**Overview**

Very often it’s very useful to see the nature of signals changes. Chart helps to take a glance from afar, notice potentially problematic ones and focus on their actual values.

ChartPonts is a trace tool giving the ability to define trace variables in Visual Studio design mode with subsequent view charts containing them during and after debug session.

**Limitations**

*Visual studio version*: MSVC 2015 (v14)

*Supported language*: C/C++

*Variables types*:

***Fundamental types***

short

short int

signed short

signed short int

unsigned short

unsigned short int

int

signed

signed int

unsigned

unsigned int

long

long int

signed long

signed long int

unsigned long

unsigned long int

long long

long long int

signed long long

signed long long int

unsigned long long

unsigned long long int

double

float

bool

signed char

char

unsigned char

***typedefs***

int8\_t

int16\_t

int32\_t

int64\_t

uint8\_t

uint16\_t

uint32\_t

uint64\_t

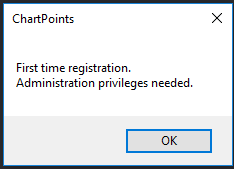
***Important:*** all other typedefs not supported

Only class variables of listed above types are supported.

*Trace points*: class methods only.

**Installation issues**

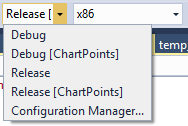
While first time Visual Studio loading after ChartPoints extension installation you will be prompted on additional installation steps.



If you are launching Visual Studio in non-administrative mode UAC will ask your confirmation.

**How it works**

New configurations based on appropriate existing ones are added: “Debug [ChartPoints]” and “Release [ChartPoints]” to both solution & project configurations.



As extra code is injected during build in this modes it is made specially to separate ordinal & “.. [ChartPoints]” outputs.

**Uninstallation issues**

Self-cleaning MSBuild task is added to \*.vcxproj files. While first build after ChartPoints extension removal it will remove all ChartPoints dependencies from \*.vcxproj file (no matter what configuration is used). Solution “.. [ChartPoints]” configurations are left. You have to remove them manually.

**Overhead (comparison of ChartPoints/ordinal execution time)**

The main goal was to minimize overhead produced by trace capabilities injection.

*Tests*

3 simultaneously running threads

10 ChartPoints per thread

Delay between thread cycles (1 millisecond - 1000 Hz [accuracy depends on OS/hardware])

Result: no overhead was detected

Surely, if you will try to trace without delay on thread cycles the overhead will be significant (“Too much <../trace> will kill you. ©”

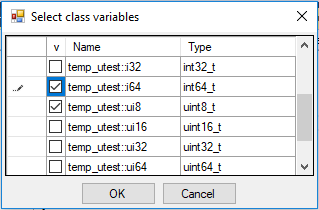
**How-to..**

*Insert/Remove ChartPoint(s)*

In code editor inside class method definition open context menu and select <***ChartPoints / Insert ChartPoint***>.



Window with all available class variables will be opened. Here you can select/deselect needed ones.



After ChartPoint(s) were added glyph  will appear on the left side of code editor window.

Selected ChartPoints will be added at the beginning of the line or immediately after ‘{‘ if you are staying at the line with beginning of the method body.

***Important***

* All ChartPoints information is actual on per solution basis (saved in \*.suo [Solution User Options] file). So moving the project to another place leads to their loss
* Only simple editing tracking is performed in design mode (add/remove lines before ChartPoint). They will be validated on build/load
* Non-validated ChartPoints will not be saved
* Changes in ChartPoints list are not tracked, so it is needed to perform rebuild after any changes in it

***Glyphs***

 - ChartPoint is switched on

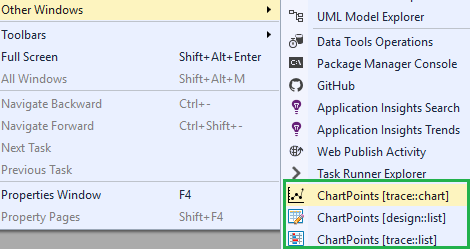
 - ChartPoint is switched off

- ChartPoint is unavailable (detected during build)

