Edelwise™

beWISE - TOOLS

User MANUAL

(Rev. B)

Title: User Manual for beWISE TOOLS.

Type of documentation: Manual

Date of creation: 02/07/2007

This technical description replaces all previous versions.

Document history

Date	**Name**	**Revision**	**Modification**
2/7/2007	edelweiss	A	First version
2/10/2007	edelweiss	B	Second version

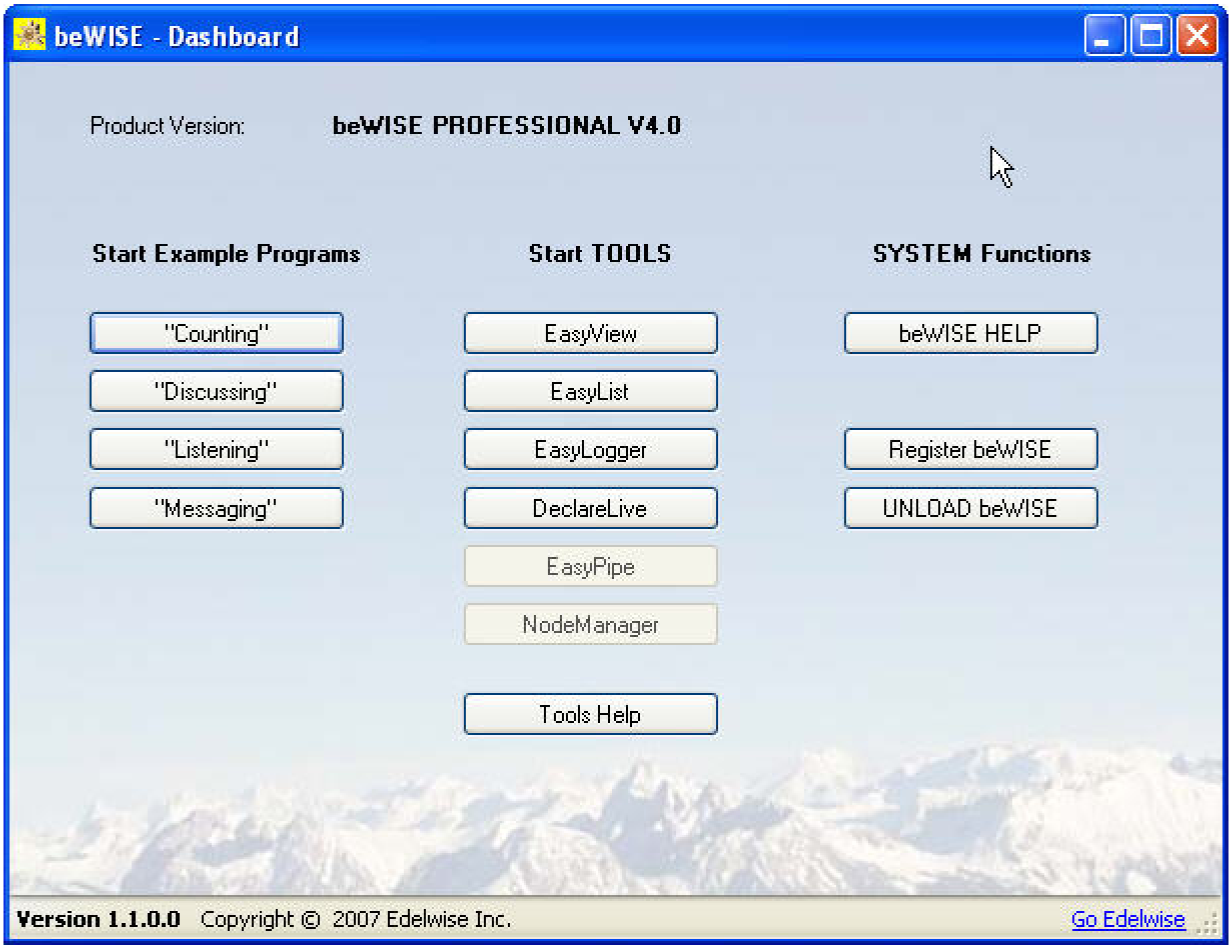
1 TABLE OF CONTENTS

2 TOOLS FOR BEWISE BASIC

2.1 DASHBOARD

The dashboard is a little board which shows you which product version you are registered to and gives you the possibility to start all beWISE example programs and beWISE tools which come with your product version.

Further you can call beWISE help, register beWISE and unload it.



Product Version: beWISE PROFESSIONAL V4.0

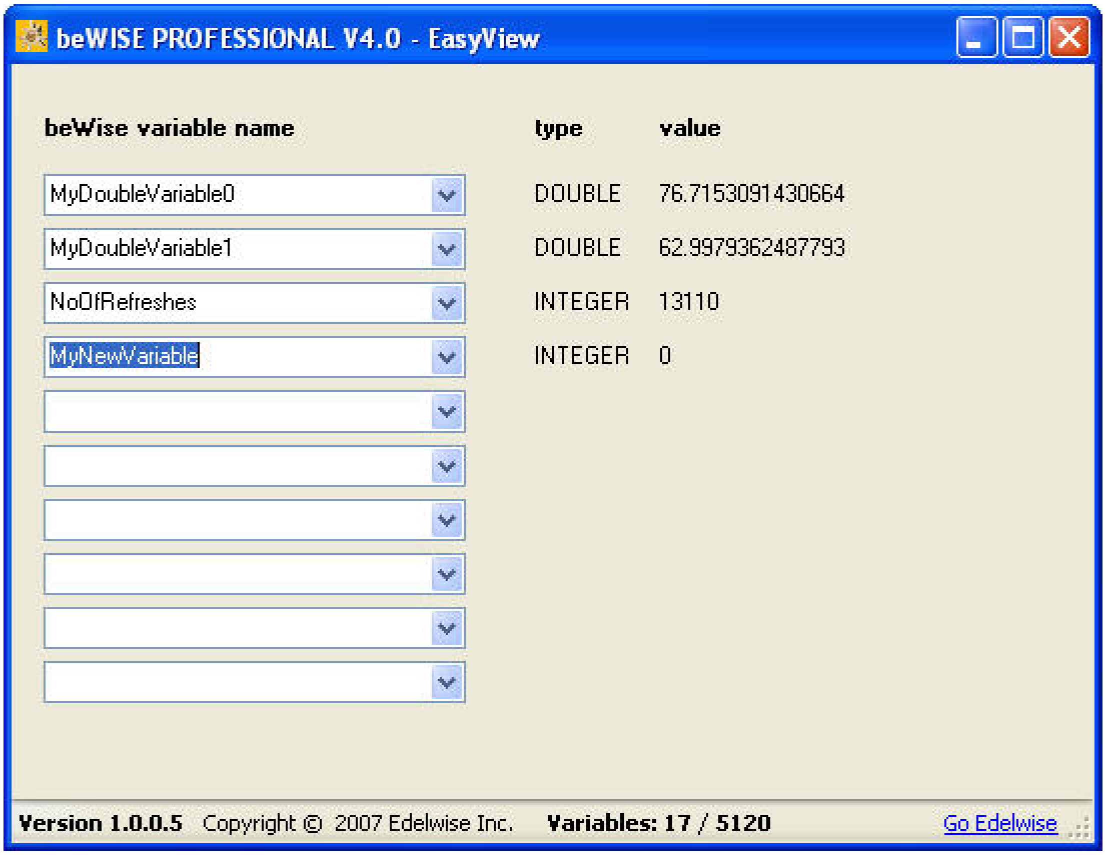
Version 1.1.0.0 Copyright © 2007 Edelwise Inc.

