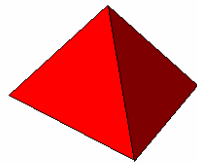


WinCPRS Protocol Language Manual



Absolute Aliens Oy

Revision 1.14

1.07.2009

WinCPRS Protocol Language

1. General

All commands are **case sensitive** and they must be written **exactly** as they have been described in the following chapter. The source code of the protocol has no line numbers but it is possible to use labels as a target of the jump and skip commands. The protocol compiler adds the line numbers and resolves the actual addresses of the labels. The maximum length of one line is 100 characters excluding leading spaces.

Labels, the targets of jump commands, must be started from the **first** column and they must be ended with the semicolon (;) character. Labels can not include any space character, and the maximum length of the label is 24 characters. There must be at least **one space** before any command, otherwise it is interpreted as a label. The maximum number of labels is 100.

All strings, like texts and signal names, must be put within apostrophes (' ') unless they do not contain any space character. Many commands consist of several parts, and parts must be separated with one or more spaces.

Some remarks on the syntax of parameters:

<parameter>	this parameter is obligatory
<parameter 1 parameter 2 ... parameter n>	one of these parameters must be used
{parameter}	this parameter is optional
<'text'>	the parameter is a string

The reserved words are written in **bold**, and the parameters set by the user are in **bold** and *italic*.

Several commands have the signal as a parameter. The signal is identified using three or four parameters:

< name >	the name of the signal exactly as it is in the Signal List dialog
< group >	the group of the signal 1 = <i>Raw signals</i> , 2 = <i>Time series&BRS</i> , 3 = <i>Spectrums</i> and 4 = <i>Miscellaneous</i>
< mode >	in the case of the spectrum group, 1 = <i>Normal</i> , 2 = <i>Sliding</i> and 3 = <i>Time-frequency</i> , with other signal groups this parameter has no meaning but it must be given
< version >	the version number of the signal

Some commands are actually functions, which return boolean value. This internal boolean value can be used e.g. to decide if the program should jump to another line. Search procedures can also return values, which can be tested or used with other commands.

2. Commands

Phase *<name>*

This command does not perform any action but it names the structural blocks of the protocol. The names of the phases are displayed on the protocol dialog in the phase window. It is not necessary to put the phase name within apostrophes even if it includes space characters.

;

This character starts the comment line, and it must be put in the first column of the line. All text after this character is interpreted as a comment line.

Comment *'text'* {**Clear**|**Left**|**Center**|**CLeft**|**CCenter**} {**Clear**}

Displays one comment line set by the parameter *text* on the protocol dialog. Options are following:

<i>Clear</i>	clears the display before writing; it can be used in combination with other options (the last option)
<i>Left</i>	text is left justified
<i>Center</i>	text is horizontally centered
<i>CLeft</i>	text is left justified and vertically centered
<i>CCenter</i>	text is horizontally and vertically centered

Comment **Clear**

Clears the protocol display.

AddLine

Adds one empty line on the protocol display.

CommentToInfo *'text'*

Adds the text line onto the Info Page.

Check *<Cont/Stop>* => **boolean value**
 <Yes/No> => **boolean value**
 <OK>

Displays various buttons on the protocol dialog:

<i>Cont/Stop</i>	Continue and Stop buttons are displayed and the protocol waits for the user response. If Continue button is pressed, this function returns <i>true</i> value. If Stop button is pressed, the function returns <i>false</i> value and the user should decide if the protocol will be ended.
<i>Yes/No</i>	Yes and No buttons are displayed and the protocol waits for the user response. If Yes button is pressed, the function returns <i>true</i> value, otherwise false.
<i>OK</i>	OK button is displayed and the protocol waits for the user response.

HideAllButtons

Hides all buttons on the protocol dialog.

ShowImage <'filename'>

Displays the picture in the protocol dialog. Only picture files with the file extension *.*bmp* (bitmaps) are possible. The picture file must locate in the same directory as the protocol file itself.

ShowImage **Clear**

The picture in the protocol dialog is removed.

RunMedia <'filename'>

Runs audio and video files. Audio files must be *wav*-type and video files *avi*-type. The audio/video file must locate in the same directory as the protocol file itself.

Goto <*label*>

Protocol jumps to another line marked with the *label*.

Skip <*label*>

Protocol jumps to another line marked with the *label* if the internal boolean result is *true*. The boolean value is set or reset by the last boolean function performed before the Skip command.

Nskip <*label*>

Protocol jumps to another line marked with the *label* if the internal boolean result is *false*. The boolean value is set or reset by the last boolean function performed before the Skip command.

