

The **huaz** package

v1.1 (2024/04/12)

Tibor Tómacs

tomacs.tibor@gmail.com

Abstract

In Hungarian there are two definite articles, “*a*” and “*az*”, which are determined by the pronunciation of the subsequent word. The definite article is “*az*”, if the first phoneme of the pronounced word is a vowel, otherwise it is “*a*”. The **huaz** package helps the user to insert automatically the correct definite article for cross-references and other commands containing text. Thus, if these change, the definite articles will also change accordingly.

Sorry, the documentation is in Hungarian. This is because only Hungarian-speaking users are interested.

1. Bevezetés

A magyar nyelvben a határozott névelő egy szó előtt aszerint „*a*” vagy „*az*”, hogy a szó kiejtve mássalhangzóval vagy magánhangzóval kezdődik. Például „*a szék*” vagy „*az asztal*”. Ez egyszerűnek tűnik, de gondoljon a következő esetekre:

- Például ha oldalszámra akar hivatkozni, akkor a `\pageref` parancsot szoktuk használni. Ha ez elé akar határozott névelőt rakni, akkor nem mindegy, hogy mi az oldalszám. Például „*az 1. oldal*”, „*a 2. oldal*”, „*az 5. oldal*”, „*a 10. oldal*”. Így egyáltalán nem biztos, hogy az „*a \pageref{<kulcs>}*” kód megfelelő eredményt ad. Hasonló a probléma minden keresztivatkozás esetén (`\ref`, `\pageref`, `\eqref`, `\cite`).
- Az is fontos, hogy egy kifejezés római szám vagy sem. Például „*az V. fejezet*”, ha *V* római szám, azaz 5 a jelentése. Ugyanakkor „*a V. fejezet*”, ha *V* betűt jelöl vagy alfanumerikus szám esetén a 22 értéket (*V* az angol ábécében a 22. betű).
- Néhány mássalhangzó speciális tulajdonságú. Például „*az M betű*”, de „*a Magyar Közlöny*”; „*az Ny betű*”, de „*a Nyugdíjfolyósító Igazgatóság*”; stb.

Először megemlítünk két eddigi megoldást a határozott névelők automatikus kezelésére, kiemelve azok hibáit és hiányosságait.

1.1. A magyar.ldf

Egyik lehetőség a `babel` csomag `magyar` opcióját beállító `magyar.ldf` fájl használata. Ennek teljes dokumentációja elérhető innen: <https://math.bme.hu/latex/magyarldf-doc.pdf>.

Az ebben definiált `\az`, `\aref`, `\apageref`, `\acite`, stb. parancsok alapvetően jól használhatóak, de van néhány fontos hiba, melyek a `huaz` csomag megírását motiválták:

- Az `\eqref` parancsnak nincs névelős verziója. Helyette az `\aref{<kulcs>}` használható, de ez dőlt betűs környezetben nem álló betűs eredményt ad, mint az `\eqref` esetében. Másik lehetőség, ami ezt a problémát megoldja, az `\az{\eqref{<kulcs>}}` parancs. Ugyanakkor egyik megoldás sem kezeli a római számozású vagy a `\tag` parancssal címkézett egyenleteket.
- Az `\az{\ref{<kulcs>}}` és `\az{\pageref{<kulcs>}}` nem kezeli a római számozású hivatkozásokat.
- Az előző hiba akkor is fennáll, amikor az `\az` parancsban egy római számmal kezdődő szöveg van. Például `\az{V.~osztály}` eredménye „*a V. osztály*”.
- Amennyiben nem kereszthivatkozáshoz, hanem egy szöveghez, vagy valamilyen szöveget tároló parancshoz kell automatikus névelő, akkor az ékezetes betűket UTF-8 kódolás esetén rosszul detektálja. A hiba alapvető oka, hogy az UTF-8 kódolású karakterek nem egy, hanem több bájtton vannak kódolva, amit a `magyar.ldf` nem vesz figyelembe. Ezért például `\az{ágy}` eredménye „*a ágy*”, mert nem betűként érzékeli az „*á*” betűt, így azt sem tudja, hogy ez magánhangzó. Meglepő módon `\az{száz}` eredménye „*az száz*” is rossz. Ennek az az oka, hogy mivel számára az „*á*” nem betű, ezért az „*sz*” betűt különállónak tekinti, ami elé valóban „*az*” kell.
- Hibának ugyan nem tekinthető, de mindenesetre kényelmetlen, hogy például `\az{\textbf{N betű}}` esetén a `\textbf` parancs megzavarja a szó betűinek detektálását, így a névelő helytelenül „*a*” lesz.