3 TOOLS FOR BEWISE ADVANCED

3.1 EASYVIEW

beWISE EasyView is a simple viewer of beWISE variables.

You can specify up to 10 different beWISE variables and EasyView will watch them for you.



beWISE PROFESSIONAL V4.0 - EasyView

beWise variable name	type	value
MyDoubleVariable0	DOUBLE	76.7153091430664
MyDoubleVariable1	DOUBLE	62.9979362487793
NoOfRefreshes	INTEGER	13110
MyNewVariable	INTEGER	0

Version 1.0.0.5 Copyright © 2007 Edelwise Inc. Variables: 17 / 5120 Go Edelwise

To choose a beWISE variable just click on one of the combo boxes. It will show you existing beWISE variables.

Choose one and you will see the type and value of this variable.

As long as this beWISE VARIABLE exists on your computer it will be refreshed and will show you the actual value. In case you delete a variable the last known value will remain and will not be changed any more.

The refresh rate is fixed to 10 times a second.

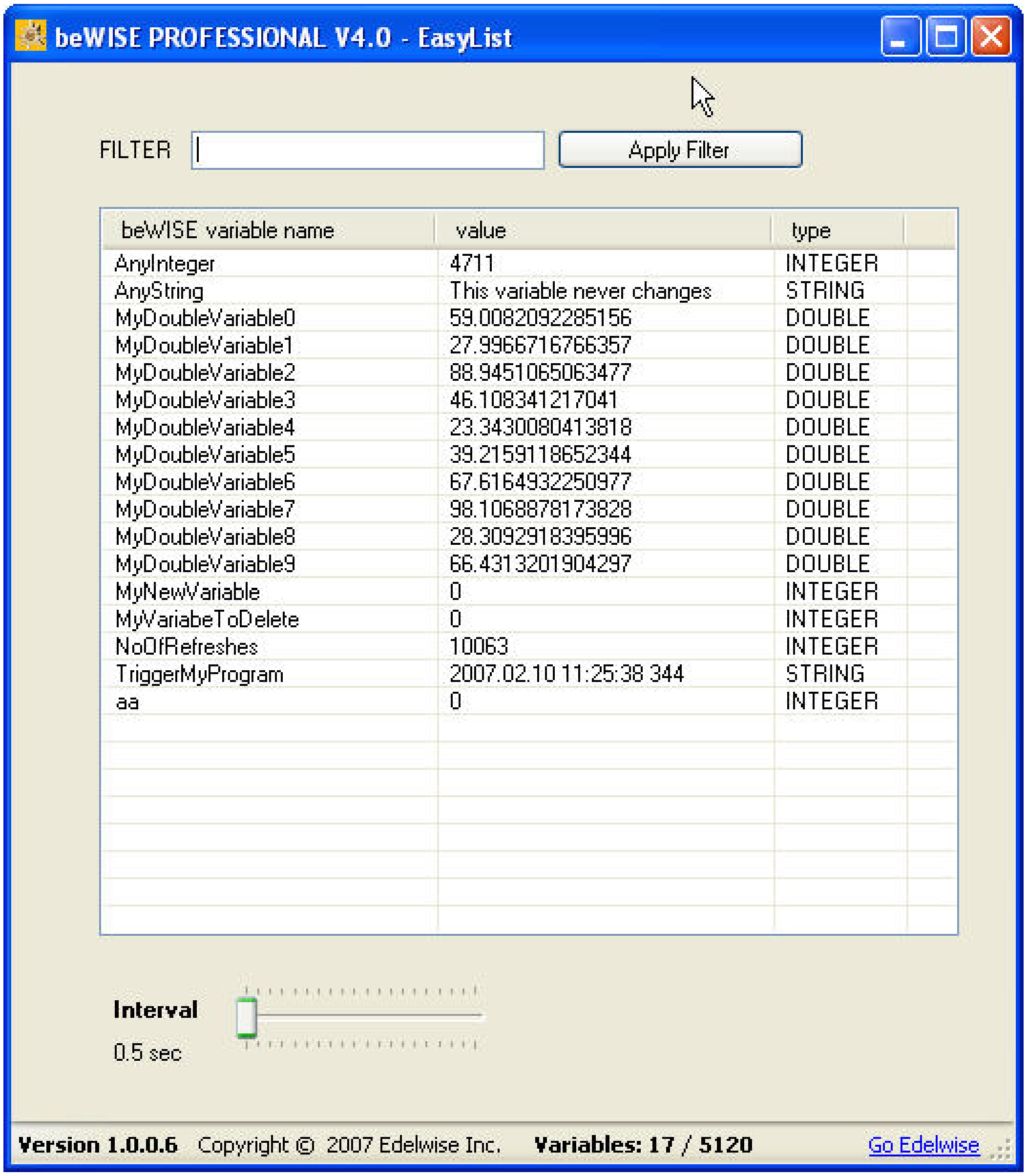
In the tool strip status line EasyView indicates you the number of available variables in your computer.

3.2 EASYLIST

beWISE EasyList is a simple viewer of beWISE variables.

In opposite to EasyView EasyList makes it even easier for you to. You do not even choose which beWISE variable you want to see.

This is the standard view when you open EasyList. The FILTER is empty, which means – show me all existing variable.



beWISE variable name	value	type
AnyInteger	4711	INTEGER
AnyString	This variable never changes	STRING
MyDoubleVariable0	59.0082092285156	DOUBLE
MyDoubleVariable1	27.9966716766357	DOUBLE
MyDoubleVariable2	88.9451065063477	DOUBLE
MyDoubleVariable3	46.108341217041	DOUBLE
MyDoubleVariable4	23.3430080413818	DOUBLE
MyDoubleVariable5	39.2159118652344	DOUBLE
MyDoubleVariable6	67.6164932250977	DOUBLE
MyDoubleVariable7	98.1068878173828	DOUBLE
MyDoubleVariable8	28.3092918395996	DOUBLE
MyDoubleVariable9	66.4313201904297	DOUBLE
MyNewVariable	0	INTEGER
MyVariabeToDelete	0	INTEGER
NoOfRefreshes	10063	INTEGER
TriggerMyProgram	2007.02.10 11:25:38 344	STRING
aa	0	INTEGER

Interval
0.5 sec

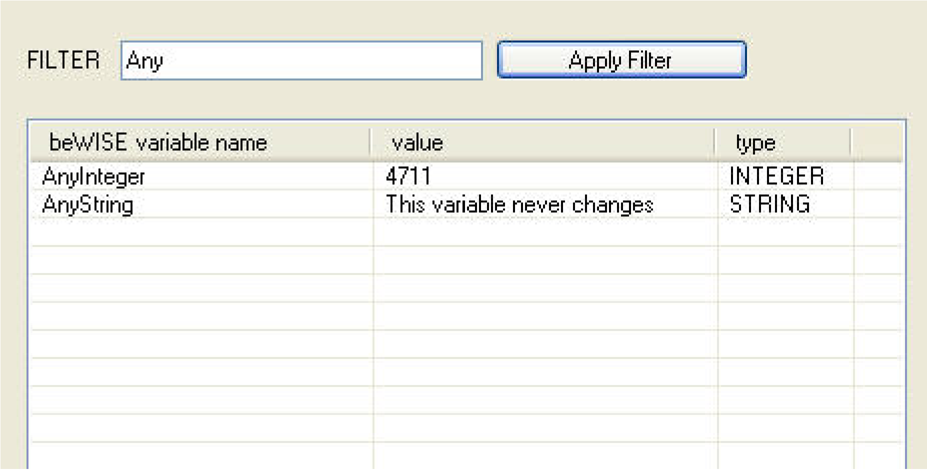
Version 1.0.0.6 Copyright © 2007 Edelwise Inc. **Variables: 17 / 5120**

EasyList - Filter:

In the filter you can enter a trailing part of the beWISE variable names and from there on you will see only these beWISE variables which meet the filter specified.

