Livrets B4x





B4x E D I

**E**nvironnement de **D**éveloppement **I**ntégré

Copyright: © 2017 Anywhere Software Édition 1.1

Dernière mise à jour : 2017.10.12

[1 B4x 5](#_Toc495585523)

[2 EDI en général 6](#_Toc495585524)

[3 Menu et barre d’outils 7](#_Toc495585525)

[3.1 Barre d’outils 7](#_Toc495585526)

[3.2 Menu Fichier 8](#_Toc495585527)

[3.3 Menu Edition 8](#_Toc495585528)

[3.4 Menu Projet 9](#_Toc495585529)

[3.4.1 Ajouter un nouveau module 10](#_Toc495585530)

[3.5 Menu outils 11](#_Toc495585531)

[3.5.1 Options de l’éditeur EDI 12](#_Toc495585532)

[3.5.1.1 Langue 13](#_Toc495585533)

[3.5.1.2 Thèmes 14](#_Toc495585534)

[3.5.1.3 Sélecteur de police 15](#_Toc495585535)

[3.5.1.3.1 Saut à la ligne 15](#_Toc495585536)

[3.5.1.4 Configurer le Timeout 15](#_Toc495585537)

[3.5.1.5 Désactiver Complétion automatique implicite 15](#_Toc495585538)

[3.5.2 Capture d’écran seulement B4A 16](#_Toc495585539)

[3.5.3 Capture de vidéo seulement B4A 17](#_Toc495585540)

[3.5.4 Nettoyer le dossier de fichiers (fichiers inutiles) 18](#_Toc495585541)

[3.5.5 Nettoyer le projet 18](#_Toc495585542)

[3.6 Menu clic droit 19](#_Toc495585543)

[3.7 Mode de compilation 20](#_Toc495585544)

[3.7.1 B4A et B4J 21](#_Toc495585545)

[3.7.1.1 Modes Release et (obfuscated) dans B4A et B4J 21](#_Toc495585546)

[3.7.2 B4i 22](#_Toc495585547)

[3.7.3 B4R 22](#_Toc495585548)

[4 Zone code 23](#_Toc495585549)

[4.1 Diviser la zone code 23](#_Toc495585550)

[4.2 Taille du texte de l’EDI 24](#_Toc495585551)

[4.3 Entêtes de code Project Attributes / Activity Attributes 25](#_Toc495585552)

[4.3.1 B4A 25](#_Toc495585553)

[4.3.1.1 Project Attributes attributs projet 25](#_Toc495585554)

[4.3.1.2 Activity Attributes attributs Activity 25](#_Toc495585555)

[4.3.1.3 Service Attributes attributs service 26](#_Toc495585556)

[4.3.2 B4i 27](#_Toc495585557)

[4.3.3 B4J 27](#_Toc495585558)

[4.3.4 B4R 27](#_Toc495585559)

[4.4 Annuler – Refaire   28](#_Toc495585560)

[4.5 Réduire une routine 28](#_Toc495585561)

[4.6 Réduire une Région 29](#_Toc495585562)

[4.7 Réduire tout le code en entier 30](#_Toc495585563)

[4.8 Basculer les contours Ctrl + 0 31](#_Toc495585564)

[4.9 Copier un texte sélectionné 32](#_Toc495585565)

[4.10 Déplacer ligne(s) Haut / Bas Alt + Up / Alt + Down 32](#_Toc495585566)

[4.11 Rechercher / Remplacer 33](#_Toc495585567)

[4.12 Commenter et décommenter du code   34](#_Toc495585568)

[4.13 Signets  35](#_Toc495585569)

[4.14 Indentation   36](#_Toc495585570)

[4.15 Format automatique 38](#_Toc495585571)

[4.16 Documentation en passant au-dessus d’éléments de code 39](#_Toc495585572)

[4.17 Complétion automatique 40](#_Toc495585573)

[4.18 Documentation intégrée 44](#_Toc495585574)

[4.18.1 Copie d’exemples de code 45](#_Toc495585575)

[4.19 Sauter vers une routine 46](#_Toc495585576)

[4.20 Mettre en évidence des occurrences de mots 47](#_Toc495585577)

[4.21 Points d’arrêt 48](#_Toc495585578)

[4.22 Sélecteur de couleur  50](#_Toc495585579)

[4.23 Sélecteur d’icônes  51](#_Toc495585580)

[4.24 Couleurs dans la marge gauche 52](#_Toc495585581)

[4.25 URLs dans commentaires Ctrl + clic 53](#_Toc495585582)

[5 Onglets 54](#_Toc495585583)

[5.1 Fenêtres onglets flottantes 55](#_Toc495585584)

[5.2 Flottante  56](#_Toc495585585)

[5.3 Cacher automatiquement  59](#_Toc495585586)

[5.4 Fermer 61](#_Toc495585587)

[5.5 Liste de Modules et routine  62](#_Toc495585588)

[5.5.1 Trouver la procédure / module / numéro de ligne (Ctrl + E) 63](#_Toc495585589)

[5.6 Gestionnaire de fichiers  B4A, B4i et B4J 64](#_Toc495585590)

[5.7 Logs  66](#_Toc495585591)

[5.7.1 Avertissements de compilation 67](#_Toc495585592)

[5.7.1.1 Ignorer des avertissements de compilation 68](#_Toc495585593)

[5.7.1.2 Liste des avertissements 69](#_Toc495585594)

[5.8 Gestionnaire de bibliothèques  77](#_Toc495585595)

[5.9 Recherche rapide  78](#_Toc495585596)

[5.10 Trouver toutes les références (F7)  80](#_Toc495585597)

[6 Navigation dans l’EDI 81](#_Toc495585598)

[6.1 Alt + Gauche / Alt + Droite 81](#_Toc495585599)

[6.2 Alt + N Menu de la pile de Navigation  81](#_Toc495585600)

[6.3 Diviser l’écran 81](#_Toc495585601)

[6.4 Fenêtres multiples 82](#_Toc495585602)

[6.5 Ctrl + E Recherche de routines ou modules 82](#_Toc495585603)

[6.6 Ctrl + Clic sur une routine ou variable 82](#_Toc495585604)

[6.7 F7 - Trouver toutes les références 82](#_Toc495585605)

[6.8 Ctrl + F Recherche Rapide 82](#_Toc495585606)

[7 Débogage B4A, B4i, B4J 83](#_Toc495585607)

[7.1 B4A, B4i, B4J 83](#_Toc495585608)

[7.1.1 Mode de compilation Debug 84](#_Toc495585609)

[7.1.1.1 Barre d’outils de débogage 84](#_Toc495585610)

[7.1.1.1.1 Exécute  F5 84](#_Toc495585611)

[7.1.1.1.2 Entrer  F8 85](#_Toc495585612)

[7.1.1.1.3 Sauter  F9 86](#_Toc495585613)

[7.1.1.1.4 Sortir  F10 86](#_Toc495585614)

[7.1.1.1.5 Stop  87](#_Toc495585615)

[7.1.1.1.6 Redémarrer  F11 87](#_Toc495585616)

[7.1.2 Fenêtre Débogueur 88](#_Toc495585617)

[7.1.2.1 Le bouton d’état 88](#_Toc495585618)

[7.1.2.2 La fenêtre points d’arrêt 88](#_Toc495585619)

[7.1.2.3 La fenêtre Regard 89](#_Toc495585620)

[7.1.2.4 La fenêtre objets 90](#_Toc495585621)

[7.1.2.5 Points d’arrêt 91](#_Toc495585622)

[7.1.2.6 Logs 93](#_Toc495585623)

[7.1.2.7 Modification de code dans le Débogueur 94](#_Toc495585624)

[7.1.3 Debug (legacy) mode B4A seulement 95](#_Toc495585625)

[7.2 Débogage B4R 96](#_Toc495585626)

[7.2.1 Débogage avec le projet FeuxSignalisation 96](#_Toc495585627)

Contributeurs principaux : Klaus Christl (klaus), Erel Uziel (Erel)

Tranduit par : Klaus Christl (klaus)

**Pour chercher un mot ou une phrase particulière veuillez utiliser la fonction Rechercher dans le menu Edition.**

Les codes source avec tous les fichiers nécessaires (layouts, images etc.) de tous les projets exemple dans ce guide figurant dans le dossier CodeSource.

Mis à jour pour les versions ci-dessous :

B4A version 7.30

B4i version 4.30

B4J version 5.90

B4R version 2.20

**Livrets B4x en français :**

B4x Premier pas

B4x Langage Basic

B4x EDI Environnement de Développement Integré

**Livrets B4x (Booklets) en anglais :**

[B4x Getting started](http://www.b4x.com/android/files/Booklets.zip)

[B4x Basic Language](http://www.b4x.com/android/files/Booklets.zip)

[B4x IDE Integrated Development Environment](http://www.b4x.com/android/files/Booklets.zip)

[B4x CustomViews](http://www.b4x.com/android/files/B4X_CustomViews.zip)

# B4x

B4x est une suite de langages de programmation BASIC pour différentes plateformes.

B4x supporte plus de plateformes que n’importe quel autre outil

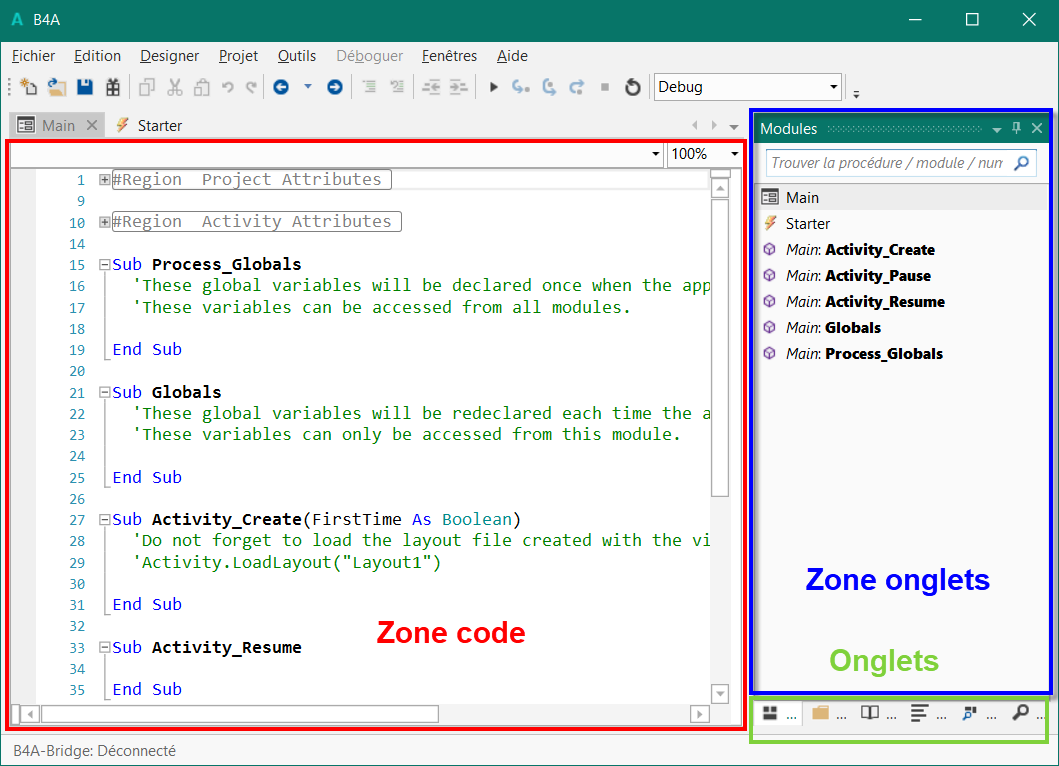
ANDROID | IOS | WINDOWS | MAC | LINUX | ARDUINO | RASPBERRY PI | ESP8266 | ET PLUS...

* **B4A**  **Android**  
  B4A inclut toutes les fonctionnalités pour développer rapidement n’importe quel type d’application pour Android.
* **B4i**  **iOS**  
  B4i est un outil de développement pour des applications natives pour iOS.  
  B4i suit les mêmes concepts que B4A, et vous permet de réutiliser la plupart du code et produire des applications pour les deux systèmes d’exploitation Android et iOS.
* **B4J**  **Java / Windows / Mac / Linux / Raspberry PI**  
  B4J est un outil de développement, **100% gratuit,** pour des applications desktop, serveurs et l’Internet des Objets (en anglais Internet of Things IoT).  
  Avec B4J vous pouvez facilement créer des applications desktop avec interface utilisateur, des applications de console (sans interface utilisateur) et des solutions serveur.  
  Les applications compilées peuvent fonctionner sur Windows, Mac, Linux et des cartes ARM (comme Raspberry Pi).
* **B4R**  **Arduino / ESP8266**B4R est un outil de développement, **100% gratuit,** pour des applications natives pour des microcontrôleurs Arduino et ESP8266.  
  B4R suit les mêmes concepts que les autres outils B4x, fournissant un outil de développement simple et puissant.  
  B4R, B4A, B4J et B4i ensemble, constituent la meilleure solution de développement pour l’Internet des Objets (en anglais Internet of Things IoT).

# EDI en général

L’**E**nvironnement de **D**éveloppement **I**ntégré.

Lorsque vous lancez l’EDI vous aurez une image similaire à celle-ci-dessous.



Toutes les images correspondent à l’EDI de B4A.

Les EDI des autres produits B4x ont des apparences similaires avec des thèmes différents.

Des images spécifiques aux autres produits seront montrées si nécessaire.

Vous avez 3 zones principales :

* Zone Code L’éditeur de code.
* Zone Onglets Le contenu de cette zone dépend de l’onglet sélectionné.
* [Onglets](#_Tabs) Onglets pour différents paramètres.

# Menu et barre d’outils





## Barre d’outils



Génère un nouveau projet vide [Ctrl + N].

 Charge un projet.

 Enregistre le projet courant [Ctrl + S].

 Exporte le projet dans un fichier zip.

 Copie le texte sélectionné dans le presse-papiers [Ctrl + C].

 Coupe le texte sélectionné et le copie dans le presse-papiers [Ctrl + X].

 Colle le texte du presse-papiers à la position du curseur [Ctrl + V].

 Annule la dernière opération [Ctrl + Z].

 Refait l’opération précédente [Ctrl + Shift + Z].

 Navigue en arrière [Alt + Gauche].

 Historique de navigation [Alt + N].

 Navigue en avant [Alt + Droite].

[Commente un bloc [Ctrl + Q]](#_Commenting_and_uncommenting).

 [Décommente un bloc [Ctrl + W]](#_Commenting_and_uncommenting).

 [Décale l’indentation du texte sélectionné](#_Indentation) vers la gauche.

 [Décale l’indentation du texte sélectionné vers la droite](#_Indentation).

Lance le compilateur [F5].

Les 5 fonctions ci-dessous ne sont actives que lorsque le débogage est actif

 Entrer [F8].

Sauter [F9].

 Sortir [F10]. .

 Stop

 Redémarrer [F11].

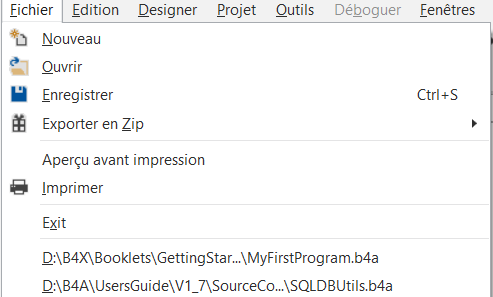


Liste des [options du compilateur](#_Compiler_mode) et [Débogage](#_Debugging).



Options de compilation conditionnelle.

## Menu Fichier

**Nouveau** Génère un nouveau projet vide.

**Ouvrir** Charge un projet.

**Enregistrer** Enregistre le projet courant.

**Exporter en Zip** Exporte le projet complet dans un fichier zip.

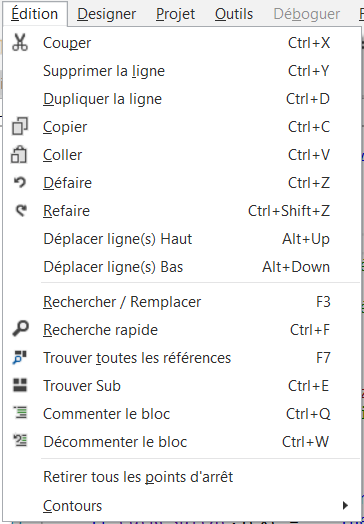
**Aperçu avant impression**.

**Imprimer** Imprimer tout le code du module sélectionné.

**Exit** Quitte l’EDI.

Liste de projets récents.

## Menu Edition

**Couper** Supprime le texte sélectionné et le copie dans le presse-papiers.

**Supprimer ligne** Supprime la ligne à la position du curseur.

**Dupliquer la ligne** Duplique la ligne courante.

**Copier** Copie le texte sélectionné dans le presse-papiers.

**Coller** Colle le texte du presse-papiers à la position du curseur.

**Défaire** Défait la dernière opération.

**Refaire** Refait l’opération précédente.

**Déplacer ligne(s) Haut** Déplace les lignes sélectionnées vers le haut.

**Déplacer ligne(s) Bas** Déplace les lignes sélectionnées vers le bas.

[Rechercher et Remplace](#_Find_/_Replace)r.

[Recherche rapide](#_Quick_Search)

[Trouver toutes les références](#_Find_All_References)

[Trouver Sub](#_Find_Sub_/)

**Commenter un bloc  
Décommenter un bloc**

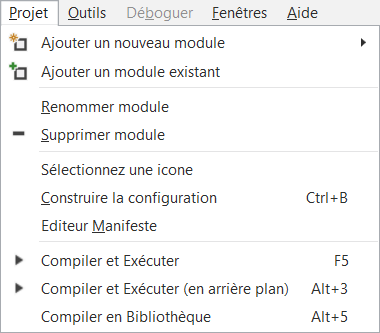
[Commenter / Décommenter les lignes](#_Commenting_and_uncommenting) sélectionnées.

**Retirer tous les** [Points d’arrêt](#_Breakpoints).

**Contours** [Collapse tout le code](#_Collapse_the_whole).

## Menu Projet

**B4A**



Ajoute un nouveau [module](#_Modules)

Ajoute un [module](#_Modules) existant

Renomme le [module](#_Modules)

Supprime le [module](#_Modules) courant

Sélection d’une icône pour le programme.

Construit la configuration.

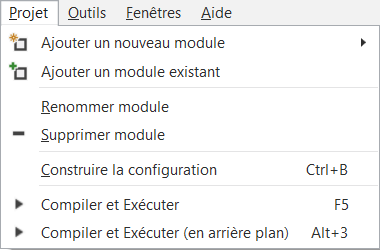
Editeur de manifeste.

Compile et exécute le projet.

Compile et exécute le projet en arrière-plan.

Compile en une bibliothèque.

**B4i, B4R**



Ajoute un nouveau [module](#_Modules)

Ajoute un [module](#_Modules) existant

Renomme le [module](#_Modules)

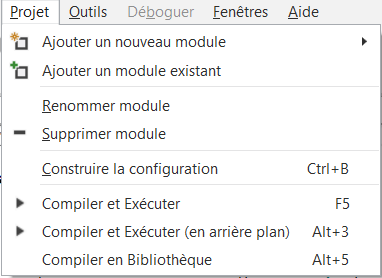
Supprime le [module](#_Modules) courant

Construit la configuration.

Compile et exécute le projet.

Compile et exécute le projet en arrière-plan.

**B4J**



Ajoute un nouveau [module](#_Modules)

Ajoute un [module](#_Modules) existant

Renomme le [module](#_Modules)

Supprime le [module](#_Modules) courant

Construit la configuration.

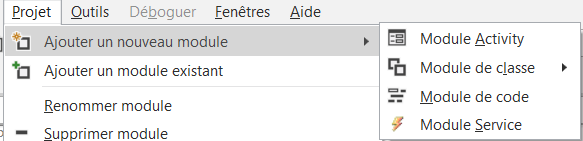
Compile et exécute le projet.

Compile et exécute le projet en arrière-plan.

Compile en une bibliothèque.

