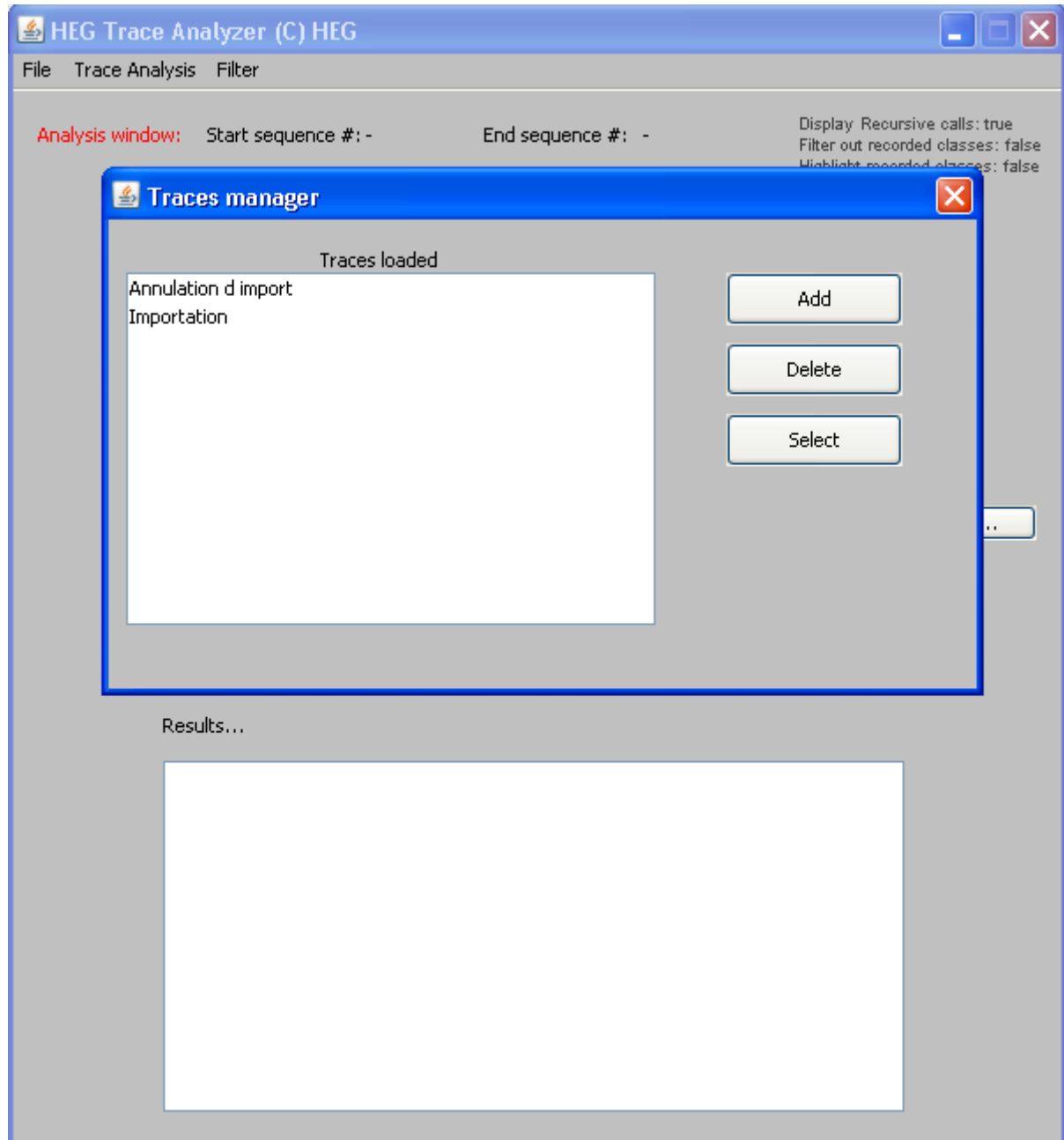


# Trace Analyzer : Manuel de l'utilisateur

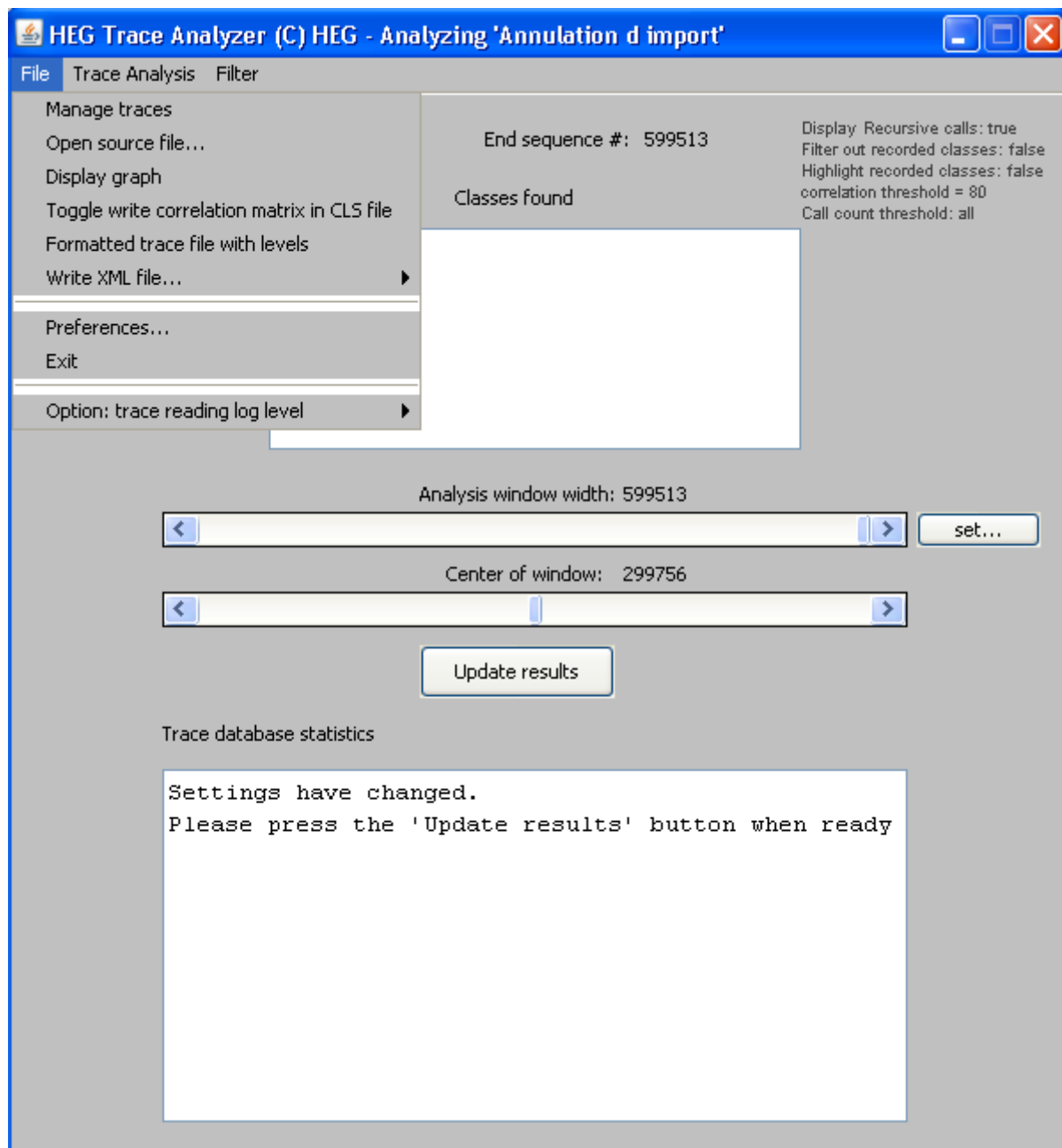
## 1 Au démarrage



La fenêtre « Traces manager » s'affiche au démarrage. Celle-ci permet de :

- Importer une nouvelle trace avec le bouton **Add**
- Sélectionner une ou plusieurs traces et la(les) supprimer avec le bouton **Delete**
- Sélectionner une ou plusieurs traces et travailler avec en appuyant sur le bouton **Select**

## 2 L'onglet « File »



## 2.1 Manage traces

Ouvrir la fenêtre « Traces manager », cf. partie « Au démarrage »

## 2.2 Open source file...

Sélectionner le code source à analyser.

