

Guide de l'utilisateur de `paresse`*

Yvon Henel[†]

7 octobre 2020

Résumé

Cette extension, reprenant un exemple de T. LACHAND-ROBERT dans [1], fournit un moyen de taper des lettres grecques isolées à l'aide du caractère `§` actif et redéfini. Au lieu de `\(\alpha\)` ou tape `§a` pour obtenir α .

Important : Il doit être chargé **après** `inputenc` si ce dernier est utilisé. De plus, il faut que le signe `§` soit une lettre pour `TEX`.

Depuis la version 4, on peut utiliser cette extension même dans un source codé en utf-8 avec `LATEX`, `LuaLATEX` ou `XƎLATEX`.

Abstract

The English documentation for the final user of the package `paresse` is available in the file `paresse-eng`.

Table des matières

1	Introduction	2
1.1	Pourquoi une 5 ^e version ?	2
1.2	Pourquoi une 4 ^e version ?	2
1.3	Pourquoi une 3 ^e version ?	3
2	Utilisation	3
3	Les options de l'extension	4
3.1	Commandes et environnement	4
3.2	Tableaux des macros	5
3.2.1	Codage originel de <code>paresse</code>	5
3.2.2	Codage de Sylvio LEVI	6
	Bibliographie	6

*Ce fichier décrit la version 5.0, dernière révision 2020/10/06. Édition *Adieu à skeyval*.

[†]E-mail : le.texnicien.de.surface@yvon-henel.fr

1 Introduction

Cette extension ne fournit qu'un accès « rapide et économique » aux lettres grecques qui s'obtiennent à l'aide d'une macro comme `\alpha` ou `\Omega`. Elle fournit un environnement et une commande qui permettent d'utiliser § pour taper ces lettres. Un `\ensuremath` nous dispense de nous placer explicitement — c'est-à-dire en tapant \$ \$ ou bien \ (\) ou encore \[\] ou tout autre chose ayant le même effet — en mode mathématique pour obtenir une lettre grecque.

L'idée de la méthode est due à T. LACHAND-ROBERT et est exposée dans [1]. Je n'ai fait qu'ajouter le `\ensuremath` bien agréable pour l'écriture de macros.

Bien entendu, on ne dispose pas de macros pour la minuscule omicron ni pour les majuscules alpha, beta... qui s'obtiennent à l'aide des latines romaines de même apparence¹. Je ne me suis pas senti le courage ni la force de fournir une solution qui permettent d'obtenir dans une formule baignant dans un texte en italique gras un alpha majuscule droit, romain, &c.

Pour finir cette introduction, glosons le nom de cette extension. *paresse* vient de ce que le signe § indique un « paragraphe » en ayant une forme lointainement apparentée au S et n'a donc aucun lien avec le *défaut* si fréquent, encore que, à bien y réfléchir...

1.1 Pourquoi une 5^e version ?

Sur mon brin de laurier, je dormais comme un loir en utilisant *paresse* presque quotidiennement quand il y a quelques jours, patatras ! Plus rien ne va, je me fais insulter rien qu'à charger l'extension. Je passe sur les détails : la faute en revient à une modification du noyau qui vient perturber le code acrobatique de *skeyval* . Comme la mise à jour de la dite extension semble extrêmement improbable, j'ai décidé de réparer *paresse* en recourant aux commandes de *expl3* .

Ce n'est donc que pour remettre en marche ce dont je me sers souvent que je viens soulever le capot de cette extension vieille de plus de 7 ans.

Avec cette 5^e version, apparaissent deux *sous*-extensions : *paresse-old* et *paresse-utf8* que l'on peut charger directement et qui offrent les mêmes options et les mêmes commandes. De fait, *paresse* charge l'une ou l'autre suivant la situation.

On utilisera *paresse-utf8* uniquement si le source est codé en utf-8 et que l'on compile avec *latex* c.-à-d. avec le moteur *T_EX* et le format *L^AT_EX* . Dans tous les autres cas de figure on utilisera *paresse-old* .

La documentation de *paresse* couvre donc l'utilisation des trois extensions.

1.2 Pourquoi une 4^e version ?

Je ne sais plus bien à quelle occasion — l'âge et les pannes de machine avec « crachage » de disque aidant — et encore moins quand — si ce n'est que cela remonte à plus d'un an maintenant² — Christian TELLECHEA me faisait part de son désir d'utiliser *paresse* avec des sources codées en utf-8 sous *L^AT_EX* — et non *X_YL^AT_EX* ou *LuaL^AT_EX* .

