for j = 2 to NumberOfColumns
                  GridPostCell(i - 1, j - 1, nthField(DataLine, Chr(9), j))
            next
      Next

      Ring
      ShowTab(3)
End If

Function GetVirtualData(Table as String) as String
      //This function extracts all data for a virtual lottery. Enter with
      //the virtual lottery's table name; exit with the data from the member
      //lotteries, sorted.
      //
      //You can copy and use this function for your own scripts.
      //

      Dim SQLStatement, SQLResults, EOL as String
      Dim i, NumberOfRecords as Integer

      EOL = GetEOL()

      //BUILD SQLSTATEMENT TO ACCESS MEMBER LOTTERIES
      SQLStatement = "select LOTTERYTABLE from LOTTERIES where VIRTUALMEMBER =
'" + Table + "'"
      SQLResults = SQLSelect(SQLStatement)
      NumberOfRecords = CountFields(SQLResults, EOL) - 1
      SQLStatement = ""
      For i = 1 to NumberOfRecords
            SQLStatement = SQLStatement + "select * from " + nTh-
Field(SQLResults, EOL, i)
            if i < NumberOfRecords then
                  SQLStatement = SQLStatement + " union "
            end if
      Next
      SQLStatement = SQLStatement + " order by DRAWDATE, DRAWTIME"

      //GET DATA FROM VIRTUAL LOTTERY
      SQLResults = SQLSelect(SQLStatement)
      Return SQLResults
```

```
End Function
```

Example 4: Graphics

The graphics tab lets you create your own column, pie, line, area, xy (scatter), bar charts and much more. You can create lines, ovals, circles and rectangles in over 16 million colors.

This example creates a 3-sector distribution pie chart of a lottery drawings.

First, we define the variable we will be using...

```
Dim TableName, GameParameters, SQLStatement, Dates, Result, EOL as String
Dim DateList, SelectedDate, LotteryName, DrawnNumbers as String
Dim MinPoolNumber, MaxPoolNumber, NumberOfRecords as Integer
Dim i, Selection, NumbersDrawn(0), Numbers as Integer
Dim Range, LowPoint, HighPoint, Divisor as Integer
Dim High, Middle, Low as Integer
Dim H, W, x, y, TH as Integer
Dim PCenterX, PCenterY, PDiameter, PRadius, Increment as Integer
Dim Angle as Double
```

... and define the EOL (end-of-line) variable:

```
EOL = GetEOL //Get system End-of-line terminator
```

Next, we have the user select the lottery he want to use, and then retrieve game parameters we will be needing:

```
TableName = GetTable(1)
GameParameters = GetGameParams(TableName)
MinPoolNumber = Val(nthField(GameParameters, ",", 3))
MaxPoolNumber = Val(nthField(GameParameters, ",", 4))
LotteryName = nthField(GameParameters, ",", 9)
```

We want to have a dropdown menu for the user to select the date of the drawing. We will lookup the last 100 drawings for this lottery...

```
//Retrieve last 100 drawing dates from lottery
SQLStatement = "select DRAWDATE from " + TableName + " order by DRAWDATE desc
limit 100"
Dates = SQLSelect(SQLStatement)
NumberOfRecords = CountFields(Dates, EOL) - 1
```

...and use that to populate the dropdown menu, and have the user select the specific date he wants to use:

```
//Get desired date from user
For i = 1 to NumberOfRecords
      DateList = DateList + nThField(Dates, EOL, i)
      If i < NumberOfRecords then
            DateList = DateList + ","
      End if
Next
Selection = GetDropDown("Select Date", "Select Date", "Select", DateList)
SelectedDate = nThField(DateList, ",", Selection)
```

Next, we get the numbers drawn for that date:

```
//Get drawing data for that drawing
SQLStatement = "select * from " + TableName + " where DRAWDATE = '" + Selected-
Date + "'"
Result = SQLSelect(SQLStatement)
Numbers = CountFields(Result, Chr(9)) - 2
for i = 3 to CountFields(Result, Chr(9))
      NumbersDrawn.Append val(nthField(Result, Chr(9), i))
      DrawnNumbers = DrawnNumbers + nThField(Result, Chr(9), i)
      if i < CountFields(Result, Chr(9)) then
            DrawnNumbers = DrawnNumbers + "-"
      end if
Next
```

At this point, we have everything we need to create the pie chart. This pie chart will be a 3-sectored distribution chart, showing how many numbers are "high" numbers, how many numbers are "middle" numbers, and how many numbers are "low" numbers. For example, if the number pool is from 1 to 39, then then numbers 1 through 13 will be "low" numbers, numbers 26 to 39 will be considered "high" numbers, and everything else, in the middle will be the "middle" numbers.

Not every lottery will be able to broken into thirds so evenly, so this algorithm will do the closest, with any skewing being sent to the middle numbers:

```
//Determine sector boundaries
Range = MaxPoolNumber - MinPoolNumber + 1
Divisor = Round(Range / 3)
LowPoint = Divisor + MinPoolNumber - 1
HighPoint = MaxPoolNumber - Divisor

//Count sector memberships
For i = 1 to Numbers
      Select Case NumbersDrawn(i)
            Case Is <= LowPoint
                  Low = Low + 1
            Case Is >= HighPoint
                  High = High + 1
            Case Else
                  Middle = Middle + 1
      End Select
Next
```

At this point, the variable `Low` contains the number of low numbers, the variable `Middle` contains the number of middle numbers, and the variable `High` contains the number of high numbers.

Now, we can start making the graph.

Absolutely essential is to precede any graphics commands and functions with the `InitializeGraphics` command:

```
InitializeGraphics
```

Next, we need to find out the size of the canvas, because the user may have resized the Scripting Laboratory window. We store the canvas width and height in the `W` and `H` variables, respectively:

```
W = GetWidth()
H = GetHeight()
```

We setup the canvas with a background color and a border. All colors are defined as Red, Green and Blue (RGB) integer values, in the range of 0 to 255 each. 256 possible values of red, 256 values of green and 256 values of blue gives us a potential of 256 x 256 x 256 = 16,777,216 different colors.

First, we color the background ivory and draw a black border around the canvas:

```
'Set background to "ivory"
SetColor(255,255,240)
FillRect(0,0,W,H)
'Make black border
SetColor(0,0,0)
DrawRect(0,0,W,H)
```

Next, we print a header at the top of the canvas: the name of the lottery, the date, and the numbers drawn on that date:

```
SetFont("Arial")
SetTextSize(14)
TH = GetTextHeight
x = 15
y = 15
SetBold(True)
DrawString("Distribution (3 Sector) Pie Chart for: " + LotteryName, x, y, W-10,
True)
SetBold(False)
y = y + TH
'Print date and drawn numbers
DrawString(GetLongDate(SelectedDate) + ": " + DrawnNumbers, x, y, W-10, True)
y = y + TH * 2
```

Next, we draw a legend on the left of the canvas, showing the three colors of the three slices of the pie chart; the value of each slice; and a description of each slice:

```
'
'Draw legend for "High" box
'
SetColor(255,0,0) 'Make "High" box red
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("High (" + Str(HighPoint) + " to " + Str(MaxPoolNumber) + ")", 100,
y + 15 + (TH / 2) - 2, 0, False)
DrawString(Str(High), x + (75 / 2) - (GetStringWidth(Str(High)) / 2), y + 15 +
(TH / 2) - 2, 0, False)
y = y + 40
'
'Draw legend for "Middle" box
'
SetColor(255,165,0) 'Make "Middle" box orange
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("Middle (" + Str(LowPoint + 1) + " to " + Str(HighPoint - 1) + ")",
100, y + 15 + (TH / 2) - 2, 0, False)
DrawString(Str(Middle), x + (75 / 2) - (GetStringWidth(Str(Middle)) / 2), y +
15 + (TH / 2) - 2, 0, False)
y = y + 40
```

```
'
'Draw legend for "Low" box
'
SetColor(255,255,0) 'Make "Low" box yellow
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("Low (" + Str(MinPoolNumber) + " to " + Str(LowPoint) + ")", 100, y
+ 15 + (TH / 2) - 2, 0, False)
DrawString(Str(Low), x + (75 / 2) - (GetStringWidth(Str(Low)) / 2), y + 15 +
(TH / 2) - 2, 0, False)
```

Now comes the drawing of the pie chart itself. We start with the `SetAntiAlias(False)` command, which works well with the `Fill` command (which colors in each slice). Then we determine the size of the pie chart, and draw the outer circle:

```
SetAntiAlias(False)
PDiameter = Round(H * 0.75) 'Make the pie chart 3/4ths of height of canvas
PRadius = Round(PDiameter / 2)
PCenterX = W / 2 + 50
PCenterY = H / 2
DrawOval(PCenterX - PRadius, PCenterY - PRadius, PDiameter, PDiameter) 'Draw
outer circle
```

Finally, we draw the three pie slices. The first part of the `If... ElseIf...` construct tests to see if the numbers drawn are all high (or all middle, or all low); if so, there is no need to draw separate slices:

```
If High = Numbers then
      Fill(PCenterX, PCenterY, 255, 0, 0)
ElseIf Middle = Numbers then
      Fill(PCenterX, PCenterY, 255, 165, 0)
ElseIf Low = Numbers then
      Fill(PCenterX, PCenterY, 255, 255, 0)
```

If the slices do need to be constructed, we calculate the lines, drawn from the center of the circle to the edge, as well as target points for the `Fill` command (for coloring in the individual slices:

```
      //Draw Pie Slices
      DrawLine(PCenterX, PCenterY, PCenterX, PCenterY - PRadius) 'First line
      If High > 0 then
            Increment = High
            Angle = Increment * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * PRadius + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * PRadius + PCenterY
            DrawLine(PCenterX, PCenterY, x, y)
            Angle = (High / 2) * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 0, 0)
            SetColor(0, 0, 0)
      End If
      If Middle > 0 then
            Increment = Increment + Middle
            Angle = Increment * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * PRadius + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * PRadius + PCenterY
            DrawLine(PCenterX, PCenterY, x, y)
```

```
            Angle = (Increment - (Middle / 2)) * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 165, 0)
            DrawPixel(x, y, 255, 165, 0)
      End If
      If Low > 0 then
            Angle = 359
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 255, 0)
      End If
```

The last steps is using the `Draw` command to force drawing onto the canvas (for computers that do not have double-buffering) and the `ShowTab(4)` command to force the Graphics tab into view:

```
Draw
ShowTab(4)
```

Example 4, Full Listing

```
//This creates a pie chart of the distribution (3 sectors) of a single drawing.

Dim TableName, GameParameters, SQLStatement, Dates, Result, EOL as String
Dim DateList, SelectedDate, LotteryName, DrawnNumbers as String
Dim MinPoolNumber, MaxPoolNumber, NumberOfRecords as Integer
Dim i, Selection, NumbersDrawn(0), Numbers as Integer
Dim Range, LowPoint, HighPoint, Divisor as Integer
Dim High, Middle, Low as Integer
Dim H, W, x, y, TH as Integer
Dim PCenterX, PCenterY, PDiameter, PRadius, Increment as Integer
Dim Angle as Double

EOL = GetEOL //Get system End-of-line terminator

TableName = GetTable(1)
GameParameters = GetGameParams(TableName)
MinPoolNumber = Val(nthField(GameParameters, ",", 3))
MaxPoolNumber = Val(nthField(GameParameters, ",", 4))
LotteryName = nthField(GameParameters, ",", 9)

//Retrieve last 100 drawing dates from lottery
SQLStatement = "select DRAWDATE from " + TableName + " order by DRAWDATE desc
limit 100"
Dates = SQLSelect(SQLStatement)
NumberOfRecords = CountFields(Dates, EOL) - 1

//Get desired date from user
For i = 1 to NumberOfRecords
      DateList = DateList + nThField(Dates, EOL, i)
      If i < NumberOfRecords then
            DateList = DateList + ","
      End if
Next
Selection = GetDropDown("Select Date", "Select Date", "Select", DateList)
SelectedDate = nThField(DateList, ",", Selection)
```

```
//Get drawing data for that drawing
SQLStatement = "select * from " + TableName + " where DRAWDATE = '" + Selected-
Date + "'"
Result = SQLSelect(SQLStatement)
Numbers = CountFields(Result, Chr(9)) - 2
for i = 3 to CountFields(Result, Chr(9))
      NumbersDrawn.Append val(nthField(Result, Chr(9), i))
      DrawnNumbers = DrawnNumbers + nThField(Result, Chr(9), i)
      if i < CountFields(Result, Chr(9)) then
            DrawnNumbers = DrawnNumbers + "-"
      end if
next

//Determine sector boundaries
Range = MaxPoolNumber - MinPoolNumber + 1
Divisor = Round(Range / 3)
LowPoint = Divisor + MinPoolNumber - 1
HighPoint = MaxPoolNumber - Divisor

//Count sector memberships
For i = 1 to Numbers
      Select Case NumbersDrawn(i)
            Case Is <= LowPoint
                  Low = Low + 1
            Case Is >= HighPoint
                  High = High + 1
            Case Else
                  Middle = Middle + 1
      End Select
Next
//
//START MAKING THE GRAPH
//
InitializeGraphics
'Get current size of canvas
W = GetWidth()
H = GetHeight()
'Set background to "ivory"
SetColor(255,255,240)
FillRect(0,0,W,H)
'Make black border
SetColor(0,0,0)
DrawRect(0,0,W,H)
'Print lottery name
SetFont("Arial")
SetTextSize(14)
TH = GetTextHeight
x = 15
y = 15
SetBold(True)
DrawString("Distribution (3 Sector) Pie Chart for: " + LotteryName, x, y, W-10,
True)
SetBold(False)
y = y + TH
'Print date and drawn numbers
DrawString(GetLongDate(SelectedDate) + ": " + DrawnNumbers, x, y, W-10, True)
y = y + TH * 2
'
```

```
'Draw legend for "High" box
'
SetColor(255,0,0) 'Make "High" box red
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("High (" + Str(HighPoint) + " to " + Str(MaxPoolNumber) + ")", 100,
y + 15 + (TH / 2) - 2, 0, False)
DrawString(Str(High), x + (75 / 2) - (GetStringWidth(Str(High)) / 2), y + 15 +
(TH / 2) - 2, 0, False)
y = y + 40
'
'Draw legend for "Middle" box
'
SetColor(255,165,0) 'Make "Middle" box orange
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("Middle (" + Str(LowPoint + 1) + " to " + Str(HighPoint - 1) + ")",
100, y + 15 + (TH / 2) - 2, 0, False)
DrawString(Str(Middle), x + (75 / 2) - (GetStringWidth(Str(Middle)) / 2), y +
15 + (TH / 2) - 2, 0, False)
y = y + 40
'
'Draw legend for "Low" box
'
SetColor(255,255,0) 'Make "Low" box yellow
FillRect(x, y, 75, 30)
SetColor(0,0,0) 'Make box border black
DrawRect(x, y, 75, 30)
DrawString("Low (" + Str(MinPoolNumber) + " to " + Str(LowPoint) + ")", 100, y
+ 15 + (TH / 2) - 2, 0, False)
DrawString(Str(Low), x + (75 / 2) - (GetStringWidth(Str(Low)) / 2), y + 15 +
(TH / 2) - 2, 0, False)

