DUTCH LINGUISTIC GUIDE

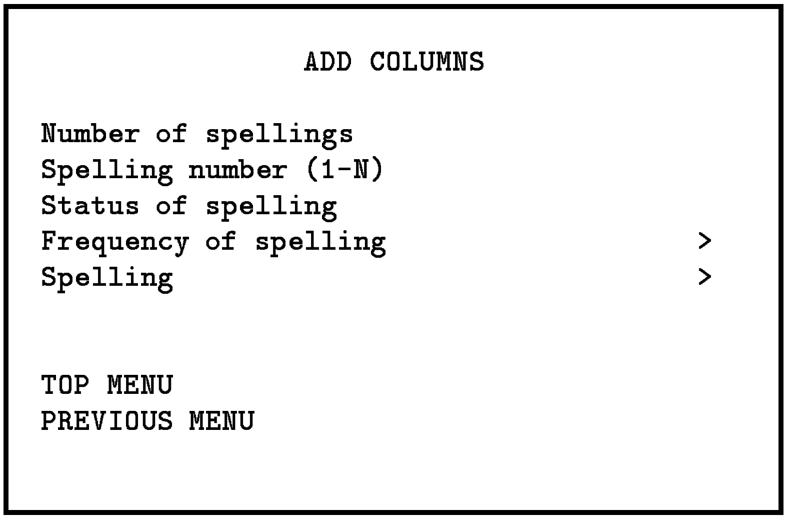
There ain't a lady livin' in the land
As I'd swap for my dear old Dutch
— **ALBERT CHEVALIER** (1861–1923) (Music Hall song)

CONTENTS

1 DUTCH ORTHOGRAPHY

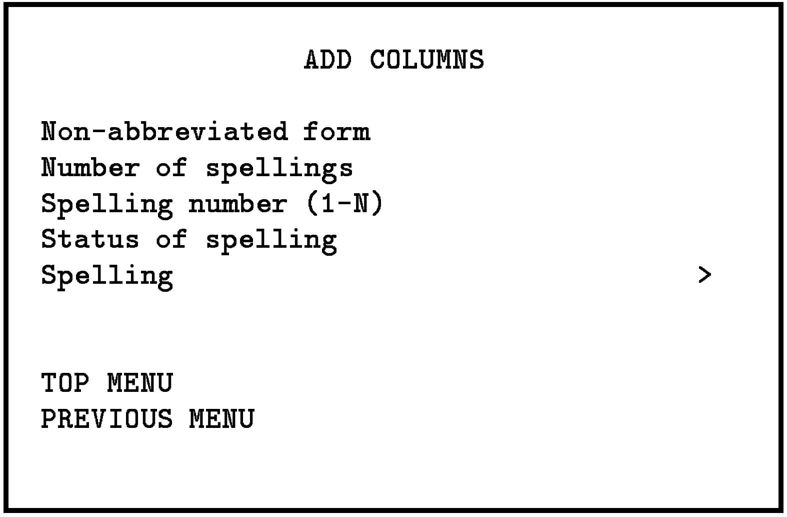
Detailed and varied information is available on the orthographic forms of lemmas (both headwords and stems) and wordforms, as well as abbreviations and INL corpus types. You can choose from a range of transcriptions: they can be syllabified or unsyllabified, they can include or omit *diacritics* (as explained below), or, in some cases, they come with the order of the letters reversed, or with the letters sorted alphabetically. In addition, there are columns which tell you the number of letters or syllables a particular transcription contains.

This FLEX window is the menu you see for a lemma or a wordform lexicon when you choose the Orthography option of the first `ADD COLUMNS` menu:



```
                  ADD COLUMNS

Number of spellings
Spelling number (1-N)
Status of spelling
Frequency of spelling                         >
Spelling                                      >


TOP MENU
PREVIOUS MENU
```

If your lexicon is of type Abbreviations, then the spelling frequency columns become unnecessary, while a column with the meaning of the abbreviation is required, so that the `ADD COLUMNS` menu you see for an abbreviation lexicon after selecting `Orthography` looks like this:



```
                    ADD COLUMNS

Non-abbreviated form
Number of spellings
Spelling number (1-N)
Status of spelling
Spelling                                >


TOP MENU
PREVIOUS MENU
```

If you are using an INL corpus type lexicon, then there is one spelling column available. It contains the words exactly as they were found in the corpus itself, with no alterations.

1.1 NUMBER OF SPELLINGS

This option in the `ADD COLUMNS` menu is a column which tells you how many ways each lemma, wordform or abbreviation (according to the type of lexicon you are using) can be spelt. For the lemma *frequenteren*, this column has the value 2, which means there are two possible ways of spelling it. Unless you construct a restriction on your lexicon, *frequenteren* will occur twice: one row using the form *frequenteren*, the other using the form *frekwenteren*.

This column is particularly useful when you want to identify words which have spelling variants. To exclude from your lexicon all items which only have one possible spelling, containing instead those which can be spelt in a number of ways, you can construct a expression restriction which simply states that the number of spellings must be greater than 1: `OrthoCnt > 1`.

The FLEX name and description of this column are as follows:

OrthoCnt (OrthoCntLemma) `Number of spellings`

1.2 SPELLING NUMBER

Just as the very first available `ADD COLUMNS` option is a number which uniquely identifies each lemma, wordform, or abbreviation (according to the type of lexicon you are using), so this column uniquely identifies every spelling to be found for each lemma, wordform, or abbreviation.

If you are using a lemma lexicon, the spelling variants are given in the form of headwords (with or without syllable markers), stems (with or without syllable markers) or abstract stems. For example, the verb *frequenteren* has two spellings: first the preferred form *frequenteren*, and second an alternative form *frekwenteren*. These have the spelling numbers 1 and 2 respectively. If you use a syllabified stem representation in place of the plain headword representation, spelling 1 takes the form *fre-quen-teer* and spelling 2 the form *fre-kwen-teer*.

This means you can use the universal sequence number to identify a particular lemma (or abbreviation, or wordform) and then the spelling number to identify the different individual spellings used for each lemma. Moreover, the spelling number allows you to identify quickly the 'preferred' spelling (as laid down in the *Woordenlijst van de Nederlandse Taal*), because such forms are always first in the list: for every lemma, wordform, or abbreviation the number 1 spelling is the 'preferred' form.

One important point to remember is that the spelling number can be used to eliminate unwanted rows from your lexicon. If you only want to see one spelling for each lemma (or whatever), you should construct a restriction which states that only rows with a spelling number equal to 1 are to be included (in the form `OrthoNum = 1`). If you don't do this, you usually end up with lexicons that are too long because they needlessly repeat certain pieces of information. Take the example *frequenteren* again, only this time imagine you want to know its pronunciation rather than the various ways it can be spelt. You create a lexicon with three columns, one giving the spelling number, one giving the orthography of the stem, and one giving the pronunciation of the stem. Without the restriction, FLEX returns two rows for *frequenteren*:

Spelling Number	**Stem**	**Pronunciation**
1	`frequenteer`	`fr@-kwEn-te:r`
2	`frekwenteer`	`fr@-kwEn-te:r`

This is unnecessary, since you are interested only in the pronunciation. The extra row merely gives you a spelling variant while the pronunciation remains the same. When you include the restriction `OrthoNum = 1`, however, only the row with the preferred spelling is included:

Spelling Number	**Stem**	**Pronunciation**
1	`frequenteer`	`fr@-kwEn-te:r`

And of course the more lemmas your lexicon contains, the greater the number of eliminated lines becomes, simply as a consequence of adding this important restriction.

If you are particularly interested in spelling variation, then do not add this `OrthoNum = 1` restriction: that way you get to see all the variant orthographic forms of each lemma. Otherwise, whenever you just want to use a simple orthographic transcription as a means of representing the lemma in your lexicon, always remember to insert it.

The FLEX name and description of this column are as follows:

***OrthoNum* (*OrthoNumLemma*)** `Spelling number`

1.3 STATUS OF SPELLING

For every different spelling there is a code which tells you whether it is the preferred spelling (`P`), a non-preferred (but still standard) spelling (`N`), or an informal, non-standard spelling (`I`). This applies to lemmas, wordforms, and abbreviations. Preferred and non-preferred forms are all standard forms, as given in the *Woordenlijst van de Nederlandse Taal*. The preferred form is the one you might expect to see most often, or which is considered the most acceptable form. The preferred form is always the first form given; that is, its spelling number is always `1`. Other informal, non-standard forms are included provided they occur in the INL 42-million word corpus. Often, they simply include or omit hyphens in a manner different from that prescribed in the *Woordenlijst*, as this table shows:

Spelling type	Status code	Example
Preferred	`P`	*preoccupatie*
Non-preferred	`N`	*preokkupatie*
Informal	`I`	*pre-occupatie*

Table 1: Orthographic status codes for Dutch spellings

The FLEX name and description of this column are as follows:

OrthoStatus (OrthoStatusLemma) `Status of spelling`

1.4 FREQUENCY OF SPELLING

There are figures available which tell you how frequently each spelling of each lemma or wordform occurs in the INL corpus, along with deviation figures which give a range of error for each frequency. They differ from the main frequency figures in that they are specific to one spelling, whereas the frequency columns proper refer to the more general frequency counts for the whole lemma (or whatever).

To arrive at these figures, the number of times every string in the INL corpus occurs must be counted. The next step is to identify which strings belong with which lemmas – a process known as *disambiguation*. Usually this is a straightforward task – a verbal wordform such as *gelezen* clearly belongs to the lemma *lezen*. However, it is sometimes possible to link one particular string to more than one lemma – the word *regent*, for example, can be the verb *rain* or the noun *ruler*. It is possible to check every occurrence of *regent* in the corpus and work out exactly how many belong to *rain* and how many to *ruler*. To a certain extent this can be done by computer program, but CELEX undertook the task by hand – reading occurrences in context and then deciding to which lemma the ambiguous string belongs. This approach clearly requires more time, but the investment yields a much more dependable result. The problem is, though, that the words which require disambiguation—and there are approximately 10,500 of them—are usually very frequent. Disambiguating all the occurrences of just one word could involve reading thousands of corpus sentences. To avoid this, a random sample of occurrences is taken from the corpus, up to a maximum of 100 (whenever the frequency is greater than 100). Disambiguating such a set produces a simple ratio which can be used to calculate the final frequency figure. Thus imagine that *regent* occurs 1320 times, and that after examining 100 occurrences of the word in the INL corpus, it seems that 86 out of the hundred meant *rain*, and the remaining 14 meant *ruler*. The ratio of one meaning to the other is thus 0.86 : 0.14. The frequency of *regent* (that is, the verb *rain*) is then 1320 multiplied by 0.86 – which is 1135, while the frequency of *regent* (that is, the noun *ruler*) is 1320 multiplied by 0.14, which is 185.

However, the story is still not complete. Occasionally it is impossible to decide which lemma a particular spelling belongs to. The plural noun *agenten* is one such word, since it can be the plural of *agente* (policewoman) or the plural of *agent* (policeman). In such cases a safe, unambiguous decision cannot be made, and the ratio is said to be 0.5 : 0.5. If there are three possible options, the ratio is 0.3333 : 0.3333 : 0.3333, and so on.

The result of this work is that you have an accurate frequency figure for each spelling of each lemma, wordform, or abbreviation in the database, and that figure is contained in this column, the FLEX name and description of which is as follows:

INLSpellFreq
(INLSpellFreqLemma)

`Spelling frequency, INL 42m word corpus`

How accurate are the figures in the ***INLSpellFreq*** column? The answer is that if there are no ambiguities to be resolved, then the figures are naturally completely accurate. This is true for most of the words in the database. From the above description, though, it's clear that in certain cases, a degree of approximation is included. When ambiguities do occur, then it is possible to calculate a deviation figure which specifies the range of error to an accuracy of at least 95%. This is the required formula:

$$N \times 1.96 \times \sqrt{\frac{p\,(1-p)}{n} \times \frac{N-n}{N-1}}$$

where N is the frequency of the word as a whole, n is the total number of words which were disambiguated in the random sample, and p is the ratio figure for the word when it belongs to one particular lemma. Thus for *regent* (the verb *rain*), N is 1320, n is 100, and p is 0.86. The formula gives 86.34 as the deviation. This means that the true frequency for this form of *regent* is almost certain—95% certain at least—to lie between 1049 and 1221 .

Occasionally you may come across cases where the deviation figure is greater than or equal to the frequency figure itself. This indicates that you are dealing with a spelling which cannot be disambiguated, as with the example *agenten* discussed above. While the frequency figures in such cases are arbitrary, the accompanying deviation figures are 100% accurate.

So while ***INLSpellFreq*** gives the disambiguated frequency figure for each spelling, this column indicates the statistical deviation of that figure. Its FLEX name and description are as follows:

InlSpellDev ***(InlSpellDevLemma)***	`95% confidence deviation`

Finally, remember that the columns described here refer only to the frequencies of individual *spellings*: most of the frequency information is dealt with in section 5 ‘Dutch Frequency’.

1.5 SPELLING

Before defining the specific spelling columns available with each of the four Dutch lexicon types, it’s worth considering a few important general features which apply to many of the columns, namely *diacritics* and *reversed transcriptions*. After that come the individual spelling columns themselves.

1.5.1 DIACRITICS

As you work your way down the `ADD COLUMN` menus, you can see that on several occasions the last menu in the series allows you to select transcriptions which contain—or omit—*diacritics*. Diacritics are the accents written above certain characters as a guide to pronunciation. In Dutch, they clarify whether a vowel written should become a vowel pronounced, for example *zeeëend*. This is a permanent feature of Dutch orthography, and thus included in the database. In other languages accents abound, and when foreign words are given in the database, the correct markers accompany them: *rücksichtslos*, *sévère*, *taalbarrière*, *doña*.

These special accented characters are eight-bit characters designed for use on certain DIGITAL terminals (the VT220 and newer terminals). If you use such a terminal, or can get your own terminal to emulate it, then you look at the diacritics columns with no problems at all. If you have a completely different terminal, you can still use diacritics columns by selecting the `MODIFY COLUMNS` option `CONVERT` to change the DIGITAL eight-bit codes to the form your terminal needs to produce the same diacritic characters.

