

Tutoriel HDL Companion



Tutoriel HDL Companion - VHDL

EDA Expert

30 Villa Remond
94250 Gentilly
France
Telephone: +33 (0) 1 49 84 86 26

Site web: www.eda-expert.com

Service clientèle: contact@eda-expert.com

Même si HDL Works BV a testé le logiciel et révisé la documentation, HDL Works BV n'offre aucune garantie ou déclaration, expresse ou implicite, à l'égard de ce logiciel et de sa documentation, qualité, performance, qualité marchande ou d'adéquation à un usage particulier.

Aucune partie de ce document ne peut être copié ou reproduit sous quelque forme ou par quelque moyen sans l'autorisation écrite préalable de HDL Works BV. Les informations contenues dans ce document sont sujettes à modification sans préavis et ne doivent pas être interprétées comme un engagement de HDL Works BV. HDL Works BV n'assume aucune responsabilité pour les erreurs qui pourraient apparaître dans ce document. HDL Works BV n'offre aucune garantie d'aucune sorte à l'égard de ce matériel, y compris, mais sans s'y limiter, les garanties implicites de qualité marchande et d'adéquation à un usage particulier. Le logiciel décrit dans ce document est droits d'auteur par HDL Works BV, Ede, aux Pays-Bas.

Le logiciel décrit dans ce document est fourni sous licence et ne peut être utilisé ou copié que conformément aux termes d'un accord de licence du logiciel.

HDL Works BV reconnaît des marques commerciales ou des marques déposées d'autres organisations pour leurs produits et services respectifs.

Version du logiciel: HDL Companion 2.8.



Table des matières

1 Introduction	4
2 Démarrage de l'application	5
3 Ajout des fichiers	7
4 Fenêtre principale d'HDL Companion	8
5 La fenêtre générale	9
6 La fenêtre détaillée	14
7 Trouver des objets et des textes	18
8 Le « Linting » d'un projet	21
9 Générer un fichier HTML	22
10 Simulation	24
11 Enregistrer et quitter	28
12 Informations contact	29

1 Introduction

Ce document est une introduction à l'utilisation de HDL Companion. Sur la base d'un exemple de code pour un contrôleur I2C, qui est inclus dans le logiciel HDL Companion, il est expliqué comment HDL Companion peut être utilisé pour obtenir rapidement un aperçu lorsque vous êtes confronté à une collection de fichiers HDL désorganisés qui forment un projet complet. Nous allons commencer par la mise en place d'un projet. Ensuite, nous allons explorer les différents points de vue offerts par HDL Companion. Nous continuerons ce tutoriel avec les fonctionnalités de recherche et de remplacement puis avec la fonction de *Linting* HDL. En outre, il est expliqué comment mettre en place un environnement de simulation et écrire des scripts Tcl qui fonctionnent sur le code HDL. Pour un aperçu plus complet des fonctions d'HDL Companion, merci de vous référer au manuel d'utilisateur.

Toutes les actions de l'utilisateur dans ce tutoriel sont précédées par le symbole 

Si vous avez des questions, veuillez nous appeler ou nous envoyer un e-mail. Nous serons heureux de vous aider !

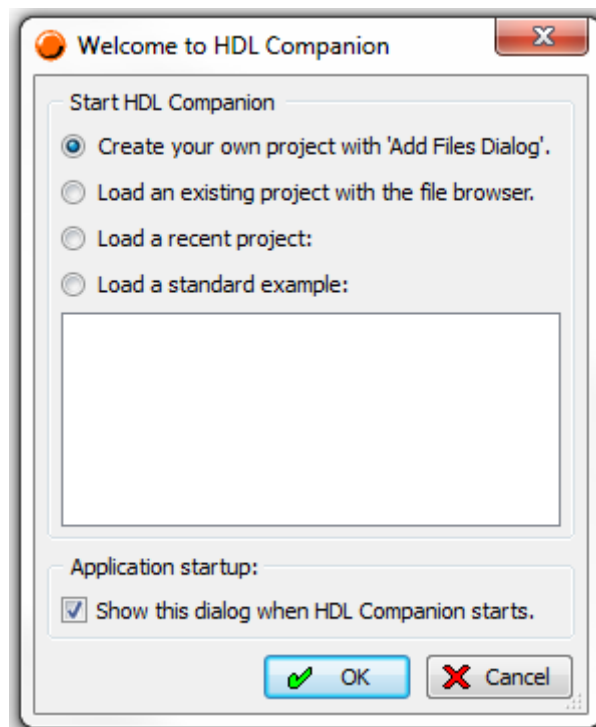
2 Démarrage de l'application

Vous pouvez choisir de créer un projet ou d'importer un projet déjà existant.

2.1 Créer votre propre projet



Lancer **HDL Companion** et sélectionnez **Créer votre propre projet** dans la boîte de dialogue **Welcome HDL Companion** et appuyez sur le bouton **OK**.

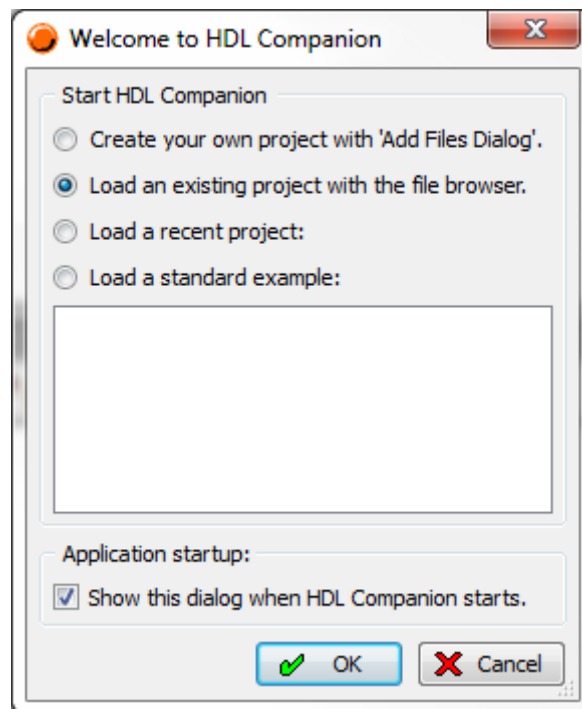


Boîte de dialogue de démarrage

2.2 Importation de votre propre projet



Pour importer votre propre projet lorsque la boîte de dialogue **Welcome HDL Companion** s'ouvre, sélectionnez **Charger un projet existant avec le navigateur de fichier** puis cliquez sur le bouton **OK**.




