

UVM_REPORTING

Sunday, March 24, 2024 7:31 AM

1) syntax of uvm info , warning, fatal, error refer pic "uvm_reporting_syntaxes"

`uvm_info(string ID, string MSG, verbosity);
`uvm_error(string ID, string MSG);
`uvm_warning(string ID, string MSG);
`uvm_fatal(string ID, string MSG);

Macros	Syntax
`uvm_info	`uvm_info(ID,MSG,VERBOSITY)
`uvm_warning	`uvm_warning(ID,MSG)
`uvm_error	`uvm_error(ID,MSG)
`uvm_fatal	`uvm_fatal(ID,MSG)

Where:
ID: message tag
MSG: A text message
VERBOSITY: If VERBOSITY is lower than configured verbosity for that reporter, it will be printed.
Note: `uvm_warning has UVM_NONE as a default verbosity.

examples:

string *string* *Point 3*
`uvm_info("TB_INFO", "hello world with uvm", UVM_LOW);
`uvm_warning("TB_WARNING", "this is a warning");
`uvm_error("TB_ERROR", "this is a error");
`uvm_fatal("TB_FATAL", "Fatal stops the simulation");

2) In uvm info we have verbosity levels of uvm info refer picture "uvm_verbosity"

the following are verbosity types

- 1) UVM_NONE
- 2) UVM_LOW
- 3) UVM_MEDIUM
- 4) UVM_HIGH
- 5) UVM_FULL
- 6) UVM_DEBUG

```
typedef enum {
    UVM_NONE = 0,
    UVM_LOW = 100,
    UVM_MEDIUM = 200,
    UVM_HIGH = 300,
    UVM_FULL = 400,
    UVM_DEBUG = 500
} uvm_verbosity;
```

UVM defines standard verbosity for report displays.

Verbosity	Description
UVM_NONE	Always printed; verbosity configuration cannot disable it.
UVM_LOW	Prints a message if verbosity is configured as UVM_LOW or above.
UVM_MEDIUM	Prints a message if verbosity is configured as UVM_MEDIUM or above.
UVM_HIGH	Prints a message if verbosity is configured as UVM_HIGH or above.
UVM_FULL	Prints a message if verbosity is configured as UVM_FULL or above.
UVM_DEBUG	Prints a message if verbosity is configured as UVM_DEBUG.

```
include "uvm_macros.svh"
import uvm_pkg::*;

module tb;

    initial begin
        $display("Default Verbosity level : %d", uvm_top.get_report_verbosity_level());
        `uvm_info("TB_TEST1", "String 1", UVM_NONE); //1
        `uvm_info("TB_TEST1", "String 2", UVM_LOW);
        `uvm_info("TB_TEST1", "String 3", UVM_MEDIUM);
        `uvm_info("TB_TEST1", "String 4", UVM_HIGH);
        `uvm_info("TB_TEST1", "String 5", UVM_FULL);
        `uvm_info("TB_TEST1", "String 6", UVM_DEBUG);

        uvm_top.set_report_verbosity_level(UVM_HIGH); //3
        $display("New Verbosity level : %d", uvm_top.get_report_verbosity_level());
        `uvm_info("TB_TEST2", "String 1", UVM_NONE);
        `uvm_info("TB_TEST2", "String 2", UVM_LOW);
        `uvm_info("TB_TEST2", "String 3", UVM_MEDIUM);
        `uvm_info("TB_TEST2", "String 4", UVM_HIGH);
        `uvm_info("TB_TEST2", "String 5", UVM_FULL);
        `uvm_info("TB_TEST2", "String 6", UVM_DEBUG);
    end
endmodule
```

initial block

prints the verbosity

3) default verbosity of a component [class] is medium refer p2.sv program

```
`uvm_info("TB_TEST1", "String 1", UVM_NONE);
`uvm_info("TB_TEST1", "String 2", UVM_LOW);
`uvm_info("TB_TEST1", "String 3", UVM_MEDIUM);
`uvm_info("TB_TEST1", "String 4", UVM_HIGH);
`uvm_info("TB_TEST1", "String 5", UVM_FULL);
`uvm_info("TB_TEST1", "String 6", UVM_DEBUG);
```

by default only uvm infos with verbosity level less than UVM_MEDIUM[<200] are printed.

