

# System Verilog 設計・検証ツール製品紹介

---

---

Document Revision: 2.3 2022.06.12



## 注意事項

ここで紹介する内容は、ソフトウェア最新版と異なる場合があります。ご了承ください。

### SystemVerilog 設計・検証ツール製品紹介

© 2018, 2022 アートグラフィックス  
〒124-0012 東京都葛飾区立石 8-14-1  
[www.artgraphics.co.jp](http://www.artgraphics.co.jp)

### SystemVerilog Tools Summary

© 2018, 2022 Artgraphics. All rights reserved.  
8-14-1, Tateishi, Katsushika-ku, Tokyo, 124-0012 Japan  
[www.artgraphics.co.jp](http://www.artgraphics.co.jp)

弊社製品、及び、業務案内に関するご質問には下記の連絡先をご利用下さい。  
連絡先：[contact.us@artgraphics.co.jp](mailto:contact.us@artgraphics.co.jp)

## 目次

|       |                                  |   |
|-------|----------------------------------|---|
| 1     | 製品の目的 .....                      | 1 |
| 2     | 製品概要 .....                       | 1 |
| 2.1   | SVDESIGNER .....                 | 1 |
| 2.2   | SVCHECKER .....                  | 1 |
| 3     | 機能紹介 .....                       | 2 |
| 3.1   | SVDESIGNER と SVCHECKER の差異 ..... | 2 |
| 3.2   | IDE としての GUI .....               | 2 |
| 3.2.1 | SVDesigner .....                 | 2 |
| 3.2.2 | SVChecker .....                  | 4 |
| 3.3   | テキストエディタの特徴 .....                | 5 |
| 3.4   | データタイプナビゲータ .....                | 5 |
| 3.5   | 自動バックアップ機能 .....                 | 6 |
| 3.6   | HTML 文書生成機能 .....                | 6 |
| 3.7   | デザインスタイルチェック .....               | 7 |
| 3.7.1 | リントとデザインスタイルチェック .....           | 7 |
| 3.7.2 | デザインスタイルチェックと UVM .....          | 7 |
| 3.8   | 参照機能 .....                       | 8 |
| 3.8.1 | コードスニペット .....                   | 8 |
| 3.8.2 | ブックマーク .....                     | 9 |

## 1 製品の目的

最先端の検証技術（VMM、UVM 等）を効果的に適用する為には、最適な GUI の使用が不可欠です。SystemVerilog IDE、及び、SVChecker は時代に即した機能を提供します。この文書では GUI 機能を中心に紹介します。更に詳細な情報をお求めの場合には、ご連絡下さい。

## 2 製品概要

### 2.1 SVDesigner

SVDesigner は設計・検証機能を統合した GUI ベースの開発環境です。以下の様な機能を備えています。

- デザインマネージャー（プロジェクトのファイルを管理し、コンパイルからシミュレーションまでの一連の処理を行ないます）
- コード開発と支援機能（シンタックスハイライト、ナビゲータ、リント、デザインスタイルチェック）
- クイック参照機能（コードスニペット、ブックマーク）
- SystemVerilog コンパイラー、及び、シミュレータ
- 制約付きランダムステイミュラス生成
- ファンクショナルカバレッジ
- アサーション
- コードカバレッジ
- 検証ビューワー（VCD、カバレッジ、アサーション）
- UVM サポート（UVM は複雑なコンストラクトである為、GUI の使用は不可欠です）
- HTML 文書生成（SystemVerilog 記述を HTML 形式に変換します）
- ワークベンチ（データ構造ウィザード、UVM クラスウィザード、テストベンチ生成等）
- 自動バックアップ機能
- ソフトウェア更新機能

### 2.2 SVChecker

SVChecker は SystemVerilog による設計・検証を支援する開発環境です。SVDesigner と同様に、SVChecker は検証機能を持ちますが、全体的に軽量 GUI として実装されています。