### Ajouter un nouveau module

**B4A**



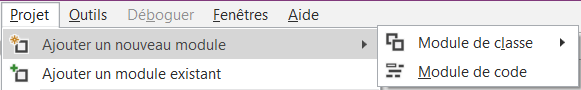
Module Activity

Module de classe

Module de code

Module Service

**B4i, B4J**



Module de classe

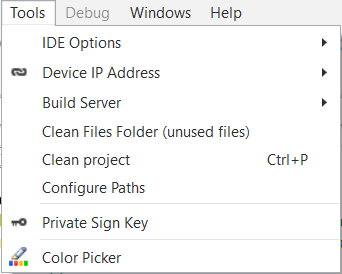
Module de code

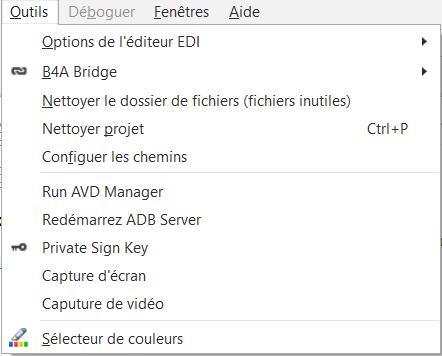
**B4R**



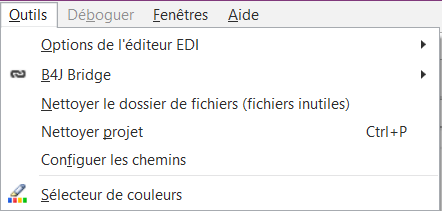
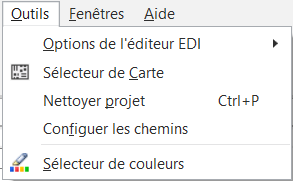
Seulement module de code

## Menu outils

**B4A B4i**



**B4R** **B4J**



Options de l’éditeur EDI Voir ci-dessous

[B4A Bridge](#_B4A_Bridge_1), connexion avec Wifi B4A

[Nettoyer le dossier…(fichiers inutiles)](#_Clean_Files_Folder) B4A, B4i, B4J

[Nettoyer le projet](#_Clean_Project) Tous

[Configurer les chemins](#_Install_and_configure) Tous

[Redémarrer ADB Server](#_Launch_an_Emulator) B4A

[Capture d’écran](#_Take_Screenshot) B4A

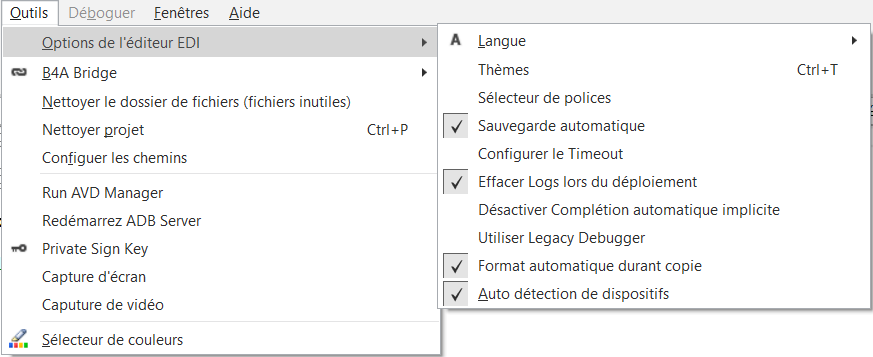
[Capture de vidéo](#_Create_Video) B4A

Affiche le [Sélecteur de couleurs](#_Color_Picker) Tous

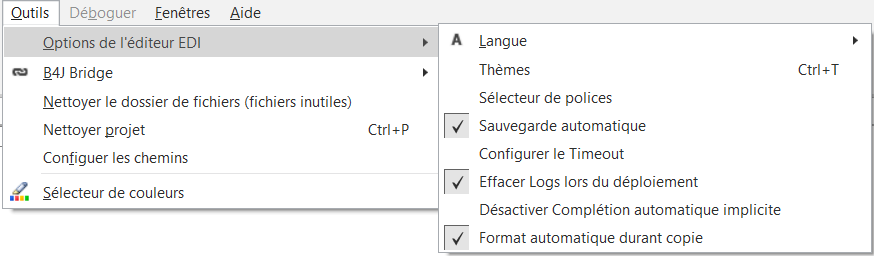
Sélecteur de carte B4R

### Options de l’éditeur EDI

**B4A**



**B4i, B4J, B4R**



**Tous**

Langue.

[Thèmes](#_Themes).

[Sélecteur de polices](#_Font_Picker).

Sauvegarde automatique Sauvegarde le programme chaque fois que vous le compilez.

Effacer Logs lors du déploiement Supprime tous les Log lors de la compilation en mode Release.

[Désactiver Complétion automatique implicite](#_Disable_Implicit_Auto).

Format automatique durant copie

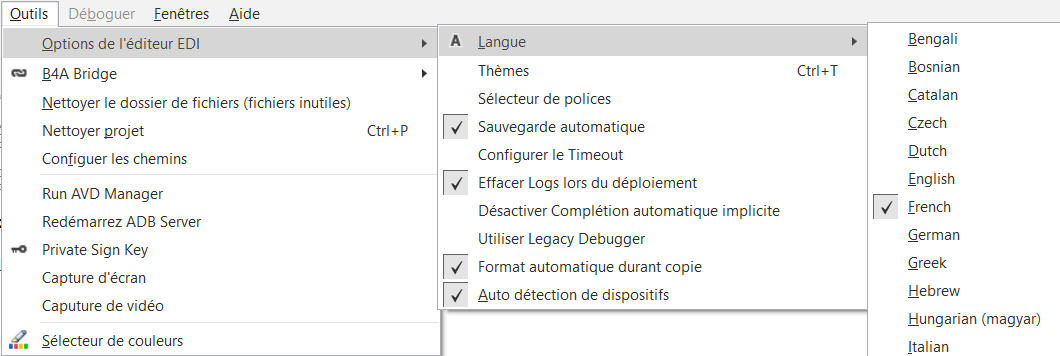
**Seulement B4A**

[Utiliser Legacy Debugger](#_Debug_(legacy)_mode) Utiliser le ‘Legacy Debugger’ au lieu du débogueur rapide.

Auto détection de dispositifs Détecte automatiquement des dispositifs connectés.

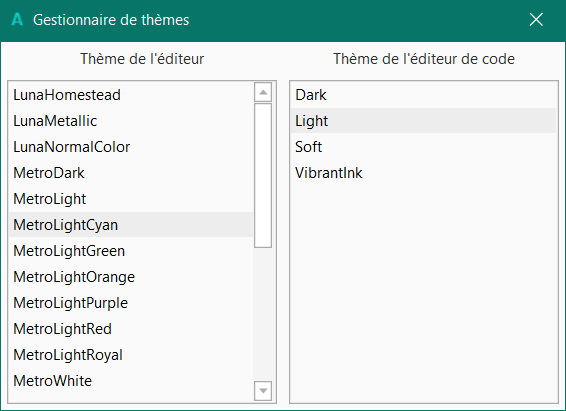
#### Langue

Vous pouvez choisir la langue pour l’EDI, dans le menu Outils / Options



Sélectionnez la langue désirée dans la liste des langues actuellement disponibles.

#### Thèmes

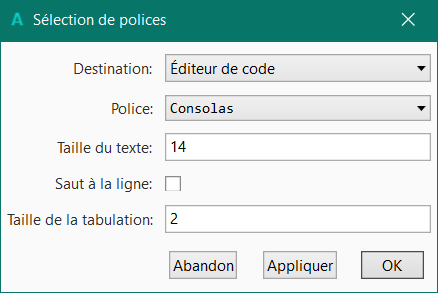


Vous pouvez choisir différents thèmes pour l’EDI.

Le thème par défaut est différent selon le produit B4x.

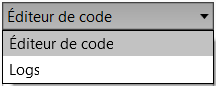
Lorsque vous en sélectionnez un, vous voyez immédiatement les nouvelles couleurs.

#### Sélecteur de police



Vous pouvez choisir la Destination :

Éditeur de Code ou Logs.



Différentes polices.

La taille du texte.

Saut à la ligne.

Taille de la tabulation.

##### Saut à la ligne

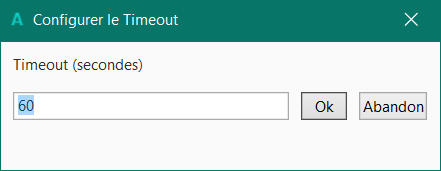


Sans saut à la ligne. La fin de la ligne est cachée.



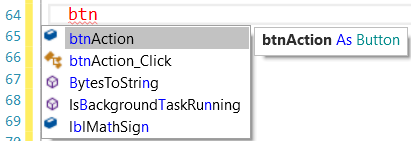
Avec saut à la ligne. La fin de la ligne est affichée dans la ligne suivante.

#### Configurer le Timeout



La compilation peut prendre plus de temps dans certains cas. Si vous obtenez un message ‘Process timeout’ vous pouvez augmenter le temps.

#### Désactiver Complétion automatique implicite



Si 

n’est pas coché, une liste déroulante avec les mots possibles sera affichée durant l’écriture.

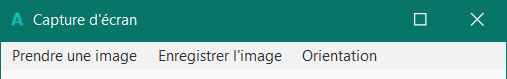
Si est coché, la liste de complétion automatique ne sera pas affichée.

### Capture d’écran seulement B4A

La fonction  peut être appelée depuis le :

* Menu Outils lorsque l’EDI est en mode édition.
* Menu Débogueur lorsque l’EDI est un mode débogage.

**Note : Cette fonction ne fonctionne qu’avec une connexion par USB, pas avec B4A-Bridge !**



Un clic sur

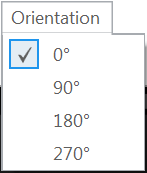
affiche la fenêtre ci-dessous.



Cliquez sur  pour prendre une image de l’écran du dispositif.

Vous pouvez changer la dimension de l’image au moyen du curseur sur la gauche.

Vous pouvez enregistrer l’image dans un fichier PNG avec .



Et vous pouvez changer son orientation.

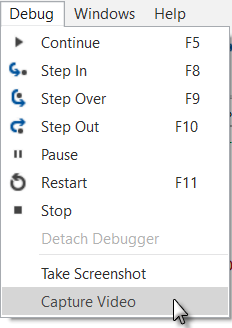
Cliquez avec le bouton droit de la souris sur l’image pour la copier dans le presse-papiers.



### Capture de vidéo seulement B4A

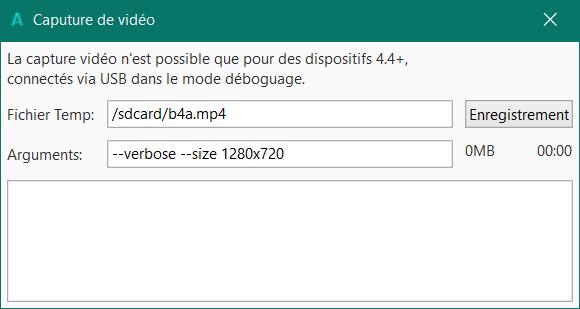
Vous pouvez lancer votre programme et enregistrer une vidéo pendant son utilisation.

**Note : Cette fonction ne fonctionne qu’avec une connexion par USB, pas avec B4A-Bridge !**



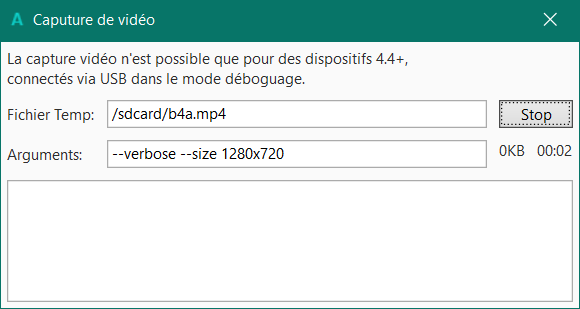
Dans le menu  cliquez sur .

La fenêtre ci-dessous sera affichée :



Cliquez sur  pour démarrer l’enregistrement.

Une fenêtre similaire à celle-ci-dessous sera affichée :



Cliquez sur  pour arrêter l’enregistrement.

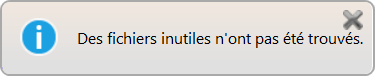
Le programme vous demandera où enregistrer le fichier sur votre ordinateur.

#### 

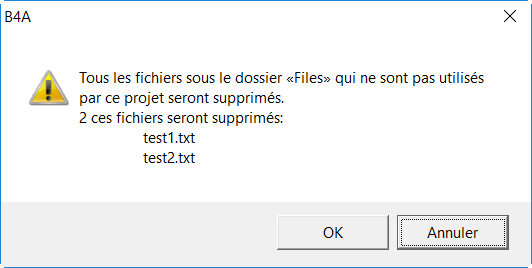
### Nettoyer le dossier de fichiers (fichiers inutiles)

Efface, dans le dossier Files du projet, tous les fichiers qui ne sont pas utilisés dans le projet (les fichiers référencés dans les layouts ne sont pas effacés). Une liste avec tous les fichiers non utilisés sera affichée avant l’opération (vous pouvez toujours abandonner l’opération).

S’il n’y a pas de fichiers non utiliés, le message ci-dessous sera affiché.



Si non, une fenêtre similaire à celle-ci-dessous sear affichée.

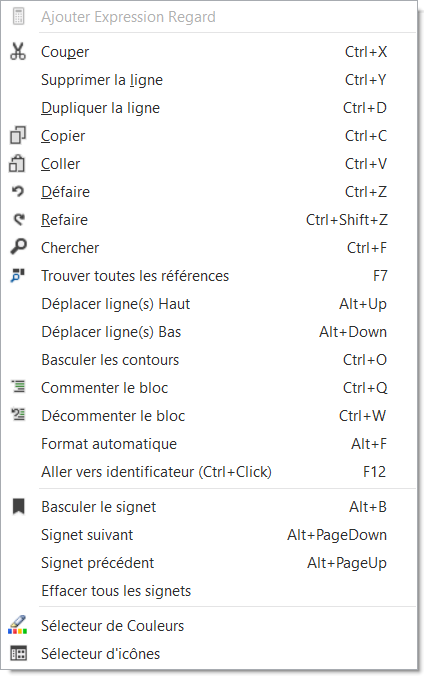


### Nettoyer le projet

Efface tous les fichiers générés lors de la compilation.

## Menu clic droit

Si vous cliquez, dans la zone de code, avec le bouton droit de la souris, le menu contextuel ci-dessous sera affiché.



Couper

Supprimer la ligne

Dupliquer la ligne

Copier

Coller

[Défaire](#_Undo_–_Redo)

[Refaire](#_Undo_–_Redo)

[Chercher](#_Find_All_References)

[Trouver toutes les références](#_Find_All_References)

[Déplacer ligne(s) Haut](#_Move_lines_up_1)

[Déplacer ligne(s) Bas](#_Move_lines_up_1)

[Basculer les Contours](#_Basculer_les_contours)

[Comment](#_Commenting_and_uncommenting_1)er le bloc

[Décommenter le bloc](#_Commenting_and_uncommenting_1)

[Format](#_Auto_format_1) automatique

[Aller vers identificateur](#_Jump_to_a)

[Basculer les signets](#_Bookmarks_1)

[Signet suivant](#_Bookmarks_1)

[Signet précédent](#_Bookmarks_1)

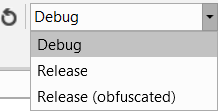
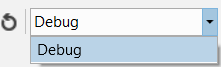
[Effacer signets](#_Bookmarks_1)

[Sélecteur de couleur](#_Color_Picker)

[Sélecteur d’icônes](#_Icon_Picker_1) Pas dans B4R.

## Mode de compilation

A côté de la barre d’outils il y a une liste déroulante pour sélectionner le mode de compilation.

[B4A / B4J](#_B4A_and_B4J) [B4i](#_B4i) [B4R](#_B4R)

Le débogage est expliqué en détail dans le chapitre [Débogage](#_Debugging_1).

### B4A et B4J

Modes de compilation :

* [Debug](#_Debug_(rapid)_mode)
* [Release](#_Release_and_Release)
* [Release (obfuscated)](#_Release_and_Release)

#### Modes Release et (obfuscated) dans B4A et B4J

**Pour distribuer votre projet, vous devez le compiler avec :**

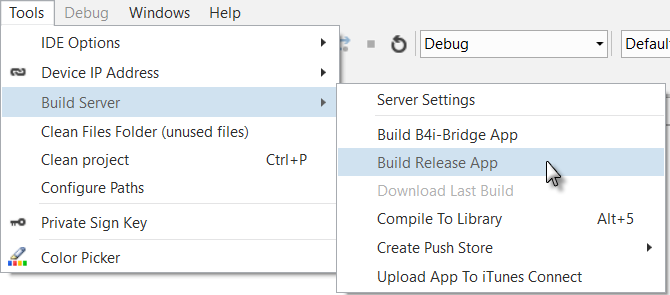
* Release  
  Le code de débogage ne sera pas ajouté au fichier apk de distribution.
* Release (obfuscated)   
  Le code de débogage ne sera pas ajouté au fichier apk de distribution,  
  mais sera modifié. Voir ci-dessous.

Pendant la compilation, B4A génère du code Java qui est, ensuite, compilé avec le compilateur Java et converti en Dalvik (code au format Android).  
Il existe des outils permettant de décompiler du code Dalvik en code Java.  
  
L’objectif de ‘l’obfuscation’ (obscurcissement) est de rendre le code décompilé mois lisible, plus difficile à comprendre et plus difficile d’en extraire du texte tel les clés de comptes développeur.  
  
Il est important de comprendre comment cet obscurcissement fonctionne.  
L’ obscurcissement consiste en deux choses :  
  
**Strings obfuscation** (obscurcissement de texte)  
Tout texte écrit dans la Sub Process\_Globals (et seulement dans cette routine) sera obscurci, rendant plus difficile l’extraction de clés importantes. Les textes sont désobscurcis pendant l’exécution.  
Notez que plusieurs clés sont utilisées pendant l’obscurcissement incluant le nom du paquet, le nom de la version et le code de la version. Une modification de ces valeurs dans l’éditeur manifeste va rompre le processus de désobscurcissement.  
  
**Variables renaming** (renommer les variables)  
Les noms des variables globales et les noms des routines sont convertis en textes non significatifs. Les variables locales ne sont pas affectées car leurs noms sont de toute façon perdus lors de la compilation.  
Les identifiants suivants ne sont **pas** convertis :  
- Identifiants comprenant un caractère soulignage (nécessaire pour les routines d’événement).  
- Les noms de routines dans les appels avec CallSub. Quand un nom de routine apparait comme un texte statique, l’identifiant est gardé tel.  
- Noms définis dans le Designer.  
  
Conseil : Si pour une raison quelconque vous voulez éviter l’obscurcissement pour un identifiant, ajoutez un caractère de soulignage dans le nom.  
  
Un fichier nommé ObfuscatorMap.txt est créé dans le dossier Objets. Ce fichier Map fait correspondre les noms des identifiants à ceux obscurcis. Cette correspondance peut être utile pour l’analyse de rapports de ‘plantages’.

### B4i

Pour distribuer votre projet vous devez le compiler en mode Release.

Cliquez sur  dans le menu Tools / Build Server.



### B4R

Il n’existe qu’un mode de compilation, le mode Release.

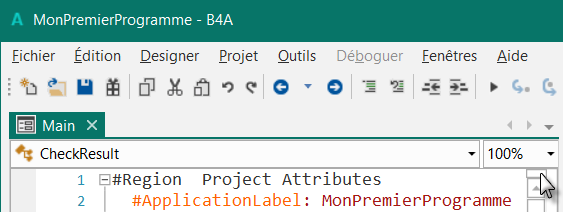
# Zone code

Le code pour le module sélectionné est affiché dans cette zone et peut être édité.

Les exemples ci-dessous sont basés sur le projet MonPremierProgramme du livret ‘Premiers Pas’.

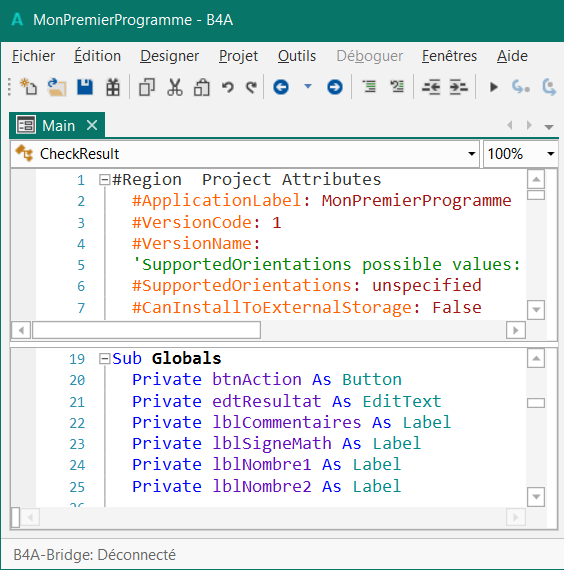
## Diviser la zone code

Il est possible de diviser la zone code en deux parties permettant d’éditer deux parties différentes du code en même temps.



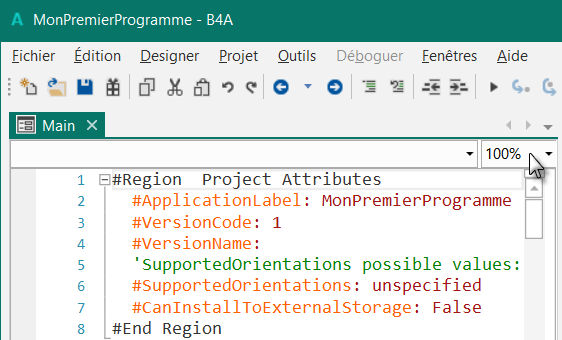
Déplacez le petit rectangle juste sous l’indication du niveau de zoom.

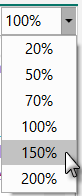
Et le résultat.