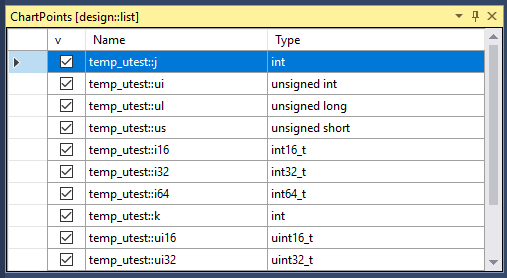
- combination of ChartPoints statuses

*Tool Windows*

<***Main Menu / View / Other Windows***>



*ChartPoints [design:list]*



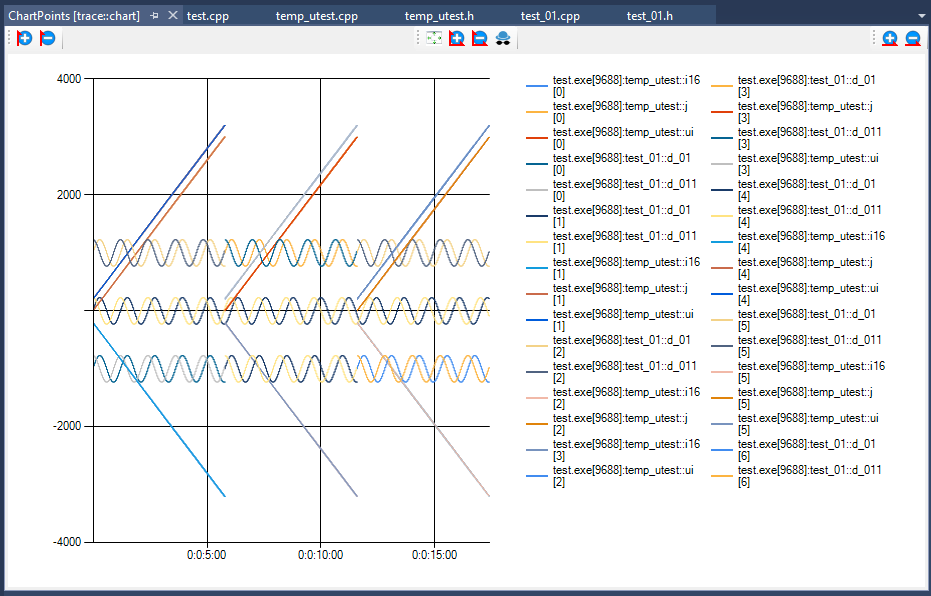
This is a place where all toggled ChartPoints are listed.

Double click on the line will open corresponding file where ChartPoint is placed and puts cursor on the corresponding line.

Checkbox allows temporary make ChartPoint unavailable (exclude from build without deleting)

<Del> key pressed on selected line will remove ChartPoint

*ChartPoint [trace:chart]*



This window contains chart with legends.

X – time axis

Y – value axis

Legend names:



chart line color | executable name | [process ID] | variable unique name | [variable instance ordinal number (in order of their initialization)]

 - zooming in/out of axises or full chart



Moving mouse wheel will zoom chart relative to cursor position.

Pressing left mouse button allows to select rectangle area and zoom to it.

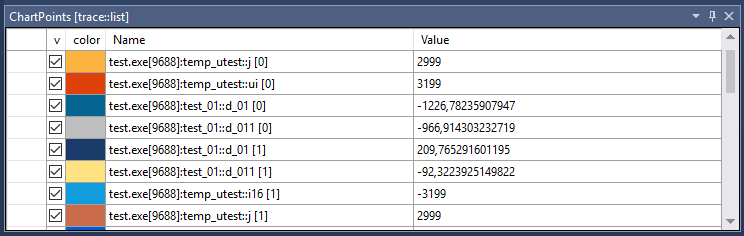
Moving cursor in chart area with right mouse button pressed scrolls current chart area if zoomed.

 - fit view

- spy mode. When this button is toggled you can move vertical cursor line along time axis and watch corresponding values in “ChartPoints [trace:list]”



*ChartPoints [trace:list]*



Contains traced variables and their values.

Checkbox allows to temporary exclude corresponding variable from chart

ChartPoint [trace:chart] & ChartPoints [trace:list] are operable during debug session & after ending it till next run.