//
//DRAW THE PIE CHART
//
SetAntiAlias(False)
PDiameter = Round(H * 0.75) 'Make the pie chart 3/4ths of height of canvas
PRadius = Round(PDiameter / 2)
PCenterX = W / 2 + 50
PCenterY = H / 2
DrawOval(PCenterX - PRadius, PCenterY - PRadius, PDiameter, PDiameter) 'Draw
outer circle

If High = Numbers then
      Fill(PCenterX, PCenterY, 255, 0, 0)
ElseIf Middle = Numbers then
      Fill(PCenterX, PCenterY, 255, 165, 0)
ElseIf Low = Numbers then
      Fill(PCenterX, PCenterY, 255, 255, 0)
Else
      //Draw Pie Slices
      DrawLine(PCenterX, PCenterY, PCenterX, PCenterY - PRadius) 'First line
      If High > 0 then
            Increment = High
            Angle = Increment * 360 / Numbers
```

```
            x = Sin(DegreesToRadians(Angle)) * PRadius + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * PRadius + PCenterY
            DrawLine(PCenterX, PCenterY, x, y)
            Angle = (High / 2) * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 0, 0)
            SetColor(0, 0, 0)
      End If
      If Middle > 0 then
            Increment = Increment + Middle
            Angle = Increment * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * PRadius + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * PRadius + PCenterY
            DrawLine(PCenterX, PCenterY, x, y)
            Angle = (Increment - (Middle / 2)) * 360 / Numbers
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 165, 0)
            DrawPixel(x, y, 255, 165, 0)
      End If
      If Low > 0 then
            Angle = 359
            x = Sin(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterX
            y = -Cos(DegreesToRadians(Angle)) * (PRadius * 0.9) + PCenterY
            Fill(x, y, 255, 255, 0)
      End If
End If

Draw
ShowTab(4)

Function DegreesToRadians(Degrees as Double) as Double
      Dim PI as Double
      PI = 3.14159265358979323846264338327795
      Return Degrees * PI / 180
End Function
```

Appendix C: LS Script Programmer's Reference Guide

DATATYPES

Integer

Integers hold negative and positive whole numbers (signed integers) or positive whole numbers only (signed integers). The default value for all integers datatypes is 0.

Int8 (-128 to 127)
Int16 (-32,768 to 32,767)
Int32 or Integer (-2,147,483,648 to 2,147,483,647)
Int64 (-2^{63} to 2^{63} -1)
UInt8 or Byte (0 to 255)
UInt16 (0 to 65,535)
UInt32 (0 to 4,294,967,295)
UInt64 (0 to 2^{64} - 1)

Single

A "single" datatype, also known as "single precision" or "float" is a number that can hold a real number. The maximum value of a single is:
$\pm 3.402823466385288598117041834845169 25^{38}$. The minimum value (towards zero) is:
$\pm 1.401298464324817070923729583289916 13^{-45}$.

The default value of a single is 0.0.

Double

A "double" datatype, also known as "double precision" is a number that can hold a real number. The maximum value of a double is:
$\pm 1.797693134862315708145274237317043 57^{308}$. The minimum value (towards zero) is:
$\pm 4.940656458412465441765687928682213 72^{-324}$.

The default value of a double is 0.0.

Boolean

Boolean datatypes can only take on the values True or False. The default value is False.

String

A "string" datatype is a series of numeric or alphabetic characters enclosed in quotes. Any kind of alphabetic or numeric information can be stored as a string. The maximum length of a string is limited only by available memory. The default value of a string is "".

CONTROL STRUCTURES

Function... End Function

Declares the name, parameters, returned value, and code that form the body of a function (method that returns a value).

Syntax

```
Function name (parameterList) as type
      local variable declarations
      statements
Return (variable)
```

Example

This example is a function that calculates the volume based on the length, width and depth passed.

```
Function CalcVolume(Length as Double, Width as Double, Depth as Dou-
ble
```

```
      Dim v as Double
      v = Length * Width * Depth
      Return v
End Function
```

Sub... End Sub

Declares the name, parameters, and code that form the body of a subroutine (method). The only difference between a Sub and a Function is that the Sub does not return a value.

Syntax

```
Sub name (parameterList)
      local variable declarations
      statements
End Sub
```

Example

This example rings the computer's bell and displays a message.

```
Sub Alert(Message as String)
      Ring
      Msg(Message)
End Sub
```

For... Next

Executes a series of statements a specified number of times.

Syntax

```
For counter = start To | DownTo End Step value
      statements
Next
```

Example 1

This example counts *up*.

```
Dim i as Integer
For i = 1 to 10
      Msg(Str(i))
Next
```

Example 2

This example counts *down*.

```
Dim i as Integer
For i = 10 DownTo 1 Step 1
      Msg(Str(i))
Next
```

Do... Loop

Repeatedly executes a series of statements while a specified condition is True.

Syntax

```
Do (Until condition)
      statements
Loop (Until condition)
```

"Condition" is any valid boolean expression.

Example
```
Dim x as Integer
x = 1
Do Until x > 100
      x = x * 3
Loop
Msg "x = " + Str(x)
```

Exit
The Exit statement causes control to exit a loop and jump to another line of code without the loop conditions being satisfied.

Syntax
```
Exit For | Do | While
      or
Exit For loopVariable
      or
Exit Sub | Function
```

"Condition" is any valid boolean expression.

Example
```
Dim i, j, myArray(255,255) as Integer
Dim Result, EOL as String
EOL = GetEOL()
//Fill myArray with random data //*****
For i = 0 to 255
      For j = 0 to 255
            myArray(i, j) =
For i= 0 to 255
      For j= 0 to 255
            If myArray(i, j) = 23 then
                  Exit For i
            End if
      Next
      Result = "i = " + Str(i) + EOL
      Result = Result + "j = " + Str(j) + EOL
      Result = Result + "myArray(i, j) = " + Str
Next
```

If... Then... End If
Conditionally executes a group of statements, depending on the value of a boolean expression.

Syntax
```
If condition Then
      statements
ElseIf condition Then
      statements
Else
      statements
End If
```

Example
```
Dim x, n as Integer
```

```
x = 50
If x < 10 then
      n = 1
Elseif n < 100 then
      n = 2
Else
      n = 3
End If
```

Select Case... End Select

Executes one of several groups of statements, depending on the value of an expression.

Syntax

```
Select Case testExpression
      Case expression
            statements
      Else
            statements
End Select
```

Example

```
Dim DayOfWeek as Integer
DayOfWeek = 3
Select Case DayOfWeek
Case 1
      Msg("Sunday")
Case 2
      Msg("Monday")
Case 3
      Msg("Tuesday")
Case 4
      Msg("Wednesday")
Case 5
      Msg("Thursday")
Case 6
      Msg("Friday")
Else
      Msg("Saturday")
End Select
```

Expressions can be a single item, a comma-separated list of times, a range or a mathematical expression. Here are some examples:

```
      Case 9 //single value
      Case 1,  2, 5, 16 //list of values
      Case 16 to 108 //a range of values
      Case < 64 //a mathematical expression
```

While... Wend

Repeatedly executes a series of statements while a specified condition is True.

Syntax

```
While condition
      statements
Wend
```

"Condition" is any valid boolean expression.

Example
```
Dim n as Integer
While n < 50
      n = n + 1
Wend
```

ARRAY FUNCTIONS

Dim
Creates a local variable or array with the name and size (in the case of an array) and data type specified.

Syntax
```
Dim VariableName as Datatype
      or
Dim ArrayName(ArraySize) as Datatype
```

Notes
- You can dimension multiple variables (or arrays) on a single line.
- For arrays with multiple dimensions separated the size or each dimension with a comma.

Example
```
Dim s as String
Dim a(20) as Integer
Dim OK as Boolean
Dim i, j, Names(20) as Integer //example of dimensioning several values
Dim States(50,10) as String //creates an array with 50 rows by 10 columns
```

ReDim
Resizes the passed array.

Syntax
```
ReDim ArrayName(NewArraySize)
```

UBound
Returns the index of the last element in an array. If an array has no elements, -1 is returned.

Syntax
```
Result = UBound(array, dimension)
```

Notes
The Ubound function is used to determine the last element of an array, but it can also be used to determine the size of an array. It may appear at first that the last element number and the size of the array are the same but in fact they are not. All arrays have a zero element. In some cases element zero is used and in other cases it is not. You will need to keep this in mind when using the Ubound function to determine the number of values you have in the array. For example, if the array is zero-based, then element zero is used to store a value and you will have to add one to the value returned by the Ubound function to get the number of values in the array.

For multi-dimensional arrays, Ubound returns the index of the last element of the dimension you specify, or, if you do not specify a dimension, it returns the value for the first dimension. The first dimension is numbered 1.

CONTROL FUNCTIONS

AppendBatchLine
This command adds a line at the end of the Batch list. If there is a batch file currently in progress, this line will be executed.

Syntax
AppendBatchLine(*Filename as String, Delay as Integer*)
where

Filename = name of LS Script file or Batch file
Delay = delay in seconds.

Notes:

- The LS Script file *must* be in the "Scripts" folder (of your Lotto Sorcerer v9 Files folder), and any batch file *must* be in the "Batch Files" folder.
- The filenames must have the extension of the file (".bas" for LS Scripts and ".bat" for batch files).
- Delay must be at least "5".

Example:
```
AppendBatchLine("Example1.bas",10)
```

Launch
Launches an executable file on your computer. Enter with the complete path and application name.

Syntax
Launch(*Path as String*)

For Mac OS X, the directory path is separated by colons (":"), and ends with a colon. For Windows, the backslash character ("\") is the directory separator as well as the trailing character.

Examples:
```
Launch("Macintosh HD:Applications:Utilities:Grapher.app:")
```
launches the built-in mathematical graphing application for Mac OS X.

```
Launch("C:\Program Files\Windows NT\Accessories\wordpad.exe\")
```
launches the built-in WordPad application in Windows XP.

Please note that both of these examples may not work on your particular computer if your pathway is different than those shown.

DATE FUNCTIONS

BuildSQLDate
Returns a string in YYYY-MM-DD format, where YYYY = the year, MM = the two-digit month and DD returns the two digit day.

Syntax
Result (as string) = BuildSQLDate(month as integer, day as integer, year as integer)

Example
```
Dim s as String
s = BuildSQLDate(10,2,2000) //returns “2000-10-02”
```

CheckDate

Checks for valid date.

Syntax

Result (as boolean) = CheckDate (SQLDate as string, NoFuture as boolean)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format
NoFuture = boolean flag (any future date is counted as an invalid date)

Note: even if the NoFuture flag is set to false, this function will return FALSE if the year is greater than 200 years in the past or the future.

Example

```
Dim dt as String
Dim OK as Boolean

dt = GetInput("Enter Date","",10,"","Enter Date")

OK = CheckDate(dt, True)
If OK = True then
   Msg "OK!"
else
   Msg "Bad date!"
end if
```

GetAbbreviatedDate

Returns the date in the user's abbreviated date format as a string based on the user's locale and formatting.

Syntax

Result (as string) = GetAbbreviatedDate (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as string
r = GetAbbreviatedDate("2005-08-10") //returns the date in the user's "Abbrevi-
ated date" format
```

GetDayOfWeek

Returns the day of the week (Sunday = 1, Saturday = 7) for the passed date. The passed date must be a string, in SQLDate format (YYYY-MM-DD).

Syntax

Result (as integer) = GetDayOfWeek (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as integer
r = GetDayOfWeek "2000-10-02") //returns 2 (for Monday)
```

GetDayOfYear

Returns the day of the year (e.g., January 1 = "1") for the passed date. The passed date must be a string, in SQLDate format (YYYY-MM-DD).

Syntax

Result (as integer) = GetDayOfYear (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as integer
r = GetDayOfYear "2000-10-02") //returns 276
```

GetLongDate

Reports the date in the user's long date format as a string based on the user's locale and formatting. The passed date must be a string, in SQLDate format (YYYY-MM-DD).

Syntax

Result (as string) = GetLongDate (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as String
r = GetLongDate "2000-10-02") //returns your computer's "Long Date" setting
```

GetShortDate

Reports the date in the user's short date format as a string based on the user's locale and formatting. The passed date must be a string, in SQLDate format (YYYY-MM-DD)

Syntax

Result (as integer) = GetShortDate (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as integer
r = GetShortDate "2000-10-02") //returns your computer's "Short Date" setting
```

GetWeekOfYear

Returns the week of the year for the passed date. The first week may be incomplete. If January 1 falls on a Saturday, then the next day is in week 2. The passed date must be a string, in SQLDate format (YYYY-MM-DD).

Syntax

Result (as integer) = GetDayOfYear (SQLDate as string)
where
SQLDate = the date in SQLDate (YYYY-MM-DD) format

Example

```
Dim r as integer
r = GetDayOfYear "2000-10-02") //returns 276
```

Today
Returns the current date, in SQLDate format (YYYY-MM-DD).

Syntax
Result (as string) = Today()

Example
```
Msg(Today)
```

Tomorrow
Returns the date for tomorrow, in SQLDate format (YYYY-MM-DD).

Syntax
Result (as string) = Tomorrow()

Example
```
Msg(Tomorrow)
```

Yesterday
Returns the date for yesterday, in SQLDate format (YYYY-MM-DD).

Syntax
Result (as string) = Yesterday()

Example
```
Msg(Yesterday)
```

DATE/TIME FUNCTIONS

GetNow
Returns the current day and time in YYYY-MM-DD HH-mm-SS format, where
- YYYY = Year
- MM = Month
- DD = Day
- HH = Hour
- mm = Minute
- SS = Second

Syntax
Result (as string) = GetNow

Example
```
"2000-10-02 130532" for October 2, 2000, 1:05:32 pm
```

GetTimestamp
Returns the current day and time in YYMMDDHHmmSS format, where
- YY = Last two digits of the year
- MM = Month
- DD = Day
- HH = Hour
- mm = Minute
- SS = Second

Syntax
Result (as string) = GetTimestamp

Example
```
"050810192054" for August 10, 2005, 7:20:54 pm
```

FILE FUNCTIONS

DeleteFile
This function deletes any file located in any of the subfolders located in your "Lotto Sorcerer v9 Files" folder (which is located in your Documents folder). *Use with caution!* No confirmation or acknowledgement will be displayed.

