# EDA Debugger

EDA debugger is used for debugging calculation of calculated vectors and functions of EDA subsystem. This feature is included in a group of graphic objects for pictures as EDA debugger.

This graphic object enables, directly in its interface, to connect to any EDA calculation in application. Calculation means a loading of calculated vector or calculation of function. Whole process of debugging is controlled in the graphical interface, i.e. in the EDA debugger.

Other method of usage of EDA debugger is initiation of calculation for debugging directly from ESL script. This method requires supporting functions of EDA debugger. In first step, user must create for debugging connection with process that executes EDA calculations. To establish this connection, use the function %HI_EDADebugOpen. The function %HI_EDADebugClose closes this connection. After creating debug connection, user may start calculation of calculated vector for debugging by calling the function %HI_EDADebugVectorRec. When calling %HI_EDADebugCalcFunctionRec, it activates debugging calculation of EDA-L formula. Functions %HI_EDADebugVectorRec and %HI_EDADebugCalcFunctionRec do not wait for finishing the calculation that was started by these functions.

Example of EDA debugger that is used in a picture opened in HI:
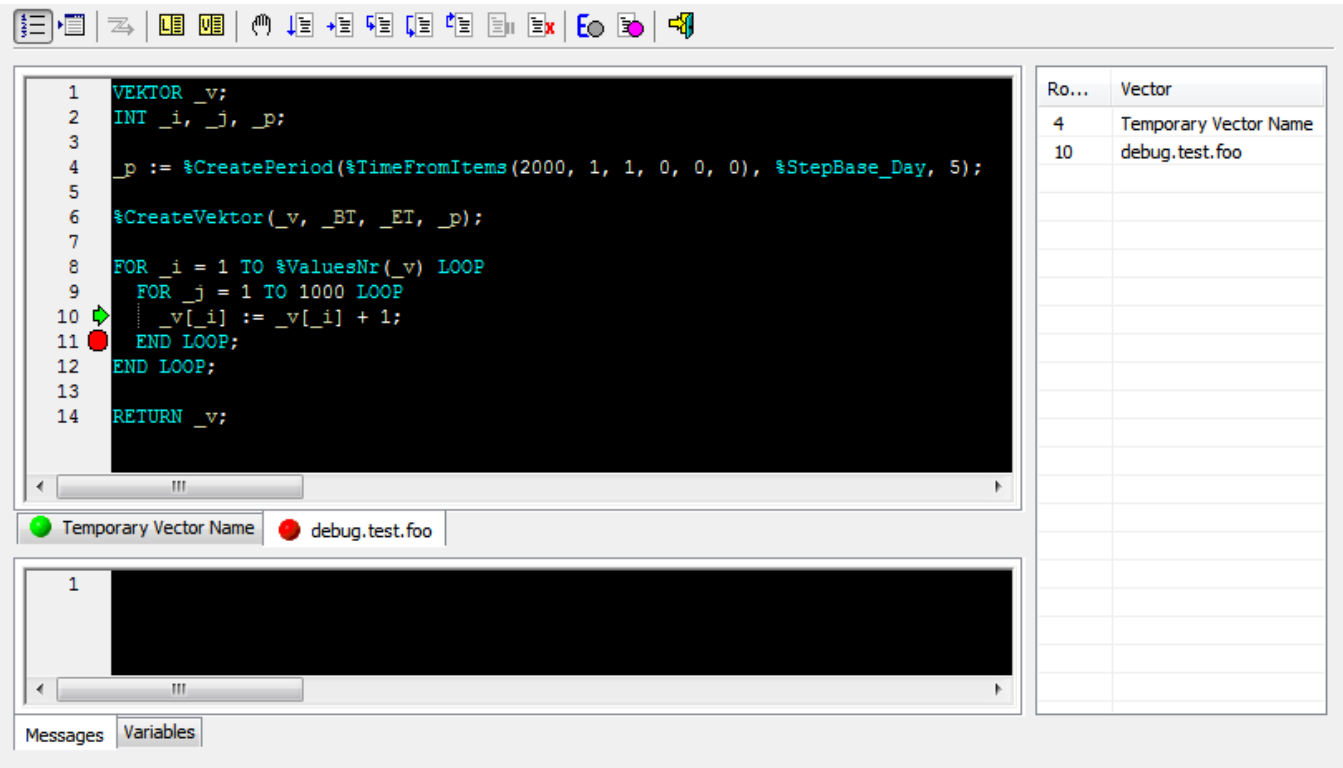
It consist of:

