

CS TimeClock

Processor Specification

Document Date: October 2013

Document Status: Version 4.22

Program Status: Implemented in CS TimeClocks version 1.15 and later.

© 2013 by CapeSoft Software

Contents

1. Introduction	4
2. Goals.....	4
3. Limitations.....	4
4. Clock Configuration.....	5
4.1 Global Settings	5
4.2 Employee Settings.....	6
4.3 Reciprocal Midpoint.....	7
5. Processing Rules & Functionality	7
6. Auto-Direction Updates	9
7. Flexitime.....	10
8. Information captured by employees and supervisors	10
9. Leave	10
10. Public Holidays	11
11. Processed Hours.....	11
12. Web Interface	12
13. Software Integration	12
14. Additional.....	13
15. Notes for Programmers	13
16. Tables	14
17. Version History.....	15
Build 2.04	15
Build 1.19c.....	15

Build 1.19d 15

1. Introduction

The CStimeClock has the ability to act as a stand-alone device (ie without additional PC software). In this mode some basic clocking-into-hours processing functionality is desirable. The CStimeClock range of clocks has this built-in processing capability, and all aspects of this ability are discussed in this document below.

This document covers information useful to the owner of the clock, as well as additional technical information aimed at programmers accessing the clock remotely.

2. Goals

1. **Simplicity:** Creating a completely flexible processing engine requires a complicated setup to be done by the client. We already have this in a PC software product, and it requires an experienced person to configure. The goal of the processing engine inside the clock is to remove all complexity. This in turn means few, if any, options available to the end user. While the operation of the clock should be intuitive, it is expressly not the goal of the processor to suit “all users”.
2. **Explainability:** The answers provided by the clock should be easily understood. In other words, regardless of their correctness, they should be easy to explain when given the input clockings. The goal here is to reduce questions by staff of payroll officers, and to reduce support necessary required from distributors.

3. Limitations

1. **7 Day Cycle.** The processor will assume the employees are working on a 7 day work cycle. (The payroll period may be longer than 7 days, but the Work cycle is 7 days.)
2. **No “Payroll Hours”.** The processor converts clocking into Daily Hours. Payroll hours are assumed to be a sum of Daily hours. Thus all payroll periods are supported, and no “Payroll Hours” rules need to be entered or applied.
3. **Leave Data collection, but not enforcement.** Employee leave and public holiday information is necessary to calculate daily hours correctly. However there are no rules governing whether an

employee is entitled to leave.

4. Flexitime collection, but not enforcement. The clock can keep a running balance of hours worked against hours expected. However no enforcement of core hours, minimum or maximum flexitime amounts are done.

4. Clock Configuration

4.1 Global Settings

1. Processing on/off. On by default.
2. Names of 9 time categories. These will default to Extra, Normal, Overtime, Saturday, Sunday, Public Holiday, Paid, Sick and Variance. While these names can be modified, the meaning is fixed (more on this below.)
3. Daily Balancing On/Off. On by default.
4. Clocking Rounding On/Off. If On then the rounding amount in minutes. Default 15 minutes.
5. Time category for “Before start of day”. Defaults to Extra time. (1)
6. Time category for “Normal Working Day”. Defaults to Normal. (2)
7. Time category for “After end of day”. Defaults to Overtime. (3)
8. Time category for Saturday work. Defaults to Saturday. (4)
9. Time category for Sunday work. Defaults to Sunday. (5)
10. Time category for public holidays. Defaults to Holiday. (6)
11. Time category for the daily Variance. Defaults to Variance (9).
12. Names for days of the week. Defaults to English.
13. Auto-End Processing On/Off. On by default.
14. Running Totals On/Off. On by default.
15. Leave Auto Approved On/Off. On by default.

16. Daily Thresholds for each category. Stored as seconds. Defaults to 0. (Requires firmware 1.19 or later).

Programmers note: 1=On, 0=Off.

4.2 Employee Settings

The following employee settings are “per day”. For each employee, these settings can be set differently for Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday.

1. Expected Time In
2. Expected Time Out
3. Time to be deducted for unclocked, unpaid, lunch breaks.
4. Expected Normal time hours for the day

Programmers note: Times are stored as “seconds since midnight”.

The following employee settings allow global settings to be overridden for each employee.