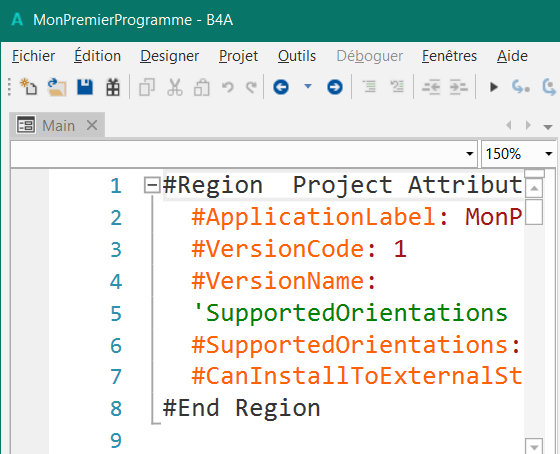
## Taille du texte de l’EDI

La taille du texte dans l’éditeur peut être choisie avec le [Sélecteur de police](#_Font_Picker) ou directement dans l’EDI :





Cliquez sur 100% et sélectionnez une des valeurs de zoom.



## Entêtes de code Project Attributes / Activity Attributes

Un entête de code, avec des réglages généraux, est ajouté au début du code.

### B4A

#### Project Attributes attributs projet

Attributs qui sont valables pour tout le projet. Affichés uniquement dans le module Main.

#Region Project Attributes  
 #ApplicationLabel: MonPremierProgramme  
 #VersionCode: 1  
 #VersionName:  
 'SupportedOrientations possible values: unspecified, landscape or portrait.  
 #SupportedOrientations: unspecified  
 #CanInstallToExternalStorage: False  
#End Region

#ApplicationLabel: Le nom qui figurera sous l’icône du programme sur le dispositif.

#VersionCode: La version du code, n’est pas affiché.

#VersionName: Vous pouvez ajouter un nom de version.

#SupportedOrientations: Vous pouvez limiter tout le programme pour une orientation donné.

Orientations possibles :

unspecified = non spécifié

landcape = paysage

portrait = portrait

#CanInstallToExternalStorage: Si vous voulez installer le programme sur une carte externe vous

devez mettre cette valeur à True.

Vous pouvez adapter les valeurs à vos besoins.

#### Activity Attributes attributs Activity

Valable pour l’Activity courante.

#Region Activity Attributes  
 #FullScreen: False  
 #IncludeTitle: True  
#End Region

Lorsque vous ajoutez un nouveau module Activity vous trouvez ces attributs au début du code.

#Region Activity Attributes

#FullScreen: False

#IncludeTitle: True

#End Region

FullScreen = plein écran, valeur par défaut False

IncludeTitle = inclure barre de titre, valeur par défaut True

#### Service Attributes attributs service

Lorsque vous ajoutez un nouveau module Service vous trouvez ces attributs au début du code.

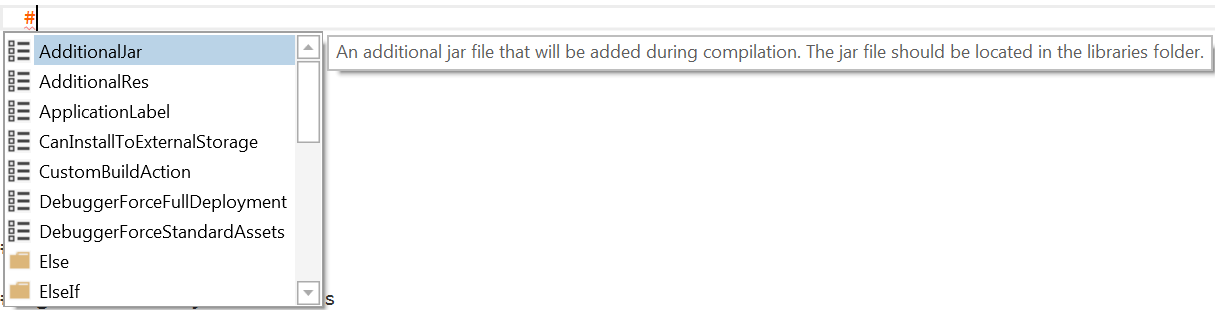
#Region Service Attributes

#StartAtBoot: False

#End Region

StartAtBoot = démarre lors du lancement

Lorsque vous voulez ajouter un nouvel attribut vous pouvez simplement écrire # et l’aide en ligne vous montre toutes les possibilités.



Notez les deux icônes différentes :

 Attributs.

 Compilation conditionnelle et mots clé Region.

Si vous chargez un programme enregistré avec une version de B4A plus ancienne que 2.5, l’entête ressemblera à ceci :

#Region Module Attributes

#FullScreen: False

#IncludeTitle: True

#ApplicationLabel: B4AExample

#VersionCode: 1

#VersionName:

#SupportedOrientations: unspecified

#CanInstallToExternalStorage: False

#End Region

### B4i

Seulement les attributs ci-dessous. Aucun autre attribut dans les modules.

'Code module

#Region Project Attributes

#ApplicationLabel: B4i Example

#Version: 1.0.0

'Orientation possible values: Portrait, LandscapeLeft, LandscapeRight and PortraitUpsideDown

#iPhoneOrientations: Portrait, LandscapeLeft, LandscapeRight

#iPadOrientations: Portrait, LandscapeLeft, LandscapeRight, PortraitUpsideDown

#Target: iPhone, iPad

#ATSEnabled: True

#MinVersion: 7

#End Region

### B4J

Seulement les deux attributs ci-dessous. Aucun autre attribut dans les modules.

#Region Project Attributes

#MainFormWidth: 600

#MainFormHeight: 600

#End Region

### B4R

Seulement les attributs ci-dessous. Aucun autre attribut dans les modules.

#Region Project Attributes

#AutoFlushLogs: True

#CheckArrayBounds: True

#StackBufferSize: 300

#End Region

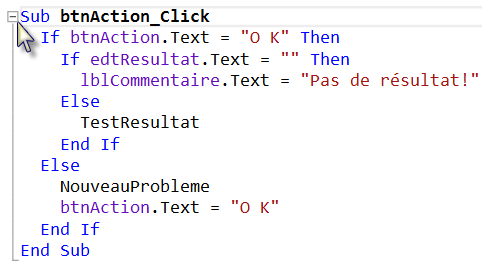
## Annuler – Refaire

Dans l’EDI il est possible d’annuler des opérations et refaire des opérations annulées.

Cliquez sur pour annuler et sur pour refaire.

## Réduire une routine

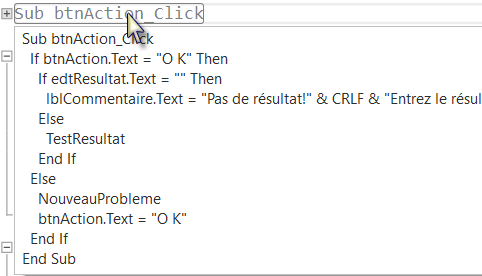
Une routine peut être réduite pour minimiser le nombre de lignes.



La routine btnAction\_Click étendue.

Cliquez sur  pour réduire la routine.

La routine btnAction\_Click réduite.

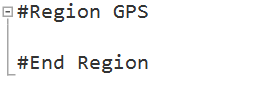


Passer avec la souris sur la routine réduite montre son contenu.

## Réduire une Région

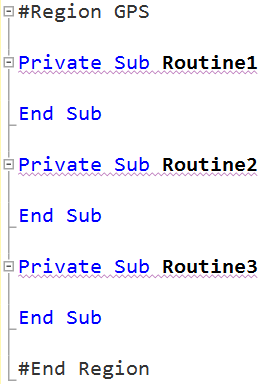
Vous pouvez définir des Régions dans votre code qui peuvent être réduites.

Exemple :

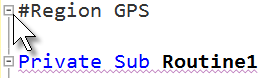


#Region GPS définit le début d’une région et

#End Region définit la fin.



Ensuite, vous pouvez ajouter des routines entre les deux limites.

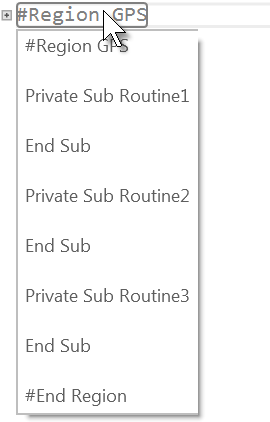


Ensuite, cliquer sur  pour réduire la région

.

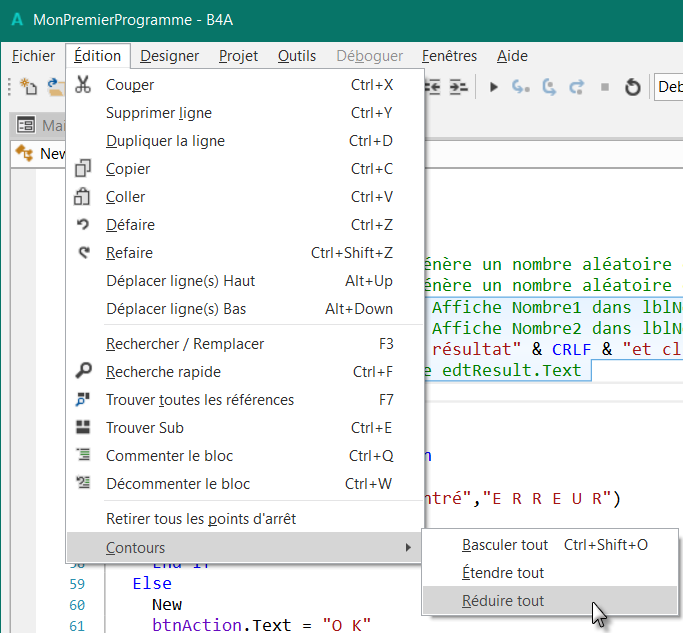


Passer au-dessus de #Region GPS



Montre le code. Pour les grandes régions tout le code n’est pas affiché.

## Réduire tout le code en entier



Dans le menu

Édition / Contours

il y trois fonctions :

- Basculer tout

Étend les routines et régions réduites et réduit les routines et régions étendues.

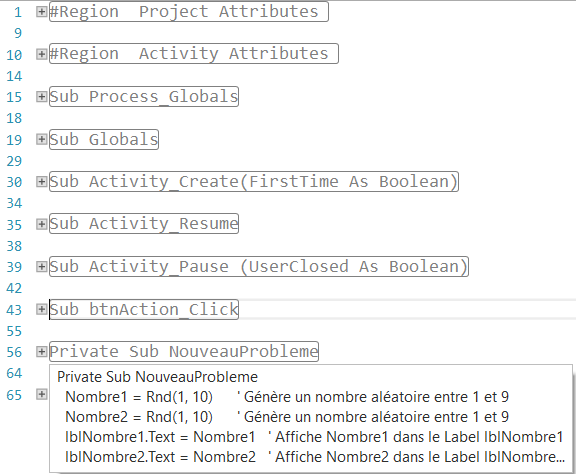
- Étendre tout

Étend tout le code.

- Réduire tout

Réduit tout le code.

Cliquez sur .



Tout le code est réduit.

Passer avec la souris au-dessus d’une routine montre le début de son contenu.

## Basculer les contours Ctrl + 0

Vous pouvez basculer les contours.

Exemple :

Sub **btnAction\_Click**

If btnAction.Text = "O K" Then

If edtResultat.Text = "" Then

lblCommentaire.Text = "Pas de résultat!" & CRLF & "Entrez le résultat" & CRLF & "et cliquez sur OK."

Else

TestResultat

End If

Else

NouveauProbleme

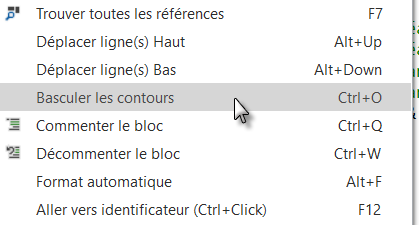
btnAction.Text = "O K"

End If

End Sub

Cliquez à l’intérieur de la routine et pressez Ctrl + 0.

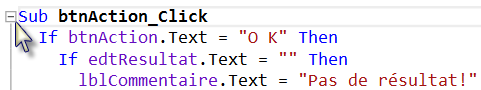
Ou cliquez avec le bouton droit de la souris pour montrer le menu contextuel et cliquez sur  pour réduire la routine.



Et le résultat.



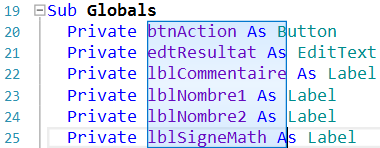
C’est la même chose que de cliquer sur .



## Copier un texte sélectionné

Il est possible de copier un bloc de texte vers le presse-papiers, pas seulement des lignes entières.

Pour sélectionner le texte, pressez Alt et déplacez la souris.



## Déplacer ligne(s) Haut / Bas Alt + Up / Alt + Down

Vous pouvez déplacer des lignes sélectionnées vers le haut ou vers le bas.

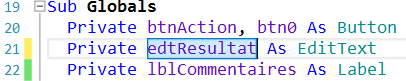
Soit avec Alt + Up ou Alt + Down.

Ou cliquez avec le bouton droit de la souris pour afficher le menu contextuel puis cliquez sur  ou .

## Rechercher / Remplacer

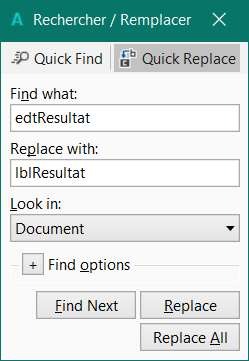
L’exemple utilise le code du projet MonPremierProgramme.

Nous allons remplacer edtResultat par lblResultat.



Dans le code, sélectionnez edtResult.

Pressez F3 ou cliquez sur  dans le menu  .

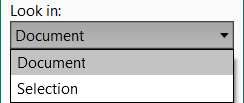


Cette fenêtre sera affichée.

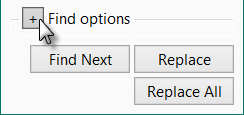
Entrez lblResult dans le champ ‘Replace with’ (remplacer par).

Vous pouvez maintenant :

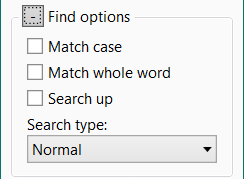
*  chercher la prochaine occurrence.
*  remplacer l’occurrence courante par le texte dans *Replace with:* et chercher la prochaine l’occurrence.
*  remplacer toutes les occurrences.



Vous pouvez chercher soit dans une portion de code sélectionnée ou dans le Document, ce qui veut dire dans le module actif et pas le programme en entier.



Vous pouvez définir des options de recherche, cliquez sur .



Match case = Sensible à la casse

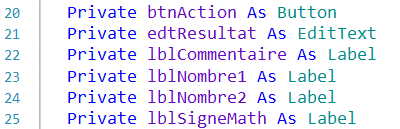
Match whole word = Correspondance mot entier

Search up = Recherche de bas en haut

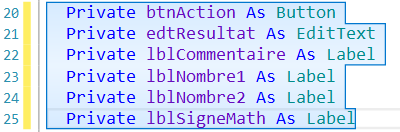
**Note : Cette fenêtre n’est actuellement pas traduite dans l’EDI.**

## Commenter et décommenter du code

Des lignes sélectionnées peuvent être commentées ou décommentées.

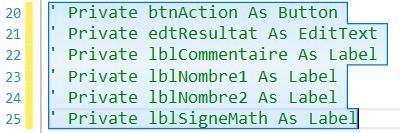


Code original.



Sélectionnez le code.

Cliquez sur ou pressez Ctrl + Q.



Les lignes sélectionnées sont commentées.

Pour décommenter des lignes, sélectionnez-les et cliquez sur ou pressez Ctrl + W.

Ou, avec un clic droit et un clic sur  ou .

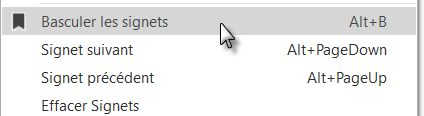
### 

## Signets

Vous pouvez activer des ‘signets’ n’importe où dans le code et naviguer en avant ou en arrière entre ces signets.

Pour activer ou effacer un signet, sélectionnez la ligne puis pressez sur Alt + B.

Ou, avec un clic droit sur la ligne concernée.

Dans le menu contextuel, cliquez sur



Pour activer ou effacer un signet.

Vous verrez cette  marque dans le bord gauche de la ligne et une petite ligne noire dans l’ascenseur à droite :



Pour sauter au prochain signet pressez Alt + PgDn.

Ou, avec un clic droit et sur 

Pour sauter au prochain signet précédent pressez Alt + PgUp.

Ou, avec un clic droit et sur 

Pour effacer tous les signets, un clic droit puis un clic sur 

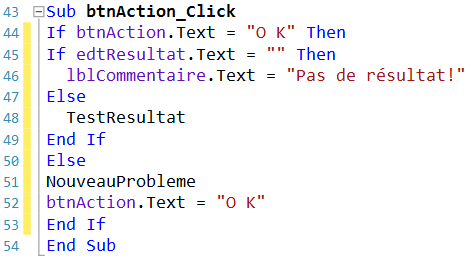
## Indentation

Une bonne pratique est d’indenter des portions de code pour une lecture plus facile.

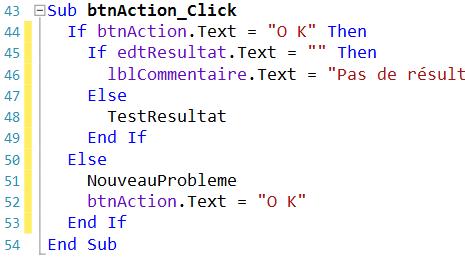
B4x le fait d’ailleurs automatiquement.

Par exemple pour les routines, boucles, structures etc.

Voyez aussi [Format automatique](#_Auto_format_1).

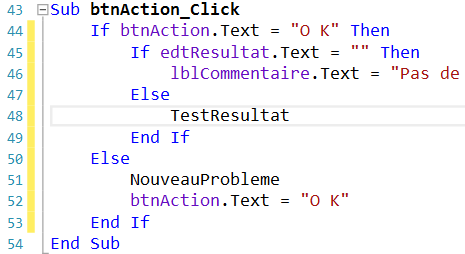


Ce code est difficile à lire car la structure de code n’est pas mise en évidence.



Ce code est facile à lire car sa structure est mise en évidence.

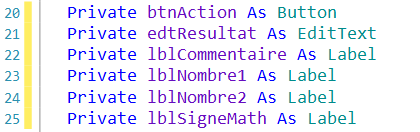
Une valeur de tabulation de 2 pour l’indentation est une bonne valeur.



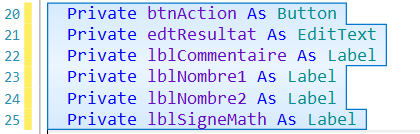
Exemple avec une indentation de 4, valeur par défaut.

Personnellement je préfère de loin la valeur de 2.

Des blocs entiers peuvent être indentés en avant ou en arrière en une fois.

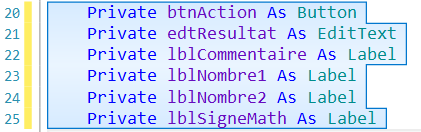


Code original.



Sélectionnez le bloc de code.

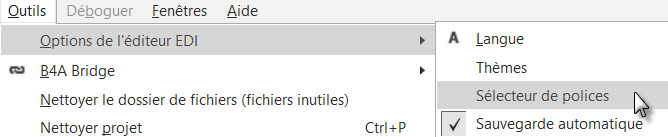
Cliquez sur .

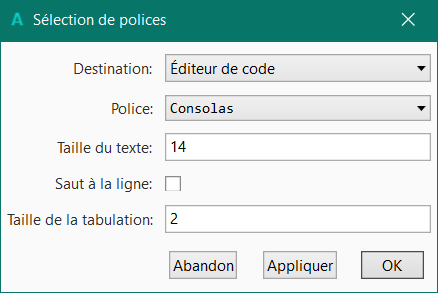
Le bloc, en entier a été déplacé d’une tabulation vers la droite.

Pour déplacer un bloc vers la gauche,

sélectionnez le bloc et cliquez sur  .

La valeur de la tabulation pour l’indentation peut être changée dans le sélecteur de police dans le menu Outils / Sélecteur de police.



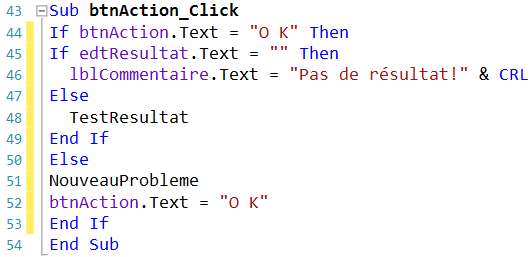


Entrez la valeur et cliquez sur .

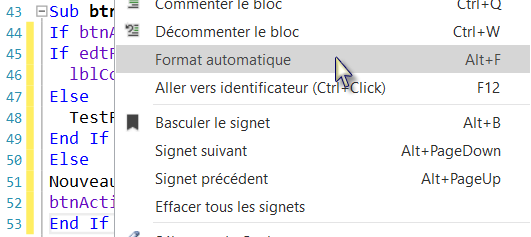
Le code est reformaté automatiquement.

## Format automatique

Vous pouvez aussi formater du code automatiquement.



Ce code est difficile à lire.

