

Premiers pas avec GPS (GNAT Programming Studio)

GPS est l'environnement de programmation graphique fournit par AdaCore pour accompagner le compilateur Ada GNAT. Il est disponible en licence GPL.

1) Installation

À partir du site "<https://libre.adacore.com>" à la page "Download GNAT GPL", télécharger le fichier suivant sur le bureau du Mac :
Compilateur 2006, PPC-DARWIN : "gnat-gpl-2006-powerpc-apple-darwin7.4.1-bin.tar.gz".

Il est nécessaire de s'enregistrer gratuitement auparavant.
Lancer le Terminal ou xterm depuis Applications/Utilitaires ou X11 et taper les commandes suivantes :

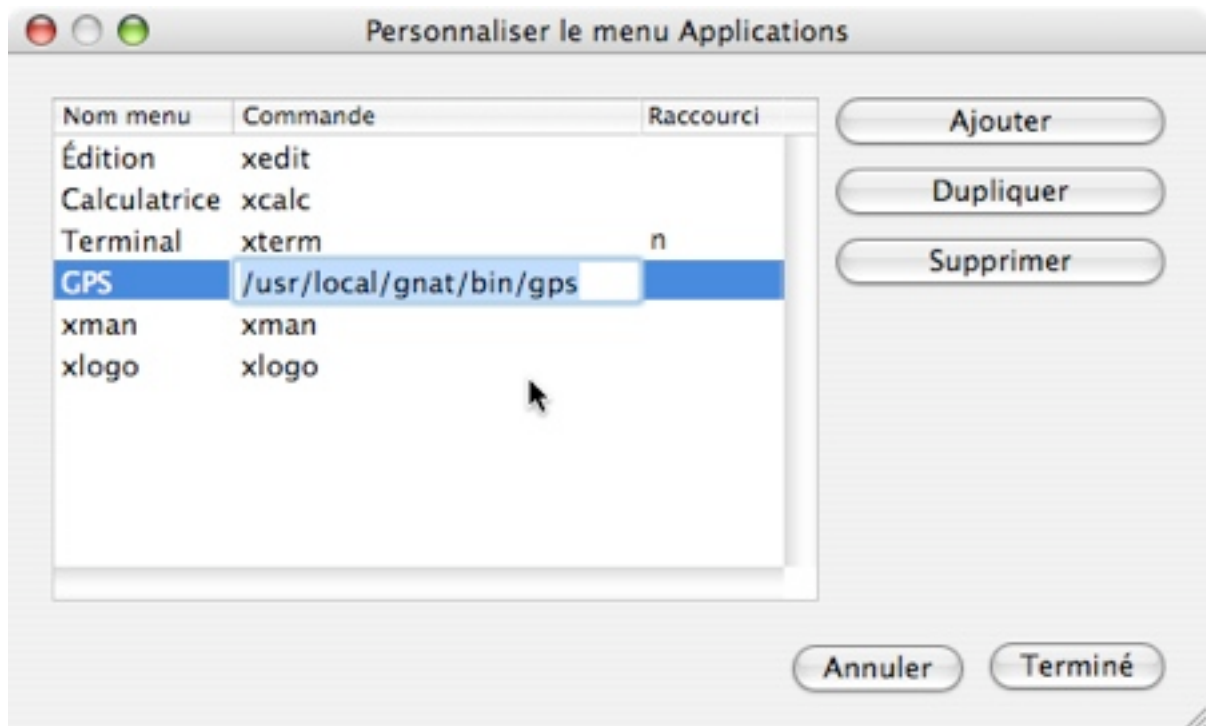
```
$ cd  
$ tar xzvf ~/Desktop/gnat-gpl-2006-powerpc-apple-darwin7.4.1-bin.tar.gz  
$ cd gnat-gpl-2006-powerpc-apple-darwin7.4.1-bin  
$ sudo doinstall
```

Le compilateur GNAT ainsi que GPS GPL 2006 sont installés sous "/usr/local/gnat".

Voir par ailleurs sur Blady pour la génération complète de la version GNAT et GPS GPL 2008 pour Mac OS 10.4.

2) Première utilisation avec X11

- . lancer "X11" depuis Applications/Utilitaires
- . sélectionner "Personnaliser le menu..." depuis le menu Applications de X11
- . cliquer sur "Ajouter"
- . double-cliquer dans la colonne "Nom menu" saisir : "GPS"
- . double-cliquer dans la colonne "Commande" saisir le chemin d'accès :
"/usr/local/gnat/bin/gps"
- . cliquer sur "Terminer"



- . sélectionner maintenant "GPS" depuis le menu Applications de X11

Cliquer OK si la question "Created config directory /Users/.../.gps" est posée.

