

SPAN Risk Manager Functions

This section describes the core processing functions of the SPAN Risk Manager. All such functions can be executed from the GUI, from the script language (via "spanitrm.exe"), and from the real-time component interface.

Note

SPAN Risk Manager functionality is a superset of that provided in PC-SPAN, so that any function available in PC-SPAN is also available in SPAN Risk Manager.

- **Processing Functions**
 - Calculate Option Implied Volatilities
 - Average Call / Put Volatilities
 - Calculate Series Volatilities
 - Calculate Option Prices (and Greeks)
 - Calculate Risk Arrays
 - Replicate Price Basis
 - Apply Volatility Skew
 - Observe Market
 - Reset Prices
- **What-If Scenario Processing**
 - Define What-If Scenario
 - Compare Scenarios
- **Typical Uses of SPAN Risk Manager**
 - Calculation of SPAN Risk Arrays, Creation of SPAN Risk Parameter Files
 - Stress Testing
 - Real-Time Pre- or Post-Execution Risk-Based Credit Controls
- **Trade Posting**
 - Update Positions Only
 - Trade-register mode
 - Bookkeeping mode

Processing Functions

Calculate Option Implied Volatilities

This function may be performed for an entire point in time, or for any of several different levels defined within a point in time: an exchange complex, a combined commodity, an option product family, an option series, or a specific option.

For each option, an attempt is made to calculate the implied volatility. If successful, the volatility for the option is set to the new value.

Average Call / Put Volatilities

This function likewise may be performed for an entire point in time, or for any of several different levels defined within a point in time: an exchange complex, a combined commodity, an option product family, an option series, or a specific option.

The function searches for call / put pairs -- *ie*, options which are identical except that one is a call and the other is a put. For each such pair:

- If the volatility is defined for both the call and the put, then the average of these two values is taken, and both the call volatility and the put volatility are re-assigned as that average.
- If the volatility is defined for the call but not for the put, then the put volatility is assigned the value of the call volatility.
- If the volatility is defined for the put but not for the call, then the call volatility is assigned the value of the put volatility.

This function may be used if it is desired to ensure that corresponding calls and puts have the same value assigned for their volatility.

Calculate Series Volatilities

This function may be performed for an entire point in time, or for an exchange complex or specific combined commodity within a point in time. The lowest level to which this function is applicable is the combined commodity. (In other words, you can't calculate a series-average volatility just for one particular series, but have to at least do it for all series within a combined commodity.)

The function assigns a value to the volatility for the entire option series, for each option series to which it is applicable, and it guarantees that such "series-average" volatilities will be assigned. The algorithm calculates implied volatilities for around-the-money options and then assigns the series volatility as the average of those strike-level volatilities for the around-the-money options.

Calculate Option Prices (and Greeks)

This function may be performed for an entire point in time, or for any of several different levels defined within a point in time: an exchange complex, a combined commodity, an option product family, an option series, or a specific option.

For selected options, the function calculates either:

- option price and delta, or
- option price, delta, vega, gamma, theta, rho

In doing so, you can choose which volatilities to use, either:

- option strike-level volatilities only, or
- option series-level volatilities only, or
- option strike-level volatilities if they are defined, or series-level volatilities if a strike-level volatility is not defined.

This last alternative is useful if it is desired to use strike-level volatilities, but if it is also possible, given the input data, that strikes may exist for which volatilities are not defined or cannot be calculated. In this case, since the algorithm for determining a series-average volatility guarantees that one will be assigned, using strike-level volatilities if they are available and series-level volatilities if not, guarantees that option prices are calculable for all options.

When calculating option prices, you can also choose to do this with "with reset" or "without reset". If you are doing this "with reset", then any previously defined option prices will be overwritten by the newly calculated prices. If you are calculating "without reset", then option prices (and greeks) will not be calculated for any option for which a price is already defined.

The "without reset" capability is useful if you have valid market prices for some options and want to calculate theoretical prices only for the options for which market prices are not available.