To do this, you need a table of the DIGITAL eight-bit codes that CELEX uses, such as the one given in part 6 of the manual, the *Appendices*). In it you can find out the hexadecimal codes of the letters you need to convert. You also need a table of the codes your terminal uses to produce the same diacritical markers. The example that follows converts all the DIGITAL eight-bit codes that are used in the Dutch database to their MS-DOS equivalents (as defined in the 1985 OLIVETTI MS-DOS User Guide). The characters which occur with diacritic markers are as follows: É, Ü, à, á, â, ä, å, ç, è, é, ê, ë, î, ï, ñ, ó, ô, ö, û, ü and °. When you reach the `MODIFY CONVERSION` window, first select a column which contains transcriptions with diacritics, then type in the following string:

```
([\x20-\x7F]+
|\xC9%\x90|\xDC%\x9A  |\xE0%\x85|\xE1%\xA0
|\xE2%\x83|\xE4%\x84  |\xE5%\x86|\xE7%\x87
|\xE8%\x8A|\xE9%\x82  |\xEA%\x88|\xEB%\x89
|\xEE%\x8C|\xEF%\x8B  |\xF1%\xA4|\xF3%\xA2
|\xF4%\x93|\xF6%\x94  |\xFB%\x96|\xFC%\x81
|\xB0%\xF8)*
```

Once installed, this pattern will convert all the diacritic characters whenever you `SHOW` or `EXPORT` the column. If you're new to the pattern matcher and its capabilities then it may appear very mysterious, but in fact it's straightforward. Read the next couple of paragraphs for a full explanation.

The first line indicates that one or more normal ASCII codes (those with hexadecimal values between 20 and 7F) are allowed.

The remaining lines indicate the changes that must be made to any 8-bit characters that occur. The pattern matcher uses the `%` sign to indicate a conversion: the element to the left of the `%` is converted to the element on the right. (This use of the `%` sign is different from the 'wildcard' function it has at other times.) The pattern matcher also uses the symbols `\x` to mean that the two characters which follow form a hexadecimal code – thus in the DIGITAL eight-bit code `\xDC` actually means `Ü`. In the MS-DOS coding set, the same `Ü` character is represented by the code `\x9A`. So to tell the pattern matcher to convert from a DIGITAL `Ü` to an MS-DOS `Ü`, you must type `\xDC%\x9A`.

So far, this accounts for one diacritic character. To convert all the diacritic characters, you have to add extra parts to the pattern as appropriate, until you end up with a pattern like the one above. Each element is separated by the OR marker `|`. The whole pattern comes between brackets followed by an asterisk at the end `(...)*`, which means 'the word may be made up of zero or more of the elements between the brackets'.

1.5.2 REVERSE TRANSCRIPTIONS

Transcriptions without diacritics are often available in *reverse order*; each item is given back to front. Thus *terug* is given as *guret*. The reason for this is that with a draft lexicon, looking up word endings can be done much more quickly when you use reverse transcriptions.

1.6 SPELLING COLUMNS

This section sets out the columns with spellings available for each lexicon type. First, the Abbreviation and INL Corpus Type lexicons are dealt with briefly, followed by the longer descriptions required for lemma and wordform lexicons.

1.6.1 TRANSCRIPTIONS FOR ABBREVIATIONS

In addition to the spelling number (described above), there is a special column available for abbreviations which gives the expanded form of each abbreviation in full. It tells you what all the abbreviations actually mean. Thus for the abbreviation *BBC*, this column contains the words *British Broadcasting Corporation*. The FLEX name and description of this column are as follows:

Meaning `Non-abbreviated form`

There are two spelling columns available. The first gives plain transcriptions which may contain both upper case and lower case letters, full stops, and hyphens. No diacritic markers are included. The FLEX name and description of this column are as follows:

Abbr `Abbreviations`

The second gives transcriptions which can include both upper case and lower case letters, hyphens, as well as two characters with diacritic markers ∘ and ö. The FLEX name and description of this column are as follows:

AbbrDia `Abbreviations, diacritics`

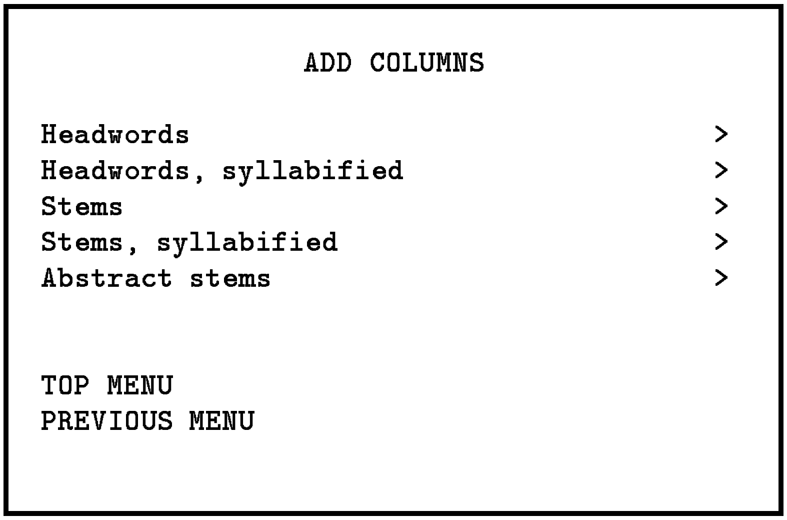
1.6.2 TRANSCRIPTIONS FOR CORPUS TYPES

One column is available. It gives plain transcriptions, which include lower case letters, hyphens, full stops, apostrophes, round brackets, and digits. If you're not sure exactly what corpus types are, check part 1 of the manual, the *Introduction* to find out. The FLEX name and description of this column are as follows:

Type `Corpus type spelling`

1.6.3 TRANSCRIPTIONS FOR LEMMAS

The number of orthographic options available means that another level in the ADD COLUMNS menu structure is required for lemmas:

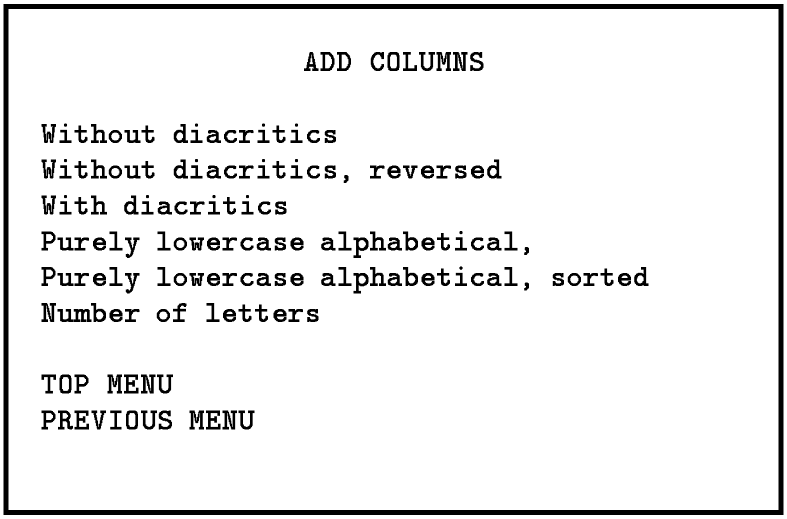


```
                 ADD COLUMNS

Headwords                                  >
Headwords, syllabified                     >
Stems                                      >
Stems, syllabified                         >
Abstract stems                             >


TOP MENU
PREVIOUS MENU
```

The subsections which follow deal in turn with each of these lemma representations.

1.6.3.1 SPELLINGS FOR HEADWORDS

A headword is that form of a lemma which most consistently corresponds to the bold-type heading used in paper dictionaries. A full description of the properties of headwords can be found in part one of the manual, the *Introduction*, under the section called 'Lexicon types'. There are six columns offered in the ADD COLUMNS menus, and each contains spellings of headwords in a different form.



```
                 ADD COLUMNS

Without diacritics
Without diacritics, reversed
With diacritics
Purely lowercase alphabetical,
Purely lowercase alphabetical, sorted
Number of letters

TOP MENU
PREVIOUS MENU
```

The first column contains information which is basic to the other five columns. It simply contains headwords composed of upper and lower case characters, hyphens and apostrophes, with no diacritics or any other alterations. The FLEX name and description of this column are as follows:

Head ***(HeadLemma)*** `Headword`

The second column contains all the headwords to be found in the first column, except that the order of the letters is reversed. The FLEX name and description of this column are as follows:

HeadRev ***(HeadRevLemma)*** `Headword, reversed`

The third column gives spellings which include diacritics as well as the basic upper and lower case characters, hyphens and apostrophes of the basic, transcriptions. The characteristics of diacritics are described in section 1.5.1 above. The FLEX name and description of this column are as follows:

HeadDia ***(HeadDiaLemma)*** `Headword, diacritics`

The next three columns all use headwords with upper case characters reduced to lower case characters and any non-alphabetic characters (hyphens, apostrophes) removed. This is particularly useful for automatic sorting programs: a column containing purely lower case alphabetical characters can be used to provide normal dictionary-like alphabetical order (i.e. not ASCII order, which differentiates between upper and lower case characters) for a lexicon, whatever the contents of its other columns. The first of these three contains the ordinary headwords of the very first column with the upper case letters replaced by the corresponding lower case letters. The FLEX name and description of this column are as follows:

HeadLow ***(HeadLowLemma)*** `Headword, lowercase, alphabetical`

The next column contains (purely lower case) headwords with their constituent letters in alphabetical order (*kaarsenstandaard* becomes *aaaaaddeknnrrst*, for example). Using this column, anagrams can be solved quickly, and searches for words containing certain numbers of letters can be carried out with ease: creating a query which looks for `aaa%` in this column can return a list of words (from another column) which contain at least three *a* characters. The FLEX name and description of this column are as follows:

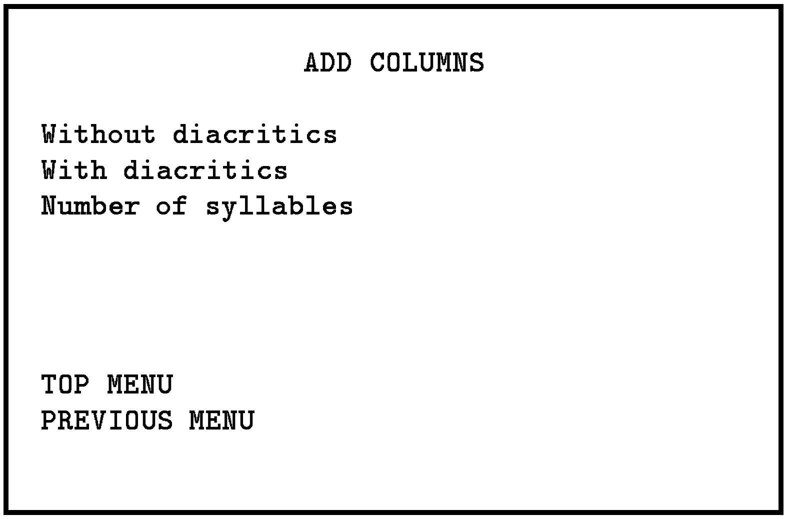
HeadLowSort ***(HeadLowSortLemma)*** `Headword, lowercase, alphabetical, sorted`

The sixth and last column contains counts of the number of letters in each headword. Here *letters* means any upper or lower case alphabetic characters, excluding hyphens and apostrophes. This means that sometimes the length of a word is different from the number of letters it contains – the number of letters in *atletiek-unie* for example is 12. The FLEX name and description of this column are as follows:

HeadCnt ***(HeadCntLemma)*** `Headword, number of letters`

1.6.3.2 SPELLINGS FOR SYLLABIFIED HEADWORDS

There are two columns which contain headwords with their orthographic syllable markers. In these columns, a hyphen marks the boundary between each pair of syllables within the headword. Thus the plain headword *abonnement* is given as *a-bon-ne-ment*. Whenever a syllable boundary and a normal hyphen occur at the same place, two hyphens are given to overcome any possible ambiguity, so *oud-katholiek* is given as *oud--ka-tho-liek*. There is a third column relating to syllabified headwords, and it tells you the number of orthographic syllables each headword has.



The first column contains the basic headwords plus syllable markers, each transcription consisting of upper and lower case characters, hyphens and apostrophes. The FLEX name and description of this column are as follows:

HeadSyl ***(HeadSylLemma)***	`Headword, syllabified`

The second column contains the same headwords as the first, except that diacritics are included where appropriate. The FLEX name and description of this column are as follows:

HeadSylDia ***(HeadSylDiaLemma)***	`Headword, syllabified, diacritics`

Some people like to use only *partially* syllabified headwords – that is, syllabified transcriptions which omit the first syllable marker if the first syllable consists of only one letter. For example, the partially syllabified transcription of *abonnement* would be *abon-ne-ment*. Such transcriptions are useful for automatic hyphenation programs, since typographic convention says that a word divided at the end of a line should consist of more than one character. To obtain transcriptions in this form, you can use the CONVERT option of the MODIFY COLUMNS menu. When you reach the MODIFY CONVERSION window, select a column containing normal syllabified headwords, and then type the following string:

```
@((-%)^-)/@*
```

This means 'first there is one character of some sort. Then, if there is a hyphen followed by a character which is *not* a hyphen, convert the hyphen into nothing; then there are zero or more other characters of some sort'. Thus whenever you SHOW or EXPORT your lexicon, the syllabified transcriptions will always appear in partially syllabified form. Two hyphens together after a first letter indicate that there is an orthographic hyphen (as opposed to a syllable marker) in the spelling at this point (as in *a-politiek* for example). They are left as two hyphens to differentiate this sort of hyphen from the other syllable markers the word might contain.

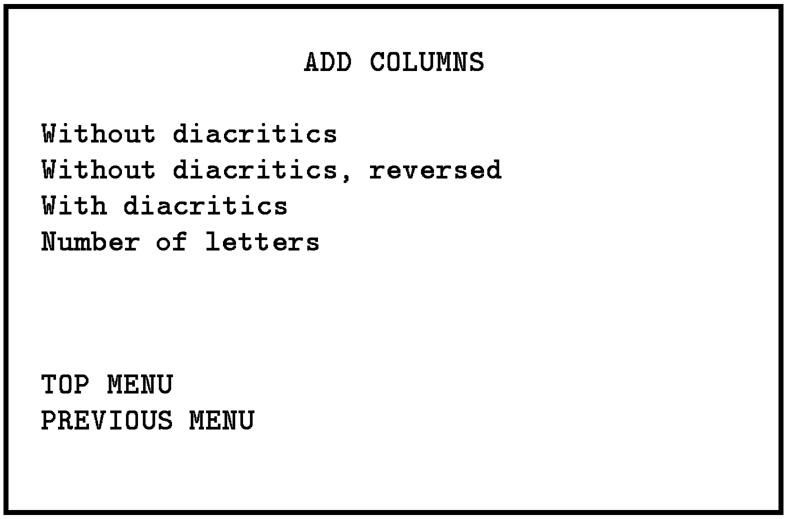
The third and last column for syllabified headwords tells you how many syllables each headword contains. The number of syllables in the word *a-bon-ne-ment*, for example, is 4. The FLEX name and description of this column are as follows:

HeadSylCnt
(HeadSylCntLemma)

```
Number of orthographic syllables
```

1.6.3.3 SPELLINGS FOR STEMS

A stem is that form of a lemma which most linguists prefer to use in their work, since it is generally the shortest occurring form in a family of inflections. A full description of the properties of stems can be found in part one of the manual, the *Introduction*, under the section called *Lexicon types*. There are four columns offered in the ADD COLUMNS menus, and each contains spellings of stems in a different form.