Example:

We entered “Any” as filter and pressed the Apply Filter button. From here on we see only those variables which name starts with the “Any”.



FILTER Any

beWISE variable name	value	type
AnyInteger	4711	INTEGER
AnyString	This variable never changes	STRING

Backside:

This seems quite a faster method as EasyView. But it has its backside as well:

- You have to declare your variables (which we encourage anyways) in logical groups you might want to see in EasyList in a glance.

4 TOOLS FOR BEWISE PROFESSIONAL

4.1 EASYLOGGER

The first part of the EasyLogger is based on the EasyView tool.

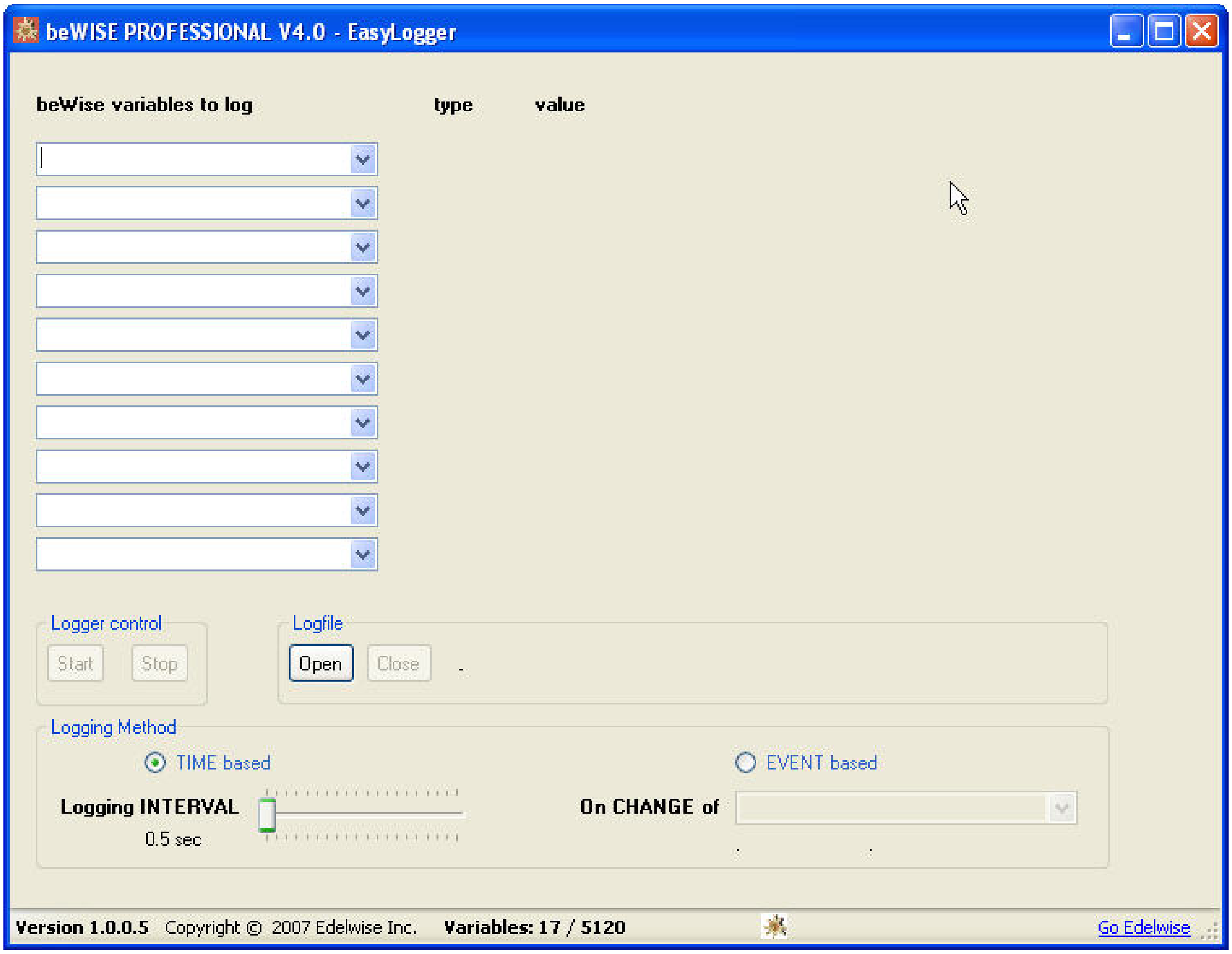
You can specify up to 10 different beWISE variables and EasyLogger will watch them for you, like you are used to from EasyView. But the EasyLogger goes some steps further:

1. EasyLogger lets you decide on how the refresh should be done:

   - by timer event (which you can choose by a slider, time frames between 0.5 – 10 seconds)
   - by change event of a beWISE variable of your choice

2. EasyLogger lets you log all variables into a *.csv file.

Here a standard view of the tool:



● TIME based

Logging INTERVAL 0.5 sec

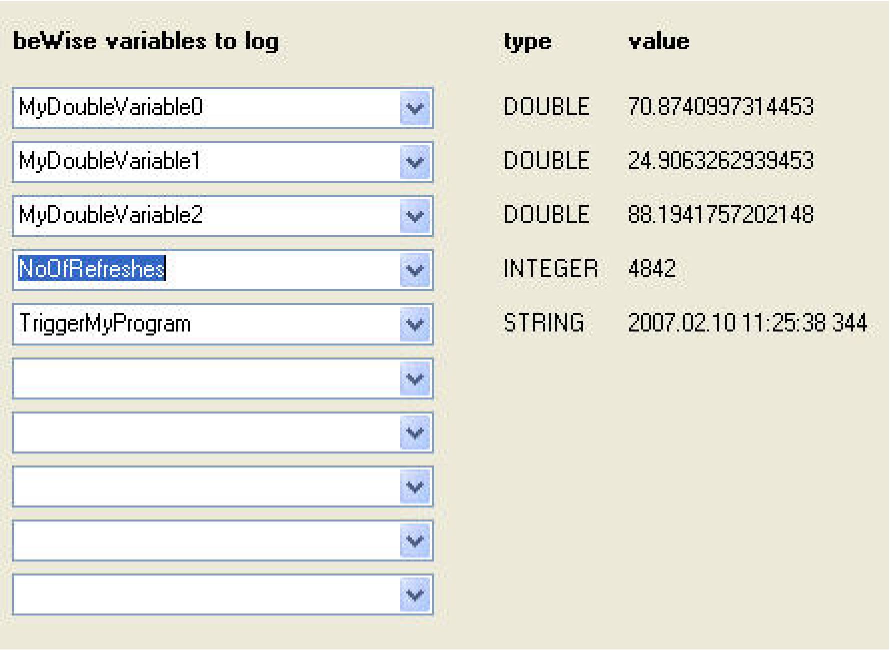
Version 1.0.0.5 Copyright © 2007 Edelwise Inc. Variables: 17 / 5120

4.1.1 EASYLOGGER - BEWISE VARIABLES TO WATCH

In the upper part of the EasyLogger tool you can choose beWISE variables you want to watch by just clicking on one of the combo boxes. When the combo box opens it will show you all existing beWISE variables at this moment.

Choose one and you will see the type and value of this variable.