Calculate Risk Arrays

This function may be performed for an entire point in time, or for any of several different levels defined within a point in time: an exchange complex, a combined commodity, a product family, an option series, or a specific contract.

For selected contract(s) this function calculates SPAN risk arrays for each SPAN rate for which price scan range and volatility scan range are defined.

In doing so, you can choose which volatilities to use (affects only option risk arrays), either:

- option strike-level volatilities only, or
- option series-level volatilities only, or
- option strike-level volatilities if they are defined, or series-level volatilities if a strike-level volatility is not defined.

As with the calculation of option prices, this last alternative is useful if it is desired to use strike-level volatilities, but if it is also possible, given the input data, that strikes may exist for which volatilities are not defined or cannot be calculated. In this case, since the algorithm for determining a series-average volatility guarantees that one will be assigned, using strike-level volatilities if they are available and series-level volatilities if not, guarantees that risk arrays are calculable for all options.

Replicate Price Basis

This function may be performed for an entire point in time, an exchange complex, an exchange, a combined commodity, or a particular non-option product family.

For futures product families (and any other non-option product type) defined at a point in time where prices are available for some contracts but not for others, this function assigns theoretical prices to those other contracts. It does so by replicating the basis differentials for this product family for the immediately prior point in time.

For example, suppose at the previous point in time the price of the lead future was X and the price of the second future was Y. Suppose further that for the current point in time the price of the lead future was assigned as A and the price of the second future has not been defined. This function would assign the price of the second future as $A + (Y - X)$.

Apply Volatility Skew

This function may be performed for an entire point in time, an exchange complex, an exchange, a combined commodity, or a particular option product family.

For each option series defined at a point in time where volatilities are available for some options but not others, this function assigns volatilities to those other options in the series. It does so by replicating the volatility skew relationship from the same series at the immediately prior point in time.

In doing this, it applies both a **horizontal shift** and a **vertical shift**: The best way to understand this is to think of a graph of strike price (on the X-axis) and volatility (on the Y-axis):

- The **horizontal shift** moves the volatility curve left or right according to how the underlying price has changed from the previous point in time to the current point in time -- ie, the price at which the at-the-money strike is defined. If the underlying price increases, for example, the curve is shifted to the right.

- The **vertical shift** moves the volatility curve up or down according to how implied volatilities for the "reference option(s)" -- the option(s) that are closest to the money -- at the new point in time have changed from the analogous volatilities at the previous point in time.

There are three different alternatives for applying the vertical shift:

- **Use separate call and put reference volatilities:** select the closest-to-the-money call at the new point in time which has a volatility defined for it. If an exact at-the-money option exists and has a volatility defined, this is the selected one. If not, then select the call which is closest-to-the-money. (If an out-of-the-money call and an in-the-money call are equidistant-from-the-money, then select the out-of-the-money one.) Use this volatility as the reference for assigning volatilities to all other calls for which volatilities are not already defined.

Separately, repeat this logic to select the closest-to-the-money put at the new point in time which has a volatility defined for it, and use this volatility as the reference for assigning volatilities to all other puts for which volatilities are not already defined.

- **Use best reference volatility:** Use the logic described above to select the closest-to-the-money call for which a volatility is defined, and the closest-to-the-money put for which a volatility is defined. If these two strikes are equidistant-from-the-money, take the average of their volatilities as the reference volatility. Otherwise, select the strike which is closest to the money, and use its volatility as the reference volatility.

This single value for best reference volatility is then used to assign volatilities to all options, calls and puts, at the new point in time for which volatilities are not already assigned.

- **No reference:** the volatility curve is shifted horizontally but not vertically.

Observe Market

This function may be performed for an entire point in time, an exchange complex, an exchange, a combined commodity, or a particular product family.