Boîte de dialogue de démarrage

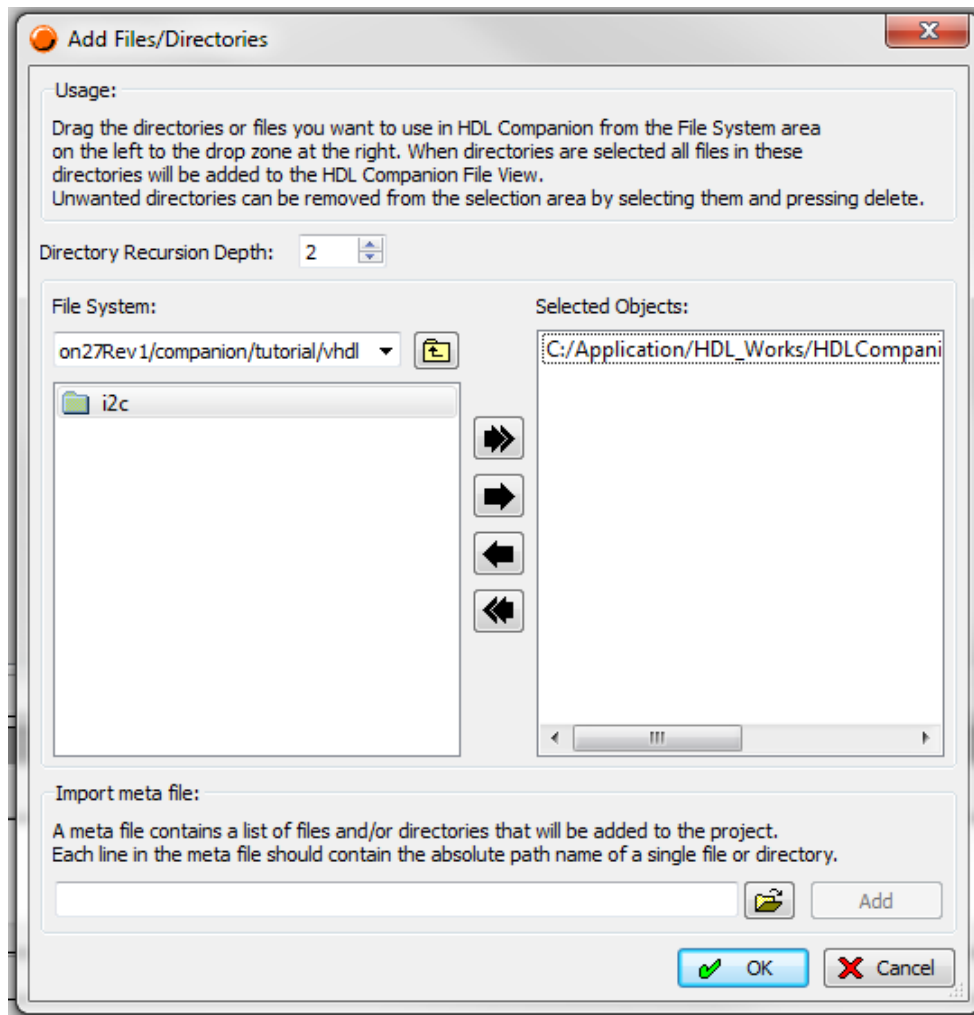
3 Ajout des fichiers

La boîte de dialogue d'ajout de fichiers s'ouvre.

 Réglez la **profondeur de récursion des dossiers** sur 2

Ce nombre signifie que tous les fichiers trouvés jusqu'à ce nombre (ici 2) de niveau en dessous du dossier sélectionné seront inclus au projet.

 Dans la colonne **File system**, naviguez jusqu'au dossier *companion/tutorial/vhdl* du dossier d'installation de HDL Companion. Faites glisser le dossier « i2c » dans la colonne des **objets sélectionnés** puis cliquez sur **OK**.



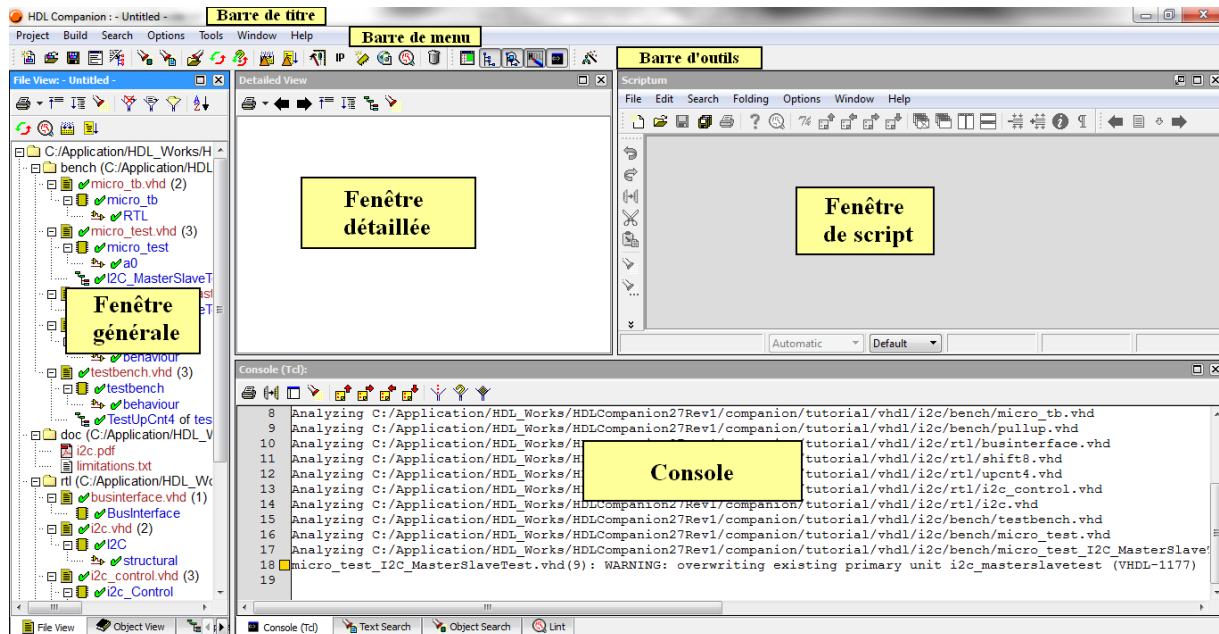
Boîte de dialogue d'ajout de fichiers

HDL Companion commence l'analyse des fichiers HDL.

Une fois l'analyse finie, la fenêtre **Project Metrics** s'ouvre, elle vous montre les informations du code HDL analysées. Si tout s'est déroulé correctement, vous pouvez voir qu'HDL Companion a trouvé 10 fichiers VHDL.

 Fermez la fenêtre Project Metrics.

4 Fenêtre principale d'HDL Companion



Fenêtre principale

Les éléments suivants sont affichés dans la fenêtre principale :

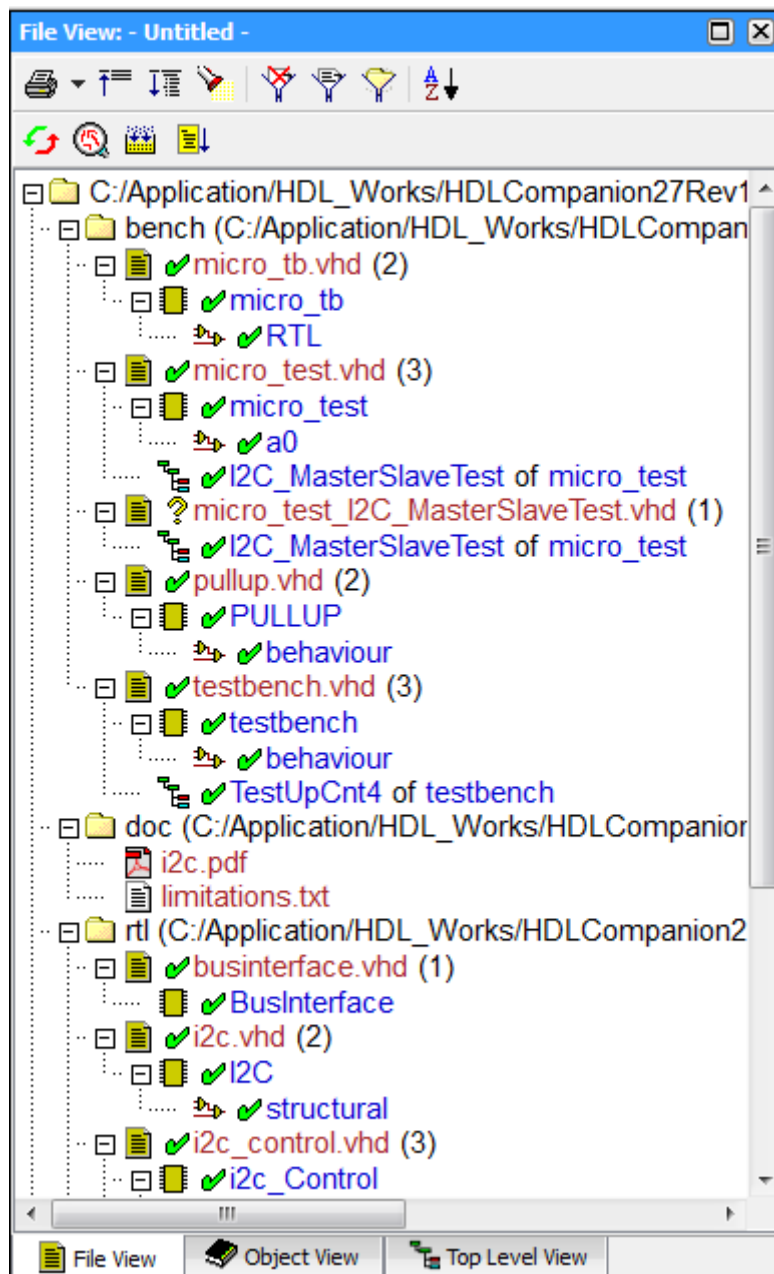
- la **barre de titre** affiche le nom du programme et le projet en cours.
- la **barre de menu** répertorie les menus principaux disponibles.
- la **barre d'outils** contient des boutons faciles à utiliser. Ils sont décrits en détail dans le manuel d'utilisateur d'HDL Companion
- la **fenêtre générale** possède trois vues : la première, la vue des fichiers qui sont actuellement dans le projet. La deuxième, une vue « objet » qui montre les objets HDL trouvés dans les fichiers. La troisième, la « Top Level View » qui elle, montre les entités, configuration et modules du design qui ne sont pas instanciés et donc considérés comme un nœud de haut niveau. Pour sélectionner la vue voulue, il suffit d'utiliser les trois onglets en bas de l'écran.
- la **fenêtre détaillée** nous permet de voir les détails d'un objet sélectionné dans la fenêtre générale.
- la **fenêtre de script** montre le code HDL ou n'importe quel autre type de fichier texte qui fait partie du projet. Cette fenêtre permet aussi d'éditer le code.
- la **console** affiche les messages générés par HDL Companion. Il est également possible de se servir de cette console pour entrer les commandes d'outils.

