

bTimer is a simple driver that can be used to determine the difference, in time, between a Start event and a Stop event. Unlike the default Control4 Timer driver that runs in countdown mode, bTimer captures incremental time. In the simplest use case (and likely the most popular), bTimer is started and when stopped will report the elapsed running time.

This driver is known as a "UI Button" driver. This class of drivers does not use C4's standard template as an interface but instead can use an icon or series of icons to communicate the driver's status - in this case, a timer.

bTimer tracks Current and Total times. When the timer is stopped, the Current properties are updated to reflect the most recent timer session. That time is then added to the Total time. The timer can be started and stopped as many times as you would like and each Current timer session will be added to the Total time.

When the timer is Reset, both the Current and Total counters are initialized to zero. If the timer was running and Reset, it will initialize and restart.

bTimer works by capturing the timestamp when started and again when stopped. There is no running process tracking the time as it ticks past. As a result, how long the timer has been running is not known by looking at the Current and Total properties until the timer is stopped. The Lap functionality helps with this limitation - when invoked, it updates the Current and Total times but does not stop the timer. If a timer is Lapped then Stopped, the Current and Total times are updated with the entire timer session. That is, Current and Total, although updated when Lapped, will reflect the time since the Start event not since being Lapped.

There are some use cases where knowing something is (still) running is useful. When the Timer Alert functionality is enabled, bTimer will fire a Timer Alert event. Timer Alert can be set to Off, Once or Repeating. bTimer will alert after Alert Interval minutes when Timer Alert is set to Once and will continue to alert every Alert Interval minutes when set to Repeating.

UI Button

As mentioned above, bTimer is a driver of type "UI Button". As such, bTimer changes its icon to reflect the activity of the timer. If an action or command (Start Timer, Stop Timer, etc.) is run on the driver, the icon is updated to reflect the current timer status. Otherwise, the icon is updated every 15 minutes.

The update works reliably on all connected Navigators (T3s, EA1, etc.) and on the Control4 app when the app is open and the icon is visible in the current view. It does not update reliably when the app is closed, or the icon is not visible in your current view.

To help address this 'feature', a single-click on the icon will invoke the 'Lap Timer' event AND update the icon to the current timer value. You will know the driver registered the click and is updating when you see the 'dot-dot-dot' icon.

bTimer will track any and all time ranges but because icons must be pre-generated and take up

bTimer Driver

space (did you notice the 30MB+ driver size?), icons for minutes, hours and days up to 99 are included. bTimer will show "1m" for Less than 1 Minute and "99+" for more than 99 days.

By default, bTimer will show the icon of the greatest "digit". For example, a timer running for:

- 1 Day, 2 Hours and 3 Minutes will display a "1d" icon
- 2 Hours and 3 Minutes will display a "2h" icon
- 3 Minutes will display a "3m" icon

Two or more clicks (within 2 seconds) will start an icon rotation beginning with largest "digit" down to the smallest "digit" followed by the default view of the largest "digit".

Usage Examples

bTimer can really be used to track incremental time between any events known to Control4. Here are two examples from my system:

I've been using bTimer to track TV usage in the house. There is an instance of the driver that starts and stops then the Family TV is turned on and off and another instance does the same for the Game Room TV. When the TVs turn off, I receive a push notification that tells me how long the TV was just on and the total time it has been on so far today. At midnight, I get an email that summaries various happenings for that day, including TV time. The timers are reset and ready for the next day. In case you didn't guess, I have kids and am a little sensitive about their screen time...

