TRADING TECHNOLOGIES

Updating the ADL Stopwatch Block

VERSION 7.X

LEGAL

Updating ADL™ Stopwatch Block

If you have an ADL algorithm that contains a Stopwatch Block, TT strongly recommends replacing the block with a newer version provided in X_TRADER® 7.17.0 or higher.

The following update method is recommended to minimize disruptions in trading activity:

1. Save backup copies of the algorithms that contain Stopwatch Block(s).
2. Install or upgrade your current X_TRADER version to 7.17.0 or higher.
3. Open the algorithms using ADL Designer Canvas and replace the outdated Stopwatch Blocks (they should display the text "Obsolete" on the surface) with new counterparts from the Toolbar Panel. Minor redesign is expected - please see the section "**Redesign Examples**" for some examples of such redesign.

Warning: Once an algorithm has been updated in this manner, that algorithm can no longer be deployed to Algo SE earlier than 7.17.0.
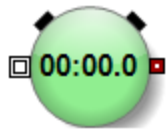
New Stopwatch Block Design

The Stopwatch Block has been enhanced for greater flexibility and efficiency, and it will operate in the following manner:

1. When the block receives a discrete event message, it will evaluate the "Stop Time" formula (depicted in its internal green canvas) to get the stop time in milliseconds. The new Stopwatch Block can achieve a maximum precision of approximately 16 ms depending on the circumstances.
2. After the specified stop time, the block will output a discrete event message.
3. The block must be reset with a new incoming discrete event message if repetition is needed.
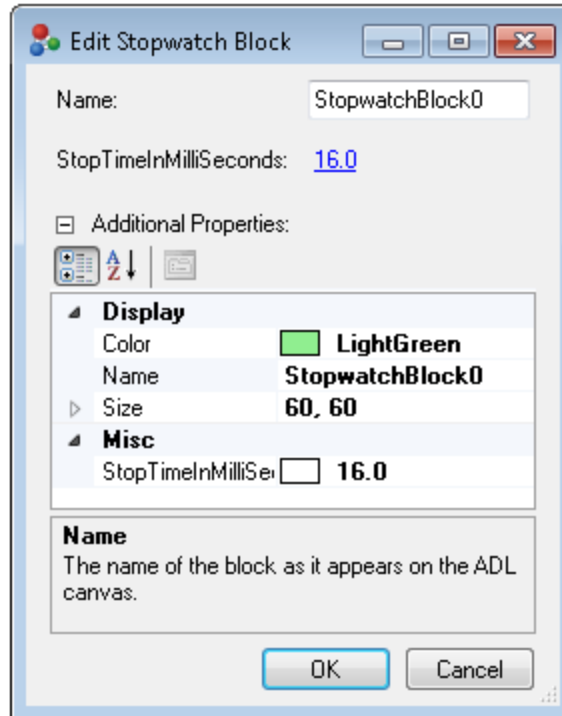
The new design confers the following benefits:

● Enables the user to set a custom timer interval instead of a fixed interval of 250 ms.
● Enables the user to set a dynamic timer interval.
● Enables the user to decide when/how to repeat the timer, thereby preventing unnecessary timer events.

Previous Design.

New Design.
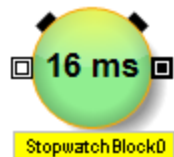
Redesign Examples

Below are some examples of replacing an outdated Stopwatch Block with the newer version provided in X_TRADER 7.17.0 (or higher).

Example 1. Using a timer to scratch a trade

The image <Before Update> displays the cover logic of a strategy that is activated by an entry fill. For 10 seconds following the entry fill, the cover order rests at a "Profit Price" which is one tick higher than the entry fill price. After 10 seconds, the cover order moves down to the entry fill price in an attempt to scratch the trade.

Before redesigning this logic using the newer version of the Stopwatch Block, it is important to note that:

- The Stopwatch Block cannot be reset (the cover logic displayed below is virtualized, which means that each fill event has an independent instance of the cover logic).
- The Stopwatch Block does not need to "tick" indefinitely - it only needs to "tick" once at the 10-second mark.