5 La fenêtre générale

Dans les prochains paragraphes nous allons décrire les 3 vues disponibles dans la vue générale.

5.1 La vue des fichiers

Si tout s'est bien passé HDL Companion a trouvé 10 fichiers VHDL et la vue de fichier ressemble à ceci.



Vue des fichiers

Les informations visibles sont basées sur les recherches des analyseurs d'HEL Companion. Elles n'ont pas nécessairement besoin d'un fichier syntaxiquement correct ou complet. Lorsque qu'un Tick Vert apparaît sur la ligne du fichier cela signifie que la syntaxe du fichier est correcte.

HDL Companion affiche tous les fichiers trouvés depuis le dossier de départ. Pour les fichiers VDHL, il montre aussi les unités (telles que les entités, architectures etc...) présentes dans le fichier. Les autres fichiers sont listés et associés à leur icône s'ils sont reconnus.

Dans le cas de ce tutoriel, 3 fichiers non-HDL sont présents. Dans le dossier *doc*, vous trouverez *i2c.pdf* et *limitations.txt*.





Double-cliquez sur l'étiquette *i2c.pdf*.

Une visionneuse de PDF devrait ouvrir le fichier. C'est un exemple de la manière dont HDL Companion gère les fichiers non-HDL. Dans l'outil d'enregistrement vous pouvez définir les noms et extensions de fichiers et la manière dont HDL Companion doit s'en charger.



Fermez la visionneuse de PDF.

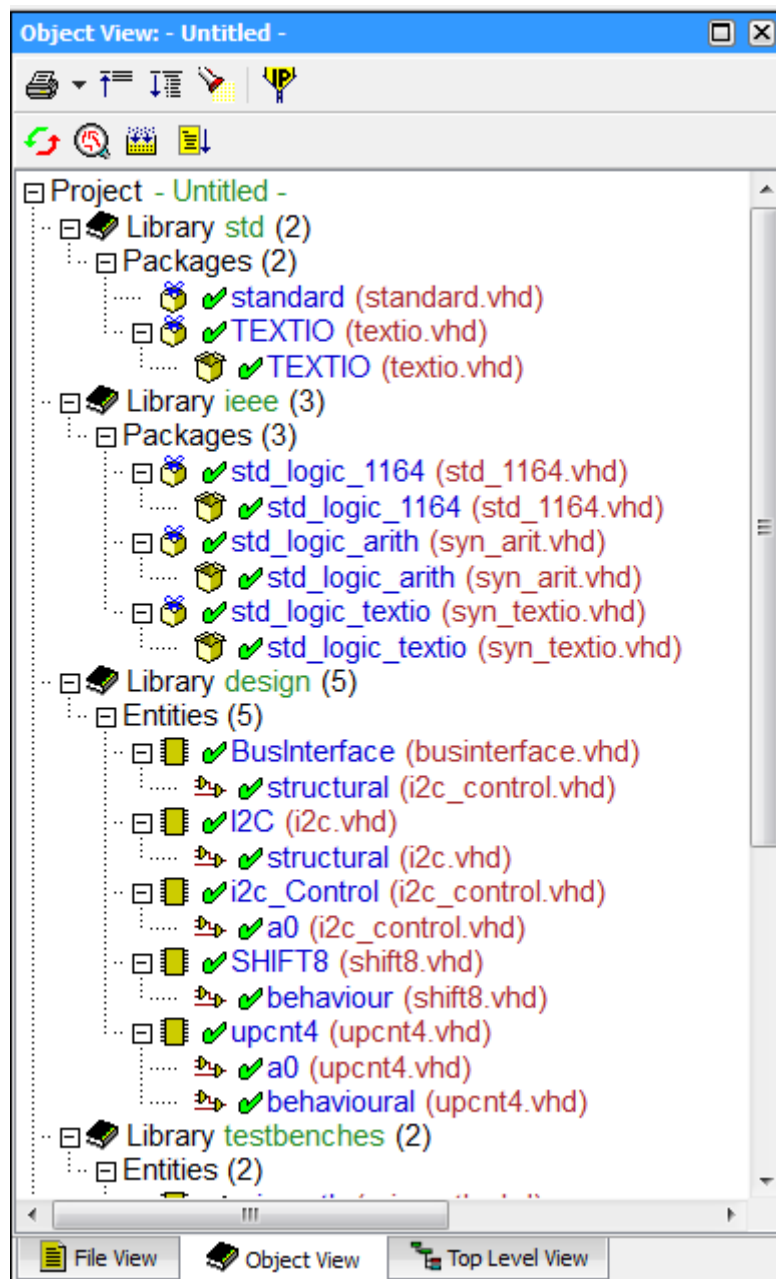
Les réglages par défaut d'HDL Companion, sont tels que tous les fichiers et dossier sont visibles. Grâce à l'outil de filtre, vous pouvez contrôler la visibilité des fichiers et des dossiers. Si vous voulez voir uniquement les fichiers VHDL, cliquez sur  pour cacher les fichiers non HDL et sur  pour cacher les dossiers vides.

5.2 La vue « objet »



Cliquez sur l'onglet Object View afin d'obtenir la vue « objet ».

Cette vue vous affiche les paquets VHDL, entités, architectures, configuration et les modules Verilog présents dans la bibliothèque. Les noms attribués dans la bibliothèque sont déduits par HDL Companion depuis les déclarations dans la bibliothèque utilisateur des fichiers VHDL.



Vue « objet »

Dans ce tutoriel, nous avons une bibliothèque nommée *work* qui contient 2 entités et 3 configurations. Allons voir la bibliothèque *testbenches*.



Sélectionnez l'étiquette bleue *mico_test*. Puis en maintenant la touche Ctrl enfoncée, sélectionnez l'étiquette bleue *testbench*. Relâchez la touche Ctrl. Gardez le bouton de souris enfoncée et faites glisser les entités sélectionnées dans la librairie verte *testbenches*. Quand la souris passe sur un objet, l'ajout d'objet est rendu possible, la forme de la souris change. Relâchez la souris sur l'étiquette *testbenches*.

HDL Companion ré-analyse à présent votre projet et place les deux entités dans la bibliothèque *testbenches*. Remarquez que les 3 configurations sont aussi analysées dans la bibliothèque *testbenches*.

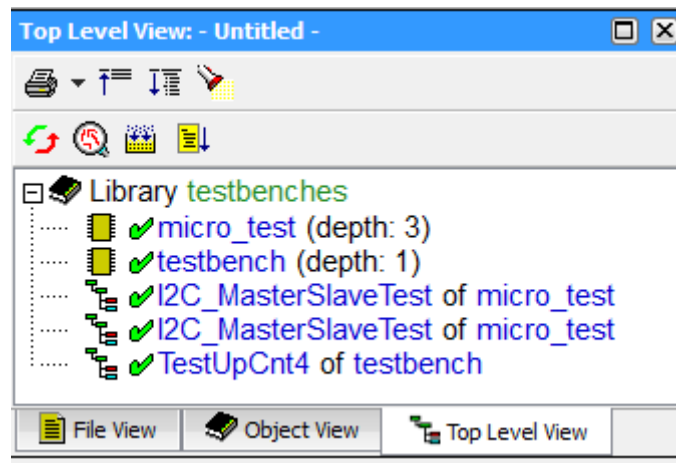
Vous pouvez créer une nouvelle bibliothèque en utilisant le raccourci du menu du projet – **Untitled-**.