Break

Stops the protocol immediately.

SetToggle <toggle num> <value>

Protocol system supports internal toggle variables which are indexed by the *toggle num* (1 .. 10). The value of the toggle is set by the parameter *value*, which must be integer. Toggle variables can be used to construct multi choice case-structures.

TestToggle <toggle num> <value> => boolean value

Tests if the toggle variable of the index *toggle num* has the value of *value*. If comparison is successful, the function returns *true*, otherwise *false*.

IncrementToggle <toggle num>

Increments by one the value of the toggle variable of the index *toggle num*.

DecrementToggle <toggle num>

Decrements by one the value of the toggle variable of the index *toggle num*.

ShowDialog <QRS|MSNA|Breathing|Database|Acquisition|Import|Log|PhysioTool> {mode}

Displays the dialog. In the case of *PhysioTool* dialog, the *mode* selects the test type.

HideDialog <QRS|MSNA|Breathing|Database|Acquisition|Import|Log|PhysioTool>

Hides the dialog.

Detect <QRS|QRSAdvanced|BP|MSNA|Breathing> <version>

Starts a detection procedure using ECG, Blood pressure, MSNA or Respiration Volume signal of the *version*.

WaitFor <QRS|QRSAdvanced|BP|MSNA|Breathing|DDEDataPoked>

Protocol waits until the detection procedure is completed or the DDE client has poked data.

CreateSignal <'name'> <group> <mode> {source version}

Creates the signal. The version of the new signal is the first free one. If the *source version* is given, the signal of that version is used when creating the new signal.

ShowSignal <'name'> <group> <mode> <version>

The signal is set to be visible.

HideSignal <'name'> <group> <mode> <version>

Hides the signal.

HideSignal All

Hides all signals.

DeleteSignal <'name'> <group> <mode> <version>

Deletes the signal.

CreateAdditionalStatistics <'name'>

Creates a new signal, which has only statistics but not actual signal data. The name has no restrictions.

DeleteAdditionalStatistics <version>

Deletes the additional statistics.

CreateReferenceStatistics <'name'>

Creates a new signal, which has only reference type of statistics but not actual data. The name has no restrictions.

DeleteReferenceStatistics <version>

Deletes the reference statistics.

SetXCursors <'name'> <group> <mode> <version> <low> <high>

Sets the positions of the x-cursors, *low* and *high*, on the signal window.

SetXCursorsToLimits <'name'> <group> <mode> <version>

Sets the positions of the x-cursors to the limits of the signal.

SetXCursorsFromVariables <'name'> <group> <mode> <version>

Sets the positions of the x-cursors using the internal variables #1 and #2.

SetXCursorByMarkerSignal <'name1'> <group1> <mode1> <version1>
 <'name2'> <group2> <mode2> <version2>
 <Left|Right'> <Up|Down'> <threshold>

Sets the position of the left or right x-cursor on the signal 1 based on marker values on the signal 2. If the option *Left/Right* is used, the left/right cursor is moved. The position of the cursor is set by searching the next crossing of the signal 2 upwards (the option *Up*) or downwards (the option *Down*) over the threshold level set by the parameter *threshold*. The search of the next crossing is always started from the current position of the left cursor on the signal 1.

MoveXCursors <'name'> <group> <mode> <version> <leftshift> <rightshift> {Rel}

Moves the position of the left and right cursors by amount (positive or negative) given by the parameters *leftshift* and *rightshift*. If the option *Rel* is used, the position of the right cursor is relative to the position of the left cursor and the distance of the cursors is set by the parameter *rightshift*.

SetYCursor <'name'> <group> <mode> <version> <level>

Sets the position of the y-cursors, *level*, on the signal window.

SetXScales <'name'> <group> <mode> <version> <min|-shift|+shift|CP> <max|-shift|+shift|CP >

Sets the x-scales of the signal window. If the scale value starts with – or + character, then it is interpreted as a relative shift to the current value. If the *CP* option is used, then the scale value is not changed but the current value is used.

SetXScalesToLimits <'name'> <group> <mode> <version>

Sets the x-scales of the signal window to the limits of the signal.

SetYScales <'name'> <group> <mode> <version> <min> <max>

Sets the y-scale of the signal window.

SetCommonYScales <'name1'> <group1> <mode1> <version1> <'name2'> <group2> <mode2>
<version2>

Sets the y-scales of two signal windows to be equal.

AutoScales

Autoscales all signal windows.