- プロジェクトマネージャー
- コード開発と支援機能
- クイック参照機能（コードスニペット、ブックマーク）
- SystemVerilog コンパイラー、および、シミュレータ
- 制約付きランダムステイミュラス生成
- ファンクショナルカバレッジ
- アサーション
- UVM サポート（UVM は複雑なコンストラクトである為、GUI の使用は不可欠です）
- HTML 文書生成（SystemVerilog 記述を HTML 形式に変換します）
- 検証ビューワー（VCD、カバレッジ、アサーション）が別アプリとして標準装備
- ワークベンチ（データ構造ウィザード、UVM クラスウィザード、テストベンチ生成等）
- 自動バックアップ機能
- ソフトウェア更新機能

### 3 機能紹介

IDE としての特徴は、使用者の生産性を向上する事を目的としている事です。コード入力の負荷軽減、ソースコードを見易くする支援、参照資料の管理、プロジェクトに必要なファイル管理等様々な観点から設計者・検証技術者の支援をします。以下では、GUI としての先進的な機能を紹介します。

#### 3.1 SVDesigner と SVChecker の差異

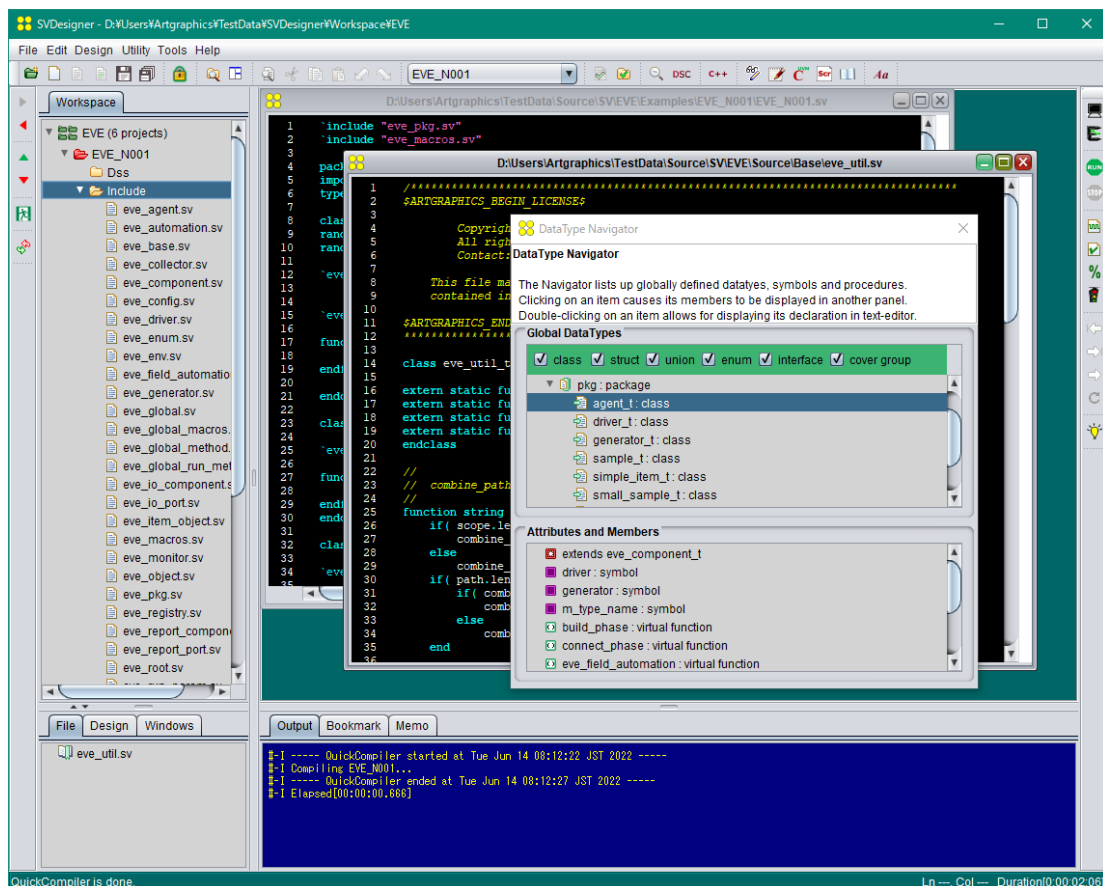
二つの製品の違いを以下の様に総括する事が出来ます。