Seule une bibliothèque vide peut-être supprimée. HDL Companion, effacera automatiquement les bibliothèques vides comme la bibliothèque *work*.

5.3 La « Top Level View »


 Cliquez sur l'onglet « Top Level View » pour obtenir la « Top Level View ».

Cette vue affiche toutes les entités et configurations VHDL et les modules Verilog qui ne sont pas instanciés dans une autre unité de design. Le niveau de hiérarchie est noté entre parenthèses.



La « Top Level View »


En regardant la « Top Level View, comme ci-dessus, nous voyons deux configurations avec le même nom : I2C_MasterSlaveTest. Jusqu'ici, nous avons ignoré les messages de la console.

 Faites défiler vers le haut dans la console jusqu'aux avertissements sur la redéfinition de la configuration de I2C_MasterSlaveTest. Double-cliquez sur les deux lignes précédées d'un carré jaune.

Les fichiers *micro_test_I2C_MasterSlaveTest.vhd* and *micro_test.vhd* sont ouverts dans la fenêtre source pour que les avertissements soient examinés. En regardant de plus près le code VHDL, nous pouvons remarquer que la configuration de I2C_MasterSlaveTest a été définie dans les deux fichiers.

 Clic droit dans la console puis sélectionnez Clear all, ainsi, vous allez réinitialiser la console.

Nous allons à présent fixer les propriétés du fichier *micro_test_I2C_MasterSlaveTest.vhd* à Unused (inutilisé).

 Sélectionnez l'onglet de vue des fichiers, clic droit sur le fichier *micro_test_I2C_MasterSlaveTest.vhd* et sélectionnez **propriétés**. Les propriétés du fichier s'affichent, sélectionnez l'option « Unused » puis fermez la boîte de dialogue en cliquant sur **OK**.


Le code HDL est alors re-analysé, cette fois, aucun avertissement n'apparaît dans la console étant donné qu'il n'y a plus qu'une seule définition de la configuration de I2C_MasterSlaveTest. Sur le symbole du fichier `micro_test_I2C_MasterSlaveTest.vhd` il y a une croix rouge pour indiquer son statut « inutilisé ».

Le modèle HDL Companion peut gérer les double définitions mais les analyseurs d'HDL remplaceront les fichiers existants par la version analysée la plus récente.

L'étape suivante est de définir l'élément « Top Level » du projet.



Clic droit sur l'étiquette de la configuration I2C_MasterSlaveTest et sélectionnez **Mark as TopLevel**.

Ce symbole va apparaître devant la configuration : 

6 La fenêtre détaillée

La fenêtre détaillée peut montrer différentes vues détaillées des unités analysées. Les unités doivent être syntaxiquement correctes afin d'afficher une vue détaillée. Chacun des points de vue (détaillée, hiérarchique et des pistes) est sélectionné en cliquant sur une étiquette bleue dans la fenêtre générale ou détaillée, ou en utilisant les objets du menu contextuel.

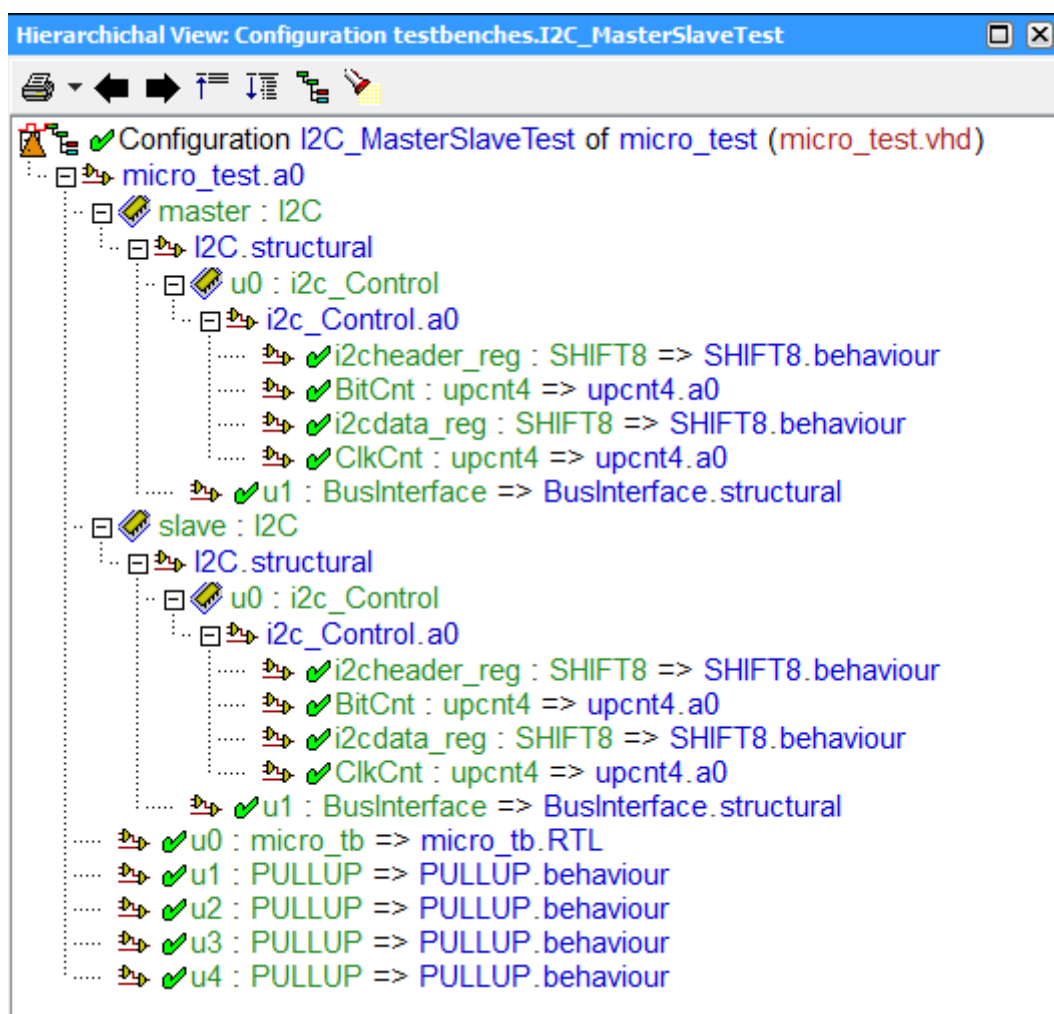
Les trois vues sont décrites dans les paragraphes suivants.

6.1 La vue hiérarchique




Cliquez sur **I2C MasterSlaveTest** dans la « Top Level View ».

La fenêtre détaillée vous affiche alors la hiérarchie complète de cette configuration.



Vue hiérarchique

La vue hiérarchique a 2 modes de visual. Soit en mode navigateur comme les vues de la fenêtre générale, soit le mode hiérarchique de bas en haut. Cette vue nécessite plus d'espace pour visualiser correctement mais offre une meilleure compréhension de la hiérarchie. Vous pouvez basculer d'un mode à l'autre avec l'outil  **Switch between graphical and tree view**.

Si vous cliquez sur une des étiquettes bleue, vous zoomerez sur la hiérarchie de l'objet correspondant. Cliquer une étiquette verte ouvrira le code VHDL dans la fenêtre source.

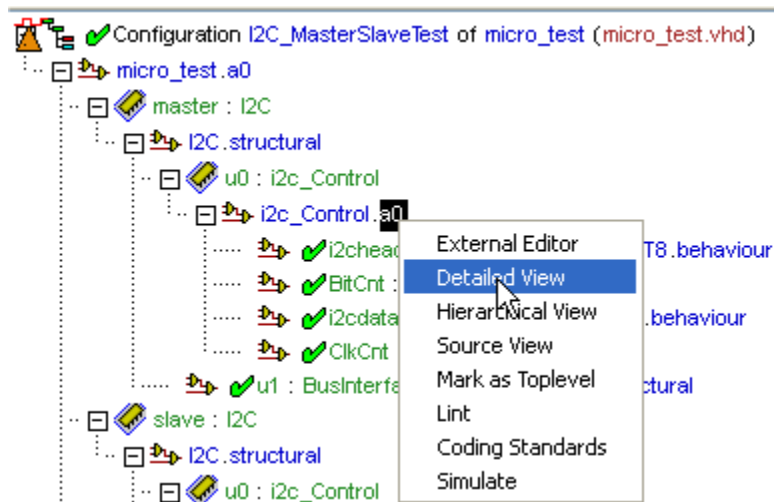
Vous pouvez utiliser les flèches **Précédent** et **Suivant** pour naviguer entre les différentes vues objet affichées dans la fenêtre détaillée.