1. Running Totals. On/Off/Use Global. Default is Use Global.
2. Balancing. On/Off/Use Global. Default is Use Global.
3. Rounding. Minutes/Use Global. Default is Use Global.
4. Auto-End processing On/Off/ Use Global. Default is Use Global.
5. Leave Auto Approved On/Off/ Use Global. Default is Use Global.

Programmers note: 1=On, 0=Off, -1=Use Global.

4.3 Reciprocal Midpoint

The *reciprocal midpoint* of a shift is the midpoint of the start and end times, plus 12 hours. This is the point at which one calendar day ends and the next begins.

1. A person working 9 am to 5 pm has a midpoint of **1pm** (half way between the start and the end). The reciprocal midpoint is thus **1 am**.
2. A person working from 6 am to 2 pm has a midpoint of **10 am**. The reciprocal midpoint is thus **10 pm**.
3. A person working from 10 pm to 6 am has a midpoint of **2 am**. The reciprocal midpoint is thus **2 pm**.
4. If an employee does not have Start and End times entered for a day then the day runs from midnight to midnight. (i.e. the reciprocal midpoint is midnight.)

5. Processing Rules & Functionality

1. Daily Rounding. If activated, all clockings are rounded as per the time set in the global setting (or the employee override). This includes all break clockings, IN clockings and OUT clockings. The rounding is deemed to be “half half”. In other words, if the rounding period is set to 15 minutes, then each clocking will be rounded to the “nearest quarter hour”.

Programmers note: Rounding in the gloproc file is stored in minutes.

2. Time earned while on public holiday, leave, or sick leave is all sent to the appropriate category. Time due is the Expected time for the person on that day, multiplied by the leave partial. The leave partial is a %. Eg If Leave Partial = 50 then only half the expected time for the day will be sent to the leave category. Amounts greater than 100 are allowed. For example, setting leave partial to 200 results in double the amount of hours expected being allocated.
3. Time worked before the expected start time of the day is typically Extra time. Category is set in global settings. Some companies may choose to send this to Normal Time. Especially in the case where employees are working flexi-time.
4. Time worked between the expected start, and expected end of the day is typically Normal time. On Saturday it is typically Saturday time and on Sunday typically Sunday time.
5. Time worked after the expected day end is typically Overtime. Category is set in global settings. Some companies may choose to send this to Normal Time. Especially in the case where

employees are working flexi-time.

6. It is possible for Leave time, and Work time to be accumulated on the same day if the employee is on leave (or Public Holiday) and also comes to work.
7. An employee may take a half-day or quarter day leave. A suitable proportion of Leave time will be applied to the record.
8. Daily Balancing. If on, an employee is expected to earn their Expected Normal Hours each day. No overtime can be earned until this amount of time is earned. Equally, time worked in excess of this amount is automatically sent to "After end of day". On Saturday and Sunday, the normal Categories are Saturday and Sunday respectively. If the expected time for a day is zero, then no balancing occurs (regardless of the start and finish times.)
9. Normal Time worked on Saturdays & Sundays is sent to the category specified in the global setting. Ie Time between the Start and End times. If the whole day is say Sunday time, then set the start to 0:00, and the end to 24:00. Expected time can still be 0.
10. All work is assigned to the calendar day of the Start clocking. If the expected time crosses midnight, then a day break at the opposite of the expected midpoint is used. For example, if the expected IN is 4 pm, and the expected OUT is 12 midnight, then the day runs from 8am to 8am.
11. Extra clockings, of the same direction, are ignored. For example in the sequence In, In, Out the second In is ignored. In the sequence In, Out, Out, the second Out is ignored.
12. Variance time contains either a positive or negative amount equal to
Normal + Overtime + Sat + Sun + Holiday + Paid + Sick – Expected Normal Hours
Note that Extra time is not included here.
13. Lunch time. If a deduction is specified for an employee, then this deduction is made, from normal time, regardless of the time spent at work, or the time of the clockings. If the employee has worked, in the whole day, less than the lunch deduction amount, then only the time worked is deducted.

The supervisor can override the lunch deduction for any employee for the day. The lunch deduction is made before daily balancing occurs for the day. Extra Time and Overtime are not included in lunch deductions even if sufficient normal time does not exist. On Saturday, and Sunday the Saturday time and Sunday time is used in place of Normal Time.