The image <After Update> shows the new Stopwatch Block set to generate an event at the 10-second mark (10,000 ms). The event is then used to change the output value of the Value Accumulator Block labeled "Flag" from 0 to 1, which in turn updates the output value of the "Work for Profit?" block from TRUE to FALSE. Blocks downstream from that point do not need to be redesigned.

Before Update
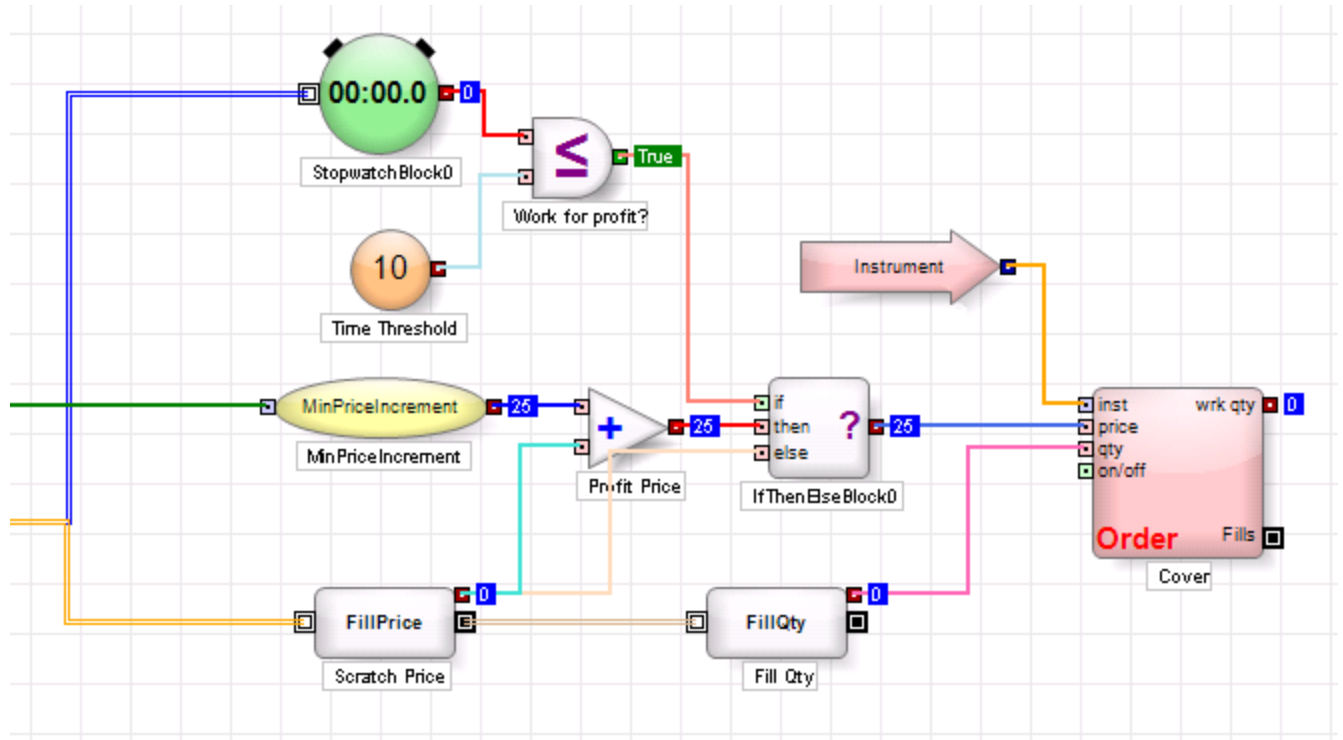
After Update
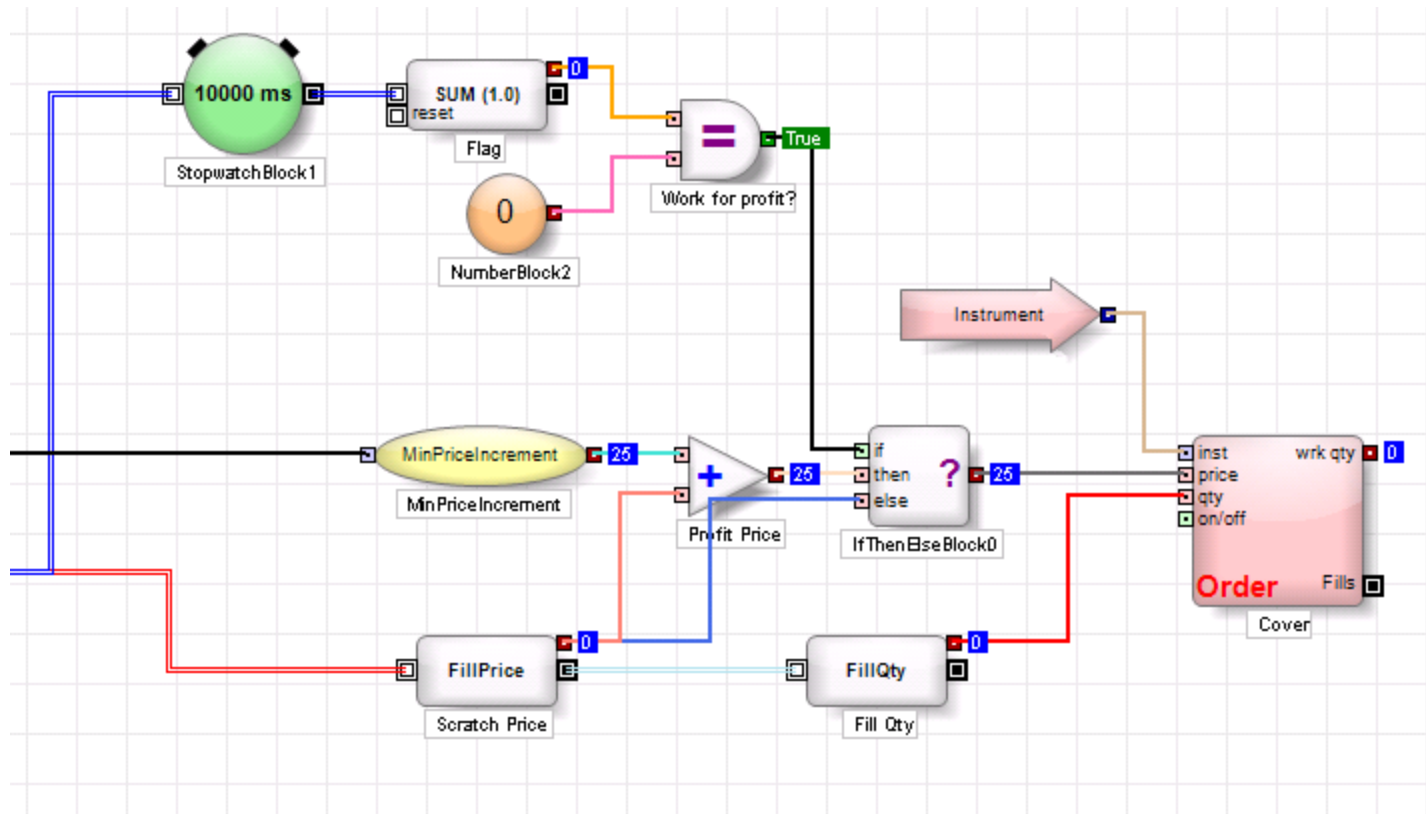
Example 2. Using a timer to progressively reprice a cover order