If you want to set there were positive of the whole component then use handle name dot set report verbosity level method with an argument of the desired verbosity

Handle_name.set_report_verbosity_level(DESIRED_VERBOSITY);

Ex: drv.set_report_verbosity_level(UVM_HIGH);

4) changing verbosity of a particular message id in a component [class] handle_name.set_report_id_verbosity("MESSAGE_ID", DESIRED_VERBOSITY);

this sets the verbosity of the particular components message id to the DESIRED_VERBOSITY for the remaining components we can set it separately according to message id or else

the default verbosity [UVM_MEDIUM] will be applied

refer p3.sv and p4.sv

inside class

```
include "uvm_macros.svh"
import uvm_pkg::*;

class driver extends uvm_driver;
    uvm_component_utils(driver);

    function new(string path, uvm_component parent);
        super.new(path, parent);
    endfunction

    task run();
        `uvm_info("DRV1", "Executed Driver1 Code", UVM_HIGH);
        `uvm_info("DRV2", "Executed Driver2 Code", UVM_HIGH);
    endtask
endclass

module tb;

    driver drv;
    initial begin
        drv = new("DRV", null);
        //1
        drv.set_report_verbosity_level(UVM_HIGH);
        drv.run();
    end
endmodule
```

VERBOSITY_LEVEL

UVM-MEDIUM
UVM-HIGH

UVM-HIGH

If this was not there then nothing would have been printed As the default verbosity will be uvm medium and in our UVM info we have the verbosity as UVM high

P3.sv

```
include "uvm_macros.svh"
import uvm_pkg::*;

class driver extends uvm_driver;
    uvm_component_utils(driver);

    function new(string path, uvm_component parent);
        super.new(path, parent);
    endfunction

    task run();
        `uvm_info("DRV1", "Executed Driver1 Code", UVM_HIGH);
        `uvm_info("DRV2", "Executed Driver2 Code", UVM_HIGH);
    endtask
endclass

module tb;

    driver drv;
    initial begin
        drv = new("DRV", null);
        //1
        drv.set_report_id_verbosity("DRV1", UVM_HIGH);
        drv.run();
    end
endmodule
```

Printed
Not printed

VERBOSITY_LEVEL
MSG_ID1 [DRV1]
MSG_ID2 [DRV2]

UVM-MEDIUM
UVM-MEDIUM UVM-HIGH
UVM-MEDIUM

P4.sv

```
include "uvm_macros.svh"
import uvm_pkg::*;

class driver1 extends uvm_driver;
    uvm_component_utils(driver1);

    function new(string path, uvm_component parent);
```

VERBOSITY_LEVEL
MSG_ID1 [DRV1.2]
MSG_ID2 [DRV1.6]
UVM-HIGH
UVM-MEDIUM
UVM-MEDIUM

```
class driver2 extends uvm_driver;
    uvm_component_utils(driver2);

    function new(string path, uvm_component parent);
        super.new(path, parent);
    endfunction
```

VERBOSITY_LEVEL
MSG_ID1 [DRV1.2]
MSG_ID2 [DRV1.6]
UVM-MEDIUM
UVM-MEDIUM

```

4  import uvm_pkg;
5
6  //////////////////////////////////////
7  class driver1 extends uvm_driver;
8      uvm_component_utils(driver1)
9
10     function new(string path , uvm_component parent);
11         super.new(path, parent);
12     endfunction
13
14     task run();
15         'uvm_info("DRV1_a", "String 3", UVM_HIGH);
16         'uvm_info("DRV1_b", "String 4", UVM_HIGH);
17     endtask
18
19 endclass
20
21
22

```

PS.10
 PS.102 (DRV1-a) UVM_MEDIUM
 PS.102 (DRV1-b) UVM_MEDIUM
 printed

```

24 class driver2 extends uvm_driver;
25     uvm_component_utils(driver2)
26
27     function new(string path , uvm_component parent);
28         super.new(path, parent);
29     endfunction
30
31
32     task run();
33         'uvm_info("DRV2_a", "String 1", UVM_HIGH);
34         'uvm_info("DRV2_b", "String 2", UVM_HIGH);
35     endtask
36
37 endclass
38
39

```

PS.102 (DRV2-a) UVM_MEDIUM
 PS.102 (DRV2-b) UVM_MEDIUM

```

40 module tb;
41
42     driver1 drv1;
43     driver2 drv2;
44
45
46     initial begin
47         drv1 = new("DRV1", null);
48         drv2 = new("DRV2", null);
49
50         //the default verbosity is UVM_MEDIUM -> none of the UVM_HIGH are printed
51         drv1.run();
52         drv2.run();
53
54         //setting the verbosity of the particular ID of that component
55         drv1.set_report_id_verbosity("DRV1_a", UVM_HIGH);
56         drv1.run();
57         drv2.run();
58     end
59 endmodule