For all contracts for which the function is performed, it does the following:

- Calculates the mark-to-market on the net position quantity from the previous value for contract price to the current value for contract price.
- Takes the sum of three values: (a) the mark-to-market amount just calculated, (b) the previous value for mark-to-market on the position stored with the position, and (c) the value for mark-to-market on trades posted since the last market observation.
- Stores this sum as the new value for mark-to-market on the position.
- Resets the value for mark-to-market on trades posted since the last market observation to zero.
- Stores the current value for contract price as the previous value for contract price.

It calculates Mark to Market amounts for all portfolios defined for the point in time. Portfolios are marked to the latest prices set for the point in time.

Reset Prices

This function may be performed for an entire point in time, an exchange complex, an exchange, a combined commodity, or a particular product family.

For all contracts for which the function is performed, it takes the current price and stores it as the "previous" price for the contract. In effect, it establishes the current price as the baseline for subsequent mark-to-market processing -- either trade posting or market observations.

What-If Scenario Processing

The SPAN Risk Manager provides a rich set of capabilities for evaluating the impact of what-if scenarios, a function also known as **stress testing**. This works as follows:

- A point in time is selected -- the **base** point in time.
- A new point in time is created -- the **what-if** point in time -- by cloning the base point in time, and then applying a defined **what-if scenario** to it. The what-if scenario allows any or all of the following to be varied: price, volatility, price scan range, volatility scan range, time to expiration, risk-free interest rate, and dividend yield.
- After the application of the what-if scenario to the what-if point in time, theoretical prices and SPAN risk arrays are calculated for it.
- Performance bond requirements and portfolio values are then calculated for all portfolios for both the base point in time and the what-if point in time.
- For each portfolio, the **impact** of applying the what-if scenario is then calculated as:
 - the amount by which the portfolio value has increased, plus
 - the amount by which the SPAN risk has decreased.

Thus, a **positive impact** means that the portfolio gained in value and/or had its risk decrease. Conversely, a **negative impact** means that the portfolio lost value and/or had its risk increase.

Define What-If Scenario

This function is always performed for an entire point in time. It is used to create a new point in time, identical to the original ("base") one except that:

- it has a descriptor to identify it and distinguish it from the original point in time -- for example, "S&P's up 20%"
- any of the following parameters have been changed: price, volatility, price scan range, volatility scan range, time to expiration, risk-free interest rate, and/or dividend yield.

The specification for how these parameters should be changed, and to which contracts the changes should be applied, constitute a **what-if scenario**.

You start by pointing to the original, base point in time, right-mouse click, and select **What-If Scenario**.

The **Create What-If Scenario** dialog box displays, prompting you to enter a descriptor for the what-if scenario. You enter the descriptor and click on **Create New**.

The full **Define What-If Scenario** dialog box now displays. This allows the specification of the changes to be applied to the above parameters:

- You specify which parameter is to be changed -- price, volatility, price scan range, volatility scan range, time to expiration, risk-free interest rate, and/or dividend yield.
- For each such parameter, you specify how it is to be changed by selecting one of the three options, and then specify the value:
 - **Set** -- sets the parameter to the specified value;
 - **Change** -- changes the parameter by the specified amount. (A negative amount decreases the parameter.)
 - **Change Percent** -- changes the parameter by the specified percentage. (A negative value decreases the parameter by the specified percentage.)
- By expanding or contracting nodes on the tree in the dialog box, you can specify the data elements to which you want the changes to be applied:
 - to the entire newly-created point in time
 - to an exchange complex within that point in time
 - to a combined commodity within that exchange complex, or
 - to a particular product family linked into that combined commodity.

After the desired what-if parameters have been defined, you can click on either **OK** or **Calculate**:

- **OK** causes the new point in time to be created and the what-if scenarios applied to it.
- **Calculate** does the following:
 - Creates the new point in time and applies the what-if scenarios to it.
 - Calculates option theoretical prices and SPAN risk arrays for the new point in time. Strike-level volatilities are used if they exist and series-level volatilities if not. Option prices are calculated "with reset" -- *ie*, prices are calculated for all options regardless of whether prices are already defined for them.
 - Calculates performance bond requirements for all portfolios at both the original ("base") point in time and the newly-created point in time.
 - Displays the **Compare Scenarios** dialog box, displaying the impact of the what-if scenario on the portfolios.