SetCaption <'name'> <group> <mode> <version> <'text'>

Sets the caption, the *text* parameter, of the signal window

ShareLimits <'name'> <group> <mode> <version>

Shares the limits of the signal window to all other windows.

SetSlopeMode <'name'> <group> <mode> <version>

Sets the signal window to the slope mode.

ClearSlopeMode <'name'> <group> <mode> <version>

Removes the slope mode of the signal window.

SetDots <'name'> <group> <mode> <version> <On|Off>

Sets or resets the dot mode of the signal window.

SetTimeStamps <'name'> <group> <mode> <version> <On|Off>

Sets or resets the time stamp mode of the signal window.

ShiftStartTime <'name'> <group> <mode> <version> <time shift>

Shift the start time of signal by the amount given by the parameter *time shift*.

BringToFront <'name'> <group> <mode> <version>

Brings the signal window in front.

LockSignals <'name1'> <group1> <mode1> <version1> <'name2'> <group2> <mode2> <version2>

Locks two signals, the first one is the master and the second one the slave.

AddMarker <'name'> <group> <mode> <version> <pos> <'text'> <textpos> <align>
<linetype> <color> {Left|Right|Both}

Creates a marker on the signal window. The parameters are:

<i>pos</i>	the position of the marker
<i>text</i>	the text line linked to the marker
<i>textpos</i>	Top = 1, Middle = 2, Bottom = 3
<i>align</i>	Left = 1, Center =2, Right = 3
<i>linetype</i>	None = 1, Solid =2, Dashed =3, Dotted = 4
<i>color</i>	1 ... 15

If some of the option *Left*, *Right* or *Both* are used, the marker is set according the corresponding cursor position, not by the parameter *pos*.

ResetLogTimer

Resets the real time clock on the Log dialog.

StopLogTimer

Stops the real time clock on the Log dialog.

AddLogInformation <'text'>

Adds one line, the *text* parameter, on the Log dialog.

FoundSignal <'name'> <group> <mode> <version> => boolean value

Returns *true* value if the signal is available, otherwise *false*.

InitialFile <'filename'>

Opens the initialization file using the current initialization file directory.

CheckCurrentInitialFile <'filename'> => **boolean**

If the current initialization file is *filename*, returns *true* value, otherwise *false*.

ReferenceFile <'filename'>

Opens the reference file using the current protocol file directory.

FoundFile <Initial|Data|Database|Import|Protocol|Reference|Append|General> <'filename'> => **boolean value**

Returns *true* value if the file can be found, otherwise *false*. The file is searched from the current directory specified by the file type parameter. If the option *General* is used, the file path must be complete including the drive letter.

ShowClock <duration> {Minutes} {Remaining}

Displays the time line. The maximum time is *duration* in seconds. The unit of the time is normally seconds but if the option *Minutes* is used, the unit is minutes. If the option *Remaining* is used, the time is counted backwards.

SetAlarm <index> <time> <'text'>

When the time line is visible, it is possible to set alarm actions, indexed from 1 to 10. The alarm is enabled when the time line reaches the value *time* (in seconds), and then the *text* is shown on the protocol display.

ClearAlarm

Clears all alarms.

AddStatistics <'name'> <group> <mode> <version>

Adds a new statistical set on the signal window using current settings and cursor positions.

AddSlopeStatistics <'name'> <group> <mode> <version>

Adds a new slope statistics on the current statistical set of the signal window using current cursor positions.

UpdateStatistics <'name'> <group> <mode> <version>

Updates the current statistical set on the signal window.

SelectStatistics <'name'> <group> <mode> <version> <set num>

Makes the statistical set indexed by the *set num* parameter as an active set on the signal window.

TransferStatisticToStatistic <'name'> <group> <mode> <version> <set num> <item num>
<target version> <'item name'> <E|Ln|Ln1|Log|Log1> <digits>

Transfer one statistic item from the signal window to the additional statistics. The statistical set is selected by the parameter *set num* and the item by the parameter *item num*. The additional statistic version is set by the parameter *target version*. The added statistical item gets the name *item name*. The value is transformed according to the last parameter: E = no conversion, Ln = natural logarithm, Log = 10 base logarithm, Ln1 = natural logarithm -1, Log1 = 10 base logarithm -1. If the parameter *digits* is negative, the number of decimals is the same as for the original statistical item, otherwise the number by the parameter is used.

