VERIFICATION ACADEMY

UVM Basics

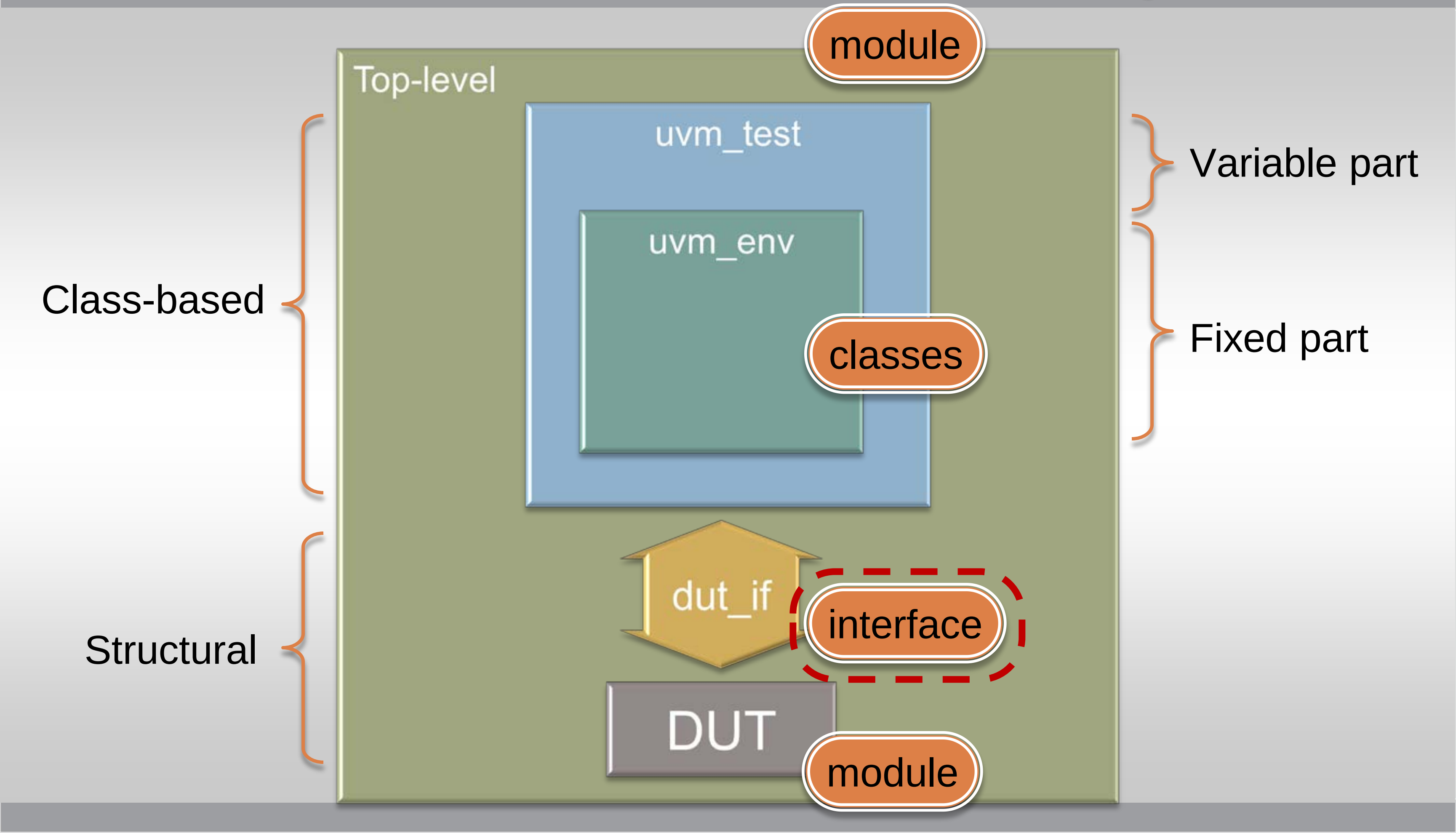
Connecting Env to DUT

John Aynsley
CTO, Doulos

academy@mentor.com
www.verificationacademy.com

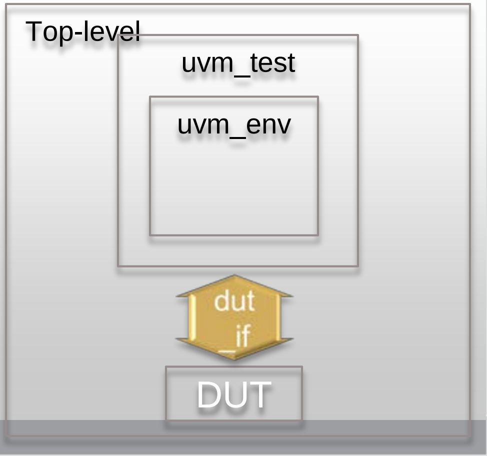
Mentor Graphics®

Connecting Env to DUT



Interface

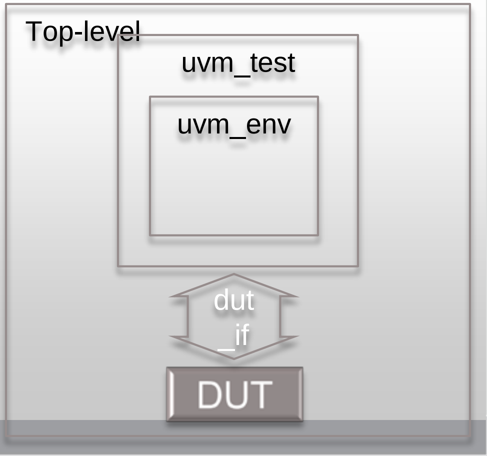
```
interface dut_if();

  logic clock, reset;
  logic data;

endinterface: dut_if
```



DUT

```
module dut(dut_if _if);

  always @(posedge _if.clock)
  begin
    ...
  end

endmodule: dut
```