Syntax
DeleteFile (*Filename as String, FolderName as String*)
where

Filename = a valid filename for the file to be read.
Foldername = a valid foldername within the Lotto Sorcerer v9 Files folder.

Example
```
DeleteFile("Test.txt", "Sandbox")
```

Valid foldernames are: "Backup Files"; "Batch Files"; "Charts"; "Combinations"; "Database"; "Datasets"; "Definition Files"; "Export Files"; "Graphs"; "Hyperlinks"; "Import Files"; "Logs"; "Notes"; "Permutations"; "Playslip Files"; "Preferences"; "Reports"; "Sandbox"; "Scripts"; "Spreadsheets"; "SQL Files"; "Suggestions"; "Temp Files" and "Wheels".

GetFile
This function opens a standard file selector, which allows the user to select a text file on the computer. If the user selects a valid file, this function returns the entire contents as a string.

Syntax
Result (as string) = GetFile ()

Example
```
Dim s as string
s = GetFile()
Print s
ShowTab(2)
```

ListFiles
This function returns a tab-delimited list of files that match the passed extension.

Syntax
Result as String = ListFiles (*FolderName as String, Extension as String*)
where

Foldername = a valid foldername within the Lotto Sorcerer v9 Files folder.
Extension = filename extension to be matched.

Example
This example lists all of the files in the "Scripts" folder that have an extension of "bas" by populating a dropdown menu:

```
Dim List, s as String
Dim Count, n as Integer
List = ListFiles("Scripts", "bas")
Count = CountFields(List, Chr(9))
If Count > 0 then
   List = ReplaceAll(List, Chr(9), ",")
   n = GetDropDown("List of Files", "Select File", "Choose", List)
   Msg("You selected " + nthField(List, ",", n))
End If
```

Valid foldernames are: "Backup Files"; "Batch Files"; "Charts"; "Combinations"; "Database"; "Datasets"; "Definition Files"; "Export Files"; "Graphs"; "Hyperlinks"; "Import Files"; "Logs"; "Notes"; "Permutations"; "Playslip Files"; "Preferences"; "Reports"; "Sandbox"; "Scripts"; "Spreadsheets"; "SQL Files"; "Suggestions"; "Temp Files" and "Wheels".

ReadFile

This function reads a text file located in any of the subfolders located in your "Lotto Sorcerer v9 Files" folder (which is located in your Documents folder). Because no interaction is required from the user, this function is useful for batch files.

If the file does not exist, the returned value will be "Error: File does not exist."

Syntax

Contents (as String) = ReadFile (*Filename as String, FolderName as String*)
where

Filename = a valid filename for the file to be read.
Foldername = a valid foldername within the Lotto Sorcerer v9 Files folder.

Example

```
Dim s as String
s = ReadFile("Test.txt", "Sandbox")
If s = "Error: File does not exist." then
   Msg(s)
else
   Print s
   ShowTab(2)
End If
```

Valid foldernames are: "Backup Files"; "Batch Files"; "Charts"; "Combinations"; "Database"; "Datasets"; "Definition Files"; "Export Files"; "Graphs"; "Hyperlinks"; "Import Files"; "Logs"; "Notes"; "Permutations"; "Playslip Files"; "Preferences"; "Reports"; "Sandbox"; "Scripts"; "Spreadsheets"; "SQL Files"; "Suggestions"; "Temp Files" and "Wheels".

SaveFile

This function opens a standard file selector, which allows the user to save a text file on the computer.

Syntax

SaveFile (*a as String*)
where

a = a text string to save as a file

Example

```
Dim s as String
s = "This is a test."
SaveFile(s)
Msg "File has been saved."
```

WriteFile

This function saves a text file in one of the subfolders located in your "Lotto Sorcerer v9 Files" folder, which is located in your Documents folder. Because no interaction is required from the user, this function is useful for batch files. If the file already exists, it will be overwritten.

Syntax

WriteFile (*Filename as String, Foldername as String, Contents as String*)
where

Filename = a valid filename for the file to be saved.
Foldername = a valid foldername within the Lotto Sorcerer v9 Files folder.
Contents = the contents to be saved.

Example

```
Dim s as String
s = "This is a test."
WriteFile("Test.txt", "Sandbox", s)
Msg "File has been saved."
```

Valid foldernames are: "Backup Files"; "Batch Files"; "Charts"; "Combinations"; "Database"; "Datasets"; "Definition Files"; "Export Files"; "Graphs"; "Hyperlinks"; "Import Files"; "Logs"; "Notes"; "Permutations"; "Playslip Files"; "Preferences"; "Reports"; "Sandbox"; "Scripts"; "Spreadsheets"; "SQL Files"; "Suggestions"; "Temp Files" and "Wheels".

GRAPHICS FUNCTIONS

ClearRect

This function clears the rectangle described by the parameters passed by filling it with the background color of the parent window.

Syntax

ClearRect(*x as Integer, y as Integer, Width as Integer, Height as Integer*)
where

x = x (horizontal) coordinate of the top left of the rectangle
y = y (vertical) coordinate of the top left of the rectangle
Width = width of the rectangle, in pixels
Height = height of the rectangle, in pixels

Example

This clears a rectangular area, 100 pixels wide and high, starting at 130 pixels from the left and 10 pixels from the top:

```
ClearRect(130, 10, 100,100)
```

Draw

This forces a refresh of the graphics engine. Although not required, *it is strongly recommended that you use the Draw command as the last command in a graphics routine* (to accommodate users whose graphics engine is not double-buffered).

Syntax

Draw

DrawCautionIcon

This draws the operating system's Caution icon at the coordinates specified.

Syntax

DrawCautionIcon*(x as Integer, y as Integer)*

Example

This draws the Caution icon at the top-left corner:

```
DrawCautionIcon(0,0)
```

DrawLine

Draws a line from X1, Y1 to X2, Y2 in the current color. The current color is set with the ForeColor property.

Syntax

DrawLine*(x1 as Integer, y1 as Integer, x2 as Integer, y2 as Integer)*

DrawNoteIcon

This draws the operating system's Note icon at the coordinates specified.

Syntax

DrawNoteIcon*(x as Integer, y as Integer)*

DrawOval

This draws the outline of an oval (or circle) in the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the oval. To draw a circle, make the width and height equal.

Syntax

DrawOval*(x as Integer, y as Integer, Width as Integer, Height as Integer)*

DrawPixel

This draws a pixel of the color of the RGB (red, green and blue) colors passed at the location of the x and y coordinates.

Syntax

DrawPixel*(x as Integer, y as Integer, r as Integer, g as Integer, b as Integer)*
where

x = x (horizontal) coordinate
y = y (vertical) coordinate
r = red value (0 to 255)
g = green value (0 to 255)
b = blue value (0 to 255)

Example

This draws a pixel, at a location 64 pixels from the left, 16 pixels from the top, in the color of "violet" (which has an RGB value of 159 red, 0 green and 255 blue):

```
InitializeGraphics
DrawPixel(64,16,159,0,255)
Draw
```

DrawRect

Draws the outline of a rectangle in the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the rectangle.

Syntax

DrawRect*(x as Integer, y as Integer, Width as Integer, Height as Integer)*

DrawRoundRect

Draws the outline of a rounded rectangle in the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the round rectangle. ArcWidth and ArcHeight control the shape of the corners in the horizontal and vertical axes, respectively. They are the distance (in pixels) from the corner at which the arc begins. Setting them to zero results in a rectangle with sharp corners (which would result in the same thing as DrawRect.

Syntax

DrawRountRect*(x as Integer, y as Integer, Width as Integer, Height as Integer, ArcWidth as Integer, ArcHeight as Integer)*

DrawStopIcon

This draws the operating system's Stop icon at the coordinates specified.

Syntax

DrawStopIcon*(x as Integer, y as Integer)*

DrawString

Draws the text at the specified location and in the current color. The current color is set with the ForeColor property. The X parameter specifies the distance from the left of the Graphics object in pixels. The Y parameter specifies the baseline for the text. The optional WrapWidth parameter specifies the width (in pixels) at which Text should wrap. The Text will wrap if WrapWidth is provided and Condense is False (The default is False). If WrapWidth is omitted, then Text will print on one line, even if the window is too narrow to contain the text. If the optional Condense property is True, DrawString truncates the string to fit into the space specified by WrapWidth and uses an ellipsis ("...") to indicate that there is additional text that is not shown. The default values of WrapWidth and Condense are zero and False, respectively. The default behavior is to print the string on one line.

Syntax

DrawString*(Text as String, x as Integer, y as Integer, WrapWidth as Integer, Condense as Boolean)*

Fill

Fills the area specified by x and y with the passed color (in RGB values). Adjacent locations are also filled if they are the same color as the x and y location. *If the area is bounded by curves, it is strongly recommended that you SetAntiAlias to "FALSE".*

Syntax

Fill*(x as Integer, y as Integer, Red as Integer, Green as Integer, Blue as Integer)*

FillOval

Draws an oval (or circle) filled with the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the oval. Set Width and Height to the same value to draw a filled circle.

Syntax

FillOval*(x as Integer, y as Integer, Width as Integer, Height)*

FillRect

Draws a rectangle filled with the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the rectangle.

Syntax

FillRect*(x as Integer, y as Integer, Width as Integer, Height)*

FillRoundRect

Draws a rounded rectangle filled with the current color. The current color is set with the ForeColor property. X and Y are the coordinates of the top-left corner. Width and Height specify the size of the round rectangle. ArcWidth and ArcHeight control the shape of the corners in the horizontal and vertical axes, respectively. They are the distance (in pixels) from the corner at which the arc begins. Setting them to zero results in a rectangle with sharp corners (which would result in the same thing as FillRect.

Syntax

FillRountRect*(x as Integer, y as Integer, Width as Integer, Height as Integer, ArcWidth as Integer, ArcHeight as Integer)*

GetHeight

Returns the current height of the drawing canvas, in pixels.

Syntax

Result *(as Integer)* = GetHeight()

GetPixel

Gets the color of the pixel at x and y. Returns a comma-delimited string value of red, green and blue values.

Syntax

Result *(as String)* = GetPixel*(x as Integer, y as Integer)*

Example

```
Dim r as String
InitializeGraphics
r = GetPixel(50,60)
Print r
ShowTab(2)
```

GetStringDirection

Returns an integer that indicates the direction in which the text is written.

Syntax

Result *(as Integer)* = GetStringDirection*(Text as String)*
where

-1 = Direction unknown
0 = Left to Right
1 = Right to Left

GetStringHeight

Returns as an Integer the height of the text based on the current font and font size (in pixels) and the passed WrapWidth (also in pixels). The WrapWidth parameter specifies the width (in pixels) at which text should wrap.

Syntax

Result *(as Integer)* = GetStringHeight*(Text as String, WrapWidth as Integer)*

GetStringWidth
Returns as an integer the width of Text in pixels.

Syntax
Result *(as Integer)* = GetStringWidth*(Text as String)*

GetTextAscent
Returns the ascent of a line of text drawn with the current font. *TextAscent* is the height of the tallest font letter above the font baseline.

Syntax
Result *(as Integer)* = GetTextAscent()

GetTextHeight
Contains the height of a line of text drawn with the current font.

Syntax
Result *(as Integer)* = GetTextHeight()

GetWidth
Returns the current width of the drawing canvas, in pixels.

Syntax
Result *(as Integer)* = GetWidth()

InitializeGraphics
This routine *must* be called *before* calling *any* other graphic routines or functions, otherwise an error *will* occur.

Syntax
InitializeGraphics()

SetAntiAlias
This is to draw smooth lines and shapes, including text where applicable. It is strongly recommended that you turn antialiasing off (that is, "`SetAntiAlias(FALSE)`") if using the Fill function with geometric entities with curves.

To use this function, either pass TRUE or FALSE to turn antialiasing on or off, respectively.

Syntax
SetAntiAlias*(Value as Boolean)*

SetBold
If passed TRUE, this sets subsequent font as Bold style.

Syntax
SetBold*(Value as Boolean)*

SetColor

The currently selected color for the Graphics object. This color is used by the various drawing methods. Pass the color as red, green and blue integer values, from 0 to 255.

Syntax

SetColor*(Red as Integer, Green as Integer, Blue as Integer)*

Example

This example draws a golden square and a turquoise circle:

```
InitializeGraphics
SetColor(218,165,32) //Goldenrod
FillRect(10,10,50,50)
SetColor(64,244,208) //Turquoise
FillOval(90,80,50,50)
Draw
ShowTab(4)
```

SetFont

Name of the font used to display the caption or text content. You can enter any font that is installed on the computer or the names of two metafonts, "System" and "SmallSystem." The System font is the font used by the system software as its default font. Different operating systems use different default fonts. If the system software supports both a large and small System font, you can also specify the "SmallSystem" font as your TextFont.

If you are planning to distribute your code to others, it is strongly recommended that you use fonts that are supplied with the operating system. Invoking a font name that does not exist on the target computer can cause unpredictable results.

Syntax

SetFont*(FontName as String)*

Example

```
InitializeGraphics
SetFont("Arial")
DrawString("Hello, World!",40,40,0,False)
SetFont("Courier New")
DrawString("Goodbye, World!",40,70,0,False)
Draw
ShowTab(4)
```

SetItalic

If passed TRUE, this sets subsequent font as Italic style.

Syntax

SetItalic*(Value as Boolean)*

SetPenHeight

The height in pixels used when drawing lines, ovals, and rectangles.

Syntax

SetPenHeight*(PenHeight as Integer)*

SetPenWidth

The width in pixels used when drawing lines, ovals, and rectangles.

Syntax
SetPenHeight*(PenWidth as Integer)*

SetTextSize
Size of the font used to when drawing text.

Syntax
SetTextSize *(Size as Integer)*

SetUnderline
If passed TRUE, this sets subsequent font as Underline style.

Syntax
SetUnderline*(Value as Boolean)*

INTERFACE INPUT FUNCTIONS

CLS
Clears the Output tab.

Syntax
CLS

GetDropDown
This function invokes a custom window, allowing the user to select a value from a custom dropdown menu. This function returns an integer. If the returned integer is 0 (zero), the user selected cancel. Otherwise, the integer represents the item number in the dropdown menu the user selected (1-based).

Syntax
Result (as integer) = GetDropDown (*FormCaption As String, Prompt As String, ButtonCaption As String, Values As String*)
where

FormCaption = a text string containing the title of the window
Prompt = a text string containing the prompt
ButtonCaption = a text string containing the caption of the affirmative button
Values = a comma-delimited string of values to populate the dropdown menu

Example
```
Dim z as String
Dim n as Integer
z = "47167,47404,48214,26041,02116,94705,46360"
n = GetDropDown("ZIPCode Selector", "Select ZIP Code","Select",z)
Print "You selected item # " + Str(n) + " (" + Chr(34) + nthField(z, ",", n) +
Chr(34) + ")"
ShowTab(2)
```

GetInput
This function invokes a custom dialog box, allowing the user to enter values.