Une discussion s'engagea, Christian me fit parvenir du matériel en ordre de marche. Il me fit même cadeau de deux versions, la deuxième améliorant la reconnaissance du codage déclaré avec *inputenc* . Toutefois, je procrastinais. Il se peut que ma vie personnelle et mon travail aient interféré avec le développement de cette extension.

Enfin, voici la chose.

1. Je ne ferai aucune remarque sur les problèmes de codage que cela pose.

2. J'ai écrit cette section à l'occasion de la sortie de la 4^e version, en 2013.

Le plus nouveau devrait échapper à l'utilisateur de Lua \LaTeX ou X \LaTeX et même de \LaTeX chargeant `inputenc` avec une option comme `latin1` ou `latin9`. Cependant, désormais, on peut utiliser cette extension avec \LaTeX en chargeant `inputenc` avec l'option `utf8`.

J'ai profité de cette nouvelle version pour ajouter une macro en $\text{\S} : \text{\SZ}$ qui produit \S , symbole déjà accessible par \S , ce qui fait que je n'ose pas vraiment parler de « raccourci ».

1.3 Pourquoi une 3^e version ?

C'est par courriel que Claudio BECCARI m'a signalé très aimablement qu'il existait un codage de l'alphabet grec en lettres latines près de 15 ans avant que je ne commette cette extension. Ce codage était dû à Sylvio LEVI qui l'avait mis au point, alors qu'il dessinait les premières fontes grecques pour \TeX , en s'appuyant sur la correspondance entre les claviers américain et grec. Il me faisait remarquer à juste titre que si quelqu'un avait pris l'habitude du codage de Sylvio LEVI, il préférerait conserver ses habitudes pour paresse.

J'ai donc décidé de fournir un nouveau couple d'options exclusives l'une de l'autre que j'ai nommées, pour l'une, `LEGACY` — pour « héritage » — qui assure le codage originel de cette extension et qui est active par défaut et, pour l'autre, `LEVI` qui assure le codage de Sylvio LEVI.

J'en profite pour faire quelques changements cosmétiques : désormais toutes les macros internes *secrètes*³ ont un nom qui commence par \GA@ ; le fichier `.dtx` est réorganisé pour faciliter la tâche de traduction de la documentation.

2 Utilisation

On charge cette extension avec $\text{\usepackage{paresse}}$. Quand on utilise \LaTeX avec un codage 8-bits (comme `latin9` par exemple), on la charge **après** l'extension `inputenc` avec l'option `idoine`. Avec un \LaTeX moderne — post 2019 — et un source codé en utf-8, on n'a pas besoin de charger explicitement `inputenc`. Dans ce cas, `paresse` suppose que le source est codé en utf-8.

Dans tous les cas il faut que le « caractère » \S soit reconnu par \TeX comme une lettre.

On obtiendra le même comportement, à l'exception — signalée dans le paragraphe intitulé **restriction** page 3 — près, avec `inputenc` et l'option `utf8`.

Il n'y a pas ce genre de restriction quand on travaille avec Lua \LaTeX ou X \LaTeX sur un source codé en utf-8.

Par défaut, l'extension est chargée avec l'option `wild` ce qui signifie que les macros de la forme \Sa sont disponibles. Si l'on préfère, on peut passer à `paresse` l'option `TAME` avec $\text{\usepackage[tame]{paresse}}$. Il faut alors utiliser la commande \ActiveLaParesse ou l'environnement `ParesseActive` pour utiliser les macros en \S .

Lorsque la « paresse est active », il suffit de taper \Sa pour obtenir α . On a de même accès à toutes les lettres grecques auxquelles sont consacrées une macro comme \alpha , voyez 3.2.1 et 3.2.2. On obtient, de même, α^β avec $\text{\S a}^\text{\S b}$ lorsque \S est active.

Restriction On notera que les accolades sont optionnelles et qu'on obtient un résultat identique avec $\text{\S a}^\text{\S b}$, à moins que l'on utilise un source codé en utf-8 avec \LaTeX .

3. À la mode $\text{\LaTeX} 2_\epsilon$, pour la version 5.0 j'utilise en partie `expl3` et, dans ce cas, le préfixe `paresse`.