## 2.3 Display graph

Afficher le graphique des éléments courants.

## ***2.4 Toggle write correlation matrix in CLS file***

Déclencher la génération d'une matrice de corrélation dans un fichier CLS si cette fonctionnalité n'a pas encore été activée. Si elle l'a déjà été, la matrice s'affiche dans l'écran du logiciel. Cette fonctionnalité nécessite l'enclenchement de la fonction "Trace Analysis/Temporal correlation matrix ».

## ***2.5 Formatted trace file with levels***

Générer un fichier texte contenant les appels de la trace et leur niveau respectif

## ***2.6 Write XML file***

Générer un fichier XML.

## ***2.7 Preferences***

Régler les préférences.

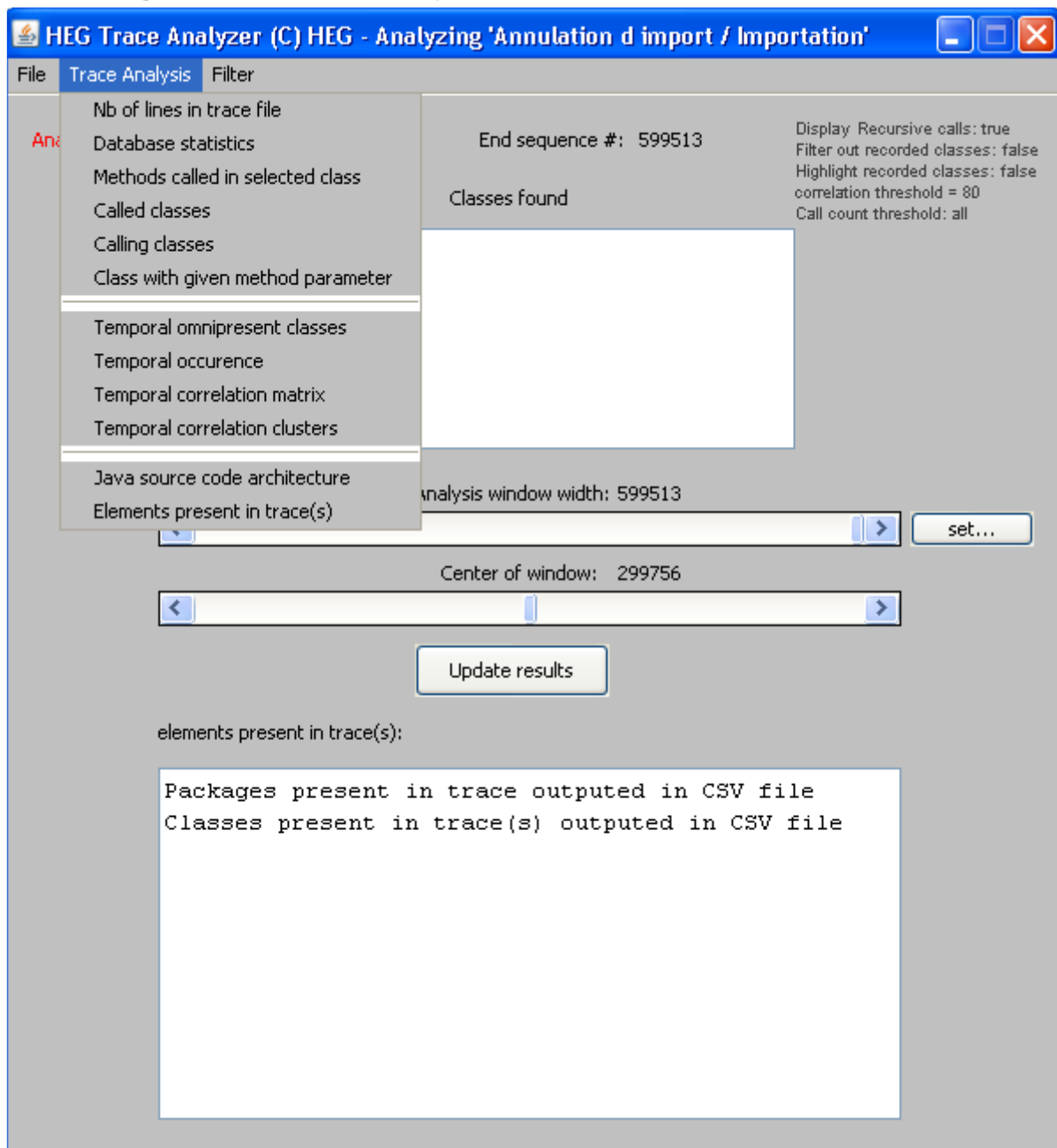
## ***2.8 Exit***

Sortir de l'application

## ***2.9 Option : trace reading log level***

Régler le contenu du log

### 3 L'onglet « Trace Analysis »



#### 3.1 Nb of lines in trace file

Afficher le nombre de lignes d'appel, de fin et de lignes totales d'une trace.