TransferStatisticToReference <'name'> <group> <mode> <version> <set num> <item num>
<target version> <'item name'> <E|Ln|Ln1|Log|Log1> <digits>

Transfer one statistic item from the signal window to the reference statistics. The statistical set is selected by the parameter *set num* and the item by the parameter *item num*. The reference statistic version is set by the parameter *target version*. The added statistical item gets the name *item name*. The value is transformed according to the last parameter: E = no conversion, Ln = natural logarithm, Log = 10 base logarithm, Ln1 = natural logarithm -1, Log1 = 10 base logarithm -1. If the parameter *digits* is negative, the number of decimals is the same as for the original statistical item, otherwise the number by the parameter is used.

TransferAdditionalStatisticToReference <version> <set num> <item num><target version>
<'item name'> <E|Ln|Ln1|Log|Log1> <digits>

Transfer one statistic item from the additional statistics to the reference statistics. The statistical set is selected by the parameter *set num* and the item by the parameter *item num*. The reference statistic version is set by the parameter *target version*. The added statistical item gets the name *item name*. The value is transformed according to the last parameter: E = no conversion, Ln = natural logarithm, Log = 10 base logarithm, Ln1 = natural logarithm -1, Log1 = 10 base logarithm -1. If the parameter *digits* is negative, the number of decimals is the same as for the original statistical item, otherwise the number by the parameter is used.

TransferVariableToStatistic <target version> <'name'> <'unit'> <digits>

Some protocol functions return numerical value, which can be transferred, to the additional statistic. The version number of the additional statistic is set by the parameter *target version*. The name and the unit of this statistical item can be set by the parameters *name* and *unit*.

TransferVariableToReference <target version> <'name'> <'unit'> <digits>

Some protocol functions return numerical value, which can be transferred to the reference statistic. The version number of the reference statistic is set by the parameter *target version*. The name and the unit of this statistical item can be set by the parameters *name* and *unit*.

AddNumericParameter <'name'> <group> <mode> <version> <'parameter name'>
<value|Var1|Var2> <digits> <'unit name'>

Adds new numeric parameter (real number) on the parameter section of the signal window. The name, value and the unit is set by the parameters *parameter name*, *value* and *unit*. If option Var1 or Var2 is used, the parameter is set by the value of the internal variable Var1 or Var2.

AddNonNumericParameter <'name'> <group> <mode> <version> <'parameter name'> <'value'>
<'unit name'>

Adds new nonnumeric parameter (string) on the parameter section of the signal window. The name, value and the unit is set by the parameters *parameter name*, *value* and *unit*.

AddNumericParameter <'name'> <group> <mode> <version> <'parameter name'>
<'value'|Var1|Var2> <digits> <'unit name'>

Adds new parameter on the parameter section of the signal window. The name, value and the unit is set by the parameters *parameter name*, *value* and *unit*. If option Var1 or Var2 is used, the parameter is set by the value of the internal variable Var1 or Var2.

TestNonNumericParameter <'name'> <group> <mode> <version> <'parameter name'> <'value'>
=> **boolean value**

Returns *true* value if the parameter section of the signal includes the *parameter* and its value is equal to *parameter value*, otherwise *false*.

OpenMeasure => boolean value

Opens the connection to the data acquisition device. If the connection can be established, this function returns *true* value, otherwise *false*.

CloseMeasure

Closes the connection to the data acquisition device.

StartMeasure => boolean value

Starts to read data from the data acquisition device. If the operation is successful, this function returns *true* value, otherwise *false*.

SaveMeasure

Data from the data acquisition device is saved to memory.

StopMeasure

Stops to save data from the data acquisition device.

Channel Total <channel number> <sampling rate>

Sets the number of channels and the sampling rate (in unit samples/second) for data acquisition.

Channel <channel index> <'name'>

Selects the type of the signal for each data acquisition channel, *channel index* must be 1 .. 8. The name of the signal must be exactly as it is written in the data acquisition dialog.

ChannelSettings <channel index> <Bipolar|Unipolar> <filter mode> <Averaging|NoAveraging> <averaging sample num>

Sets the data acquisition parameters of the channel, the *channel index* must be 1 .. 8. The *filter mode* can be either 1 (no filtering), 2 (50 Hz filter) or 3 (60 Hz filter). The number of samples using in the averaging procedure is set by the parameter *averaging sample num*.

TileSignals <column num> <screen width ratio>

Tiles all visible signal windows using columns set by the parameter *column num*. Columns can fill the application window up to *screen width ratio* (0 ..100) percentage of the total area.

CascadeSignals

Cascades all visible signal windows.

PositionSignal <'name'> <group> <mode> <version> <Bottom|Top> <Left|Right>

Positions the signal window within the client area of WinCPRS.

TrimRawSignalTimeRanges

Cuts all raw signals in such a manner that their have equal start and end time.