2) Création du premier projet

a) Accueil

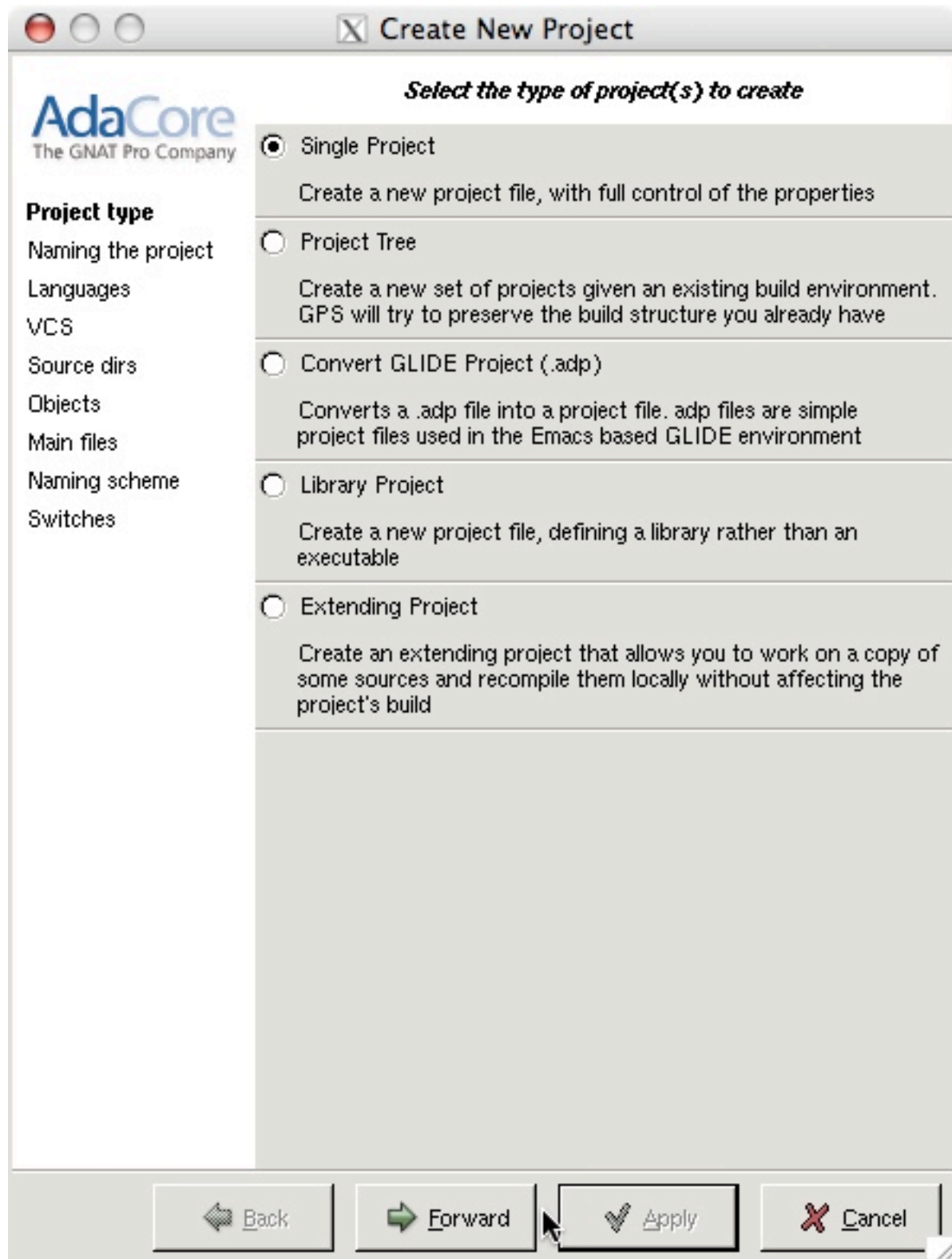
Sélectionner "Create new project with wizard", cliquer sur OK.



De nouveaux projets pourront être également créer plus tard de la même façon en sélectionnant Project->New... dans GPS.

b) Project type

"Single Project" est sélectionné par défaut, cliquer sur "Forward"



c) Naming the project

Saisir le nom du projet et son emplacement en cliquant sur "Browse", cliquer plusieurs fois sur "Forward" jusqu'au menu "Objects".

d) Objects

Saisir l'emplacement des fichiers compilés en cliquant sur "Browse" en face de "Build Directory", cliquer sur "Forward".

