



LVM REINDENTATION TOOL

Version 1.01 (April 2025)





Overview

Overview

About LVM Reindentation Tool

Before AND After of Reindentation Tool

Steps to Integrate LVM Reindentation Tool

Terms and Conditions

About LVM Verification Enterprise



About LVM Reindentation Tool

WHAT IS Reindentation Tool FROM LVM

- System Verilog and UVM File Reindentation Tool that can reindent the file with configurable indent size.

OBJECTIVE OF LVM Reindentation Tool

- To help user to reindent the system verilog / UVM testbench / RTL files so that it is tidy, easy to read and avoid confusion.



Before AND After of Reindentation Tool

```
task x;
  if(a)
    for(1)
      if(a)
        b;
    else
      c;
endtask
```



```
task x;
  if(a)
    for(1)
      if(a)
        b;
    else
      c;
endtask
```

```
if(aaa)
  case(xx)
  h:
    if(1)
      case(1)
        1: begin
          a;
          b;
        end
      endcase
    endcase
end
```



```
if(aaa)
  case(xx)
  h:
    if(1)
      case(1)
        1: begin
          a;
          b;
        end
      endcase
    endcase
end
```

```
task x;
a;
l;
if()begin
b;
// xyz
if (1) begin
bb;
kk;
while (
(
a ||
b
)
)
@(posedge clk);
end
else
20us; // testing
testing;
end
endtask
```



```
task x;
a;
l;
if()begin
b;
// xyz
if (1) begin
bb;
kk;
while (
(
a ||
b
)
)
@(posedge clk);
end
else
20us; // testing
testing;
end
endtask
```



Before AND After of Reindentation Tool

```
if(a)
  if(b)
    if(c)
      1;
    else
      2;
```

```
if(a)
  if(b)
    if(c)
      1;
    else
      2;
```

```
interface #
(
  .a(1),
  .b(1)
)
(
  input a,
  input b
);
  a;
  b;
endinterface
```

```
interface #
(
  .a(1),
  .b(1)
)
(
  input a,
  input b
);
  a;
  b;
endinterface
```

```
`ifdef A
always @(posedge clk)
`else
always @(negedge clk)
`endif
testing;
```

```
`ifdef A
always @(posedge clk)
`else
always @(negedge clk)
`endif
testing;
```

```
if(
(
  (
    a &&
    b
  )
  ||
  (
    c &&
    d
  )
)
&&
a
&&
b
)
testing;
```

```
if(
(
  (
    a &&
    b
  )
  ||
  (
    c &&
    d
  )
)
&&
a
&&
b
)
testing;
```

```
for /* TODO */ (
  a;
  f;
  b /* TODO */
) begin
  aaa; // if
  bbb; // begin
end
```

```
for /* TODO */ (
  a;
  f;
  b /* TODO */
) begin
  aaa; // if
  bbb; // begin
end
```





Before AND After of Reindentation Tool

```
repeat(10)
while()
aaa;

if() `uvm_info()
else `uvm_error()
```

```
repeat(10)
    while()
        aaa;

    if() `uvm_info()
    else `uvm_error()
```

```
class ex_trial extends trial;
function display();
super.display();
$display ("0x%0h", value);
endfunction;

function new();
super.new ();
endfunction
endclass
```

```
class ex_trial extends trial;
    function display();
        super.display();
        $display ("0x%0h", value);
    endfunction;

    function new();
        super.new ();
    endfunction
endclass
```

```
sequence ev_B;
@(posedge clk)
B == 0;
endsequence
```

```
sequence ev_B;
    @(posedge clk)
    B == 0;
endsequence
```

```
`define AAA () \
if(aaa)\
if(bbb)\
```

```
`define AAA () \
    if(aaa)\
        if(bbb)\
```

```
case()
a: begin
    kkk;
end
b: lll;
c: begin kkk; end
d: begin
    kkk;
end
e:
    lll;
f:
    if(1)
    2;
g:
    for(1)
        2;
h:
    for(1)
        case(1)
            1: begin
                a;
                b;
            end
        endcase
i:
    `uvm_info()
j:
    `uvm_info( if ()
)
k:
    case(x)
        h:
            for(1) 2;
        i:
            `uvm_info()
    endcase
endcase
```

```
case()
a: begin
    kkk;
end
b: lll;
c: begin kkk; end
d: begin
    kkk;
end
e:
    lll;
f:
    if(1)
        2;
g:
    for(1)
        2;
h:
    for(1)
        case(1)
            1: begin
                a;
                b;
            end
        endcase
i:
    `uvm_info()
j:
    `uvm_info( if ()
)
k:
    case(x)
        h:
            for(1) 2;
        i:
            `uvm_info()
    endcase
endcase
```