Syntax

Result (as string) = GetInput (*Prompt As String, DefaultValue As String, LimitText As Integer, Mask As String, DialogTitle as String*)
where

Prompt = a text string containing the prompt
DefaultValue = a text string containing the default value
LimitText = an integer containing the maximum number of characters allowed
DialogTitle = a string containing the title of the dialog box
Mask = a string containing the mask. Use the Mask property to filter user input on a character-by-character basis and add formatting characters. For example, a mask for a Telephone number field can add parentheses, spaces, and dashes as literals, that are used for formatting, and the digit mask symbol '#' to restrict entry to numbers only. It uses the same mask characters as Visual Basic.

MASK	DESCRIPTION	NOTES
#	Single digit placeholder	The user can type only a digit (numeric) character in this position. For example, the mask "(###) ###-####" accepts the entry 5551212121" and returns "(555) 121-2121".
.	Decimal separator	The decimal placeholder that is actually used is specified in the user's International settings. The character is treated as a literal (formatting) character for masking purposes. For example, the mask "##.##" accepts the entry "2344" and returns "23.44" (for US systems).
,	Thousands separator	The thousands separator that is actually used is specified in the user's International settings. The character is treated as a literal (formatting) character for masking purposes. For example, the mask "####,###" accepts the entry "123456" and returns "123,356".
:	Time separator	The time separator that is actually used is specified in the user's International settings. The character is treated as a literal (formatting) character for masking purposes.
/	Date separator	The date separator that is actually used is specified in the user's International settings. The character is treated as a literal (formatting) character for masking purposes. For example, the mask "99/99/\2099" accepts the entry "123109" and returns "12/31/2009". The "\20" enters the default century and decade and only accepts the year in the first decade of the century.
\	Mask escape character	Treats the next character in the mask as a literal. The escape character enables you to use the '#', '&', 'A', '?' (and so on) characters in the mask. The escapted character is treated as a literal (formatting) character. For example, the mask "\C\C-9999" accepts the entry "1234" and returns "CC-1234".
&	Character or space placeholder	Valid values are the ASCII characters 32-126 and the non-ASCII characters 128-255. For example, the mask "&&-99999" accepts "li20520" and returns "li-20520".
C	Optional character or space placeholder	Character or space placeholder, where entry is optional. It operates like the '&' placeholder. For example, the mask "CCCC-CC" formats "1233ed" as "1233-ed".
>	Convert to uppercase	Uppercasing works beyond the ASCII range where appropriate, e.g., ü becomes Ü. For example, the mask ">&&-#####" accepts the string "li20520" and returns "LI-20520".
<	Convert to lowercase	Lowercasing works beyond the ASCII range where appropriate, e.g., Ü becomes ü.
A	Mandatory alphanumeric placeholder	For example, the spec "AAA" specifies three alphanumeric characters.
a	Optional alphanumeric placeholder	Alphanumeric character placeholder, where entry is optional.

0	Literal zero	For example, the mask "99.00" formats the entry "22" as "22.00". The mask "\C\C0-9999" accepts the entry "1234" and returns it as "CC0-1234". The mask "##,###.00" accepts the entry "12345" and returns "12,345.00". The mask "99.00" accepts "21" and returns "21.00".
9	A single numeric digit	
?	Alphabetic placeholder	Entry is optional. For example, the mask "???" accepts three alphabetic characters. It rejects numeric characters.
Any literal	All other symbols displayed as literals	All other symbols are displayed as literals for formatting purposes. For example, the mask "999" accepts the entry "333" and returns "333"

Example
This example forces the user to enter only numbers (with a maximum of eight digits).

```
Dim r as String
r = GetInput("Enter date","20050810", 8, "99999999", "Enter Date")
Print "You entered: " + r
ShowTab(2)
```

IMessage
This function invokes an interactive message box, allowing the user to select specific buttons.

Syntax
Result (as integer) = MsgBox (*Message As String, Buttons As Integer*)
where

Message = a text string containing the message to be displayed
Buttons = a bitwise integer for selecting buttons, the icon and the default button

Result:

1 = OK pressed
2 = Cancel pressed
3 = Abort pressed
4 = Retry pressed
5 = Ignore pressed
6 = Yes pressed
7 = No pressed

Number and Type of Buttons

Value	Description
0	Display OK button only
1	Display OK and Cancel buttons
2	Display Abort, Retry and Ignore buttons
3	Display Yes, No and Cancel buttons
4	Display Yes and No buttons
5	Display Retry and Cancel buttons

Icon to Be Displayed *(Microsoft Windows only)*

Value	Description
0	No icon
16	Stop sign icon
32	Question icon
48	Caution triangle icon

64	Note icon

Selection of Default Button

Value	Description
0	First button of Group 1 list is the default
256	Second button of Group 1 list is the default
512	Third button of Group 1 list is the default
768	No button is the default

Since the value of the button parameter is the sum, you make a selection from each of the three tables, add up their values and pass that value. For example, if you want the buttons to be the "Abort, Retry, and Ignore" set, with a Caution icon, and Retry as the default, you would add up 2 + 48 + 256 = 306.

Example

```
Dim m as String
Dim n as Integer
m = "Do you really want to delete the database record?"
n = IMessage(m, 52) //"52" means display "Yes" or "No", with a Caution icon
Select Case n
      Case 6 //"Yes" pressed
            Msg("You pressed 'yes'")
      Case 7 //"No" pressed
            Msg("You pressed 'no'")
End Select
```

Msg

This function shows a simple message box.

Syntax

Msg(*Message as String*)

MsgDialog

This function is used to design and display customized message dialog boxes.

Syntax

Result (as integer) = MsgDialog (*Tiles as String, Icon as Integer, Message As String, Explanation as String, Button1Caption as String, Button2Caption as String, Cancel as Boolean*)

where

Title = a text string containing the title of the dialog window

Icon = an integer representing the type of icon used in the dialog window:

Value	Description
0	No icon
1	Note icon
2	Caution icon
3	Stop icon
4	Question icon

Message = a text string containing the primary message

Explanation = a text string containing additional message

Button1Caption = a text string containing the caption for the first button

Button2Caption = a text string containing the caption for the second button

Cancel = a boolean value determining whether the cancel button is visible or not

This function returns an integer value:

Value	Description
0	Button 1 pressed
1	Button 2 pressed
2	Cancel button pressed

Example

```
Dim m1, m2, b1, b2 as String
Dim n as Integer
m1 = "Do you really want to delete the database record?"
m2 = "This cannot be undone."
b1 = "Sure!"
b2 = "No way!"
n = MsgDialog("Delete Record",2,m1,m2,b1,b2,False)
Select Case n
      Case 0 //First button pressed
            Msg("You pressed '" + b1 + "'")
      Case 1 //Second button pressed
            Msg("You pressed '" + b2 + "'")
End Select
```

INTERFACE OUTPUT FUNCTIONS

GridAddRow

This adds a row to the grid (and populates the first cell of the grid).

Syntax

GridAddRow (*CellContents as String*)

Example

```
GridAddRow("Test")
ShowTab(3)
```

GridClearAll

This clears the grid.

Syntax

GridClearAll

Example

```
GridClearAll
ShowTab(3)
```

GridColAlignment

This sets the text alignment for a column in the grid.

Syntax

GridColAlignment (*Column as Integer, Alignment as Integer*)
where

Alignment = 0 (default alignment)
Alignment = 1 (left alignment)
Alignment = 2 (center alignment)
Alignment = 3 (right alignment)
Alignment = 4 (decimal alignment)

Notes
"Column" is zero-based. So the first column would be "0", the second column would be "1", the third column would be "2", et cetera.

"Decimal alignment" aligns the decimal separator to the right edge of the column. You need to use *GridColAlignOffset* to move the alignment point in the column.

GridColAlignOffset
Modifies the alignment point and is especially useful for decimal alignment. n pixels from the right edge of the column. The first column is numbered zero.

The value is the distance in pixels from the right edge of the column. A negative value moves the decimal separator to the left, i.e., into the body of the column. Columns are zero-based.
GridColAlignOffset (*Column as Integer, Offset as Integer*)

GridColWidths
This sets the widths of all columns in the grid.

Syntax
GridColWidths (*WidthDescription as String*)

A list of comma separated values, with each value controlling the width of the associated column. A value can be an absolute value (in pixels), a percentage, a relative length expressed as i* where i is an integer, or an "*" that indicates "fill in the remaining width." If you use percentages, you can use non-integer values to specify fractions of a percent, e.g., 43.52%. The percentage value can be greater than 100%.

If you use pixels, the last column doesn't grow to the size of the rest of the ListBox. You should set the width of the last column to "*" and it will automatically take up the remaining width of the grid.

Without any column width specifications, the headers will be divided evenly. If there are fewer column widths specified than the total number of columns, the remaining columns will divide up the remaining width equally.

An element with a length of "3*" will be allotted three times the space of an element with length "1*". The value "*" is equivalent to "1*" and can be used to mean "fill the remaining space."

GridPostCell
This adds a value to a cell in the grid.

Syntax
GridPostCell(*Row as Integer, Column as Integer, Contents as String*)
where

Row = Row number (zero-based)
Column = Column number (zero-based)
Contents = The string contents to be added to the cell.

Note that the grid only hold string values, so numbers will need to be converted to strings (using `STR()`, `CSTR()` or `FORMAT()` before being added. Also, you must have added the appropriate row to the grid by using the `GridAddRow()` function first.

GridSetColNumber
This sets the number of columns in the grid.

Syntax
GridSetColNumber(*NumberOfColumns as Integer*)

This number is 1-based.

GridSetHeadings
This sets the headings of the grid.

Syntax
GridSetHeadings(*ColumnNumber as Integer, Heading as String*)
where
ColumnNumber = column number (zero-based)
Heading = String containing the heading title
Example
GridSetHeadings(0, "Date")

Print
This clears the Output box (found on the second tab of the Scripting Laboratory) and prints the passed string to it.

Syntax
Print(*StringToPrint as String*)

Example
```
Print "Hello, World."
ShowTab(2)
```

Say
Uses the computer's speech synthesizer to pronounce the passed text string.

Syntax
Say(*StringToSay as String)*

Example
```
Say("This is a test")
```

ShowTab
This forces the desired tab in the Scripting Laboratory into view. This is extremely useful, because users may not be aware that the results of the program are in another tab.

Syntax
ShowTab(*TabNumber as Integer*)
where
1 = "Source Code" tab
2 = "Output" tab
3 = "Grid" tab
4 = "Graphics" tab

Example
```
ShowTab(2)
```
INTERFACE SETTINGS

SetBackground
This sets the background of the Output tab. Enter with red, green and blue integer values (0-255).

Syntax
SetBackground (*Red as Integer, Green as Integer, Blue as Integer*)

Example
This example sets the output screen as a clone of the classic Commodore 64 computer:

```
SetBackground(67,67,231)
SetForeground(165,165,255)
Print "**** COMMODORE 64 BASIC V2 ****"
Print "64K RAM SYSTEM  38911 BASIC BYTES FREE"
Print "READY."
ShowTab(2)
```

SetForeground
This sets the foreground (text color) of the Output tab. Enter with red, green and blue integer values (0-255).

Syntax
SetForeground(*Red as Integer, Green as Integer, Blue as Integer*)

For an example, please see the "SetBackground" function (above).

INTERNET FUNCTIONS

GetHTTP
Returns the results of the passed http URL as a string. Note: *only* http:// protocal is handled.

Syntax
GetHTTP(*URL as String*)

Example
```
Print(GetHTTP("http://www.excite.com"))
ShowTab(2)
```
LOTTO SORCERER FUNCTIONS

Count3FactorNumbers
Returns the number of 3-factor numbers in passed dash-delimited number string.

Syntax
Result (as integer) = Count3FactorNumbers([dash-delimited number string])

Example
```
Dim n as integer
n = Count3FactorNumbers("01-07-08-11-13-18")
```

```
Msg Str(n) //Returns "1", because "18" is the only 3-factor number in the
passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountAbundantNumbers

Returns the number of abundant numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountAbundantNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountAbundantNumbers("01-07-08-11-13-18")
Msg Str(n) //Returns "1", because "18" is the only abundant number in the
passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountAdjacentNumbers

Returns the number of adjacent ("back-to-back") numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountAdjacentNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountAdjacentNumbers("1-3-5-6-15")
Msg Str(n) //Returns "2", because "5" and "6" are adjacent to each other.
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountCompositeNumbers

Returns the number of composite numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountCompositeNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountCompositeNumbers("1-3-5-6-15")
Msg Str(n) //Returns "2", because only "6" and "15" are composite numbers.
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountDeficientNumbers

Returns the number of deficient numbers in passed dash-delimited number string.

Syntax
Result (as integer) = CountDeficientNumbers([dash-delimited number string])

Example
```
Dim n as integer
n = CountDeficientNumbers("1-3-5-6-15")
Msg Str(n) //Returns “4”, because only “6” is not a deficient number.
```

Note:s:
- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountFibonacciNumbers
Returns the number of Fibonacci numbers in passed dash-delimited number string.

Syntax
Result (as integer) = CountDeficientNumbers([dash-delimited number string])

Example
```
Dim n as integer
n = CountFibonacciNumbers("1-3-5-6-15")
Msg Str(n) //Returns “3”, because “1”, “3” and “5” are the only Fibonacci num-
bers in the passed string
```

Note:s:
- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountPadovanNumbers
Returns the number of Padovan numbers in passed dash-delimited number string.

Syntax
Result (as integer) = CountPadovanNumbers([dash-delimited number string])

Example
```
Dim n as integer
n = CountPadovanNumbers("1-3-5-6-15")
Msg Str(n) //Returns “3”, because “1”, “3” and “5” are the only Padovan numbers
in the passed string
```

Note:s:
- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountPentagonalNumbers
Returns the number of pentagonal numbers in passed dash-delimited number string.

Syntax
Result (as integer) = CountPentagonalNumbers([dash-delimited number string])

Example
```
Dim n as integer
```

```
n = CountPentagonalNumbers("1-3-5-6-15")
Msg Str(n) //Returns "2", because "1" and "5" are the only pentagonal numbers
in the passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountPerfectNumbers

Returns the number of perfect numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountPerfectNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountPerfectNumbers("1-3-5-6-15")
Msg Str(n) //Returns "3", because "3", "6" and "15" are the only perfect num-
bers in the passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountPrimeNumbers

Returns the number of prime numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountPrimes([dash-delimited number string])

Example

```
Dim n as integer
n = CountPrimeNumbers("1-3-5-6-15")
Msg Str(n) //Returns "2", because "1" and "3" are the only prime numbers in the
passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountRepeatNumbers

Returns the number of repeated numbers in passed dash-delimited number strings.

Syntax

Result (as integer) = CountRepeatNumbers([first dash-delimited number string, second dash-delimited string])

Example

```
Dim n as integer
n = CountRepeatNumbers("1-3-5-10-15", "4-6-7-9-10")
Msg Str(n) //Returns "1", because the number "10" is the only repeating number.
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed strings do not need to be sorted.