The first column contains information basic to the other three columns. It simply contains stems composed of upper and lower case characters, hyphens and apostrophes, with no diacritics or any other alterations. The FLEX name and description of this column are as follows:

Stem
(StemLemma)

```
Stem
```

The second column contains the same stems as the first, except that the characters are given in reverse order. (This enables you to look for word endings more quickly and with greater ease.) The FLEX name and description of this column are as follows:

StemRev
(StemRevLemma)

```
Stem, reversed
```

The third column contains the plain stem (containing upper and lower case letters, hyphens, and apostrophes) complete with diacritic markers (as described in section 1.5.1 above). The FLEX name and description of this column are as follows:

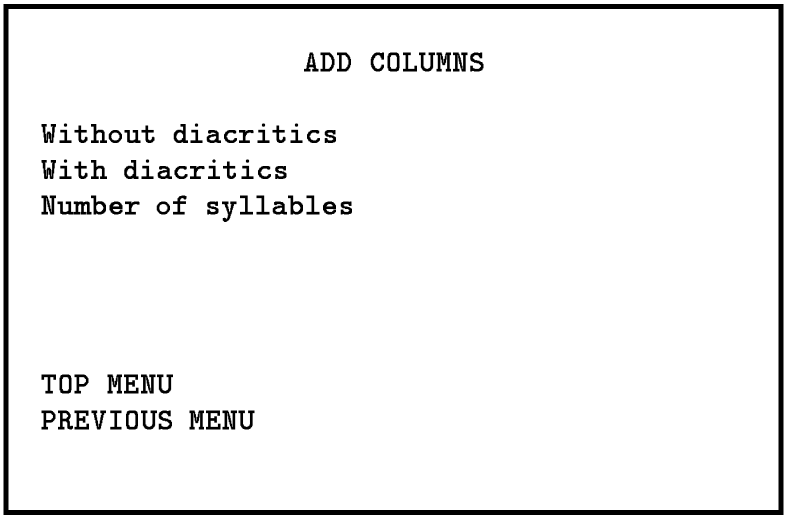
StemDia
(StemDiaLemma) `Stem, diacritics`

The fourth and last plain stem column contains counts of the number of letters in each stem. Here *letters* means any upper or lower case alphabetic characters, excluding hyphens and apostrophes. This means that sometimes the length of a word is different from the number of letters it contains – the number of letters in *atletiek-unie* for example is 12. The FLEX name and description of this column are as follows:

StemCnt
(StemCntLemma) `Stem, number of letters`

1.6.3.4 SPELLINGS FOR SYLLABIFIED STEMS

There are two columns which contain stems with their orthographic syllable markers. In these columns, a hyphen marks the boundary between each pair of syllables within the stem. Thus the plain stem *abonneer* is given as *a-bonneer*. Whenever a syllable boundary and a normal hyphen occur at the same place, two hyphens are given to overcome any possible ambiguity, so *oud-katholiek* is given as *oud--ka-tho-liek*. There is a third column relating to syllabified stems, and it tells you the number of orthographic syllables each stem has.



```
                 ADD COLUMNS

Without diacritics
With diacritics
Number of syllables





TOP MENU
PREVIOUS MENU
```

The first column simply contains stems composed of upper and lower case characters, hyphens and apostrophes, with no diacritics. Each syllable boundary is marked by a hyphen. The FLEX name and description of this column are as follows:

StemSyl ***(StemSylLemma)***	`Stem, syllabified`

The second column contains the plain stem (containing upper and lower case letters, hyphens, and apostrophes) complete with diacritic markers (as described in section 1.5.1 above). The FLEX name and description of this column are as follows:

StemSylDia ***(StemSylDiaLemma)***	`Stem, syllabified, diacritics`

Some people like to use only *partially* syllabified stems – that is, syllabified transcriptions which omit the first syllable marker if the first syllable consists of only one letter. For example, the partially syllabified transcription of *abonneer* would be *abon-neer*. Such transcriptions are useful for automatic hyphenation programs, since typographic convention says that a word divided at the end of a line should consist of more than one character. To obtain transcriptions in this form, you can use the `CONVERT` option of the `MODIFY COLUMNS` menu. When you reach the `MODIFY CONVERSION` window, select a column containing normal syllabified stems, and then type the following string:

```
@<testchars>@*#<testchars>=--|-%|
```

This means 'each row is made up of one single character of any sort, followed by something called testchars, which may be followed by zero or more characters of any sort'. After the 'begin definition' `#` character comes the definition of testchars. 'if there are two hyphens together (indicating that there is an orthographic, rather than syllabic, hyphen at this point), then leave both in the transcription. If, on the other hand, there is one hyphen, then re-write that hyphen as nothing. Otherwise, if there are no hyphens, interpret testchars as having no characters.' Thus when you `SHOW` or `EXPORT` your lexicon, the syllabified transcriptions will always appear in partially syllabified form, and the double 'orthographic' hyphens which coincide with the second-letter syllable boundary are retained to differentiate them from syllable marker hyphens.

The third and last column for syllabified stems tells you how many syllables each stem contains. For the word *a-bon-neer*, for example, the number of syllables is 3. The FLEX name and description of this column are as follows:

***StemSylCnt* (*StemSylCntLemma*)** `Stem, number of orthographic syllables`

1.6.3.5 SPELLINGS FOR ABSTRACT STEMS

Normally, the form of a stem can be easily determined, using the rules set out in the section called 'Lexicon types' in part 1 of the manual, the *Introduction*. Usually, these rules yield stems which seem natural enough; they look like real words (*kat*, for example, is in real life a singular noun, and for electronic dictionary users, it also happens to be a stem). Sometimes, though, the resulting stem doesn't always seem so natural. Take the verb *lezen*, for example: in the present tense, the first person singular form is 'ik *lees*', but the first person plural form is 'wij *lezen*'. So, in accordance with the rules, the ordinary stem is *lees*, but for those who like to use a more abstract stem, the columns described here give the form *leez*. The 'Abstract Stem Rule' applies only to stems ending in *-s* or *-f*, and it is as follows:

If, in any inflectional forms of the stem, the final *-f* becomes a *-v*, or the final *-s* a *-z*, then likewise make the stem abstract by using *-v* or *-z*.

Here are a few more examples to illustrate this concept: the verb *leven* has as its abstract stem *leev*, because the forms 'ik *leef*' and 'wij *leven*' occur in the language. In the same way *kaaz* is the abstract stem of the noun *kaas*, plural form *kazen*.

Three columns containing abstract stems are available. In all of them, the abstract stem is given in place of the plain stem when the rules indicate that an abstract form is appropriate, and in all other cases the ordinary stem is given.

The first column gives stems and abstract stems when appropriate using upper case and lower case letters, along with hyphens and apostrophes. The FLEX name and description of this column are as follows:

AbStem `Abstract stem`
(AbStemLemma)

The second column contains the same transcriptions as the previous column, only this time diacritics are included when appropriate. The FLEX name and description of this column are as follows:

AbStemDia `Abstract stem, diacritics`
(AbStemDiaLemma)

The third and final abstract stem column contains counts of the number of letters in each abstract stem, excluding hyphens and apostrophes. The number of letters in *kaaz*, for example, is 4. The FLEX name and description of this column are as follows;

AbStemCnt `Abstract stem, number of letters`
(AbStemCntLemma)

1.6.4 TRANSCRIPTIONS FOR WORDFORMS

Wordforms are the words which we use in everyday speech and writing, the inflected forms of the stems and headwords listed in dictionaries and databases. A full description of the properties of wordforms can be found in part one of the manual, the *Introduction*, under the section called 'Lexicon types'. Transcriptions are available either with or without syllable markers.

1.6.4.1 SPELLINGS FOR WORDFORMS

There are six columns offered in the `ADD COLUMNS` menus, and each contains spellings of wordforms in a different form.

```
                    ADD COLUMNS

 Without diacritics
 Without diacritics, reversed
 With diacritics
 Purely lowercase alphabetical,
 Purely lowercase alphabetical, sorted
 Number of letters

 TOP MENU
 PREVIOUS MENU
```

The first column contains information which is basic to the other five columns. It simply contains wordforms composed of upper and lower case characters, hyphens and apostrophes, with no diacritics or any other alterations. The FLEX name and description of this column are as follows:

Word `Word`

The second column contains all the wordforms to be found in the first column, except that the order of the letters is `reversed`. The FLEX name and description of this column are as follows:

WordRev `Word, reversed`

The third column gives spellings which include diacritics as well as the basic upper and lower case characters, hyphens and apostrophes of the basic transcriptions. The characteristics of diacritics are described in section 1.5.1 above. The FLEX name and description of this column are as follows:

WordDia `Word, diacritics`

The next three columns all give wordforms with upper case characters reduced to lower case characters and any non-alphabetic characters (hyphens, apostrophes) removed. This is particularly useful for automatic sorting programs: a column containing purely lower case alphabetical characters can be used to provide normal dictionary-like alphabetical order (i.e. not ASCII order, which differentiates between upper and lower case characters) for a lexicon, whatever the contents of its other columns. The first of these three contains the ordinary wordforms of the very first column with the upper case letters replaced by the corresponding lower case letters. The FLEX name and description of this column are as follows:

WordLow `Word, lowercase, alphabetical`

The next column contains (purely lower case) wordforms with their constituent letters in alphabetical order (*kaarsenstandaard* becomes *aaaaaddeknnrrst*, for example). Using this column, anagrams can be solved quickly, and searches for words containing certain numbers of letters can be carried out with ease: creating a query which looks for `aaa%` in this column can return a list of words (from another column) which contain at least three *a* characters. The FLEX name and description of this column are as follows:

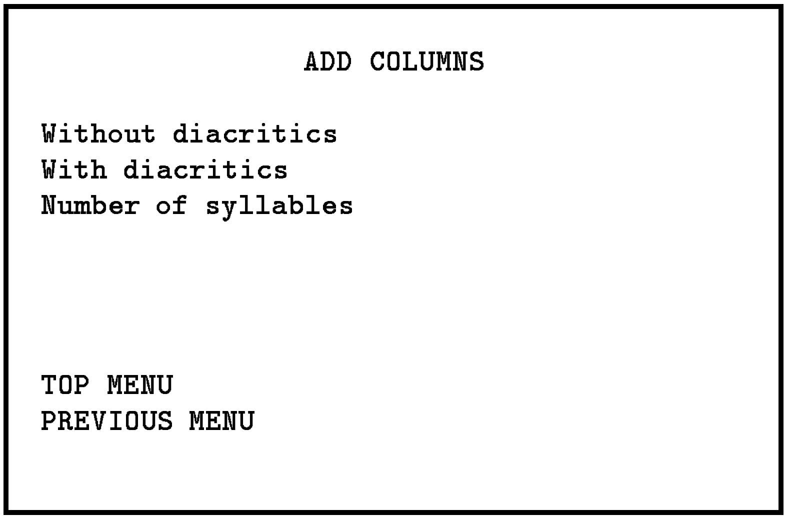
WordLowSort `Word, lowercase, alphabetical, sorted`

The sixth and last column contains counts of the number of letters in each wordform. Here *letters* means any upper or lower case alphabetic characters, excluding hyphens and apostrophes. This means that sometimes the length of a word is different from the number of letters it contains – the number of letters in *atletiek-unie* for example is 12. The FLEX name and description of this column are as follows:

WordCnt `Word, number of letters`

1.6.4.2 SPELLINGS FOR SYLLABIFIED WORDFORMS

There are two columns which contain wordforms with their orthographic syllable markers. In these columns, a hyphen marks the boundary between each pair of syllables within the wordform. Thus the plain wordform *abonnement* is given as *a-bon-ne-ment*. Whenever a syllable boundary and a normal hyphen occur at the same place, two hyphens are given to overcome any possible ambiguity, so *oud-katholiek* is given as *oud--ka-tho-liek*. There is a third column relating to syllabified wordforms and it tells you the number of orthographic syllables each wordform has.



The first column contains wordforms plus syllable markers. Each transcription consisting of upper and lower case characters, hyphens and apostrophes. The FLEX name and description of this column are as follows:

WordSyl `Word, syllabified`

The second column contains the same wordforms as the first, except that diacritics are included where appropriate. The FLEX name and description of this column are as follows:

WordSylDia `Word, syllabified, with diacritics`

Some people like to use only *partially* syllabified headwords – that is, syllabified transcriptions which omit the first syllable marker if the first syllable consists of only one letter. For example, the partially syllabified transcription of *abonnement* would be *abon-ne-ment*. Such transcriptions are useful for automatic hyphenation programs, since typographic convention says that a word divided at the end of a line should consist of more than one character. To obtain transcriptions in this form, you can use the `CONVERT` option of the `MODIFY COLUMNS` menu. When you reach the `MODIFY CONVERSION` window, select a column containing normal syllabified wordforms, and then type the following string:

```
@.{part1}-/@*.{part2}%{part1}{part2}
```

This basically means 'call the first character by the name *part1*. The second character may or may not be a hyphen. Any subsequent characters are called *part2*. Re-write the whole word as *part1* plus *part2*.' Only the parts of the word assigned to *part1* or *part2* are re-written, thus excluding the first hyphen whenever it occurs, because it is not assigned to any variable. When you `SHOW` or `EXPORT` your lexicon, the syllabified transcriptions will always appear in partially syllabified form. However note that on this occasion, when a double 'orthographic' hyphen occurs after the first letter, only one of the two hyphens is written. So if you ever do see a hyphen as the second letter in your converted column, you know for sure that it is actually an orthographic and not a syllabic hyphen. (For patterns which retain both hyphens in such a position, check the two previous column conversion examples.)