Note that you can also choose to define and apply Risk Scenarios to an existing point in time rather than creating a new one:

- Point to the desired point-in-time. Right-mouse click on it, select **What-If Scenario**, then click on **Modify**.

Compare Scenarios

This function is likewise always performed for a particular point in time. It allows the comparison of the impacts associated with any particular what-if scenario. To use it:

- Point to the what-if point in time.
- Right-mouse click, then select **Compare Scenarios**.
- The **Select Point in Time** dialog box pops up. Select the "base" point in time, then click on **OK**.
- The **Compare Scenarios** dialog box displays.

Typical Uses of SPAN Risk Manager

Calculation of SPAN Risk Arrays, Creation of SPAN Risk Parameter Files

An exchange or a clearing organization implementing SPAN needs a way to calculate SPAN risk arrays for all of its products and publish at least one daily SPAN risk parameter file. A typical set of operations would be:

- Via **spanitrm** and a script file:
 - Load the "null" risk parameter file, in which SPAN risk arrays and composite delta values are all zero (because they have not yet been calculated.)
 - Calculate series-average volatilities for all option series.
 - Calculate individual strike-level implied volatilities for all options.
 - Average corresponding call and put volatilities.
 - Calculate risk arrays for all contracts using strike-level volatilities if they are defined and series-level volatilities if not.
 - Save the SPAN document, thereby creating the XML-based SPAN risk parameter file.
- If desired, use the **x2p** conversion utility to create the SPAN risk parameter file in the expanded-unpacked or standard-unpacked format.

Stress Testing

Evaluating the impact on a set of portfolios of various hypothetical market occurrences over the next trading day is always an important requirement. Such "stress testing" allows the impact of common hypothetical market occurrences to be determined in advance and thereby allows rapid response in the event any such scenario occurs.

To do this in an automated way:

- Via **spanitrm** and a script file:
 - Create and save the "base" SPAN risk parameter file.
 - Load portfolios, calculate margins and portfolio values, save the results to a "base case" SPAN document file.
 - For each "scenario set" for which it is desired to perform stress testing:
 - From the base point in time, create a new point-in-time with a descriptor identifying the scenario set.
 - Apply the changes to prices, volatilities, price and volatility scan ranges, interest rates, and time to expiration associated with the scenario set.
 - Apply the volatility skew with "no reference" to shift the volatilities horizontally for each option series according to the move in the underlying price.
 - Recalculate theoretical prices for all options.
 - Recalculate risk arrays for all contracts.
 - Recalculate margin requirements and portfolio values for all portfolios.
 - Save the results to an overall SPAN document file.
- Open the resulting document file in SPAN Risk Manager and view impacts.

Real-Time Pre- or Post-Execution Risk-Based Credit Controls

If an order to execute a trade has been placed, it is often desirable to evaluate, in real-time, the impact of such a trade on the portfolio risk, margin requirement, and value. Based upon this impact, again in real time, the order may be accepted or rejected. Even if such "pre-execution" credit controls are not implemented, in many cases it will be extremely valuable to implement such real-time checking post-execution. One of SPAN Risk Manager's most important capabilities is the ability to implement this.

Typically this would be implemented via a multi-platform strategy -- one platform in order to update the risk profile of the various products in a "continuous batch" process, and another platform or set of platforms in order to apply the results to portfolios in real time as orders are provided and/or trades are executed.

On one platform for the updating the risk profile:

- Via either **spanitrm** or the real-time component interface for the SPAN Risk Manager:
 - Load the SPAN risk parameter file for the immediately previous point in time.
 - Clone the data for the previous point in time to a new point in time.
 - Update market prices for the most actively trading futures, physicals and options.

