

# Les outils Ada

## 1) Outil de traduction de Pascal en Ada :

L'outil "p2ada" est disponible dans la distribution de gnat-osx et sur le site d'origine.

Dans la distribution de gnat-osx, l'outil sert essentiellement à transformer les fichiers d'interfaces Pascal pour Carbon en interfaces Ada. Il a donc été modifié par Alan Reynolds en conséquence. Il s'utilise de la façon suivante :

```
% p2ada < QuickTime.p > quicktime.ada
```

Il est auparavant utile de transformer les retours à la ligne Mac en Unix et d'expurger les nombreuses macros conditionnelles dont Apple est friand. Ces utilitaires se trouvent au paragraphe 4).

Pour convertir un module Pascal complet il vaut mieux utiliser la distribution d'origine. L'utilitaire s'utilise de la même façon :

```
% p2ada < BasicDef.p > BasicDef.ada
```

Voir le texte spécialement consacré à "p2ada" à la page "À savoir".

## 2) Outil de formatage du code source Ada :

L'outil "reformat" est disponible dans la distribution de gnat-osx.

Son exécution sans paramètre nous affiche judicieusement les dits paramètres:

```
% reformat
```

```
Usage reformat [options] infile [outfile]
```

```
options =
```

```
-id Upper_Case|Lower_Case|Mixed_No_Force_Lower|Mixed_Force_Lower
```

```
-rw Upper_Case|Lower_Case|Mixed_No_Force_Lower|Mixed_Force_Lower
```

```
-c Colorize_Only|Colorize|Bold|Bold_Only|No_Colorize
```

```
-i N
```

```
-nip
```

```
-id specifies case for identifiers
```

```
-rw specifies case for reserved words
```

```
-i specifies amount to indent each level
```

```
-nip specifies not to indent based on parenthesis depth
```

```
-nar specifies not to do advanced reformat
```

```
-nrt specifies not to reformat types
```

```
-c specifies whether to output colorized or bolded RTF, or plain ASCII
```

\* Pour les simples besoins de clarification du code source j'utilise la syntaxe suivante :

```
% reformat -id Mixed_No_Force_Lower -rw Lower_Case -i 2 BasicDef.ada  
BasicDef.ada
```

\* Pour les besoins d'impression du code source j'utilise la syntaxe suivante :

```
% reformat -id Mixed_No_Force_Lower -rw Lower_Case -i 2 -c Colorize  
BasicDef.ada BasicDef.rtf
```