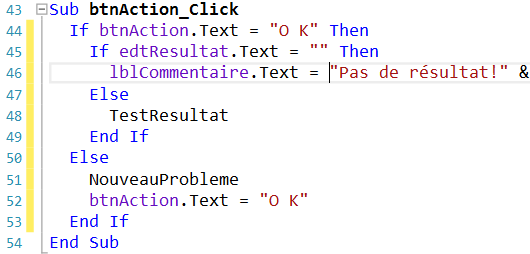


Sélectionnez le code.

Cliquez avec le bouton droit de la souris pour afficher le menu contextuel.

Et cliquez sur 

ou pressez Alt + F.



Et le résultat.

La grandeur de la tabulation dépend de vos réglages, voir page précédente.

## Documentation en passant au-dessus d’éléments de code

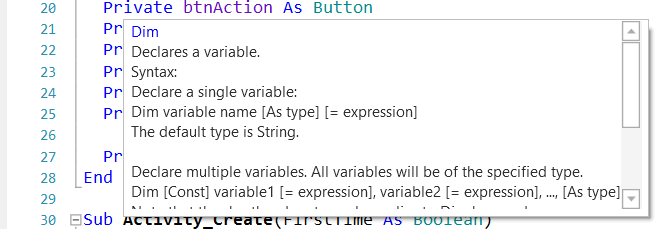
Lorsque vous passez avec la souris au-dessus d’un élément de code, l’aide en ligne sera affichée.

Exemples :

Passez au-dessus de Globals :



Passez au-dessus de Private :

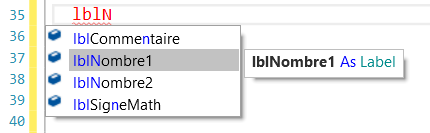


## Complétion automatique

Un outil très utile est la fonction de complétion automatique.

Attention : Assurez-vous que , dans le menu Options / Options de l’éditeur EDI, ne soit pas coché !

Exemple avec le code du projet MonPremierProgramme :



Écrivons lblN.

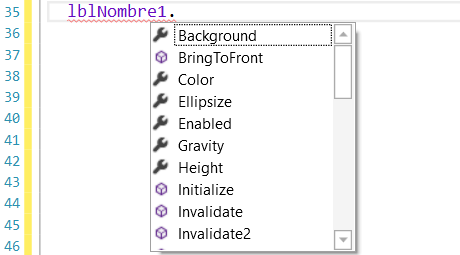
Toutes les variables, views et propriétés dont le nom commence par les lettres déjà écrites sont affichées dans un menu contextuel.

Pour choisir lblNumber1 pressez Entrée.

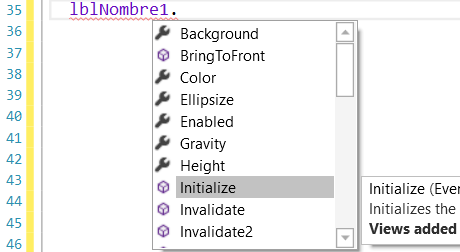


Le nom sélectionné est complété.

Pour choisir lblNumber2 double cliquez sur ou pressez la flèche vers le bas et pressez Entrée.



Ajoutez un point ".", et toutes les propriétés et méthodes de l’objet sélectionné sont affichées dans le menu contextuel.



Lorsqu’on sélectionne un élément, son aide en ligne est affichée

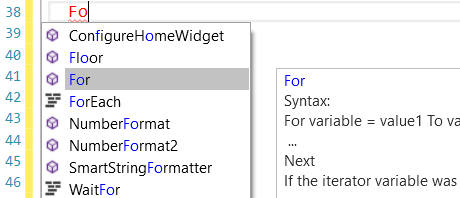
En pressant sur les flèches haut ou bas on peut sélectionner l’élément précédent ou suivante avec son aide en ligne.

En pressant sur un caractère la liste est mise à jour avec les paramètres commençant par ce caractère.

Les structures sont aussi complétées.

Exemples :

**For / Next**

 Écrivez ‘Fo’.

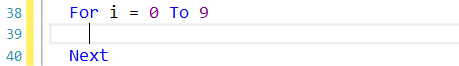
Vous obtenez For avec son aide en ligne.

Pressez Entrée.

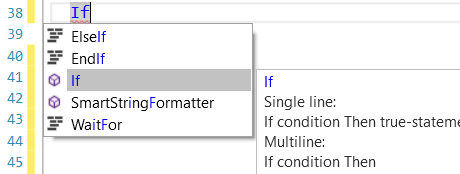
For est complété.

Écrivez le reste de l’instruction.

Et pressez Entrée.

Next est automatiquement ajouté et le curseur se trouve au début de la ligne suivante indenté.

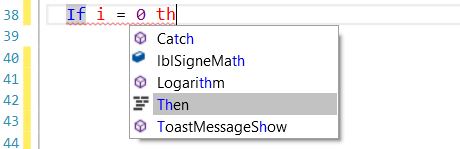
**If / Then**



Écrivez ‘if’.

Vous obtenez If avec son aide en ligne.

Pressez Entrée et écrivez la suite comme dans l’exemple.



Après ‘th’ vous obtenez Then avec son aide en ligne.

Pressez Entrée.

Et pressez Entrée encore une fois.

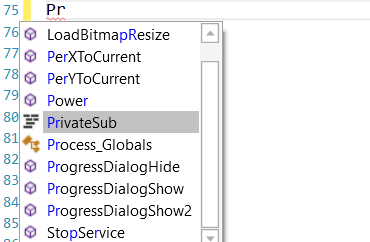
 End If est automatiquement ajouté et le curseur se trouve au début de la ligne suivante indenté.

**Le meilleur moyen pour apprendre ces fonctions est de ‘jouer’ avec elles.**

Une autre fonction de complétion automatique vous permet de créer des routines d’événement.

Dans l’exemple ci-dessous nous voulons créer la routine d’événement pour un bouton bntOK.

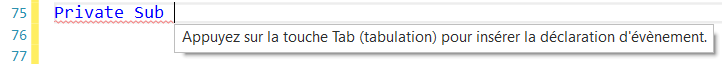
Écrivez Pr et l’aide propose les mots clé qui contiennent les deux lettres Pr.



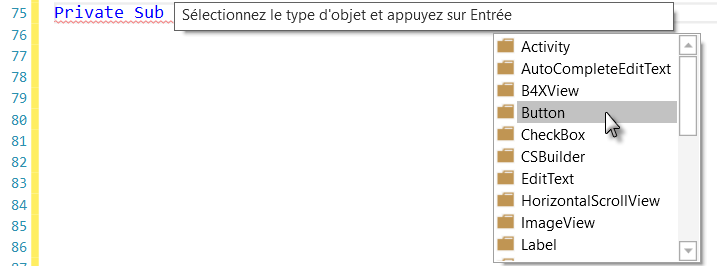
Pressez Entrée pour sélectionner Private Sub.



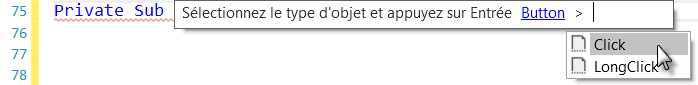
Pressez la barre de caractère vide.



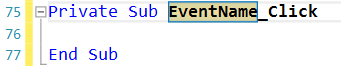
Pressez Tab et sélectionnez le type d’objet, Button dans notre cas.



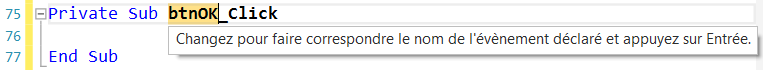
Tous les événements pour l’objet Button sont affichés, sélectionnez .



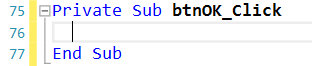
Le cadre de la routine est généré.



Remplacez 'EventName' par le nom de l’événement pour le Button, dans notre exemple btnOK.



Pressez Entrée et la routine est terminée.



Le curseur se trouve dans la première ligne et indenté.

## Documentation intégrée

Une autre fonction utile est la documentation intégrée.  
  
Des commentaires au-dessus d’une routine, tel que :

'Dessine une croix aux coordonnées données avec la couleur donnée

'x any y = coordinées en pixels

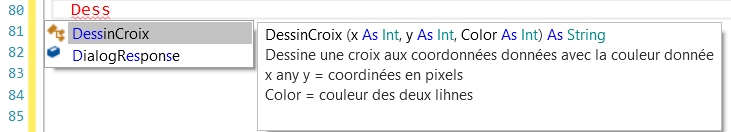
'Color = couleur des deux lihnes

Sub **DessinCroix**(x As Int, y As Int, Color As Int)

Private d = 3dip As Int

cvsLayer.DrawLine(x - d, y, x + d, y, Color, 1)

cvsLayer.DrawLine(x, y - d, x, y + d, Color, 1)

End Sub  
  
Seront automatiquement affichés dans la fenêtre d’aide :  
  


Si vous voulez ajouter un exemple de code, utilisez <code> </code> :

'Dessine une croix aux coordonnées données avec la couleur donnée

'x any y = coordinées en pixels

'Color = couleur des deux lihnes

'Exemple de code: <code>

'DessinCroix(20dip, 50dip, Colors.Red)

'</code>

Sub **DessinCroix** (x As Int, y As Int, Color As Int)

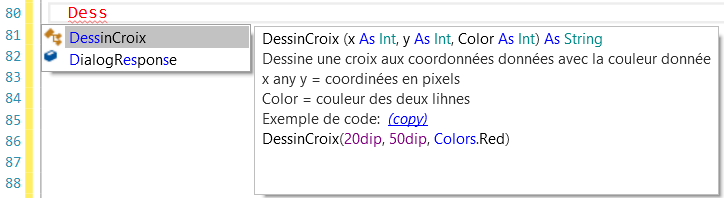
Private d = 3dip As Int

cvsLayer.DrawLine(x - d, y, x + d, y, Color, 1)

cvsLayer.DrawLine(x, y - d, x, y + d, Color, 1)

End Sub

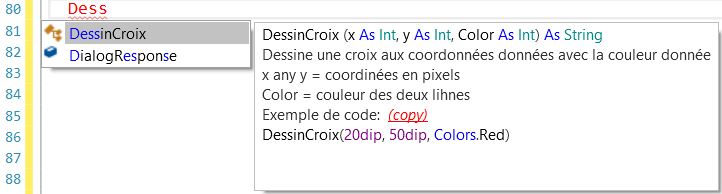
Le code sera affiché avec la syntaxe en couleurs :



### Copie d’exemples de code

Vous pouvez copier les exemples de code depuis la fenêtre d’aide.

En passant avec la souris au-dessus de *(copy)*, qui passe en rouge, vous pouvez copier l’exemple de code dans le presse-papiers en cliquant sur *(copy)*.



Supprimez Dess



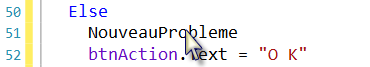
Et copiez le presse-papiers.



## Sauter vers une routine

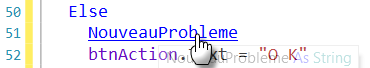
Quelque fois il est utile de pouvoir sauter depuis un appel à une routine vers sa définition.

Ceci se fait facilement :

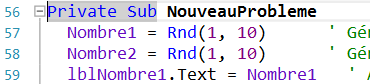


Pointez avec la souris le texte de l’appel à la routine.

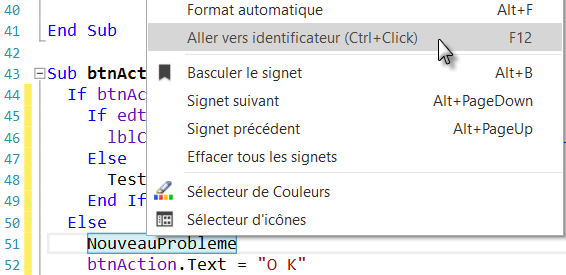
Puis pressez Ctrl, le nom de l’appel sera souligné.



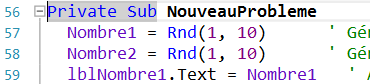
Et cliquez.

  
Et vous y êtes.

Une autre méthode.

Sélectionnez le texte de l’appel à la routine.

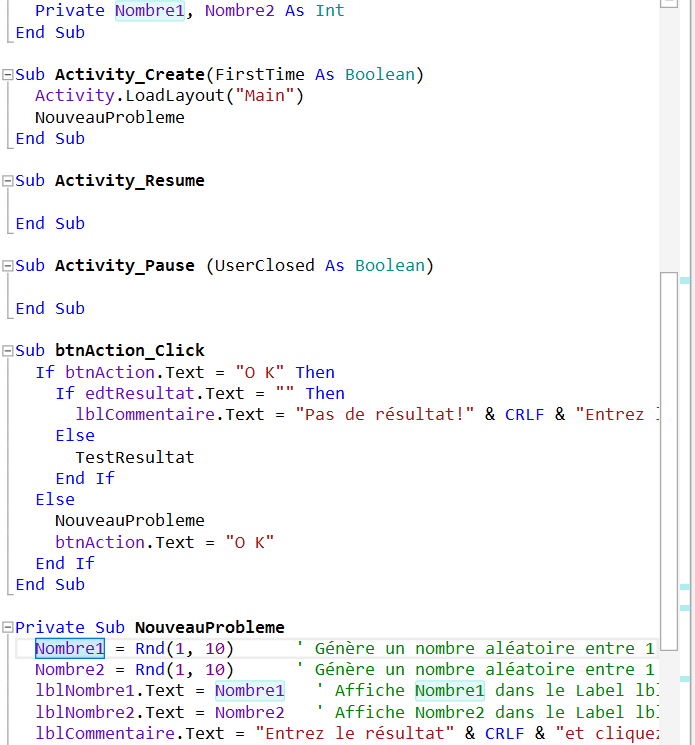
Cliquez avec le bouton droit de la souris sur le texte sélectionné pour afficher le menu contextuel.  
  
Cliquez sur.   


  
  
  
Et vous y êtes.

## Mettre en évidence des occurrences de mots

Lorsque vous sélectionnez un mot, il est mis en évidence par une couleur bleu foncé et toutes les autres occurrences de ce mot sont mises en évidence par une couleur bleu clair et à côté de l’ascenseur à droite de petites lignes bleu clair sont affichées.

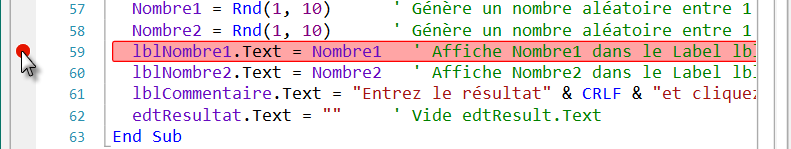
Vous pouvez vous déplacer dans le code avec l’ascenseur vers les autres occurrences du mot sélectionné.



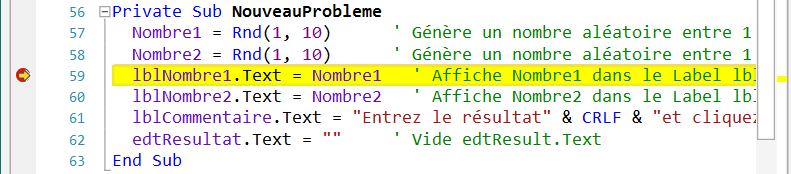
## Points d’arrêt

Cliquez dans la marge gauche d’une ligne pour ajouter un point. Lors de l’exécution du programme celui-ci s’arrête au premier point d’arrêt.

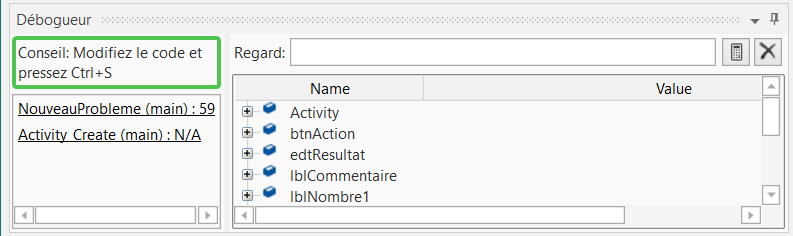
**Des points d’arrêt dans les routines Globals, Process\_Globals et Activity\_Pause sont ignorés.**



Lancez le programme, il s’arrête sur le point d’arrêt et l’éditeur ressemble à l’image ci-dessous. La ligne dans laquelle le programme s’arrête et mise en évidence par une couleur jaune.



Au bas de l’éditeur vous trouvez d’autres informations.

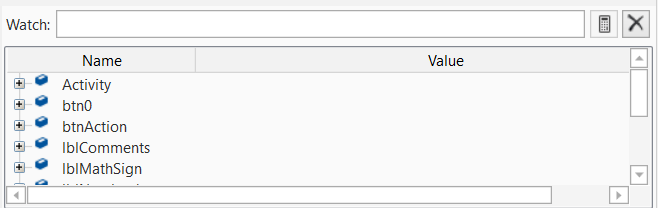


Le débogueur est connecté. Dans sa partie gauche nous avons :

*  Un bouton pour mettre à jour le code après une modification.
*  Le nom de la routine dans laquelle le programme s’est arrêté.  
   NouveauProbleme dans le module Main à la ligne 59.
* Routine d’appel de la routine ‘NouveauProbleme’ :   
   Activity\_Create dans le module Main.

Un clic sur une de ces lignes déplace le curseur vers celle-ci.

Dans la partie droite du débogueur nous trouvons une liste d’objets et de variables avec leurs valeurs.



Dans la barre d’outils, sur le haut de l’éditeur, les boutons de navigations sont activés.



 Entrer F8 Exécute l’instruction suivante.

 Sauter F9 Saute l’appel d’une routine, mais exécute son code sans y entrer.

 Sortir F10 Exécute le reste de la routine et sort.

 Stop Arrête le programme.

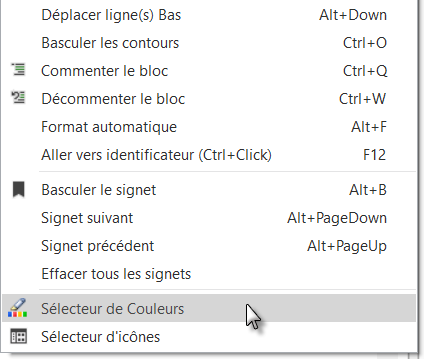
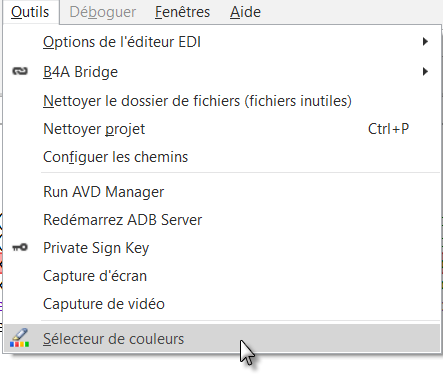
 Redémarrer F11 Redémarre le programme.

Plus de détails dans le chapitre [Débogage](#_Débogage__B4A,).

## Sélecteur de couleur

Dans le code, clic droit pour afficher

le menu contextuel ci-dessous. Ou dans le menu Outils.

Cliquez sur  pour afficher le Sélecteur de Couleurs.

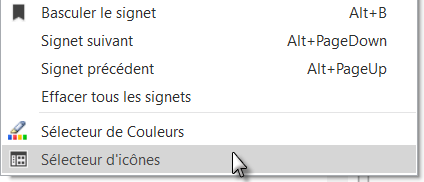


Vous pouvez :

* Déplacer le curseur dans les zones carrée et rectangulaire.
* Entrer directement les valeurs A R G B  
  A = Alpha, R = Rouge, G = Vert, B = Bleu
* Copier la valeur dans le presse-papiers.

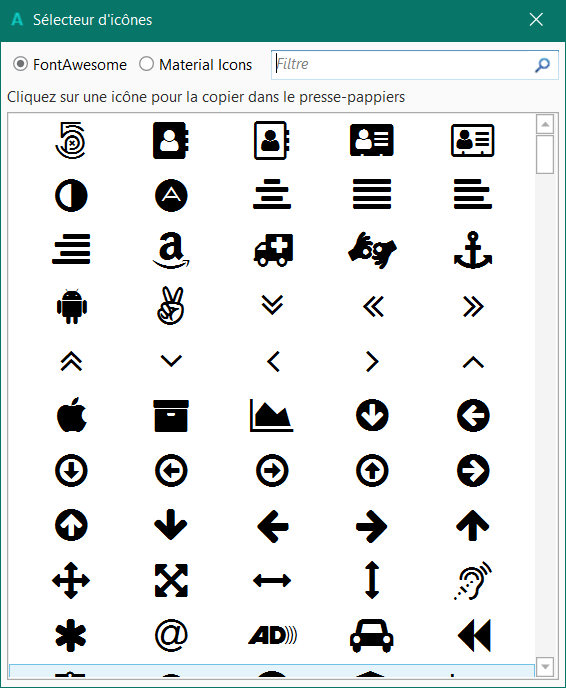
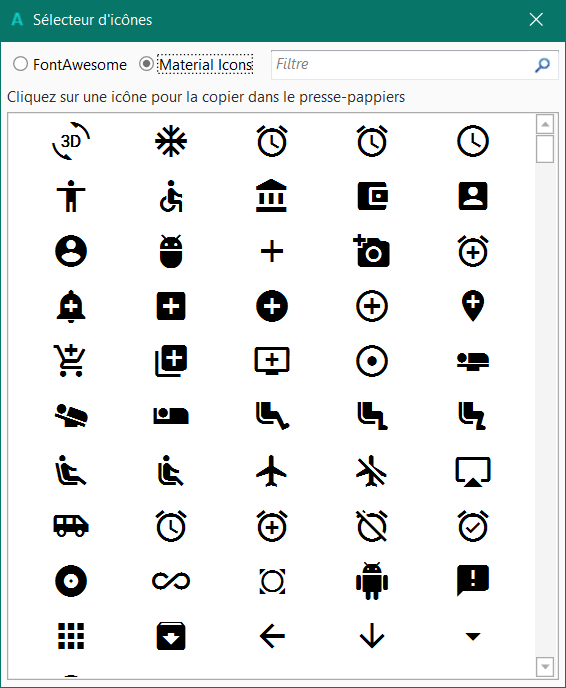
Ensuite vous pouvez copier la valeur dans le code.