|       | SVDesigner                           | SVChecker                          |
|-------|--------------------------------------|------------------------------------|
| 画面管理法 | MDI<br>(Multiple Document Interface) | TDI<br>(Tabbed Document Interface) |
| 開発機能  | 有り                                   | 有り                                 |
| 検証機能  | 有り                                   | 有り                                 |

#### 3.2 IDE としての GUI

##### 3.2.1 SVDesigner

SVDesigner は下図に示す様な外観を持ちます。



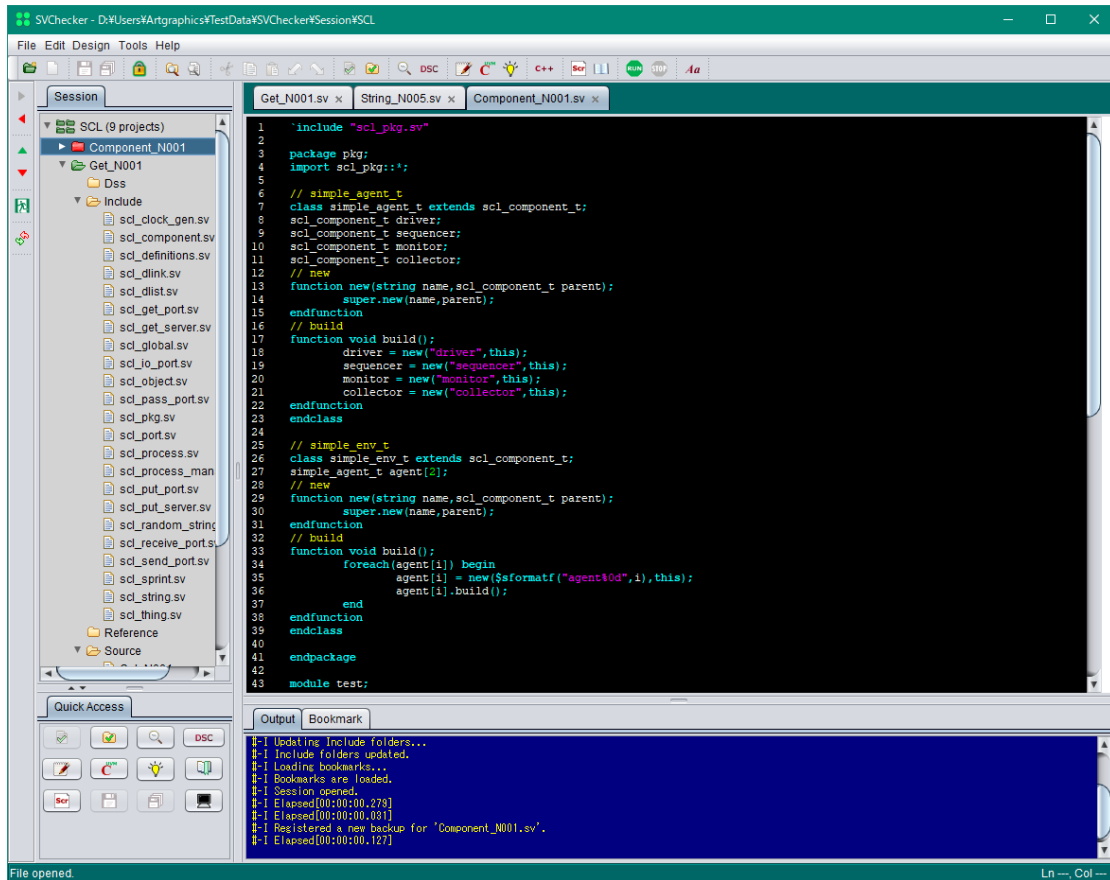
テキストエディタのウィンドウは、ファイル毎に独立した大きさと配置を持つので、必要な数のファイルを同時に開いて作業をする形態に適しています。