```

nothing is printed
 1
 2

PS.11

```

5  'include "uvm_macros.svh"
6  import uvm_pkg;
7
8  //////////////////////////////////////
9  class driver1 extends uvm_driver;
10     uvm_component_utils(driver1)
11
12     function new(string path , uvm_component parent);
13         super.new(path, parent);
14     endfunction
15
16     task run();
17         'uvm_info("DRV1_a", "DRV1_A String 1", UVM_NONE);
18         'uvm_info("DRV1_a", "DRV1_A String 2", UVM_LOW);
19         'uvm_info("DRV1_a", "DRV1_A String 3", UVM_MEDIUM);
20         'uvm_info("DRV1_a", "DRV1_A String 4", UVM_HIGH);
21         'uvm_info("DRV1_a", "DRV1_A String 5", UVM_FULL);
22         'uvm_info("DRV1_a", "DRV1_A String 6", UVM_DEBUG);
23
24         'uvm_info("DRV1_b", "DRV1_B String 1", UVM_NONE);
25         'uvm_info("DRV1_b", "DRV1_B String 2", UVM_LOW);
26         'uvm_info("DRV1_b", "DRV1_B String 3", UVM_MEDIUM);
27         'uvm_info("DRV1_b", "DRV1_B String 4", UVM_HIGH);
28         'uvm_info("DRV1_b", "DRV1_B String 5", UVM_FULL);
29         'uvm_info("DRV1_b", "DRV1_B String 6", UVM_DEBUG);
30     endtask
31 endclass

```

verbosity_level UVM_HIGH
 PS.102 (DRV1-a) UVM_MEDIUM
 PS.102 (DRV1-b) UVM_MEDIUM

```

35 class driver2 extends uvm_driver;
36     uvm_component_utils(driver2)
37
38     function new(string path , uvm_component parent);
39         super.new(path, parent);
40     endfunction
41
42
43     task run();
44         'uvm_info("DRV2_a", "DRV1_A String 1", UVM_NONE);
45         'uvm_info("DRV2_a", "DRV1_A String 2", UVM_LOW);
46         'uvm_info("DRV2_a", "DRV1_A String 3", UVM_MEDIUM);
47         'uvm_info("DRV2_a", "DRV1_A String 4", UVM_HIGH);
48         'uvm_info("DRV2_a", "DRV1_A String 5", UVM_FULL);
49         'uvm_info("DRV2_a", "DRV1_A String 6", UVM_DEBUG);
50
51         'uvm_info("DRV2_b", "DRV2_B String 1", UVM_NONE);
52         'uvm_info("DRV2_b", "DRV2_B String 2", UVM_LOW);
53         'uvm_info("DRV2_b", "DRV2_B String 3", UVM_MEDIUM);
54         'uvm_info("DRV2_b", "DRV2_B String 4", UVM_HIGH);
55         'uvm_info("DRV2_b", "DRV2_B String 5", UVM_FULL);
56         'uvm_info("DRV2_b", "DRV2_B String 6", UVM_DEBUG);
57     endtask
58 endclass

```

verbosity_level UVM_MEDIUM
 PS.102 (DRV2-a) UVM_MEDIUM
 PS.102 (DRV2-b) UVM_MEDIUM

```

60 module tb;
61
62     driver1 drv1;
63     driver2 drv2;
64
65     //comment this initial block and uncomment the 2nd initial block
66     //uncomment both initial blocks
67     //check both scenario outputs
68     initial begin
69         drv1 = new("DRV1", null);
70         drv2 = new("DRV2", null);
71
72         //the default verbosity is UVM_MEDIUM -> none of the UVM_HIGH are printed
73         drv1.run();
74         drv2.run();
75
76         $display("*****\n\n");
77         #2;
78         //setting the verbosity of the particular ID of that component
79         drv1.set_report_id_verbosity("DRV1_a", UVM_HIGH);
80         drv1.run();
81         drv2.run();
82         $display("*****\n\n");
83
84         #10; // here the previous verbosity is also active and the one which we are putting below is also active
85         drv2.set_report_id_verbosity("DRV2_b", UVM_HIGH);
86         drv1.run();
87         drv2.run();
88         $display("*****\n\n");
89
90         // we don't use the set_report_id_verbosity() function when using uvm classes because everytime we change the code we will have to compile the file
91         // so instead we set these things in simulation options
92     end
93     //uvm_verbosity_initialised commented
94     /*
95     initial begin
96         #1;
97         drv2.set_report_id_verbosity("DRV1_a", UVM_HIGH);
98         drv1.run();
99         drv2.run();
100         $display("*****\n\n");
101     end
102     */
103 endmodule