## Sélecteur d’icônes



Cliquez avec le bouton droit de la souris dans la zone code pour afficher le menu contextuel ci-contre et cliquez sur .

Vous pouvez choisir entre des icônes *Font Awesome* ou *Material*.

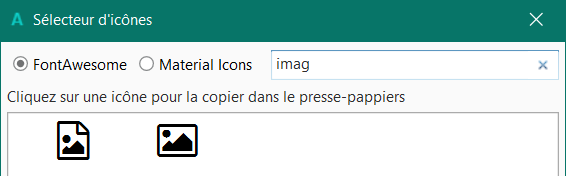
Icônes *Font Awesome*. Icônes *Material*.

Cliquez sur une icône pour la copier dans le presse-papier.

Ensuite vous pouvez la copier dans le code comme ci-dessous.

L’icône est représentée par son code caractère, Chr(0xE632).

lblResult.Text = Chr(0xE632)

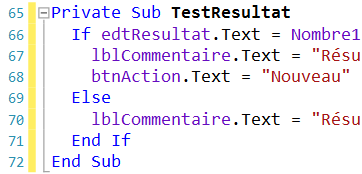


Vous pouvez filtrer les icônes.

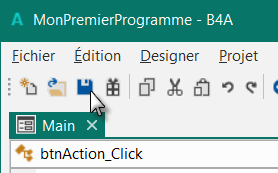
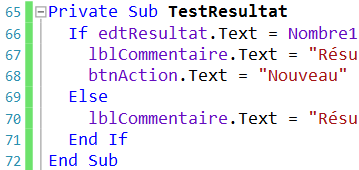
## Couleurs dans la marge gauche

Vous verrez régulièrement des lignes verticales jaunes ou vertes dans la marge gauche de l’éditeur.

Dès que vous ajoutez ou modifiez une ligne, celle-ci sera marquée en jaune juste à côté de numéro de la ligne et signifie que cette ligne a été modifiée.



Si nous cliquons sur  pour enregistrer le projet, les lignes jaunes passent en vert indiquant que le code a été modifié mais déjà enregistré. Vous pouvez aussi presser Ctrl + S pour enregistrer le projet.

Lorsque nous quittons l’éditeur et chargeons le programme à nouveau les lignes vertes disparaissent.

## URLs dans commentaires Ctrl + clic

Des URLs dans des commentaires ou des objets String peuvent être exécutés avec Ctrl et un clic.

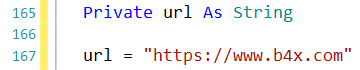
Dans un commentaire :



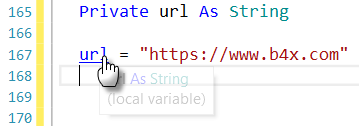
Si le curseur se trouve sur la ligne et que vous pressez Ctrl l’url est mis en évidence en bleu et si vous cliquez sur l’url il sera exécuté. Passer avec le curseur au-dessus d’un url tout en maintenant la touche Ctrl pressée, l’url sera aussi mise en évidence en bleu.



Dans un objet String :

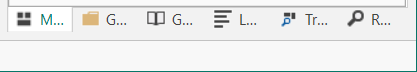


Le curseur doit se trouver au-dessus de la variable de l’objet String et non sur le texte.



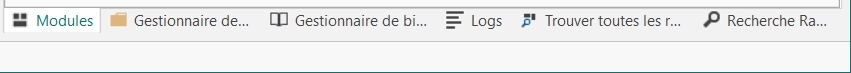
# Onglets

Il y 6 onglets dans le coin inférieur droit de l’EDI affichant différentes fenêtres.



La version courte.

La version longue.



Les 6 onglets sont :

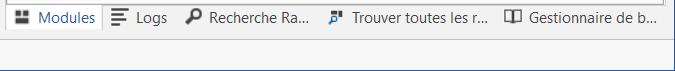
* Modules
* Gestionnaire de fichiers
* Gestionnaire des bibliothèques
* Logs
* Trouver toutes les références
* Recherche Rapide

Chaque onglet a sa propre fenêtre.

Par défaut elles sont affichées dans la zone des onglets dans la partie droite de l’EDI, seulement une fenêtre à la fois.

Ces fenêtres peuvent être fermées, cachées ou flottantes, voir le chapitre suivant.

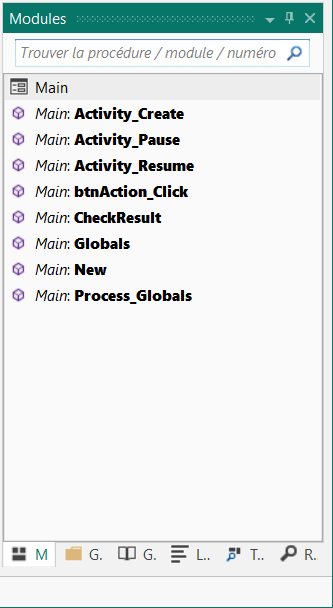
**B4R**



Dans B4R il n’y a que 5 onglets, l’onglet Gestionnaire de fichiers n’existe pas.

## Fenêtres onglets flottantes

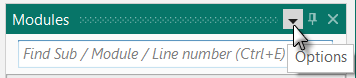
Lorsque vous lancez l’EDI d’origine, toutes le fenêtres d’onglet sont amarrés dans la zone onglets.



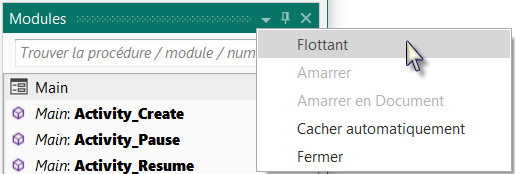
Vous pouvez définir chaque fenêtre comme fenêtre flottante.

## Flottante

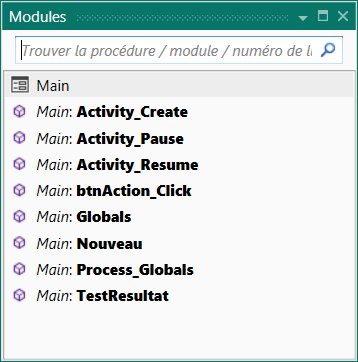
Pour rendre la fenêtre Modules flottante, cliquez dans le titre sur .



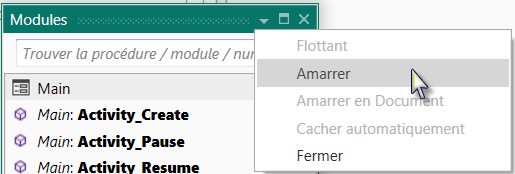
Et cliquez sur Flottant .



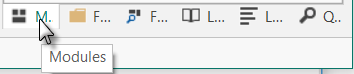
La fenêtre Modules est maintenant flottante, vous pouvez la placer n’importe où sur l’écran même sur un deuxième écran.



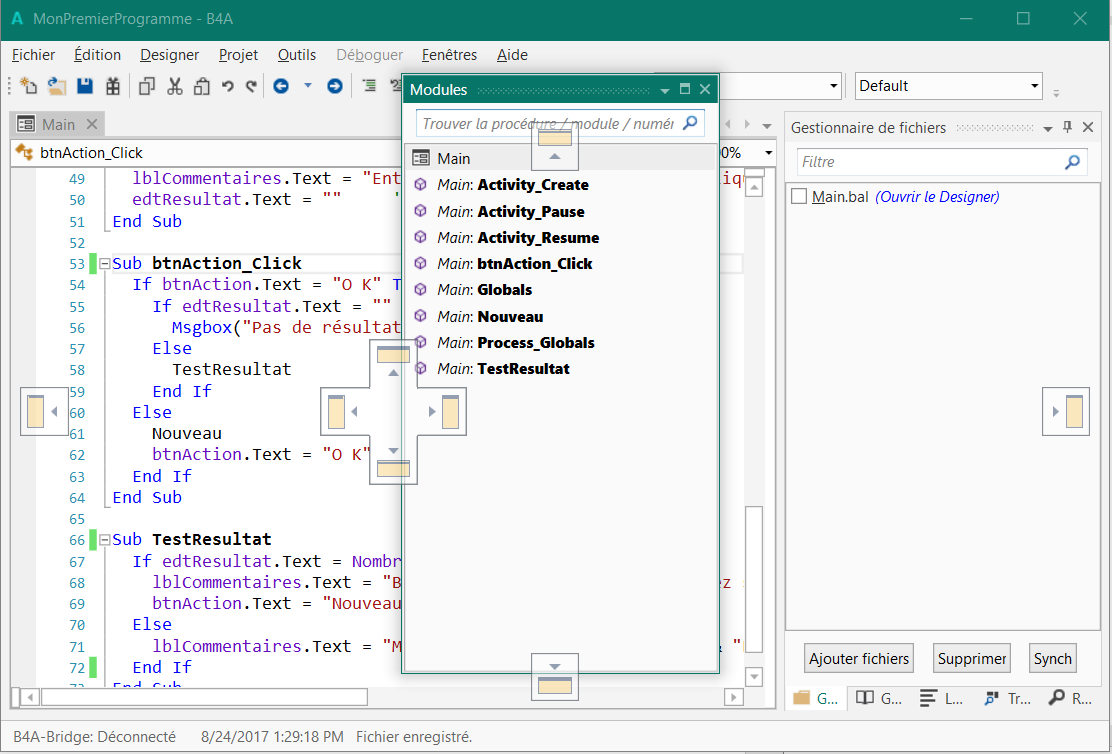
Pour l’amarrer à nouveau, cliquez sur .



Vous pouvez aussi cliquer sur l’onglet et le déplacer tout en maintenant la touche de la souris enfoncée.



Ceci vous montre les zones d’amarrage possibles.



Zones d’amarrage :



En haut



A gauche

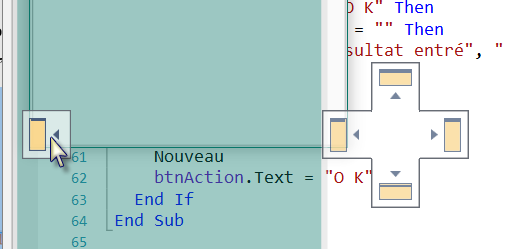


A droite

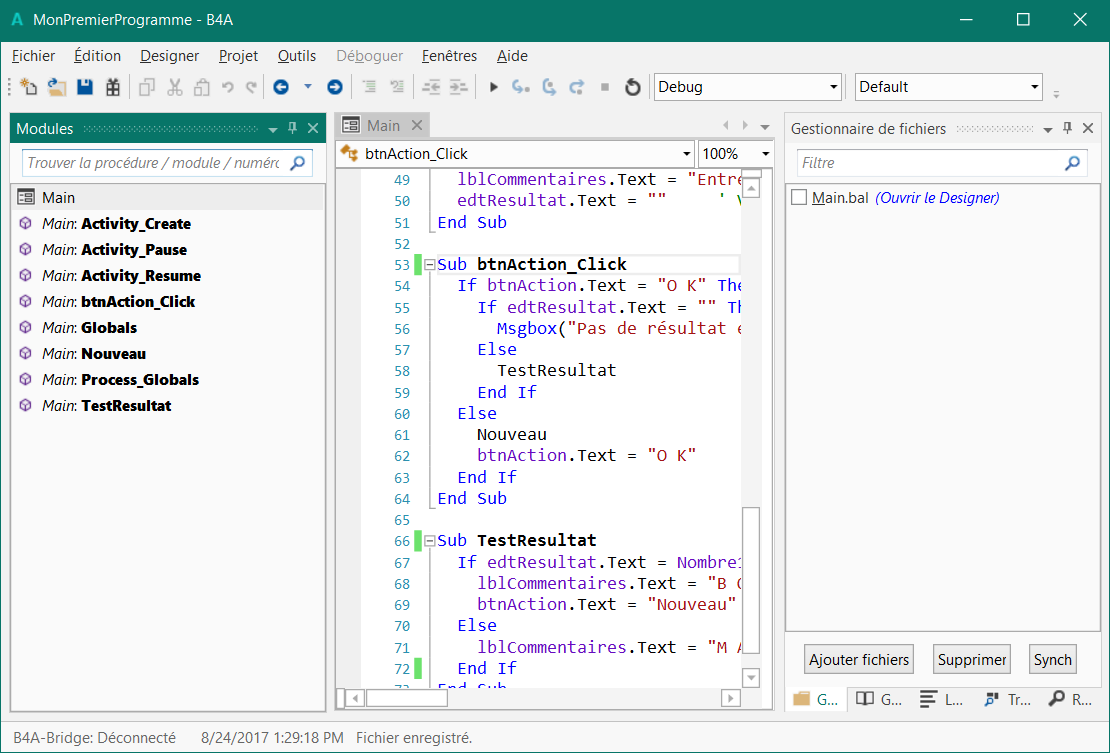


En bas

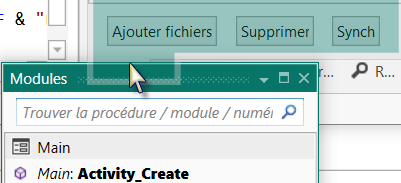
Lorsque vous déplacez la souris sur un des symboles de zone d’amarrage la fenêtre se trouvera soit en haut, à gauche, à droite ou en bas.



Et le résultat.

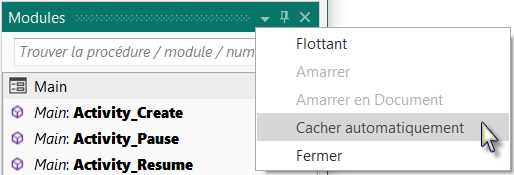


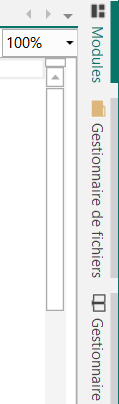
Pour replacer la fenêtre à sa position d’origine, cliquez sur le titre de la fenêtre et déplacez la vers les onglets.



## Cacher automatiquement

Cliquez sur  dans le titre ou cliquez sur  dans les options.

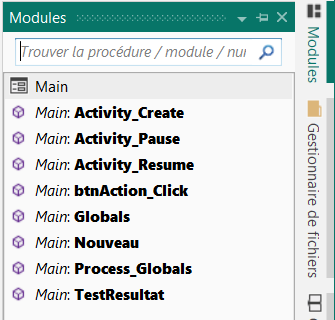




Les onglets se déplacent du bas de l’écran, verticalement, vers la droite de l’écran et la fenêtre onglet est cachée.

Passer au-dessus d’un onglet le met en évidence en vert.

Cliquez sur l’onglet pour montrer sa fenêtre.

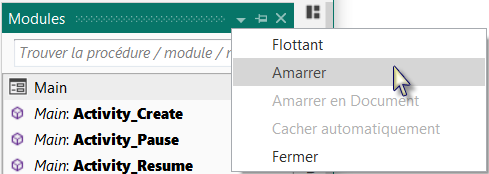


La fenêtre de l’onglet sélectionné est affichée.

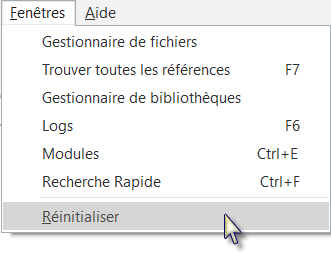
Dès que vous cliquez sur quelque chose dans la zone code le fenêtre ongle sera à nouveau cachée.

Pour déplacer un onglet à nouveau dans le coin inférieur droit :

Cliquez sur  dans les Options.



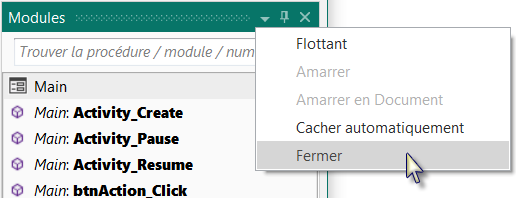
Ou cliquez sur  dans le menu Fenêtres de l’EDI.

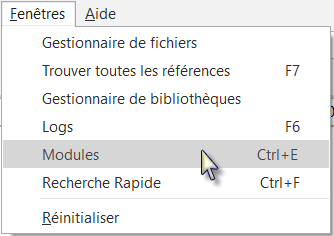


## Fermer

Vous pouvez fermer une fenêtre, la cacher.

Cliquez sur  dans le titre ou sur  dans les Options.

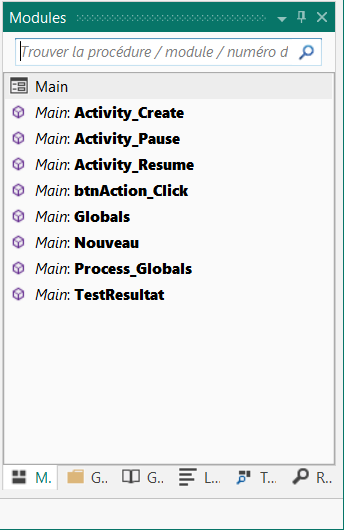




Pour l’afficher à nouveau, cliquez, dans le menu Fenêtres, sur le nom de l’onglet que vous voulez afficher à nouveau,  dans notre exemple.

## Liste de Modules et routine

Tous les modules du projet et toutes les routines du module actif sont affichés dans la fenêtre de l’onglet Modules. La hauteur de l’image ci-dessous a été réduite.



[Trouver la procédure / module / numéro de ligne (Ctrl + E)](#_Find_Sub_/)

Liste des modules sur le haut.

Cliquez sur un module pour afficher son code.

Trouver la procédure… (Ctrl + E) voir page suivante

Liste des routines du module sélectionné.

Cliquez sur une routine pour l’afficher au milieu de la zone code.

Pour afficher un module caché, cliquez sur son nom dans la liste.

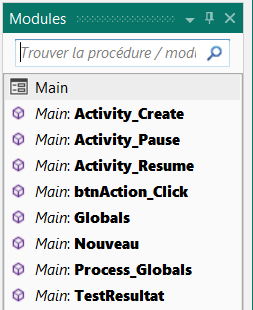
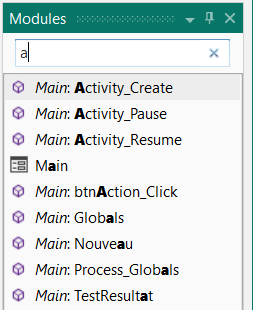
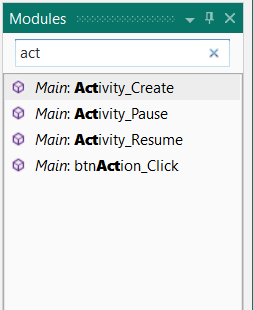
### Trouver la procédure / module / numéro de ligne (Ctrl + E)

La fonction *Trouver la procédure / module / numéro de ligne* est un moteur de recherche, sur le haut de la fenêtre de l’onglet Modules, pour trouver des procédures ou des modules avec un nom ou une partie de nom donné.

Vous pouvez presser Ctrl + E dans le code pour sélectionner l’onglet Modules avec la fonction *Trouver la procédure / module / numéro de ligne*.

Exemple avec le code du projet MonPremierProgramme.

Pas de texte seulement un caractère ‘a’ texte ‘act’

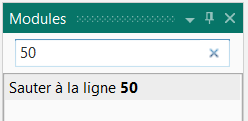
  

Montre tous les modules et Montre tous les modules et Montre tous les modules et

routines du Module actif. routines contenant ‘a’. routines contenant ‘act’.

Cliquez sur un des éléments dans la liste pour afficher son code.

Pour afficher un numéro de ligne au milieu de la zone code, entrez le numéro :



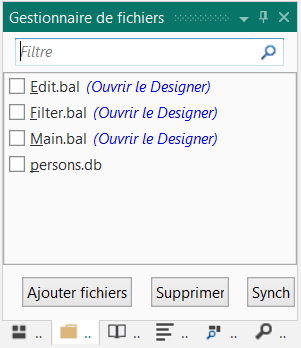
Puis pressez Entrée ou cliquez sur .

## Gestionnaire de fichiers B4A, B4i et B4J

Cette fenêtre affiche tous les fichiers qui ont été ajoutés au projet.

Ces fichiers sont enregistrés dans le dossier ‘Files’ du projet.

Ces fichiers peuvent être de n’importe quel type : layouts, images, textes, etc.