Avec le résultat suivant :

```
package BasicDef is  
  with  
    Files;  
  type TClef is String;  
  type TElement is FSSpec;  
  function NonDefini return TElement;  
end BasicDef;  
  
package body BasicDef is  
  function ResultNonDefini return TElement is  
    ResultNonDefini : TElement;
```

```

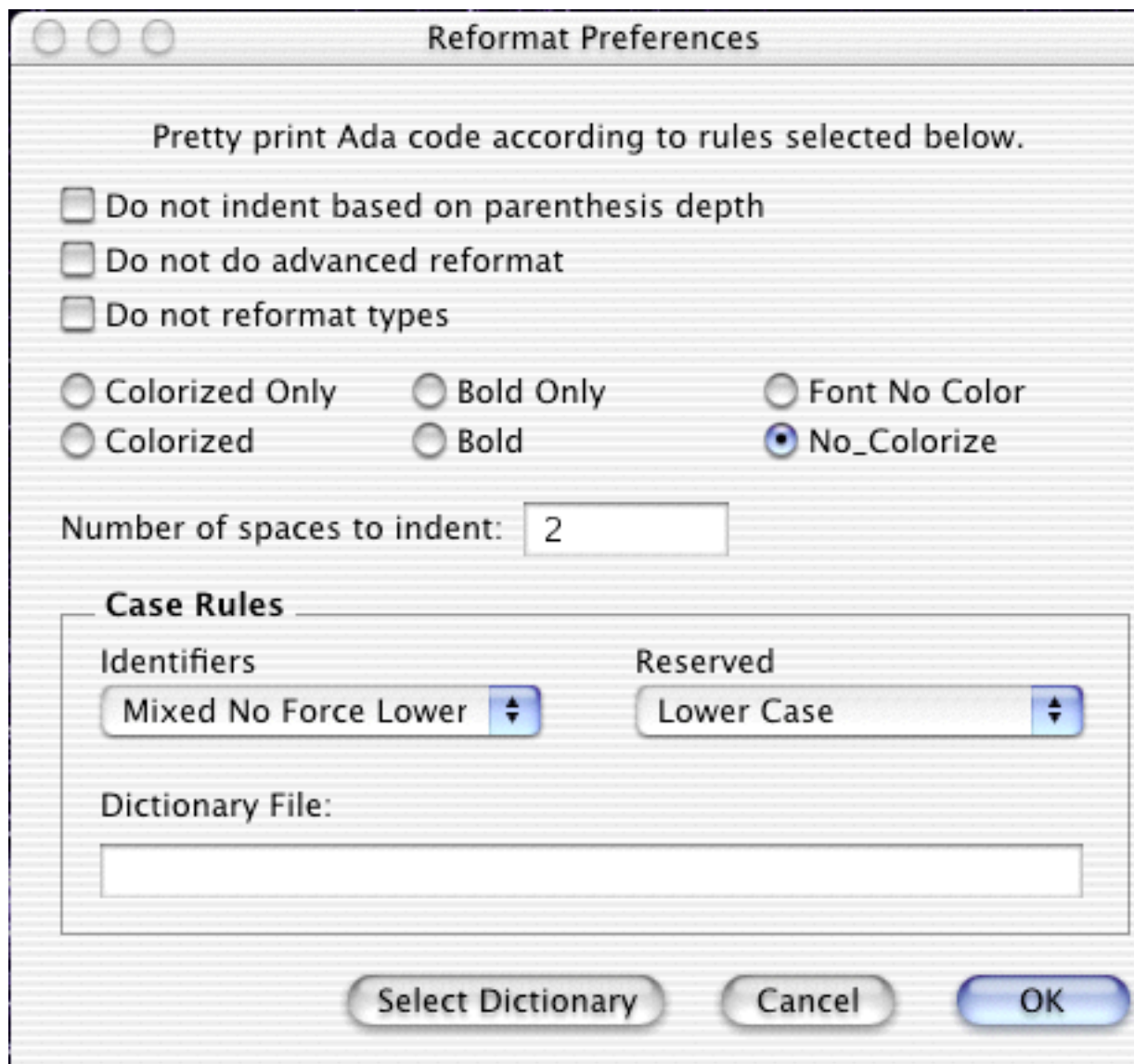
begin
  declare
    ND : TElement := (VRefNum => - 1, ParID => - 1, Name => "");
  begin
    ResultNonDefini := ND;
    null;
  end;
  return ResultNonDefini;
end;
end BasicDef;

```

L'outil Reformat est maintenant proposé sur le site Gnat-OSX avec une interface utilisateur : Reformat\_GUI.

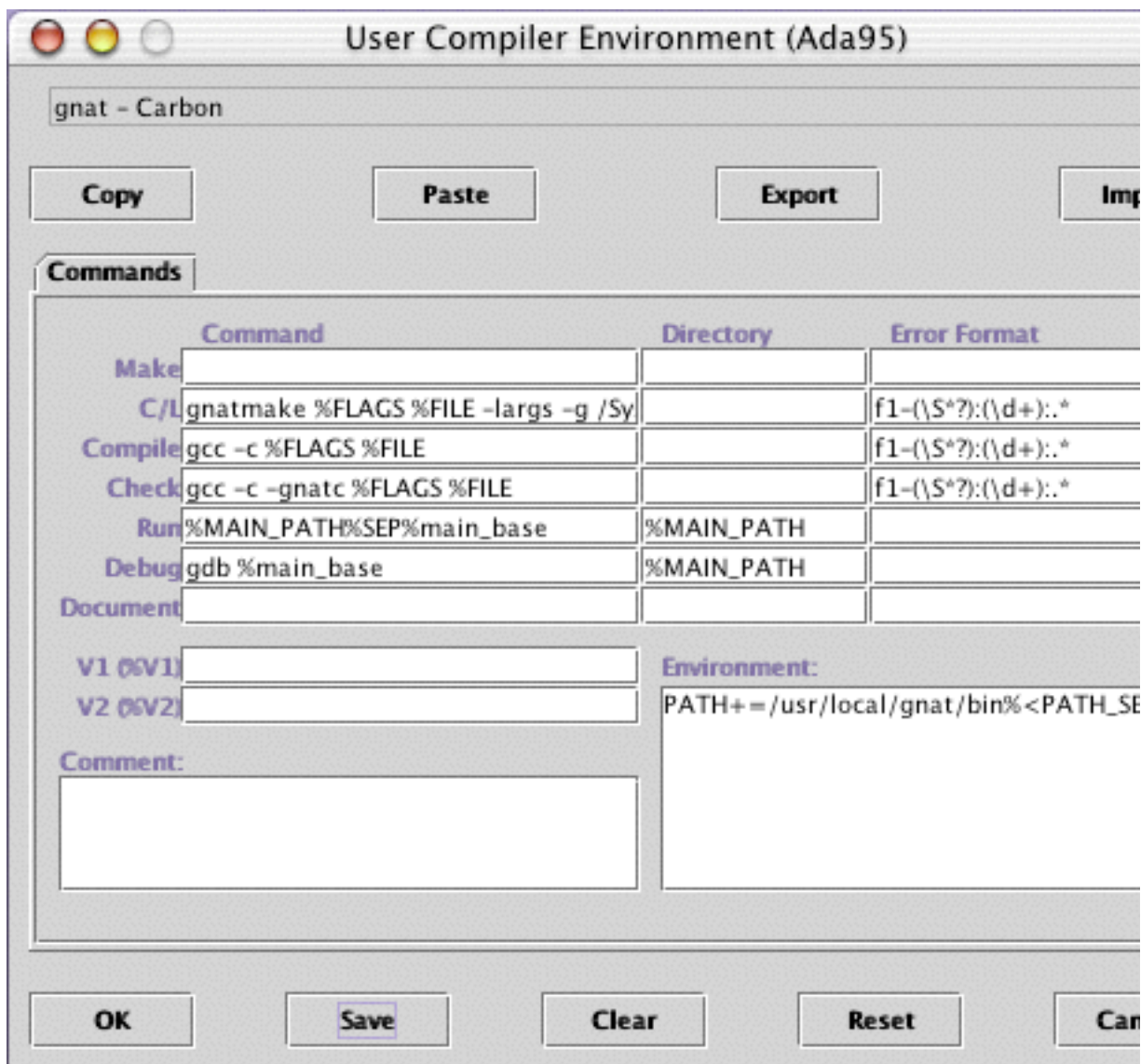
<http://www.adapower.net/macros/tools.html>