The image <Before Update> displays an enhanced version of the price determination portion from Example 1. The cover order is *progressively* repriced closer to the scratch price as more time elapses. For 10 seconds following the entry fill, the cover order rests at a "Tier 1 Price" which is 2 ticks higher than the entry fill price. After 10 seconds, the cover order moves down to "Tier 2 Price" which is one tick higher than the entry fill price. Finally, after 15 seconds, the cover order moves down to "Tier 3 Price" which is the entry fill price, in an attempt to scratch the trade.

Before redesigning this logic using the newer version of the Stopwatch Block, it is important to note that:

- The Stopwatch Block cannot be reset (the cover logic displayed below is virtualized, which means that each fill event has an independent instance of the cover logic).
- The Stopwatch Block does not need to "tick" indefinitely - it only needs to "tick" twice at the 10-second mark and the 15-second mark.

The image <After Update> shows 2 new Stopwatch Blocks. The upper Stopwatch Block labeled "Tier 1 Stopwatch Block" is set to generate an event at the 10-second mark (10,000 ms). The event is then used to change the output value of the Value Accumulator Block labeled "Tier 1

Flag" from 0 to 1, which in turn updates the output value of the "In Tier 1" block from TRUE to FALSE. Blocks downstream from that point do not need to be redesigned.

Note that the lower Stopwatch Block labeled "Tier 2 Stopwatch Block" is activated *only after* the upper Stopwatch Block ticks. To be more specific, the discrete event message generated by the upper Stopwatch Block activates the lower Stopwatch Block.

The lower Stopwatch Block is set to generate an event at at the 5-second mark (5,000 ms). The event is then used to change the output value of the Value Accumulator Block labeled "Tier 2 Flag" from 0 to 1, which in turn updates the output value of the "In Tier 2" block from TRUE to FALSE. Blocks downstream from that point do not need to be redesigned.
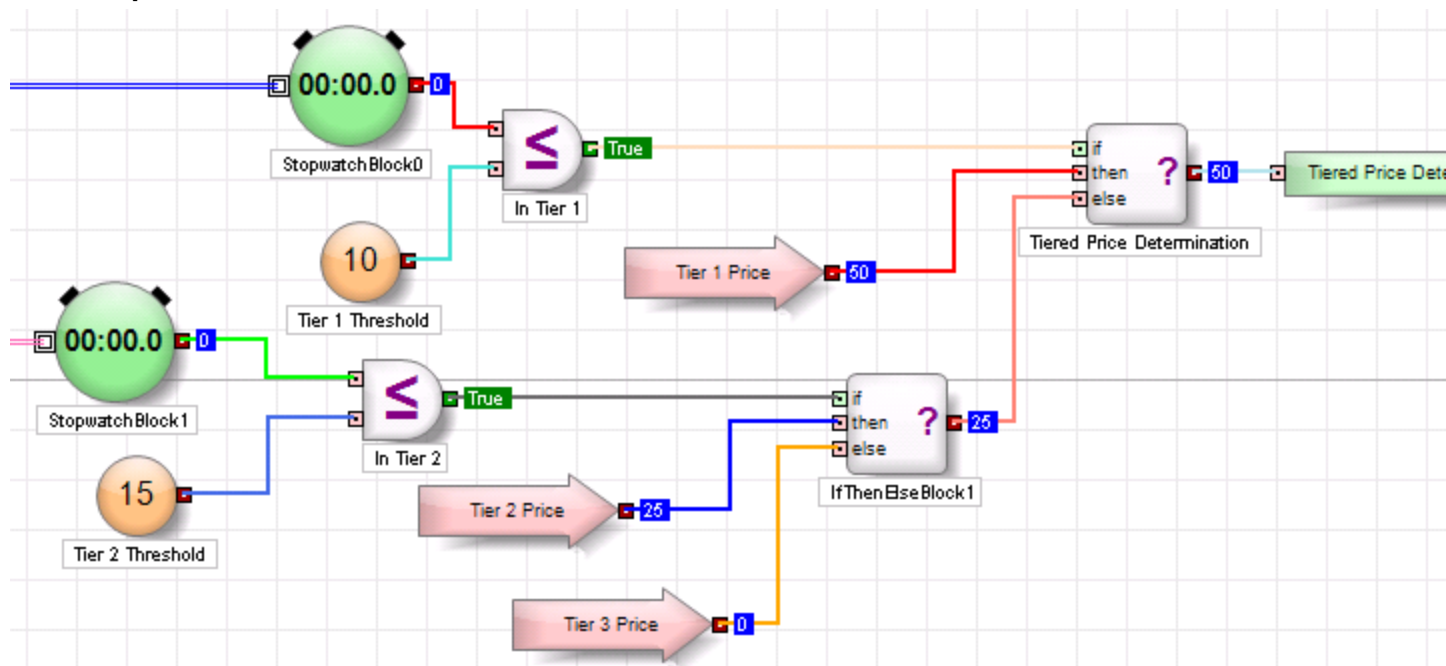
Before Update