The third and last column for syllabified wordforms tells you how many syllables each wordform contains. The number of syllables in the word *a-bon-ne-ment*, for example, is 4. The FLEX name and description of this column are as follows:

WordSylCnt `Word, number of orthographic syllables`

2 DUTCH PHONOLOGY

Phonetic and phonological transcriptions are available for lemmas and wordforms, along with the appropriate CV patterns, stress patterns, and phoneme and phonetic syllable counts. In addition, when you are using a wordform lexicon, you can get phonetic information (and other information too) about the lemmas of any wordforms you look at in the morphology `add columns` menus. Abbreviation and INL corpus type lexicons cannot contain any phonetics columns. Phonetics Columns— While phonetic transcriptions are available for most of the wordforms, headwords and stems, none are available as yet for proper nouns. So occasionally null values do occur in the transcription columns, and the value 0 (zero) is given for such words in the numeric columns which contain counts of various sorts.

2.0.1 COMPUTER PHONETIC CHARACTER SETS

Four different sets of phonetic character codes are available from CELEX. The first three sets are SAM-PA, CELEX and CPA, and they can be thought of as computerized versions of IPA. They use standard ASCII codes—those which can be typed in and read on almost any terminal—to represent certain of the IPA characters. As far as possible, these sets have been designed to resemble IPA; a lot of the characters you type or read look like their IPA counterparts. As with IPA, diphthongs and affricates are represented by writing the two appropriate characters next to each other, and long vowels are indicated by length markers. In some cases, however, these conventions can lead to ambiguity: are the two vowels shown next to each other *really* a diphthong, or are they in fact two separate vowels? To overcome such problems, there are columns which contain transcriptions with syllable markers, and also columns available which have a delimiter placed after each consonant, affricate, vowel, long vowel or diphthong. So, these sets of computer codes for phonetic transcription can provide a readable approximation of IPA, with extra provision made to overcome the possibility of ambiguity.

The first of these three sets is the SAM-PA set. It was developed in connection with a European Community research program, and it has been presented in the *Journal of the International Phonetic Association* (1987) 17 : 22, pp. 94–114 as a widely-agreed computer-readable phonetic character set suitable for use with Danish, Dutch, English, French, German and Italian. For technical reasons, the version of SAMPA implemented by CELEX has to include one change: the \ character (ASCII code 92) representing the 'half-open front rounded' vowel sound has been implemented as / (ASCII code 47). The second is a set originally designed for use within CELEX. The third is CPA, the *Computer Phonetic Alphabet*, or *Esprit 291*, which was developed in the Ruhr Universität Bochum, West Germany.

The fourth set is the DISC set, so called because it is a computer phonetic alphabet made up of distinct single characters. It is fundamentally different from the other three in that it assigns one ASCII code to each distinct phonological segment in the sound systems of Dutch, English and German. Here *segment* means a consonant, an affricate, a short vowel, a long vowel or a diphthong. There are two main advantages to this set. First, it provides one character for one segment – in contrast to the other three sets which use extra characters for long vowels, affricates and diphthongs. Second, there is no possibility of ambiguous transcriptions. A diphthong is always shown as a diphthong, and two separate vowels in proximity to each other (say on either side of a syllable boundary) can thus no longer be confused with a real diphthong; an affricate is always shown as such, and not as two consonants. For both these reasons, those interested in processing phonetic transcriptions—as opposed to reading transcriptions in a character set that resembles the familiar IPA—may well choose transcriptions in this character set. Its most basic codes correspond to SAM-PA; all the SAM-PA codes which represent short vowels and consonants are included in this set. The remaining long vowels, diphthongs and affricates have been assigned codes not already in use for other purposes. The resulting character set thus does not look as elegant and IPA-like as the other three sets. However, if you are mainly interested in the computer processing of transcriptions, such æsthetic considerations might not be so important.

Clearly, you have a wide choice of transcriptions available to you. The type you choose will depend on the nature of the task you have in mind. For IPA-like readability and non-ambiguous transcriptions, use the SAM-PA, CELEX or CPA sets. For computer processing tasks which need one-character-to-one-segment-correspondence, use the DISC set. In Appendix I there is a table which sets out DISC and how it relates to Dutch, English and German.

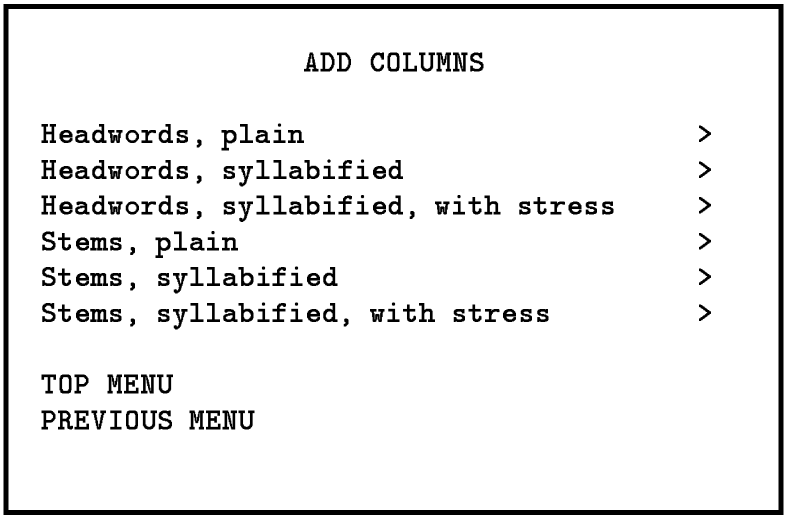
The table on the next page lists the basic set of segments for Dutch. Each line gives an IPA character alongside a word which exemplifies the sound and the equivalent characters in the four computer-usable sets available with CELEX.

2.1 PHONETIC TRANSCRIPTIONS

Phonetic transcriptions are available for lemmas (headwords and stems) and also for wordforms. They are written using the four computer phonetic alphabets described in the previous section. In addition, there are columns containing CV patterns, and also some phonological representations for stems in the CELEX and CPA computer phonetic alphabets. There are no phonetic transcriptions for abbreviations or INL corpus types.

2.1.1 LEMMA TRANSCRIPTIONS

The first choices you must make in your search for phonetic transcriptions concern the form of the lemma you want to use (headword or stem) and whether you want your transcription to contain stress markers and/or syllable markers:



```
                    ADD COLUMNS

Headwords, plain                              >
Headwords, syllabified                        >
Headwords, syllabified, with stress           >
Stems, plain                                  >
Stems, syllabified                            >
Stems, syllabified, with stress               >

TOP MENU
PREVIOUS MENU
```

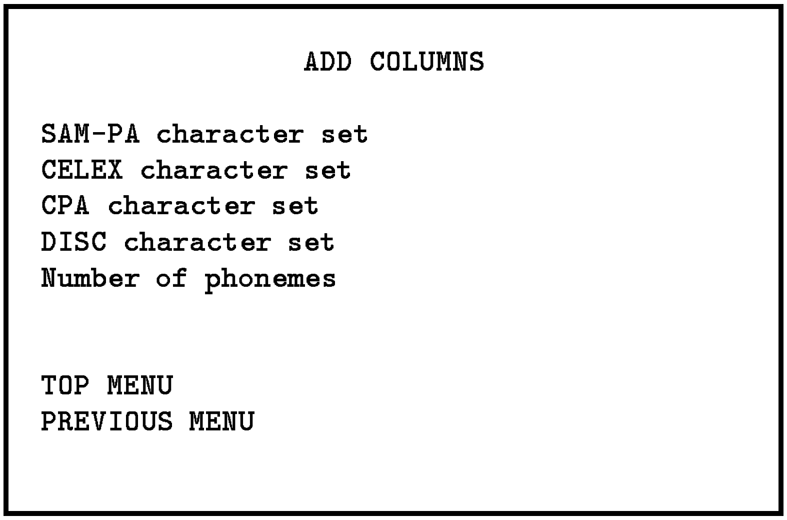
IPA	example	SAM-PA	CELEX	CPA	DISC
p	**p**ut	p	p	p	p
b	**b**ad	b	b	b	b
t	**t**ak	t	t	t	t
d	**d**ak	d	d	d	d
k	**k**at	k	k	k	k
g	**g**oal	g	g	g	g
ŋ	la**ng**	N	N	N	N
m	**m**at	m	m	m	m
n	**n**at	n	n	n	n
l	**l**at	l	l	l	l
ʀ, r	**r**at	r	r	r	r
f	**f**iets	f	f	f	f
v	**v**at	v	v	v	v
s	**s**ap	s	s	s	s
z	**z**at	z	z	z	z
ʃ	**sj**aal	S	S	S	S
ʒ	rava**g**e	Z	Z	Z	Z
j	**j**as	j	j	j	j
x	li**ch**t, **g**aat	x	x	X	x
ɣ	re**g**en	G	G	G	G
h	**h**ad	h	h	h	h
ʋ	**w**at	w	w	w	w
ʤ	**j**azz	dZ	dZ	J/	_
iː	l**ie**p	i:	i:	i:	i
yː	b**uu**t	y:	y:	y:	y
eː	l**ee**g	e:	e:	e:	e
øː	d**eu**k	|:	&:	q:	|
aː	l**aa**t	a:	a:	a:	a
oː	b**oo**m	o:	o:	o:	o
uː	b**oe**k	u:	u:	u:	u
ɪ	l**i**p	I	I	I	I
ɛ	l**e**g	E	E	E	E
ɑ	l**a**t	A	A	A	A
ɔ	b**o**m	O	O	O	O
ʉ	p**u**t	}	U	Y/	}
ə	g**e**lijk	@	@	@	@
iːː	anal**y**se	i::	i::	i::	!
yːː	centrif**u**ge	y::	y::	y::	(
ɛː	sc**e**ne	E:	E:	E:	)
œː	fr**eu**le	/:	U:	Q:	*
ɒː	z**o**ne	Q:	O:	o:	<
ɛi	w**ij**s	EI	EI	y/	K
œy	h**ui**s	/I	UI	q/	L
ɑu	k**ou**d	Au	AU	A/	M

Table 2: Computer codes for Dutch phonetic transcriptions

The columns available with each of these options are described in full in the six subsections which follow. If you want to see how all these different types of transcriptions look, then consult table 3 on page 3–41: it gives a couple of examples from all the columns described below so that you can see at a glance the differences between them.

2.1.1.1 TRANSCRIPTIONS FOR HEADWORDS

This first set of columns offers *plain* transcriptions – that is, transcriptions which do not have any syllable markers or stress markers, written in each of the four coding systems already described:



However three of these columns have one special feature: *each phonetic segment ends with a delimiter*. Here a *segment* means a vowel, a consonant, a long vowel, a diphthong, or an affricate. Using a delimiter avoids any possibility of ambiguity between the two parts of a diphthong or an affricate – something which FLEX requires when it is working on `TOOLBOX` options such as `NEIGHBOURS` or `COHORTS`. These delimiter transcriptions are available in the SAM-PA, CELEX, and CPA character sets. Delimiters are not given with DISC transcriptions since the unique single-character nature of that set obviates the need to delimit each segment in this way.

The first plain headword transcription column uses the SAM-PA character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonSAM (PhonSAMLemma) `Phonetic headword, SAM-PA character set`

The second column uses the CELEX character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonCLX (PhonCLXLemma) `Phonetic headword, CELEX character set`

The third column uses the CPA character set, and full stops (.) as delimiters. (Normally CPA uses full stops as syllable markers, but here of course, no syllable markers are used.) The FLEX name and description of this column are as follows:

PhonCPA (PhonCPALemma) `Phonetic headword, CPA character set`

The fourth column uses the DISC set. No delimiters, syllable markers or stress markers are included, since each character equals one segment. The FLEX name and description of this column are as follows:

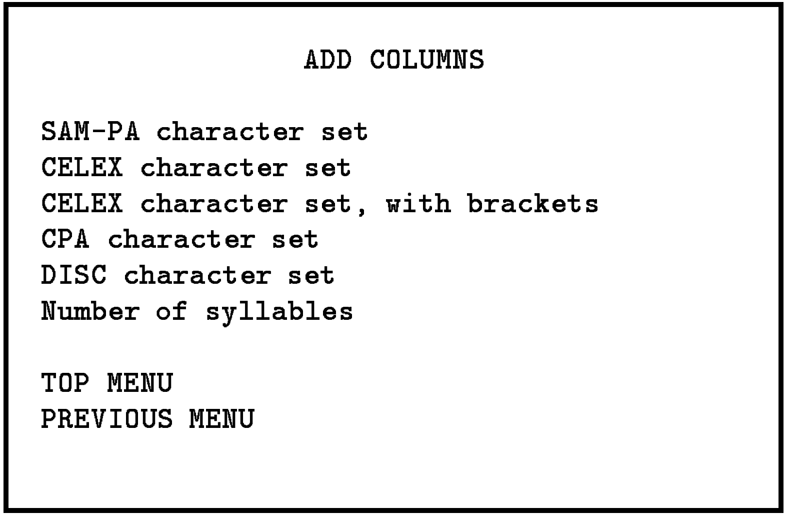
PhonDISC (PhonDISCLemma) `Phonetic headword, DISC character set`

The last column in this subsection gives you counts of the number of phonemes in each headword. Here *phoneme* means the same as *segment* – one phoneme equals a vowel, a consonant, a long vowel, a diphthong, or an affricate. Thus for the word *makelij* the number of phonemes is given as 6, while for *makkelijk* the number is 7. The FLEX name and description of this column are as follows:

PhonCnt (PhonCntLemma) `Headword, number of phonemes`

2.1.1.2 TRANSCRIPTIONS FOR SYLLABIFIED HEADWORDS

This set of transcriptions uses the same basic transcriptions as the first set, except that instead of segment markers, there are characters that mark each phonetic syllable. These are the columns which contain syllabified phonetic transcriptions of each headword:



```
                    ADD COLUMNS

SAM-PA character set
CELEX character set
CELEX character set, with brackets
CPA character set
DISC character set
Number of syllables

TOP MENU
PREVIOUS MENU
```

In most cases transcriptions are syllabified by putting a hyphen (or, in the case of CPA, a full stop) at every syllable boundary within each word. A second method, available with the CELEX character set, is to enclose each syllable within square brackets. The advantage of the brackets notation is that so-called 'ambisyllabic consonants' can be clearly identified. Ambisyllabic consonants are those consonants which come between two syllables, and which belong to both of those syllables. For example, the first `[k]` of *makkelijk* is part of the first syllable and the second syllable, whereas the `[k]` of *makelij* belongs to the second syllable only.