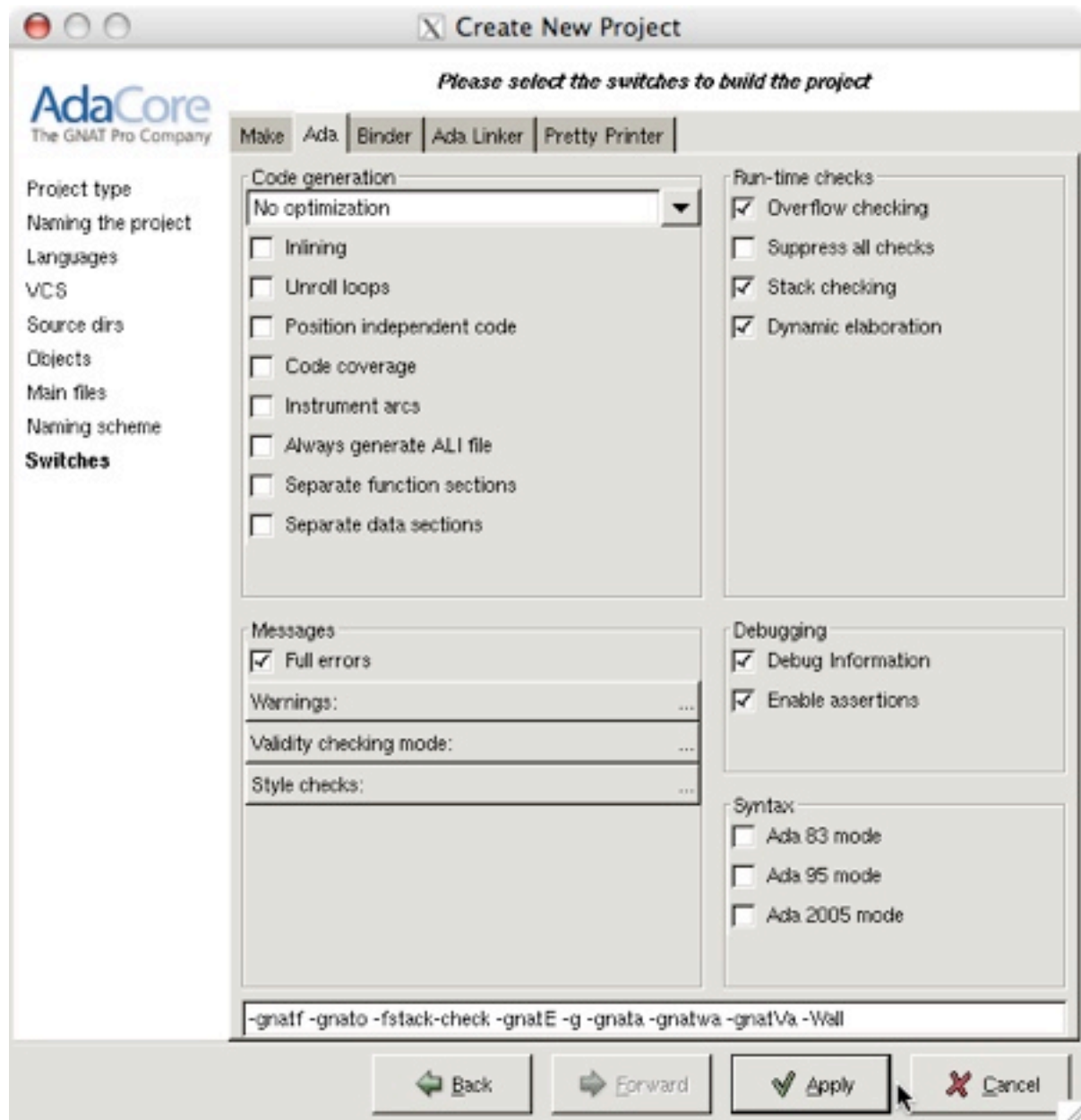
e) Main files

Si le programme principal existe déjà, ajouter le en cliquant sur "Add" sinon nous pourrons le fait ultérieurement en sélectionnant Project->Edit Project Properties.

Cliquer sur "Forward" jusqu'au menu "Switches".

f) Switches

Cliquer sur l'onglet Ada positionner les options comme sur la figure.
Ajouter "-gnatwa -gnatVa -Wall" dans la ligne du bas. Cliquer sur "Apply".



