



SOFTWARE CONTROL SERVICES (PTY) LTD

475 King's Highway, Lynnwood
P.O.Box 36675, Menlo Park
Pretoria, South Africa
0102

(t) +27 12 348 7301
(f) +27 12 348 1129
(e) techsupport@softconserv.com
www.softconserv.com

Timers Setup Guide

Version 0. 1

Prepared by: Michael Davis- Hannibal

Softcon Software Control Services (Pty) Ltd.

7 March 2017



Revision History

Name	Date	Reason For Changes	Version
MDH	7-Jul-10	Initial document	0.1

Contents

1. General Information	3
2. Setup	3

1. General Information

The system provides for numerous system timers (up to 65000) that generate set events on time-out. On events, timers are set to start (loads set, pre-set or current values), stop or set current values. Timers can be set or pre-set to cycle.

Typical examples:

- If operator does not accept an alarm within 60 seconds, sound a siren. If not accepted within 2 minutes, send an SMS.
- While a door is open, give an audible alarm every minute, and send an EMAIL if door remains open for longer than 10 minutes.
- After hours – if no motion is detected for more than 10 minutes, enable the alarm system.

These timers are set-up via the TIMER SET-UP list editor with the data below. Timing out and events sections describe how the system automatically changes certain data. The (how often) timers are checked are set in the PC set-up, setting which PC does the checking and a delay after which the checking is started after the programs starts (allowing for events to be received that may stop / reload timers (general set-up).

2. Setup

1. Click setup, setup editor, PC setup.
 - a. For the PC that will process the timers(Normally the server) under Timers Proc insert a tick, as well as set the Timers period(time between checking the timers)

Activity GRP	Timers Proc	Timers Period	Ma...
	✓	yyyy-mm-dd hh:mm:01	0
		yyyy-mm-dd hh:mm:ss	0
		www-mm-dd hh:mm:ss	0

Figure 1: PC Setup

2. Click setup, general setup. Locate the 'timers start after' field.
 - a. Timers are started (checked) after set seconds after client programs start. This allows for events to be received that may stop/reload timers.