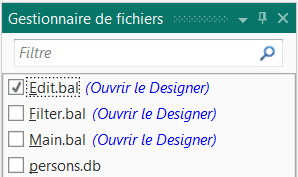


Pour des fichiers layout, vous pouvez cliquer sur pour ouvrir le Designer avec le fichier sélectionné.

Cliquez sur  pour ajouter des fichiers à la liste.

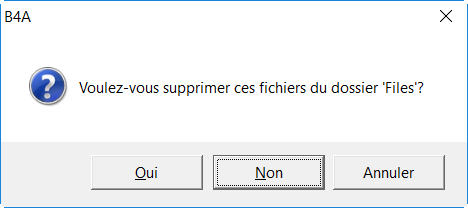
Les fichiers dans ce dossier sont accessibles depuis le code avec comme référence de dossier File.DirAssets.

Ou cliquez sur  pour ajouter tous les fichiers du dossier ‘Files’ du projet dans l’onglet.



Pour supprimer des fichiers, cochez-les, et cliquez sur le bouton .

Cette fonction supprime les fichiers cochés de la liste et, si vous le désirez, aussi du dossier ‘Files’ du projet.



Le programme vous demande si vous voulez supprimer les fichiers cochés du dossier 'Files' du projet.

**Assurez-vous d’avoir une copie des fichiers que vous supprimez, car ils sont supprimés du dossier ‘Files’ mais pas transférés dans la corbeille, ce qui veut dire qu’ils sont définitivement perdus sans copie.**

Dans le haut du Gestionnaire de fichier vous pouvez filtrer la liste des fichiers.



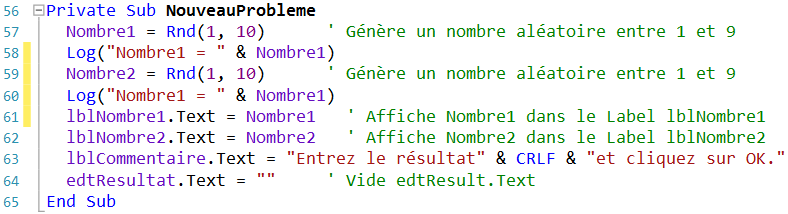
Entrez ‘bal’ pour filtrer les fichiers layout.

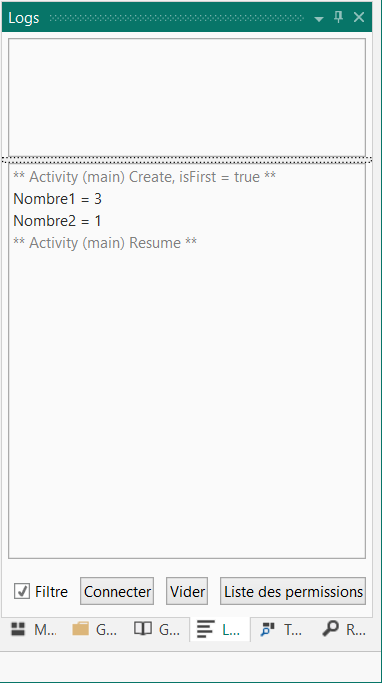
## Logs

La fenêtre onglet ‘Logs’ affiche des commentaires générés par le programme durant son exécution.

Dans le projet MonPremierProgramme, nous ajoutons deux lignes dans la routine Nouveau.

Les numéros des lignes peuvent différer des vôtres.



Lancez le programme.

Cliquez sur  si nécessaire pour connecter l’enregistreur (logger).

Dans la partie supérieure des [Avertissements de compilation](#_Compile_/_Warnings) sont affichés, voir chapitre suivant.

Dans la partie inférieure sont affichés des informations sur l’évolution de l’exécution du programme.

\*\* Activity (main) Create, isFirst = true \*\*

Nombre1 = 3 Premier message Log

Nombre2 = 1 Deuxième message Log

\*\* Activity (main) Resume \*\*

Lorsque *Filtre* est coché, vous ne voyez que les messages relatifs à votre programme. Lorsque c’est non coché, vous voyez tous les messages qui circulent dans le système.

Si vous rencontrez une erreur, mais sans message relatif, il est utile de décocher l’option de filtre pour voir s’il y des messages d’erreur.

Cliquez sur  pour vider la fenêtre Logs.

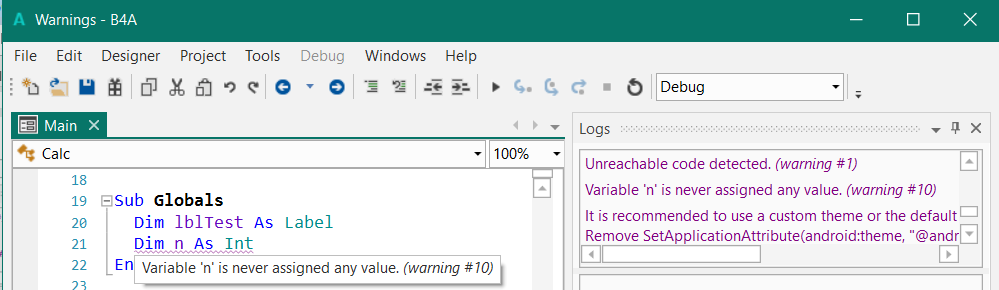
#### 

### Avertissements de compilation

B4x contient un moteur d’avertissement. L'objectif du moteur d'avertissement est de trouver des erreurs de programmation potentielles le plus tôt possible.

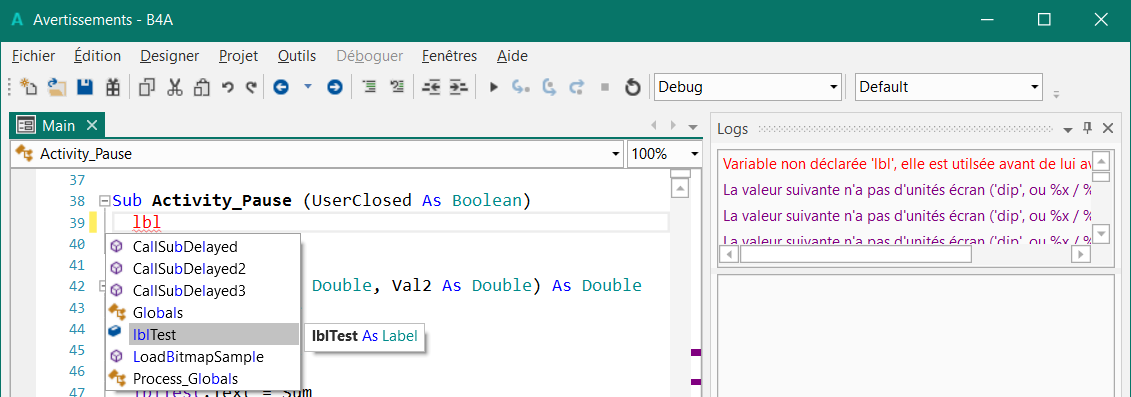
Les exemples utilisent le projet Avertissements.

Les avertissements de compilation sont affichés dans le haut de le fenêtre ‘Logs’ et dans le code lorsqu’un passe au-dessus de la ligne de code avec le curseur.

Les lignes de code générant un avertissement sont soulignées .  
  


Cliquez sur la ligne d’avertissement pour afficher la ligne de code concernée au milieu de l’éditeur.

Le moteur d’avertissement est actif en tout temps et dès que vous écrivez.



Écrivez, par exemple, ‘lbl’ au début d’une ligne et vous aurez :

* lbl en rouge, cas lbl n’est pas déclaré.
* Un avertissement 
* La fenêtre de complétion automatique avec les suggestions d’objets et variables contenant les caractères écrits.

#### Ignorer des avertissements de compilation

En tant que développeur, vous pouvez choisir d’ignorer des avertissements. Ajoutez un commentaire ‘ignore sur la ligne pour ignorer l’avertissement :

Vous pouvez aussi ignorer tous les avertissements d’un type donné en ajoutant l’attribut #IgnoreWarnings: dans la région des attributs du module concerné.

Par exemple, ignorer les avertissements #10 et #12 :

#Region Activity Attributes

#FullScreen: False

#IncludeTitle: True

**#IgnoreWarnings: 10, 12**

#End Region

Vous trouvez les numéros des avertissements à la fin de chaque ligne d’avertissement.

#### Liste des avertissements

Les avertissements peuvent être différents selon le produit.

1: Code inaccessible détecté.  
2: Toutes les parties du code ne renvoient une valeur.  
3: Le type (dans la définition de la Sub) doit être défini explicitement.  
4: Valeur de renvoi absente. Une valeur par défaut sera utilisée.  
5: Déclaration de type de variable absent. Un objet string sera utilisé.  
6: La valeur suivante n'a pas d'unités écran ('dip', ou %x / %y): {0}.  
7: L'objet a été converti en une variable 'String'. C'est sûrement une erreur de programmation.  
8: Variable non déclarée '{0}'  
9: Variable pas utilisée '{0}'.  
10: Aucune valeur n'a été attribuée à la variable '{0}'.  
11: La variable '{0}' n'a pas été initialisée.  
12: La Sub '{0}' n'est pas utilisée.  
13: La variable '{0}' doit être déclarée dans la Sub Process\_Globals.  
14: Le fichier '{0}' dans le dossier 'Files' n'a pas été ajouté dans le Gestionnaire de fichiers.  
Vous devez soit l'effacer ou l'ajouter au projet.  
Vous pouvez utiliser Outils - Nettoyer le dossier de fichiers (fichiers inutiles)  
15: Le fichier '{0}' n'est pas utilisé.  
16: Le fichier layout '{0}' n'est pas utilisé. Est-ce qu'il manque un appel Activity.LoadLayout?  
17: Le fichier '{0}' est absent dans l'onglet Gestionnaire de fichiers.  
18: Les valeurs de TextSize ne doivent pas être mises à l'échelle, car elles le sont automatiquement.  
19: Bloc 'Catch' vide. Vous devez ajouter au moins Log(LastException.Message).  
20: La 'View' {0} a été définie dans le Designer. Vous ne devez pas l'initialiser dans le code.  
21: Impossible d'accéder aux dimensions de la 'View' avant qu'elle n'ait été ajoutée à son parent.  
22: Les types ne correspondent pas.  
23: Des dialogues modaux ne sont pas autorisés dans Sub Activity\_Pause. Ils sont ignorés.  
24: Accéder à des champs d'autres modules dans Sub Process\_Global peut être dangereux car l'ordre d'initialisation n'est pas déterministe.

25: Sub '{0}' pas trouvée.

26: Ajoutez android:targetSdkVersion="19" dans l'éditeur manifeste (après minSdkVersion).

27 : AndroidManifest.xml est en lecture seule ou si l'option Ne pas écraser le fichier manifeste est cochée. Utilisez plutôt l'éditeur manifeste.

28: Il est recommandé d'utiliser un thème personnalisé ou le thème par défaut.  
Supprimez SetApplicationAttribute (android: theme, "@android: style / Theme.Holo") dans l'éditeur manifeste.

32: La bibliothèque '{0}' n'est pas utilisée.  
33: DoEvents est obsolète. Cela peut entraîner des problèmes de stabilité. Utilisez Sleep (0) à la place (si c'est vraiment nécessaire).

Avertissements lors de l’exécution  
1001: Panel.LoadLayout ne doit être appelé que lorsque le 'Panel' a été ajouté à son parent.  
1002: Le même objet a été ajouté à la liste. Vous devriez appeler Dim encore une fois pour créer un nouvel objet.  
1003: L'objet a déjà été initialisé.  
1004: Les propriétés FullScreen (plein écran) ou IncludeTitle (inclure titre) dans le fichier layout ne correspondent pas aux paramètres des attributs d'activité.

1005: Le fichier layout '{0}' n'est pas utilisé. Est-ce qu'Il manque un appel de Page.RootPanel.LoadLayout?

1006: La valeur suivante manque d'unités d'écran ('dip'): {0}.

1007: Le fichier layout '{0}' n'est pas utilisé. Est-ce qu'Il manque un appel de MainForm.RootPane.LoadLayout?

1008: Les noms de fichier sont sensibles à la casse. La casse est incorrecte.

**1: Code inaccessible détecté.**

Il y a du code qui ne sera jamais exécuté.

Ceci peut arriver, dans une routine, si vous avez du code après le mot clé Return, qui sort de la routine.

**2: Toutes les parties du code ne renvoient une valeur.**

Sub **Calc**(Val1 As Double, Val2 As Double, Operation As String) As Double

Select Operation

Case "Add"

Return (Val1 + Val2)

Case "Sub"

Return (Val1 - Val2)

Case "Mult"

Return (Val1 \* Val2)

Case "Div"

End Select

End Sub

Dans Case "Div" aucune valeur n’est renvoyée !

Autre exemple :

Mauvais code.

Sub **Activity\_KeyPress**(KeyCode As Int) As Boolean

Private Answ As Int

Private Txt As String

If KeyCode = KeyCodes.KEYCODE\_BACK Then' Vérifie si le KeyCode est BackKey

Txt = "Voulez-vous vraiment qutter le programme ?"

Answ = Msgbox2(Txt,"A T T E N T I O N","Oui","","Non",Null) ' MessageBox

If Answ = DialogResponse.POSITIVE Then ' Si la valeur renvoyée est Oui alors

Return False ' Return = False l’événement ne sera pas consommé

Else ' nous quittons le programme

Return True ' Return = True l’événement ne sera pas consommé

End If ' pour ne pas quitter le programme

End If

End Sub

Code correct.

Sub **Activity\_KeyPress**(KeyCode As Int) As Boolean

Private Answ As Int

Private Txt As String

If KeyCode = KeyCodes.KEYCODE\_BACK Then' Vérifie si le KeyCode est BackKey

Txt = " Voulez-vous vraiment qutter le programme ?"

Answ = Msgbox2(Txt,"A T T E N T I O N","Oui","","Non",Null) ' MessageBox

If Answ = DialogResponse.POSITIVE Then ' Si la valeur renvoyée est Oui alors

Return False ' Return = False the Event will not be consumed

Else ' nous quittons le programme

Return True ' Return = True l’événement ne sera pas consommé

End If ' pour ne pas quitter le programme

Else

Return True ' Return = True l’événement ne sera pas consommé

End If ' pour ne pas quitter le programme

End Sub

**3: Le type (dans la définition de la Sub) doit être défini explicitement.**

Mauvais code.

Sub **Calc**(Val1 As Double, Val2 As Double, Operation As String)

Code correct.

Sub **Calc**(Val1 As Double, Val2 As Double, Operation As String) As Double

Le type de renvoi doit être spécifié !

**4: Valeur de renvoi absente. Une valeur par défaut sera utilisée.**

Mauvais code.

Sub **CalcSum**(Val1 As Double, Val2 As Double) As Double

Private Sum As Double

Sum = Val1 + Val2

Return

End Sub

Code correct.

Sub **CalcSum**(Val1 As Double, Val2 As Double) As Double

Private Sum As Double

Sum = Val1 + Val2

Return Sum

End Sub

**5: Déclaration de type de variable absent. Un string sera utilisé.**

Mauvais code.

Sub **Calc**(Val1, Val2 As Double, Operation As String) As Double

Code correct.

Sub **Calc**(Val1 As Double, Val2 As Double, Operation As String) As Double

Dans les déclarations de routines, chaque variable doit avoir sa propre déclaration de type.

Dans les déclarations utilisant les mots clé Private, Public ou Dim c’est autorisé.

Dans la ligne ci-dessous, les deux variables sont du type Double :

Private Val1, Val2 As Double

**6: La valeur suivante n'a pas d'unités écran ('dip', ou %x / %y): {0}.**

Mauvais code.

Activity.AddView(lblTest, 10, 10, 150, 50)

Code correct.

Activity.AddView(lblTest, 10dip, 10dip, 150dip, 50dip)

Dans l’exemple ci-dessus vous aurez quatre avertissements, un pour chaque valeur.

Pour les dimensions de Views vous devez utiliser des valeurs dip, %x ou %y.

**7: L'objet a été converti en une variable 'String'.**

**C'est sûrement une erreur de programmation.**

**8: Variable non déclarée '{0}.**

Mauvais code.

Sub **SetHeight**

h = 10dip

End Sub

Code correct.

Sub **SetHeight**

Private h As Int

h = 10dip

End Sub

La variable h n’a pas été déclarée. Vous le voyez aussi à sa couleur rouge.

**9: Variable pas utilisée '{0}'.**

Sub **SetHeight**

Private h As Int

h = 10dip

End Sub

Cet avertissement indique que la variable h n’est pas utilisée.

Elle est déclarée et on lui a attribué une valeur, mais elle n’est pas utilisée !

Ce code ne provoque pas d’avertissement car la variable est utilisée :

Sub **SetHeight**

Private h As Int

h = 10dip

lblTest.Height = h

End Sub

**10: Aucune valeur n'a été attribuée à la variable '{0}'.**

Sub **Test**

Private h As Int

End Sub

Cet avertissement indique que la variable h est déclarée, mais qu’aucune valeur ne lui a été attribuée.

Code correct, voir ci-dessus.

**11: La variable '{0}' n'a pas été initialisée.**

Mauvais code.

Private lst As List

lst.Add("Test1")

Code correct.

Private lst As List

lst.Initialize

lst.Add("Test1")

Des objets tel List ou Map doivent être initialisées avant de pouvoir les utiliser.

Les Views ajoutées dans le code doivent aussi être initialisées avant de pouvoir être ajoutées à leurs objets parent.

**12: La Sub '{0}' n'est pas utilisée.**

Cet avertissement est affiché si une routine n’est utilisée nulle part.

**13: La variable '{0}' doit être déclarée dans la Sub Process\_Globals.**

Mauvais code.

Sub **Globals**

Public Timer1 As Timer

Public GPS1 As GPS

Code correct.

Sub **Process\_Globals**

Public Timer1 As Timer

Public GPS1 As GPS

Certains objets tel Timer ou GPS doivent être déclarés dans la routine Process\_Globals, pas dans Globals.

**14: Le fichier '{0}' dans le dossier 'Files' n'a pas été ajouté dans l’onglet Gestionnaire de fichiers. Vous devez soit l'effacer ou l'ajouter au projet.**

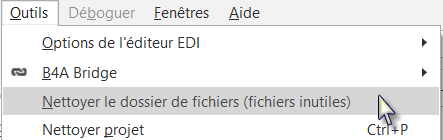
Vous utilisez un fichier se trouvant dans le dossier Files du projet, mais n’a pas été ajouté dans l’onglet Gestionnaire de fichiers.

Vous devez soit :

- Le supprimer du dossier Files. N’oubliez pas de le sauvegarder au part avant.

- L’ajouter dans l’onglet Gestionnaire de fichiers.

- Utiliser la fonction *Nettoyer le dossier de fichiers (fichiers inutiles)* dans le menu Outils.



**15: Le fichier '{0}' n'est pas utilisé.**

Vous avez des fichiers dans le dossier Files du projet qui ne sont pas utilisés.

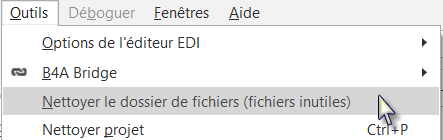
Vous devez les supprimer, ou nettoyer le dossier Files, voir ci-dessus.

**16: Le fichier layout '{0}' n'est pas utilisé. Est-ce qu'il manque un appel Activity.LoadLayout?**

Vous aveZ un fichier layout dans le dossier Files qui n’est pas utilisé.

Vous pouvez ajouter une instruction LoadLayout ou le supprimer du dossier Files.

Ou nettoyer le dossier.



**17: Le fichier '{0}' est absent dans l'onglet Gestionnaire de fichiers.**

Le ficher spécifié figure dans l’onglet Gestionnaire de fichiers mais ne se trouve pas dans le dossier Files du projet. Vous devez l’ajouter.

Voir la chapitre [Gestionnaire de fichiers](#_Files_1).

**18: Les valeurs de TextSize ne doivent pas être mises à l'échelle, car elles le sont automatiquement.**

Mauvais code.

lblTest.TextSize = 16dip

Code correct.

lblTest.TextSize = 16

Les valeurs de la propriété TextSize (taille du texte) sont des valeurs indépendantes des pixels et de la densité. Leur unité est le [point typographique](https://fr.wikipedia.org/wiki/Point_(unit%C3%A9)), une unité typographique, qui doit avoir une valeur absolue et non des valeurs *dip*.

**19: Bloc 'Catch' vide. Vous devez ajouter au moins Log(LastException.Message).**

Mauvais code.

Try

imvImage.Bitmap = LoadBitmap(File.DirRootExternal, "image.jpg")

Catch

End Try

Code correct.

Try

imvImage.Bitmap = LoadBitmap(File.DirRootExternal, "image.jpg")

Catch

Log(LastException.Message)

End Try

Il est recommandé d’ajouter au moins l’instruction Log(LastException.Message) dans le bloc Catch au lieu de le laisser vide.

**20: La 'View' {0} a été définie dans le Designer. Vous ne devez pas l'initialiser dans le code.**

Une View définie avec le Designer dans un fichier layout ne doit pas être initialisé !

Seules des Views ajoutées dans le code doivent être initialisées.