After Update
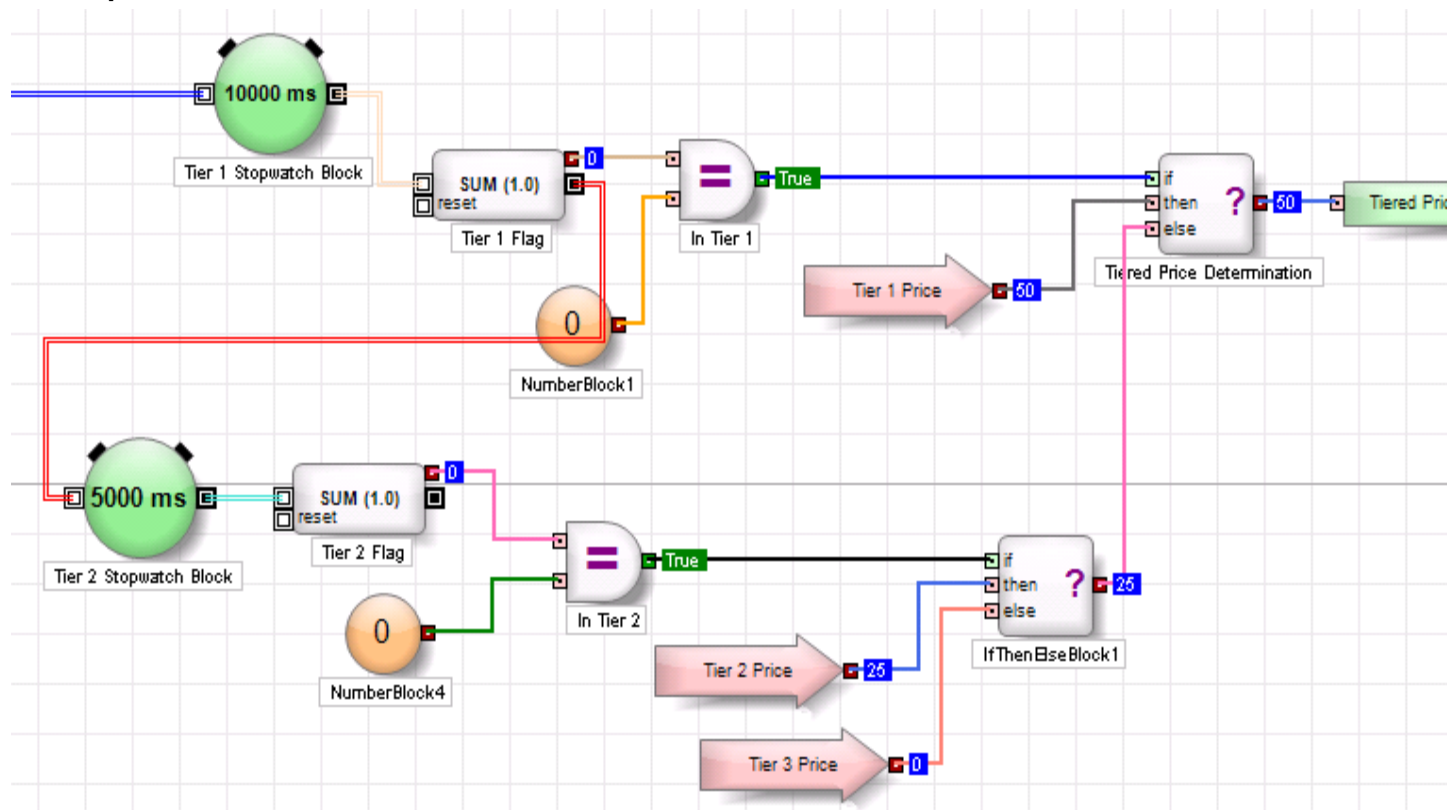
Example 3. Measuring the elapsed time between the last two trades

The image <Before Update> displays logic that is used to measure the time elapsed between the last 2 trades. Before redesigning this logic using the Generator Block, it is important to note that:

- Unlike the previous examples, the Stopwatch Block can be reset.
- Unlike the previous examples, the Stopwatch Block needs to tick indefinitely.