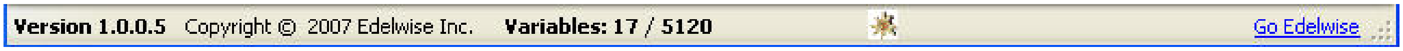
As long as this beWISE VARIABLE exists on your computer it will be refreshed and will show you the actual value. In case you delete a variable (in any other of your applications) the last known value will remain and will not be changed any more.



beWise variables to log	type	value
MyDoubleVariable0	DOUBLE	70.8740997314453
MyDoubleVariable1	DOUBLE	24.9063262939453
MyDoubleVariable2	DOUBLE	88.1941757202148
NoOfRefreshes	INTEGER	4842
TriggerMyProgram	STRING	2007.02.10 11:25:38 344

4.1.2 EASYLOGGER - NUMBER OF BEWISE VARIABLES IN YOUR SYSTEM

In the tool strip status line EasyLogger indicates the number of available / maximum number of variables on your system.



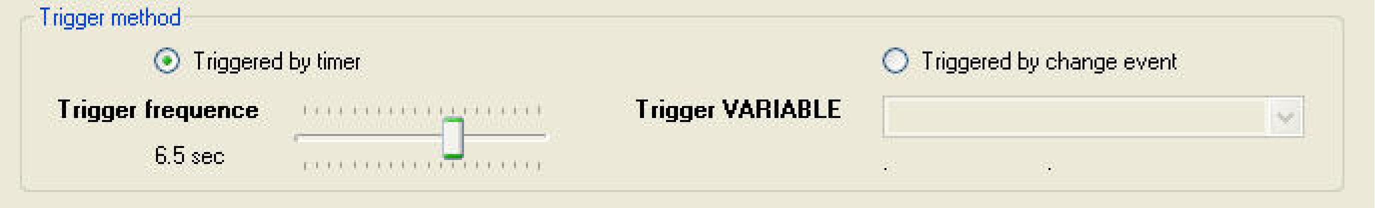
Version 1.0.0.5 Copyright © 2007 Edelwise Inc. **Variables: 17 / 5120**

4.1.3 EASYLOGGER - TRIGGER METHOD

One of the advantages of EasyLogger is that you are able to choose by yourself on when the variables displayed will be refreshed and written into the log file (see below).

The trigger method can be:

Triggered by timer
Choose with the slider a time interval between 0.5 seconds – 10 seconds (e.g. every 6.5 seconds the refresh (and the logging) will be done)



Trigger method

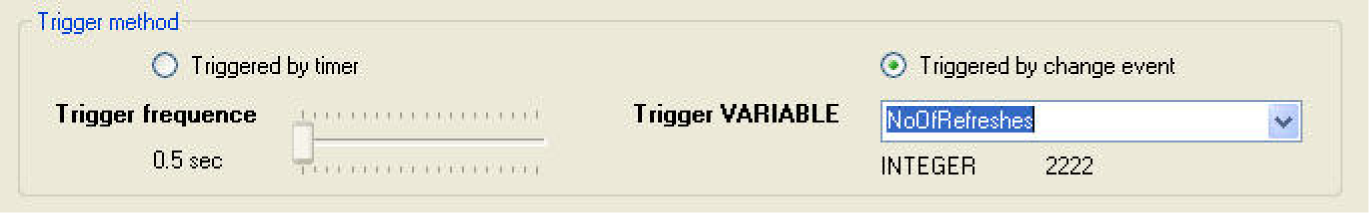
● Triggered by timer

○ Triggered by change event

Trigger frequence 6.5 sec

Trigger VARIABLE

Triggered by change event
Choose your beWISE variable which will act as refresh trigger when it changes. You also see the variable's type and the value underneath.
Every change of this variable triggers will refresh the display and writes a log record into the log file. If the variable does not change then there is no refresh of all variables defined above, neither will be anything written into the log file.



Trigger method

○ Triggered by timer

● Triggered by change event

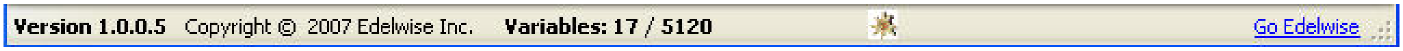
Trigger frequence 0.5 sec

Trigger VARIABLE NoOfRefreshes

INTEGER 2222

Trigger indicator

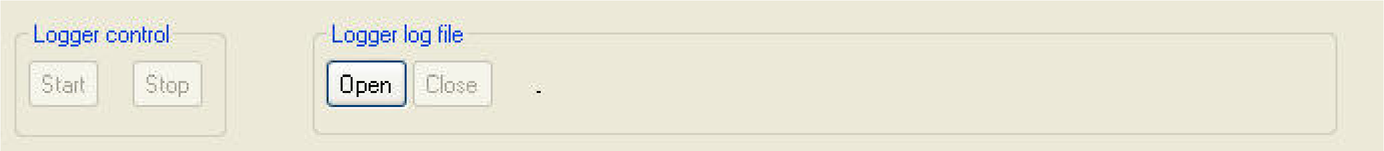
Just as a visual indicator of triggers (time or event triggers) are fired you will see a little edelweiss in the tool strip status line.



Version 1.0.0.5 Copyright © 2007 Edelwise Inc. **Variables: 17 / 5120**

4.1.4 EASYLOGGER - LOGGING INTO THE LOG FILE

The second big advantage of EasyLogger to the tools before is the possibility to write into a log file.



You got two control groups here:

Logger log file control

With **open** you can open a log file in a directory of your choice. You can create a new log file, or you can open an existing one. The contents of an existing one will be appended by the new logs. The file format is a csv file. Which is a comma separated sequential file, but has the advantage that MS Excel can open it immediately as well. So testing or documentation of any tests done is much more convenient for you.

With **close** you can (have to) close an open log file before you want to start with a new one.

Logger control

With **start** you start writing into the log file.

With **stop** you stop writing into the log file.

You can start / stop logging as many times as you want as long a log file is open. You can close a log file only when you stopped logging.

Remarks:
The logger does not influence the refresh of the variables you watch. So you can stop logging, watch the variables for a while and later on you start logging again.