14. If no employee setting record exists for the day, then Normal time is simply In-Out.

15. If an employee is still IN at the end of his working day, then the OUT clocking is assumed to be the expected end time, if the Auto End Processing setting is on.
16. If an employee has expected time for a day, but does not clock at all, then the `awol` field for that day is set to 1.
17. Where days overlap (ie rolling from say night to day shift) all clockings before the night shift day end belong to the night shift. The day end is the reciprocal midpoint of the night shift.
18. If the amount of time earned in a category is less than the daily threshold for that category, then the category time is set to zero. For example, if the threshold for overtime is set to 30 minutes (1800 seconds), and the employee works 29 minutes or less of overtime, then overtime is set to 0. If they work more than 30 minutes then the time is processed as per the other processing rules. Threshold limits are applied after the rounding of clockings takes place.

Programmers note: Thresholds in the `gloproc` file are stored in seconds.

6. Auto-Direction Updates

1. If a person is still IN at the end of the day, then their direction is automatically set to OUT at that time.
2. If a person is in an area other than AREA 0 at the end of the day, then their area is automatically set to 0 at that time.
3. The “End of the Day” is defined as the reciprocal midpoint between the expected start, and expected end time of the day. (See above for reciprocal midpoint definition.)

Programmers note: If bit 2 in `autoproc` field in `gloproc` table is set (i.e. `autoproc & 2`) then auto-direction updates are disabled. Firmware version 1.19d or later required for this setting to work. Bit positions are counted from 1, i.e. 1=1, 2=2, 3=4, 4=16 and so on.

7. Flexitime

1. Because we know the expected time for each employee, we can keep track if they are ahead, or behind their expected hours for the month. This is the total of the Variance category.
2. At the end of each month the current total for the employee is stored. This total can be edited by a supervisor.
3. There are no core hours, or other flexitime enforcement.
4. If an employee is allowed to, and chooses to, then a whole day can be taken as “Flexitime Leave”.
5. Flexitime can be enabled, or disabled via the Global setting called Running Totals.

8. Information captured by employees and supervisors

1. Clock In & Clock Out. Plus breaks if required.
2. Supervisor can override clocking times.
3. Supervisor can override employee lunch deduction.
4. Supervisor can override calculated hours. (No “Add” or “Subtract” hours, and no distinct “Processing” or “Override” hours. There is a single daily processing record, per employee, per day, containing both the calculated, and overriding hours.)
5. Leave taken. (Paid, Sick and Flexi required. Unpaid optional.)
6. Public Holidays.
7. Supervisor can override the flexitime total for a day, or the accumulated time carried into the following month.

9. Leave

1. There are five default types of leave. Paid, Sick, Flexi, Unpaid and Other. Additional leave types can be added as required. The default leave types can be edited. The category for time earned

on each leave type can be set by the user.

2. Captured as Fromdate to Todate.
3. Leave records may not overlap
4. For paid leave (paid, sick, flexi) expected hours are assigned to employee.
5. A partial day (as a percentage) can be used as a fractional factor for expected hours. Numbers greater than 100 are allowed.
6. If the employee is on leave, on a public holiday, then no leave hours are earned, rather Public Holiday hours are earned.
7. If the employee clocks while on leave then that time is processed as normal. However leave time, in the appropriate category will be added in as well.

10. Public Holidays

1. Each holiday must be entered every year. There is no support for automatically re-occurring holidays.
2. All holidays entered into the clock are considered to be “paid” holidays. If a holiday is Unpaid then do not enter the holiday as a public holiday. If desired place employees on unpaid leave for the day.
3. If the employee clocks while on public holiday then that time is processed as normal. However public holiday time, in the appropriate category will be added in as well.

11. Processed Hours

1. Processed hours are created automatically from the clockings, employee & global rules, in conjunction with Leave and Public Holiday records. The processed hours record is created (with zero time in it) at the beginning of each day. The record is updated automatically whenever the employee clocks.
2. To force the re-processing of a daily-hours record, set the `processed` field to 0.

3. Only one processed-hours record, per-employee, per-day, is allowed.
4. The user can override the processed hours or the allocated lunch deduction. If altering the allocated lunch deduction, then the record is reprocessed so that the new deduction can be taken into account. To force reprocessing set the `processed` field to 0 whenever another field in the record is changed.
5. The hours record includes a description of the “expected” clockings, as well as the “actual” clockings.