**21: Impossible d'accéder aux dimensions de la 'View' avant qu'elle n'ait été ajoutée à son parent.**

Vous devez d’abord ajouter une View à son objet parent avant de pouvoir accéder à ses dimensions.

Lorsque vous ajoutez des Views dans le code leurs dimensions sont définies lors de l’ajout avec la fonction AddView.

**22: Les types ne correspondent pas.**

**23: Des dialogues modaux ne sont pas autorisés dans Sub Activity\_Pause. Ils sont ignorés.**

Des dialogues modaux tel MsgBox ne sont pas autorisés dans la routine Activity\_Pause.

**24: Accéder à des champs d'autres modules dans Sub Process\_Global peut être dangereux car l'ordre d'initialisation n'est pas déterministe.**

**25: Sub '{0}' pas trouvée.**

La routine spécifiée n’a pas été trouvée.

**26: Ajoutez android:targetSdkVersion="19" dans l'éditeur manifeste (après minSdkVersion).**

<uses-sdk android:minSdkVersion="5" android:targetSdkVersion="19"/>

Au lieu de :

<uses-sdk android:minSdkVersion="5""/>

**27: AndroidManifest.xml est en lecture seule ou si l'option Ne pas écraser le fichier manifeste est cochée. Utilisez plutôt l'éditeur manifeste.**

**28: Il est recommandé d'utiliser un thème personnalisé ou le thème par défaut.  
Supprimez SetApplicationAttribute (android: theme, "@android: style / Theme.Holo") dans l'éditeur manifeste**.

Ceci était ajouté automatiquement dans des versions anciennes de B4A, mais n’est plus nécessaire.

**32: La bibliothèque '{0}' n'est pas utilisée.**

Supprimez les bibliothèques inutilisées.

**33: DoEvents est obsolète. Cela peut entraîner des problèmes de stabilité. Utilisez Sleep (0) à la place (si c'est vraiment nécessaire).**

## Gestionnaire de bibliothèques

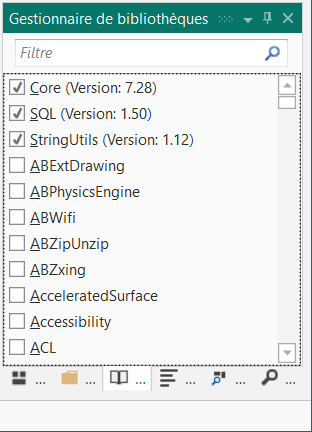
La fenêtre du ‘Gestionnaire de bibliothèques’ contient la liste des bibliothèques disponibles pouvant être utilisées dans le projet.

Les bibliothèques dans la liste dépendent du produit B4x utilisé.

Les images correspondent à une exemple B4A.

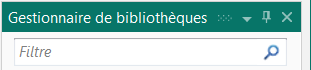
Cochez les bibliothèques dont vous avez besoin.

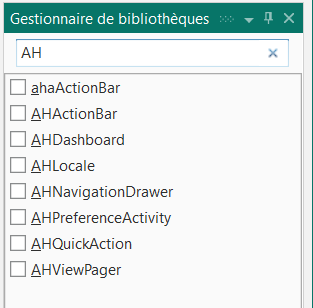
Assurez-vous que vous avez les dernières versions des bibliothèques.



Les bibliothèques utilisées sont affichées dans le haut de la liste.

Dès que vous sélectionnez une bibliothèque elle est déplacée vers le haut.

Sur le haut, vous trouvez un champ permettant de filtrer la liste de bibliothèques.



Entrez ‘AH’ et vous trouvez toutes les bibliothèques commençant par AH.

Une liste des bibliothèques additionnelles peut être trouvé ici :

[B4A](https://www.b4x.com/android/documentation.html), [B4i](https://www.b4x.com/b4i/documentation.html), [B4J](https://www.b4x.com/b4j/documentation.html), [B4R](https://www.b4x.com/b4r/documentation.html)

Les liens ci-dessus montrent la page de documentation du site.

Les bibliothèques sont expliquées en détail dans le livret

B4x Langage Basic.

## Recherche rapide

La Recherche Rapide permet de rechercher n’importe quel texte dans le code de tout le projet.

Exemples avec le code de MonPremierProgramme.

Il y a plusieurs possibilités pour sélectionner la fonction Recherche Rapide :

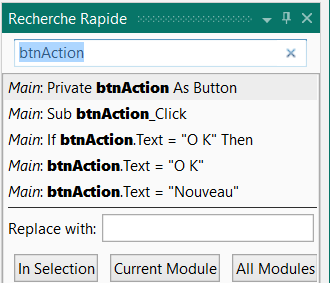
* Ctrl + F, le moyen le plus simple et efficace.
* Cliquer sur l’ongletdans le coin inférieur droit de l’EDI.
* Cliquer sur  dans le menu Edition.

Exemple :



Dans le code, double cliquez sur btnAction pour sélectionner le texte et pressez Ctrl + F.

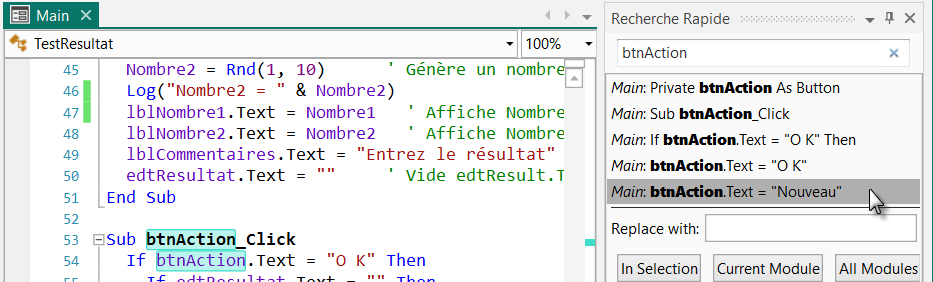
Vous obtenez la fenêtre onglet ci-dessous.

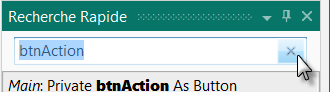


La liste contient toutes les occurrences du texte sélectionné dans tous les modules

Dans chaque ligne vous trouvez le nom du module et le contenu de la ligne.

Un clic sur une ligne déplace le curseur directement dans la ligne code concernée.



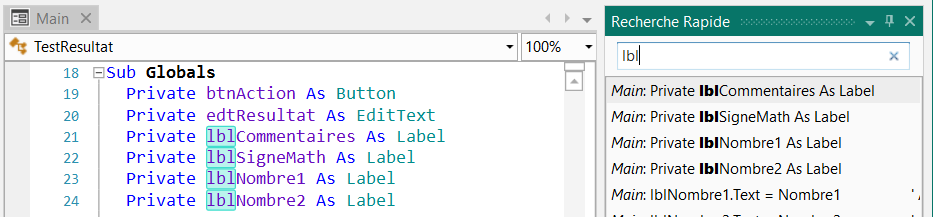


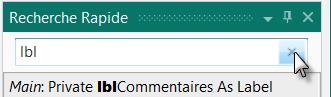
Pour supprimer la recherche cliquez sur  dans le coin supérieur droit de la fenêtre Recherche Rapide.

Vous pouvez aussi entrer n’importe quel texte dans le champ de recherche :

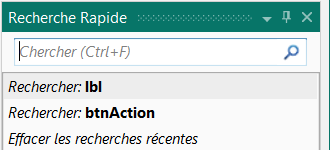
Par exemple, entrez *lbl* et vous obtenez la liste des lignes contenant le texte, *lbl* dans notre cas.

Le texte de recherche est mis en évidence En cliquant sur une des lignes, celle-ci dans les lignes contenant ce texte. est affichée au milieu de l’EDI.

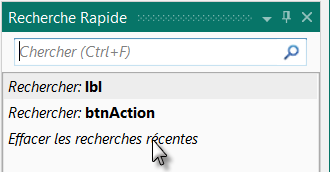




Cliquez sur  pour supprimer la recherche.



La liste des dernières recherches est affichée.

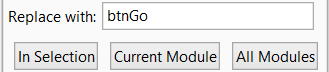


Cliquez sur  pour supprimer les dernières recherches.

Des éléments sont ajoutés à la liste des dernières recherches lorsque :  
1. Vous sélectionnez un des résultats ou pressez Entrée qui sélectionnera le premier résultat.  
2. Vous sélectionnez un texte dans le code et pressez Ctrl + F pour effectuer la recherche.

Vous pouvez remplacer du texte soit dans le code sélectionné , dans le module actif  ou dans tous les modules .

Entrez le nouveau texte et cliquez sur un de ces boutons ,  ou .



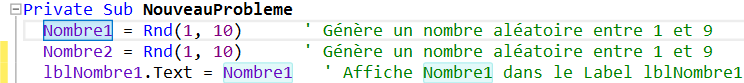
## Trouver toutes les références (F7)

Ceci est un moteur de recherche pour trouver toutes les références d’un objet (view, variable) donné.

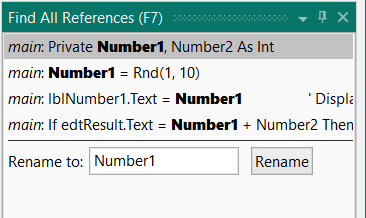
Cliquez sur l’onglet  ou pressez F7 pour obtenir la fenêtre ci-dessous montrant une liste avec toutes les lignes de code contenant le texte sélectionné.

Exemple avec le code du projet MonPremierProgramme.

Sélectionnez dans le code, dans la ligne, Nombre1.



Cliquez sur  ou pressez F7 et vous obtenez la liste ci-dessous avec toutes les lignes contenant l’objet sélectionné.



Un clic sur une des lignes de la liste affiche cette ligne au centre de la zone code de l’EDI.

Vous pouvez changer le nom de l’objet sélectionné.

Entrez le nouveau nom et cliquez sut .

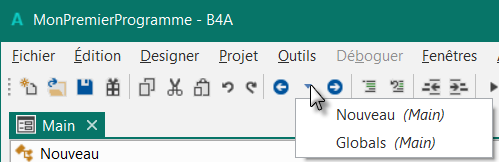
# Navigation dans l’EDI

Conseils donnés par Erel dans le forum.

## Alt + Gauche / Alt + Droite

Vous permet de vous déplacer en avant ou en arrière dans le code vers les derniers endroits où vous étiez qui sont mémorisés dans la pile de navigation (voir ci-dessous). Ceci peut être utile pour naviguer entre les dernières routines où vous étiez.

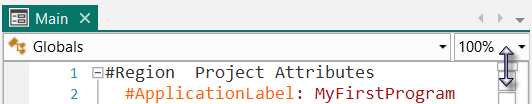
## Alt + N Menu de la pile de Navigation

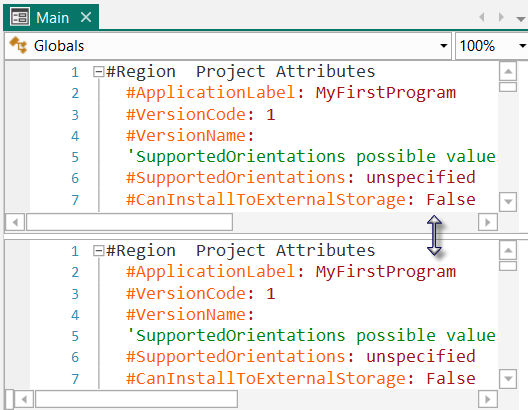
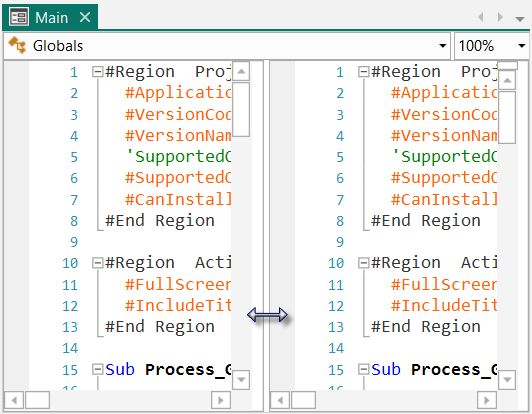


Alt + N ouvre le menu de la pile de Navigation. Vous pouvez sélectionner l’endroit où vous voulez aller.

## Diviser l’écran

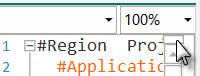
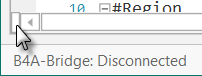
Si vous travaillez à deux endroits différents dans le même module vous pouvez diviser la zone de code :

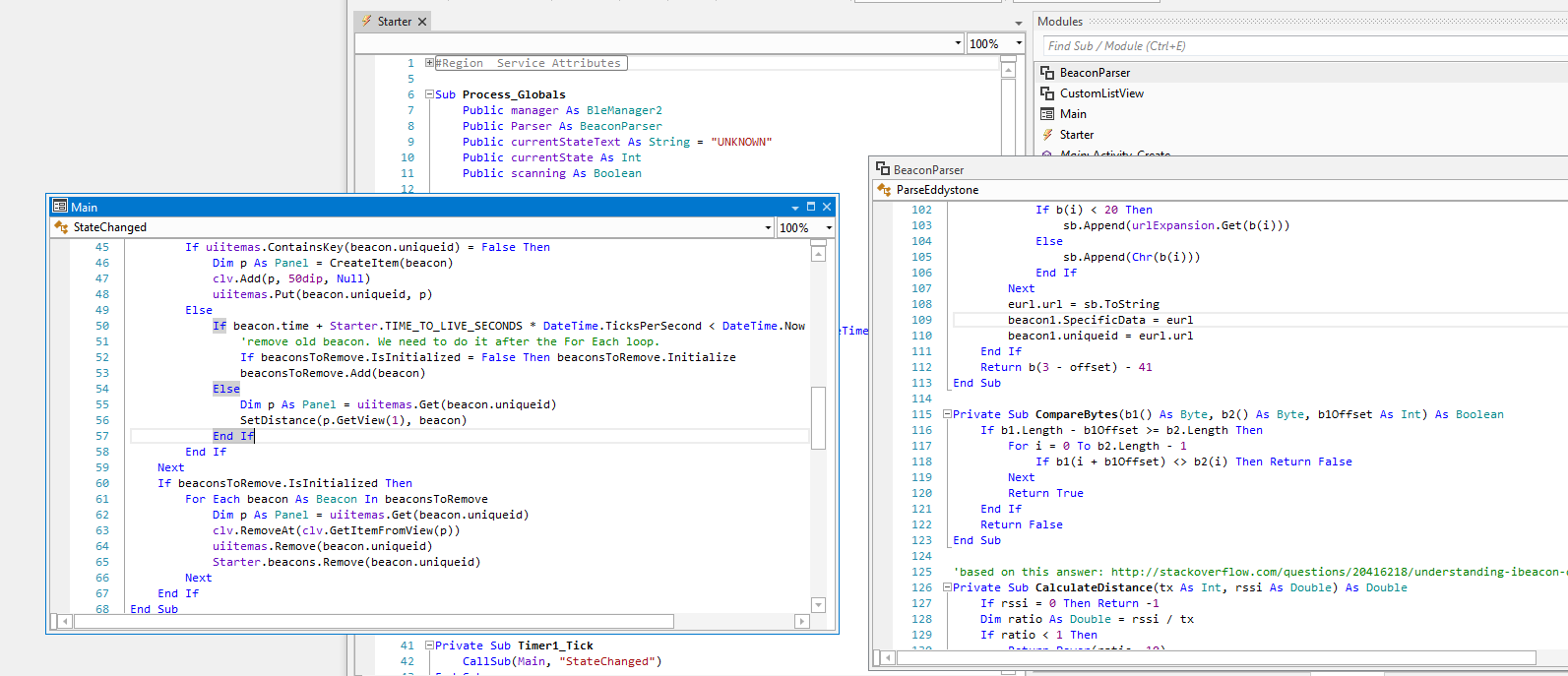
Horizontalement Verticalement

Vous pouvez aussi double cliquer sur les petits rectangles pour diviser l’écran.



## Fenêtres multiples

Si vous travaillez dans différents modules vous pouvez déplacer les fenêtres hors de la fenêtre principale de l’EDI en tant que fenêtres indépendantes.



## Ctrl + E Recherche de routines ou modules

Ctrl + E Recherche de routines ou de modules. Très utile pour les grands projets.

## Ctrl + Clic sur une routine ou variable

Ctrl + Clic sue une routine ou une variable saute vers le ligne de sa déclaration.

## F7 - Trouver toutes les références

Pas directement en relation avec la navigation, mais peut être utile pour les grands projets.

Details dans [Trouver toutes les références](#_Find_All_References).

## Ctrl + F Recherche Rapide

Ctrl + F Recherche rapide, détails dans [Recherche Rapide](#_Quick_Search).

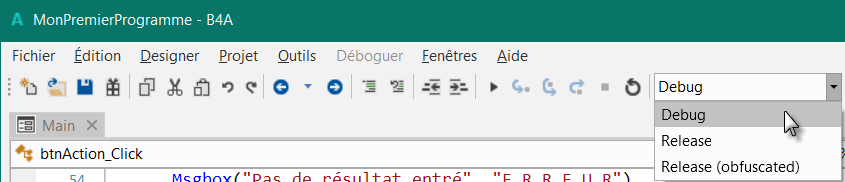
# Débogage B4A, B4i, B4J

Le débogage est une part importante lors du développement.

Le débogage dans B4R est différent par rapport à celui dans B4A, B4i and B4J.

## B4A, B4i, B4J

Pour activer le mode de débogage vous devez sélectionner *Debug* pour le mode de compilation.



**Notes à propos du débogage (seulement B4A) :**

* Des points d’arrêt (breakpoints) sont ignorés dans les routines suivantes :   
  Globals, Process\_Globals et Activity\_Pause.
* Services – Des points d’arrêt après un appel à StartService sont ignorés. Des points d’arrêt dans des Service\_Create et Service\_Start arrête l’exécution pour un temps spécifié (environ 12 secondes) pour prévenir que le système d’exploitation ne le détruise.
* Les événements qui sont générés lorsque le programme est en veille sont exécutés. Les points d’arrêt dans la routine événement seront ignorés (seulement si le programme est en veille).
* La taille des données envoyées par le dispositif vers l’EDI est limitée. Des textes longs peuvent être tronqués.

Les deux principaux utilitaires pour le débogage sont :

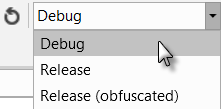
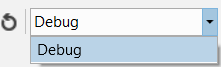
[Points d’arrêt](#_Breakpoints) (Breakpoints) – Vous pouvez marquer des lignes en tant que points d’arrêt en cliquant dans la marge gauche de la ligne. Le programme s’arrête lorsqu’il arrive sur un point d’arrêt et vous permet d’inspecter l’état courant du programme.

[Logging](#_Logs) – L’onglet Logs dans la partie droite de l’EDI est très utile. Y sont affichés des messages sur la durée de vie de certaines composantes ainsi que des messages définis avec le mot clé Log. Vous pouvez cliquer sur le bouton Connecter pour connecter le dispositif si nécessaire. Notez qu’il y a une case à cocher Filtre. Quand elle est cochée, vous ne verrez que les messages relatifs à votre programme. Quand elle est décochée vous verrez tous les messages générés dans le système. Si vous rencontrez un problème et ne voyez pas de message pertinent dans l’onglet, il peut être utile de décocher le filtre pour voir s’il y un message pertinent.

Notez que les Logs (enregistrements) sont maintenus dans le dispositif. Lorsque vous vous connectez à un dispositif vous verrez les messages précédents.

### Mode de compilation Debug

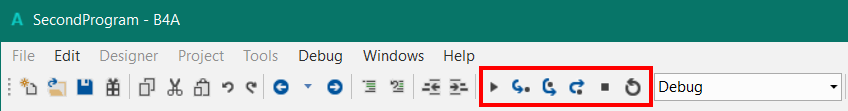
Les modes de compilation sont différents dans les différents produits :

B4A, B4J B4i, seulement Debug

#### Barre d’outils de débogage

La barre d’outils de débogage se trouve à droite de la barre d’outils.



**Barre d’outils de débogage :** 

 Exécuter F5 Exécute le programme, pas d’action en Debug.

 Entrer F8 Exécute l’instruction suivante.

 Sauter F9 Saute l’appel d’une routine, mais exécute son code sans y entrer.

 Sortir F10 Exécute le reste de la routine et sort.

 Stop Arrête le programme.

 Redémarrer F11 Redémarre le programme.

Les exemples ci-dessous utilisent le projet MonPremierProgramme.

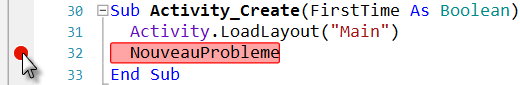
##### Exécute F5

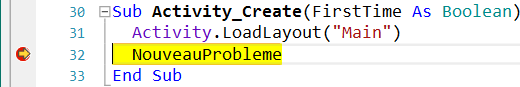
Exécute le programme.

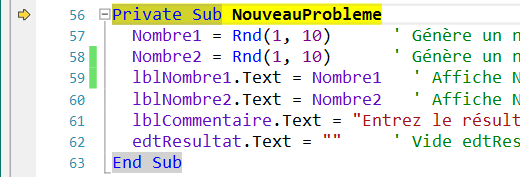
Si le programme est arrêté à un point d’arrêt, il continue jusqu’au point d’arrête suivant ou termine l’exécution.