```

note this may id doesn't exists -> no compile error reported
 1
 2
 3
 4

- if there are child classes declared inside the component whose verbosity we are changing set_report_verbosity_level() will not effect their verbosity
- changing the verbosity of the entire hierarchy of the component environment class handle env is having 2 child classes mon and drv
 env.set_report_verbosity_level_hier(UVM_HIGH);
 handle_name.set_report_verbosity_level_hier(DESIRED_VERBOSITY);
 This will set the verbosity of the entire component along with its child to DESIRED_VERBOSITY
 in this case parent class env along with its child mon and drv will have DESIRED_VERBOSITY

```

8 "include "uvm_macros.svh"
9 import uvm_pkg::*;
10
11 //////////////////////////////////////////////////
12 class driver extends uvm_driver;
13   uvm_component_utils(driver);
14
15   function new(string path , uvm_component parent);
16     super.new(path, parent);
17   endfunction
18
19   task run();
20     "uvm_info("DRV", "Executed Driver Code", UVM_HIGH);
21   endtask
22 endclass
23 //////////////////////////////////////////////////

```

```

25 class monitor extends uvm_monitor;
26   uvm_component_utils(monitor);
27
28   function new(string path , uvm_component parent);
29     super.new(path, parent);
30   endfunction
31
32   task run();
33     "uvm_info("MON", "Executed Monitor Code", UVM_HIGH);
34   endtask
35 endclass
36 //////////////////////////////////////////////////
37
38

```

```

29 class env extends uvm_env;
30   uvm_component_utils(env);
31
32   driver drv;
33   monitor mon;
34
35   function new(string path , uvm_component parent);
36     super.new(path, parent);
37     drv = new("DRV", this);
38     mon = new("MON", this);
39   endfunction
40
41   task run();
42     drv.run();
43     mon.run();
44   endtask
45 endclass

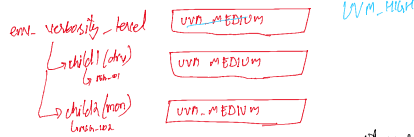
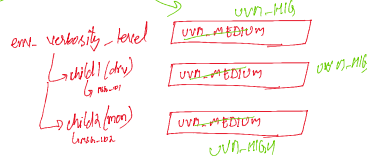
```

```

module tb;
  env e;

  initial begin
    e = new("ENV", null);
    e.set_report_verbosity_level_hier(UVM_HIGH);
    e.run();
  end
endmodule

```



difference b/w set_report_verbosity_level() , set_report_verbosity_level_hier()

PS-5

```

1 "include "uvm_macros.svh"
2 import uvm_pkg::*;
3
4 //////////////////////////////////////////////////
5 class driver extends uvm_driver;
6   uvm_component_utils(driver);
7
8   function new(string path , uvm_component parent);
9     super.new(path, parent);
10   endfunction
11
12   task run();
13     "uvm_info("DRV", "Executed Driver Code", UVM_HIGH);
14     "uvm_info("DRV_a", "DRV1_A String 1", UVM_NONE);
15     "uvm_info("DRV1_a", "DRV1_A String 2", UVM_LOW);
16     "uvm_info("DRV1_a", "DRV1_A String 3", UVM_MEDIUM);
17     "uvm_info("DRV1_a", "DRV1_A String 4", UVM_HIGH);
18     "uvm_info("DRV1_a", "DRV1_A String 5", UVM_FULL);
19     "uvm_info("DRV1_a", "DRV1_A String 6", UVM_DEBUG);
20     $display();
21   endtask
22 endclass

```

```

25 class monitor extends uvm_monitor;
26   uvm_component_utils(monitor);
27
28   function new(string path , uvm_component parent);
29     super.new(path, parent);
30   endfunction
31
32   task run();
33     "uvm_info("MON", "Executed Monitor Code", UVM_HIGH);
34     "uvm_info("MON1_a", "MON1_A String 1", UVM_NONE);
35     "uvm_info("MON1_a", "MON1_A String 2", UVM_LOW);
36     "uvm_info("MON1_a", "MON1_A String 3", UVM_MEDIUM);
37     "uvm_info("MON1_a", "MON1_A String 4", UVM_HIGH);
38     "uvm_info("MON1_a", "MON1_A String 5", UVM_FULL);
39     "uvm_info("MON1_a", "MON1_A String 6", UVM_DEBUG);
40   endtask
41 endclass
42
43