ファイルはプロジェクト単位に管理されているので、コンパイルするために必要な操作は IDE が全て行います。ユーザは、コンパイルボタンをクリックするだけで済みます。コンパイルが終了すると、実行モジュールの使用が可能となるので、何時でもシミュレーションを開始できます。

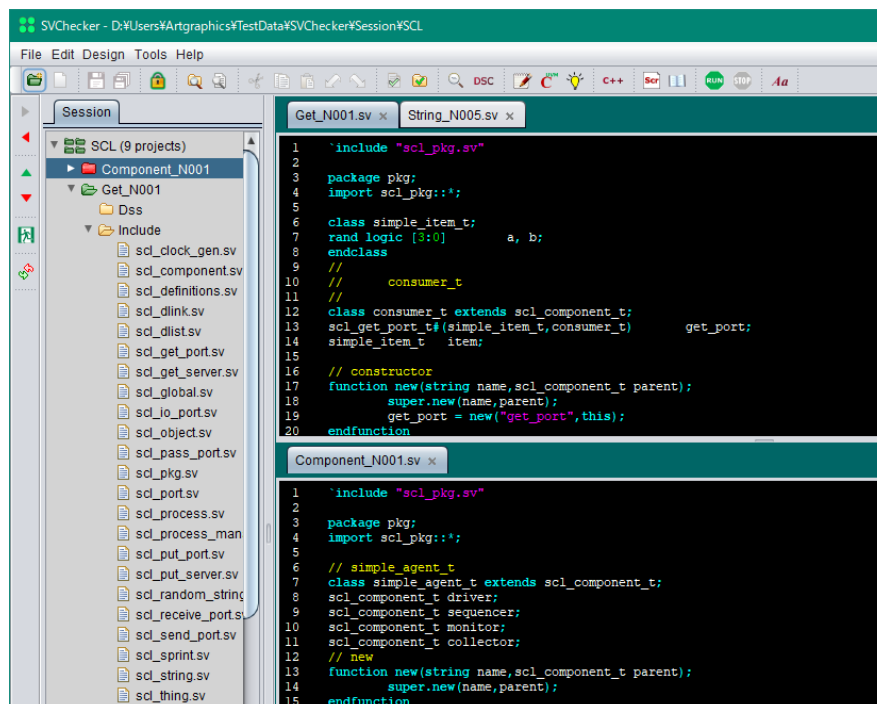


### 3.2.2 SVChecker

SVChecker は下図に示す様な外観を持ちます。

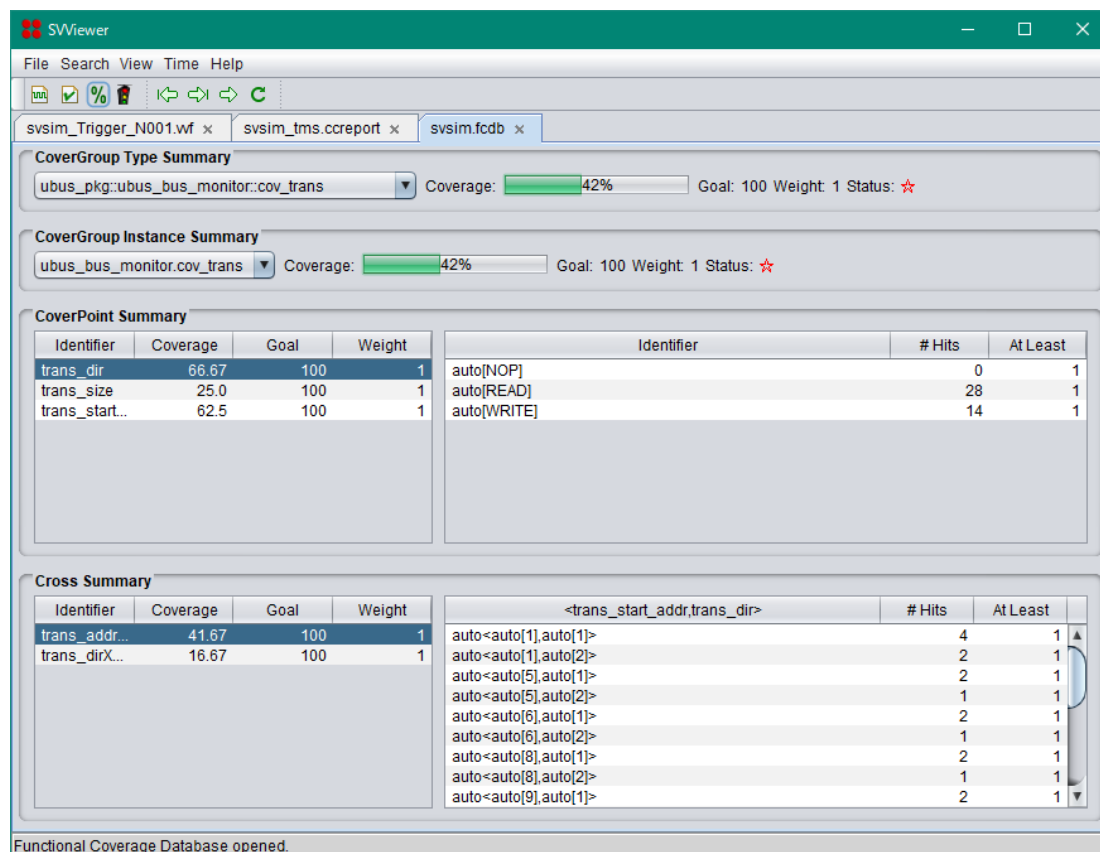


テキストウィンドウは、キャンバス全体に表示されるのが標準的な動作ですが、タブを操作してウィンドウの配置を変更できます。IDE は今日の UI の慣習に従っているため、初心者でも操作は簡単です。下図は、タブ操作により配置を変更した状態を示しています。



SVDesigner と同様に、ファイルはプロジェクト単位に管理されているので、コンパイル作業は簡単です。シミュレーション実行のステップも SVDesigner と全く同様です。但し、検証ビューワーは、別のアプリケーションとなっています。これにより、SVChecker のレイアウトは見易くなっています。