The first syllabified headword transcription column uses the SAM-PA character set, and syllable boundaries within words are shown by hyphens. The FLEX name and description of this column are as follows:

PhonSylSAM
(PhonSylSAMLemma)
`Syllabified phonetic headword, SAM-PA character set`

The next two columns both use the CELEX character set. The first marks every syllable boundary within each transcription with a hyphen. The FLEX name and description of this column are as follows:

PhonSylCLX
(PhonSylCLXLemma)
`Syllabified phonetic headword, CELEX character set`

The other CELEX syllabified phonetic headword column uses the brackets notation as described above, and its FLEX name and description are as follows:

PhonSylBCLX ***(PhonSylBCLXLemma)***	`Syllabified phonetic headword, CELEX character set (brackets)`

The next column gives syllabified headword transcriptions in the CPA character set. Every syllable boundary within each word is marked by a full stop. The FLEX name and description of this column are as follows:

PhonSylCPA ***(PhonSylCPALemma)***	`Syllabified phonetic headword, CPA character set`

The fifth column uses the DISC character set, and here every syllable boundary within each word is marked by a hyphen. The FLEX name and description of this column are as follows:

PhonSylDISC ***(PhonSylDISCLemma)***	`Syllabified phonetic headword, DISC character set`

The last column in this subsection gives counts of the phonetic syllables which occur in each transcription. For example, both *makkelijk* and *makelij* contain 3 syllables. The FLEX name and description of this column are as follows:

SylCnt ***(SylCntLemma)***	`Headword, number of phonetic syllables`

2.1.1.3 TRANSCRIPTIONS FOR STRESSED AND SYLLABIFIED HEADWORDS

This set of columns gives syllabified transcriptions that also mark the points of primary stress in each headword. These are the columns you can choose from:

```
                ADD COLUMNS

SAM-PA character set
CELEX character set
CPA character set
DISC character set
Stress Pattern


TOP MENU
PREVIOUS MENU
```

The first column uses the SAM-PA character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show points of primary stress by means of the 'double quote' character ("). This character is placed immediately before a stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsSAM `Syllabified phonetic headword, with stress`
(PhonStrsSAMLemma) `marker, SAM-PA character set`

The second column uses the CELEX character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show the points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsCLX `Syllabified phonetic headword, with stress`
(PhonStrsCLXLemma) `marker, CELEX character set`

The third column uses the CPA character set, including full stops to mark syllable boundaries, and these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsCPA `Syllabified phonetic headword, with stress`
(PhonStrsCPALemma) `marker, CPA character set`

The fourth column uses the DISC character set, and along with hyphens to mark syllable boundaries, these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsDISC `Syllabified phonetic headword, with stress`
(PhonStrsDISCLemma) `marker, DISC character set`

The last column in this subsection contains a simple stress pattern for each headword. A *stress pattern* is a string which shows how each phonetic syllable is stressed in speech. Each syllable is represented by one numeric character: either `0` or `1`. `1` indicates that the syllable receives primary stress, and `0` that it does not receive primary stress. Thus the three-syllable word *makkelijk* has the stress pattern `100` and *makelij* has the pattern `001`. The FLEX name and description of this column are as follows:

StrsPat `Headword, stress pattern`
(StrsPatLemma)

2.1.1.4 SOME EXAMPLE TRANSCRIPTIONS

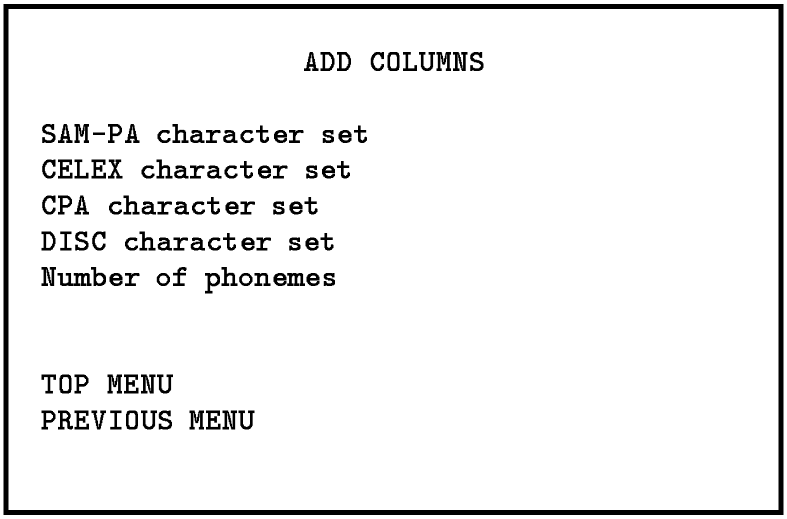
Column	Examples	
	makkelijk	*makelij*
PhonSAM	`m.A.k.@.l.@.k.`	`m.a:.k.@.l.EI.`
PhonCLX	`m.A.k.@.l.@.k.`	`m.a:.k.@.l.EI.`
PhonCPA	`m.A.k.@.l.@.k.`	`m.a:.k.@.l.y/.`
PhonDISC	`mAk@L@k`	`mak@lK`
PhonSylSAM	`mA-k@-l@k`	`ma:-k@-lEI`
PhonSylCLX	`mA-k@-l@k`	`ma:-k@-lEI`
PhonSylBCLX	`[mA[k]@][l@k]`	`[ma:][k@][lEI]`
PhonSylCPA	`mA.k@.l@k`	`ma:.k@.ly/`
PhonSylDISC	`mA-k@-L@k`	`ma-k@-lK`
PhonStrsSAM	`"mA-k@-l@k`	`ma:-k@-"lEI`
PhonStrsCLX	`'mA-k@-l@k`	`ma:-k@-'lEI`
PhonStrsCPA	`'mA.k@.l@k`	`ma:.k@.'ly/`
PhonStrsDISC	`'mA-k@-L@k`	`ma-k@-'lK`

Table 3: Example phonetic transcriptions

The table on the previous page lets you see the difference stress or syllable markers make to the appearance of your transcriptions. Use it in conjunction with the column descriptions to decide what sort of transcription you want to use. Although this table uses the names of the headword columns described above, the examples are exactly the same for the stem columns described below.

2.1.1.5 TRANSCRIPTIONS FOR STEMS

This first set of columns offers *plain* transcriptions – that is, transcriptions which do not have any syllable markers or stress markers, written in each of the four coding systems already described:



However three of these columns have one special feature: *each phonetic segment ends with a delimiter*. Here a *segment* means a vowel, a consonant, a long vowel, a diphthong, or an affricate. Using a delimiter avoids any possibility of ambiguity between the two parts of a diphthong or an affricate – something which FLEX requires when it is working on `TOOLBOX` options such as `NEIGHBOURS` or `COHORTS`. These delimiter transcriptions are available in the SAM-PA, CELEX, and CPA characters sets. Delimiters are not given with DISC transcriptions since the unique single-character nature of that set obviates the need to delimit each segment in this way.

The first plain stem transcription column uses the SAM-PA character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonStSAM `Phonetic stem, SAM-PA character set`
(PhonStSAMLemma)

The second column uses the CELEX character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonStCLX `Phonetic stem, CELEX character set`
(PhonStCLXLemma)

The third column uses the CPA character set, and full stops (.) as delimiters. (Normally CPA uses full stops as syllable markers, but here of course, no syllable markers are used.) The FLEX name and description of this column are as follows:

PhonStCPA `Phonetic stem, CPA character set`
(PhonStCPALemma)

The fourth column uses the DISC set. No delimiters, syllable markers or stress markers are included, since each character equals one segment. The FLEX name and description of this column are as follows:

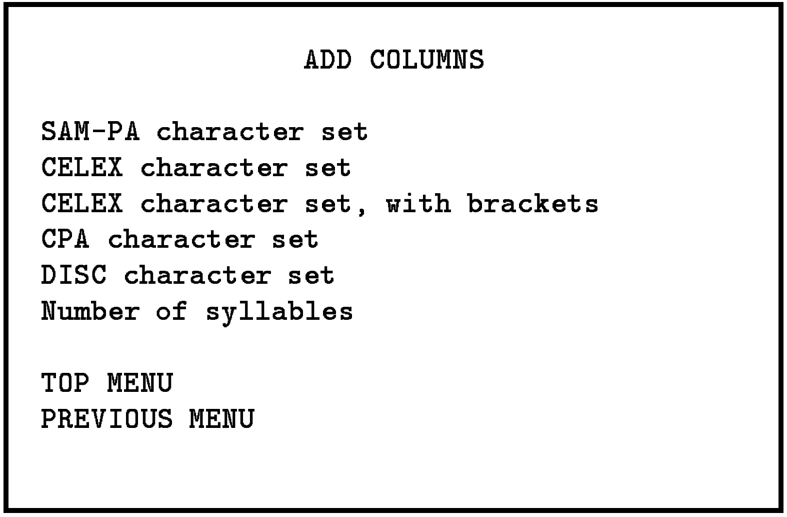
PhonStDISC `Phonetic stem, DISC character set`
(PhonStDISCLemma)

The last column in this subsection gives you counts of the number of phonemes in each stem. Here *phoneme* means the same as *segment* – one phoneme equals a vowel, a consonant, a long vowel, a diphthong, or an affricate. Thus for the word *makelij* the number of phonemes is given as 6, while for *makkelijk* the number is 7. The FLEX name and description of this column are as follows:

PhonStCnt `Stem, number of phonemes`
(PhonStCntLemma)

2.1.1.6 TRANSCRIPTIONS FOR SYLLABIFIED STEMS

This set of transcriptions uses the same basic transcriptions as the first set, except that instead of segment markers, there are characters that mark each phonetic syllable. These are the columns which contain syllabified phonetic transcriptions of each stem:



```
                ADD COLUMNS

SAM-PA character set
CELEX character set
CELEX character set, with brackets
CPA character set
DISC character set
Number of syllables

TOP MENU
PREVIOUS MENU
```

In most cases transcriptions are syllabified by putting a hyphen (or, in the case of CPA, a full stop) at every syllable boundary within each word. A second method, available with the CELEX character set, is to enclose each syllable within square brackets. The advantage of the brackets notation is that so-called 'ambisyllabic consonants' can be clearly identified. Ambisyllabic consonants are those consonants which come between two syllables, and which belong to both of those syllables. For example, the first `[k]` of *makkelijk* is part of the first syllable and the second syllable, whereas the `[k]` of *makelij* belongs to the second syllable only.

The first syllabified stem transcription column uses the SAM-PA character set, and syllable boundaries within words are shown by hyphens. The FLEX name and description of this column are as follows:

PhonSylStSAM
(PhonSylStSAMLemma)
`Syllabified phonetic stem, SAM-PA character set`

The next two columns both use the CELEX character set. The first marks every syllable boundary within each transcription with a hyphen. The FLEX name and description of this column are as follows:

PhonSylStCLX
(PhonSylStCLXLemma)
`Syllabified phonetic stem, CELEX character set`

The other CELEX syllabified phonetic stem column uses the brackets notation as described above, and its FLEX name and description are as follows:

PhonSylStBCLX (PhonSylStBCLXLemma) `Syllabified phonetic stem, CELEX character set (brackets)`

The next column gives syllabified stem transcriptions in the CPA character set. Every syllable boundary within each word is marked by a full stop. The FLEX name and description of this column are as follows:

PhonSylStCPA (PhonSylStCPALemma) `Syllabified phonetic stem, CPA character set`

The fifth column uses the DISC character set, and here every syllable boundary within each word is marked by a hyphen. The FLEX name and description of this column are as follows:

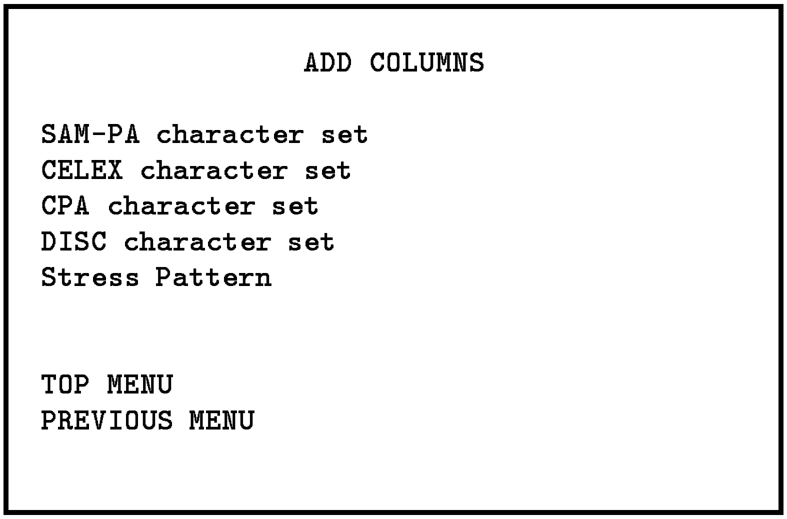
PhonSylStDISC (PhonSylStDISCLemma) `Syllabified phonetic stem, DISC character set`

The last column in this subsection gives counts of the phonetic syllables which occur in each transcription. For example, both *makkelijk* and *makelij* contain 3 syllables. The FLEX name and description of this column are as follows:

StSylCnt (StSylCntLemma) `Stem, number of phonetic syllables`

2.1.1.7 TRANSCRIPTIONS FOR STRESSED AND SYLLABIFIED STEMS

This set of columns gives syllabified transcriptions that also mark the points of primary stress in each stem. These are the columns you can choose from:



```
                    ADD COLUMNS

 SAM-PA character set
 CELEX character set
 CPA character set
 DISC character set
 Stress Pattern


 TOP MENU
 PREVIOUS MENU
```

The first column uses the SAM-PA character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show points of primary stress by means of the 'double quote' character ("). This character is placed immediately before a stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsStSAM `Syllabified phonetic stem, with stress marker,`
(PhonStrsStSAMLemma) `SAM-PA character set`

The second column uses the CELEX character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show the points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsStCLX `Syllabified phonetic stem, with stress marker,`
(PhonStrsStCLXLemma) `CELEX character set`

The third column uses the CPA character set, including full stops to mark syllable boundaries, and these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsStCPA `Syllabified phonetic stem, with stress marker,`
(PhonStrsStCPALemma) `CPA character set`

The fourth column uses the DISC character set, and along with hyphens to mark syllable boundaries, these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