ResetRawSignalStartTimes

Resets the start time of all raw signals.

ValidSignal <'name'> <group> <mode> <version> <min> <max> => boolean value

Returns *true* value if the data values of the selected signal are between *min* and *max* values, otherwise false.

OpenData

Opens the Open file dialog.

OpenDataFromDatabase

Opens the data file linked to the current record of the database.

SaveAsData

Opens the Save As dialog.

SaveAsDataFromDatabase {Name} {IDNumber} {Protocol} {Session} {Date}

Saves the data using the patient name, ID number, protocol name and date from the current record of the database if the corresponding option is selected.

CloseData

Closes the data file.

LinkToDatabase

Links current data file to the current record of the database.

DatabaseOpened => boolean value

Returns *true* value if the database has been opened, otherwise *false*.

CheckDatabaseEditingMode => boolean value

Returns *true* value if the database is in the editing or inserting mode, otherwise *false*.

SetDatabaseEditMask <IDNumber|Phone> <mask>

Sets the mask property for the selected field on the database record.

Sound <frequency> <duration>

Creates a sound with *frequency* in Hz and *duration* in milliseconds using either the system loudspeaker or sound card (see SoundWave command).

SoundPattern <frequency1> <duration 1> <frequency 2> <duration 2> {DisplayS|DisplayL} {Pie}

Creates 2-tone metronome sound using either the system loudspeaker or sound card (see SoundWave command). *Frequencies* are in Hz and *durations* in milliseconds. If the option *DisplayS* or *DisplayL* is selected, also the *In* and *Out* labels are shown on the protocol dialog. The size of the labels is small with the *DisplayS* option and large with the *DisplayL* option. If the *Pie* option is set, also the pie diagram of the sound progress is shown.

SoundPatternFromFile <'filename'> <duration 1> <starttime 1> <endtime 1> <duration 2> <starttime 2> <endtime 2> {DisplayS|DisplayL}{Pie}

Plays two sounds from the wav-file *filename*. *Durations*, *starttimes* and *endtimes* are in milliseconds. *Starttime* and *endtime* determines the beginning and the end of the corresponding sound in the sound record. If the option *DisplayS* or *DisplayL* is selected, also the *In* and *Out* labels are shown on the protocol dialog. The size of the labels is small with the *DisplayS* option and large with the *DisplayL* option. If the *Pie* option is set, also the pie diagram of the sound progress is shown. It should be noticed that this command always assumes that the computer has the sound card.

SoundPattern Stop

Stops metronome sound.

SoundWave <On|Off>

Sets mode of the sound generation: if *Off*, the sounds are produced using the system loudspeaker, if *On*, the sounds are taken from the specific sound files and playing through the sound card of the computer.

ShowDataBar <'name'> <version> <min> <max> <limit min> <limit max>

Shows bar display. The signal must be one of the raw signals. There can be only one bar display. The minimum and maximum values of the display are set by the parameters *min* and *max*. The target limits are set by the parameters *limit min* and *limit max*.

ShowDataBar <'name'> <version> <min> <max> Var <limit min> <limit max>

Shows bar display. The signal must be one of the raw signals. There can be only one bar display. The minimum and maximum values of the display are set by the parameters *min* and *max*. The target limits are set by the parameters *limit min* times Var2 and *limit max* times Var2.

HideDataBar

Hides the bar display.

ShowPeriodDisplay <'name'> <version> <type> <threshold|Auto>

Shows period display. The signal must be one of the raw signals. There can be only one period display. If *type* = 1, the period is shown in seconds, if *type* = 2, in bpm. The period is calculated using threshold detection method, the level is set by the parameter *threshold*. If *Auto* mode is set, the threshold is calculated automatically from the signal itself.

HidePeriodDisplay

Hides the period display.

ShowTrendDisplay <version> <mode> <threshold|Auto>

Shows trend display. There must be the ECG signal (all modes) and blood pressure signal (modes 3 and 4) available. The period display cannot be used simultaneously. There are four different modes: 1 - RRI; 2 - RRI, HR; 3 - RRI, SAP, DAP; 4 - RRI, HR, SAP, DAP. The trends is calculated using threshold detection method on the ECG signal, the level is set by the parameter *threshold*. If *Auto* mode is set, the threshold is calculated automatically from the signal itself.

HideTrendDisplay

Hides the trend display.

DialogWidth <Normal|Wide>

Sets the width of the protocol dialog to normal or wide.

CreateProtocolFrame <'protocol name'> <protocol type>

Creates protocol frame for result pages. There are no restrictions for the *protocol name*. The version of the new protocol frame is the first free one. The *protocol type* is reserved for future use, it should be set to value 1.