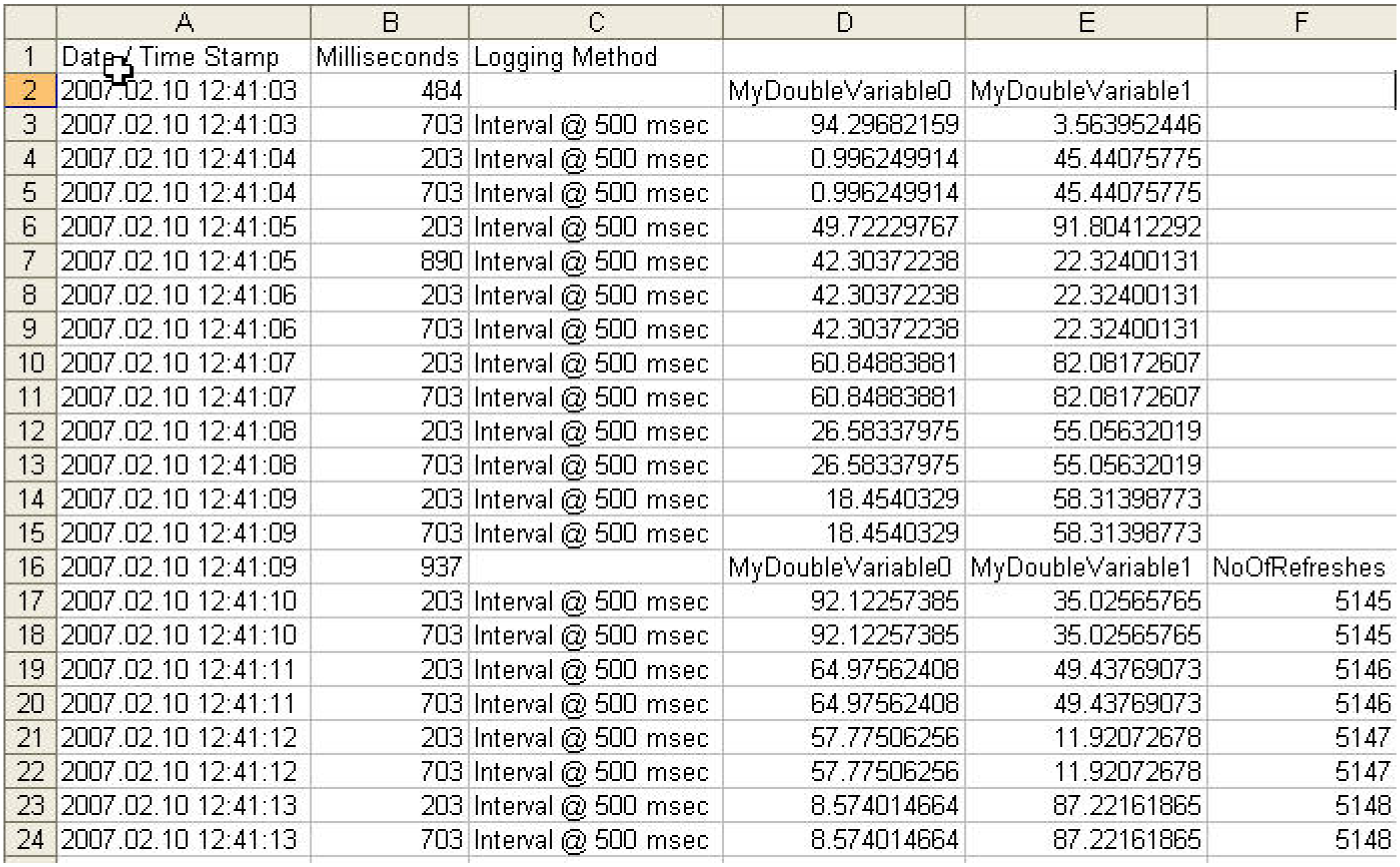
You can change the beWISE variables to be watched during logging is active. A header line with the name of all variables will be written into the log files every time you change the watcher's contents.

You can change from time- to event triggered watching during logging is active. A header line is written as well into the log file. Also the event trigger variable's name will be written into the log file.

4.1.5 EASYLOGGER - LOG FILE CONTENTS

Now let’s see a couple of examples of a log file:

Example A: Triggered by timer



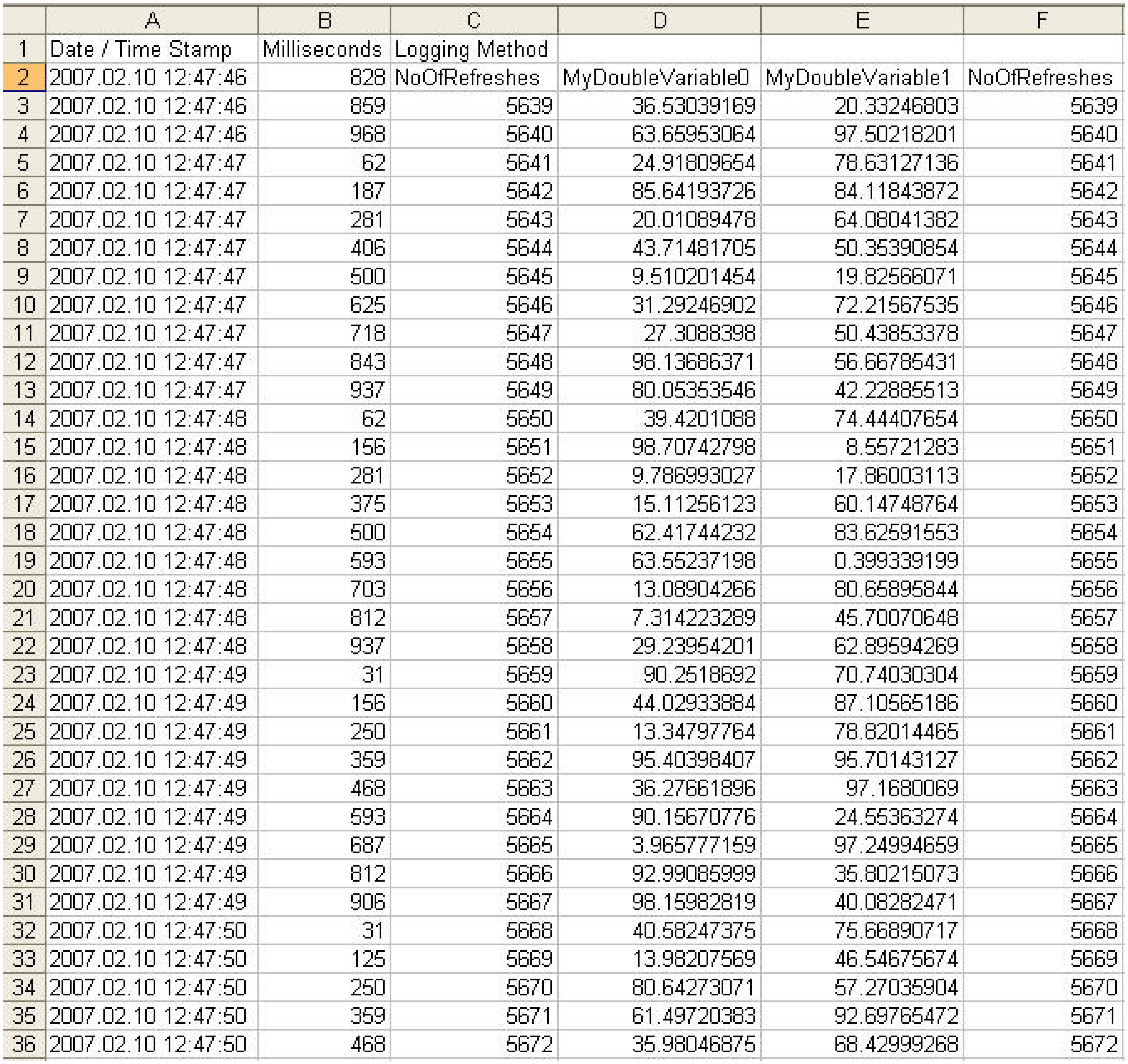
	A	B	C	D	E	F
1	Date / Time Stamp	Milliseconds	Logging Method			
2	2007.02.10 12:41:03	484		MyDoubleVariable0	MyDoubleVariable1	
3	2007.02.10 12:41:03	703	Interval @ 500 msec	94.29682159	3.563952446	
4	2007.02.10 12:41:04	203	Interval @ 500 msec	0.996249914	45.44075775	
5	2007.02.10 12:41:04	703	Interval @ 500 msec	0.996249914	45.44075775	
6	2007.02.10 12:41:05	203	Interval @ 500 msec	49.72229767	91.80412292	
7	2007.02.10 12:41:05	890	Interval @ 500 msec	42.30372238	22.32400131	
8	2007.02.10 12:41:06	203	Interval @ 500 msec	42.30372238	22.32400131	
9	2007.02.10 12:41:06	703	Interval @ 500 msec	42.30372238	22.32400131	
10	2007.02.10 12:41:07	203	Interval @ 500 msec	60.84883881	82.08172607	
11	2007.02.10 12:41:07	703	Interval @ 500 msec	60.84883881	82.08172607	
12	2007.02.10 12:41:08	203	Interval @ 500 msec	26.58337975	55.05632019	
13	2007.02.10 12:41:08	703	Interval @ 500 msec	26.58337975	55.05632019	
14	2007.02.10 12:41:09	203	Interval @ 500 msec	18.4540329	58.31398773	
15	2007.02.10 12:41:09	703	Interval @ 500 msec	18.4540329	58.31398773	
16	2007.02.10 12:41:09	937		MyDoubleVariable0	MyDoubleVariable1	NoOfRefreshes
17	2007.02.10 12:41:10	203	Interval @ 500 msec	92.12257385	35.02565765	5145
18	2007.02.10 12:41:10	703	Interval @ 500 msec	92.12257385	35.02565765	5145
19	2007.02.10 12:41:11	203	Interval @ 500 msec	64.97562408	49.43769073	5146
20	2007.02.10 12:41:11	703	Interval @ 500 msec	64.97562408	49.43769073	5146
21	2007.02.10 12:41:12	203	Interval @ 500 msec	57.77506256	11.92072678	5147
22	2007.02.10 12:41:12	703	Interval @ 500 msec	57.77506256	11.92072678	5147
23	2007.02.10 12:41:13	203	Interval @ 500 msec	8.574014664	87.22161865	5148
24	2007.02.10 12:41:13	703	Interval @ 500 msec	8.574014664	87.22161865	5148