To meet the above 2 conditions, the old Stopwatch Block has been replicated using the Generator Block (set to generate events on a Timer Interval), as shown in the images <After Update> and <Inside Group Block "Custom Stopwatch Block">.

You can use the logic shown in the image <Inside Group Block "Custom Stopwatch Block"> to replicate the old Stopwatch Block in its entirety, but total replication should be reserved to cases like the one shown in this example where the Stopwatch Block *must* repeat indefinitely.

In most cases, the Stopwatch Block only needs to "tick" for a finite number of times, and therefore, it is far more efficient to design the new Stopwatch Block to "tick" as needed and then stop. Such efficient design will prevent the Stopwatch Block from generating unnecessary discrete event messages.

Before Update
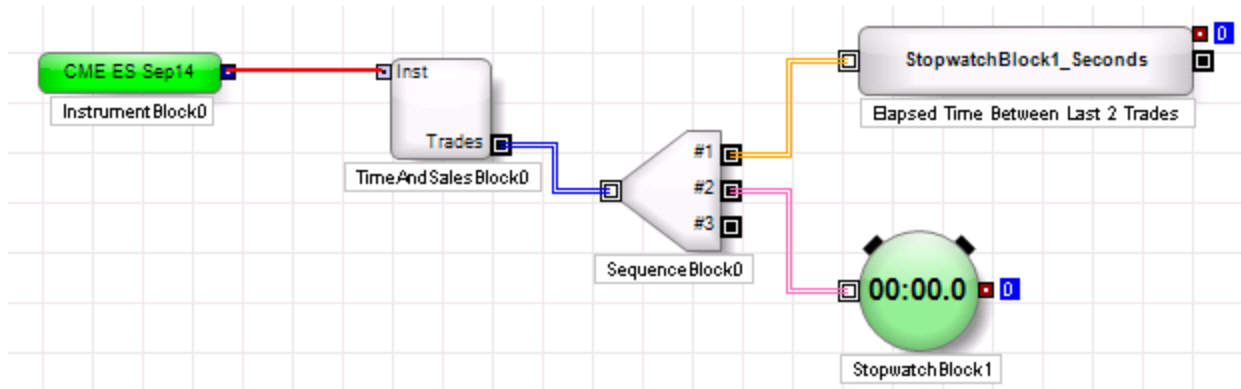
After Update
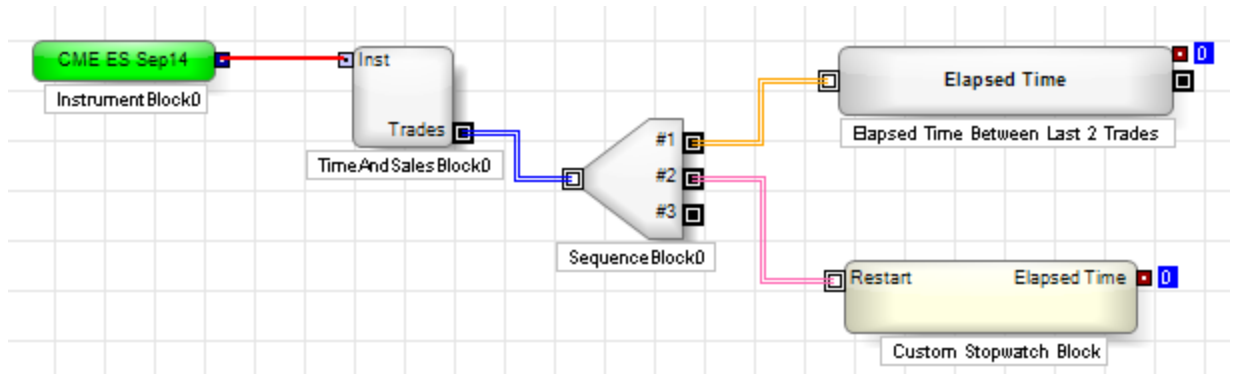
Inside Group Block "Custom Stopwatch Block"