Before AND After of Reindentation Tool

```
constraint aaa {  
  if() { bbb; }  
  else {  
    ccc;  
  }  
}
```

```
constraint aaa {  
  if() { bbb; }  
  else {  
    ccc;  
  }  
}
```

```
task a(  
  bit a, bit b,  
  bit c  
);  
if(a)  
  v();  
  
if(b)  
  bb;  
endtask
```

```
task a(  
  bit a, bit b,  
  bit c  
);  
if(a)  
  v();  
  
if(b)  
  bb;  
endtask
```

```
`ifdef AAA  
always () begin  
  `uvm_info ()  
`else  
always () begin  
  `uvm_info ()  
`endif  
  
end // always
```

```
`ifdef AAA  
always () begin  
  `uvm_info ()  
`else  
always () begin  
  `uvm_info ()  
`endif  
  
end // always
```

```
fork  
begin  
  if()  
    b;  
  for() c;  
end  
begin  
  a;  
end  
for()  
  a;  
if(a) begin  
  d;  
  e;  
end  
`uvm_info(asd)  
`uvm_info(asd  
)  
test;  
fork  
  fork  
    a;  
    b;  
  join  
    begin xx; end  
  begin  
    repeat(3) g;  
  end  
  
  repeat(3) g;  
  
  repeat(3)  
  g;  
join  
join
```

```
fork  
begin  
  begin  
    if()  
      b;  
    for() c;  
  end  
begin  
  a;  
end  
for()  
  a;  
if(a) begin  
  d;  
  e;  
end  
`uvm_info(asd)  
`uvm_info(asd  
)  
test;  
fork  
  fork  
    a;  
    b;  
  join  
    begin xx; end  
  begin  
    repeat(3) g;  
  end  
  
  repeat(3) g;  
  
  repeat(3)  
  g;  
join  
join
```



Steps to Integrate LVM Reindentation Tool

Step 1: Request a copy of LVM Reindentation tool from LVM (developer@lvmvip.com) while mentioning what is your EDA tool (vcs/xrun/irun/vsim)

Step 2: Copy the LVM Reindentation tool files to your bin / script area

```
cp <downloaded path>/lvm_reindentation <your bin / script area>
cp <downloaded path>/lvm_reindent.pl <your bin / script area>
chmod -R 755 <your bin / script area>/lvm_reindent*
```

Step 3: Modify the lvm_reindent.pl, by filling <fill your path here>/lvm_reindentation,

```
Before      : <fill your path here>/lvm_reindentation
After       : <your bin / script area>/lvm_reindentation
For example: /usr/bin/lvm_reindentation
```

Step 4: Execute the tool

indent size

Filename

Reindent on
original file

```
<your bin / scripts area>/lvm_reindent.pl -ts 4 -f <your sv file> -replace
```




Terms and Conditions

IMPORTANT NOTICE

LVM reserves the right to make changes without further notice to any product or specifications herein.

LVM does not assume any responsibility for use of any its products for any particular purpose, nor does LVM assume any liability arising out of the application or use of any its products.





About LVM VIP

1. At LVM, our mission is to enhance the performance and efficiency of design verification work in the industry by providing ultra-high-quality, low-cost VIPs. We offer a range of AMBA VIPs and serial VIPs, including:

AXI4 / AXI4-LITE, AHB, APB, TCM, USART, JTAG, AXI Stream

2. To learn more about our services, please contact us at

developer@lvmvip.com.

3. You can also visit our LinkedIn page at

<https://www.linkedin.com/in/lvm-vip-3444b21b5/>