1.2. A nevelok csomag

2015-ben készült a `nevelok` csomag, ami szintén az automatikus határozott névelők kezelésére készült. A szerzője valószínűleg nem ismerte a `magyar.ldf` már akkor létező lehetőségeit, ezért készíthette ezt a jóval kevesebbet tudó csomagot.

2022-ben telepített \TeX -rendszeren tesztelve azt tapasztaltam, hogy nem kezeli az ékezetes betűk semmilyen formáját és sok esetben ad kereszthivatkozásoknál is rossz névelőt. Az utóbbinak az az oka, hogy közvetlenül a `\ref` parancsot vizsgálja, ami nem kifejthető parancs.

Ezen dokumentáció elkészültéig semmilyen fejlesztése nem történt. Jelen állapotában nem javasolt ennek a csomagnak a használata.

1.3. A huaz csomag célja és működése

A **huaz** csomag célja segíteni a felhasználót abban, hogy a kereszthivatkozásokhoz és szöveget tartalmazó parancsokhoz automatikusan beszúrja a megfelelő határozott névelőt. Így ha ezek változnak, akkor a határozott névelők is megfelelően változnak.

A másik cél ezt úgy megtenni, hogy a `magyar.ldf` és a `nevelok.sty` hibáit és hiányosságait korigálja illetve pótolja.

A **huaz** csomag a következő esetekben illeszt az adott szöveg elé „az” névelőt:

1. Az első betű magánhangzó (legyen az kis- vagy nagybetű, ékezetes vagy sem, magyar vagy sem, UTF-8, ISO-8859-2 karakterrel vagy repülő ékezettel megadva).
2. Az első betű olyan kis vagy nagy mássalhangzó, amit önmagában kiejtve magánhangzóval kezdünk (például F, L, M, stb.) míg a második karakter (ha van) nem betű, hanem szám, írásjel vagy szóköz. Például „M-10”. Ide soroltunk néhány nem magyar ékezetes mássalhangzót is. Például „Ň.1”.
3. Az első két karakter egy olyan kis vagy nagy kétjegyű mássalhangzó, amit önmagában kiejtve magánhangzóval kezdünk (például Ny, Ly, Sz, stb.) míg a harmadik karakter (ha van) nem betű, hanem szám, írásjel vagy szóköz. Például „NY betű”.
4. Az első karakter 5.
5. 1, 4, 7 vagy 10 jegyű számmal kezdődik és az első számjegy 1 (egy, ezer, egymillió, egymilliárd).

Ha a szó elején található karakterek római számként is értelmezhetőek, akkor megválasztható, hogy azt konvertálja arab számra és a névelőt ahhoz határozza meg vagy sem. Például „XII/A” esetén.

2. A huaz csomag használata

A **huaz** csomagot a szokott módon kell betölteni:

```
\usepackage{huaz}
```

Csomagopciók nincsenek. UTF-8 (`utf8`) és ISO-8859-2 (`latin2`) kódolású forrásfájl esetén működik, de a repülő ékezeteket is jól kezeli. Kompatibilis a `pdflatex`, `xelatex` és `lualatex` fordítókkal is. Például `pdflatex` fordítóval a következő betöltés megfelelő:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{huaz}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
```