Voici les réglages que j'utilise :



### 3) Environnement de développement Ada jGrasp :

jGrasp est un environnement de développement écrit en Java. Il prend en compte plusieurs langages de programmation dont Ada, Java, Objective-C, C, C++. Il a la particularité d'afficher graphiquement en marge du code source la structure du programme. La possibilité de replier les structures de code est très pratique pour les longs programmes. Il permet aussi d'afficher graphiquement la "complexité" du source (voir le manuel utilisateur). Pour pouvoir lancer directement la compilation il est nécessaire de modifier quelques options de l'environnement dans le menu "Settings / Compilers Settings / global". Faites une copie de l'environnement Ada95 "gnat - generic" en "gnat - macosx" (par exemple) et éditer comme ci-dessous :



Ajouter "PATH+=/usr/local/gnat/bin%<PATH\_SEP>" dans la boîte "Environment".  
Pour pouvoir utiliser la localisation automatique des erreurs il convient de  
modifier le script de décodage des erreurs en "f1-(\S\*?):(\d+):.\*" dans la boîte  
"Error Format".

Sauvegardez, appliquez puis activez le menu "File / Save Current Workspace".  
Attention certains raccourcis s'obtiennent avec la touche contrôle et non  
Pomme.

#### 4) Les scripts utiles :

Conversion des retours à la ligne de type Mac (CR) en type Unix (LF) :  
(Reformat\_GUI le fait également)

```
#!/bin/tcsh
foreach f ($argv)
  set ft = $f:t
  echo Traduction (Mac/Unix) $ft
  cat $f | tr '\015' '\012' > $ft.tmp
  mv $ft.tmp $ft
end
```

Conversion des interfaces de programmation Pascal en interfaces Ada :

```
#!/bin/tcsh
foreach f ($argv)
  set ft = $f:t
  echo Traduction (Pascal/Ada) $ft
  set ft = `echo $ft | tr '[:upper:]' '[:lower:]'`
  gpcpp -ki -C -P -DTARGET_API_MAC_CARBON -DMWERKS=200
-DPOWERPC=TRUE -DMAC68K=FALSE -DPROCTYPE=TRUE -lang-pascal $f >
$ft:r.i
  p2ada < $ft:r.i > $ft:r.ads
  reformat -id Mixed_No_Force_Lower -rw Lower_Case -i 2 $f:r.ads $f:r.ads
end
```

Formatage des sources Ada :

```
#!/bin/tcsh
foreach f ($argv)
  echo Formatage $f
  reformat -id Mixed_No_Force_Lower -rw Lower_Case -i 2 $f $f
end
```

Note : les adresses Internet des utilitaires se trouvent sur le site Blady en page "Liens" sauf "gpcpp" qui se trouve sur la page "Créations" à l'article sur GPC.

Pascal Pignard Mars 2002, Décembre 2002.