CountSemiPerfectNumbers

Returns the number of semi-perfect numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountSemiPerfectNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountSemiPerfectNumbers("1-3-5-6-15")
Msg Str(n) //Returns "1", because "6" is the only semi-perfect number in the
passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountSemiPrimeNumbers

Returns the number of semi-prime numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountSemiPrimeNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountSemiPrimeNumbers("1-3-5-6-15")
Msg Str(n) //Returns "2", because "6" and "15" are the only semi-prime numbers
in the passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountSquareNumbers

Returns the number of square numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountSquareNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountSquareNumbers("1-3-5-6-15")
Msg Str(n) //Returns "1", because "1" is the only square number in the passed
string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountTriangularNumbers

Returns the number of triangular numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountTriangularNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountTriangularNumbers("1-3-5-6-15")
Msg Str(n) //Returns “4”, because “5” is the only number in the passed string
that is not triangular
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

CountUlamNumbers

Returns the number of ulam numbers in passed dash-delimited number string.

Syntax

Result (as integer) = CountUlamNumbers([dash-delimited number string])

Example

```
Dim n as integer
n = CountUlamNumbers("1-3-5-6-15")
Msg Str(n) //Returns “3”, because “1”, “3” and “6” are the only Ulam numbers in
the passed string
```

Note:s:

- This function only analyzes numbers from 0 to 99.
- Passed string does not need to be sorted.

GenerateSuggestions

Causes the "Start" button in the Main Window to be pressed (effectively starting the suggestion generation process). This will work only if a lottery has been selected in the Main Window.

GetAnalysis

Returns the contents of the Analysis in Lotto Sorcerer's Main Window.

Syntax

Result (as string) = GetAnalysis()

Example

```
Dim r as string
r = GetAbbreviatedDate("2005-08-10") //returns the date in the user's "Abbrevi-
ated date" format
```

GetAnalysisEngine_Engine

Returns the type of analysis engine in the Projection Parameters of the Main Window.
where

0 = Pattern Recognition (neural network)
1 = Deep Pattern Recognition (neural network)
2 = Forecast 1 (small segments)
3 = Forecast 2 (large segments)

Syntax
Result (as integer) = GetAnalysisEngine_Engine ()

GetAnalysisEngine_Mode

Returns the type of analysis mode in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetAnalysisEngine_Engine ()
where

0 = Pool Temperature
1 = Parity
2 = Distribution

GetAnalysisEngine_SamplingSize

Returns the Sampling Size in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetAnalysisEngine_SamplingSize ()
where

0 = Sampling size of "3"
1 = Sampling size of "4"
2 = Sampling size of "5"
3 = Sampling size of "6"
4 = Sampling size of "7"
5 = Sampling size of "8"
6 = Sampling size of "9"
7 = Sampling size of "10"

GetAnalysisEngine_Sectors

Returns the Sectors value in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetAnalysisEngine_Sectors ()
where

0 = Sampling size of "2"
1 = Sampling size of "3"
2 = Sampling size of "4"
3 = Sampling size of "5"
4 = Sampling size of "6"
5 = Sampling size of "7"

GetAssertionFilters

Returns the Assertion Filters value (as a bitwise integer) in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetAssertionFilters ()
where

Bit #1 = "Adjacent 2"
Bit #2 = "Repeat 1"
Bit #3 = "Prime 1+"
Bit #4 = "Factor of 3"
Bit #5 = "Triangular Numbers 1+"
Bit #6 = "Ulam Numbers 1+"
Bit #7 = "Calculations"

GetLastDraw

Returns the last drawing date in the lottery in SQLDate (YYYY-MM-DD) format.

Syntax

Result (*as String*) = GetLastDraw(*TableName as String*)

Example

```
Print GetLastDraw(GetTable(7))
ShowTab(2)
```

GetLimitationDev

Returns the current setting from the Limitation Deviation radio buttons in the Projection Parameters of the Main Window.

Syntax

Result (*as Integer*) = GetLimitationDev
where

0 = ±1
1 = ±2
2 = ±3

Example

```
Print Str(GetLimitationDev)
ShowTab(2)
```

GetLimitationFilters

Returns the Limitation Filters value (as a bitwise integer) in the Projection Parameters of the Main Window.

Syntax

Result (as integer) = GetLimitationFilters ()
where

Bit #1 = "Arithmetic Mean"
Bit #2 = "Geometric Mean"
Bit #3 = "Harmonic Mean"
Bit #4 = "Median"
Bit #5 = "Population (SD)"
Bit #6 = "Range"
Bit #7 = "Truncated Mean"
Bit #8 = "Variance (SD)"
Bit #9 = "Variance (SPD)"
Bit #10 = "Winsorized Mean"

GetNeuralDepth

Returns the Neural/Analysis value in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetNeuralDepth ()

GetNotes

Returns the contents of the Notes box in the Notes tab of the Main Window.

Syntax
Result (as String) = GetNotes ()

GetRejectionFilters

Returns the Rejection Filters value (as a bitwise integer) in the Projection Parameters of the Main Window.

Syntax
Result (as integer) = GetRejectionFilters ()
where

Bit #1 = "Prior Drawn Numbers"
Bit #2 = "Adjacent 2+"
Bit #3 = "Adjacent 3+""
Bit #4 = "Adjacent 4+""
Bit #5 = "Skewed Hot"
Bit #6 = "Skewed Medium"
Bit #7 = "Skewed Cold "
Bit #8 = "Skewed Even"
Bit #9 = "Skewed Odd"
Bit #10 = "Skewed High"
Bit #11 = "Skewed Middle"
Bit #12 = "Skewed Low"
Bit #13 = "Repeat"
Bit #14 = "User Defined"

GetScopeEnd

Returns the ending date of the Scope setting in the Projection Parameters of the Main Window, in SQLDate (YYYY-MM-DD) format.

Syntax
Result (as String) = GetScopeEnd ()

GetScopeStart

Returns the starting date of the Scope setting in the Projection Parameters of the Main Window, in SQLDate (YYYY-MM-DD) format.

Syntax
Result (as String) = GetScopeStart ()

GetSuggestions

Returns the generated suggestions from the Projection Results tab in the Main Window. If there are more than one suggestion, then each suggestion is separated by the End-of-line terminator.

Syntax
Result (*as String*) = GetSuggestions()

PostSuggestions

This function sends passed suggestions to the Projection Results tab in the Main Window. If more than one suggestion is passed, it is expected that suggestions be separated by an End-of-line terminator string. *Note:* if there are any suggestions currently existing in the Projection Results box, this function will delete them.

Syntax
PostSuggestions(*Suggestions as String*)

Example
```
Dim Suggestions(4), EOL, s as String
Dim i as Integer
EOL = GetEOL()
Suggestions(0) = "01-02-09-11-13-19"
Suggestions(1) = "02-05-08-10-11-25"
Suggestions(2) = "16-24-31-34-37-39"
Suggestions(3) = "03-06-25-26-33-35"
Suggestions(4) = "04-17-18-24-28-30"
//Combine into one, separated by end-of-line
For i = 0 to 4
      s = s + Suggestions(i)
      if i < 4 then
            s = s + EOL
      end if
Next
PostSuggestions(s)
```

SelectLottery
This function programmatically selects the desired lottery from the "Select Lottery" dropdown menu in the Main Window. Enter this function with the number of the lottery. For example, the first lottery in the dropdown menu is "1", the second is "2", et cetera.

Syntax
SelectLottery (*LotteryNumber as Integer*)

Example
```
SelectLottery(2)
```

SetAnalysisEngine_Engine
This programmatically sets the Analysis Engine dropdown menu in the Projection Parameters in the Main Window.

Syntax
SetAnalysisEngine_Engine (*EngineNumber as Integer*)
where

0 = Pattern Recognition (neural network)
1 = Deep Pattern Recognition (neural network)
2 = Forecast 1 (small segments)
3 = Forecast 2 (large segments)

Example
```
SetAnalysisEngine_Engine(3) //To set to "Forecast 2 (large segments)
```

SetAnalysisEngine_Mode

This programmatically sets the Analysis Engine Mode dropdown menu in the Projection Parameters in the Main Window.

Syntax

SetAnalysisEngine_Mode (*ModeNumber as Integer*)
where

0 = Pool Temperature
1 = Parity
2 = Distribution

Example

```
SetAnalysisEngine_Mode(2) //To set to "Distribution"
```

SetAnalysisEngine_SamplingSize

This programmatically sets the Sampling Size dropdown menu in the Projection Parameters in the Main Window.

Syntax

SetAnalysisEngine_SamplingSize (*SamplingSizeNumber as Integer*)
where

0 = Sampling Size of "3"
1 = Sampling Size of "4"
2 = Sampling Size of "5"
3 = Sampling Size of "6"
4 = Sampling Size of "7"
5 = Sampling Size of "8"
6 = Sampling Size of "9"
7 = Sampling Size of "10"

Example

```
SetAnalysisEngine_SamplingSize(4) //To set to "7"
```

SetAnalysisEngine_Sectors

This programmatically sets the Sectors dropdown menu in the Projection Parameters in the Main Window.

Syntax

SetAnalysisEngine_Sectors (*SectorNumber as Integer*)
where

0 = Sampling Size of "2"
1 = Sampling Size of "3"
2 = Sampling Size of "4"
3 = Sampling Size of "5"
4 = Sampling Size of "6"
5 = Sampling Size of "7"

Example

SetAnalysisEngine_Sectors (3) //To set to "5"

SetAssertionFilters

This sets the Assertion filters in the Projection Parameters tab in the Main Window.

Syntax

SetAssertionFilters(*FilterValues as Integer*)

To set, add up the values of the filters you want to set, and pass this value to the function.

Filter	Value
Adjacent 2	1
Repeat 1	2
Prime 1+	4
Factor of 3	8
Triangular Numbers 1+	16
Ulam Numbers 1+	32
Calculations	64

For example, if you wanted the "Repeat 1", "Prime 1+" and the "Ulam Numbers 1+", add the values 2 + 4 + 32 = 38. So you would pass this value as follows:

Example
```
SetAssertionFilters(38)
```

SetLimitationDeviation
This sets the Limitation Deviation "radio buttons" in the Projection Parameters tab in the Main Window.

Syntax
SetLimitationDeviation (*SettingValue as Integer*)

To set, use these values:

Setting	Value
±1	0
±2	1
±3	2

Example
```
SetLimitationDeviation(2) //sets the ±3 radio button
```

SetLimitationFilters
This sets the Limitation filters in the Projection Parameters tab in the Main Window.

Syntax
SetLimitationFilters (*FilterValues as Integer*)

To set, add up the values of the filters you want to set, and pass this value to the function.

Filter	Value
Arithmetic Mean	1
Geometric Mean	2
Harmonic Mean	4
Median	8
Population (SD)	16
Range	32
Truncated Mean	64
Variance (SD)	128
Variance (SPD)	256
Winsorized Mean	512

For example, if you wanted the "Arithmetic Mean", "Range" and the "Winsorized Mean", add the values 1 + 32 + 512 = 545. So you would pass this value as follows:

Example
```
SetLimitationFilters (545)
```

SetNeuralDepth
This sets the Neural/Analysis Depth slider in the Projection Parameters in the Main Window. Pass a value between 1 and 256, inclusive.

Syntax
SetNeuralDepth(*Neural/AnalysisValue as Integer*)

Example
```
SetNeuralDepth(64)
```

SetNote
This fills the content of the Note box in the Notes tab in the Main Window. Note that using this function will overwrite whatever is already there.

Syntax
SetNote(*Note as String*)

SetRejectionFilters
This sets the Rejection filters in the Projection Parameters tab in the Main Window.

Syntax
SetRejectionFilters (*FilterValues as Integer*)
To set, add up the values of the filters you want to set, and pass this value to the function.

Filter	Value
Prior Drawn Number	1
Adjacent 2+	2
Adjacent 3+	4
Adjacent 4+	8
Skewed Hot	16
Skewed Medium	32
Skewed Cold	64
Skewed Even	128
Skewed Odd	256
Skewed High	512
Skewed Middle	1024
Skewed Low	2048
Repeat	4096
User Defined	8192

For example, if you wanted the "Prior Drawn Numbers", "Skewed Even", "Skewed Odd" and "Repeat", add the values 1 + 128 + 256 + 4096 = 4481. So you would pass this value as follows:

Example
```
SetRejectionFilters (4481)
```

SetScopeEnd

This function sets the ending Scope date control in the Projection Parameters tab in the Main Window. Pass the date as a SQLDate formatted string (i.e., YYYY-MM-DD).

Syntax

SetScopeEnd(*Date as String*)

Example

```
SetScopeEnd("2005-08-10")
```

SetScopeStart

This function sets the starting Scope date control in the Projection Parameters tab in the Main Window. Pass the date as a SQLDate formatted string (i.e., YYYY-MM-DD).

Syntax

SetScopeStart(*Date as String*)

Example

```
SetScopeStart("2005-08-10")
```

MATH FUNCTIONS

Abs

Returns the absolute value of the number specified.

Syntax

Result as Double = Abs(*Value*)
where
Value = any number type

Example

```
Dim d as Double
d=Abs(23.9)  //returns 23.9
d=Abs(-23.9) //returns 23.9
```

Acos

Returns the arccosine of the value specified. The arccosine is the angle whose cosine is value. The returned angle is given in radians.

Syntax

Result as Double = Acos(*Value as Double*)

Example

```
Dim d, PI as Double
PI=3.14159265358979323846264338327950
d=Acos(.5) //returns 1.0471976
d=Acos(.5)*180/PI //returns 60
```

Asin

Returns the arcsine of the value specified.

Syntax

Result as Double = Asin(*Value as Double*)

Example
```
Dim d, PI as Double
PI=3.14159265358979323846264338327950
d=Asin(.5)  //returns 0.5235988
d=Asin(.5)*180/PI  //returns 30
```

Atan
Returns the arctangent of the value specified. The arctangent is the angle whose tangent is *value*.