3 Les options de l'extension

<code>tame / wild</code>	— TAME s'oppose à WILD qui est l'option par défaut. Lorsque TAME règne, il faut un environnement ParesseActive ou une commande <code>\ActiveLaParesse</code> pour utiliser les macros à §.
<code>legacy / Levi</code>	— LEVI s'oppose à LEGACY qui est l'option par défaut. Avec l'option LEGACY on utilise le codage « originel » de paresse tel que le donne la table 3.2.1. Sinon, le codage est celui défini par Sylvio LEVI, cf. la table 3.2.2.
<code>ttau / ttheta</code>	— TTAU s'oppose à TTHETA qui est l'option par défaut. Lorsque TTHETA est active \$t donne θ dans le cas contraire \$t donne τ . En tous cas, θ est accessible par \$v et τ par \$y . Cette option est inopérante si l'option LEVI a été choisie. Remarque : quand l'option LEGACY a été choisie, Θ est obtenu par \$V de manière « régulière » et <i>également</i> par \$T quelle que soit l'option choisie. Dans le cas de l'option LEVI, \$V ne correspond à aucune lettre grecque.
<code>epsilon / varepsilon</code>	— EPSILON s'oppose à VAREPSILON qui est l'option par défaut. Avec EPSILON, \$e donne ϵ sinon \$e donne ε . — Se comportent comme le couple EPSILON, VAREPSILON les couples suivants THETA et VARTHETA ; PI et VARPI ; RHO et VARRHO ; SIGMA et VARSIGMA « \$s donne ς » ; PHI et VARPHI. Par défaut on a VAREPSILON, THETA, PI, RHO, SIGMA, VARPHI, WILD et LEGACY. Cela assure que cette version 3, se comporte, par défaut, comme la précédente.

3.1 Commandes et environnement

<code>\makeparesseletter</code>	Cette macro donne au « caractère » § le catcode d'une lettre. Après cela, on peut se servir de § dans un nom de macro, par exemple. C'est le pendant de <code>\makeatletter</code> .
<code>\makeparesseother</code>	Cette macro donne au caractère § le catcode <i>other</i> . C'est le « contraire » de la précédente. Cela équivaut au <code>\makeatother</code> . Cette macro est inactive avec un codage utf-8 sous L ^A T _E X. Elle n'aurait d'ailleurs pas vraiment de sens. Son utilisation produit un avertissement dans le fichier <code>.log</code> .
<code>\ActiveLaParesse</code>	Cette macro active le caractère § et permet ainsi d'accéder aux macros dont le nom commence par § comme \$a . Pour une liste de ces macros et leurs significations, voyez les tableaux 3.2.1 et 3.2.2.
<code>ParesseActive</code>	Dans cet environnement le caractère § est actif ce qui permet d'utiliser les macros en §. On utilisera cet environnement si l'on veut utiliser les macros quand on a chargé l'extension paresse avec l'option tame .

3.2 Tableaux des macros

3.2.1 Codage originel de paresse

C'est le codage actif lorsque l'on a choisit les options `LEGACY` et `TTHETA` qui sont des options par défaut.

Cette version 5.0 ajoute ς obtenu avec `\Sj`.

<code>\Sa</code>	α	<code>\Sb</code>	β	<code>\Sg</code>	γ	<code>\Sd</code>	δ
<code>\Se</code>	ε	<code>\Sz</code>	ζ	<code>\Sh</code>	η	<code>\Sv</code>	θ
<code>\Si</code>	ι	<code>\Sk</code>	κ	<code>\Sl</code>	λ	<code>\Sm</code>	μ
<code>\Sn</code>	ν	<code>\Sx</code>	ξ	<code>\Sp</code>	π	<code>\Sr</code>	ρ
<code>\Ss</code>	σ	<code>\Sy</code>	τ	<code>\Su</code>	υ	<code>\Sf</code>	φ
<code>\Sc</code>	χ	<code>\Sq</code>	ψ	<code>\Sw</code>	ω		
<code>\SG</code>	Γ	<code>\SD</code>	Δ	<code>\SV</code>	Θ	<code>\SL</code>	Λ
<code>\SX</code>	Ξ	<code>\SP</code>	Π	<code>\SS</code>	Σ	<code>\SU</code>	Υ
<code>\SF</code>	Φ	<code>\SQ</code>	Ψ	<code>\SW</code>	Ω	<code>\SZ</code>	\S