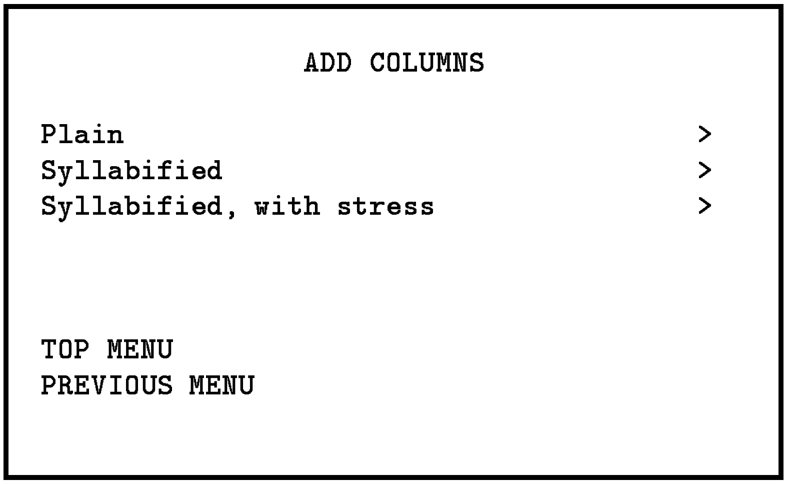
PhonStrsStDISC `Syllabified phonetic stem, with stress marker,`
(PhonStrsStDISCLemma) `DISC character set`

The last column in this subsection contains a simple stress pattern for each stem. A *stress pattern* is a string which shows how each phonetic syllable is stressed in speech. Each syllable is represented by one numeric character: either 0 or `1`. `1` indicates that the syllable receives primary stress, and `0` that it does not receive primary stress. Thus the three-syllable word *makkelijk* has the stress pattern `100` and *makelij* has the pattern `001`. The FLEX name and description of this column are as follows:

StStrsPat `Stem, stress pattern`
(StStrsPatLemma)

2.1.2 WORDFORM TRANSCRIPTIONS

A full range of phonetic transcriptions is available for wordforms. In addition, there are columns with phoneme and syllable counts and stress patterns for each wordform at appropriate points. You can choose them in your preferred computer phonetic character set, as described in section 2.0.1, but one small point to remember is that wordforms like *ga weg* which include a space in their spelling also include a space in their phonetic transcription, thus `xa: wEx`. The first choice you have to make is whether you want plain transcriptions, syllabified transcriptions, or stressed and syllabified transcriptions:



2.1.2.1 TRANSCRIPTIONS FOR WORDFORMS

This first set of columns offers *plain* transcriptions – that is, transcriptions which do not have any syllable markers or stress markers, written in each of the four coding systems already described:

```
                    ADD COLUMNS

SAM-PA character set
CELEX character set
CPA character set
DISC character set
Number of phonemes


TOP MENU
PREVIOUS MENU
```

However three of these columns have one special feature: *each phonetic segment ends with a delimiter*. Here a *segment* means a vowel, a consonant, a long vowel, a diphthong, or an affricate. Using a delimiter avoids any possibility of ambiguity between the two parts of a diphthong or an affricate – something which FLEX requires when it is working on `TOOLBOX` options such as `NEIGHBOURS` or `COHORTS`. These delimiter transcriptions are available in the SAM-PA, CELEX, and CPA characters sets. Delimiters are not given with DISC transcriptions since the unique single-character nature of that set obviates the need to delimit each segment in this way.

The first plain wordform transcription column uses the SAM-PA character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonSAM `Phonetic wordform, SAM-PA character set`

The second column uses the CELEX character set, and full stops (.) as segment delimiters. The FLEX name and description of this column are as follows:

PhonCLX `Phonetic wordform, CELEX character set`

The third column uses the CPA character set, and full stops (.) as delimiters. (Normally CPA uses full stops as syllable markers, but here of course, no syllable markers are used.) The FLEX name and description of this column are as follows:

PhonCPA `Phonetic wordform, CPA character set`

The fourth column uses the DISC set. No delimiters, syllable markers or stress markers are included, since each character equals one segment. The FLEX name and description of this column are as follows:

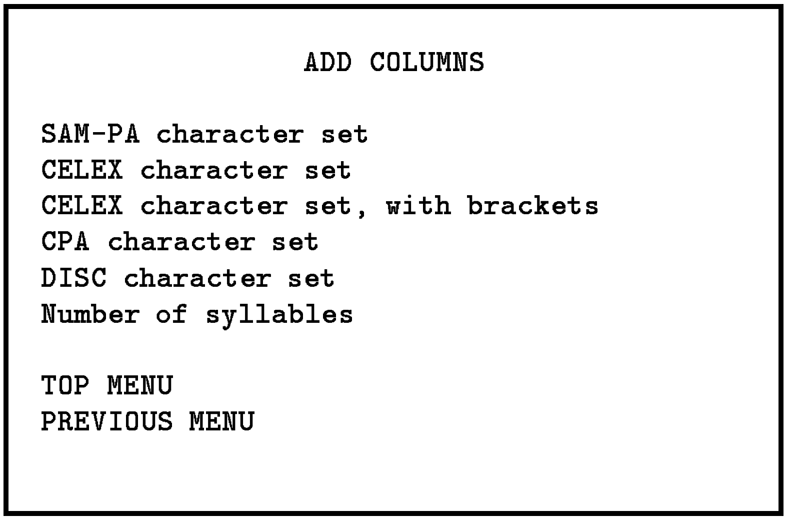
PhonDISC `Phonetic wordform, DISC character set`

The last column in this subsection gives you counts of the number of phonemes in each wordform. Here *phoneme* means the same as *segment* – one phoneme equals a vowel, a consonant, a long vowel, a diphthong, or an affricate. Thus for the word *makelij* the number of phonemes is given as 6, while for *makkelijk* the number is 7. The FLEX name and description of this column are as follows:

PhonCnt `Wordform, number of phonemes`

2.1.2.2 TRANSCRIPTIONS FOR SYLLABIFIED WORDFORMS

This set of transcriptions uses the same basic transcriptions as the first set, except that instead of segment markers, there are characters that mark each phonetic syllable. These are the columns which contain syllabified phonetic transcriptions of each wordform:



```
                ADD COLUMNS

SAM-PA character set
CELEX character set
CELEX character set, with brackets
CPA character set
DISC character set
Number of syllables

TOP MENU
PREVIOUS MENU
```

In most cases transcriptions are syllabified by putting a hyphen (or, in the case of CPA, a full stop) at every syllable boundary within each word. A second method, available with the CELEX character set, is to enclose each syllable within square brackets. The advantage of the brackets notation is that so-called 'ambisyllabic consonants' can be clearly identified. Ambisyllabic consonants are those consonants which come between two syllables, and which belong to both of those syllables. For example, the first [k] of *makkelijk* is part of the first syllable and the second syllable, whereas the [k] of *makelij* belongs to the second syllable only.

The first syllabified wordform transcription column uses the SAM-PA character set, and syllable boundaries within words are shown by hyphens. The FLEX name and description of this column are as follows:

PhonSylSAM Syllabified phonetic wordform, SAM-PA character set

The next two columns both use the CELEX character set. The first marks every syllable boundary within each transcription with a hyphen. The FLEX name and description of this column are as follows:

PhonSylCLX Syllabified phonetic wordform, CELEX character set

The other CELEX syllabified phonetic wordform column uses the brackets notation as described above, and its FLEX name and description are as follows:

PhonSylBCLX Syllabified phonetic wordform, CELEX character set (brackets)

The next column gives syllabified wordform transcriptions in the CPA character set. Every syllable boundary within each word is marked by a full stop. The FLEX name and description of this column are as follows:

PhonSylCPA Syllabified phonetic wordform, CPA character set

The fifth column uses the DISC character set, and here every syllable boundary within each word is marked by a hyphen. The FLEX name and description of this column are as follows:

PhonSylDISC Syllabified phonetic wordform, DISC character set

The last column in this subsection gives counts of the phonetic syllables which occur in each transcription. For example, both *makkelijk* and *makelij* contain 3 syllables. The FLEX name and description of this column are as follows:

SylCnt `Wordform, number of phonetic syllables`

2.1.2.3 TRANSCRIPTIONS FOR STRESSED AND SYLLABIFIED WORDFORMS

This set of columns gives syllabified transcriptions that also mark the points of primary stress in each wordform. These are the columns you can choose from:

```
                 ADD COLUMNS

SAM-PA character set
CELEX character set
CPA character set
DISC character set
Stress Pattern


TOP MENU
PREVIOUS MENU
```

The first column uses the SAM-PA character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show points of primary stress by means of the 'double quote' character ("). This character is placed immediately before a stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsSAM `Syllabified phonetic wordform, with stress marker, SAM-PA character set`

The second column uses the CELEX character set, and as well as using hyphens to mark syllable boundaries, these transcriptions show the points of primary stress with an inverted comma (') immediately before the stressed syllable.

The FLEX name and description of this column are as follows:

PhonStrsCLX `Syllabified phonetic wordform, with stress marker,`
`CELEX character set`

The third column uses the CPA character set, including full stops to mark syllable boundaries, and these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsCPA `Syllabified phonetic wordform, with stress marker, CPA character set`

The fourth column uses the DISC character set, and along with hyphens to mark syllable boundaries, these transcriptions show points of primary stress with an inverted comma (') immediately before the stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsDISC `Syllabified phonetic wordform, with stress marker, DISC character set`

The last column in this subsection contains a simple stress pattern for each wordform. A *stress pattern* is a string which shows how each phonetic syllable is stressed in speech. Each syllable is represented by one numeric character: either `0` or `1`. `1` indicates that the syllable receives primary stress, and `0` that it does not receive primary stress. Thus the three-syllable word *makkelijk* has the stress pattern `100` and *makelij* has the pattern `001`. The FLEX name and description of this column are as follows:

StrsPat `Wordform, stress pattern`

2.2 PHONETIC PATTERNS

Phonetic patterns here means CV patterns: the consonant and vowel patterns for the phonetic transcription (as opposed to the orthographic or phonological transcriptions) of any lemma (headword or stem) or wordform you select. Instead of the basic CV pattern, which uses hyphens to mark phonetic syllable boundaries within words, you may want to use the alternative notation which delimits syllables by means of square brackets. The phonetic CV pattern used here represents each *short vowel* as `V`, each *long vowel* and *diphthong* as `VV`, and each *consonant* and *affricate* as `C`.

In addition, special consideration is made for *ambisyllabic* consonants, such as the `[k]` in the word *makkelijk*. (Ambisyllabic consonants are those consonants which seem to 'belong' to two syllables at once.) The `[k]` is replaced by one `C` at the end of the first syllable, and *another* `C` at the beginning of the second syllable. Thus its CV pattern is `CVC-CV-CVC`. With a brackets notation, the ambisyllabic nature of the consonant can be made clearer: `[CV[C]V][CVC]`.

This table illustrates the two different formats you can choose for you CV patterns:

		CV pattern	CV pattern with brackets
makelij	`[ma:-k@-lEI]`	`CVV-CV-CVV`	`[CVV][CV][CVV]`
makkelijk	`[mA-k@-l@k]`	`CVC-CV-CVC`	`[CV[C]V][CVC]`

2.2.1 PHONETIC CV PATTERNS FOR LEMMAS

For headwords, the basic phonetic CV patterns include hyphens as syllable markers. The FLEX name and description of this column are as follows:

PhonCV
(PhonCVLemma)

`Headword, phonetic CV pattern`

Alternatively you can choose phonetic CV patterns of headwords which use square brackets to delimit the syllables. This column has the following FLEX name and description:

PhonCVBr
(PhonCVBrLemma)

`Headword, phonetic CV pattern, with brackets`

For stems, the basic CV pattern with hyphens as syllable markers are given in the column whose FLEX name and description are as follows:

PhonStCV
(PhonStCVLemma)

`Stem, phonetic CV pattern`

The other column with phonetic CV patterns for stems includes square brackets to delimit syllables. Its FLEX name and description are as follows:

PhonStCVBr
(PhonStCVBrLemma)

`Stem, phonetic CV pattern, with brackets`

2.2.2 PHONETIC CV PATTERNS FOR WORDFORMS

Two phonetic CV pattern columns are available for wordforms. The first uses hyphens to mark syllable boundaries within wordforms, and its FLEX name and description are as follows:

PhonCV `Wordform, phonetic CV pattern`

The second uses square brackets to delimit the syllables in each wordform. Its FLEX name and description are as follows:

PhonCVBr `Wordform, phonetic CV pattern, with brackets`

2.3 PHONOLOGICAL TRANSCRIPTIONS FOR STEMS

The phonological representations provided have been automatically generated using the available CELEX phonological and morphological information. They are available only for the stem form of certain lemmas. Not all stems have phonological representations, but only those with enough information, both phonological and morphological, to make the automatic formation of a transcription possible. The transcriptions given are not necessarily the definitive underlying forms in the strict linguistic sense, though they are certainly abstract (they leave out the information which can be formulated by applying certain phonetic rules to them).

Every transcription gives a phonological representation of each *morpheme* in the stem. When the word consists of more than one morpheme, the boundary between two morphemes is marked in one of two ways: either *type 1* (shown by the symbol +) or *type 2* (shown by the symbol #).

A *type 1* morpheme boundary means (amongst other things) that when the two elements are joined, the morpheme boundary given normally does *not* coincide with the phonetic syllable boundary. Such boundaries usually occur between a stem and a suffix – the transcription for *werker* (i.e. the stem *werk* plus the affix -*er*) is **`wErk+@r`** (CELEX character set).

A *type 2* morpheme boundary means (amongst other things) that when the two elements are joined, the morpheme boundary given often does coincide with the syllable boundary. Such boundaries usually occur between prefixes and stems, or between two stems – the transcription for *werkplaats* (i.e. the stem *werk* plus the stem *plaats* is **wErk#pla:ts** (CELEX character set).

The provision of these two distinct types of morpheme boundary is helpful when you want to investigate rules which govern sound changes in complex words. Each morpheme is given in its original 'underlying' (i.e. a phonological not phonetic) state. The complex word *werkgever* thus has as its transcription **wErk#xe:v+@r**, where the underlying phonological form of the stem *geef* is **xe:v**. Table 4 below sets out the phonological and phonetic transcriptions of the examples so far discussed (plus a few extra) to illustrate the difference between phonological transcriptions and phonetic syllabified transcriptions.