下図は、別アプリケーションの検証ビューワーが、ファンクショナルカバレッジを表示している状態を示しています。



### 3.3 テキストエディタの特徴

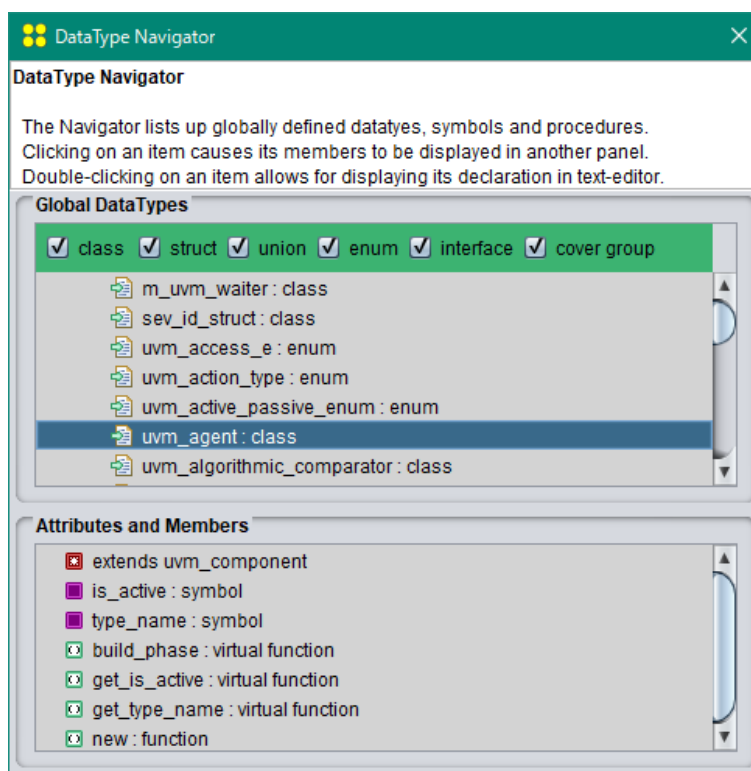
テキストエディタの特徴を以下の様に纏める事が出来ます。

- SystemVerilog を意識したシンタックスハイライト機能を備えています。
- 行番号を最新状態に保ちます。
- 行の折り畳み機能を備えています。
- テキストエディタの画面を2分割する事が出来ます。
- 直前に編集した行、及び、選択文字列を管理します。
- 懇切丁寧にコンパイルエラーを表示します。
- モジュール、クラス、メソッド等のヘッダにハイライトを設定する事が出来ます。
- トークン上でマウスを静止するとトークンの情報が表示されます。
- 入力に従いナビゲータがガイドします。
- 各種のショートカットキーを備えています。
- その他クラシックな編集機能を備えています。

### 3.4 データタイプナビゲータ

強力なクロスリファレンス機能です。特に UVM を使用する場合、必須の機能です。UVM Reference マニュアルの代用としての機能があります。下図は、データタイプ・ナビゲータを起動して、uvm\_pkg を表示している状態を示しています。





ナビゲータは、uvm\_pkg に定義されている全てのクラスにアクセスする機能を提供します。それぞれのクラス名をクリックするとクラス内で定義されているメンバー、及び、メソッドが下のパネルに表示されます。クラス名をダブルクリックするとクラスの内容がテキストエディタに表示されます。

### 3.5 自動バックアップ機能

SystemVerilog IDE は自動バックアップ機能を備えています。この機能により、ファイルが壊滅状態になってもセッション開始時点のファイルを復元する事が出来ます。バックアップは自動的に行なわれる為、使用者には手間がかかりません。何よりも、バックアップを取る機会を忘れる事はありません。

バックアップはシステム固有の場所に保管されます。バックアップの作成は自動的に行なわれる為、ユーザに求められる特別な操作はありません。ユーザがバックアップファイルを直接操作する為には、BackupMgr を使用します。以下の操作が可能です。

- ① バックアップファイルの内容を表示する事
- ② ソースファイルとバックアップファイルの差異を調べる事
- ③ バックアップファイル同士の比較を行う事
- ④ ソースファイルをバックアップファイルから復元する事
- ⑤ ソースファイルから新しいバックアップを作成する事
- ⑥ 不要になったバックアップを削除する事
- ⑦ プロジェクトに使用されているファイルのバックアップを整理する事
- ⑧ バックアップが保管されているディレクトリを知る事
- ⑨ バックアップが使用しているディスクスペースの大きさを知る事