What we got here:

We see a **timestamp** (“Date/Time Stamp”, “Milliseconds”) in ***column A*** and ***column B***.
We see a **trigger variable** (“Logging method”) in ***column C***. In our case here this is always set to timer - meaning it was triggering with timer events. Exactly every 500 milliseconds (0.5 seconds) a record was written.

In ***row 2*** we started the logging while watching two beWISE variables “MyDoubleVariable0” and “MyDoubleVariable1”. EasyLogger has written a header line to let us know which variables we were watching.
At ***row 16*** we decided to watch the beWISE variable “NOOfRefreshes” also from now on. That’s why we have here again a header line.

Example B: Triggered by change event



	A	B	C	D	E	F
1	Date / Time Stamp	Milliseconds	Logging Method			
2	2007.02.10 12:47:46	828	NoOfRefreshes	MyDoubleVariable0	MyDoubleVariable1	NoOfRefreshes
3	2007.02.10 12:47:46	859	5639	36.53039169	20.33246803	5639
4	2007.02.10 12:47:46	968	5640	63.65953064	97.50218201	5640
5	2007.02.10 12:47:47	62	5641	24.91809654	78.63127136	5641
6	2007.02.10 12:47:47	187	5642	85.64193726	84.11843872	5642
7	2007.02.10 12:47:47	281	5643	20.01089478	64.08041382	5643
8	2007.02.10 12:47:47	406	5644	43.71481705	50.35390854	5644
9	2007.02.10 12:47:47	500	5645	9.510201454	19.82566071	5645
10	2007.02.10 12:47:47	625	5646	31.29246902	72.21567535	5646
11	2007.02.10 12:47:47	718	5647	27.3088398	50.43853378	5647
12	2007.02.10 12:47:47	843	5648	98.13686371	56.66785431	5648
13	2007.02.10 12:47:47	937	5649	80.05353546	42.22885513	5649
14	2007.02.10 12:47:48	62	5650	39.4201088	74.44407654	5650
15	2007.02.10 12:47:48	156	5651	98.70742798	8.55721283	5651
16	2007.02.10 12:47:48	281	5652	9.786993027	17.86003113	5652
17	2007.02.10 12:47:48	375	5653	15.11256123	60.14748764	5653
18	2007.02.10 12:47:48	500	5654	62.41744232	83.62591553	5654
19	2007.02.10 12:47:48	593	5655	63.55237198	0.399339199	5655
20	2007.02.10 12:47:48	703	5656	13.08904266	80.65895844	5656
21	2007.02.10 12:47:48	812	5657	7.314223289	45.70070648	5657
22	2007.02.10 12:47:48	937	5658	29.23954201	62.89594269	5658
23	2007.02.10 12:47:49	31	5659	90.2518692	70.74030304	5659
24	2007.02.10 12:47:49	156	5660	44.02933884	87.10565186	5660
25	2007.02.10 12:47:49	250	5661	13.34797764	78.82014465	5661
26	2007.02.10 12:47:49	359	5662	95.40398407	95.70143127	5662
27	2007.02.10 12:47:49	468	5663	36.27661896	97.1680069	5663
28	2007.02.10 12:47:49	593	5664	90.15670776	24.55363274	5664
29	2007.02.10 12:47:49	687	5665	3.965777159	97.24994659	5665
30	2007.02.10 12:47:49	812	5666	92.99085999	35.80215073	5666
31	2007.02.10 12:47:49	906	5667	98.15982819	40.08282471	5667
32	2007.02.10 12:47:50	31	5668	40.58247375	75.66890717	5668
33	2007.02.10 12:47:50	125	5669	13.98207569	46.54675674	5669
34	2007.02.10 12:47:50	250	5670	80.64273071	57.27035904	5670
35	2007.02.10 12:47:50	359	5671	61.49720383	92.69765472	5671
36	2007.02.10 12:47:50	468	5672	35.98046875	68.42999268	5672

So let’s see what we got here:

We see a **timestamp** in ***column A*** and ***column B*** and a **trigger variable** (“Logging Method”) in ***column C***.

In ***row 2*** we started the logging while watching three beWISE variables “MyDoubleVariable0”, “MyDoubleVariable1” and “NOOfRefreshes” (***column D – F***).

But this time triggered on change of the Trigger Variable “NoOfRefreshes”. This beWISE variable “NoOfRefreshes” changed quite fast – about 10 times a second. And for every change of this variable we got an entry into the log file.

4.2 DECLARELIVE

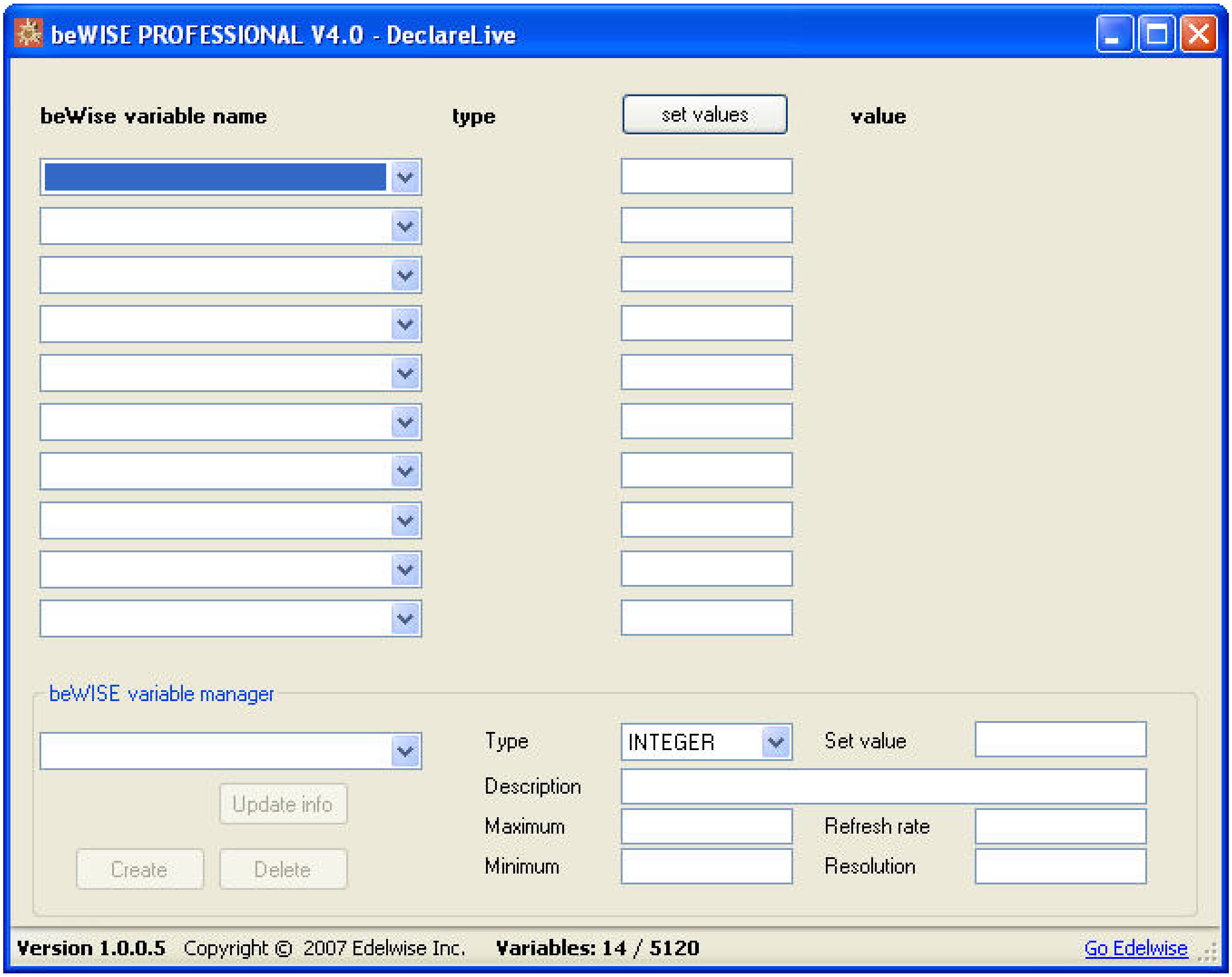
The first part of the DeclareLive looks like the EasyLogger tool. But there is one <u>important difference</u>:

While you can decide with EasyLogger if the refresh of all beWISE variables specified is done by timer or by change event and all are displayed together (synchronal) DeclareLive actually follows a totally other concept:

<u>Each beWISE variable (you have specified) in DeclareLive has its own triggers!</u>

Which means the display is updated for each beWISE variable individually whenever the beWISE variable changes.

Here a standard view of DeclareLive:



Type INTEGER

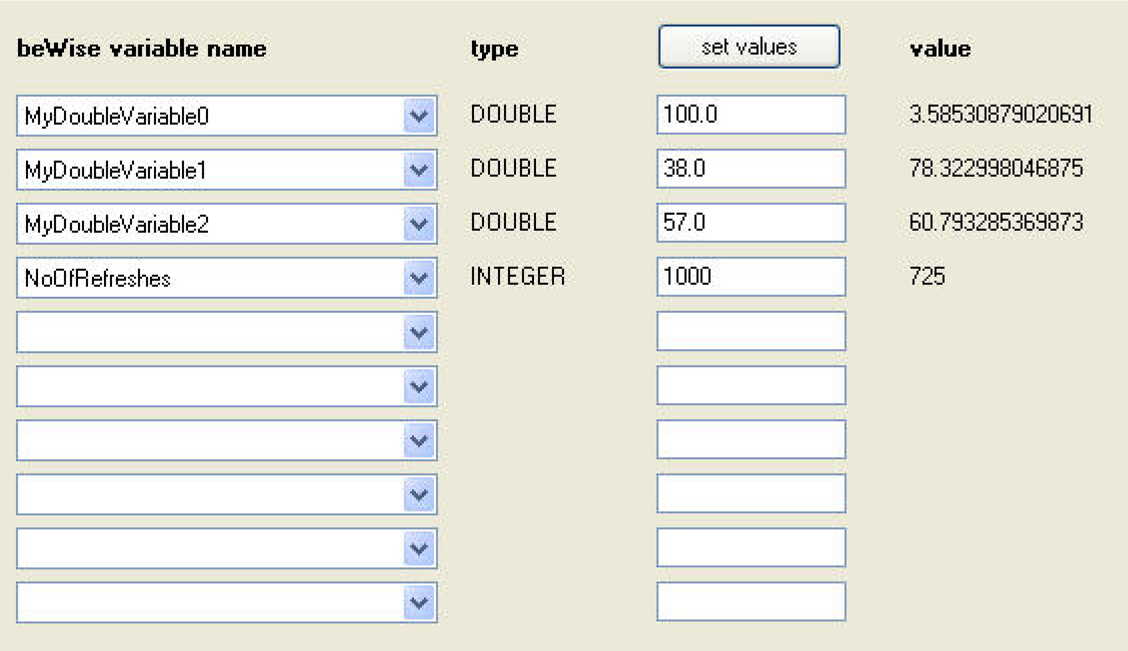
Version 1.0.0.5 Copyright © 2007 Edelwise Inc. Variables: 14 / 5120

With DeclareLive you can do the following actions:

- Watch beWISE variables value changing (but triggered individually)
- Set values for all beWISE variables specified at one time.
- Define a new beWISE variable
- Delete an existing beWISE variable
- Update an existing beWISE variable attributes, including the actual value.

4.2.1 DECLARELIVE - WATCHING AND CHANGING VALUES OF BEWISE VARIABLES

Now we want to watch four beWISE variables and set all their values at the same time:



beWise variable name	type	set values	value
MyDoubleVariable0	DOUBLE	100.0	3.58530879020691
MyDoubleVariable1	DOUBLE	38.0	78.322998046875
MyDoubleVariable2	DOUBLE	57.0	60.793285369873
NoOfRefreshes	INTEGER	1000	725

By pressing the set values button all these variables will get a new value set. The sequence of updating the variables is from top to bottom.

You will see immediately the values on the right being changed.

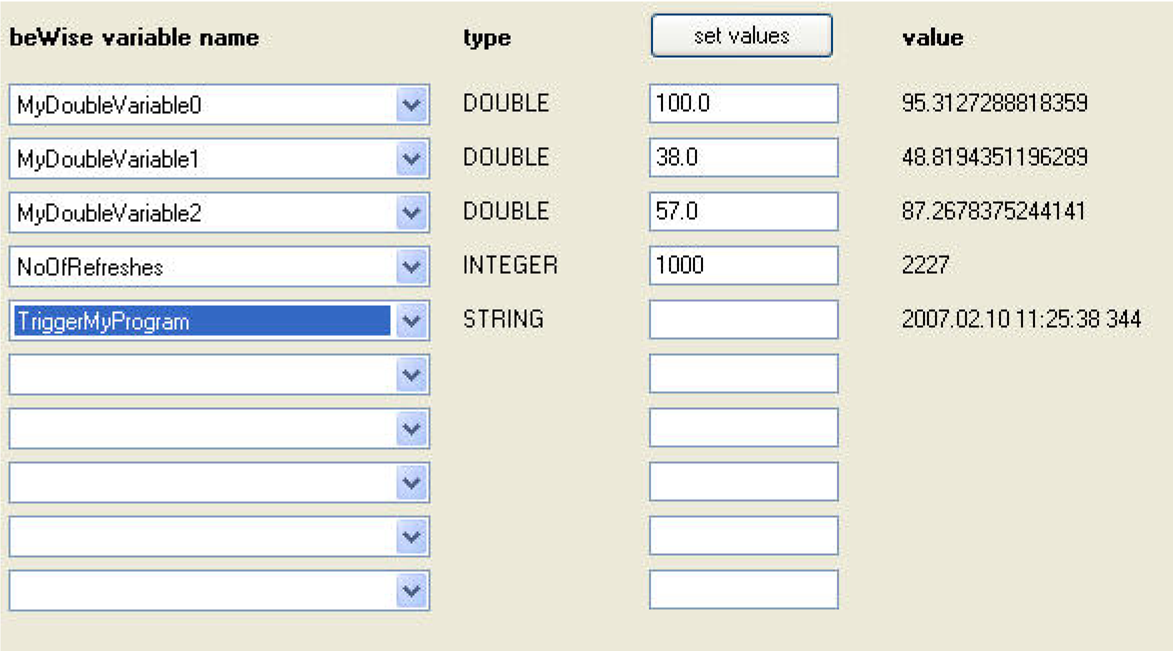
This very nice feature to test and debug interfaces of applications. You could actually put the input and the output variables into this screen (if you have more as 10 you are just use more beWISE tools at the same time), change the input variables to a specific value and look what's the application's output.