Stem	Phonological Transcription	Phonetic Transcription
werker	**wErk+@r**	**[wEr][k@r]**
werkplaats	**wErk#pla:ts**	**[wErk][pla:ts]**
werkgever	**wErk#xe:v+@r**	**[wErk][xe:][v@r]**
werkgeefster	**wErk#xe:v#st@r**	**[wErk][xe:f][st@r]**
makkelijk	**mAk+@#l@k**	**[mA[k]@][l@k]**
roodachtig	**ro:d#Ax-t@G**	**[ro:t][Ax][t@x]**

Table 4: Phonological vs. phonetic transcriptions

There are a total of 67,911 phonological transcriptions available, so clearly not every stem in the database has such a transcription. There are three reasons why a stem may not be accompanied by a phonological transcription. First, there may not be enough morphological information available to give a full analysis of a particular word. (The Dutch morphological stem column ***Status*** indicates whether or not a complete analysis is available.) Second, there may not be enough phonological information to give a complete transcription. The absence of information for one morpheme in a particular word means that no transcription can be given. Compounds which include abbreviations or proper nouns, for example, thus have no phonological transcriptions. Third, some morphological processes were not covered by the programs which generated the CELEX phonological transcriptions. These include inflectional morphology, stem allomorphy, and affix substitution.

Finally, it should be emphasized that you are dealing here with automatically-generated information; detailed correction by knowledgeable humans has not been carried out. In general, though, these tentative transcriptions are correct so long as the word is regular.

You can choose transcriptions in the CELEX or CPA phonetic character coding sets (see table 2 in section 2.0.1 above). Phonological transcriptions are not available in DISC, however, since that coding set uses the boundary marker codes (# and +) as character codes in their own right. You should note that phonological representations are available only for stems, not headwords or wordforms. Phonological transcriptions are thus available in lemma lexicons, and the names of these columns are the first of the two names given in the margin with each definition. There are no phonological transcriptions for wordforms, but you can see the phonological information for each wordform's stem by using the lemma information given with the morphology columns for Dutch wordforms. The names of these columns are the ones given in brackets directly underneath the lemma lexicon names.

First, the FLEX name and description of the column which gives phonological transcriptions in the CELEX character set:

PhonolCLX
(PhonolCLXLemma)

`Phonological deep structure, CELEX character set`

And second, the FLEX name and description of the column which gives phonological transcriptions in the CPA character set:

PhonolCPA
(PhonolCPALemma)

`Phonological deep structure, CPA character set`

3 DUTCH MORPHOLOGY

Morphological information for Dutch is available with lemma lexicons and wordform lexicons. If you are interested in inflectional morphology, then you should use a wordforms lexicon, and if you are interested in derivational and compositional morphology, you should use a lemma lexicon.

3.1 MORPHOLOGY OF DUTCH LEMMAS

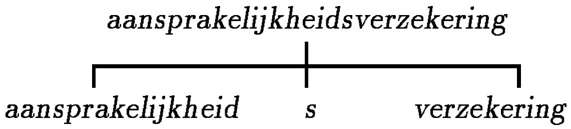
The morphological analyses given for lemmas in the CELEX databases always use the *stem* form of the lemma, because this form is usually the shortest in any inflectional paradigm, without any visible inflectional endings. Before finding out details about each of the columns available, you should look at the sections below which try to give some explanation of the methods used to obtain the analyses given in the database. You will then know what CELEX means by terms such as *immediate segmentation*, *hierarchical segmentation*, *compound*, *derivation*, and *derivational compound*. You will also know how CELEX treats the special 'problem' compound cases which can be treated as derivational compounds *and* ordinary compounds. After all that, you'll understand more clearly what each of the various columns has to offer.

3.1.1 HOW TO SEGMENT A STEM

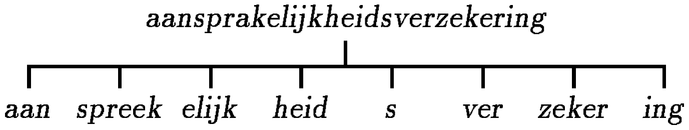
The first and most fundamental type of segmentation is *immediate segmentation*. This simply involves splitting a stem into its largest constituent parts. If you continue to carry out immediate segmentation until there is nothing left to segment, you arrive at the stem's *complete segmentation*. Depending on your requirements, you can look at a complete segmentation in two forms. The first is the *flat* form, which shows every morpheme that makes up the stem. The second is the *hierarchical* form, which, as well as pointing out the individual morphemes in a stem, also shows all the analyses which have to be made to identify those morphemes. The flat segmentation gives the conclusion reached; the hierarchical segmentation shows the working.

To illustrate the three types of segmentation, take as an example the word *aansprakelijkheidsverzekering*.

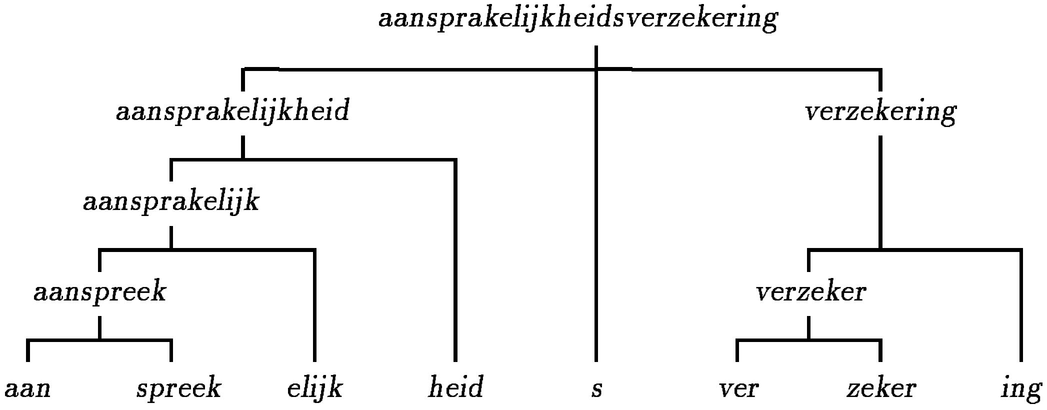
The first type of analysis 'Immediate segmentation' gives the stem *aansprakelijk* plus the affix ('link morpheme') -*s*- plus the stem *verzekering*:



The second type of analysis 'complete segmentation (flat)' shows you what you get if you keep applying immediate segmentation, namely the constituent morphemes of *aansprakelijkheidsverzekering*: the affix *aan* plus the stem *spreek* plus the affix *elijk* plus the affix *heid* plus the affix ('link morpheme') *s* plus the affix *ver* plus the stem *zeker* plus the affix *ing*.



The third type 'complete segmentation (hierarchical)' shows you the full analysis of the word, including each individual immediate segmentation carried out. It gives you enough information to produce a hierarchical tree diagram like this one:



For most stems in the database, representations of each of these three types of segmentation are available. Sometimes there is more than one representation, because certain stems can have more than one immediate segmentation. To explain this fully, the next section describes the basic analyses that result from immediate segmentation.

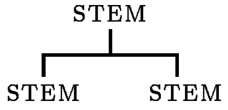
3.1.2 HOW TO ASSIGN AN ANALYSIS

When you attempt to split a stem into its biggest component parts, the result is always some combination of *stems* plus *affixes*. The most straightforward case of all is a stem which consists of only one (free) morpheme: it is *monomorphemic*, and clearly can't be split up. Every other stem, however, consists of one smaller stem plus at least one affix or one other stem, and can be termed either a *Compound*, or a *Derivation*, or a *Derivational Compound*. It is important to understand the differences between these three terms, since they are at the heart of the morphological information CELEX provides. So, in the subsections below, each is defined in terms of stems and affixes. Examples are given, and simple 'tree' diagrams illustrate the appropriate immediate analyses.

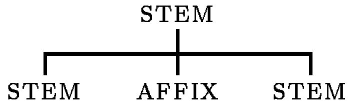
3.1.2.1 THE COMPOUND

A COMPOUND is the joining of two stems into one new stem. The immediate analysis always takes one of two forms:

(i) a binary split into two stems (the word *weerborstel* for example: *weer* + *borstel*).



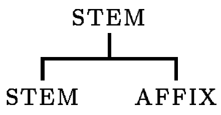
(ii) a triform split into a stem, an affix (simply a 'link' morpheme), and a stem (the word *paardeborstel* for example: *paard* + *e* + *borstel*).



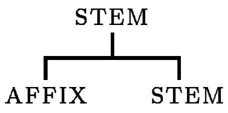
3.1.2.2 THE DERIVATION

A DERIVATION involves affixation, whereby affixes can be added to an existing stem to form a new stem. The immediate analysis always takes one of three possible forms:

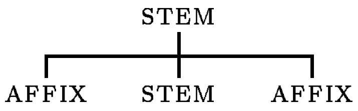
(i) a binary split into a stem and an affix (the word *baanloos*, for example: *baan* + *loos*).



(ii) a binary split into an affix and a stem (the word *medebroeder* for example: *mede* + *broeder*).



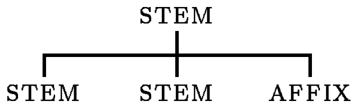
(iii) a triform split into an affix, a stem, and an affix (the word *begenadig* for example: *be* + *genade* + *ig*).



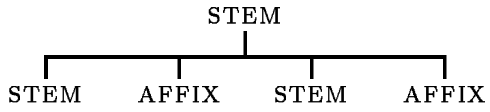
3.1.2.3 THE DERIVATIONAL COMPOUND

A DERIVATIONAL COMPOUND is a compound which can only be formed in combination with a derivational affix (as opposed to a simple link morpheme). The immediate analysis always takes one of two forms:

(i) a triform split into a stem, a stem, and an affix (the word *salarisverhoging* for example: *salaris* + *verhoog* + *ing*).

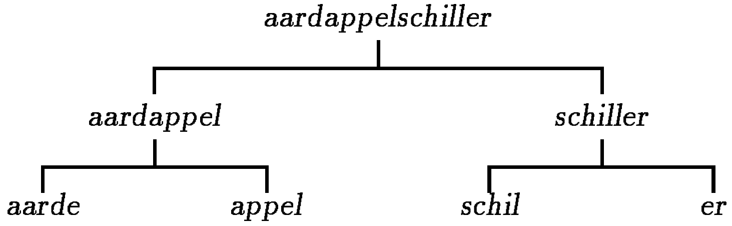


(ii) a quaternary split into a stem, an affix, a stem, and an affix (the word *stationsverbouwing* for example: *station* + *s* + *verbouw* + *ing*).

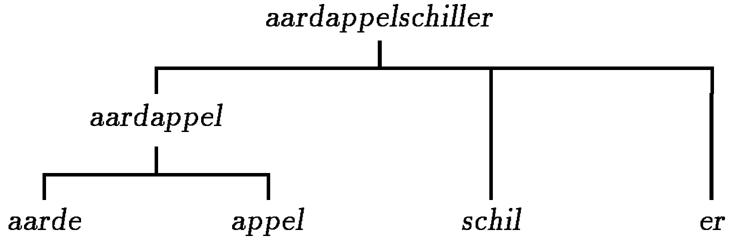


3.1.2.4 COMPOUND OR DERIVATIONAL COMPOUND?

The general definition of a derivational compound is normally sufficient, but when the second stem is a verbal form, things become more complicated. A stem which comprises a noun plus a verb plus an affix can normally be considered a derivational compound, but some people may want to treat it as an ordinary compound. The distinction is important, since it can affect not only the appearance of a single immediate segmentation branch, but also the appearance of a complete hierarchical tree. The stem *aardappelschiller* is such a 'problem' compound. If you consider it to be an ordinary compound (the stem *aardappel* plus the stem *schiller)*, its complete hierarchical tree looks like this:



But if you consider it to be a derivational compound, the first immediate segmentation gives you the stem *aardappel* plus the stem *schil* plus the affix -*er*, which gives the full hierarchical tree a different appearance:



So, when you're faced with a compound that includes a verbal component and an affix, how do you decide whether it's an ordinary compound, a derivational compound, or both? To illustrate the principles used in analysing the information to you, consider the computer program-like algorithms set out below. They take as their initial premise that the word you are looking at can be analysed as a noun, an adverb, an adjective, or a preposition plus a verb and an affix. As the algorithms show, just because they *can* be analysed this way, it is not always true that they *should* be analysed this way. When you come to select columns containing morphological analyses from the database, you can choose for yourself the analysis you want to see. Figuring out these algorithms now will help you to understand the options you can choose from.

First, here are the variables used in the algorithms and their definition:

n is a noun
v is a verb
a is an adjective or an adverb
$prep$ is a preposition
aff is an affix

```
$[n + v + aff]$

if $n$ is the direct object of $v$
then if $[n + v + aff]$ is a specific sort of $v + aff$
        then $[n + v + aff]$ is a COMPOUND
                              and a DERIVATIONAL COMPOUND
        else  $[n + v + aff]$ is a DERIVATIONAL COMPOUND
else  $[n + v + aff]$ is a COMPOUND $[n + n]$
```

How do these rules apply in practice? Take as an example the word *motorrijder*. The first question is whether the noun *motor* is the direct object of the verb *rijd*. The answer is yes, so move to the 'then' clause for the next question: is *motorrijder* a specific sort of *rijder*? Again, the answer is yes, so on moving to the next 'then' clause, you get the answer that *motorrijder* is one of those words which can be treated as an ordinary compound *and* as a derivational compound. Its immediate analysis can be noun plus noun *(motor + rijder)* or, as originally suspected, noun plus verb plus affix *(motor + rijd + er)*. In such cases, the CELEX database offers you both analyses of the stem. Using the 'status of analysis' columns, your lexicon can include either sort of analysis or both of them, according to your preference.

Another example: *naamgeving*. The first question is whether the noun *naam* is the direct object of the verb *geef*. The answer is yes, so move to the 'then' clause for the next question: is *naamgeving* a specific sort of *geving*? Here the answer has to be no, since the word *geving* does not exist by itself. So, move to the 'else' clause to discover that *naamgeving* can only be a derivational compound. Its immediate analysis is thus noun plus verb plus affix: *naam + geef + ing*.

One last example: *gewoontedrinker*. The first question is whether the noun *gewoonte* is the direct object of the verb *drink*. The answer this time is quite clearly no, so move straight to the last 'else' for the answer: *gewoontedrinker* is just an ordinary compound with the simple binary split into a noun plus a noun: *gewoonte + drinker*.

There is also a simple algorithm for stems which can be analysed as adjective or adverb plus verb plus affix:

```
$[a + v + aff]$

if $[a + v + aff]$ is a specific sort of $[v + aff]$
and if $[a + v + aff]$ means the same as $[(det)\ a\ n]$
then $[a + v + aff]$ is a COMPOUND $[n + n]$
else $[a + v + aff]$ is a DERIVATIONAL COMPOUND
```

This time there are two questions which have to be answered together. If one answer, or neither answer, is positive, then the stem is a derivational compound. If both answers are positive, then the stem is an ordinary compound. Thus with the stem *hoogspringer*, the first question is whether it is a particular type of *springer*—and the answer is yes. The second question is whether *hoogspringer* means the same as *(een) hoge springer*—and the answer is no. So, since one of the two answers is negative, you must go to the 'else' clause. This tells you that the stem is a derivational compound.