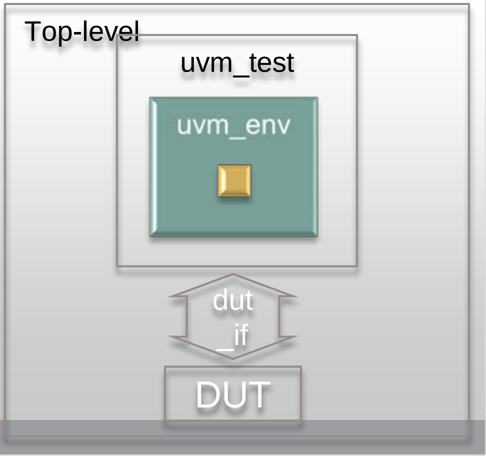
DUT can access the interface



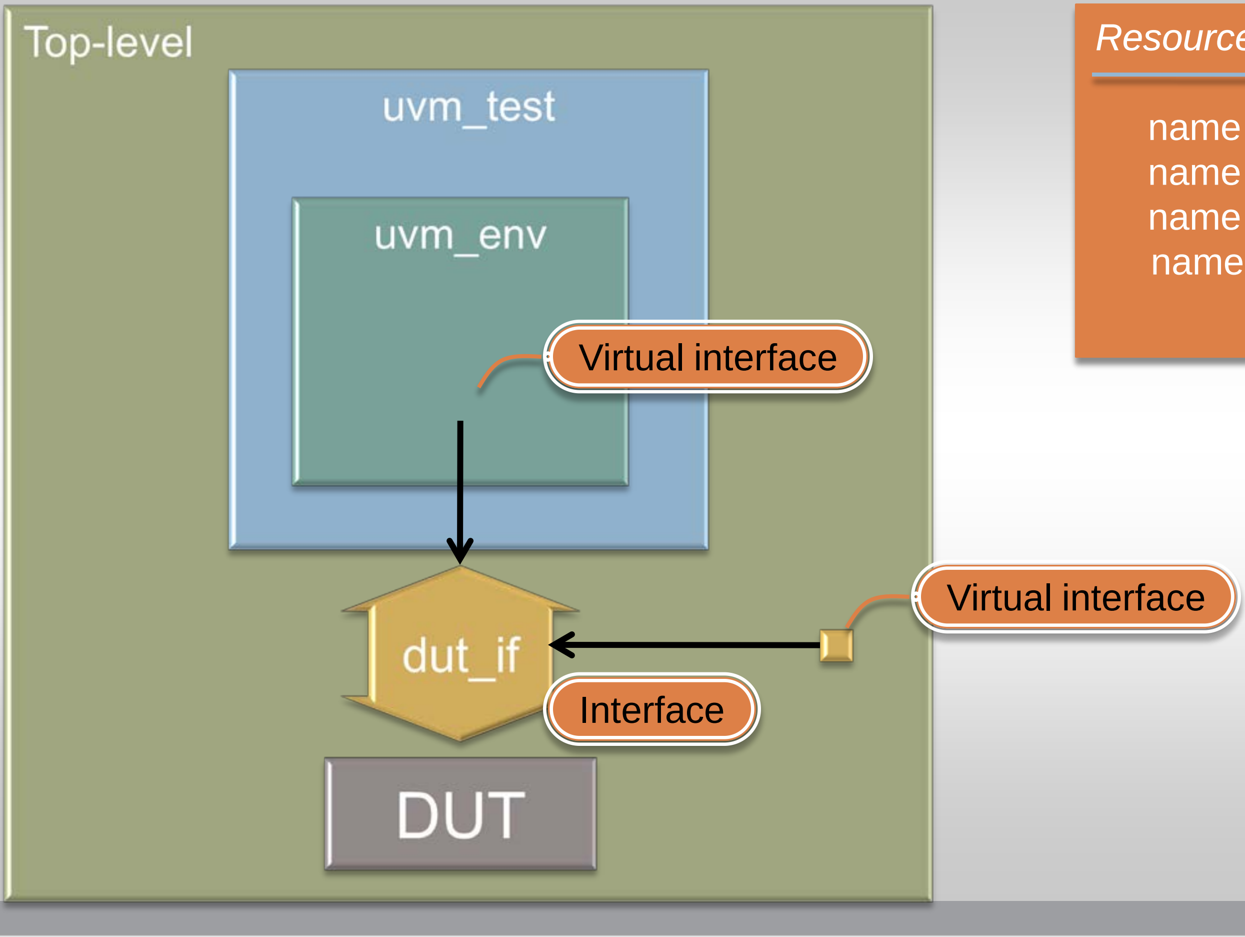
Accessing Pins through Virtual Interface

```
class my_env extends uvm_env;
 `uvm_component_utils(my_env)

 virtual dut_if dut_vi;

 function new(string name, uvm_component parent);
  ...
 function void build_phase(uvm_phase phase);
  ...
 task run_phase(uvm_phase phase);
  phase.raise_objection(this);
  #10 dut_vi.data = 0;
  #10 dut_vi.data = 1;
  #10 phase.drop_objection(this);
 endtask: run_phase
 ...
```