FoundProtocolFrame <'protocol name'> <protocol version> => boolean

Returns the value *true* if the protocol frame is found, otherwise the value *false*.

DeleteProtocolFrame <'protocol name'> <protocol version>

Deletes the protocol frame.

OpenProtocolFrame <'protocol name'> <protocol version>

Opens protocol frame from the protocol list and sets it as a current protocol frame.

ShowProtocolFrame

Displays currently open protocol frame.

HideProtocolFrame

Hides currently open protocol frame.

CreateProtocolPages <'name 1'> <'name 2'> <'name 3'> <'name 4'> <'name 5'> <'name 6'>

Creates results pages into the current protocol frame. There can be 1 .. 6 pages which have the name set by the parameters *name n*. If the page name is null string, that page is not created. Each page consists of six output modules (numbered 1 .. 6) which can be used to display signals, statistics, parameters and comments.

CreateProtocolSignalModuleNormal <page> <module> <'name'> <group> <mode> <version>
{NoXScale}

Creates module with signal window graphics on the *module* of the *page*. If the option *NoXScale* is selected,

the signal is drawn using the current x-scales.

CreateProtocolSignalModuleLarge <page> <module> <'name'> <group> <mode> <version>
{NoXScale}

Creates pair of module (two adjacent modules) with signal window graphics on the *page*. The *module* must be the left one. If the option *NoXScale* is selected, the signal is drawn using the current x-scales.

CreateProtocolStatisticModule1 <page> <module> <'name'> <group> <mode> <version>
<'set name'> <set num>

Creates the module with the signal statistic on the *module* of the *page*. Statistic set is positioned on the *left* side of the module. Set number is determined by the parameter *set num*, and the name of the set by the parameter *set name*.

CreateProtocolStatisticModule2 <page> <module> <'name'> <group> <mode> <version>
<'set name'> <set num>

Creates the module with the signal statistic on the *module* of the *page*. Statistic set is positioned on the *right* side of the module. Set number is determined by the parameter *set num*, and the name of the set by the parameter *set name*.

CreateProtocolAdditionalStatisticModule1 <page> <module> <version> <'set name'> <set num>

Creates the module with the additional statistic on the *module* of the *page*. Statistic set is positioned on the *left* side of the module. Set number is determined by the parameter *set num*, and the name of the set by the parameter *set name*.

CreateProtocolAdditionalStatisticModule2 <page> <module> <version> <'set name'> <set num>

Creates the module with the additional statistic on the *module* of the *page*. Statistic set is positioned on the *right* side of the module. Set number is determined by the parameter *set num*, and the name of the set by the parameter *set name*.

CreateProtocolReferenceStatisticModule <page> <module> <version> <'set name'> <set num>

Creates the module with the reference statistic on the *module* of the *page*. Set number is determined by the parameter *set num*, and the name of the set by the parameter *set name*.

CreateProtocolReferenceGraphicalModule <page> <module> <version> <'set name'> <set num>

Creates the module with the reference graphics on the *module* of the *page*. Set number is determined by the

Updates single Valsalva analysis statistics with the RRI and SAP signals.

StepValsalva *<version> {Next|Previous}*

Steps to the next or previous Valsalva section in the RRI signal.

StepValsalvaWithSAP *<version> {Next|Previous}*

Steps to the next or previous Valsalva section in the RRI and SAP signals.

AnalyseDeepBreathing *<version>*

Analyzes Deep Breathing test using the RRI signal.

UpdateDeepBreathing *<version>*

Updates Deep Breathing test analysis statistics.

AnalyseOrthostatic *<version>*

Analyzes Active Orthostatic test using the RRI signal.

UpdateDeepBreathing *<version>*

Updates Active Orthostatic test analysis statistics.

AnalyseHandgrip *<version>*

Analyzes Handgrip test using the RRI signal.

AnalyseHandgripWithSAP *<version>*

Analyzes Handgrip test using the RRI and SAP/DAP signals.

AnalyseAutoHandgrip *<version> <target pressure>*

Analyzes Handgrip test using the Handgrip pressure and RRI signals.

AnalyseAutoHandgripWithSAP *<version> <target pressure>*

Analyzes Handgrip test using the Handgrip pressure, RRI and SAP/DAP signals.

UpdateHandgrip <version>

Updates single Handgrip analysis statistics with the RRI signal.

UpdateHandgripWithSAP <version>

Updates single Handgrip analysis statistics with the RRI and SAP/DAP signals.