Syntax
Result as Double = Atan(*Value as Double*)

Example
```
Dim d, PI as Double
PI=3.14159265358979323846264338327950
d=Atan(1) //returns 0.785398 (PI/4 radians)
d=Atan(1)*180/PI // returns 45
```

Atan2
Returns the arctangent of the point whose coordinates are *x* and *y*. The arctangent is the angle from the x-axis to a line drawn through the origin (0,0) and a point with coordinates *x*, *y*.

Syntax
Result as Double = Atan(*y as Double, x as Double*)

Example
```
Dim d, PI as Double
PI=3.14159265358979323846264338327950
d=Atan2(1,0) //returns 1.57
d=Atan2(1,0)*180/PI //returns 90
```

CDbl
Returns the numeric equivalent of the passed string. This function is the same as the Val function but is international-savvy. Use Val if you control the string that is passed, and use CDbl if the string comes from the user. You should use CDbl instead of Val if the string contains separators.

Syntax
Result as Double = CDbl(*Value as String*)

Example
```
Dim n As Double
n = CDbl("12345")  //returns 12345
n = CDbl("54.05car45") //returns 54.05
n = CDbl("123.45") //returns 123.45
n = CDbl("123 45") //returns 123
n = CDbl("123,456") //returns 123456
n = CDbl("auto") //returns 0
```

Ceil
Returns the value specified rounded up to the nearest integer.

Syntax
Result as Integer = Ceil(*Value as Double*)

Example

```
Dim d as Double
d=Ceil(1.234) //returns 2
```

Cos

Returns the cosine of the given angle.

Syntax

Result as Double = Cos(*Value as Double*)

Example

```
Dim d, PI as Double
PI=3.14159265358979323846264338327950
d=Cos(45*PI/180) //returns 0.707
```

Dec

Decrements passed variable by a value of 1.

Syntax

Result (as integer) = Inc(*value as integer*)

Example

```
Dim n as integer
n = 5
Msg(Str(Dec(n))) //Displays "4"
```

Exp

Returns "e" to the power of the value specified.

Syntax

Result as Double = Exp(*Value as Double*)

Example

```
Dim d as Double
d=Exp(10) //returns 22026.4657948
```

Floor

Returns the value specified rounded down to the nearest integer.

Syntax

Result as Integer = Floor(*Value as Double*)

Example

```
Dim d as Double
d=Floor(1.234)  //returns 1
```

GenRandom

Returns a random number within a specified range.

Syntax

Result (as integer) = GenRandom(*low as integer, high as integer*)
where

low = the lowest number in the range
high = the highest number in the range

Example
```
Dim n as integer
n = GenRandom(1,10) //returns an integer between 1 and 10, inclusive
```

Inc
Increments passed variable by a value of 1.

Syntax
Result (as integer) = Inc(*value as integer*)

Example
```
Dim n as integer
n = 6
Msg(Str(Inc(n))) //Displays "7"
```

Log
Returns the natural logarithm of the value specified

Syntax
Result as Double = Log(*Value as Double*)

Example
```
   Dim d As Double
   d=Log(10) //returns 2.302585
```

Hex
Returns as a String the hexadecimal version of the number passed.

Syntax
Result as String = Hex(*Value as Integer*)

Example
```
   Dim hexVersion As String
   hexVersion=Hex(5) //returns "5"
   hexVersion=Hex(75) //returns "4B"
   hexVersion=Hex(256) //returns "100"
```

Max
Returns the largest value passed to it.

Syntax
Result as Double = Max(*Value1, value2, ... valueN*)

Example
```
   Dim d As Double
   d=Max(3.01, 4.05)   //returns 4.05
   d=Max(3.012, 3.011, 1.56) //returns 3.012
```

Min
Returns the smallest value passed to it.

Syntax
Result as Double3 = Min(*Value1, value2, ... valueN*)

Example
```
Dim d As Double
d=Min(3.01, 4.05)   //returns 3.01
d=Min(3.012, 3.011) //returns 3.011
```

Oct
Returns as a String the octal version of the number passed.

Syntax
Result as String = Oct(*Value as Integer*)

Example
```
Dim OctVersion As String
OctVersion=Oct(5)    //returns "5"
OctVersion=Oct(75)   //returns "113"
OctVersion=Oct(256)  //returns "400"
```

Pow
Returns the value specified raised to the power specified.

Syntax
Result as Double = Pow(*Value, Power*)

Example
```
Dim d As Double
d=Pow(4,7) //returns 16384 (four raised to the power of seven)
```

Round
Returns the passed value rounded to the nearest Integer.

Syntax
Result as Integer = Round(*Value as Double*)

Example
```
Dim d as Double
d=Round(1.499) //returns 1
d=Round(1.500) //returns 2
```

Sin
Returns the sine of the given angle.

Syntax
Result as Double = Sin(*Value as Double*)

Example
```
Dim d, PI as Double
d=Sin(0.5) //returns 0.4794255
d=Sin(30*PI/180) //returns .5
```

Sqrt
Returns the square root of the given angle.

Syntax
Result as Double = Sqrt(*Value as Double*)

Example
```
Dim d as Double
d=Sqrt(16) //returns 4
```

Tan
Returns the tangent of the given angle.

Syntax
Result as Double = Tan(*Value as Double*)

Example
```
Dim d, PI as Double
d=Tan(45*PI/180) //returns 1.0
```

Val
Returns the numeric form of a string

Syntax
Result as Double = Val(*Value as String*)

Example
```
Dim n As Double
n = Val("12345")  //returns 12345
n = Val("54.05car45") //returns 54.05
n = Val("123.45") //returns 123.45
n = Val("123 45") //returns 123
n = Val("123,456") //returns 123
n = Val("auto") //returns 0
n = Val("&hFFF")  //returns 4095
n = Val("&b1111")  //returns 15
```

REGULAR EXPRESSION FUNCTIONS

RegExReplace
Syntax
Result (as String) = RegExReplace(*SearchSource as String, SearchString as String, ReplaceString as String*)
where

SearchSource = String of text to be searched
SearchString = Regular Expression to apply
ReplaceString = String to replace

The following example shows how to strip HTML tags from a string by replacing anything between brackets (and the brackets themselves) with nothing:

Example

```
Dim Source, Pattern, Result as String

RegEx_ReplaceAllMatches(TRUE)
Source = "<B>2010-08-10:</B> 01-02-03-09-19-20<BR>"
Pattern = "<[^<>]+>"

Result = RegExReplace(Source, Pattern, "")
Print Result //Result is " 2010-08-10: 01-02-03-09-19-20"
ShowTab(2)
```

RegExSearch

Syntax

Result (as String) = RegExReplace(*SearchSource as String, SearchString as String*)
where

SearchSource = String of text to be searched
SearchString = Regular Expression to apply

If more than one item is found, it is separated by the End-of-Line character.

RegExSearch is an powerful tool that allows you to extract unknown data from text. All that is required is that you need to know the pattern of the data. For example, suppose you wanted to extract lottery numbers from a web page. You do not know the numbers, but you know that the numbers will be in the format of six two-digit numbers, separated by a dash. This example shows it in action:

Example

```
Dim Source, Pattern, Result as String

Dim s, p, result as String
RegEx_ReplaceAllMatches(TRUE)
s = "2010-08-10: 01-02-03-09-19-20<BR><BR>2010-08-11: 03-09-10-25-30-31"
p = "[0-9]{2}-[0-9]{2}-[0-9]{2}-[0-9]{2}-[0-9]{2}-[0-9]{2}"

Result = RegExSearch(s, p)
Print Result
ShowTab(2) //Result is "01-02-03-09-19-20" and "03-09-10-25-30-31"
```

RegExCaseSensitive

Syntax

RegEx_CaseSensitive(*Setting as Boolean*)

Specifies if case is to be considered when matching a string. The default is "False".

RegEx_DotMatchesAll

Syntax

RegEx_DotMatchesAll (*Setting as Boolean*)

Normally, the period matches everything except a new line, this option allows it to match new lines. The default is "False".

RegEx_Greedy

Syntax

RegEx_Greedy (*Setting as Boolean*)

"Greedy" means the search finds everything from the beginning of the first delimiter to end of the last delimiter and everything in-between. The default is "True".

RegEx_ LineEndType

Syntax

RegEx_LineEndType (*Setting as Integer*)

Changes the way `\n` (newline) is expanded for replacement patterns. This property has no effect on search patterns if `RegEx_TreatTargetAsOneLine` is "True". The default is zero ("0").

- 0 = any line ending (Windows, Macintosh, or Unix). This is the default.
- 1 = The default for the current platform. If running on Macintosh, the same as 2; if running on Windows, the same as 3, if running on Linux, the same as 4.
- 2 = Mac ASCII 13 or \r
- 3 = Win32 ASCII 10 or \n
- 4 = Unix ASCII 10 or \n

RegEx_MatchEmpty

Syntax

RegEx_MatchEmpty (*Setting as Boolean*)

Indicates whether patterns are allowed to match the empty string. The default is "True".

RegEx_ReplaceAllMatches

Syntax

RegEx_ReplaceAllMatches (*Setting as Boolean*)

Indicates whether all occurrences of the pattern are to be replaced. The default is "False".

RegEx_StringBeginIsLineBegin

Syntax

RegEx_StringBeginIsLineBegin (*Setting as Boolean*)

Indicates whether a string's beginning should be counted as the beginning of a line. The default is "True".

RegEx_StringEndIsLineEnd

Syntax

RegEx_StringEndIsLineEnd (*Setting as Boolean*)

Indicates whether a string's end should be counted as the end of a line. The default is "True".

RegEx_TreatTargetAsOneLine

Syntax

RegEx_TreatTargetAsOneLine (*Setting as Boolean*)

Ignores internal newlines for purposes of matching against '^' and '$'. The default is "False".

SQL FUNCTIONS

GetGameParams

Enter with the table name; returns, as a comma-delimited string, the game parameters of the specified lottery. Note that the lottery must have been setup first.

Syntax

Result (as String) = GetGameParams(*TableName as String*)
where

Value 1 = Numbers drawn
Value 2 = Numbers played
Value 3 = Minimum pool number
Value 4 = Maximum pool number
Value 5 = Minimum bonus pool number 1
Value 6 = Maximum bonus pool number 1
Value 7 = Minimum bonus pool number 2
Value 8 = Maximum bonus pool number 2
Value 9 = Lottery name

Example

```
Dim r as String
r = GetTable(7)
Print (GetGameParams(r))
ShowTab(2)
```

GetRecordCount

This function returns the number of records in a table.

Syntax

Result (*as Integer*) = GetRecordCount(*TableName as String*)

Example

```
Print (Str(GetRecordCount(GetTable(7))))
ShowTab(2)
```

GetTable

This invokes a window, where the user selects a lottery that he has already set up. You can filter which type of lottery will appear in the list of choices by passing a filter integer. This function return a tablename.

Filter	Lottery Type
1	Lotto-type lotteries*
2	Pick-type lotteries
3	Lotto-type lotteries* and pick-type lotteries
4	Virtual lotteries
5	Lotto-type lotteries* and virtual lotteries
6	Pick-type lotteries and virtual lotteries
7	All lotteries

*includes bonus-type lotteries, lotteries with extra numbers and keno-type lotteries

Syntax

Result (as String) = GetTable(Filter as Integer)

Example
```
Dim r as String
r = GetTable(7)
Print r
ShowTab(2)
```

SQLExecute

Used for sending a SQL command. Returns either the string "OK" or an applicable error message.

Syntax
Result (as string) = ExecuteSQL(*SQL_statement as string*)
where
SQL_statement as valid SQL command

Example
```
Dim s, r as String
s = "update LOTTERIES set DRAWTIME = '1200' where TABLENAME = 'D108'"
r = ExecuteSQL(s) //returns "OK" if no errors; otherwise the error message
```

SQLSelect

Used for retrieving data from the database. Pass a valid SQL statement. Returns either the values requested (as a string)or the exact phrase "RecordSet is NIL."

The data will have the tab characters (ASCII 9) between fields and the system End-of-Line character between records.

Syntax
Results (*as String*) = SQLSelect(*SQLStatement as String*)

Example
```
Dim SQLStatement as String
Dim Result as String

SQLStatement = "select * from D154 order by DRAWDATE limit 5"
Result = SQLSelect(SQLStatement)
Print Result
ShowTab(2)
```

SYSTEM FUNCTIONS

GetClipboard

This function returns the contents of the System Clipboard, if, and only if, the System Clipboard contains text.

Syntax
Result(*as String*) = GetClipboard

GetEOL

This retrieves the End-Of_Line string of the current system (ASCII 13 and ASCII 10 for Windows, ASCII 10 for Mac).

Syntax
Result(as String) = GetEOL

Ring
This function rings the computer's "bell".

Syntax
```
Ring
```

SetClipboard
This function sends the passed text to the System Clipboard.

Syntax
```
Dim t as String
t = "This is a test."
SetClipboard(t)
```

ShowVer
This shows the current version of LSScript in a message box.

Syntax
```
ShowVer
```

STRING FUNCTIONS

Asc
Returns as an Integer, the ASCII value for the first character of a String.

Notes: The Asc function returns the code point for the first character in the passed String. Characters 0 through 127 are the standard ASCII set. They will be the same on practically every platform. Asc returns the code point for whatever encoding the string is in. If you need to get the ASCII code of the first byte of the string rather than the first character, use the AscB function.

Syntax
Result (*as Integer*) = Asc(*String*)

Example
```
   Dim a as Integer
   a = Asc("@") //returns 64
```

AscB
Returns as an Integer, the value for the first byte of a String.

Syntax
Result (*as Integer*) = AscB(*String*)

Example
```
   Msg Str(AscB("a")) //returns 97
   Msg Str(AscB("A")) //returns 65
```

Chr
Returns the character whose ASCII value is passed.

Syntax
Result (*as String*) = Chr(*Value as Integer*)

Example

```
Dim Tab,CR as String
Tab=Chr(9)  //returns a tab
CR=Chr(13) //returns carriage return
```

ChrB

Returns a single byte string whose value is passed.

Syntax

Result (*as String*) = ChrB(*Value as Integer*)

Example

```
Dim s as String
s=ChrB(32)  //returns a space
s=Chr(10) //returns a line feed
```

CStr

Use to convert the passed data type to a String. For real numbers, CStr returns 7 significant digits.

Syntax

Result (*as String*) = CStr(*Data as Variant*)

Example

```
Dim s as String
Dim d as New Date
s=CStr(1) //returns "1", as a string
s=CStr(d.Day) //returns the day number as a string
s=CStr(True) //returns "True"
```

Format

Returns as a string a formatted version of the number passed based on the parameters specified. The Format function is similar to the way spreadsheet applications format numbers. Format will use the information based on the user's locale even if the user's locale is a Unicode-only locale.