_vi = virtual interface



Resource Database



Resource Database

name = value
name = value
name = value
name =   .

uvm_resource_db ... set

```
module top;
...
dut_if dut_if1 ();

initial
begin: blk
  uvm_resource_db #(virtual dut_if)::set(

                    "dut_ifs",   "dut_vi",   dut_if1);
```

Type of value

Scope

Field name

Value

uvm_resource_db ... set

```
module top;
...
dut_if dut_if1 ();

initial
begin: blk
  uvm_resource_db #(virtual dut_if)::set(
                            "dut_ifs", "dut_vi", dut_if1);

  run_test("my_test");
end

endmodule: top
```

uvm_resource_db ... read_by_name

```
class my_test extends uvm_test;
  ...
  virtual dut_if dut_vi;
  ...
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);

    if( !uvm_resource_db#(virtual dut_if)::read_by_name(
                          "dut_ifs", "dut_vi", dut_vi) )
      ... // report error
```

Scope

Field name

Value

uvm_config_db ... set

```
class my_test extends uvm_test;
  ...
  virtual dut_if dut_vi;
  ...
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);

    if( !uvm_resource_db#(virtual dut_if)::read_by_name(
                          "dut_ifs", "dut_vi", dut_vi) )
      ... // report error
```

Type of value

```
    uvm_config_db #(virtual dut_if)::set(

                  this,   "*",   "dut_vi",   dut_vi);
```

Prefix

Path

Field name

Value

uvm_config_db ... set

```
class my_test extends uvm_test;
  ...
  virtual dut_if dut_vi;
  ...
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);

    if( !uvm_resource_db#(virtual dut_if)::read_by_name(
                              "dut_ifs", "dut_vi", dut_vi) )
      ...


    uvm_config_db #(virtual dut_if)::set(

                  this,    "*",    "dut_vi",    dut_vi);
```

"Flat"

Scope

Hierarchical

uvm_config_db ... set

```
class my_test extends uvm_test;
  ...
  virtual dut_if dut_vi;
  ...
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);

    if( !uvm_resource_db#(virtual dut_if)::read_by_name(
                          "dut_ifs", "dut_vi", dut_vi) )
      ... // report error
    uvm_config_db #(virtual dut_if)::set(
                         null, "*", "dut_vi", dut_if1);

    my_env_h = my_env::type_id::create("my_env_h", this);
  endfunction: build_phase
```

uvm_config_db ... get

```
class my_env extends uvm_env;
  ...
  virtual dut_if dut_vi;
  ...

  function void build_phase(uvm_phase phase);
    super.build_phase(phase);

    assert( uvm_config_db #(virtual dut_if)::get(

                this,   "",   "dut_vi",   dut_vi) );
```