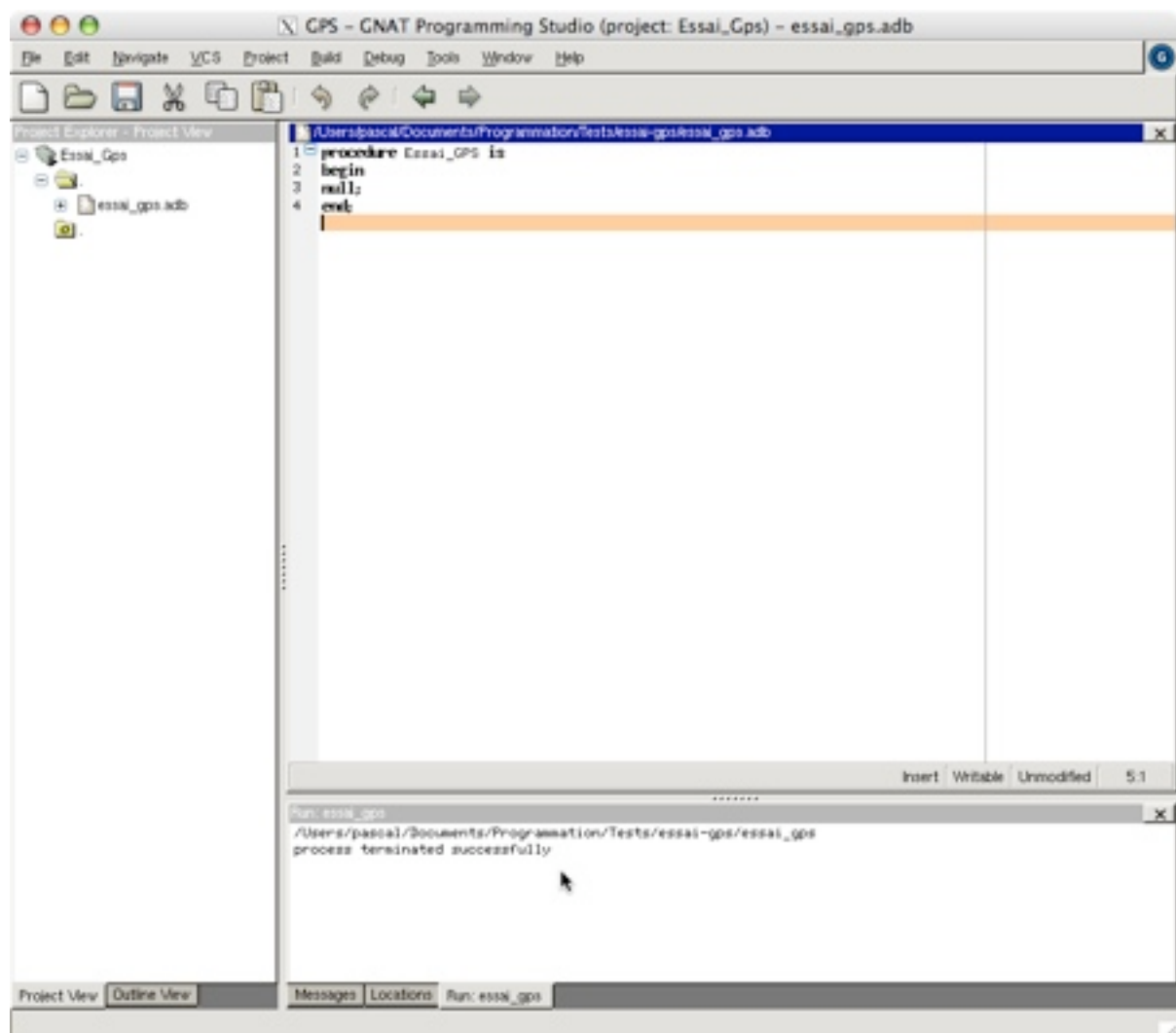
Si le répertoire du projet n'est pas déjà créer alors cliquer sur Yes à la demande de création.

3) Création du premier programme

Si le répertoire ne contient aucun source Ada, sélectionner File->New
Ajouter le code Ada puis sélectionner File->Save et entrer un nom de fichier.
Nous allons également l'ajouter comme programme principal en sélectionnant
Project->Edit Project Properties puis l'onglet Main Files, cliquer sur le bouton
Add, cocher la case en face du fichier Ada et cliquer OK

Sélectionner "Project View" dans le menu Project
Dans la fenêtre Project Explorer - Project View double-cliquer sur le dossier
présent puis sur un des sources pour l'éditer

Sélectionner Build->Make->nom_programme.adb (en face de F4) pour compiler.
Noter que les fichiers modifiés sont automatiquement sauvegardés.
Sélectionner Build->Run->nom_programme (en face de Shift+F2) puis OK pour
exécuter le programme.



4) Édition d'un code source

Le code source est affiché avec la syntaxe mise en valeur : mot-clé, commentaires, chaînes de caractères, nombres, caractères...