#### 3.2 Database statistics

Afficher les statistiques relatives à la trace, telles que le nombre d'appels, de fonctions, de procédures et de classes impliquées.

#### 3.3 Methods called in selected class

Afficher les méthodes appelées des classes sélectionnées, dans la fenêtre d'affichage.

### **3.4 Called classes**

Afficher les classes appelées par les classes sélectionnées, dans la fenêtre d’affichage.

### **3.5 Calling classes**

Afficher les classes qui appellent les classes sélectionnées, dans la fenêtre d’affichage.

### **3.6 Class with given method parameter**

Rechercher les classes qui ont comme valeur de paramètre le mot clé saisi.

### **3.7 Temporal omnipresent classes**

Permet la segmentation de la trace en un nombre de morceaux paramétrable grâce au contrôle se situant sous le label « Sampling window width ». En déplaçant le curseur à gauche on augmente le nombre de segments et inversement. Les classes omniprésentes sont affichées dans l’écran d’affichage du haut et le nombre de segment est inscrit dans celui du bas.

### **3.8 Temporal occurrence**

Comparer l’occurrence des modules sélectionnés dans l’écran d’affichage supérieur en réglant le nombre de segments.

### **3.9 Temporal correlation matrix**

Afficher dans l’écran inférieur une matrice représentant la corrélation des classes sélectionnées dans l’écran supérieur. La taille de fenêtrage est également paramétrable pour cette fonctionnalité. La fonctionnalité 2.3 (Display graph) est utilisable ici.

### **3.10 Temporal correlation clusters**

Afficher dans l’écran inférieur une matrice représentant les clusters de corrélation des classes sélectionnées dans l’écran supérieur. La taille de fenêtrage ainsi que le taux de corrélation sont paramétrables. Si le code source a été sélectionné, un fichier csv est généré. Celui-ci représente les clusters sur l’axe vertical et les packages sur l’axe horizontal. Un 0 dans une case liant à package à un cluster signifie que ce package n’est pas présent dans le cluster en question. Un 1 signifie l’inverse.

### **3.11 Java source code architecture**

Générer un fichier csv contenant les packages du code source sélectionné. Afin d’effectuer cette analyse il est nécessaire d’avoir auparavant sélectionné le code source.

