



## DAY-93

### #100DAYSRTL

---

## “UVM: UVM Object(Part-1)”

- Core methods available in UVM:- Print, Copy, Compare, clone, Pack/Unpack, record, Create
- The space where we add field macros to our data members there we use utils
- The flags mentioned as arguments in a function, Check it

<i>UVM_ALL_ON</i>	Set all operations on.
<i>UVM_DEFAULT</i>	This is the recommended set of flags to pass to the field macros. Currently, it enables all of the operations, making it functionally identical to <i>UVM_ALL_ON</i> . In the future however, additional flags could be added with a recommended default value of <i>off</i> .
<i>UVM_NOCOPY</i>	Do not copy this field.
<i>UVM_NOCOMPARE</i>	Do not compare this field.
<i>UVM_NOPRINT</i>	Do not print this field.
<i>UVM_NOPACK</i>	Do not pack or unpack this field.
<i>UVM_REFERENCE</i>	For object types, operate only on the handle (e.g. no deep copy)
<i>UVM_PHYSICAL</i>	Treat as a physical field. Use physical setting in policy class for this field.
<i>UVM_ABSTRACT</i>	Treat as an abstract field. Use the abstract setting in the policy class for this field.
<i>UVM_READONLY</i>	Do not allow setting of this field from the <code>set_*_local</code> methods or during <code>uvm_component::apply_config_settings</code> operation.

## “Code Practising”:-

```
`include "uvm_macros.svh"
import uvm_pkg::*;
class transaction extends uvm_sequence_item;
    rand bit [3:0] data;
    function new (string inst="transaction");
        super.new(inst);
    endfunction
    `uvm_object_utils_begin(transaction)
    `uvm_field_int(data,UVM_DEFAULT)
    `uvm_object_utils_end
endclass
module tb;
    transaction t;
    initial begin
        t=new();
        t.randomize();
        t.print();
    end
endmodule
```

## “Result”:-

```
-----
Name           Type           Size  Value
-----
transaction    transaction  -      @335
  data         integral     4      'h6
-----
simulation has finished. There are no more test vectors to simulate.
```

- We can change the radix by using the following conventions

<i>UVM_BIN</i>	Print / record the field in binary (base-2).
<i>UVM_DEC</i>	Print / record the field in decimal (base-10).
<i>UVM_UNSIGNED</i>	Print / record the field in unsigned decimal (base-10)
<i>UVM_OCT</i>	Print / record the field in octal (base-8).
<i>UVM_HEX</i>	Print / record the field in hexadecimal (base-16).
<i>UVM_STRING</i>	Print / record the field in string format.
<i>UVM_TIME</i>	Print / record the field in time format.

## “Code Practising”:-

```
`include "uvm_macros.svh"
import uvm_pkg::*;
class transaction extends uvm_sequence_item;
    rand bit [3:0] data;
    function new (string inst="transaction");
        super.new(inst);
    endfunction
    `uvm_object_utils_begin(transaction)
    `uvm_field_int(data,UVM_DEFAULT|UVM_BIN) //**
    `uvm_object_utils_end
endclass
module tb;
    transaction t;
    initial begin
        t=new();
        t.randomize();
        t.print();
    end
endmodule
```

## “Result”:-

Name	Type	Size	Value
transaction	transaction	-	@335
data	integral	4	'b110

## “Code Practising”:-

```
`include "uvm_macros.svh"
import uvm_pkg::*;
class transaction extends uvm_sequence_item;
    rand bit [3:0] data;
    function new (string inst="transaction");
        super.new(inst);
    endfunction
    `uvm_object_utils_begin(transaction)
    `uvm_field_int(data,UVM_DEFAULT|UVM_BIN) //**
    `uvm_object_utils_end
endclass
module tb;
    transaction t;
    initial begin
        t=new();
        t.randomize();
        t.print(uvm_default_tree_printer); //Tree Printer
    end
endmodule
```

## “Result”:-

```
transaction: (transaction@335) {  
  data: 'b110  
}  
simulation has finished. There are no more test vectors to simulate.
```

- Inbuilt implementations are less efficient than compared to do

## “Do Hooks Print”:-

## “Code Practising”:-

```
`include "uvm_macros.svh"  
import uvm_pkg::*;  
class transaction extends uvm_sequence_item;  
  rand bit [3:0] data;  
  function new (string inst="transaction");  
    super.new(inst);  
  endfunction  
  `uvm_object_utils_begin(transaction)  
  `uvm_field_int(data,UVM_DEFAULT|UVM_BIN) /**  
  `uvm_object_utils_end  
  virtual function void do_print(uvm_printer printer);  
    super.do_print(printer);  
    printer.print_field("data",data,$bits(data),UVM_DEC);  
  endfunction  
endclass  
module tb;  
  transaction t;  
  initial begin  
    t=new("t");  
    t.randomize();  
    t.print(uvm_default_tree_printer);//Tree Printer  
  end  
endmodule
```

## “Result”:-

```
t: (transaction@335) {  
  data: 'b110  
  data: 'd6  
}
```