Ces mises en valeurs sont paramétrables dans le menu préférences.

En parcourant le code source avec le curseur de la souris, il apparaît des fenêtres contextuelles avec des informations sur l'objet survolé.

```

125  procedure Suppression (O : in out TInstance; Element : out TElement) is
126  begin
127  if O.Queue = 0 then
128  raise Vide;
129  end if;
130  Element := O.Tab (0.Tete);
131  O.Tete := (O.Tab (0.Tab));
132  if (O.Queue mod 2) = 0 then
133  O.Queue := 0;
134  end if;
135  --Put_Line (O.Tab);
136  end Suppression;
137
138  end Suite;

```

The context menu for the variable `0.Tab` contains the following information:

- prog_avan2.Suite.TInstance.Tab
- local array declared at prog_avan2.adb:78
- Suites contiguës bouclée (anneau)
- Exemple 5.4 et exercice 5.1
- Type: TTab

Ces informations seront d'autant plus complètes avec une compilation aboutie.

Un **<ctrl>-clic** (ou clic bouton droit de la souris) fait apparaître un menu contextuel avec de nombreuses possibilités correspondant à l'objet cliqué.

```

72  procedure Adjonction (O : in out TInstance; Element : TElement);
73  procedure Suppression (O : in out TInstance; Element : out TElement);
74  Vide, Plein : Boolean;
75  private
76  type TTab is array of TElement;
77  type TInstance is record
78  Tab : TTab;
79  Tete : Positive;
80  Queue : Natural;
81  Plein : Boolean;
82  end record;
83  end Suite;
84
85  -- Le code des procédures
86
87  function PGCD (X, Y : Positive) return Positive;
88  A : Positive := X;
89  B : Positive := Y;
90  C : Natural;

```

The context menu for the procedure `Adjonction` contains the following options:

- Goto declaration of Adjonction
- Goto body of Adjonction
- Goto file spec->body
- Adjonction calls
- Adjonction is called by
- References
- Browsers
- Locate in Project View: prog_avan2.adb
- Edit with external editor
- Refactoring
- Documentation
- Coverage
- Casing
- Compute metrics for prog_avan2.adb
- Properties...

Pour sélectionner un bloc utiliser la souris en tirant sur le bloc à sélectionner ou utiliser **<shift>** avec les touches fléchées.

Pour ré-indenter une ligne ou un bloc du code source faire **<ctrl><tab>**.

Pour compléter automatiquement un nom faire **<ctrl><espace>**.

Si c'est la première fois une boîte de dialogue affiche :

"You are using smart completion for the first time.

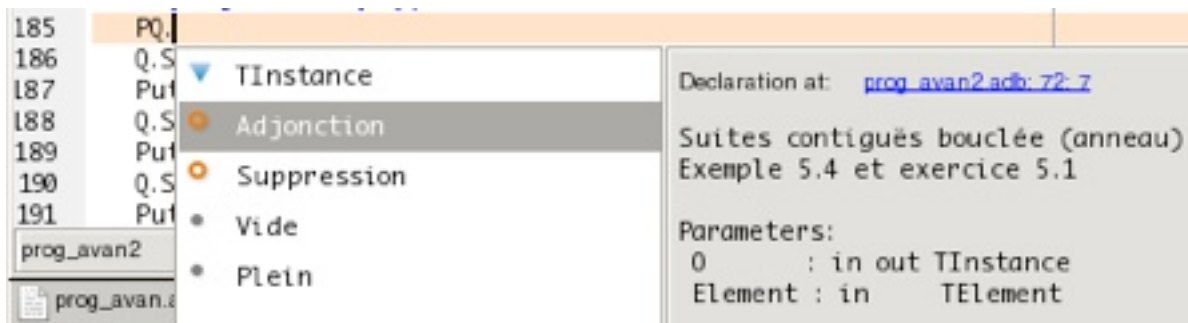
...

Do you want to activate this now?"

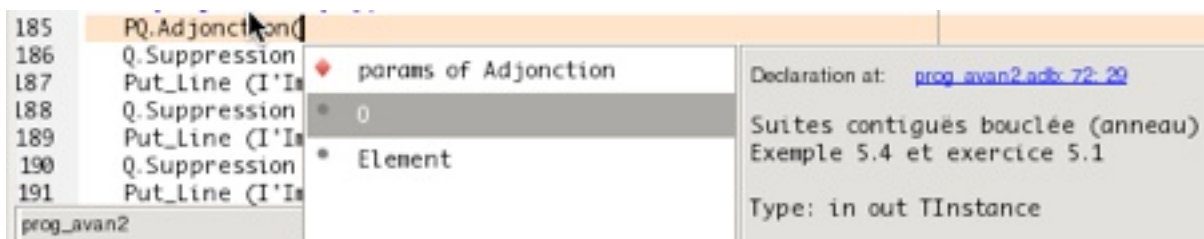
Cliquer sur "Oui".

