Software requirements for caTools

1 Requirements

1.1. The following tools shall be available:

caget Reads and format value from channels.

cagets Writes 1 to PROC field of a channel and reads it back after channel has finished processing.

caput Writes a value to a channel.

caputq Writes a value to a channel, but does not wait for the processing to finish.

cainfo Displays detailed information about a channel.

camon Monitors a set of channels and outputs their values on each change.

cado Writes 1 to a channel, but does not wait for the processing to finish.

cawait Monitors a channel and waits until specified conditions for the channel match. Then the value is displayed.

1.2. caTools shall be written in C programming language.

1.3. caTools shall compile on the following platforms:

- SL6-x86 SL6-x86_64,
- eldk52-e500v2,
- eldk42-ppc4xxFP.

1.4. Input and expected output shall be backwards compatible with existing PSI tools, except for **caInfo** tool and array handling.

1.5. **caInfo** tool shall display all the information available in `DBR_CTRL_XXX` compound data type.

1.6. The user shall be able to request specified compound data type (eg. `DBR_TIME_CHAR`).

1.7. The user shall be able to select channel access timeout (eg. how long to wait for channel to connect).

1.8. Date and time of the record execution shall be available (CA server date/time).

1.9. Date and time of the record printout shall be available (client local data/time).

1.10. **camon** shall have additional incremental time-stamps options:

- Time elapsed since start of program.
- Time elapsed since last update.
- Time elapsed since last update, by channel.

1.11. **camon** shall have an option to exit monitoring after user specified number of updates.

1.12. **cawait** shall have an option to exit after a user specified timeout.

1.13. The format of the `decimal` values shall be settable:

- Round the value to closest integer, round to next integer, round to previous integer.
- Scientific notation (mantissa/exponent) with specified precision - %e format.
- Decimal floating point notation with specified precision - %f format.
- Use the shortest representation between %e and %f.
- Override the PREC field defined in the record.

1.14. The user shall be able to set the format of the `integer` values to hex, binary or octal.

1.15. The user shall be able to force interpretation of `enum` and `char` values as integers or characters / array of characters.

1.16. Arrays shall be printed as a list of values, where the formatting shall be based on the value type (eg. char type is displayed as an ASCII character, double type as a decimal number).

1.17. Array values separator shall be configurable.

1.18. The user shall be able to set the requested amount of array elements to read or write.

1.19. Severity and status shall be displayed when not in `NO_ALARM` state.

1.20. The user shall be able for force the severity and status to always be displayed.

1.21. The user shall be able for force the severity and status to never be displayed.

1.22. The user shall be able to hide channel name from the output.

1.23. The user shall be able to hide engineering units (EGU) from displaying next to values.

1.24. The default display output format will be in the following form:

```
[date] [time] [channel name] [value] [EGU] [severity and status]
```