- Toolbar – contains buttons that represents debug commands.
- Tabs with calculations – contains source formulas of vector calculation.
- Info tabs:
    - *Messages* – contains messages that have been generated during calculation.
    - *Variables* – contains monitored variables of selected vector.
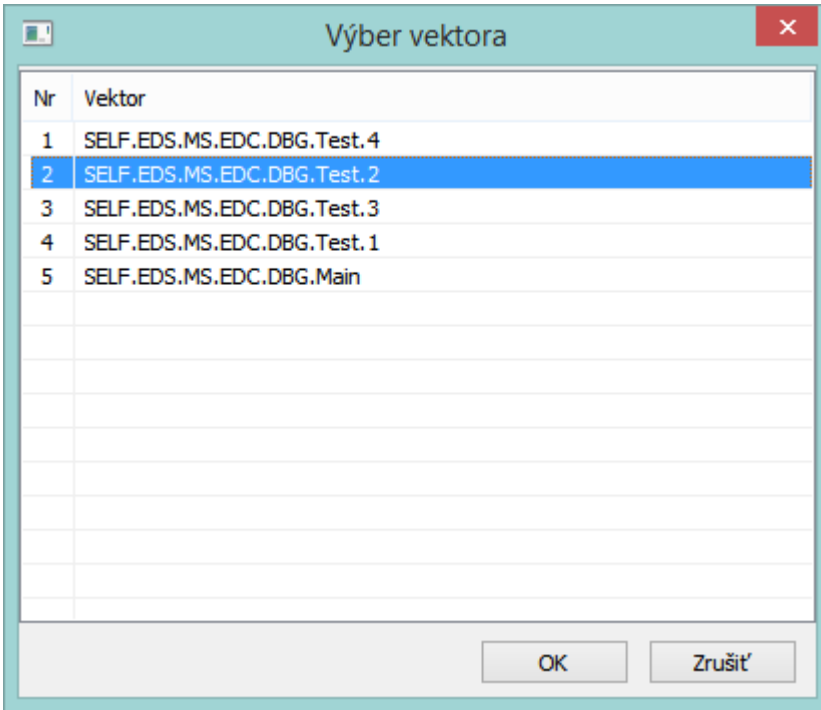- Call stack – contains current list of nested vectors of aborted calculation.

Toolbar

Buttons represents the commands that control debugging of calculation.
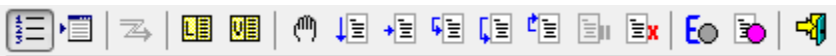
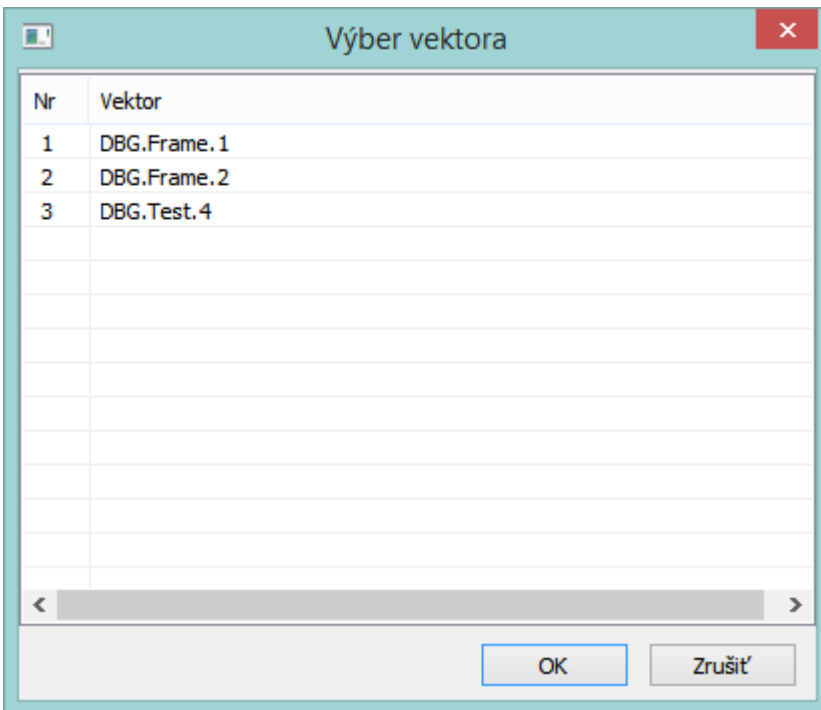|	
	Enable / disable numbering of rows in the formula.
	Go to row in displayed vector.