Une boîte de sélection apparaît, cliquer dans le choix désirer. Si aucun ne convient taper <ESC>.

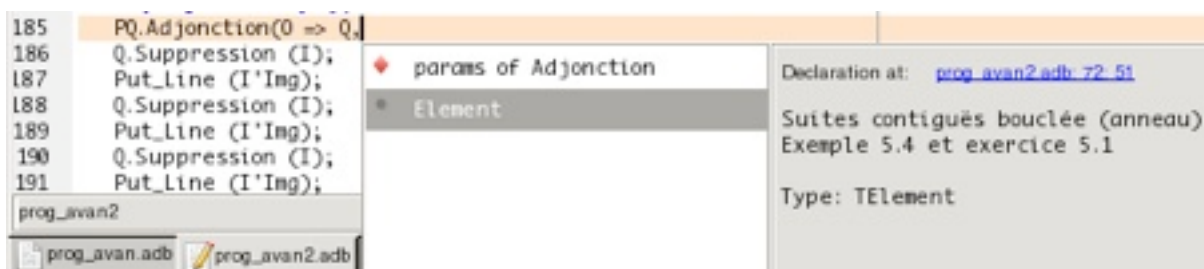
La fenêtre de choix apparaît également automatiquement après un point :



Pour compléter une liste de paramètres d'une procédure ou une fonction taper parenthèse, la liste des paramètres apparaît.



S'il y a plusieurs paramètres, taper virgule.