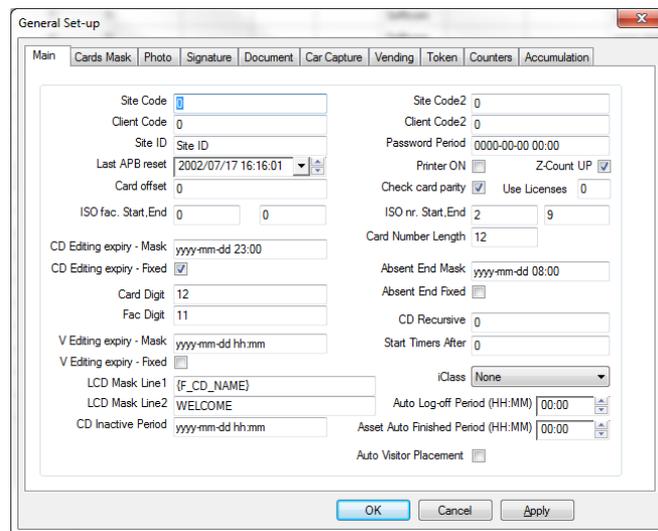


Figure 2: General Setup

3. Click setup, setup editor timers setup.
 - a. **Reference.** Timer (reference) number.
 - b. **Name.** Descriptive name given to the timer to identify it.
 - c. **Status.** Current status of the timer (automatically changed by the system):
 - 0=stopped
 - 1=running
 - 2=cycling
 - 3=timed-out.
 - d. **Status DateTime.** Last time the timer status was changed (automatically updated).
 - e. **Timer reload.** A preset reload value in seconds.
 - f. **Time remain.** Current value of the timer in seconds. The value can be set manually or by the system. The display is only updated when entering the menu, when the timer starts, stops or when timing-out - if the list is set to live, by selecting F5 (refresh).
 - g. **Time-out DT.** If status is running or cycling, contains when the time-out will occur if not stopped. Automatically updated when timer is starts running or cycling or when time remain is edited.
 - h. **Cycle reload.** Number of pre-set time-out cycles. 0 is not cycle, -1 is endless cycle.
 - i. **Cycle remain.** Number of remaining cycles on time-out. 0 is not cycle, -1 is endless cycle. If positive value, automatically decremented by the system when recycle starts.
 - j. **Cycles done.** Number of cycles done since started. Automatically incremented when timed-out.
 - k. **Event trigger.** Event that caused the timer start event (automatically saved by the system in event-event, cleared if started any other way – e.g. timed event, operator, etc.).
 - l. **Time-out result.** If not an empty string, the event that is automatically generated when time-out occurs. Event parameters are fixed or can be loaded from the Event trigger, or contain the reload or the started value by entering a ~ (insert) character as follows (case insensitive):
 - ~t Trigger type.
 - ~s Trigger status.
 - ~n Trigger system number.
 - ~v Trigger value.
 - ~x Trigger xref.
 - ~z1..10 Trigger z.
 - ~R Timer reload value.
 - ~C Timer remain (current value).
 - m. **Cycle result.** If not an empty string, the event automatically generated when cycling occurs, parameters as for time-out result event.
 - n. **Timing out** automatically occurs when timer status is not stopped or timed out (thus running or cycling) and the real time is after the time-out DT. Note that on starting of the client program, timer events are checked for the first time after a preset number of seconds (general set-up) after starting (allowing all events, timed events to be completed first). The PC and period of time-out checking is set.
 - o. On timing out, the following is done (in order):
 - Cycles done is incremented.
 - Time remain=0.
 - If cycles remain > 0 Decrement.
 - If cycles remain <> 0 Time remain=timer reload.

 - If time remain <> 0: Set timer status=cycling.
 - Set time-out DT.
 - Generate Cycling event.
 - Generate Cycle result event (if rep is event or event-tg

active).

or

If time remain = 0: Set timer status=timed-out.

Generate timed-out event.

Generate Timed-out result event (if rep is event or event-tg

active).

- p. **Timer events:** In any of the values below negatives -1 means forever. Parameter that are checked for legal values and changed, are indicated in red.

q. Control events:

80 Timer set. The system does the following (in order):
Load time reload: unchanged (z1-2), with z1. z1<-2 set to 0.
Load cycle value: unchanged (z2-2), with z2. z2<-2 set to 0.
Load time remain: unchanged (v-2), with time reload (v-3), with v. v<-3 set to 0.
Load cycle remain: unchanged (x-2), with cycle value (x-3), with x. x<-3 set to 0.

81 Timer start. Irrespective of the current status, the system does the following (in order):
(Event-event will have saved the triggering event to event trigger string).
V<-4, v set to -2. X <-3 set to -2.
Load time remain: unchanged (v-2), with time reload value (v-3), recalculate from time-out DT (v-4), with v.
Load cycle remain: unchanged (x-2), with cycle value (x-3), with x.
Clear cycles done.
Set time-out DT (of when time-out will occur, real time plus time remain).
DT of zero means forever (time remain negative).
Set status DT with real time.
If cycles remain=0, set status=running.
If cycles remain<>0, set status=cycling.
Generate started event (see below).

82 Timer stop. Irrespective of the current status, the system des the following (in order):
V<-3 set to -2. x <-3 set to -2.
If time remain is not negative, update (seconds from now till time-out date-time).

Set status DT with real time.
Set status=stopped.
Generate Stopped event (see below).
Set time remain: unchanged (v-2), with reload value (v-3), with v.
Set cycle remain: unchanged (x-2), with cycle value (x-3), with x.

r. Current Status events:

83 Timer in status. This is used as an event status trigger (as in level 55), being true when v equals the timer status (see timer status above).

- s. The following events are automatically generated by timers, with current values loaded:

v time remain
x cycles remain
q1 time preload
q2 cycle value
q3 cycles done.

t. Status changed events:

85 Timer Started.

- 86 Timer Stopped.
- 87 Timer Timed-out. See timing out above.
- 88 Timer Cycling. See timing out above.

u. Timer generated events are set to be logged, displayed, and printed and to trigger other events. A corresponding time-group sets when the event is an alarm and when the timed report function is active. See Event Reporting.

v. **Control Rep, TG** Control events – set, start and stop.

w. **Started Rep, TG** Started.

x. **Cycling Rep, TG** Cycling.

y. **Stopped Rep, TG**. Stopped.

z. **Timed-out Rep, TG**. Timed-out.

aa. **Time-out result Rep, TG**. Time-out generated event.

Event is only generated if rep is event or event-tg active (or tg 0).

Alarm or not is set in event.

bb. **Cycle result Rep, TG**. Cycle generated event.

Event is only generated if rep is event or event-tg is active (or tg 0).

Alarm or not is set in event.