```
\usepackage[magyar]{babel}
\begin{document}
...
\end{document}
```

A `hyperref` csomaggal is kompatibilis, így az is betölthető a **huaz** mellé.

A `magyar.ldf defaults=hu-min` opciója bekapcsol néhány olyan opciót is, amely elérhetővé teszi a saját automatikus névelőparancsait. Ezt ugyan nem kötelező, de hasznos a **huaz** csomag használata esetén kikapcsolni. Ehhez a

```
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
```

sor helyett használja a következőt:

```
\PassOptionsToPackage{defaults=hu-min,az=no,
shortrefcmds=no,hunnewlabel=no}{magyar.ldf}
```

A `xelatex` és `lualatex` fordítók esetén `babel` helyett a

```
\usepackage{polyglossia}
\setdefaultlanguage{hungarian}
```

is használható.

A **huaz** csomag felhasználja az `xstring`, `refcount` és `iftex` csomagok szolgáltatásait, így ezek is betöltődnek. Másrészt felhasználásra kerül néhány 2021. október 10-én bevezetett változás a \LaTeX kerneljében, így csak az ezután telepített rendszereken működik megfelelően a csomag.

2.1. Parancsok

`\az{<szöveg>}`

A `<szöveg>` elé a megfelelő határozott névelő kerül kisbetűs alakban. Ha a `<szöveg>` elején római számként értelmezhető karakterek vannak, akkor a névelő az arab megfelelőjéhez igazodik. Például

```
Idén \az{V.B}~osztály rendezi a farsangot.
```

eredménye „Idén az V.B osztály rendezi a farsangot.”.

A `<szöveg>` lehet szöveget tároló parancs is. Például

```
\newcommand{\osztaly}{V.B}
Idén \az{\osztaly}~osztály rendezi a farsangot.
```

eredménye „Idén az V.B osztály rendezi a farsangot.”.

A `<szöveg>` tartalmazhat szövegformázó parancsokat is (bővebben lásd a 2.3. alszakaszban). Például

```
\newcommand{\osztaly}{V.B}
Idén \az{\textbf{\osztaly}}-osztály rendezi a farsangot.
```

eredménye „Idén az **V.B** osztály rendezi a farsangot.”.

A `<szöveg>` lehet standard kereszthivatkozás is (`\ref`, `\pageref`, `\eqref`, `\cite`).
Például

```
\section{Cím}\label{seca}
\section{Cím}\label{secb}
\az{\ref{seca}}.~szakaszban, \az{\textbf{\ref{secb}}}.~szakaszban
```

eredménye „az 1. szakaszban, a **2.** szakaszban”. Ha a `section` számláló kiírása római számozásra van állítva a `\renewcommand{\thesection}{\Roman{section}}` paranccsal, akkor az előző kód eredménye „az I. szakaszban, a **II.** szakaszban”.

Korlátozások

1. A `<szöveg>` elején csak a `\ref`, `\pageref`, `\eqref`, `\cite` kereszthivatkozásokkal működik helyesen. Például a `hyperref` csomag `\ref*` és `\pageref*` parancsai nem működnek közvetlenül `<szöveg>`-ként, de ez megoldható a később ismertetett `\az*` paranccsal.
2. A `<szöveg>` elején a `\cite` parancs jól működik a `natbib` csomaggal és `bibtex` esetében is. A `biblatex` csomag használata esetén akkor működik jól, ha a `style` vagy `citestyle` opciók értéke `numeric`, `numeric-verb`, `alphabetic`, `alphabetic-verb` vagy `authoryear`. Akkor is jól működik, ha nem adjuk meg a `style` illetve `citestyle` opciók egyikét sem.
3. A pdf vázlatfájába nem lehet `\az` paranccsal szöveget beilleszteni. Tehát például a következő kód nem ad helyes vázlatfát, ha `hyperref` vagy `bookmark` csomagot használ:

```
\section{... \az{\ref{sec}} ...}
```

Viszont a cím a szövegben, fejlécben és a tartalomjegyzékben jól fog megjelenni. A problémát a később ismertetett `\azsaved` paranccsal lehet megoldani.

`\az*{<szöveg>}`

Ugyanaz mint a `*` nélküli esetben, de ekkor csak a névelő kerül kiírásra. Például `hyperref` csomag használatával:

```
\section{Cím}\label{sec}
\az*{\ref{sec}}~\ref*{sec}.~szakaszban
```

`\azv{<szöveg>}`

Ugyanaz mint `\az{<szöveg>}`, de ha a `<szöveg>` elején római számként értelmezhető karakterek vannak, akkor a névelő nem az arab megfelelőjéhez igazodik, hanem mint egyszerű karakterekhez. Például

```
\renewcommand{\thesection}{\Alph{section}}
\setcounter{section}{21}
\section{Cím}\label{sec}
\az{\ref{sec}}.~szakaszban, \azv{\ref{sec}}.~szakaszban
```

eredménye „az V. szakaszban, a V. szakaszban”, mert az első esetben a V betűt római számként értelmezte, a második esetben pedig nem. Mivel most alfanumerikus számra van beállítva a `section` számláló, ezért a második eset a jó.

`\azv*{<szöveg>}`

Ugyanaz mint a `*` nélküli esetben, de ekkor csak a névelő kerül kiírásra.

A parancsok nevében az „a” betű kicserélhető „A” betűre. Ekkor a határozott névelő nagybetűvel fog kezdődni, ami mondatok elején szükséges.

`\Az{<szöveg>}`
`\Az*{<szöveg>}`
`\Azv{<szöveg>}`
`\Azv*{<szöveg>}`

Például

```
\section{Cím}\label{sec}
\Az{\ref{sec}}.~szakaszban áttekintjük a téma rövid történetét.
```

eredménye „Az 1. szakaszban áttekintjük a téma rövid történetét.”.

`\azsaved`

Amikor kiadja az előző parancsok bármelyikét, akkor generálódik egy kifejthető `\azsaved` parancs is. Ennek eredménye az a határozott névelő amelyik a szó elé kell.

Amikor `hyperref` vagy `bookmark` csomagot használ, akkor ahogyan azt korábban említettük, a következő kód esetén a cím, fejléc, tartalomjegyzék rendben lesz, de a pdf vázlatfája nem:

```
\section{\Az{\ref{sec}}...}
```

Ez az `\azsaved` paranccsal a következő módon oldható meg:

```
\section{\texorpdfstring{\Az{\ref{sec}}...}{\azsaved~\ref{sec}...}}
```

Ekkor a vázlatfába az

```
\azsaved~\ref{sec}...
```

kód kerül, ami már helyes eredményt ad. Ez a probléma a `\texorpdfstring` helyett az

`\aznotshow`

paranccsal egyszerűbben is megoldható. Ugyanis ezt helyezve az `\az` (vagy bármelyik verziója) elé, az eredmény nem jelenik meg, csak az `\azsaved` generálódik le a megfelelő határozott névelővel. Így

```
\aznotshow\Az{\ref{sec}}
\section{\azsaved~\ref{sec}...}
```

szintén helyes eredményt ad a vázlatfában is.

2.2. Rövidítések

```
\aref{<kulcs>}      ≡ \az{\ref{<kulcs>}}
\aref*{<kulcs>}    ≡ \az*{\ref{<kulcs>}}
\avref{<kulcs>}    ≡ \azv{\ref{<kulcs>}}
\avref*{<kulcs>}  ≡ \azv*{\ref{<kulcs>}}

\aeqref{<kulcs>}   ≡ \az{\eqref{<kulcs>}}
\aeqref*{<kulcs>} ≡ \az*{\eqref{<kulcs>}}
\aveqref{<kulcs>} ≡ \azv{\eqref{<kulcs>}}
\aveqref*{<kulcs>} ≡ \azv*{\eqref{<kulcs>}}

\apageref{<kulcs>} ≡ \az{\pageref{<kulcs>}}
\apageref*{<kulcs>} ≡ \az*{\pageref{<kulcs>}}
\avpageref{<kulcs>} ≡ \azv{\pageref{<kulcs>}}
\avpageref*{<kulcs>} ≡ \azv*{\pageref{<kulcs>}}