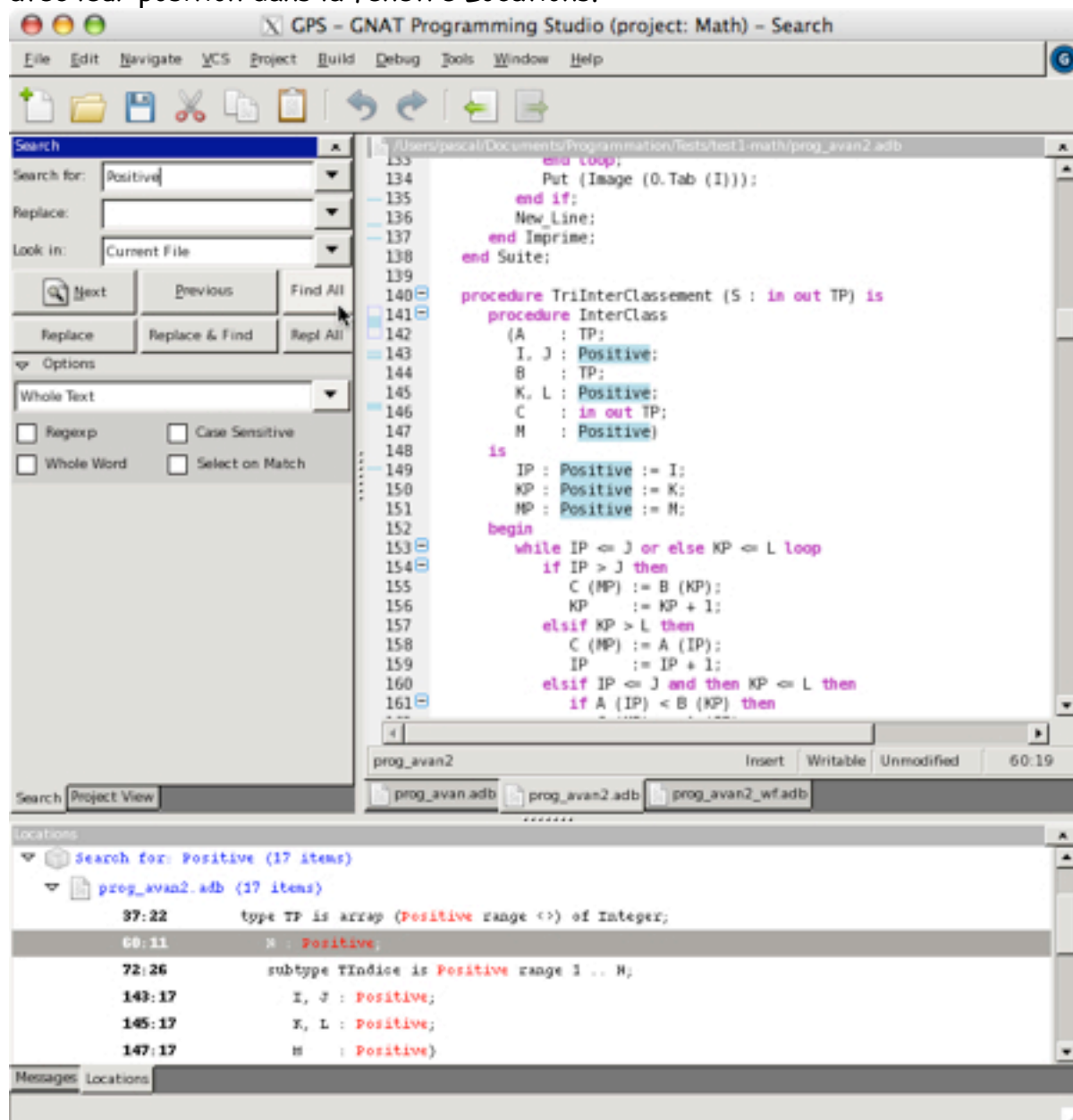
5) Recherche dans un code source

a) Un mot

Double-cliquer sur le mot à rechercher et cliquer sur le menu Navigate->Find or Replace... ou plus rapidement sur <ctrl>-F. Le panneau de recherche s'affiche. Cliquer sur Next ou <ctrl>-N, chaque mot trouvé est surligné. Cliquer sur Previous ou <ctrl>-P pour revenir sur le précédent mot trouvé.

Plusieurs options sont à découvrir comme la recherche dans plusieurs fichiers en sélectionnant Look in ou Options.

Cliquer sur Find All affiche en couleur surligné l'ensemble des mots trouvés avec leur position dans la fenêtre Locations.



Un clic-droit ou <ctrl>-clic sur le menu contextuel Clear Locations View dans la fenêtre Locations supprime les surlignages.

b) Une déclaration, une définition

Positionner le curseur sur un identificateur du code source et cliquer sur le menu Navigate->Goto Declaration affiche l'endroit où l'identificateur est déclaré, cliquer sur Navigate->Goto Body affiche l'endroit où l'identificateur est défini (parfois il s'agit du même endroit). Le menu contextuel clic-droit ou <ctrl>-clic propose les mêmes fonctions.

c) Les appels

Positionner le curseur sur un identificateur du code source et afficher le menu contextuel avec clic-droit ou <ctrl>-clic, le nom de l'identificateur s'affiche avec calls pour afficher la liste des procédures appelées :

The screenshot displays the GNAT Programming Studio (GPS) interface. The main editor shows the source code for 'prog_avan2.adb'. A context menu is open over the procedure 'Simulation_T' (lines 341-359). The menu options are:

- Goto declaration of **Simulation_T**
- Goto file spec -> body
- Simulation_T calls** (selected)
- Simulation_T is called by
- References
- Browsers
- Locate in Project View: prog_avan2.adb
- Edit with external editor
- Refactoring
- Documentation
- Coverage
- Casing
- Compute metrics for prog_avan2.adb
- Properties...

The 'Call Tree' window at the bottom shows the following calls to 'Simulation_T':

Line	Column	File
413	14	prog_avan2.adb
425	34	prog_avan2.adb
434	34	prog_avan2.adb
445	40	prog_avan2.adb
451	40	prog_avan2.adb
458	31	prog_avan2.adb

et avec called by pour afficher la liste des procédures appelant:

The screenshot shows the GNAT Programming Studio interface. The main editor displays Ada code for a procedure named 'Futur'. A context menu is open over the code, with 'Futur is called by' selected. Below the editor, the 'Call Tree' view is visible, showing the call stack for the selected procedure.

Code in the editor:

```

422     New_Line;
423     Put_Line ("Longueur des queues 1 et 2 : " & L1'Img & L2'Img);
424     if L1 > 0 then
425         Futur (Temps + Poisson (TempsService), FinServ1);
426     end if;
427     when FinServ2 =>
428         L2 := L2 - 1;
429         Put ("Fin de service chez le deuxième serveur à ");
430         Put (Temps, 5, 2, 0);
431         New_Line;
432         Put_Line ("Longueur des queues 1 et 2 : " & L1'Img & L2'Img);
433         if L2 > 0 then
434             Futur (Temps + Poisson (TempsService), FinServ2);
435         end if;
436     when Nouve =>
437         Put ("A");
438         Put (Temps, 5, 2, 0);
439         New_Line;
440         Put_Line ("Longueur des queues 1 et 2 : " & L1'Img & L2'Img);
441         if L1 > 0 then
442             Futur (Temps + Poisson (TempsService), FinServ1);
443         end if;
444         if L2 > 0 then
445             Futur (Temps + Poisson (TempsService), FinServ2);
446         end if;