<code>\Sa</code>	α	<code>\Sb</code>	β	<code>\Sc</code>	χ	<code>\Sd</code>	δ	<code>\Se</code>	ε	<code>\Sf</code>	φ
<code>\Sg</code>	γ	<code>\Sh</code>	η	<code>\Si</code>	ι	<code>\Sj</code>	ς	<code>\Sk</code>	κ	<code>\Sl</code>	λ
<code>\Sm</code>	μ	<code>\Sn</code>	ν	<code>\So</code>		<code>\Sp</code>	π	<code>\Sq</code>	ψ	<code>\Sr</code>	ρ
<code>\Ss</code>	σ	<code>\St</code>	θ	<code>\Su</code>	υ	<code>\Sv</code>	θ	<code>\Sw</code>	ω	<code>\Sx</code>	ξ
<code>\Sy</code>	τ	<code>\Sz</code>	ζ	<code>\SA</code>		<code>\SB</code>		<code>\SC</code>		<code>\SD</code>	Δ
<code>\SE</code>		<code>\SF</code>	Φ	<code>\SG</code>	Γ	<code>\SH</code>		<code>\SI</code>		<code>\SJ</code>	
<code>\SK</code>		<code>\SL</code>	Λ	<code>\SM</code>		<code>\SN</code>		<code>\SO</code>		<code>\SP</code>	Π
<code>\SQ</code>	Ψ	<code>\SR</code>		<code>\SS</code>	Σ	<code>\ST</code>	Θ	<code>\SU</code>	Υ	<code>\SV</code>	Θ
<code>\SW</code>	Ω	<code>\SX</code>	Ξ	<code>\SY</code>		<code>\SZ</code>	\S				

Remarques : à l'exception de θ , τ , ψ et ς les lettres utilisées dans les noms des macros sont chargées de vertu mnémotechniques :-)) et les capitales grecques, quand elles diffèrent des capitales latines, s'obtiennent à l'aide de la majuscule correspondante.

3.2.2 Codage de Sylvio Levi

On active ce codage avec l'option LEVI.

<code>\Sa</code>	α	<code>\Sb</code>	β	<code>\Sg</code>	γ	<code>\Sd</code>	δ
<code>\Se</code>	ε	<code>\Sz</code>	ζ	<code>\Sh</code>	η	<code>\Sj</code>	θ
<code>\Si</code>	ι	<code>\Sk</code>	κ	<code>\Sl</code>	λ	<code>\Sm</code>	μ
<code>\Sn</code>	ν	<code>\Sx</code>	ξ	<code>\Sp</code>	π	<code>\Sr</code>	ρ
<code>\Ss</code>	σ	<code>\St</code>	τ	<code>\Su</code>	υ	<code>\Sf</code>	φ
<code>\Sq</code>	χ	<code>\Sy</code>	ψ	<code>\Sw</code>	ω	<code>\Sc</code>	ς
<code>\SG</code>	Γ	<code>\SD</code>	Δ	<code>\SJ</code>	Θ	<code>\SL</code>	Λ
<code>\SX</code>	Ξ	<code>\SP</code>	Π	<code>\SS</code>	Σ	<code>\SU</code>	Υ
<code>\SF</code>	Φ	<code>\SY</code>	Ψ	<code>\SW</code>	Ω	<code>\SZ</code>	\S

Le codage de Sylvio LEVI donne accès directement à `\varsigma` (ς) avec χ et ne diffère du codage originel que pour les lettres θ , τ , χ et ψ . Voici un résumé de ces différences :

lettres grecques	θ	τ	χ	ψ	Θ	Ψ	ς
codage originel	<code>\sv/St</code>	<code>\sy/St</code>	<code>\sc</code>	<code>\sq</code>	<code>\SV/ST</code>	<code>\SQ</code>	<code>\sj</code>
codage de S. LEVI	<code>\sj</code>	<code>\st</code>	<code>\sq</code>	<code>\sy</code>	<code>\SJ</code>	<code>\SY</code>	<code>\sc</code>

Références

- [1] T. LACHAND-ROBERT. *La maîtrise de T_EX et L^AT_EX*. Masson, Paris, Milan, Barcelone, 1995. ISBN : 2-225-84832-7.

Le TeXnicien de Surface scripsit.