I also use bTimer to track the Spa usage. Specifically, one instance tracks total Spa On time and another instance tracks how long it took the Spa to reach the SetPoint. I use the Timer Alert functionality on the total Spa On timer to send me push notifications every hour the Spa is on. And yes, the spa may take a few hours to heat up and then only enjoyed for 5 minutes and may have been left on for long periods of time, unused. See previous note about having kids :-)

Properties

Status - Shows timer's most recent action

Start Time - epoch Start Time

Stop Time - epoch Stop Time

Current Elapsed Time - Time, in seconds, between Start and Stop Time

Current Elapsed Time Days, Hrs, Mins, Secs - Components of Current Time

Current Elapsed Time Friendly - User-friendly display for Push and Email notifications

Total Elapsed Time - Cumulative time, in seconds, timer has been running. Does not include currently running timer unless timer has been Lapped

Total Elapsed Time Days, Hrs, Mins, Secs - Components of Elapsed Time

Total Elapsed Time Friendly - User-friendly display for Push and Email notifications

Timer Alert - Set Alert mode (Off, Once, Repeating)

Alert Interval - How often, in minutes, Timer Alerts

Alert Count - Counter for number of Timer Alerts sent

Variables

TIMER_RUNNING - Boolean, indicates if timer is currently running

All other Properties are accessible programmatically via their paired Variables (START_TIME, STOP_TIME, CURRENT_ELAPSED_TIME, CURRENT_ELAPSED_TIME_DAYS, CURRENT_ELAPSED_TIME_HRS, CURRENT_ELAPSED_TIME_MINS, CURRENT_ELAPSED_TIME_SECS, CURRENT_ELAPSED_TIME_FRIENDLY, TOTAL_ELAPSED_TIME, TOTAL_ELAPSED_TIME_DAYS, TOTAL_ELAPSED_TIME_HRS,

bTimer Driver

TOTAL_ELAPSED_TIME_MINS, TOTAL_ELAPSED_TIME_SECS,
TOTAL_ELAPSED_TIME_FRIENDLY, TIMER_ALERT, ALERT_INTERVAL, ALERT_COUNT)

Events

bTimer fires the following events based on actions performed on the driver: TimerStarted, TimerStopped, TimerReset, TimerLap and TimerAlert

Known Issues and Limitations

- This driver must be installed from a file named "bTimer.c4z" otherwise bLicense (the driver used to purchase activation keys) will not recognize it as a valid driver. This is especially important with subsequent driver versions as many OSs will not overwrite downloaded files but will rename the newly downloaded file to something like driver(1).c4z.

Upgrade Mode

BNet Solutions drivers can automatically update themselves. New driver functionality or capability is typically packaged as an incremental "Major" version (v3, v4 etc). "Minor" versions (v2.3, v2.4) are typically maintenance releases that update underlying libraries, address a specific issue or usability concern.

Automatic - When "Automatic" is selected, the driver will upgrade/update itself when a new version is available. "Automatic" Upgrade Mode is strongly suggested.

Minor Only - Restricts the driver from upgrading between major versions but allows updates between minor versions.

Update Now - Checks for and upgrades to any newer Major or Minor version.

Troubleshooting

All BNet Solutions products have an additional 'Submit' Debug Mode. With this mode selected, the driver creates a unique log file to capture the Lua output based on the selected Debug Level (usually set to "5 - Debug"). Once 'Submit' Debug Mode is deselected, either manually or when the Debug Timer expires, the Submit Debug Log is uploaded to the BNet Solutions Server for analysis.

The server notifies me when Submit files are uploaded but if you have not purchased a license, I will have no way to reach back out to you for troubleshooting so please email me your contact information.

Trial and Purchase

All BNet Solutions drivers are fully functional for a 7-day Trial Period. To continue usage after the Trial Period, you will need to purchase a license. To do so, install the bLicense driver from <https://bnet4solutions.com> and follow the documented purchase steps. There are no refunds so please review the documentation for any limitations or known issues and determine acceptance during the Trial Period.

Contact Information

You can reach me at blucas@bnet4solutions.com for comments or questions.

bTimer Driver

Change Log

v1 - 3/19

v2 - 5/19

- Updated license and Submit Debug Mode

v3 - 10/19

- Updated architecture modules

v4 - 11/19

- Added auto-update functionality