Type of value

Prefix

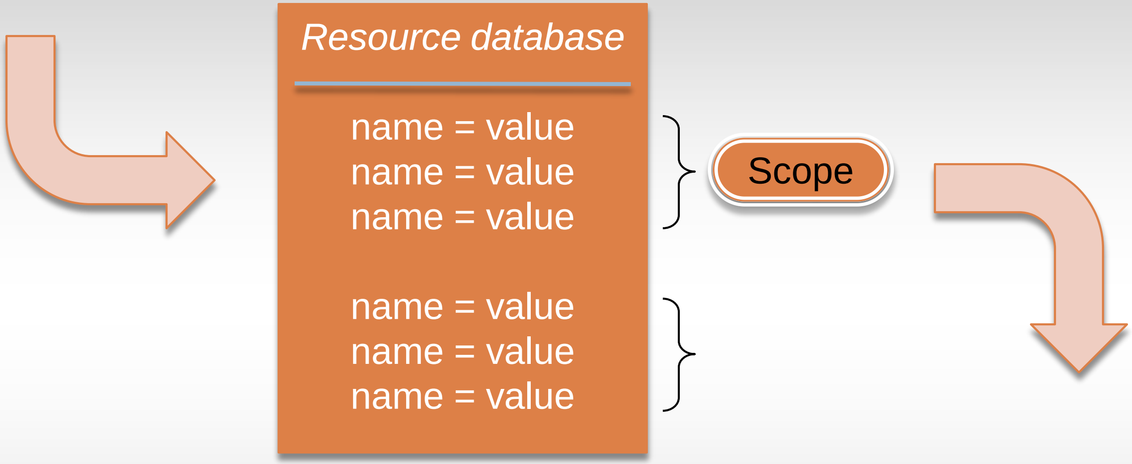
Path

Field name

Value

Setting and Getting Configuration Values

```
uvm_config_db #(T)::set(this, "path", "name", value);
```



Resource database

name = value
name = value
name = value

name = value
name = value
name = value

Scope

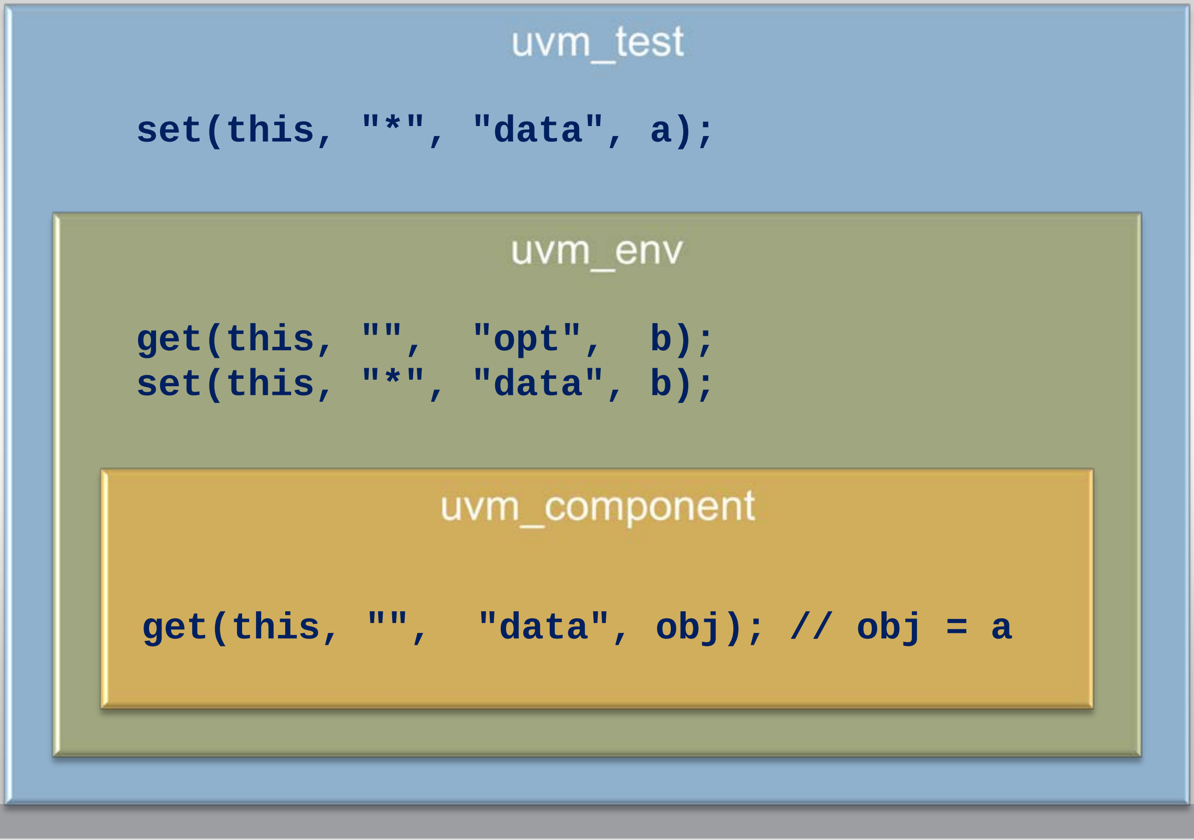
```
uvm_config_db #(T)::get(this,     "",  "name", value);
```