Syntax

Result (*as String*) = Format(*Number, formatSpec as String*)

Notes

The *formatSpec* is a string made up of one or more special characters that control how the number will be formatted:

Character	Description
#	Placeholder that displays the digit from the value if it is present. If fewer placeholder characters are used than in the passed number, then the result is rounded.
0	Placeholder that displays the digit from the value if it is present. If no digit is present, 0 (zero) is displayed in its place.
.	Placeholder for the position of the decimal point.
,	Placeholder that indicates that the number should be formatted with thousands separators.
%	Displays the number multiplied by 100.
(	Displays an opening parentheses.
)	Displays a closing parentheses.
+	Displays the plus sign to the left of the number if the number is positive or a minus sign if the number is negative.
-	Displays a minus sign to the left of the number if the number is negative. There is no effect for positive numbers.

E or e	Displays the number in scientific notation.
\character	Displays the character that follows the backslash.

The absolute value of the number is displayed. You must use the + or - signs if you want the sign displayed.

Although the special formatting characters are U.S. characters, the actual characters that will appear are based on the current operating system settings. For example, Windows uses the settings in the user's Regional and Language Options Control Panel. Formatting characters are specified in similar ways on other operating systems.

The *formatSpec* can be made up of up to three formats separated by semicolons. The first format is the format to be used for positive numbers. The second format is the format to be used for negative numbers and the third format is the format to be used for zero.

Examples

Format	Number	Formatted String
#.##	1.786	1.79
#.0000	1.3	1.3000
0000	5	0005
#%	0.25	25%
###,###.##	145678.5	145,678.5
#.##e	145678.5	146e+5
-#.##	-3.7	-3.7
+#.##	3.7	+3.7
#.##;(#.##);\z\e\r\o	3.7	3.7
#.##;(#.##);\z\e\r\o	-3.7	(3.7)
#.##;(#.##);\z\e\r\o	0	zero

The following example returns the number 3560.3 formatted as $3,560.30:

```
Dim s as String
s=Format(3560.3, "\$###,##0.00")
```

InStr

Returns the position of the first occurrence of a string inside another string. The first character is numbered 1.

Syntax

Result (*as Integer*) = InStr(*Start as Integer [optional], Source as String, Find as String*)

Notes

If the find string is not found within the source string, 0 (zero) is returned. If the find string is an empty string, then start is returned. That is, `InStr("This","")` returns 1 and `InStr(3,"This","")` returns 3.

InStr is case-insensitive, even with accented Roman characters and non-Roman alphabets.

If you need to find the byte position of the find string within the source string or need a case-sensitive function, use the InStrB function.

Example

```
Dim first As Integer
first = InStr("This is a test", "t")       //returns 1
first = InStr("This is a test", "is")     //returns 3
first = InStr(4, "This is a test", "is")  //returns 6
first = InStr("This is a test", "tester") //returns 0
```

InStrB

Returns the byte position of the first occurrence of a string inside another string. The first character is numbered 1.

Syntax
Result (*as Integer*) = InStrB(*Start as Integer [optional], Source as String, Find as String*)

Notes
If the find string is not found within the source string, 0 (zero) is returned. InStrB is case-sensitive; it treats source as a series of raw bytes. It should be used instead of InStr when the string represents binary data or when your application will run in a one-byte character set (such as the US system) and you want case-sensitivity.

If you need to find the character position of the find string within the source string, use the InStr function.

Example
```
Dim first As Integer
first = InStrB("This is a test", "T")      //returns 1
first = InStrB("This is a test", "t")      //returns 11
first = InStrB("This is a test", "is")     //returns 3
first = InStrB(4, "This is a test", "is")  //returns 6
first = InStrB("This is a test", "tester") //returns 0
first = InStrB("This Is a test", "Is")     //returns 6
```

Left
Returns the first *n* characters in a source string.

Syntax
Result (*as String*) = Left(*Source as String, count as Integer*)

Example
```
Dim s As String
s=Left("Hello World", 5) //returns "Hello
```

LeftB
Returns the first *n* bytes in a source string.

Syntax
Result (*as String*) = LeftB(*Source as String, count as Integer*)

Example
```
Dim s As String
s=LeftB("Hello World", 5) //returns "Hello
```

Len
Returns the number of characters in the specified string

Syntax
Result (as Integer) = Len(*Source as String*)

Example
```
Dim n As Integer
n=Len("Hello world") //returns 11
```

LenB
Returns the number of bytes in the specified string

Syntax

Result (as Integer) = LenB(*Source as String*)

Example
```
Dim n As Integer
n=LenB("Hello world") //returns 11
```

Lowercase
Converts all characters in a string to lowercase characters.

Syntax
Result (*as String*) = Lowercase(*Source as String*)

Example
```
Dim s As String
s=Lowercase("tHe Quick fOX") //returns "the quick fox"
s=Lowercase("THE 5 LAZY DOGS") //returns "the 5 lazy dogs"
```

LTrim
Returns the string passed with leading (left side) whitespaces removed.

Syntax
Result (*as String*) = LTrim(*Source as String*)

Example
```
Dim s as String
s=LTrim("  Hello World  ") //Returns "Hello World  "
```

Mid
Returns a portion of a string (by counting characters). The first character is numbered 1.

Syntax
Result (*as String*) = Mid(*Source as String, Start as Integer, Length as Integer*)

Example
```
Dim s As String
s = Mid("This is a test", 6) //returns "is a test"
s = Mid("This is a test", 11, 4) //returns "test"
```

MidB
Returns a portion of a string (by counting bytes). The first byte is numbered 1.

Syntax
Result (*as String*) = MidB(*Source as String, Start as Integer, Length as Integer*)

Example
```
Dim s As String
s = MidB("This is a test", 6) //returns "is a test"
s = MidB("This is a test", 11, 4) //returns "test"
```

NthField
Returns a field from a row of data, treated as character data. The first field is numbered 1.

Syntax
Result (*as String*) = nThField(*Source as String, Separator as String, fieldNumber as Integer*)

Example
```
   Dim field As String
   field=NthField("Dan*Smith*11/22/69*5125554323*Male","*",2) //Returns "Smith"
```

NthFieldB
Returns a field from a row of data, treated as binary data. The first field is numbered 1.

Syntax
Result (*as String*) = nThFieldB(*Source as String, Separator as String, fieldNumber as Integer*)

Example
```
   Dim field As String
   field=NthFieldB("Dan*Smith*11/22/69*5125554323*Male","*",2) //Returns
"Smith"
```

ParseString
This function, when passed a string, replaces all non-numeric adjacent characters with a single dash ("-"). This is useful when parsing HTML data for lottery drawings.

Syntax
Result (*as String*) = ParseString(*Text as String*)

Example
```
Dim r, s as String
s = "<b>02</b> <b>10</b> <b>11</b> <b>25</b></b> <b>32</b>
r = ParseString(s) //Returns "02-10-11-25-32"
```

Replace
Replaces the first occurrence of a string with another string by matching characters.

Syntax
Result (*as String*) = Replace(*sourceString, oldString, newString*)

Example
```
   Dim result As String
   result=Replace("The quick fox","fox","rabbit") //returns "The quick rabbit"
   result=Replace("The quick fox","f","b") //returns "The quick box"
   result=Replace("The quick fox","quick","") //returns "The fox"
```

ReplaceB
Replaces the first occurrence of a string with another string by matching bytes.

Syntax
Result (*as String*) = ReplaceB(*sourceString, oldString, newString*)

Example

```
Dim result As String
result=ReplaceB("The quick fox","fox","rabbit") //returns "The quick rabbit"
result=ReplaceB("The quick fox","f","b") //returns "The quick box"
result=ReplaceB("The quick fox","quick","") //returns "The fox"
```

ReplaceAll

Replaces all occurrences of a string with another string by matching characters.

Syntax

Result (*as String*) = ReplaceAll(*sourceString, oldString, newString*)

Example

```
Dim result As String
result=ReplaceAll("The quick fox","fox","rabbit") //returns "The quick rab-
bit"
result=ReplaceAll("The quick fox","f","b") //returns "The quick box"
result=ReplaceAll("The quick fox","quick","") //returns "The fox"
```

ReplaceAllB

Replaces all occurrences of a string with another string by matching bytes.

Syntax

Result (*as String*) = ReplaceAllB(*sourceString, oldString, newString*)

Example

```
Dim result As String
result=ReplaceAllB("The quick fox","fox","rabbit") //returns "The quick rab-
bit"
result=ReplaceAllB("The quick fox","f","b") //returns "The quick box"
result=ReplaceAllB("The quick fox","quick","") //returns "The fox"
```

Right

Returns the last *n* characters from the string specified.

Syntax

Result (*as String*) = Right(*Source as String, Count as Integer*)

Example

```
Dim s As String
s=Right("Hello World", 5)  //returns "World"
```

RightB

Returns the last *n* bytes from the string specified.

Syntax

Result (*as String*) = RightB(*Source as String, Count as Integer*)

Example

```
Dim s As String
s=RightB("Hello World", 5)  //returns "World"
```

RTrim
Returns the string passed with trailing (right side) whitespaces removed.

Syntax
Result (*as String*) = RTrim(*Source as String*)

Example
```
Dim s as String
s=RTrim("  Hello World  ") //Returns "  Hello World"
```

StripExtra
This function strips all extraneous elements out of Source, returns Separator-delimited string.

Syntax
Result (*as String*) = StripExtra(*Source as String, Separator as String*)

Example
```
Dim r as String
r = StripExtra("01,02,03,04,05 BB:06", "-") //Returns "01-02-03-04-05-06"
```

Str
Returns the string form of the value passed.

Syntax
Result (*as String*) = Str(*Value as Number*)

Example
```
Dim s As String
s=Str(123)      //returns "123"
s=Str(-123.44)  //returns "-123.44"
s=Str(123.0045) //returns "123.0045"
```

StrComp
Makes a binary (case-sensitive) or text (lexicographic) comparison of the two strings passed and returns the result.

Syntax
Result (*as Integer*) = StrComp(*String1 as String, String2 as String, Mode as Integer*)

Part	Type	Description
result	Integer	If string1 < string2 the function returns -1 If string1 = string2 the function returns 0 If string1 > string2 the function returns 1
string1	string	The first string for comparison
string2	string	The second string for comparison
mode	integer	0 = binary (case-sensitive) 1 = text (lexicographic)

Example
```
StrComp("Spam", "spam", 1) //Returns -1
```

Titlecase

Converts all characters in a string to Titlecase characters (that is, the first letter in each word is capitalized; everything else is in lower case).

Syntax
Result (*as String*) = Titlecase(*Source as String*)

Example
```
Dim s As String
s=Titlecase("tHe Quick fOX") //returns "The Quick Fox"
s=Titlecase("THE LAZY DOG") //returns "The Lazy Dog"
```

Trim
Returns the string passed with leading (left side) and trailing (right side) whitespaces removed.

Syntax
Result (*as String*) = Trim(*Source as String*)

Example
```
Dim s as String
s=Trim("  Hello World  ") //Returns "Hello World"
```

Uppercase
Converts all characters in a string to uppercase characters.

Syntax
Result (*as String*) = Uppercase(*Source as String*)

Example
```
Dim s As String
s=Uppercase("tHe Quick fOX") //returns "THE QUICK FOX"
s=Uppercase("the 5 lazy dogs") //returns "THE 5 LAZY DOGS"
```

TIME FUNCTIONS

GetLongTime
Reports the current time in the user's "long time" format as a string based on the user's locale and formatting.

Syntax
Result (as integer) = GetLongTime

GetShortTime
Reports the current time in the user's "short time" format as a string based on the user's locale and formatting.

Syntax
Result (*as integer*) = GetShortTime

Microseconds
Returns the number of microseconds (1,000,000th of a second) that have passed since the user's computer was started.

Syntax
Result (*as Double*) = Microseconds

Ticks
Returns the number of ticks (60th of a second) that have passed since the user's computer was started.

Syntax
Result(*as Integer*) = Ticks

Appendix D: Scripting Function Index

H

I

L

M

N

O

P

R

S

T

U

V

W

Y

Appendix E: Choosing a Suggestion Generation Strategy

Lotto Sorcerer versions 6 and under did not give many choices in a suggestion strategy. Your only option was a three-sectored Pool Temperature: hot numbers, cold numbers and everything in between; and neural depth.

Lotto Sorcerer v9 gives you an abundance of choices.

You can choose:

- Starting and ending date of the analysis ("Scope").
- Neural/Analysis Depth (up to 256).
- Ten Neural /Analysis modes to choose from.
- Eight analysis methods to choose from.
- Multiple sectors (from 2 to 7).
- Sampling size.
- 17 Assertion filters.
- 15 Rejection filters.
- 10 Limitation filters.
- The ability to independently limit the limitation filters from 1 to 3 standard deviations.

Here is what we recommend on what settings to choose:

First, if at all possible, you should try to keep the sectors to the largest value that is an even divisor of the pool size. For example, if you lottery draws six numbers from 1 to 48, the recommend value for the sectors is "4", because 4 goes into 48 evenly (48 divided by 4 is 12). 5 does not: 48 divided by 5 is 9.6. Of course, with some lotteries, you cannot do this, like a lottery that draws from 1 to 29, where nothing goes into it evenly (other that 1 and 29).

Why is this important? This keeps all of the pools the same size. Otherwise, the pools will be unevenly sized, and this will skew towards the largest pool.

There is one downside to this: pick lotteries, which draws 10 numbers from 0 to 9 (like "Daily 3" and "Pick 4") means that you would choose a sector of 5 (because that goes into 10 evenly). These means each pool will only have two members to it (10 divided by 5 is 2). So this limits the number of suggestions you can play: Pick 3 is limited to 8 (2^3). Pick 4 is limited to 16 suggestions (2^4), and so on.

Next, experiment, experiment, experiment! If possible, run 100 or so suggestions on different settings (with all limitation filters off), and print out the suggestions. Make a note on the printout as to what settings generated those suggestions. When the lottery makes its draw, count which strategy had the most wins. This would be the strategy to try.

Try out the Virtual Lottery concept, if possible. Virtual lotteries let you join two lotteries together to be treated, and analyzed as, one lottery. For example, if you have a lottery that draws both at noon and in the evening, you can join those two lotteries together. Then compare results: compare your success rate with running the lotteries separately with running suggestions as a virtual lottery. You can setup a virtual lottery under the "Lottery Structure" menu.

Finally, don't forget the powerful tools (located in the Tools menu): *Lottery Number Oracle, Lotto Augur, Lotto Seer* and *Pick Lottery Frequency Distribution.* They can give you valuable insights into fine-tuning the number suggestion process.

Appendix F: Using the Help System

There are two ways of accessing the built-in Help System:

1. Choosing a Help topic from the menu; or
2. Clicking the Help icon (a white question mark within a blue circle), located at the bottom right of most windows.