ShowInfoPage

Displays Info Page.

HideInfoPage

Hides Info Page.

ClearInfoPage

Clears the content of Info Page.

ShowStatisticPage

Display Statistic Page.

HideStatisticPage

Hides Statistic Page.

Filter <'name'> <group> <mode> <version> <Notch|Notch50|Notch60|BandPass|LowPass|HighPass>
<freq 1> <freq 2>

Filters the signal using one of the methods. Filter frequencies are set by the parameters *freq 1* and *freq 2*.

Baseline <'name'> <group> <mode> <version> <Global|Local> <range> <order>

Removes the baseline of the signal either using a global or local polynomial fitting. The order of the polynomial is given by the parameter *order*. When using the local fitting option, the parameter *range* determines the range of the fitting. This parameter must be given even if global fitting method is used.

AveragedECG

Creates averaged ECG using the ECG and RRI signals of version 1.

SlidingStatistic <'name'> <group> <mode> <version> <type> <width> <step> <position>

Creates the sliding statistic signal. The type of the statistics is set by the parameter *type* (1 .. 6), the width of the calculation window by parameter *width*, the step of the window by the parameter *step*, and the temporal position of the value within the window by the parameter *position* (1, 2 or 3).

AskValue <'name'> <'unit'> <default> <min> <max> => **Var1**

Opens the dialog where the user is asked to give some value. The name, unit and the default value are set by the parameters *name*, *unit* and *default*. The valid range of the value is checked using the minimum and maximum values *min* and *max*. The value is saved to the internal variable **Var1**.

ShowLargeLabel <'text'> <On|Off>

Displays or hides the large label on the top right of the protocol dialog. The label is set by the parameter *text* and the it is shown with the option *On* and hidden with the option *Off*.

SendDDECommand <'text1'> <'text2'> <'text3'>

Sends command to the DDE client.

ClearDDEDataPoked

Clears the internal event for poked data from the DDE client.

LockBreathToPeriod <ratio>

Locks the In and Out texts of the protocol dialog into the heart beat.

UnlockBreathToPeriod

Stops to display heart beat locked In and Out texts.

Language <Finnish|English>

Sets the language of the buttons and some internal messages of the protocol dialog.

ProtocolNameToDatabase

Transfers the name of the current protocol to the protocol field of the current record of the database.

ShowAlarmButton

Displays the alarm button by which the alarm stamp to the Log and Info dialog can be written.

HideAlarmButton

Hides the alarm button.

TimeFromInfoToVariable <'text'> <1|2> => **boolean value / Var1 or Var2 time value**

Search the string *text* on contents of the Info dialog and if successful returns the value *true* and the corresponding time value in the internal variable Var1 or Var2, otherwise it returns the value *false*.

SetWinAcqValsalvaOffset

Resets the base level of the Valsalva pressure signal.

SetWinAcqHandgripOffset

Resets the base level of the Handgrip pressure signal.

Multiply <Var1|Var2> <value>

Multiplies the internal variable var1 or Var2 by the *value*.

Nop

No operation.

ExportSignalToFile <'name'> <group> <mode> <version> <New|Old> <file name> <column mode> <MaxDataPointNum>

Exports the signal to the text file. Only those data points are exported which are between the x-cursors. The signal is exported to the new file if the option *New* is selected, otherwise it is appended to the end of the previous export file. If the new file is used and no *file name* is given, the normal Windows file open dialog is

opened and the user can give the name of the file. If the parameter *column mode* is 1, only the data values of the signal is written into the file (one column format), otherwise also the x-coordinate is written (two column format). If the parameter *MaxDataPointNum* is nonzero, the maximum number of data points is set by this parameter (within the x-cursors). Data values are written in the scientific format. If there are two columns, the column separator is tab character. The decimal separator of the data values is the same as used by the Windows system.

ReadMultiFileName

Opens the normal Windows file open dialog where the user can select one or more files. The selected file names are used by some other file operations, for example the command *ImportGenericASCIIRRI*. The number of selected files is set to the value of the internal toggle variable 1. This value can be used for creating operation loops. The toggle value determines the current file name from the name list.

ImportGenericASCIIRRI <ms|s> {CumTime}

Imports RRI time series which has the format of Generic ASCII RRI. The time format of the RRI value is set by the parameter *ms* or *s*. If the file includes the cumulative time column the option *CumTime* should be selected.

SaveAllStatistic <mode>

Saves all statistical results to the file. The file name is the same as the current WinCPRS file name but the extension is set by the parameter *Statistic Page file* found from the Options|General tabsheet. The parameter *mode* (1 .. 4) selects the output format of the statistical results.