```

```

class env extends uvm_env;
  uvm_component_utils(env);

  driver drv;
  monitor mon;

  function new(string path , uvm_component parent);
    super.new(path, parent);
    drv = new("DRV", this);
    mon = new("MON", this);
  endfunction

  task run();
    "uvm_info("ENV", "Executed Driver Code", UVM_HIGH);
    "uvm_info("ENV1_a", "ENV1_A String 1", UVM_NONE);
    "uvm_info("ENV1_a", "ENV1_A String 2", UVM_LOW);
    "uvm_info("ENV1_a", "ENV1_A String 3", UVM_MEDIUM);
    "uvm_info("ENV1_a", "ENV1_A String 4", UVM_HIGH);
    "uvm_info("ENV1_a", "ENV1_A String 5", UVM_FULL);
    "uvm_info("ENV1_a", "ENV1_A String 6", UVM_DEBUG);
    $display();

    drv.run();
    mon.run();
  endtask
endclass

```

```

module tb;
  env e;

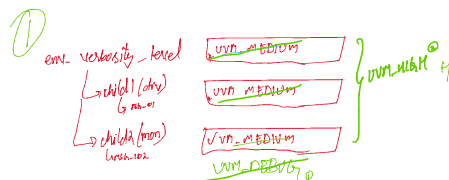
  initial begin
    e = new("ENV", null);

    /*
    //the order in which these 2 are written matter as hier will effect the verbosity of entire hierarchy
    //whereas level will effect verbosity of that component
    //here it matters because the hierarchy contains the child class mon

    //order 1
    e.mon.set_report_verbosity_level(UVM_DEBUG);
    e.set_report_verbosity_level_hier(UVM_HIGH);

    //order 2
    e.set_report_verbosity_level_hier(UVM_HIGH);
    e.mon.set_report_verbosity_level(UVM_DEBUG);
    */
  end
endmodule

```



e.mon.set_report_verbosity_level(UVM_DEBUG); //only sets the verbosity of child mon ->UVM_DEBUG
e.set_report_verbosity_level_hier(UVM_HIGH); //only sets the verbosity of parent e->UVM_HIGH and not effect the child verbosity

```
//here it matters because the hierarchy contains the child class mon
//order 1
e.mon.set_report_verbosity_level(UVM_DEBUG);
e.set_report_verbosity_level_hier(UVM_HIGH);

//order 2
e.set_report_verbosity_level_hier(UVM_HIGH);
e.mon.set_report_verbosity_level(UVM_DEBUG);

*/

//the order of these 2 doesn't matter as its setting only for that component
e.mon.set_report_verbosity_level(UVM_DEBUG);
e.set_report_verbosity_level(UVM_HIGH);

e.run();
end
endmodule
```

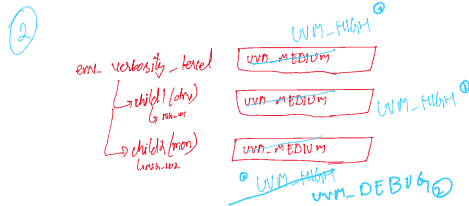
```
/*
e.mon.set_report_verbosity_level(UVM_DEBUG); //only sets the verbosity of child mon ->UVM_DEBUG
e.set_report_verbosity_level(UVM_HIGH); //only sets the verbosity of parent e->UVM_HIGH and not effect the child verbosity
//as child drv verbosity is not not it uses default verbosity as UVM_MEDIUM

output:
UVM_INFO p8.sv(64) @ 0: ENV [ENV] Executed ENV Code
UVM_INFO p8.sv(65) @ 0: ENV [ENV1_a] ENV1_A String 1
UVM_INFO p8.sv(66) @ 0: ENV [ENV1_a] ENV1_A String 2
UVM_INFO p8.sv(67) @ 0: ENV [ENV1_a] ENV1_A String 3
UVM_INFO p8.sv(68) @ 0: ENV [ENV1_a] ENV1_A String 4

UVM_INFO p8.sv(14) @ 0: ENV.DRV [DRV] Executed Driver Code
UVM_INFO p8.sv(15) @ 0: ENV.DRV [DRV1_a] DRV1_A String 1
UVM_INFO p8.sv(16) @ 0: ENV.DRV [DRV1_a] DRV1_A String 2
UVM_INFO p8.sv(17) @ 0: ENV.DRV [DRV1_a] DRV1_A String 3
UVM_INFO p8.sv(18) @ 0: ENV.DRV [DRV1_a] DRV1_A String 4

UVM_INFO p8.sv(36) @ 0: ENV.MON [MON] Executed Monitor Code
UVM_INFO p8.sv(37) @ 0: ENV.MON [MON1_a] MON1_A String 1
UVM_INFO p8.sv(38) @ 0: ENV.MON [MON1_a] MON1_A String 2
UVM_INFO p8.sv(39) @ 0: ENV.MON [MON1_a] MON1_A String 3
UVM_INFO p8.sv(40) @ 0: ENV.MON [MON1_a] MON1_A String 4
UVM_INFO p8.sv(41) @ 0: ENV.MON [MON1_a] MON1_A String 5
UVM_INFO p8.sv(42) @ 0: ENV.MON [MON1_a] MON1_A String 6
```

