

# Comment debugger Lasso ?

Dans la suite le programme qui manifeste le bug sera nommé test et le répertoire racine des sources de lasso \$LASSO\_SRC\_HOME.

On peut déjà dans un premier temps activer les options de debug du compilateur:

```
./configure --enable-debugging --enable-maintainer-mode
```

Ensuite utiliser les options de debug de lasso pour repérer les désallocations:

```
LASSO_FLAG=memory-debug ./test
```

Si on pense que c'est un problème de signature, on utilisera ce flag:

```
LASSO_FLAG=no-verify-signature ./test
```

Les flags sont cumulables séparés par un espace, une virgule ou un deux-points.

Pour débogger les messages logués via la librairie GLib (WARNING ou CRITICAL), le plus simple est d'activer les exceptions en cas de message et d'utiliser gdb:

```
G_DEBUG=fatal_warnings gdb --args ./test  
G_DEBUG=fatal_criticals gdb --args ./test
```

Pour les écritures ou lecture depuis des pointeurs invalides, une seule solution: valgrind, mais il faut bien utiliser les fichiers de suppressions surtout avec les scripts python. Sinon gare à l'overdose de faux positifs:

```
valgrind --suppressions=$LASSO_SRC_HOME/tests/valgrind/openssl.supp --suppressions=/usr/lib/valgrind/python.supp python ./test.py
```

Pour obtenir une demande d'ouverture de gdb pour chaque erreur détectée on utilisera l'option --db-attach=yes.

Un outil bien pratique pour détecter les leaks de GObject est refdbg que l'on trouvera à cette adresse <http://refdbg.sourceforge.net/>. Il nécessite de recompiler la glib avec une option ./configure --disable-visibility. Son utilisation de base est la suivante:

```
refdbg -c 'addrule <LassoNode> D:ALL' ./test
```

Cette commande affichera tous les évènements d'allocation et comptage de référence (NEW/REF/UNREF/FINALIZE/etc..) pour tous les objets dont la classe est ou descend de LassoNode. On peut s'en servir avec gdb:

```
refdbg -c 'addrule <LassoNode> B:Finalize' ./test
```

Cette commande fera s'arrêter gdb sur toute Finalization d'un objet dont la classe est ou descend de LassoNode.