|	
	Connect to calculation – connection to running calculation which should be debugged. Dialog window displays the list of all running EDA calculations in the application with information such as on which EDA server the calculation runs, from which client was started, initiate vector that is calculated.
	List of local variables in displayed vector - enables to add variables into **Variable** tab.
	List of vectors used when debugging (closed vector included) – enables to open (select) vector which will be displayed.
	Icon that informs status of calculation: 1. Grey – calculation is stopped, 2. Red – calculating.
	Continue in the calculation.

|	
	Continue in calculation up to row with cursor (if on the row is not any script command, calculation continues up to next row with command).
	Step inside – if the row contains loading of calculated vector, process continues within it.
	Step outside – if the row contains loading of calculated vector, loading is executed and calculation stops on next row.
	Continue up to return from vector.
	Abort calculation.
	Stop calculation.
	Aborted when occurring an exception – indicates two statuses differentiated by color: 1. Grey – do not abort, 2. Red – abort.
	Delete all breaks in all vectors.
	Finish of calculation debugging (calculation will continue).

Tabs with calculation

Tabs contains the vector formulas that were used when debugging of connected calculation. Vector formula occurs in tabs automatically after aborting the calculation - tabs contain all vectors that are in current call stack. User may manually close (hide) the tab with vector formula and then open again (via list of vectors).
Tabs shows name and version of vector and indicates its status (3 options):

- Grey circle – vector, at the time of aborting calculation, does not occurs in call stack.
- Green circle – vector, at the time of aborting calculation, occurs in current call stack.
- Red circle – vector, at the time of aborting calculation, occurs in current call stack and contains a command that will be executed.
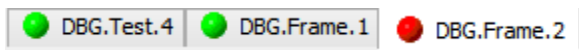
Tab with vector formula
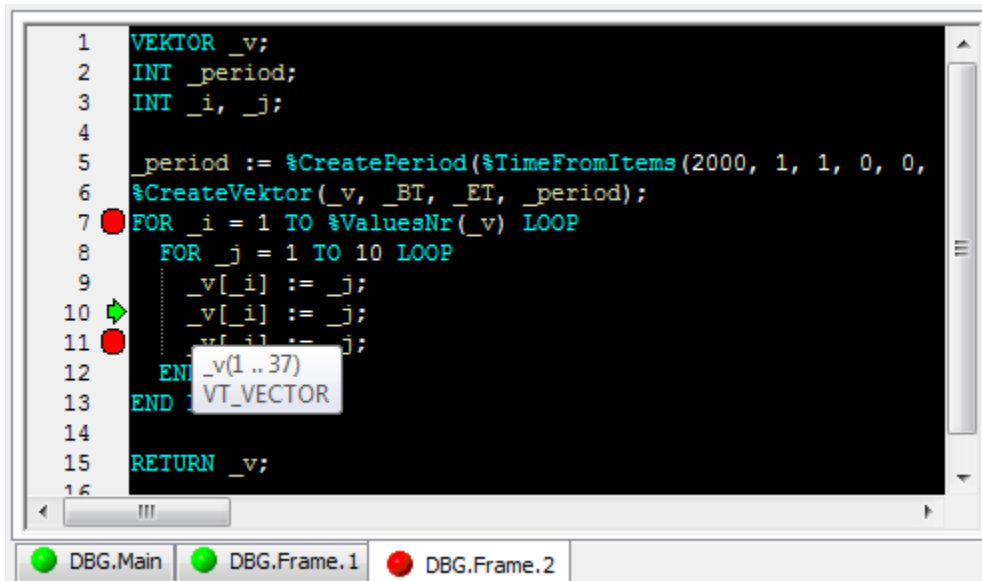
It contains:

- Optional row numbering
- Breakpoints – red circles
- Mark for aborting execution of calculation – green arrow
- Vector formula with syntax highlighting:

- key words and functions – turquoise color,
- local variable – yellow color,
- delimiters – white color.
  - Description of variable and its value (for one-element variables) with cursor.

Popup menu that occurs over formula enables:

- to add selected local variable into the list of watched variables of vector,
- to hide tabs of vector – watching variables and breakpoints stays activated. Tabs may be displayed again via list of vectors.

Info tabs

Messages

Tab contains the messages that were generated during calculation. User may write some notes in the tab and delete the selected texts.

Variable

Tab that contains the values of watched local variables of selected vector. The sign +/- collapses / expands the vector variables.