From <https://www.edaplayground.com/s/A_dc>



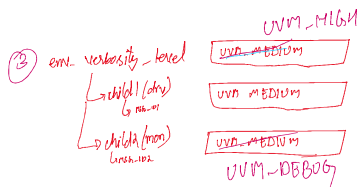
```
e.set_report_verbosity_level_hier(UVM_HIGH); // sets the verbosity of parent e->UVM_HIGH and also the child verbosity too that is child drv,mon->UVM_HIGH
e.mon.set_report_verbosity_level(UVM_DEBUG); //only sets the verbosity of child mon ->UVM_DEBUG

//here the hierarchy verbosity is over written by component verbosity for mon mon->UVM_DEBUG
```

```
OUTPUT:
UVM_INFO p8.sv(64) @ 0: ENV [ENV] Executed ENV Code
UVM_INFO p8.sv(65) @ 0: ENV [ENV1_a] ENV1_A String 1
UVM_INFO p8.sv(66) @ 0: ENV [ENV1_a] ENV1_A String 2
UVM_INFO p8.sv(67) @ 0: ENV [ENV1_a] ENV1_A String 3
UVM_INFO p8.sv(68) @ 0: ENV [ENV1_a] ENV1_A String 4

UVM_INFO p8.sv(14) @ 0: ENV.DRV [DRV] Executed Driver Code
UVM_INFO p8.sv(15) @ 0: ENV.DRV [DRV1_a] DRV1_A String 1
UVM_INFO p8.sv(16) @ 0: ENV.DRV [DRV1_a] DRV1_A String 2
UVM_INFO p8.sv(17) @ 0: ENV.DRV [DRV1_a] DRV1_A String 3
UVM_INFO p8.sv(18) @ 0: ENV.DRV [DRV1_a] DRV1_A String 4

UVM_INFO p8.sv(36) @ 0: ENV.MON [MON] Executed Monitor Code
UVM_INFO p8.sv(37) @ 0: ENV.MON [MON1_a] MON1_A String 1
UVM_INFO p8.sv(38) @ 0: ENV.MON [MON1_a] MON1_A String 2
UVM_INFO p8.sv(39) @ 0: ENV.MON [MON1_a] MON1_A String 3
UVM_INFO p8.sv(40) @ 0: ENV.MON [MON1_a] MON1_A String 4
UVM_INFO p8.sv(41) @ 0: ENV.MON [MON1_a] MON1_A String 5
UVM_INFO p8.sv(42) @ 0: ENV.MON [MON1_a] MON1_A String 6
```



```
e.mon.set_report_verbosity_level(UVM_DEBUG); //only sets the verbosity of child mon ->UVM_DEBUG
e.set_report_verbosity_level(UVM_HIGH); //only sets the verbosity of parent e->UVM_HIGH and not effect the child verbosity
//as child drv verbosity is not not it uses default verbosity as UVM_MEDIUM
```

From <https://www.edaplayground.com/s/A_dc>

```
OUTPUT:
UVM_INFO p8.sv(64) @ 0: ENV [ENV] Executed ENV Code
UVM_INFO p8.sv(65) @ 0: ENV [ENV1_a] ENV1_A String 1
UVM_INFO p8.sv(66) @ 0: ENV [ENV1_a] ENV1_A String 2
UVM_INFO p8.sv(67) @ 0: ENV [ENV1_a] ENV1_A String 3
UVM_INFO p8.sv(68) @ 0: ENV [ENV1_a] ENV1_A String 4

UVM_INFO p8.sv(15) @ 0: ENV.DRV [DRV1_a] DRV1_A String 1
UVM_INFO p8.sv(16) @ 0: ENV.DRV [DRV1_a] DRV1_A String 2
UVM_INFO p8.sv(17) @ 0: ENV.DRV [DRV1_a] DRV1_A String 3

UVM_INFO p8.sv(36) @ 0: ENV.MON [MON] Executed Monitor Code
UVM_INFO p8.sv(37) @ 0: ENV.MON [MON1_a] MON1_A String 1
UVM_INFO p8.sv(38) @ 0: ENV.MON [MON1_a] MON1_A String 2
UVM_INFO p8.sv(39) @ 0: ENV.MON [MON1_a] MON1_A String 3
UVM_INFO p8.sv(40) @ 0: ENV.MON [MON1_a] MON1_A String 4
UVM_INFO p8.sv(41) @ 0: ENV.MON [MON1_a] MON1_A String 5
UVM_INFO p8.sv(42) @ 0: ENV.MON [MON1_a] MON1_A String 6

VCS Simulation Report
```