6.2 La vue détaillée

Nous allons un peu plus explorer le code.

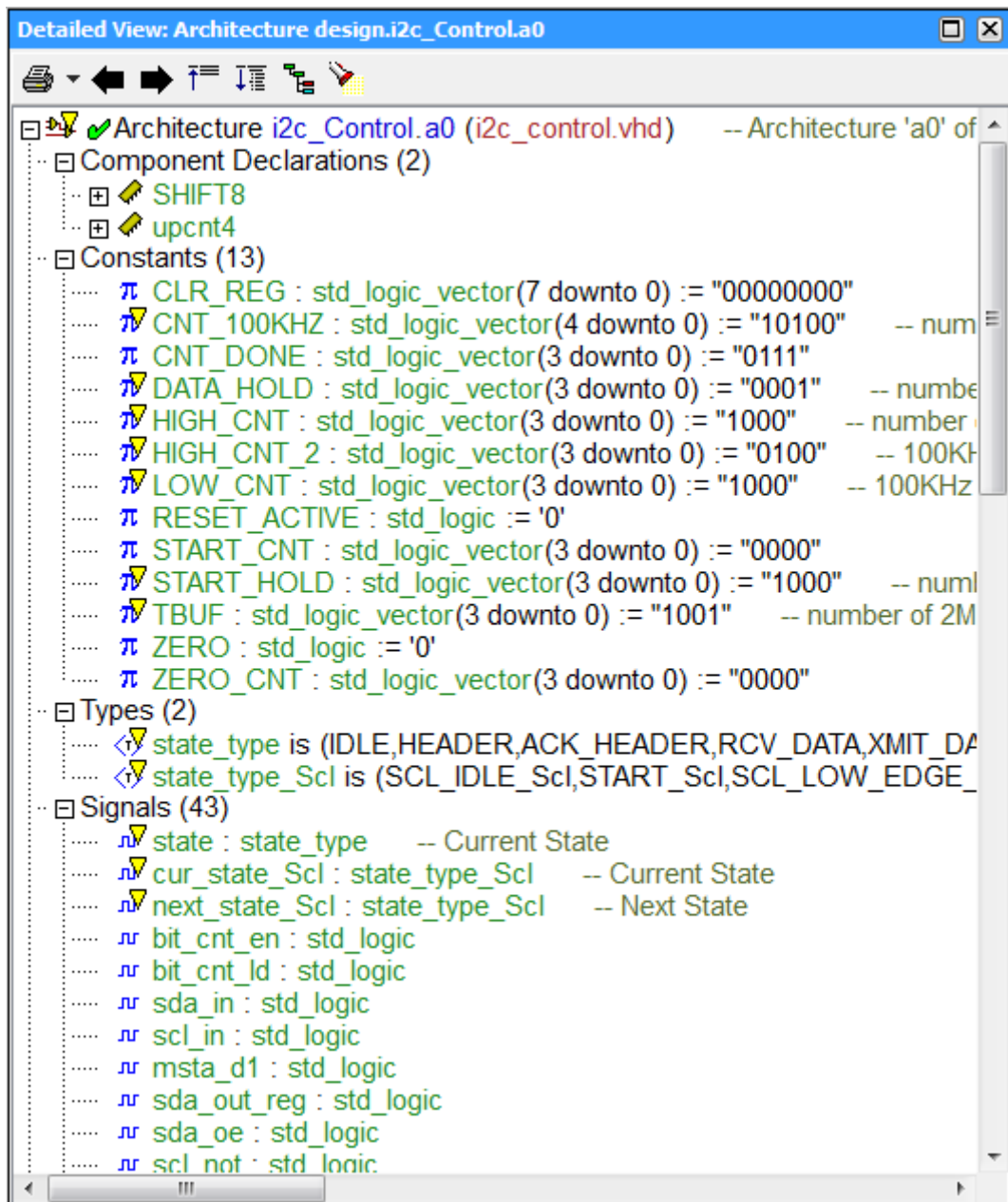


Clic droit sur l'étiquette **a0** de l'architecture **i2c_Control.a0** dans la vue hiérarchique de **I2C_MasterSlaveTest** et sélectionnez **Vue Détaillée**.



Sélectionnez la vue détaillée

A présent, la fenêtre détaillée affiche les données de l'architecture : les composants utilisés, déclarés et instanciés, les constantes, les types, les signaux et les process.



Vue détaillée

En cliquant sur une étiquette verte le fichier HDL s'ouvre dans la fenêtre source à la ligne appropriée.

En cliquant sur une étiquette bleue, vous ouvrirez l'objet correspondant dans la vue détaillée.



Cliquez sur l'étiquette de l'entité `i2c_Control` sur la ligne supérieure de l'architecture.

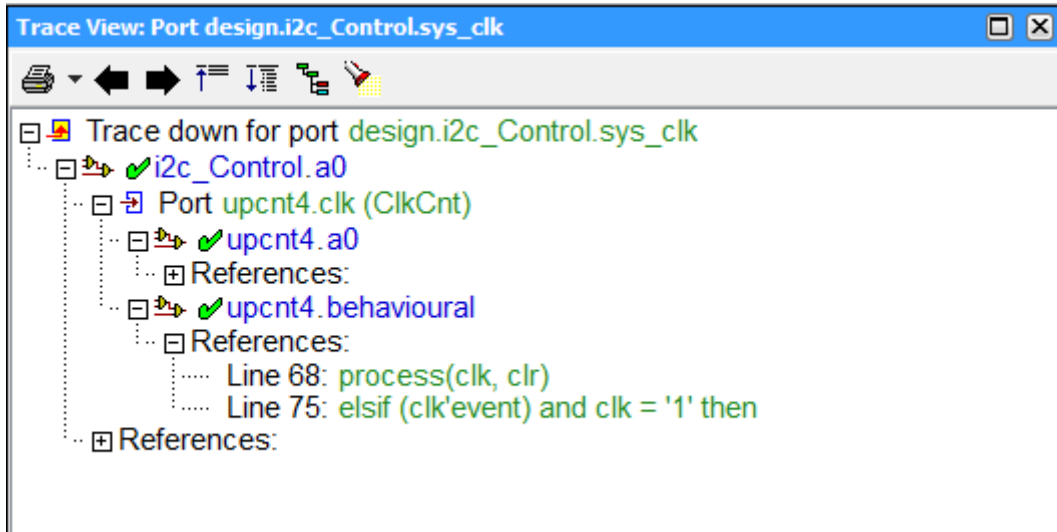
La vue détaillée a changé pour la vue entité. On peut y voir les architectures, les génériques, les ports et les instances d'usage de l'entité.

Remarquez que certains ports de symboles ont des icônes différentes puisqu'ils ont été reconnus comme étant des signaux d'horloge.

6.3 La vue des pistes

 Clic droit sur l'étiquette `sys_clk` et sélectionnez **Trace Down**.

Vous aurez donc une vue des pistes pour le signal à travers la hiérarchie de l'entité `i2c_Control`.



Vue des pistes

Les 'références' sont initialement fermées. Dans la figure ci-dessus, nous avons ouvert les références pour le `upcnt4.behavoudrial` ce qui nous montre les 2 lignes qui utilisent `clk`.

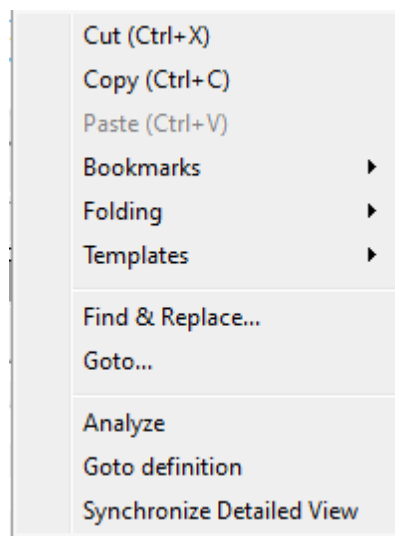
La vue des pistes est un moyen efficace pour suivre un signal à travers la hiérarchie et explorer les utilisations des signaux dans une architecture.