- Execute the "Replicate Price Basis" function in order to assign theoretical prices to futures for which current market prices were not available.
- Calculate implied volatilities for the actively trading options for which market prices were available.
- Execute the "Apply Volatility Skew" function in order to assign volatilities for all other options, taking into account both the implied volatilities for the options with current market prices, and the volatility skew from the previous point in time.
- Calculate theoretical prices for all options except the ones for which current market prices were available.
- Recalculate risk arrays for all instruments.
- Clear the data for the previous point in time from memory.
- Save the results to a SPAN document file for the new point in time.

Total time required to execute such a cycle will depend on the number of instruments, the pricing models used, and the overall speed and memory capacity of the execution platform. In many or most cases, however, total execution time will be measured in seconds. In other words, execution time is not the limiting factor and this part of the process can be executed as often as desired during the trading day. As often as every minute or two, or as infrequently as several times per day, the process can be executed in order to obtain an updated SPAN document file reflecting current market conditions. The more volatile the market, and the more it is desired to have the process reflect current market conditions, the more important it is to increase the frequency of such updates.

The process of applying the orders or trades to the SPAN document reflecting updated market conditions would typically be executed in real-time, via the real-time component interface, on one or more separate execution platforms:

- Via the real-time component interface:
 - Open the SPAN document file for the latest point in time.
 - Load portfolios current up to the latest point in time.
 - As orders are placed and/or trades are executed:
 - Post the trade, thereby updating the position quantities.
 - Calculate margin requirement and portfolio value.
 - Get the updated margin requirement and portfolio value. Take appropriate action based upon how the trade or order has caused the risk or the portfolio value to change.
 - When a new SPAN document file is available reflecting updated market conditions, pause the real-time trade poster, and re-initiate the process with that new SPAN document file.

Trade Posting

The SPAN Risk Manager software will very shortly add full "trade posting" capability, providing new position management and bookkeeping functions.

Trades may be posted by loading a file of trades, from the GUI, the batch interface, or the real-time component interface. Alternatively, trades may be posted one-by-one through the component interface.

Trade posting may be performed in any of three modes:

- **Update positions only**
- **Trade-register mode**
- **Bookkeeping mode**

Update Positions Only

When trade posting is performed in this mode, as each trade is posted, the following is done:

- The position quantity is updated
- The mark-to-market on the trade is calculated from trade price to current market price, and:
 - For the position, the mark-to-market amount on trades posted since the last market observation is incremented by this result.
 - For the product family, the overall amount for mark-to-market amount on trades posted since the last observation is similarly incremented by this result.

- At higher levels (the combined commodity, exchange complex, and portfolio as a whole), the mark-to-market amount on trades posted since the last observation is similarly incremented, with the caveat that separate buckets are maintained for mark-to-market on trades for futures-style products, and for non-futures-style products.
- If the product is a premium-style product (for example, a premium-style option, or an equity or debt security), the option premium (ie, price obligation) is calculated for the trade, and:
 - For the position, the option premium (accumulated price obligation) field is incremented for this result.
 - For the product family, the option premium (accumulated price obligation) field is similarly incremented for this result.
 - At higher levels (the combined commodity, exchange complex, and portfolio as a whole), the accumulated premium (price obligation) is similarly incremented, with the caveat that separate buckets are maintained for accumulated premium on premium-style options, and accumulated price obligation for non-options such as equity and debt securities and other physicals.

In other words, aggregate mark-to-market amounts are broken out into separate fields for true settlement variation which will be banked for futures-style products, and mark-to-market amount which is provided for information for non-futures-style products. Premium (price obligation) is calculated for any product which is not valued futures-style, and is aggregated separately for premium-style options and other premium (equity) style products.