From <https://www.edaplayground.com/s/A_dc>

7)there are 4 severities

1]uvm_info 2]uvm_warning 3]uvm_error 4]uvm_fatal

8)changing the severity level of a uvm reporting

handle_name.set_report_severity_override(CURRENT_SEVERITY,DESIRED_SEVERITY)

example:

d.set_report_severity_override(UVM_WARNING, UVM_ERROR);

```
5 "include "uvm_macros.svh"
6 import uvm_pkg::*;
7
8 //////////////////////////////////////////
9 class driver extends uvm_driver;
10 "uvm_component_utils(driver)
11
12 function new(string path , uvm_component parent);
13 super.new(path, parent);
14 endfunction
15
16 task run();
17 "uvm_info("DRV", "Informational Message", UVM_INFO);
18 "uvm_warning("DRV", "Potential Error");
19 "uvm_error("DRV", "Real Error");
20 #10;
21 "uvm_fatal("DRV", "Simulation cannot continue DRV");
22 #10;
23 "uvm_fatal("DRV1", "Simulation Cannot Continue DRV1");
24 endtask
25
26 endclass
```

U I I
W E E
E E E
F E F
as 1st fatal stops simulation. this will not print

```
module tb;
driver d;

initial begin
d = new("drv", null);
// d.set_report_severity_override(UVM_FATAL, UVM_ERROR);
d.set_report_severity_id_override(UVM_FATAL, "DRV", UVM_ERROR);
d.run();
end

endmodule
```

output without any severity conversion

```
OUTPUT:
UVM_INFO p8.sv(13) @ 0: DRV [DRV] Informational Message
UVM_WARNING p8.sv(14) @ 0: DRV [DRV] Potential Error
UVM_ERROR p8.sv(15) @ 0: DRV [DRV] Real Error
UVM_FATAL p8.sv(17) @ 10: DRV [DRV] Simulation cannot continue DRV
UVM_INFO japps/vcs/mvcs/S-2021.09/ebc/uvim-1.2/sr0/bas0/uvim_report_server.svh(904) @ 10: reporter
[UVMREPORT/SERVER]
--- UVM Report Summary ---

** Report counts by severity
UVM_INFO : 2
```

```

UVM_WARNING: 1
UVM_ERROR: 1
UVM_FATAL: 1
** Report counts by id
[DRV] 4
[UVM/RELNOTES] 1

```

From <https://www.edaplayground.com/x/A_dc>

```

/*
d.set_report_severity_override(UVM_FATAL, UVM_ERROR); // converts all uvm_fatal to uvm_error

```

```

OUTPUT:
UVM_INFO p0.sv(13) @ 0: DRV [DRV] Informational Message
UVM_WARNING p0.sv(14) @ 0: DRV [DRV] Potential Error
UVM_ERROR p0.sv(15) @ 0: DRV [DRV] Real Error
UVM_ERROR p0.sv(17) @ 10: DRV [DRV] Simulation cannot continue DRV
UVM_ERROR p0.sv(19) @ 20: DRV [DRV] Simulation Cannot Continue DRV

```

From <https://www.edaplayground.com/x/A_dc>

```

d.set_report_severity_id_override(UVM_FATAL, "DRV", UVM_ERROR); //converts the `uvm_fatal messages with message id "DRV" to uvm_error

```

```

OUTPUT:
UVM_INFO p0.sv(13) @ 0: DRV [DRV] Informational Message
UVM_WARNING p0.sv(14) @ 0: DRV [DRV] Potential Error
UVM_ERROR p0.sv(15) @ 0: DRV [DRV] Real Error
UVM_ERROR p0.sv(17) @ 10: DRV [DRV] Simulation cannot continue DRV
UVM_FATAL p0.sv(19) @ 20: DRV [DRV] Simulation Cannot Continue DRV
UVM_INFO /apps/vcs/mx/vcs/S-2021.09/etc/uvmm-1.2/src/base/uvm_report_server.svh(904) @ 0: reporter [UVM/REPORT/SERVER]
--- UVM Report Summary ---
** Report counts by severity
UVM_INFO: 2
UVM_WARNING: 1
UVM_ERROR: 2
UVM_FATAL: 1
** Report counts by id
[DRV] 4
[DRV] 1
[UVM/RELNOTES] 1