7 Trouver des objets et des textes

Il y a différentes manières de chercher ou trouver la définition d'un objet dans un projet HDL Companion. Lorsqu'un fichier HDL est correctement analysé, vous pouvez utiliser la fonction **Goto Definition** dans l'éditeur de script pour aller à la définition ou la déclaration d'un identifiant.

7.1 La fonction Goto Definition

Lorsqu'un fichier HDL est correctement analysé et ouvert dans l'éditeur de script, une info-bulle s'ouvre lorsque l'identifiant est reconnu. Pour voir la définition ou la déclaration, vous pouvez utiliser la fonction **Goto Definition** depuis le menu contextuel. Le script va alors ouvrir la définition et aller directement à la ligne appropriée. Cette fonction facilite l'accès à divers définitions.




Menu contextuel

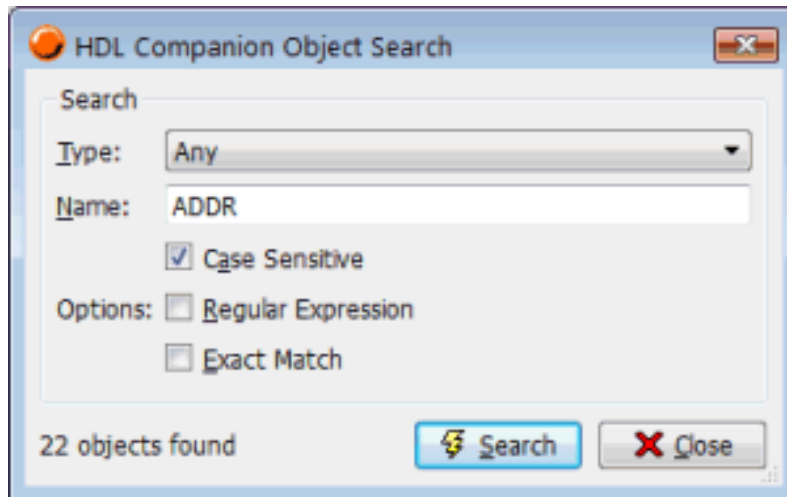
7.2 La recherche d'objet

Une manière appropriée de chercher des objets HDL est d'utiliser l'**outil de recherche d'objet**. Par exemple, pour trouver où et comment l'adresse de l'esclave I2C est définie :

 Cliquez sur l'outil de **localisation d'un objet** 

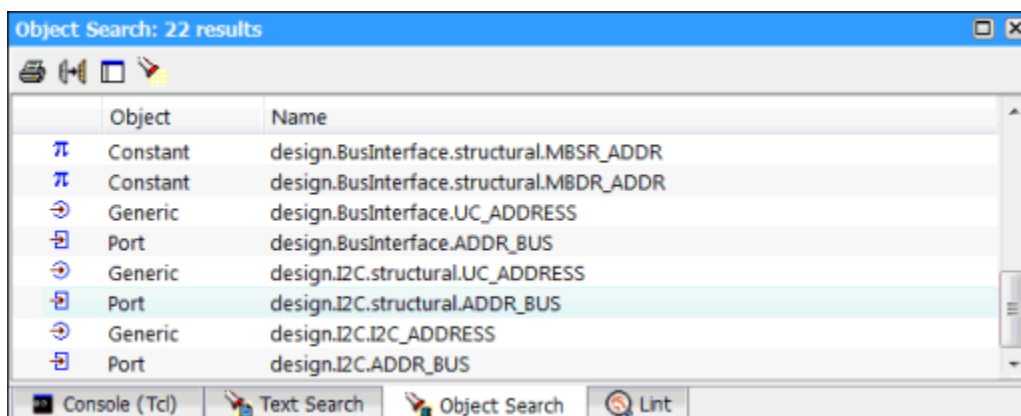
La fenêtre de recherche d'objet apparaît.

 Dans la fenêtre de recherche d'objet tapez « ADDR » au champ « nom ». Fixez les options comme sur la copie d'écran suivante puis cliquez sur **chercher**.



Recherche de « ADDR »

Les résultats de la recherche sont affichés dans l'onglet recherche d'objet de la console.



Résultats de la recherche

Les icônes et noms des objets nous informent du type d'objet correspond aux recherches. Le nom nous donne le nom hiérarchique dans l'unité. Double cliquez sur le résultat et le contexte dans lequel l'objet a été trouvé apparaîtra dans la fenêtre détaillée. Le fichier contenant la définition sera ouvert dans la fenêtre source à la ligne où l'objet est défini.

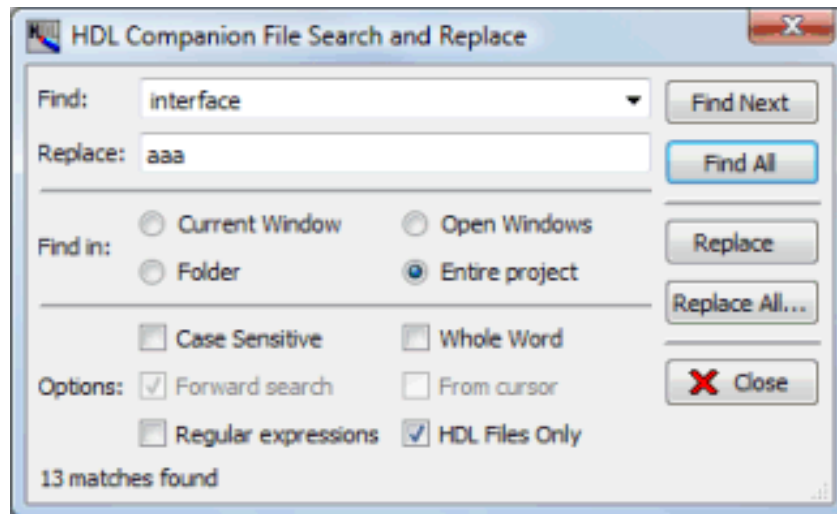
7.3 Recherche de fichier et remplacement.

Dans certains cas, la recherche d'un texte générique est plus propice à la production de résultats. L'outil de **recherche et remplacement de fichier** permet une recherche de texte générique.



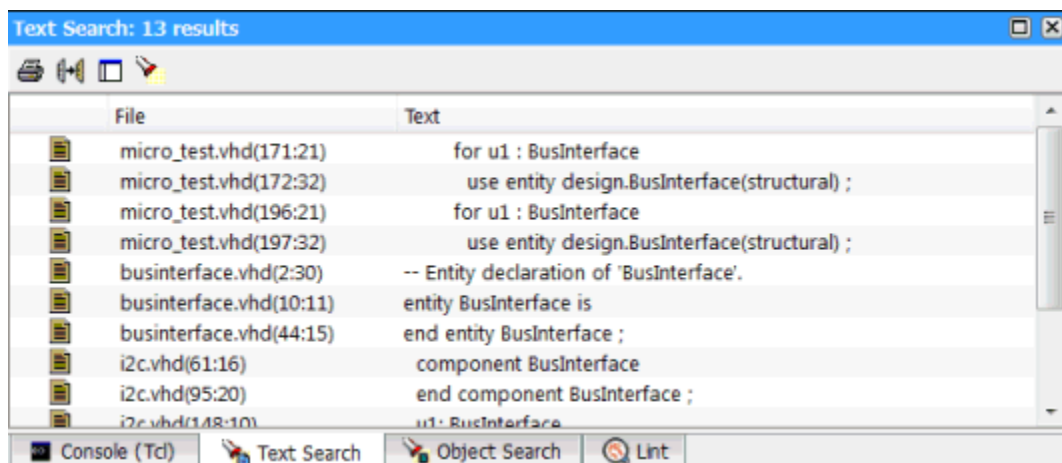
Cliquez sur  l'**outil de recherche et remplacement dans un fichier**. Dans l'outil qui s'ouvre tapez « interface » dans le champ 'trouver'. Sélectionnez le projet entier dans la liste

'trouver dans' et fixez les options comme sur la copie d'écran suivante. Fermez ensuite la fenêtre en cliquant sur « trouver tout ».



Recherche et remplacement d ' « interface »

Les résultats s'affichent dans l'onglet recherche de texte de la console.