### 3.6 HTML 文書生成機能

HTML 文書生成機能が改訂されました。この機能は以下の様な特徴を持ちます。

- ① モジュール、クラス等のヘッダ直前に記述されているコメントから仕様を抽出して文書

- を生成します。
- ② コメント部分に記述する仕様には **HTML** タグを使用する事が出来るので見易い文書を作成する事が出来ます。
  - ③ ファイル、及び、ディレクトリ単位に文書を生成する事が出来ます。
  - ④ プロジェクト全体の文書を生成するには、プロジェクトのルートディレクトリを指定するだけで全ての文書を作成する事が出来ます。
  - ⑤ コンパイラを使用せずに文書を作成する為、未定義のモジュール、及び、クラスを参照していてもエラーは発生しません。また、`include しているファイルが存在しなくても構いません。
  - ⑥ 文書は **CSS** ファイルを使用しています。**CSS** ファイルを編集する事により、ユーザ固有の文書書式を実現する事が出来ます。

下図は、生成した文書を Google Chrome で表示している例です。

The screenshot shows a web browser displaying the 'SystemVerilog Reference' website. The page title is 'SystemVerilog Reference' and the main heading is 'Module fifo\_controller'. On the left, there is a sidebar with a list of modules: FIFO, fifo\_controller, temp\_converter, temp\_filter, temp\_monitor\_unit, and temp\_register. The main content area includes a 'Synopsis' section with a description of the signal 'select' and its behavior, followed by a 'Module Declaration' section containing the Verilog code for the 'fifo\_controller' module. The code defines inputs for data, select, read, write, and full signals, and outputs for data and read signals. At the bottom of the page, a footer states: 'This document has been generated by Artgraphics' SystemVerilog Reference.'

### 3.7 デザインスタイルチェック

#### 3.7.1 リントとデザインスタイルチェック

リント機能は、環境及び状況に依存しない一般的に適用されるチェック機能を意味します。一方、デザインスタイルチェックは、設計手法、環境、設計条件等に依存するチェック機能を意味します。前者はチェック条件を必要としませんが、後者はチェック条件を必要とします。リント機能とデザインスタイルチェック機能を併用する事により、SystemVerilog 記述上の隠れた落とし穴を早期に解消する事が出来ます。

#### 3.7.2 デザインスタイルチェックと UVM

デザインスタイルチェック (DSC) に UVM 検証コードを確認する機能を追加しました。この機能を利用する事により、基本的で、且つ、見落とし易い問題を早期に解決する事が出来ます。UVM には、守るべきルールが幾つか存在しますが、それらの多くに気が付かないのが現

状です。

例えば、メソッドロジークラス `uvm_driver` を利用してドライバーを定義する時、``uvm_component_utils` マクロを忘れる事があります。忘れても、SystemVerilog コンパイラーではエラーとなりませんが、その後、ファクトリを使用してドライバーのインスタンスを作成するとエラーが発生します。この時点でのエラーの原因究明は複雑になります。マクロが引き起こした問題である為、エラーメッセージの内容は、一般的には、意味不明で解決に時間を要します。

別の例を挙げます。ドライバーを定義すると `virtual interface` の宣言が必要です。宣言を忘れても、SystemVerilog コンパイラーではエラーとなりません。然し、後（例えば、シミュレーション時）にエラーが判明します。更に、話を複雑にするのは ``uvm_field_` マクロです。宣言した `virtual interface` に対して ``uvm_field_` マクロを定義すると、今度は SystemVerilog コンパイラーがエラーを発行します。`interface` はクラスではない為、``uvm_field_` マクロを使用するのは間違いです。然し、一般に、UVM に関する深い知識が無いとエラーを解析する事は難しいです。これらの例が示す様に、UVM を簡単に使用する事は出来ません。技術者にはフラストレーションが溜るばかりです。

