

# Le module `drac`\*

Josselin Noirel et Yvon Henel

14 août 2008

## Résumé

Ce module définit `\DeclareRobustActChar` et `\ReDeclareRobActChar`.

On utilise `\DeclareRobustActChar` comme `\DeclareRobustCommand` mais elle protège le caractère actif, voir `ltxdefns.dtx` dans `/latex/base/`. Cela permet, par exemple, d'écrire ce caractère dans un fichier auxiliaire sans qu'il soit développé. Il est donc devenu « robuste » au sens des arguments « mobiles » de  $\text{\LaTeX}$ .

`\ReDeclareRobActChar` permet de redéfinir le caractère actif défini préalablement par `\DeclareRobustActChar` à la manière de `\renewcommand`.

## Abstract

This package provides `\DeclareRobustActChar` and `\ReDeclareRobActChar` macros.

You should have downloaded with this file a file `drac-en.dtx` and a file `drac.pdf` which give the English documentation of this package.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Utilisation</b>	<b>2</b>

## 1 Introduction

Lorsqu'elle apparut pour la première fois, le 8 avril 2004, sur `fr.comp.text.tex` dans le fil de discussion « la question à 100 euros », le nom de la macro était `\DeclareRobustActiveCharacter` qui avait l'avantage d'être totalement descriptif mais l'inconvénient d'être long. Pour alléger quelque peu le travail du codeur mais *surtout* pour éviter un joyeux bazar dans l'index de cette documentation, j'ai pris sur moi de la renommer `\DeclareRobustActChar`.

À la lecture du fil mentionné ci-dessus, on s'apercevra sans peine que ce que je — je c.-à-d. Y. Henel,  $\text{\TeX}$ nicien de surface — fis ce fut de passer commande d'un code que Josselin Noirel me fournit très aimablement et tout aussi efficacement.

Je fis mumuse un moment avec le code fourni par Josselin puis quelques mois plus tard la vie m'asséna un de ces coups dont la fréquentation assidue des classiques auraient du me prévenir mais, pour citer un de ses fameux (pseudo?) proverbe chinois, « l'expérience n'éclaire que le chemin parcouru ». Ce qui fait que même si Josselin, de son côté assez occupé avec la fin de ses études, me fit parvenir, dans un courriel de février 2005, une 2<sup>e</sup> version dans laquelle apparut `\ReDeclareRobActChar` dans un courriel de février 2005 je n'étais guère en état d'en tirer profit.

---

\*Ce document correspond au fichier `drac v1`, du 2008/08/14.

Puis le temps a passé... j'ai sorti la tête de l'eau et Josselin est parti vers d'autres cieux.

Le code existe et fonctionne. Alors, plutôt que de le laisser moisir dans un coin obscur de mon disque dur, et puisque j'ai passé quelques heures à mettre à jour mes extensions, je me suis décidé à lancer `drac` dans le vaste monde. Je me contente de fournir l'emballage et porter le bébé dans mes bras velus jusqu'au serveur CTAN le plus proche.

On aura compris que je ne m'engage pas du tout à assurer la maintenance de ce code si jamais le besoin s'en faisait sentir. J'arrive à peu près à comprendre ce que fait Josselin en suivant pas à pas ses explications mais je suis bien incapable de me hausser à de telles hauteurs. Il est donc temps maintenant de céder le clavier à Josselin.

Le  $\TeX$ nicien de surface

## 2 Utilisation

La macro `\DeclareRobustActChar` a la syntaxe habituelle des `\newcommand` et consœurs : elle admet une forme étoilée selon que le caractère actif acceptera ou non des arguments `\longs` (c.-à-d. un argument pourra comporter une fin de paragraphe) ; ensuite vient le caractère actif ; un nombre optionnel d'arguments, un argument optionnel qui est la valeur optionnelle du premier argument, enfin la définition proprement dite. Voilà les syntaxes valides :

```
\DeclareRobustActChar{<caractère actif>}[n]%  
  [⟨valeur par défaut du premier argument⟩]{⟨définition⟩}  
\DeclareRobustActChar{<caractère actif>}[n]{⟨définition⟩}  
\DeclareRobustActChar{<caractère actif>}{⟨définition⟩}  
\DeclareRobustActChar*{<caractère actif>}[n]%  
  [⟨valeur par défaut du premier argument⟩]{⟨définition⟩}  
\DeclareRobustActChar*{<caractère actif>}[n]{⟨définition⟩}  
\DeclareRobustActChar*{<caractère actif>}{⟨définition⟩}
```

Suite à quoi le caractère actif pourra être utilisé dans les arguments mobiles (dans les arguments de `\caption` par exemple) et dans les autres cas qui requièrent des commandes robustes (`\protected@edef`, `\protected@write`, etc.)

De même `\ReDeclareRobActChar` a la syntaxe de `\renewcommand` d'où les syntaxes possibles :

```
\ReDeclareRobActChar{<caractère actif>}[n]%  
  [⟨valeur par défaut du premier argument⟩]{⟨définition⟩}  
\ReDeclareRobActChar{<caractère actif>}[n]{⟨définition⟩}  
\ReDeclareRobActChar{<caractère actif>}{⟨définition⟩}  
\ReDeclareRobActChar*{<caractère actif>}[n]%  
  [⟨valeur par défaut du premier argument⟩]{⟨définition⟩}  
\ReDeclareRobActChar*{<caractère actif>}[n]{⟨définition⟩}  
\ReDeclareRobActChar*{<caractère actif>}{⟨définition⟩}
```

## Changements

v0		la première version publique . . . 1
	General : A failli être la première version publique . . . . . 1	Ajout de <code>\ReDeclareRobActChar</code> . . . . . 1
v0.1		v1
	General : A manqué de peu d'être	General : 1 <sup>re</sup> version publique . . . . 1