Résultats

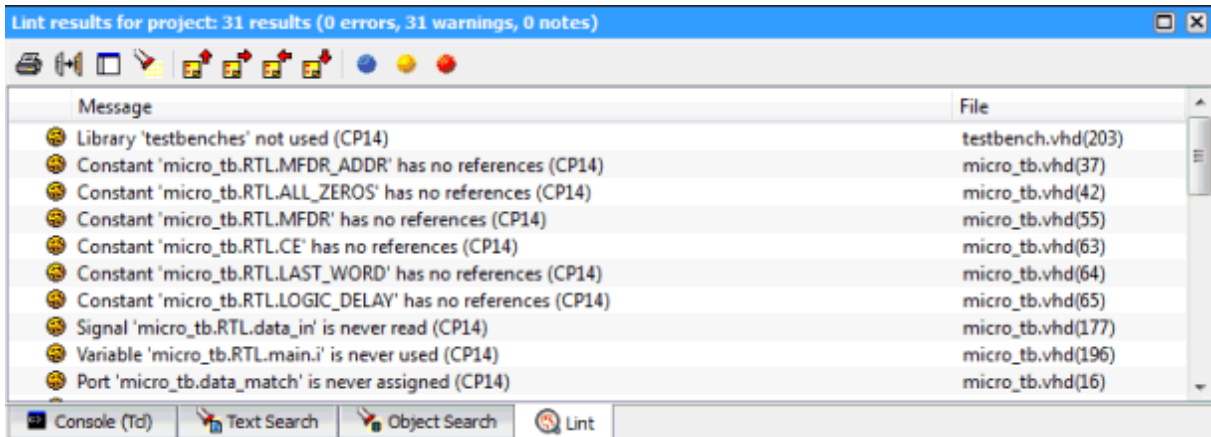
8 Le « Linting » d'un projet

Le Linting est une vérification formelle d'un certain nombre de règles. Il offre une méthode simple mais rapide de la vérification de l'exactitude d'un code HDL.



Cliquez sur le bouton de **Linting**  pour vérifier le respect des règles de codage HDL d'un projet complet.

Les résultats se trouvent dans l'onglet linting de la console.



Résultats

Les règles de linting peuvent être contrôlées dans les **propriétés du projet**.

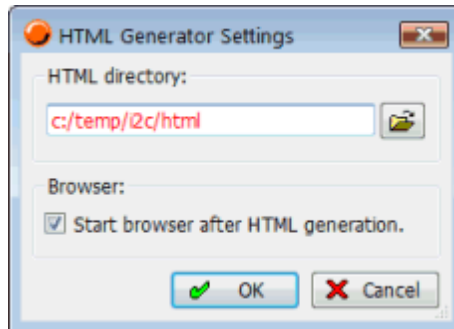
9 Générer un fichier HTML

Vous pouvez générer un fichier html de votre projet.



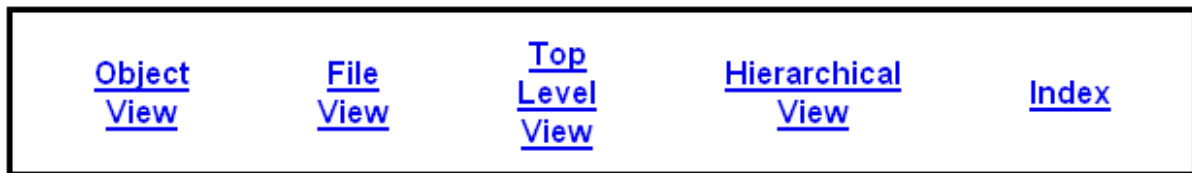
Cliquez sur l'outil de génération de fichier html

Une fenêtre apparaît elle vous permet de sélectionner le dossier dans lequel vous voulez que le fichier html soit généré.



Fenêtre de réglages

Lorsque l'option navigateur après génération d'un fichier html est autorisé, le navigateur par défaut se lancera lorsque le fichier HTML sera généré.



Object View: - Untitled -

- Project - Untitled -
 - 📁 Library std (2)
 - Packages (2)
 - 📁 [standard \(standard.vhd\)](#)
 - 📁 [TEXTIO \(textio.vhd\)](#)
 - 📁 [TEXTIO \(textio.vhd\)](#)
 - 📁 Library ieee (3)
 - Packages (3)
 - 📁 [std_logic_1164 \(std_1164.vhd\)](#)
 - 📁 [std_logic_1164 \(std_1164.vhd\)](#)
 - 📁 [std_logic_arith \(syn_arit.vhd\)](#)
 - 📁 [std_logic_arith \(syn_arit.vhd\)](#)
 - 📁 [std_logic_textio \(syn_textio.vhd\)](#)
 - 📁 [std_logic_textio \(syn_textio.vhd\)](#)
 - 📁 Library design (5)
 - Entities (5)
 - 📁 [BusInterface \(businterface.vhd\)](#)
 - 📁 [structural \(i2c_control.vhd\)](#)
 - 📁 [I2C \(i2c.vhd\)](#)
 - 📁 [structural \(i2c.vhd\)](#)

La vue objet du fichier HTML généré

Le fichier HTML est généré pour toutes les vues d'HDL Companion.

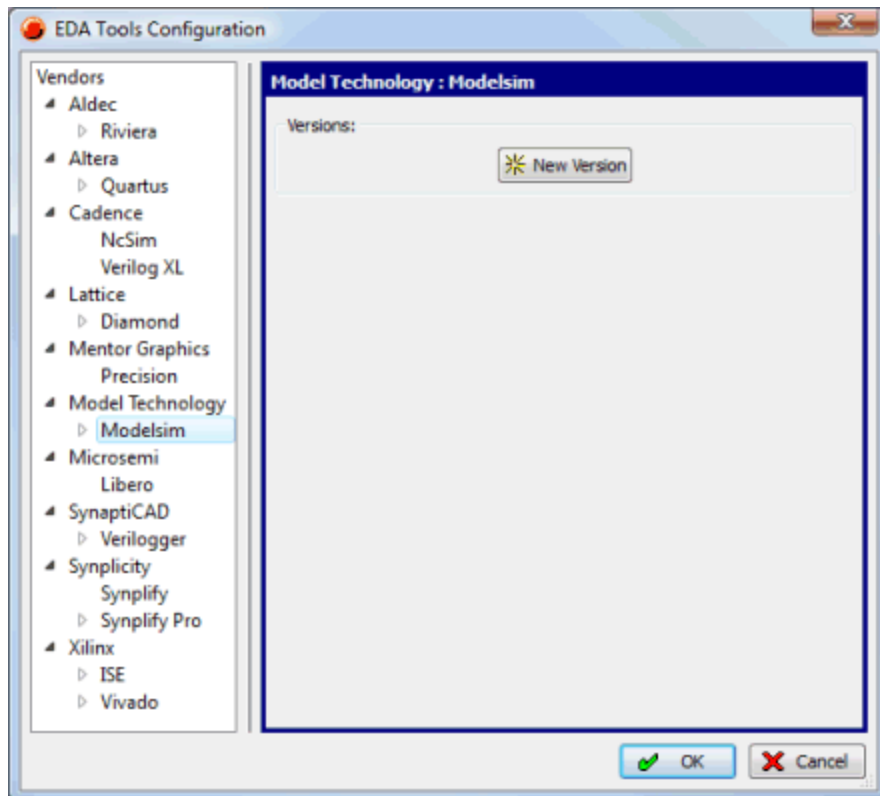
Vous pouvez naviguer dans les fichiers HTML de la même manière que dans HDL Companion.

10 Simulation

Dans les sections précédentes, HDL Companion était utilisé pour explorer les unités. Dans ce chapitre, nous utiliserons HDL Companion pour faire une simulation rapide. Étant donné que tous les fichiers ont été analysés, HDL Companion peut facilement déterminer la séquence correcte dans laquelle les fichiers doivent être compilés.



