

Hi All -

With respect to BUG ID #26 - I am inclined to propose that importing enumerated types with the same literal as existing or other imported enumerate types shall be an error, but importing with the wildcard package import is not necessarily an error. Examples follow.

If all agree, I will write up the proposal and include the following examples.

Regards - Cliff

### Example

```
package p;
  typedef enum { FALSE, TRUE } bool_t;
endpackage

package q;
  typedef enum { ORIGINAL, FALSE } teeth_t;
endpackage

module tmp1d;
  import p::bool_t;
  import q::teeth_t;    // ERROR - bool_t & teeth_t both have an
                       // enumeration literal called "FALSE"
  ...
endmodule

module tmp2d;
  import p::*;
  import q::teeth_t;   // OK
  teeth_t myteeth;

  initial begin
    myteeth = FALSE; // OK - Direct reference to FALSE refers to the
                    // FALSE literal imported from q
  end
endmodule

module tmp3d;
  import q::teeth_t;
  typedef enum ( TRUE, FALSE ) story_t;
                       // ERROR - teeth_t from package q and
                       // local typedef for story_t both have
                       // an enumeration literal called "FALSE"
  ...
endmodule
```

```
module tmp4d;
  import p::*;
  typedef enum ( TRUE, FALSE ) story_t;
  story_t tale;

  initial begin
    tale = FALSE;    // OK - Direct reference to FALSE refers to the
                    // FALSE literal from the locally declared story_t
  end
endmodule

module tmp5d;
  import p::*;
  bool_t test;      // uses bool_t from package p
  typedef enum ( TRUE, FALSE ) story_t;
                    // ERROR - bool_t from package p and
                    // local typedef for story_t both have
                    // an enumeration literal called "FALSE"

  ...
endmodule
```