Author's advice

Make sure the beWISE variable which triggers the application sits at the end of your list. So you can be sure all the other parameters are set before the trigger fires the application does something.

Usually it's a program-to-program (task-to-task) communication what you are testing here. If you have a program which gets a punch of variables as input parameters from an other program make sure you set all the parameters first and then fire the "GO" event. In a lot of cases I usually use a trigger beWISE variable by creating a beWISE string and put a timestamp (date, time, milliseconds) into it to fire the event. Advantage of this timestamp: It's always unique, so it will always fire the event and I don't have to care about a unique counter or something else like this.

Just like this example:



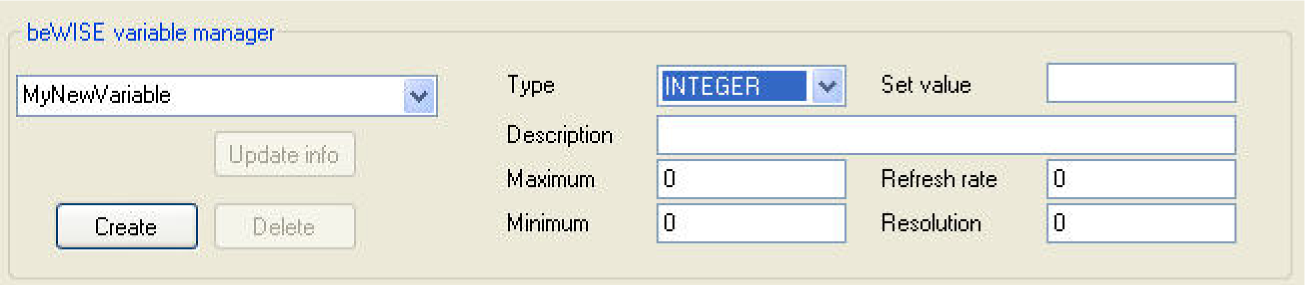
beWise variable name	type	set values	value
MyDoubleVariable0	DOUBLE	100.0	95.3127288818359
MyDoubleVariable1	DOUBLE	38.0	48.8194351196289
MyDoubleVariable2	DOUBLE	57.0	87.2678375244141
NoOfRefreshes	INTEGER	1000	2227
TriggerMyProgram	STRING		2007.02.10 11:25:38 344

Of course here you are just testing and it doesn't really matter what you enter as long it's changing to fire a trigger. But when you make a task-to-task communication, where the trigger has to be created automatically in the sender task, it helps.

4.2.2 DECLARELIVE – BEWISE VARIABLE MANAGER

4.2.2.1 DECLARELIVE – DEFINING NEW BEWISE VARIABLE

Now we want to define a new beWISE variable:



beWISE variable manager

MyNewVariable

Type: INTEGER

Set value:

Description:

Maximum: 0

Refresh rate: 0

Minimum: 0

Resolution: 0

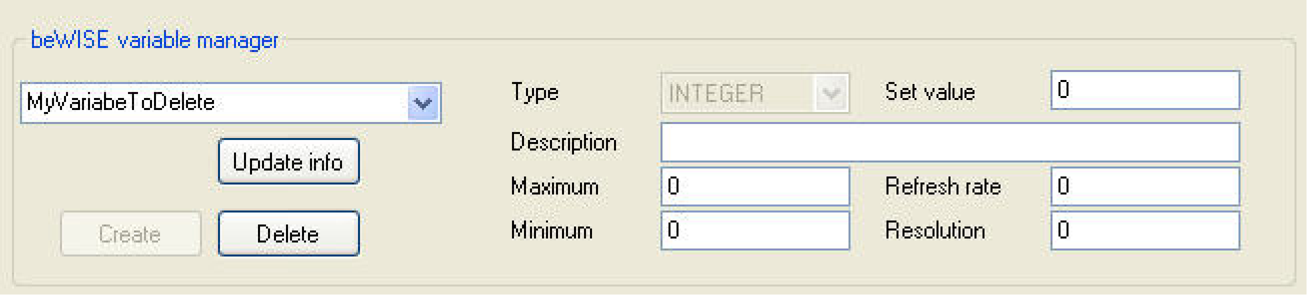
Enter the name of a new beWISE variable (max. length 128 characters), enter all information and then press the Create button.

Type	{INTEGER, DOUBLE, STRING} default: <<the last type used>>
Set value	You can enter a set value default: <<0>> for Integer and Double, << >> for String
Description	You can enter a description of this variable (max. 256 characters)
Maximum	You can enter a maximum value for Integer and Double beWISE variables. Type of this variable is according to the type you specified. *Be aware that this is only an information attribute of this beWISE variable.*
Minimum	You can enter a minimum value for Integer and Double beWISE variables. . Type of this variable is according to the type you specified. *Be aware that this is only an information attribute of this beWISE variable.*
Refresh rate	You can enter a refresh rate value for beWISE variables. Type is integer. Be aware that this is only an information attribute of this beWISE variable.
Resolution	You can enter a resolution value for beWISE variables. Type is integer. Be aware that this is only an information attribute of this beWISE variable.

Only specification **you have to make** about a new variable is the **type**.

4.2.2.2 DECLARELIVE – DELETE A BEWISE VARIABLE

Just choose the variable for all existing beWISE variable and press the delete button.



MyVariabeToDelete

Type INTEGER

Set value 0

Description

Maximum 0

Refresh rate 0

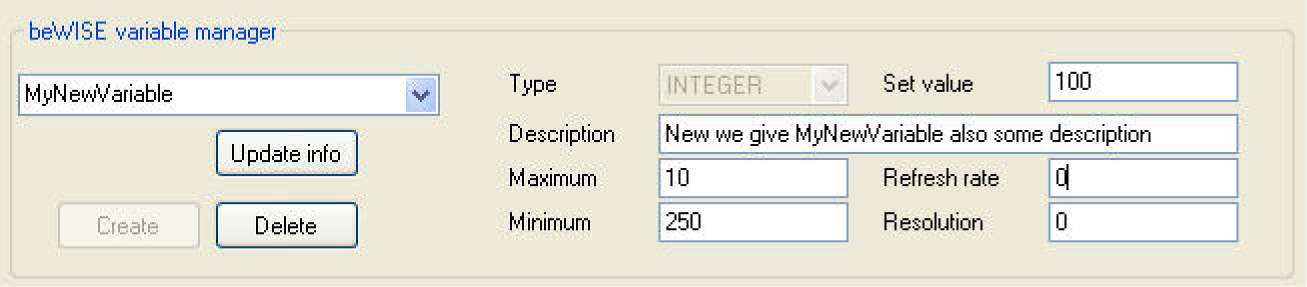
Minimum 0

Resolution 0

4.2.2.3 DECLARELIVE – UPDATE A BEWISE VARIABLE

Just choose the variable for all existing beWISE variable, change all the information you want to, except the type, which can not be changed with update and press the Update info button.

In order to change the type you have to delete the variable and create a new one from the type you want. Be aware (and that's why I made it here more complicated for changing the type) that in all you programs using this variable you have to change the type of receiving variables as well.



MyNewVariable

Type INTEGER

Set value 100

Description New we give MyNewVariable also some description

Maximum 10

Refresh rate 0

Minimum 250

Resolution 0