Figure 109.

When the Help Window appears, you will see that it is divided into two sections. The left section shows the Help topic categories, and the right section shows the Help topic.

The higher level Help topic categories have a disclosure arrow (Mac OS X) or plus/minus signs (Windows). Clicking on these lets you collapse or reveal help topics within that category.

Appendix G: Using the Date Selector

The Date Selector is used in several functions to select the date. The appearance and function varies between Mac OS X and Windows.

In both operating systems, you can use the Tab key to move between the fields (month, day and year); once in a field, you can type directly to set the value.

Mac OS X Date Selector

The Mac OS X version has arrows to the right of the Date Selector. Use these arrows to increment or decrement the field you have selected.

Figure 110.

Windows Date Selector

The Windows Date Selector has a calendar icon
and arrow to the right of the Selector. Clicking this invokes the System Calendar as shown in Figure 72.

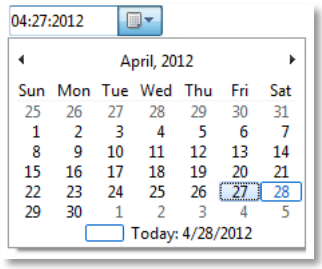


Figure 111.

The highlighted date is the selected date; the date with a border around it is the current date.

Appendix I: Using the Calendar Control

The Calendar Control is used in the Main Window.

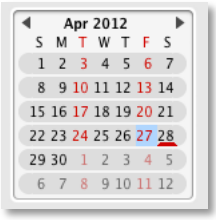


Figure 112.

Features

- The days of selected drawings are marked in a red type face.
- The day you have selected is in a blue background.
- The current day is underlined.
- Extraneous days are shown in ghosted text.

Changing the Date

- To change the year, click on the year; small arrows will appear, letting you increment or decrement the year.
- To change the month, click on the arrow at the top left to decrement the month, or click on the arrow at the top right to increment the month.

Modifying the Calendar

A couple of the features of the calendar can be modified in the Preferences window (see page 177):

1. Extraneous days can be made visible or invisible.
2. The first day of the week can be changed.

Appendix J: Using the System Clipboard

Lotto Sorcerer gives you the option to use the System Clipboard in many of its functions. The System Clipboard is built into the operating system, allowing you to send and receive data from different programs (or from within the same program).

There are three functions to the System Clipboard:

1. Cut: removes the source data, and send it to the Clipboard.
2. Copy: copies the source data to the Clipboard.
3. Paste: copies the source data from the Clipboard to the target. The source material data in the Clipboard.

Only one item can be in the Clipboard at a time. Cutting or copying to the Clipboard will overwrite any existing data.

Appendix K: Web Scraping

Web scraping is a unique feature of Lotto Sorcerer that can help you enter past drawings into Lotto Sorcerer.

The process consists of:

1. Copying the drawing data from the website by highlighting the text, and copying the text to the System Clipboard;
2. Clicking the "Scrape" button in Lotto Sorcerer. If successful, the drawing data will be inserted into the text boxes of the Main Window.

Notes

- This feature will work only with text data; it will not work with image data.
- Some web browsers present data in scrapable text format better than others. We highly recommend Mozilla's *Firefox* as an especially good at this.
- The Web Scraping feature is very forgiving, and will consider any non-numeric data as a separator.

Appendix L: Glossary

Arithmetic Mean: also known as the mean or average, is the central tendency of a collection of numbers taken as the sum of the numbers divided by the size of the collection.

Given the sample space $\{a_1, \cdots, a_n\}$:

$$A := \frac{1}{n}\sum_{i=1}^{n} \mathrm{a}_i$$

Comma Delimited: where a comma is the separator between values. For example, "3,5,13,19,22" ia s comma delimited string.

Harmonic Mean: also known as the subcontrary mean, is one of the three Pythagorean means. It is appropriate for situations when the average of rates is desired.

The harmonic mean H of the positive real numbers $\{x_1, x_2, \cdots x_n\}$:

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Median: the numerical value separating the higher half of a sample, a population, or a probability distribution, from the lower half.

Population (Standard Deviation): the variation or "dispersion" from the average.

$$\sigma = \sqrt{\frac{\Sigma_{i=1}^{n} a_i^2}{n} - \left(\frac{\Sigma_{i=1}^{n} a_i}{n}\right)^2}$$

Range: the length of the smallest interval that contains all the data.

Real Lottery: a lottery that is not a virtual lottery; a discrete lottery. Virtual lotteries are made up of real lotteries. Most of the dropdown menus within Lotto Sorcerer will show only real lotteries.

System Clipboard: a reserved section of computer memory that is used as a temporary, behind-the-scenes staging area for data that is copied (using copy and paste) or moved (using cut and paste) from one application to another. Each time data are transferred into the clipboard, the previous contents of the clipboard is deleted.

Variance (Standard Deviation): this value is calculated similarly to the "Population (Standard Deviation)", but the sigma is not rooted:

$$\sigma = \frac{\Sigma_{i=1}^{n} a_i^2}{n} - \left(\frac{\Sigma_{i=1}^{n} a_i}{n}\right)^2$$

Variance (Standard Population Deviation): this value is calculated similarly to the "Variance (Standard Deviation)", but the sample size is incremented:

$$\sigma = \frac{\Sigma_{i=1}^{n} a_i^2}{n+1} - \left(\frac{\Sigma_{i=1}^{n} a_i}{n+1}\right)^2$$

Virtual Lottery: a lottery that is comprised of at least two real lotteries. For example, a midday Pick 3 and evening Pick 3 lottery can be joined together as one virtual lottery.

Winsorized Mean: a measure of central tendency, involving calculating the mean after replacing given parts of a sample at the high and low end with the most extreme remaining values.

Appendix M: Database Schema

LOTDEF Table

This is the table that Lotto Sorcerer uses to create lotteries that you have selected by using the Lottery Setup Wizard. It is not intended to be modified by the user.

FIELD	TYPE	KEY	NOTES
LOTID	CHAR	Primary	
LOTTERYTABLE	CHAR		Name of table
LOTTERYNAME	CHAR	Normal	Name of lottery
LOTTERYTYPE	INT		0 = standard lotto
			1 = lotto + 1 bonus number from 1 pool
			2 = lotto + 2 bonus numbers from 1 pool
			3 = pick 3 type lottery
			4 = pick 4 type lottery
			5 = lotto + 2 bonus numbers from 2 pools
			6 = keno type lottery
			7 = lotto + 1 extra number
			8 = lotto + 2 extra numbers
			9 = lotto + 3 extra numbers
			10 = pick 5 type lottery
			11 = pick 6 type lottery
			12 = pick 7 type lottery
			13 = pick 8 type lottery
			14 = lotto + 4 extra numbers
			15 = pick 2 type lottery
			16 = pick 1 type lottery
DRAWINGDAYS	INT		Bitwise map (7 bits)
DRAWTIME	CHAR		Drawing time (24 hr clock)
MINPOOLNUMBER	INT		Minimum pool number
MAXPOOLNUMBER	INT		Maximum pool number
MINBONUSPOOLNUMBER1	INT		Minimum bonus number 1
MAXBONUSPOOLNUMBER1	INT		Maximum bonus number 1
MINBONUSPOOLNUMBER2	INT		Minimum bonus number 2
MAXBONUSPOOLNUMBER2	INT		Maximum bonus number 2
NUMBERSDRAWN	INT		Numbers drawn (incl. bonus/extra numbers)
NUMBERSPLAYED	INT		Numbers played (incl. bonus numbers)
UPDATED	INT		1 = updateable via subscription service
			0 = not updateable via subscription service
STATE	CHAR	Normal	Name of state/country of lottery
URL	CHAR		Website of lottery
FIELDSIZE	INT		Size of drawing field

LOTTERIES Table

This is the dynamic table that contains how each of your lotteries is setup, plus in-work settings (filters used, engine settings, playslip nudge settings, virtual lottery settings, etc.)

FIELD	TYPE	KEY	NOTES
LOTID	CHAR	Unique	
LOTTERYTABLE	CHAR	Primary	Name of table
LOTTERYNAME	CHAR	Normal	Name of lottery
TABLETYPE	CHAR	Normal	D = built-in
			L = custom
			V = virtual
LOTTERYTYPE	INT	Normal	0 = standard lotto
			1 = lotto + 1 bonus number from 1 pool
			2 = lotto + 2 bonus numbers from 1 pool
			3 = pick 3 type lottery
			4 = pick 4 type lottery
			5 = lotto + 2 bonus numbers from 2 pools
			6 = keno type lottery
			7 = lotto + 1 extra number
			8 = lotto + 2 extra numbers
			9 = lotto + 3 extra numbers
			10 = pick 5 type lottery
			11 = pick 6 type lottery
			12 = pick 7 type lottery
			13 = pick 8 type lottery
			14 = lotto + 4 extra numbers
			15 = pick 2 type lottery
			16 = pick 1 type lottery
VIRTUALMEMBER	CHAR	Normal	LOTTERYTABLE tied to this virtual lottery
DRAWINGDAYS	INT		Bitwise map (7 bits)
DRAWTIME	CHAR		Drawing time (24 hr clock)
MINPOOLNUMBER	INT		Minimum pool number
MAXPOOLNUMBER	INT		Maximum pool number
MINBONUSPOOLNUMBER1	INT		Minimum bonus number 1
MAXBONUSPOOLNUMBER1	INT		Maximum bonus number 1
MINBONUSPOOLNUMBER2	INT		Minimum bonus number 2
MAXBONUSPOOLNUMBER2	INT		Maximum bonus number 2
NUMBERSDRAWN	INT		Numbers drawn (incl. bonus/extra numbers)
NUMBERSPLAYED	INT		Numbers played (incl. bonus numbers)
UPDATED	INT		1 = updateable via subscription service
			0 = not updateable via subscription service
STATE	CHAR		Name of state/country of lottery
URL	CHAR		Website of lottery
FIELDSIZE	INT		Size of drawing field
CHECKNUMBERS	CHAR		Used by Check Numbers function
ASSERTCALC	CHAR		Used by the Assert Calculation function
NUDGE	CHAR		Used by playslip settings
REJ	CHAR		Used by User-defined rejection filter
FLAG1	INT		Reserved for future use
FLAG2	INT		Reserved for future use
FLAG3	INT		Reserved for future use
FLAG4	CHAR		Reserved for future use
FLAG5	CHAR		Reserved for future use
FLAG6	CHAR		Reserved for future use

WHEELS Table

This table is used to maintain Lotto Sorcerer's wheels.

FIELD	TYPE	KEY	NOTES
WHEELNAME	CHAR	Primary	Name of wheel
WHEELDESC	CHAR		Description of wheel
NUMBERSDRAWN	INT	Normal	NUMBERS PER TICKET (k)
NUMBERSPERWHEEL	INT	Normal	Numbers in the wheel (v)
NUMBEROFTICKETS	INT	Normal	Number of wheels in file
NUMBERSMATCH	INT	Normal	How many numbers match… (t)
IFNUMBERSDRAWN	INT		…if numbers drawn
WHEELLOCATION	CHAR		PROG = Program location
			DOCS = Documents file

Appendix N: Differences Between the Evaluation Version and the Registered Version

The Evaluation Version has four limitations:

1. The Evaluation Version is limited to a Neural/Analysis Depth of eight.
2. The Evaluation Version is limited to 12 uses.
3. All export functions in the Evaluation version are limited to exporting 50 records.
4. Lotto Scribe prints "DEMO" on the playslips in the Evaluation Version.

The Registered Versions do not have these limitations.

Appendix O: Concerning Microsoft Excel Compatibility

All of the functions within Lotto Sorcerer v9 that export to Microsoft Excel are compatible with Microsoft Excel 2002 (or higher) or OpenOffice 4 (or higher).

Appendix P: Gamble Responsibly

If you choose to gamble, we urge you to:

- Gamble responsibly.
- Play within your means.
- Gamble with no more than your discretionary income.
- Don't "chase your losses".
- Set your play limit and stick to it.

Symptoms of Gambling Addiction

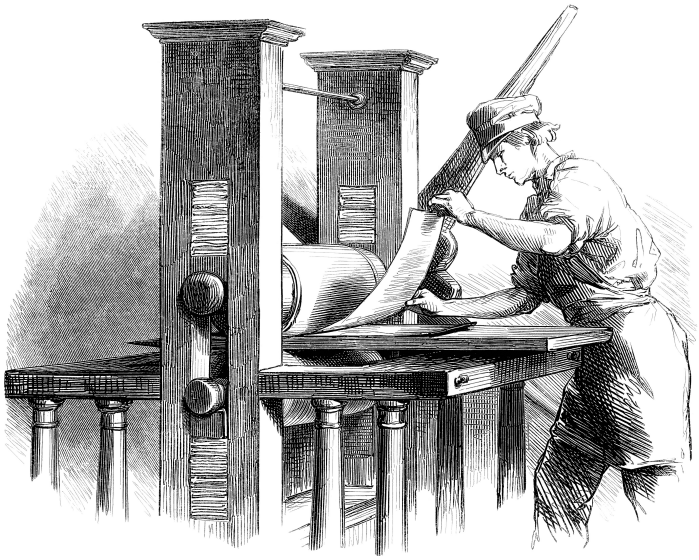
1. Regret over the amount of money or time spent on gambling.
2. Spending more than you can afford on gambling.
3. Spending money on gambling while financial responsibilities are neglected (neglect of paying bills, rent, etc.).
4. Spending money on gambling while personal responsibilities are neglected (neglect of family, pets, charities, commitments, etc.).
5. "Cutting back" on a necessity (food, medicine, etc.) in order to pay for gambling.
6. Covering up or lying about the amount of money spent on gambling.
7. Purchasing lottery tickets at multiple locations to allay suspicions of sales clerk that you may have a gambling problem.
8. Having a compulsion to purchase tickets regularly; to never miss a drawing of your favorite lottery.
9. Breaking the law in order to get gambling money or recover gambling losses (stealing, fraud, etc.).
10. Asking for financial assistance as a result of gambling.
11. Continued gambling despite negative consequences: loss of job, relationships or opportunities.
12. Denial of a gambling problem or lying to friends or family about behavior.
13. Rationalizing one's gambling, or believing that this list of *Symptoms of Gambling Addiction* does not apply to you. Remember, "rationalize" means "rational lies".

Gambling addiction is a continuum, from "at-risk gambler" to "problem gambler" to "compulsive gambler". Matching even one symptom, listed above, is a red flag. The more symptoms you match, the more addicted you may be.

Finis

This Document has been Produced by:

Satori Publishing
In-House Document Division
Print Shop



Document number: LSV900.R1
April 7, 2018