A `comment` field is provided in the hours field so that if the hours are overridden the supervisor can capture reasons for the change. The supervisor making the change is recorded in the `suen` and `stime` fields.

Programmers note: If bit 1 in the `autoproc` field in the `gloproc` table is not set (ie `autoproc & 1 = 0`), then hours processing is disabled. In firmware versions before 1.19c the whole `autoproc` field must be set to 0 to disable auto-processing. In firmware 1.19d and later bit 1 turns Processing ON if set, and bit 2 sets auto-directions OFF if set.

12. Web Interface

1. Users can use the web interface to add, edit, view and export processed hours.
2. Users can use the web interface to set global, and employee settings.
3. The web interface can be used to add, and edit employee leave records, and public holiday records.

13. Software Integration

These features are suggested software features, and are not implemented as features in the CSTimeClock.

1. By turning off the processing (ie bit 1 in `autoproc` in `GloProc` table set to 0) all processing of clocking into hours is negated. So employee, and global processing rules become unimportant. However external PC software can still choose to populate the hours table with processed time,

and can still collect changes to the hours, leave and public holiday tables. This allows the web interface to remain functional for remote users.

2. If you have turned off processing on the clock, then you have also probably not populated the schedules for the employees on the clock. If you also have night-workers then you almost certainly want to set bit 2 `autoproc` in the `GloProc` table on, which turns off auto-direction updates. Failure to do this will result in night workers having their direction reset at midnight. This setting requires firmware version 1.19d or later. Unfortunately the setting is not backwards compatible – if you set this value to 2 on a clock with firmware earlier than 1.19c then auto-processing on the clock will be turned on, and auto-directions will not be turned off. You can check the version of the firmware on the clock by checking the `dbversion` field in the `_info` table. This should be `>= 1194`.

14. Additional

1. If the bit 1 in the `autoproc` field in the `GloProc` table is set to 0 then processing is turned off. If the bit 2 is set (and firmware `>= 1.19d`) then auto-direction is turned off. For backward compatibility reasons note that bit 1 is “set it to turn processing ON” and bit 2 is set to “set it to turn auto-directions OFF”. These bits are both live. In other words setting `autoproc` to the value 3 (both bits on) will result in processing being on, and auto-direction off. This is useful where you will not enter schedules for the employees, but there are some night workers.
2. Hour records will be created for every day for every employee. Records are created early in the day.
3. All clocking times are truncated downwards to the nearest minute.
4. All clockings are processed immediately they are entered.
5. All “non-clocking-days” (ie where the employee does not clock at all) are processed just before, and just after midnight.
6. Only NORMAL and PAYROLL clocking types are taken into account when processing. Deleted clockings (`DEL = 0`) are also ignored by the processor.

15. Notes for Programmers

1. If `gloproc`, holiday or leavetype table changed then send a refresh command (`.r;`)

2. If a clocking is changed set pr = 0.
3. If an hour, `accumtime1` or leave record is changed, set `processed=0`.
4. To include the processor in a program;


```
#include "libpro.h"
proc_init(argc,argv);
result = (proc_load(db,FORCE) < 0);
```
5. To process a specific employee on a specific date call


```
result = proc_process(db,char *p_en,time_t p_date);
```
6. To process any outstanding clockings call


```
struct next_clocking_type clocking;
result = proc_waiting(db,&clocking);
```

16. Tables

Tables in the database used wholly or partly by the processing engine are;

Table	Description
Accumtime ¹	Accumulated running total for employees on last day of the month.
Clocking	Clockings from which hours are processed.
EmpProc	Employee processing options.
GloProc	Global processing options.
Holiday	List of Public Holidays
Hour	Processed Hours.
Leave	Employee leave records.
LeaveType	Supported leave types.

For details of the various tables used and explanations, of the individual fields, see the CStimeClock SDK documentation.

¹ Accumtime table is deprecated in clocks ver 2.04 or later.

17. Version History

New features and changes added after the initial release are listed here. Documentation for these features is included in the document above.

Build 2.04

Accumtime table has been deprecated.

Build 1.19c

Support for Minimum Daily Thresholds added

Support for disabling direction updates added via GloProc table, AutoProc field setting.

Build 1.19d

Autoproc changed to individual bits so that auto-processing can be on even if auto-direction is turned off.