\acite[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \az{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}
\acite*[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \az*{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}
\avcite[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \azv{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}
\avcite*[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \azv*{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}
```

A paranccsok nevében az első „a” betű kicserélhető „A” betűre. Ekkor a határozott névelő nagybetűvel fog kezdődni, ami mondatok elején szükséges:

```
\Aref \Avref \Aeqref \Aveqref \Apageref \Avpageref \Acite \Avcite
```

Például

```
\section{Cím}\label{seca}
\section{Cím}\label{secb}
\Aref{seca}.~és \Aref{secb}.~szakaszban
```

eredménye „Az 1. és a 2. szakaszban”.

2.3. A huaz kapocs

Amikor a **huaz** csomag megkeresi a megfelelő határozott névelőt, akkor a szövegben található kereszthivatkozásokat lecseréli a kifejthető verziójukra, továbbá a

kifejtés során a formázó parancsokat (`\emph`, `\textbf`, `\small`, stb.) figyelmen kívül hagyja. Emiatt lehetséges, hogy például a következő kódok működnek:

```
\newcommand{\osztaly}{V.B}
Idén \az{\textbf{\osztaly}}-osztály rendezi a farsangot.
```

```
\section{Cím}\label{sec}
\az{\textbf{\ref{sec}}}
```

`\AddToHook{huaz}{<kód>}`

A **huaz** csomag azon szövegformázó parancsokat hagyja figyelmen kívül, melyek egy előre definiált listában szerepelnek. Amennyiben olyan formázó parancsot használ, ami még ezen a listán nem található, akkor az a felhasználó által is bővíthető. Például

```
\newcommand{\myfont}[1]{\usefont{T1}{yv1d}{m}{n}#1}
\newcommand{\mytext}{X.A-osztály}
\az{\myfont{\mytext}}
```

esetén a `\myfont` nincs ezen a listán, ezért ez a kód hibával fordul. A listára az

```
\AddToHook{huaz}{\def\myfont{}}
```

módon vehető fel. Ezzel azt érjük el, hogy a névelő megállapítása során a `\myfont` parancs ne jelentsen semmit. Tehát a következő kód már megfelelően működik:

```
\AddToHook{huaz}{\def\myfont{}}
\newcommand{\myfont}[1]{\usefont{T1}{yv1d}{m}{n}#1}
\newcommand{\mytext}{X.A-osztály}
\az{\myfont{\mytext}}
```

Az előző eset a `huaz` kapocs nélkül így is megoldható:

```
\newcommand{\myfont}[1]{\usefont{T1}{yv1d}{m}{n}#1}
\newcommand{\mytext}{X.A-osztály}
\az*{\mytext}~\myfont{\mytext}
```

Ha a `\myfont` a `\mytext` definíciójában szerepel, akkor a `huaz` kapocs használata nem kerülhető meg:

```
\AddToHook{huaz}{\def\myfont{}}
\newcommand{\myfont}[1]{\usefont{T1}{yv1d}{m}{n}#1}
\newcommand{\mytext}{\myfont{X.A-osztály}}
\az{\mytext}
```

Kereszthivatkozásnál, ha maga a számozás van standardtól különböző parancssal formázva, akkor a `huaz` kapocs használata szintén nem kerülhető meg. Például


```
\AddToHook{huaz}{\def\myfont{}}
\DeclareRobustCommand{\myfont}[1]{\usefont{T1}{yv1d}{m}{n}#1}
\renewcommand{\thesection}{\myfont{\arabic{section}}}
\section{Cím}\label{sec}
\aref{sec}
```

Itt a `\myfont` azért lett erős parancsként definiálva, mert mozgó argumentumba kerül. Az előző példa esete ritkán fordul elő, mert a számláló típusának megadásánál (arab, római, stb.) nem szokás szövegformázó parancsot is alkalmazni.