##### Entrer F8

Le débogueur exécute le code pas à pas.

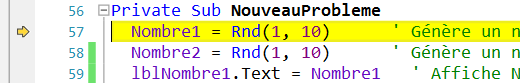
Dans le projet MonPremierProgramme nous mettons un point d’arrêt à la ligne NouveauProbleme.

Lors de l’exécution, le programme s’arrête sur la ligne 31 Nouveau.



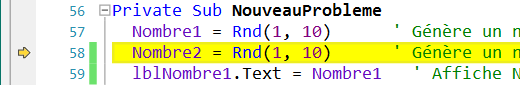
Cliquez sur .

Le débogueur exécute la ligne suivante, Sub Nouveau dans notre cas.



Cliquez à nouveau sur.

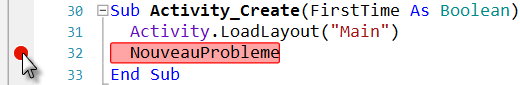
Le débogueur exécute la ligne suivante, Nombre1 =…

Cliquez encore une fois sur.

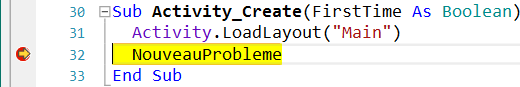
Le débogueur exécute la ligne suivante, Nombre2 =…

##### Sauter F9

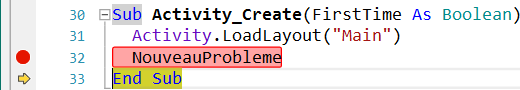
Si la ligne courante est un appel à une routine, le débogueur exécutes le code de la routine puis s’arrête à la ligne suivant la ligne d’appel.



Dans le projet MonPremierProgramme nous mettons un point d’arrêt à la ligne NouveauProbleme.



Lors de l’exécution, le programme s’arrête sur la ligne 31 Nouveau.

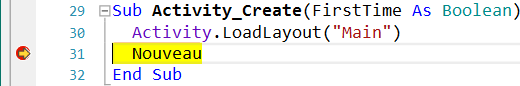
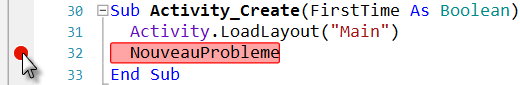
Cliquez sur .

Le débogueur exécute le code de Nouveau et saute à la ligne suivante qui est End Sub de Activity\_Create.

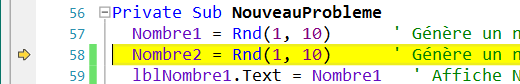
##### Sortir F10

Si la ligne courante se trouve à l’intérieur d’une routine, le débogueur finit l’exécution de cette routine et saute à la ligne suivant celle de l’appel de la routine.

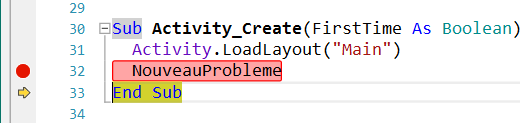
Dans le projet MonPremierProgramme nous mettons un point d’arrêt à la ligne 31 Nouveau.



Lors de l’exécution, le programme s’arrête sur la ligne NouveauProbleme.



Nous allons pas à pas avec   
à la ligne suivante dans la routine.

Cliquez sur.

Le débogueur exécute le reste du code de la routine Nouveau et saute à la ligne suivant celle de l’appel de la routine qui est

End Sub de Activity\_Create.

##### Stop

Arrête le programme est quitte le débogueur.

##### Redémarrer F11

Redémarre le programme tout en restant dans le débogueur.

Exécute :

**B4A** Process\_Globals, Globals, Activity\_Create et recharge le layout.

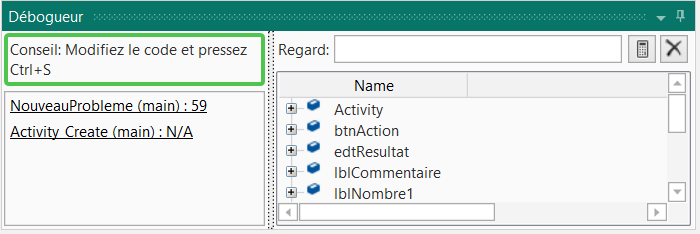
**B4i** Process\_Globals.

**B4J** Process\_Globals.

Ceci est utile si vous changez un fichier layout.

*Redémarrer* est différent du [bouton d’état](#_Le_bouton_d’état)  expliqué dans le chapitre suivant.

### Fenêtre Débogueur



Dans la fenêtre Débogueur nous avons, exemple avec MonPremierProgramme, et un point d’arrêt sur la ligne 45 :

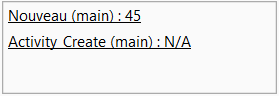
#### Le bouton d’état



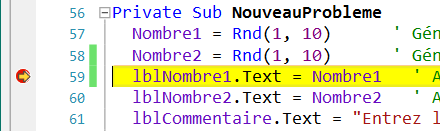
Montre que le programme s’exécute, le bord est en vert.

Lorsque vous changez le code, le bord du bouton devient rouge. Pour mettre le code à jour, cliquez sur le bouton ou pressez Ctrl + S.

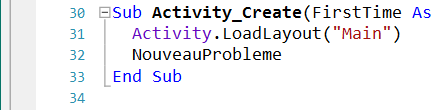
#### La fenêtre points d’arrêt



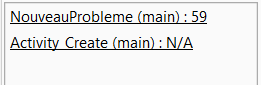
La fenêtre des points d’arrêt montre où le programme s’est arrêté.



  
Le programme s’est arrêté à la  
ligne , dans la routine Nouveau dans le module Main.



  
La routine d’appel Activity\_Create, et le numéro de ligne (pas disponible dans ce cas).

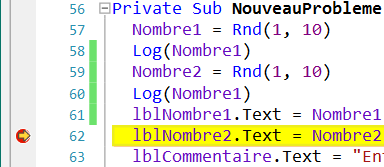


Si vous cliquez sur une des lignes, le débogueur saute vers cette ligne dans l’éditeur.

#### La fenêtre Regard

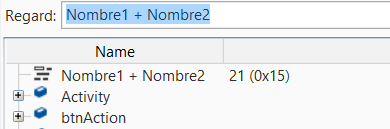


La fenêtre Regard permet de vérifier des fonctions plus complexes lors de tests et de débogage.



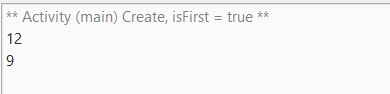
Dans le programme MonPremierProgramme ajoutez les 2 lignes Log et mettez un point d’arrêt sur la ligne 48.

Lancez le programme.

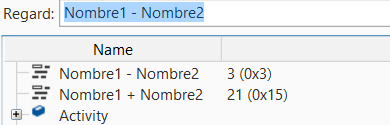
Dans le champ de la fenêtre Regard, entrez :

Nombre1 + Nombre2

Cliquez sur  pour afficher le résultat dans le haut de la liste.



Les valeurs des deux nombres, Nombre1 et Nombre2, sont affichés dans la fenêtre Log.



Vous pouvez entrer une nouvelle fonction   
Nombre1 - Nombre2   
et afficher le résultat.

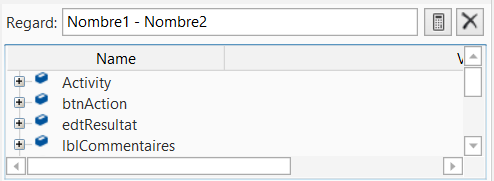
Cliquez sur  pour supprimer les fonctions. Ceci supprime toutes les fonctions.

Nous aurions, bien entendu, pu afficher ces fonctions avec des Logs avec :

Log(Nombre2 + Nombre2)

Log(Nombre2 - Nombre2)

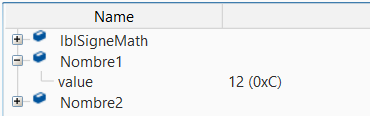
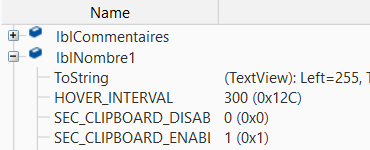
#### La fenêtre objets

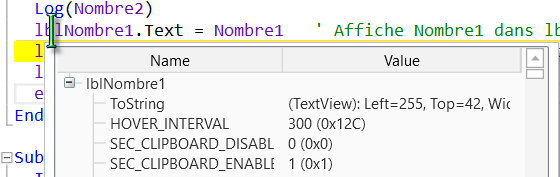


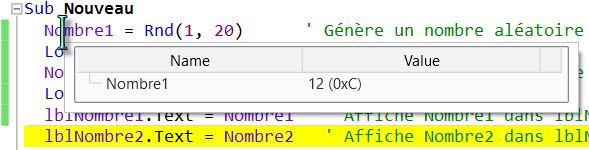
La fenêtre objets affiche tous les objets et variables triés par ordre alphabétique.

Cliquez sur  pour afficher les détails de l’objet :

Exemples :

*  Nombre1  
    
    
  Affiche la valeur courante (7).
*  lblNombre1  
    
    
  Affiche toutes les propriétés de l’objet, un Label dans cet exemple.  
    
    
    
    
    
  Dans la ligne compète on trouve les propriétés Left, Top, Width, Height, Tag et Text qui vaut 12, la valeur de Nombre1.



Vous obtenez la même information si vous passez avec le souris au-dessus de l’objet, lblNombre1 dans l’exemple.

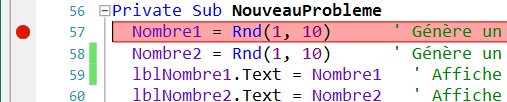
Nombre1

#### Points d’arrêt

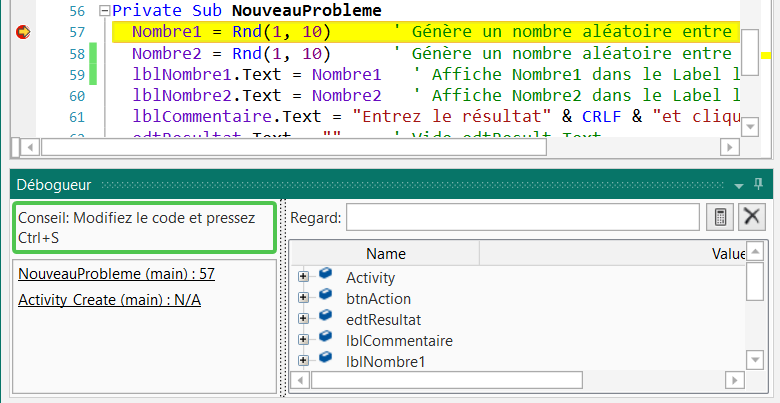
Les points d’arrêt facilitent grandement le débogage. Vous pouvez définir des points d’arrêt n’importe où dans le code.

**B4A, des points d’arrêt dans ces routines Process\_Globals et Globals sont refusés.**

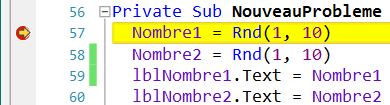
Cliquez dans la marge gauche dans la ligne dans laquelle vous voulez définir un point d’arrêt. Lors de l’exécution, le programme s’arrête au premier point d’arrêt.



Lancez le prgramme, il s’arrête sur la ligne qui contient un point d’arrêt et l’éditeur ressemble à l’image ci-dessous. La ligne où le programme s’arrête est mise en évidence en jaune.

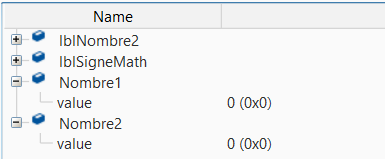


Au bas de l’éditeur vous trouvez la fenêtre Débogueur.

 Exemple avec MonPremierProgramme :

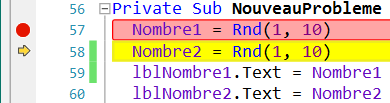
Définissez un point d’arrêt dans la ligne 43.

Lancez le programme, il s’arrête sur la ligne 43.

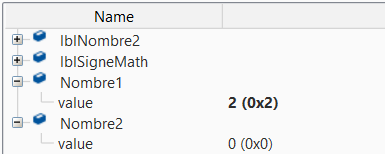
Dans la fenêtre Objets regardez Nombre1 et Nombre2 :

Si vous voyez ceci  à gauche de Nomber1 ou Nombre2, cliquez sur ce signe pour montrer les détails.

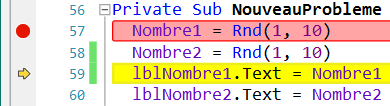
Les deux valeurs valent 0.

Cliquez sur .

Le programme exécute la ligne et saute à la suivante.

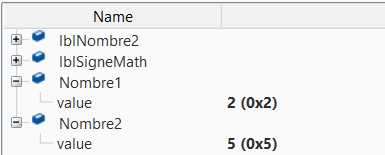


Vous voyez que la valeur de Nombre1 a changé.



Cliquez encore une fois sur  .

Le programme exécute la ligne et saute à la suivante.

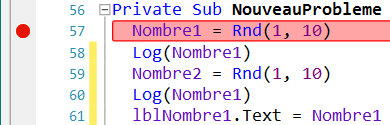


Vous voyez que la valeur de Nombre2 a changé.

Le meilleur moyen pour apprendre à déboguer est ; tester, tester et tester !

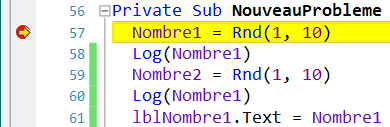
#### Logs

Exemple avec MonPremierProgramme.

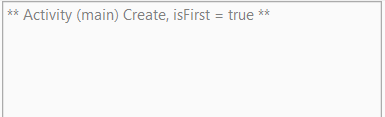


Ajoutons deux lignes avec des Log pour afficher les valeurs des deux nombres dans l’onglet Logs.

Nous ajoutons un point d’arrêt à la ligne 57 pour voir ce qui se passe.

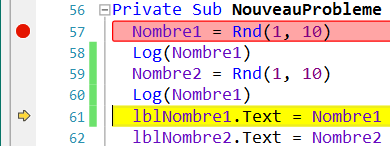


Lancez le programme, il s’arrête sur la ligne 57.

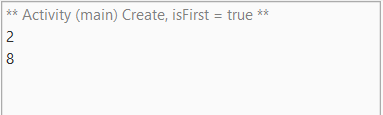


Dans l’onglet Logs nous ne voyons pour le moment que 

indiquant que le programme s’exécute.

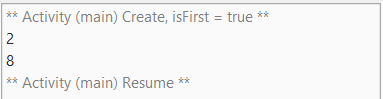


Cliquez quatre fois sur  jusqu’à la ligne 61.

Dans l’onglet Logs nous voyons les valeurs des deux nombres.

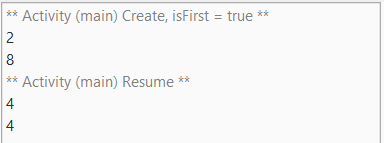
Cliquez sur  pour terminer l’exécution.

Et supprimez le point d’arrêt.





est ajouté à l’affichage indiquant que le programme a exécuté la routine Activity\_Resume.

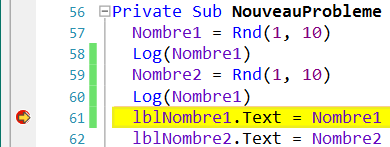


Lorsque vous utilisez le programme, deux nouveaux nombres seront affichez chaque fois que le programme exécute la routine Nouveau.

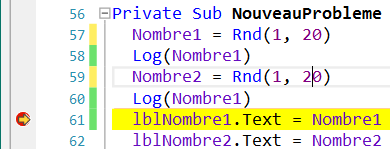
#### Modification de code dans le Débogueur

Il est possible de changer le code dans le débogueur et voir l’évolution sans redémarrer le programme.

Avec le programme MonPremierProgramme, les deux lignes Log et un point d’arrêt sur la ligne 61.



Lancez le programme, il s’arrête sur la ligne 61.



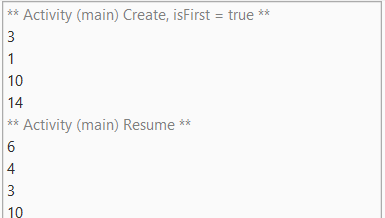
Dans les deux lignes avec Rnd(1, 10) nous changeons les deux nombres 10 en 20.

Le bouton d’état a changé de couleur confirmant que le code a été changé.

Appuyez sur Ctrl + S pour mettre le code à jour.

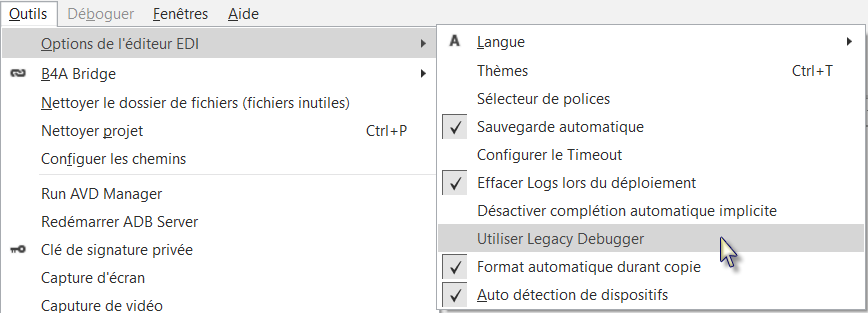
Le bouton d’état revient en vert indiquant que le programme a été mis à jour et est de nouveau en cours d’exécution

Supprimez le point d’arrêt et utilisez le programme, vous voyez que les nombres peuvent être entre 1 et 19 maintenant.



### Debug (legacy) mode B4A seulement

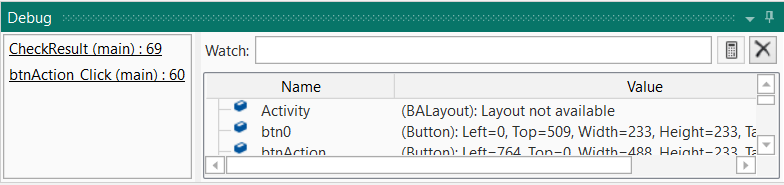
Dans certains cas, il peut être utile d’utiliser le débogueur ‘legacy’, vous pouvez l’activer dans le menu Outils / Options de l’éditeur EDI.



**Debug(legacy) :** Lorsque cette option est choisie, le fichier compilé contient du code de débogage.

Le nom du fichier compilé .apk se termine avec \_DEBUG.apk. Vous ne devez pas distribuer ce fichier vu qu’il contient du code supplémentaire pour le débogage.

Pour distribuer votre projet vous devez le compiler en mode *Release* ou *Release (obfuscated)*.



Les boutons de la barre d’outils débogage sont activés .

Ils fonctionnent de manière similaire que dans le mode Debug (rapid).

## Débogage B4R

Le débogage est une partie importante dans le développement.

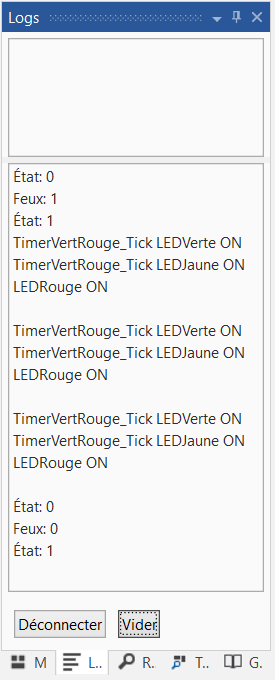
Dans B4R il n’y a pas de mode Debug comme dans les autres langages B4x.

Le débogage ne peut se faire qu’avec des [Logs](#_Logs).

Dans l’onglet Logs à droite de l’EDI sont affichés des messages relatifs à l’exécution du programme ainsi que des messages affichés avec le mot clé Log. Vous devez cliquer sur le bouton Connecter pour connecter le dispositif.

### Débogage avec le projet FeuxSignalisation

Dans le projet FeuxSignalisation j’ai ajouté plusieurs lignes pour afficher des commentaires montrant l’évolution de l’exécution du programme.



Au lancement du programme l’onglet Logs est vide.

Puis :

En pressant le bouton > État: 0

Les feux sont sur ON > Feux: 1 feu rouge ON

En relâchant le bouton > État: 1

Les ‘Logs’ du Timer (minuteur) :

Au début LEDVerte = False

Dans TimerVertRouge\_Tick nous changeons de rouge à vert :

> TimerVertRouge\_Tick LEDVerte ON

Prochain Tick, LEDVerte = True nous appelons FinJaune et changeons de vert à jaune:

> TimerVertRouge\_Tick LEDJaune ON

Après le délai, FinJaune est exécuté où nous éteignons la LED jaune et allumons la LED rouge :

> LEDRouge ON

Fin du premier cycle.

Début du cycle suivant : > TimerVertRouge\_Tick LightGreen

En pressant le bouton > État: 0

Les feux sont sur OFF > Light: 0

En relâchant le bouton > État: 1