+/-	Variable	Value type	Value	Time	N_A	Valid	Manual	Weak	Attribute A	Attribute B	Attribute C	Attribute D	Locked
	_p	VT_INT	-1001	N/A	☐	☑	☐	☐	☐	☐	☐	☐	☐
-	_v (1 - 38)	VT_VECTOR											
	|- _v(1)	VT_VECTOR	1.00000000000000E+05	28-12-2015 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(2)	VT_VECTOR	1.00000000000000E+05	02-01-2016 00:00:00	☐	☑	☐	☑	☐	☐	☐	☐	☐
	|- _v(3)	VT_VECTOR	1.00000000000000E+05	07-01-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(4)	VT_VECTOR	1.00000000000000E+05	12-01-2016 00:00:00	☐	☑	☐	☐	☐	☑	☐	☐	☐
	|- _v(5)	VT_VECTOR	1.00000000000000E+05	17-01-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(6)	VT_VECTOR	500	22-01-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(7)	VT_VECTOR	1.00000000000000E+05	27-01-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(8)	VT_VECTOR	1.00000000000000E+05	01-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(9)	VT_VECTOR	1.00000000000000E+05	06-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(10)	VT_VECTOR	1.00000000000000E+05	11-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(11)	VT_VECTOR	1.00000000000000E+05	16-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(12)	VT_VECTOR	1.00000000000000E+05	21-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐
	|- _v(13)	VT_VECTOR	1.00000000000000E+05	26-02-2016 00:00:00	☐	☑	☐	☐	☐	☐	☐	☐	☐

Messages | Variables

It is filled from the list of local variables via command line or popup menu that appears over selected variable in the window with vector formula.

Nr	Premenné	Typ	Parametre
1	_BT	VT_TIME	
2	_ET	VT_TIME	
3	_STEP	VT_INT	
4	_i	VT_INT	
5	_j	VT_INT	
6	_period	VT_INT	
7	_v(1 .. 37)	VT_VECTOR	bIntegral => FALSE, (beginTime => 2013-12-27T23:00:00.000Z, timeZone => 3600, stepSize => (base => STEPBASE_DAY, count => 10))

Variables, the values of which are already watched, are differentiated by color (turquoise color). Selecting the variable of Array type, user may enter a range of values that will be watched:

Selection may be defined by range and name. To display the items to the last one, enter the zero at the end of range, e.g. 3,11-14,18,25-0.

Name	ValueType	Value	N_A	Valid	Manual	Weak	Flag A	Flag B	Flag C	Flag D	Locked
_intVar	VT_INT	0									
_realVar	VT_REAL	5.00000000000000E-01									
_boolVar	VT_BOOL	TRUE									

The monitored variable / group of variables can be removed from the list with the help of popup menu. The value can be changed directly in the list of watched variables.
Popup menu that appears over the list of variables enables to:

- remove selected variable / group of variables,
- remove all variables,
- define new values for variables,
- delete the change of value of selected variables.

Call stack

Call stack displays the current list of vector nesting of aborted calculation. The first vector of the calculation is at the top of list. Double clicking over the list enables to select the tab with vector formula. The row, on which the calculation was aborted, will be selected automatically.

Example of starting calculation of vector for debugging from script

```
INT _serverId, _clientId
INT _retCode

ENTRY BTN_DebugVectorRec_OnClick
 RECORD NOALIAS (SD._System_HI_EDADebugVectorRec) _paramsRec
 TIME _bt, _et

; connection to process that executes EDA calculations
 _retCode := %HI_EDADebugOpen(_DBG, _serverId, _clientId)

_bt := %StrToTimeEx("2015-01-01 00:00:00", "rrrr-mm-dd hh:mi:ss")
_et := %StrToTimeEx("2015-02-01 00:00:00", "rrrr-mm-dd hh:mi:ss")
_paramsRec[1]^structVersion := 1

; activating calculation of vector for debugging
_retCode := %HI_EDADebugVectorRec(_DBG, "DBG.Main", _bt, _et, _paramsRec\HBJ)
END BTN_DebugVectorRec_OnClick

ENTRY BTN_DebugClose_OnClick
; closing debug connection
_retCode := %HI_EDADebugClose(_DBG)
END BTN_DebugClose_OnClick

BEGIN
; getting ID of process EDA Client, which uses this HI, and EDA Server, to which the client is connected
; it is expected that connection to EDA already exists
CALL %EDA_GetProcessId(_serverId, _clientId, _retCode)
END
```

ⓘ **Related pages:**

Graphic object manipulation functions