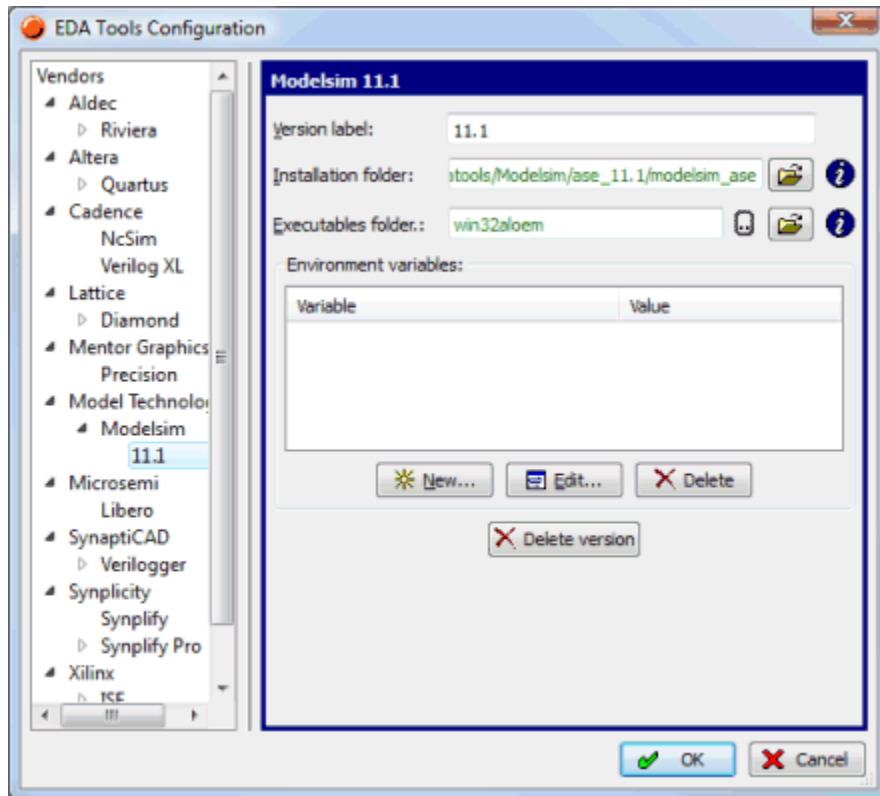
Cliquez sur l'outil de **configuration des designs électroniques** . Dans la fenêtre, sélectionnez la page Modelsim et cliquez sur **nouvelle version**.



Outil de configuration des designs électroniques, page modelsim




Dans la fenêtre, remplissez les champs **version** et **répertoire**.




Nouvelle version de Modelsim

Si vous utilisez une version OEM de Modelsim, changez le Bin dir. en répertoire win32 oem. Les champs répertoire et Bon dir. devraient apparaître vert, cela prouve qu'HDL Companion a reconnu les deux champs.

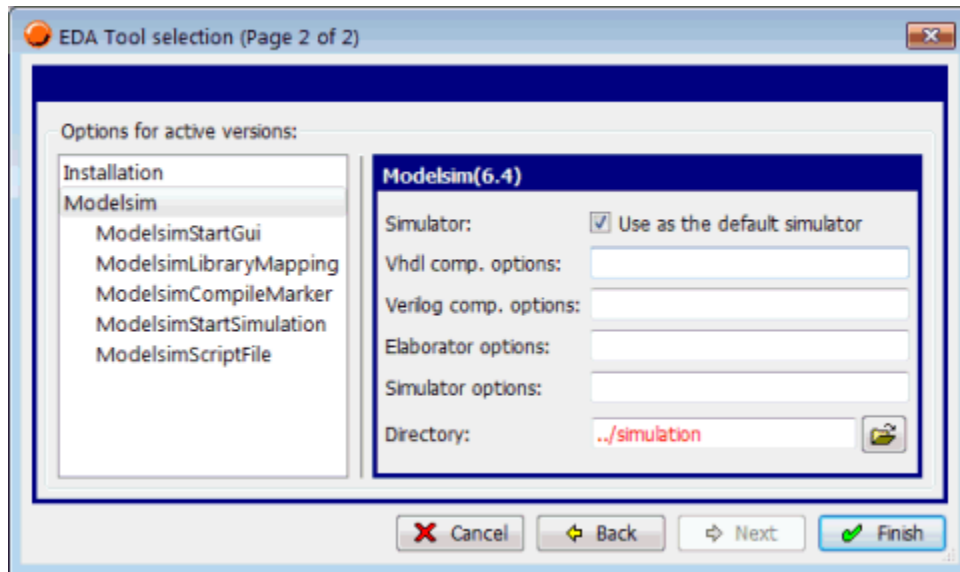
 Cliquez sur **OK**, dans la fenêtre suivante, cliquez sur **oui**.

L'assistant de sélection de l'outil de conception électronique apparaît.

 Sélectionnez la version de Modelsim que vous avez précédemment définie de la liste et cliquez sur **suivant**.

La seconde page de l'assistant s'ouvre.

 Sélectionnez la page **Modelsim**.

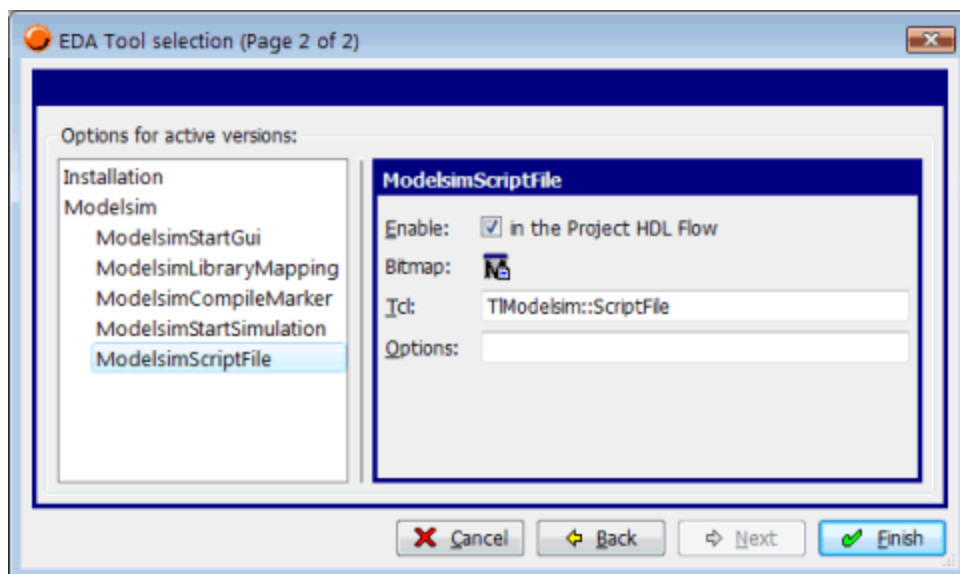


Réglages généraux de Modelsim

Sur cette page vous pouvez, par exemple, entrer des options par défaut d'un compilateur VHDL. Il est aussi recommandé de spécifier un répertoire de travail. Par défaut, le répertoire de simulation est relatif au fichier projet d'HDL Companion.



Sélectionnez la page ModelsimScriptFile.



Réglages individuels de Modelsim



Cliquez sur finir pour fermer l'assistant.

Cinq boutons ont été ajoutés à la barre d'outils.

Nous allons devoir compiler tous les fichiers.



Cliquez sur  l'**outil de compilation des fichiers HDL**

Les résultats apparaissent dans la console. Si tout a bien été compilé, vous pouvez débiter la simulation.



Cliquez sur  l'**outil de démarrage de simulation**.

Le simulateur Modelsim va démarrer et charger de design. Vous pouvez démarrer des cycles de fonctionnement en tapant '1 ms' dans le simulateur.



Fermez le simulateur Modelsim

11 Enregistrer et quitter

La dernière étape est d'enregistrer le projet HDL Companion pour pouvoir conserver les réglages que vous avez fait et les réutiliser une prochaine fois. Les réglages sont enregistrés au format xml. Aucune modification n'est effectuée sur les codes HDL. Seuls les fichiers et répertoires chargés dans la vue des fichiers sont conservés.



Cliquez sur  **enregistrer votre projet HDL Companion**. Naviguez jusqu'au répertoire voulu.

Par défaut, l'extension des projets HDL companion est .companion.

Ceci termine notre tutoriel HDL companion. . Nous espérons que vous avez apprécié la lecture de ce dernier.

Plus d'informations peuvent être trouvées dans le manuel de l'utilisateur **HDL Companion** et sur le site web: www.hdlworks.com.

12 Informations contact

EDA Expert

30 Villa Remond
94250 Gentilly
France

Téléphone: +33 (0) 1 49 84 86 26

Site web: www.eda-expert.com

Service clientèle: contact@eda-expert.com