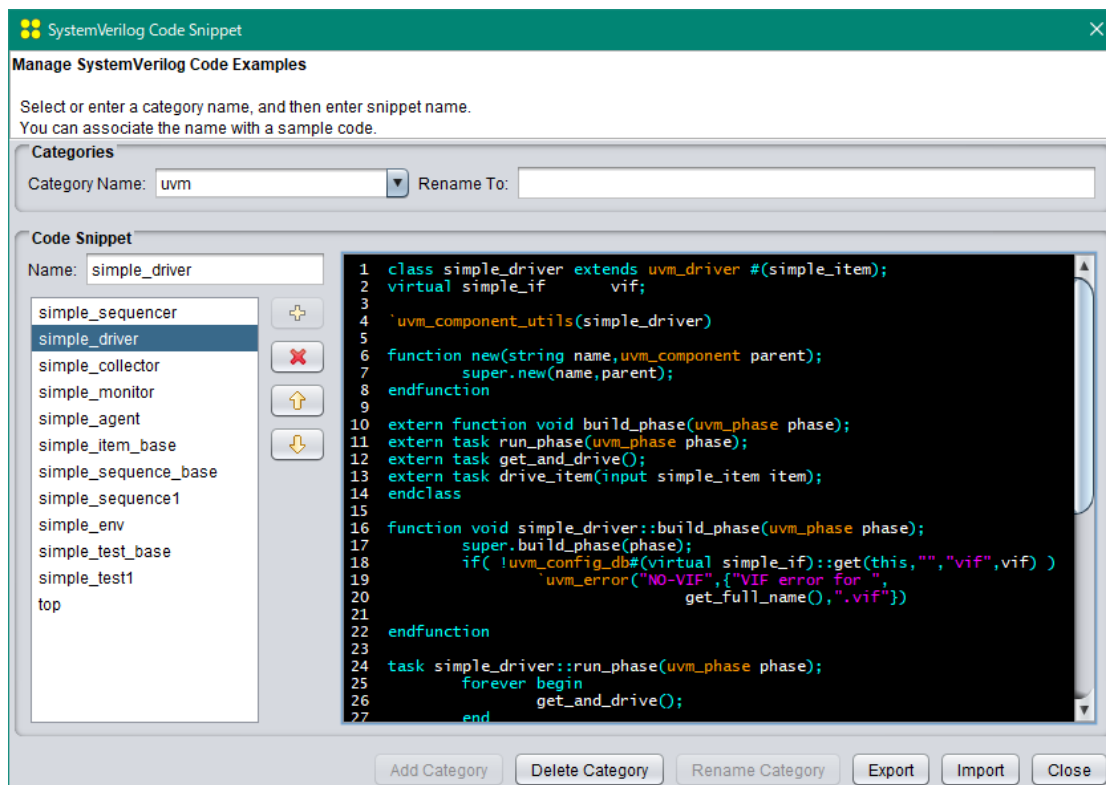
DSCはこの様な潜在的な問題を未然に防ぎ、技術者は冷静さを保つ事が出来ます。UVM 検証コードを記述したら、DSC を実行して下さい。DSC は基本的なルールチェックを施工します。

### 3.8 参照機能

#### 3.8.1 コードスニペット

スニペットは、テキスト文字列から構成された情報群を意味します。例えば、サンプルコードの断片がスニペットに相当します。マニュアルに書かれた複雑なシンタックスを読み返すよりも過去の使用例を見た方が確実に理解が速いのが常です。

登録したコードスニペットをソースコードに簡単に貼り付ける事が出来ます。また、Export、及び、Import 機能を使用する事によりコードスニペットを他の人と共有する事が出来ます。下図は、UVM のスニペット例です。



### 3.8.2 ブックマーク

コードスニペットはユーザ指定の名称に対してコードの断片をマップする機能です。それに対して、ブックマーク機能はユーザ指定の名称に対してソースファイル内の行番号を割り当てます。コードスニペット機能、及び、ブックマーク機能の組み合わせを利用する事により、コード開発時の生産性を向上する事が出来ます。次の様に機能します。

- ブックマークをダブルクリックするとファイルが開き、該当する行がテキストエディタ内に選択表示されます。
- ブックマークを目的別に分類する事が出来ます。
- ブックマークに名称を設定する事が出来ます。

頻繁にアクセスするファイルをブックマークとして登録しておくると便利です。