```

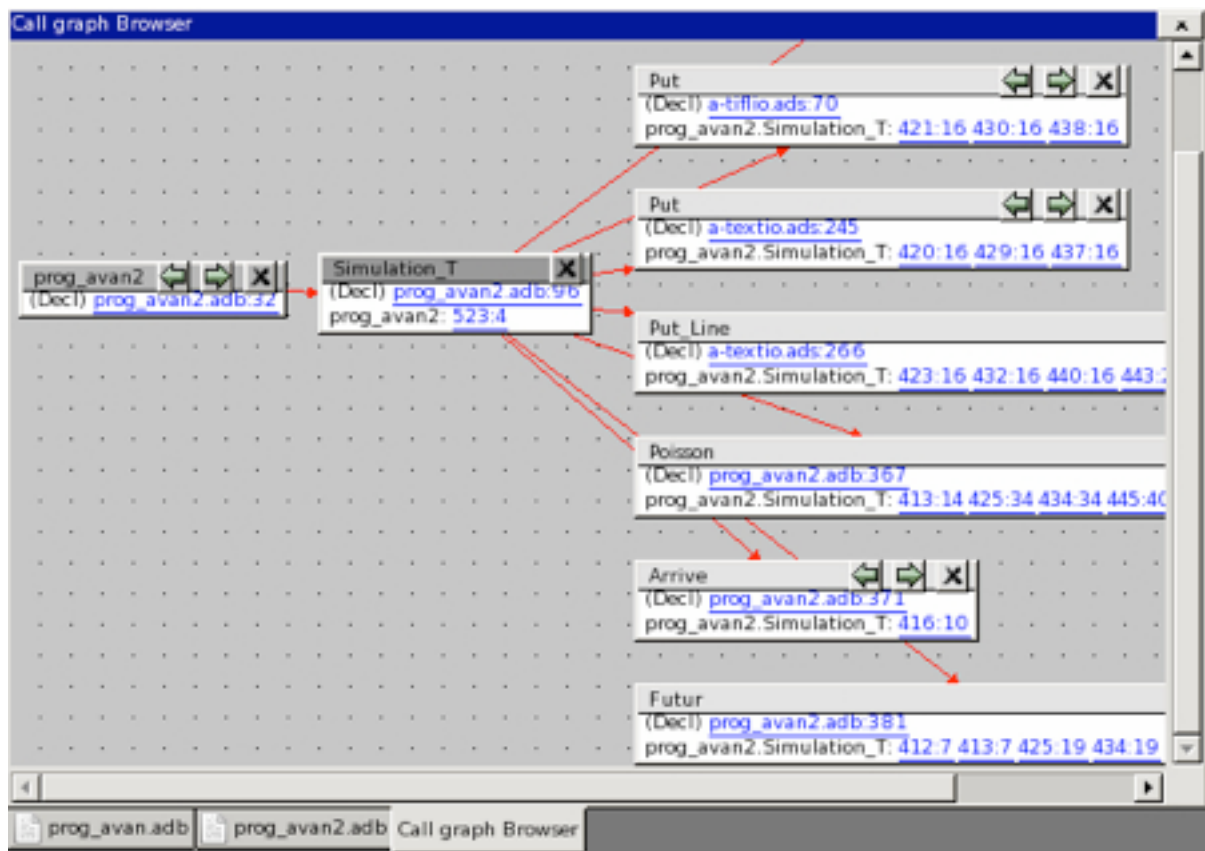
Context Menu Options:

- Goto declaration of **Futur**
- Goto body of **Futur**
- Goto file spec <-> body
- Futur** calls
- Futur is called by** (selected)
- References
- Browsers
- Locate in Project View: **prog_avan2.adb**
- Edit with external editor
- Refactoring
- Documentation
- Coverage
- Casing
- Compute metrics for **prog_avan2.adb**
- Properties...

Call Tree View:

Procedure	Line	File
Futur is called by	412	prog_avan2.adb
Simulation_T	413	prog_avan2.adb
prog_avan2	425	prog_avan2.adb
prog_avan2	434	prog_avan2.adb
prog_avan2	445	prog_avan2.adb
prog_avan2	451	prog_avan2.adb
prog_avan2	458	prog_avan2.adb

Ces deux informations sont également disponibles graphiquement en sélectionnant Browsers->Calls ou Called by dans le menu contextuel :



d) Une ligne

Cliquer sur le menu Navigate->Goto Line ou <ctrl>-G pour afficher ligne spécifique.

Le mois prochain nous verrons quelques astuces de correction d'erreur dans un source Ada.

Pascal Pignard, novembre-décembre 2008, février 2009.