3. Example of the protocol file

```
; Example analysis protocol for ECG morphology
; TK/Absolute Aliens Oy
;
    Phase Start
    Comment Clear
    AddLine
    AddLine
    Comment 'ECG analysis protocol' Center
    AddLine
    Comment 'Version 1.00' Center
    AddLine
    Comment 'ECG morphology' Center
    Check Cont/Stop

; OPEN THE DATA -----

    Phase Open data
    Comment 'Open the data file' CCenter Clear
    OpenData
    Check Cont/Stop

    FoundSignal ECG 1 0 1
    Skip Cont1
    Comment 'ECG signal can not be found' CCenter Clear
    Comment 'Protocol must be terminated' Center
    Check OK
    Break

Cont1:    FoundSignal RRI 2 0 1
    Skip Cont2
    Comment 'RRI signal can not be found' CCenter Clear
    Comment 'Protocol must be terminated' Center
    Check OK
    Break

; ANALYSIS -----

Cont2:    Phase Range checking
    HideSignal All
    ShowSignal ECG 1 0 1
    ShowSignal RRI 2 0 1
    SelectStatistics RRI 2 0 1 1
    CascadeSignals
    BringToFront RRI 2 0 1
    Comment 'Check that the time range on RRI is correct' CCenter Clear
    Check Cont/Stop

    Phase Advanced QRS morphology
    Comment 'Generating advanced QRS signals' CCenter Clear
    ShareLimits RRI 2 0 1
    Detect QRSAdvanced 1
    WaitFor QRSAdvanced
    CascadeSignals
```

Phase Statistics
Comment 'Calculating statistics' CCenter Clear
ShareLimits RRI 2 0 1

UpdateStatistics 'QTa time' 2 0 1
UpdateStatistics 'QT time' 2 0 1
UpdateStatistics 'QTc time' 2 0 1
UpdateStatistics 'TaT time' 2 0 1
UpdateStatistics 'RTa time' 2 0 1

FoundSignal 'PaR time' 2 0 1
Skip Found
Comment 'Can not found P-wave related signals'
Check Cont/Stop
Goto NoFound

Found: UpdateStatistics 'PaR time' 2 0 1
UpdateStatistics 'PaTa time' 2 0 1
UpdateStatistics 'PPI' 2 0 1

NoFound: UpdateStatistics 'T amplitude' 2 0 1
UpdateStatistics 'T area' 2 0 1
UpdateStatistics 'R amplitude' 2 0 1
UpdateStatistics 'R area' 2 0 1
UpdateStatistics 'R/T ratio' 2 0 1
UpdateStatistics 'ST level' 2 0 1

Phase Averaged ECG
AveragedECG

Phase Transfer results
Comment 'Transfer statistical results' CCenter Clear
ShowStatisticPage
Check Cont/Stop

; DATA SAVING -----

Phase Data save
Comment 'Save the data' CCenter Clear
SaveAsData
Check OK
CloseData

; END OF PROTOCOL -----

Phase Termination
Comment 'Analysis has been terminated' CCenter Clear
Check OK
Comment Clear
Check OK

4. Structure of the reference file

The reference file consists of reference data blocks, each of them corresponds to one variable. The blocks must be separated with at least one empty line. The comment lines must be separated with braces { }.

The structure of the reference data block is following:

```
Variable name { :Male|:Female }  
Minimum-age      Maximum-age  
Age1            Mimimum-value   Mean-value  Maximum-value  
Age2            Mimimum-value   Mean-value  Maximum-value  
...  
AgeN            Mimimum-value   Mean-value  Maximum-value
```

There can be separated reference data blocks for males and females marked with the options :Male or :Female. The Minimum-age and Maximum-age parameters are obligatory, and they must correspond to first and last reference data lines. There are no limitations for the step of the age between two lines, but the reference lines must be ordered in increasing age. If the age is lower than the Minimum-age, the Minimum-age values are used, and correspondingly for the age higher than the Maximum-age.

Examples:

RRI Mean

```
10 80  
10 800 1000 1100  
20 810 1000 1100  
30 820 1010 1200  
40 820 1010 1200  
50 820 1010 1200  
60 810 1000 1100  
70 800 1000 1100  
80 780 980 1080
```

MAP Mean:Male

```
10 100  
10 100 120 140  
20 105 125 145  
30 105 125 145  
40 105 120 140  
50 100 120 140  
60 100 110 135  
70 90 100 125  
80 90 100 120  
90 85 95 110  
100 85 95 110
```