```

From <https://www.edaplayground.com/x/A_dc>

```

incase you want to convert all warnings to erros
d.set_report_severity_override(UVM_WARNING, UVM_ERROR);

```

Severity	Default Action
UVM_INFO	UVM_DISPLAY
UVM_WARNING	UVM_DISPLAY
UVM_ERROR	UVM_DISPLAY UVM_COUNT
UVM_FATAL	UVM_DISPLAY UVM_EXIT

Action	Available actions	Description
UVM_DISPLAY		Displays messages on console
UVM_COUNT		Increments global error count
UVM_EXIT		Terminates the simulation immediately
UVM_LOG		Captures messages into specified log file
UVM_CALL_HOOK		Calls callback method
UVM_NO_ACTION		No Action will be taken
UVM_STOP		Causes \$stop to be executed, puts simulation into interactive mode

9)the actions which a severity can do are as follows

1]UVM_NO_ACTION 2] UVM_DISPLAY 3]UVM_LOG 4]UVM_COUNT
5]UVM_EXIT 6] UVM_CALL_HOOK 7]UVM_STOP 8] UVM_RM_RECORD

10)there are default actions see the picture attached "uvm_reporting_severity_default_actions"

11]changing the actions of particular severity

```
handle_name.set_report_severity_action(UVM_SEVERITY, UVM_ACTION1 | UVM_ACTION2 |...);
```

example:-

```
d.set_report_severity_action(UVM_INFO,UVM_NO_ACTION);
```

```
d.set_report_severity_action(UVM_INFO,UVM_DISPLAY | UVM_EXIT);
```

12) the action UVM_COUNT when added to a severity will increment the quit count whenever that severity message is executed

as by default this counts the `uvm_error messages count

whenever this count variable reaches a certain value it exits the simulation

in order to set this certain value to the count variable we use

```
handle_name.set_report_max_quit_count(DESIRED_COUNT_NUMBER_VALUE);
```

example:

```
d.set_report_max_quit_count(3);
```

```

1  "include "uvm_macros.svh"
2  import uvm_pkg::*;
3
4  //////////////////////////////////////////////////
5  class driver extends uvm_driver;
6  "uvm_component_utils(driver)
7
8  function new(string path , uvm_component parent);
9  super.new(path, parent);
10 endfunction
11
12 task run();
13 $display("start of run");
14
15 "uvm_info("DRV", "Informational Message", UVM_NONE);
16 $display("after 1st uvm_info");
17 "uvm_warning("DRV", "Potential Error");
18 "uvm_error("DRV", "Real Error"); //uvm_count
19 "uvm_error("DRV", "Second Real Error");
20
21 //
22 "uvm_fatal("DRV", "Simulation cannot continue DRV"); // uvm_exit
23
24 "uvm_fatal("DRV", "Simulation Cannot Continue DRV");
25
26 endtask
27
28 endclass

```

```

module tb;
driver d;

initial begin
d = new("DRV", null);
d.set_report_severity_action(UVM_INFO,UVM_NO_ACTION);
d.run();

$ddisplay("*****");
d.set_report_severity_action(UVM_INFO,UVM_DISPLAY | UVM_EXIT);
d.run();

$ddisplay("*****");
d.set_report_max_quit_count(3);
d.run();
end
endmodule

```

start of run

after 1st uvm_info

UVM_WARNING p10.sv(16) @ 0: DRV [DRV] Potential Error

UVM_ERROR p10.sv(17) @ 0: DRV [DRV] Real Error

UVM_ERROR p10.sv(18) @ 0: DRV [DRV] Second Real Error

start of run

UVM_INFO p10.sv(14) @ 0: DRV [DRV] Informational Message

UVM_INFO /apps/vcs/mx/vcs/S-2021.09/etc/uvmm-1.2/src/base/uvm_report_server.svh(904) @ 0: reporter

[UVM/REPORT/SERVER]

--- UVM Report Summary ---

** Report counts by severity

UVM_INFO: 2

UVM_WARNING: 1

UVM_ERROR: 2

UVM_FATAL: 0

** Report counts by id
[DRV] 4
[UVM/RELNOTES] 1

From <https://www.edaplayground.com/x/A_dc>