Typical usage for real-time risk intraday real-time risk management would be as follows:

- Load the end-of-day SPAN file for the previous business day.
- Load the beginning of day positions.
- As trades occur, in real time, post them, causing the positions, the mark-to-market amount on trades, and the accumulated premium (price obligation) to be updated.
- As often as desired, bring in an updated set of market prices, and perform a market observation. This causes the positions to be marked-to-market from previous market price to current market price, increments the updated mark-to-market amount on the position by the amount on trades posted since the last market observation, and then sets the mark-to-market amounts on such trades to zero.
- Monitor the accumulated mark-to-market and price obligation fields in real-time, taking action as needed.

Trade-register mode

In addition to the functions performed in "Update positions only" mode, Trade Register mode saves, for each posted trade, the mark-to-market amount and, if it is for a premium-style product, the price obligation. This data is saved to a separate Trade Register file, from which various Trade Register reports and datafiles can be prepared.

Typical end-of-day usage would be as follows:

- Produce the end-of-day SPAN risk parameter file.
- Produce a trade file consisting of:
 - beginning of day positions, with trade price specified as end-of-day settlement prices for the previous day, for futures-style products, and as zero for premium-style products
 - all trades for the day just ended.
- Post the trades from that trade file, specifying that a Trade Register file be created.
- From the resulting Trade Register file, produce reports showing:
 - for futures-style positions:
 - the beginning of day position and its mark-to-market from previous-day's settlement price to current day's settlement price
 - for each trade, the mark-to-market from trade price to current day's settlement price.
 - for premium-style positions:
 - the beginning of day position
 - for each trade, the premium (price) obligation.

Bookkeeping mode

In addition to the functions performed in "Update positions only" mode, Bookkeeping mode tracks individual Open Trades behind each net position and does FIFO position offsetting and profit & loss accounting. The result are datafiles from which basic bookkeeping reports can be produced.

The SPAN document file itself in this case stores the individual Open Trades behind each net position. For each open trade, the following is stored:

- the original date on which the Open Trade was created
- the original open quantity and original trade price
- the currently open quantity, ie, the remaining open quantity
- the unrealized P&L -- *ie*, the accumulated profit or loss on the remaining open quantity
- the realized P&L -- *ie*, the accumulated profit or loss on the closed quantity.

When the open trade has been fully liquidated, the currently open quantity will be zero as will the unrealized P&L. All P&L will have been realized.

At any instant in time, then, there will be one or more open trades behind each net position. The open quantities on each open trade with nonzero value for currently open quantity, must of necessity be of the same sign as that of the current value for net position.

For example, suppose on day one you start out flat in a particular instrument and post two buy trades for quantities of 10 at different prices. The result is a net position of +20, with two open trades behind that position, each for an original quantity and a currently open quantity of +10.

Suppose on the next day you clear a sell trade for a quantity of -30. The first -10 will offset the first open trade, causing its open quantity to be set to zero and all P&L on the open trade to be realized. In effect, it is a "closed trade". The second -10 will similarly offset the second open trade. The last -10 will establish a new open trade, at its trade price, with an open quantity of -10, and the mark-to-market from trade price to end of day market price stored as unrealized P&L. The overall net position will be -10.

In addition to the updating of the net positions and of the open trade(s) behind the net positions, bookkeeping mode causes a Bookkeeping Register datafile to be produced. This lists, for each trade posted, what occurred:

- If the trade quantity is on the same side of the market as the current net position, then the trade establishes a new Open Trade behind the position.
- If the trade quantity is on the opposite side of the market as the current net position, then:
 - the trade posting will cause one or more Open Trades to be partially or completely liquidated, bringing the currently open quantity back towards zero and causing unrealized P&L to be updated and transferred to realized P&L.
 - if the absolute value of the trade quantity was greater than the absolute value of the net position, then all previously open Open Trades will be complete liquidated, and a new Open Trade will be established on the opposite side of the market for the excess.

From the SPAN document file and the Bookkeeping Register datafile, bookkeeping reports can be produced.