Scope

Path and field names can contain wildcards (glob pattern matching)

Hierarchical Configuration using uvm_config_db

Config information flows top-down, higher levels overriding lower levels

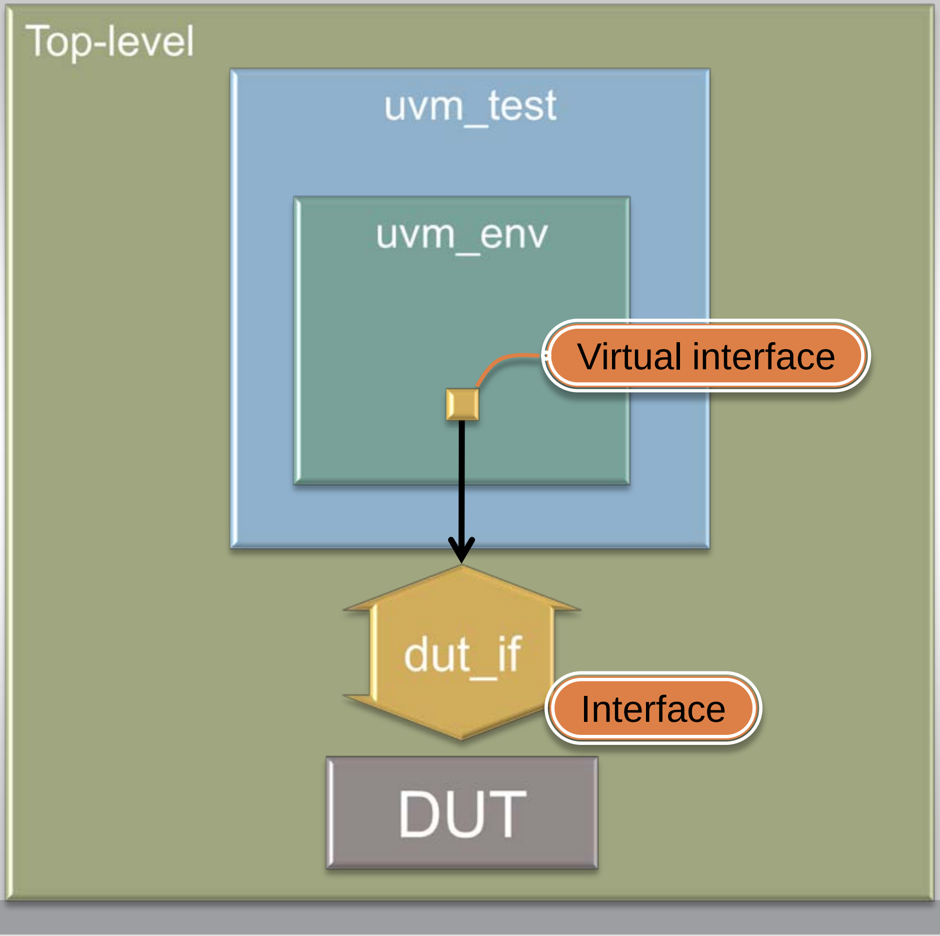


```
set(this, "*", "data", a);
```

```
get(this, "",  "opt",  b);
set(this, "*", "data", b);
```

```
get(this, "",  "data", obj); // obj = a
```

Summary



Resource database

name = value
name = value
name = value
name = ■ .

VERIFICATION ACADEMY

UVM Basics

Connecting Env to DUT

John Aynsley
CTO, Doulos

academy@mentor.com
www.verificationacademy.com

Mentor Graphics®