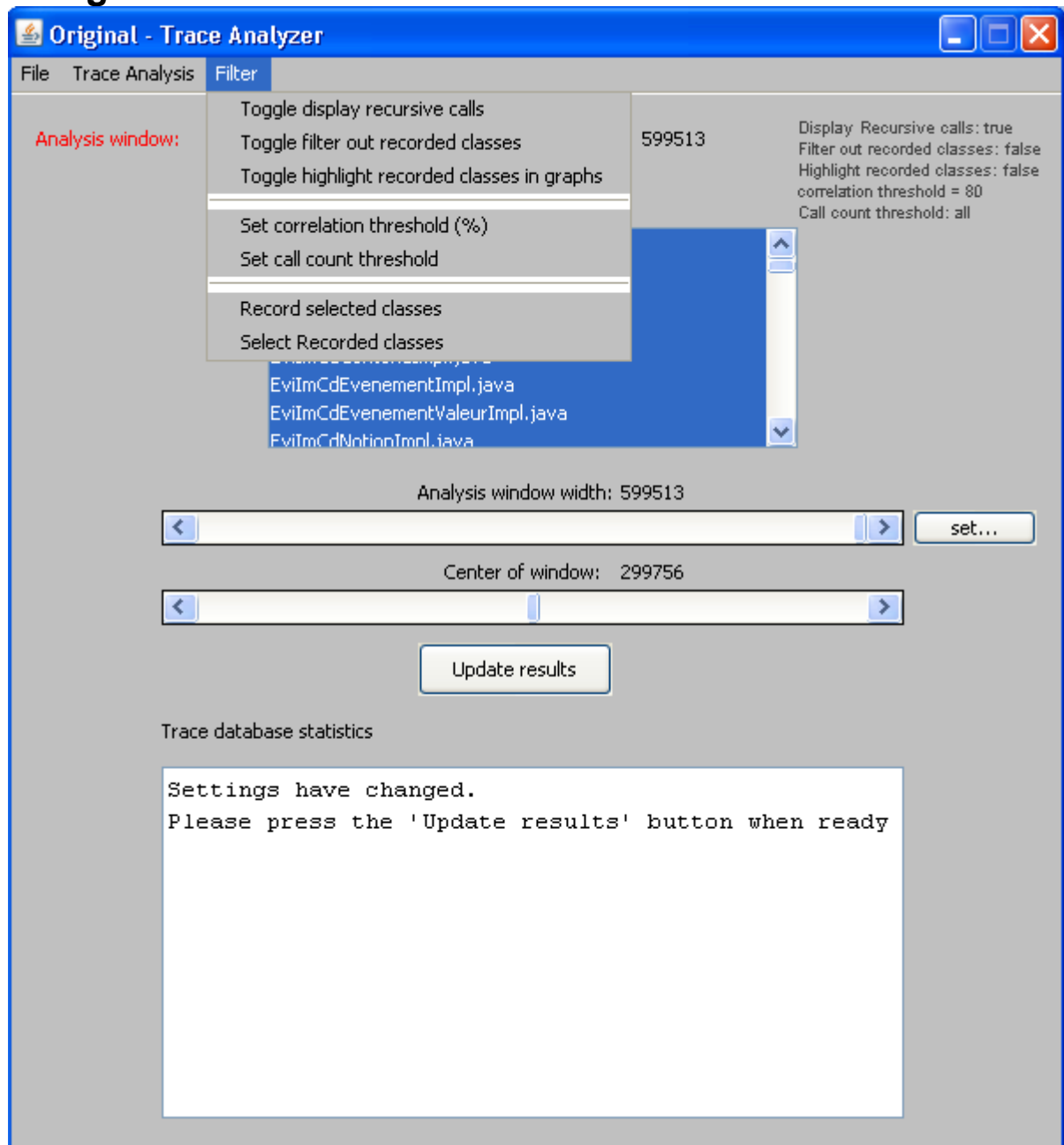
### **3.12 Elements present in trace(s)**

Générer 2 fichiers csv, le premier représente les packages présent dans la ou les traces analysée(s). Pour chaque package le nombre inscrit représente le nombre de trace dans lesquels il est présent. Afin de connaître de quel package il s’agit, il faut auparavant tirer l’architecture java avec la fonction « Java source code architecture ».

Le second fichier généré contient, sur l'axe horizontal, le nom des traces analysées et, sur l'axe vertical, le nom des classes du code source. Un 1 dans une case signifie que la classe correspondante est présente dans la trace en question, alors qu'un 0 signifie qu'elle ne l'est pas.

Afin d'effectuer cette analyse il est nécessaire d'avoir auparavant sélectionné le code source.

## 4 L'onglet « Filter »



### 4.1 Toggle display recursive calls

Bouton bascule permettant de désactiver ou pas les appels récurifs. L'état de cette propriété est visible dans le coin supérieur droit.

#### ***4.2 Toggle filter out recorded classes***

Bouton bascule permettant de filtrer ou pas les classes enregistrées. L'état de cette propriété est visible dans le coin supérieur droit.

#### ***4.3 Toggle highlight recorded classes in graphs***

Bouton bascule permettant de mettre en évidence ou pas les classes enregistrées dans les graphes générés. L'état de cette propriété est visible dans le coin supérieur droit.

#### ***4.4 Set correlation threshold (%)***

Paramétrer le seuil de corrélation des classes.

#### ***4.5 Set call count threshold***

Paramétrer les types de liens qui seront présents dans les graphes générés.

#### ***4.6 Record selected classes***

Enregistre les classes sélectionnées.

#### ***4.7 Select recorded classes***

Sélectionne les classes enregistrées.