In fact, most adjective-or-adverb-plus-verb-plus-affix stems are derivational compounds; you won't often find a stem that produces a positive answer to both the questions.

Another important category to consider here is the preposition plus verb plus affix combination. Usually, they can be analysed simply as verb plus affix, i.e. as simple derivations. However on occasions such stems can better be analysed as derivational compounds. The algorithm below indicates when:

```
$[prep + v + aff]$

if $[prep + v]$ is an existing verbal stem with
        the equivalent meaning
then $[prep + v + aff]$ is a DERIVATION
else $[prep + v + aff]$ is a DERIVATIONAL COMPOUND
```

Take as an example the word *afbetaler*. The question is whether the verb *afbetaal* is a verb that exists in its own right, and the answer is yes. Naturally this analysis takes account of the meaning of the word – if *afbetaler* did not mean *iemand die afbetaalt* then clearly the analysis would be wrong. So, the answer yes lets you move onto the 'then' clause, where you find out that the stem is in fact a derivation with an immediate two-part analysis of verb plus affix.

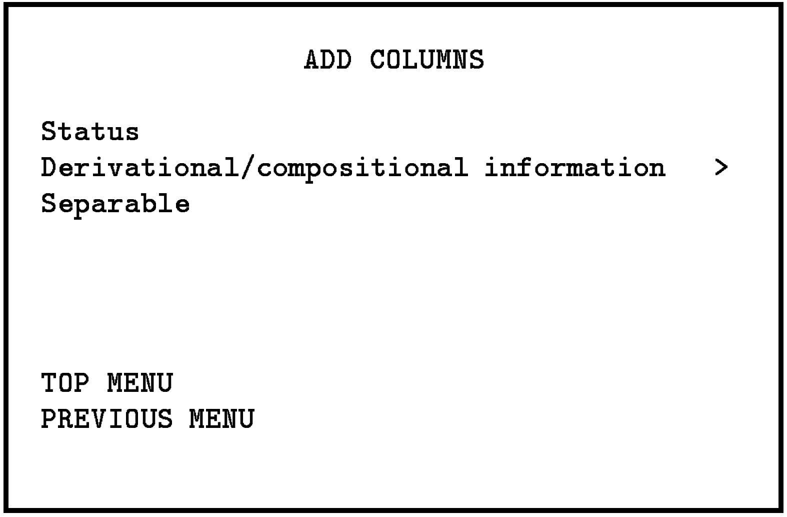
Another example is the word *bijrijder*. Here the verb *bijrijden* does not exist, so the 'else' option indicates that this word is a derivational compound with a triform immediate analysis of preposition plus verb plus affix.

These detailed definitions and explanations are given so you know what to expect when you ask for morphological analyses of stems. You can control the number of analyses you see for each stem, as well as the type of analyses, by means of restrictions on the 'number' and 'status' columns which are defined below. You can decide for yourself whether your lexicon should contain just one 'default' analysis per stem, or whether it should contain more than one analysis per stem. In cases where a stem can be analysed as a compound or a derivational compound, you can choose to include whichever type you prefer, leaving out the other type. In short, you have the freedom to build lexicons which contain morphological information in the form you most prefer.

Having set out much of the theory behind the morphological analyses provided by CELEX, it's now possible to discuss the columns themselves, and this is done in the sections which follow.

3.1.3 STATUS AND SEPARABLE

The first `ADD COLUMNS` menu you see after you select the 'Morphology' option is this one:



```
                    ADD COLUMNS

Status
Derivational/compositional information   >
Separable




TOP MENU
PREVIOUS MENU
```

Before dealing with the various derivational/compositional information columns, which form the bulk of the available morphological information, the first and third columns can be quickly dealt with here.

The first column simply tells you by means of a single code whether each stem is morphologically simple, morphologically complex, or why it is as yet unanalysed. These are the codes that are used:

Status	Code	Example
Morphological analysis available:		
Morphologically complex	`C`	*voetbalschoen*
Monomorphemic	`M`	*beer*
Morphological analysis unavailable:		
Morphology irrelevant	`I`	*Aalsmeer*
Lexicalised flection	`F`	*aaneenschakelend*
Morphology undetermined	`U`	*halvarine*

Table 5: Derivational morphology status codes

If a stem contains at least one stem plus at least one other stem or affix, then it is said to be morphologically complex. Details of how the stem can be analysed are given in the derivational/compositional segmentation columns described in the section below. Thus if a stem has the morphological status code `C` for ‘complex’, you know that information about its derivational and/or compositional morphology are available in the database.

If a stem is monomorphemic, then it contains only one morpheme, and no further analysis is required. The morphological status code M means 'monomorphemic', and you know that a simple one-stem analysis is given as the derivational and/or compositional morphology for each stem with this code.

Sometimes morphological analysis is not appropriate for a particular stem. Usually this is true when the stem involves a proper noun in some way (*Ajaccio* or *Engels*, for example), or when the stem has an extended or sentence-like structure (such as the phrase *jan-in-de-zak*), or when the stem is an interjection (for example *asjemenou*). Thus when a stem has the code I for 'irrelevant', you know that a morphological analysis isn't considered necessary, and that its entries in the segmentation columns described below are therefore empty.

On occasions, a particular flectional form of a stem occurs very frequently, or acquires a meaning slightly different from that of the original stem. For this reason, they can be given stem status in their own right, rather than being considered mere flections. Typically, present and past participles become independent adjectives. In the *Woordenlijst van de Nederlandse Taal*, the word *begrensd* is listed as a bold-type entry in its own right as well as a flection of the verb *begrenzen*. Forms such as these are called *lexicalised flections*. For the CELEX database, any such word which appears as a bold-type headword in either the *Woordenlijst van de Nederlandse Taal* or the *Van Dale Groot Woordenboek van Hedendaags Nederlands* is given the morphological status code F for 'flection'. The Morphological properties of such words are given with the inflectional information available in the 'Morphology of Dutch wordforms' columns. For this reason, no analyses are given for them with the compositional and derivational information.

The last of the morphological status codes is the one which covers everything else. It simply means that the stems in question couldn't be satisfactorily analysed, for a variety of reasons. Some stems use classical affixes, which don't behave quite like normal Dutch affixes (*megafoon* for example), other stems are recent foreign loanwords which aren't always normal productive Dutch stems (as in *half-time*), and others are just plain weird (as in *hakkepoffer*). In all such cases the morphological status code is U for 'undetermined', and no analyses are given.

This column can be used to eliminate from your lexicon stems for which there are no morphological analyses, allowing you to concentrate on those which do. Simply add a restriction which states that you only want stems which are morphologically complex: `MorphStatus = C`.

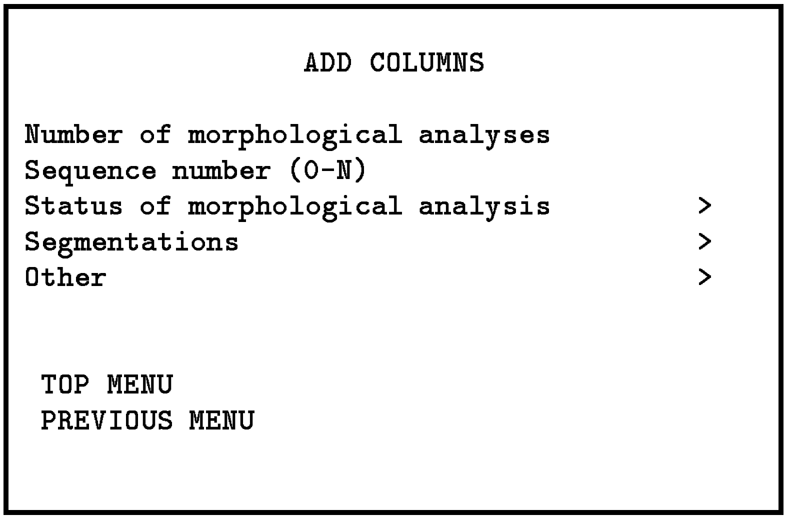
The column which contains these morphological status codes has the following FLEX name and description:

MorphStatus
(MorphStatusLemma) `Morphological status`

The third option deals with *separable* stems: those stems—mostly verbs—whose wordforms sometimes split into two parts, depending on the structure of the sentence they are used in. The stem *tegenzit*, for example, is the same stem whether it occurs in a phrase like *Als het weer tegenzit, dan gaan we niet* or in a phrase like *De omstandigheden zitten tegen, dus beleggen we voorlopig geen geld op de Beurs*. So, if any wordforms of a stem can occur in this way, this column includes the code `Y`. If not, the code given is `N`. This column can be used in the construction of a restriction which specifically includes such stems in your lexicon or specifically excludes them from your lexicon. The FLEX name and description of this column are as follows:

Sepa
(SepaLemma) `Separable`

3.2 DERIVATIONAL/COMPOSITIONAL INFORMATION



```
                    ADD COLUMNS

Number of morphological analyses
Sequence number (0-N)
Status of morphological analysis         >
Segmentations                            >
Other                                    >


 TOP MENU
 PREVIOUS MENU
```

These options give you information about the derivational and compositional morphology of *stems*, including how many analyses are available for each stem, a unique number for each analysis, an indication of the way in which each analysis has been made, and a marker for the 'default' analyses for each stem.

The first option is a column which simply indicates how many analyses have been made for each stem. For example, *bebouwbaar* has one analysis, *varkensfokker* has two, and *kerkhervorming* three. The number of analyses for each stem also equals the number of rows that stem can have with distinct analyses, since each morphological analysis is assigned to its own individual row.

You can use this column to construct restrictions for your lexicon. A simple example would be one that includes in your lexicon only those stems which have more than one analysis. This would take the form `MorphCnt > 1`. The FLEX name and description of this column are as follows:

MorphCnt
(MorphCntLemma) `Number of morphological analyses`

The second option is a column which identifies each analysis of a particular stem. Each different morphological analysis of a stem is assigned to a different row, and this column gives the number of the row. Thus lemma number 108138 (the number in the column ***idNum***, whose stem is *varkensfokker*), has two rows: one has the ***MorphNum*** 1, the other has the ***MorphNum*** 2. The FLEX name and description of this column are as follows:

MorphNum
(MorphNumLemma) `Morphological analysis ID`

Under the 'status of morphological analysis' option there are three 'yes/no'-type columns which, when you use them to construct restrictions, can help you extract the analyses you want from the many stem segmentations available.

Each distinct morphological analysis of each stem has a number, and is given (in several different forms) on its own row in the database. These columns give simple information about each analysis, and are particularly useful whenever a stem is a 'problem' compound, or whenever it contains a 'problem' compound. (A problem compound, as discussed in section 3.1.2.4 , can correctly be analysed as a derivational compound or an ordinary compound.) The three columns in question are called ***DerComp, Comp***, and ***Def***.

Whenever ***DerComp*** contains a `Y`, you know that 'yes, any problem compounds which occur anywhere in this stem are analysed as derivational compounds'. And naturally, `N` means that problem compounds *aren't* analysed as derivational compounds

DerComp (DerCompLemma) `Derivational compound analysis method`

Whenever ***Comp*** contains a `Y`, you know that 'yes, any problem compounds which occur anywhere in this stem are analysed as ordinary compounds'. And again, `N` means that any problem compounds *aren't* analysed as ordinary compounds.

Comp (CompLemma) `Compound analysis method`

Whenever ***Def*** contains a `Y`, you know that 'yes, this analysis is the default analysis'. If a stem includes a problem compound, then there are *two* default analyses with a `Y` in this column, one with the derivational compound type analysis, the other with the ordinary compound type analysis.

Def (DefLemma) `Default analysis`

To illustrate how you can use these columns, imagine that you have chosen ***Imm*** as the form of morphological analysis you want to see (this column, and the other columns containing the same analysis in different forms, are described in the sections following this one). Then say that you are interested in the stem *hartvervetting*, which has four different analyses. First, it is one of the problem compounds which can be a derivational compound or an ordinary compound, which accounts for two analyses. And second, the stem *vet* is analysed as a noun, but might also be thought of as an adjective, which gives the other two analyses.

First you can decide whether you want just one default analysis, or whether you want to see all the available analyses.

If you want to see all its possible segmentations, then you don't need to add extra restrictions. As the ***MorphCnt*** column indicates, there are 4 analyses given for this stem, *hartvervetting*, so this is what the unrestricted example lexicon looks like:

Stem	MorphNum	DerComp	Comp	Def	Imm
hartvervetting	1	Y	N	Y	hart+vervet+ing
hartvervetting	2	Y	N	N	hart+vervet+ing
hartvervetting	3	N	Y	Y	hart+vervetting
hartvervetting	4	N	Y	N	hart+vervetting

Analyses numbers 1 and 2 are derivational compounds, so in these two cases ***DerComp*** contains `Y`, and ***Comp*** contains `N`. Analyses numbers 3 and 4 are ordinary compounds, so there ***Comp*** contains `Y`, and ***DerComp*** contains `N`.

However, rather than including all four forms in your lexicon, you might want to ignore the ordinary compound analyses, and just see the derivational compound analyses. To do this for all the stems in the database, you should add an 'expression' restriction to your lexicon which states that `DerComp = Y`. In the example lexicon, this one restriction produces the following result:

Stem	MorphNum	DerComp	Comp	Def	Imm
hartvervetting	1	Y	N	Y	hart+vervet+ing
hartvervetting	2	Y	N	N	hart+vervet+ing

In the same way, if you want to ignore the derivational compound analyses in favour of the ordinary compound analyses, you should add an 'expression' restriction to your lexicon which states that `Comp = Y`. In the example lexicon, this restriction produces the following result:

Stem	MorphNum	DerComp	Comp	Def	Imm
hartvervetting	3	N	Y	Y	hart+vervetting
hartvervetting	4	N	Y	N	hart+vervetting

Rather than seeing a number of analyses, you might prefer to look at just one straightforward default analysis, no matter how many alternatives are given in subsequent rows. Again, you can quickly construct restrictions to make this possible. The quickest way is to use the ***MorphNum*** column, which gives a number to each analysis of each stem. You can say `MorphNum = 1`, which means that only the very first analysis of each stem appears in your lexicon. And whenever a stem is a problem compound, you should remember that the first analysis is always the derivational compound form rather than the ordinary compound form.

Another way to get a single analysis for each stem with problem compounds treated as derivational compounds is to add these two restrictions: `Def = Y` and `DerComp = Y`. Here you are saying explicitly that you want the default form of the stem (in the example lexicon that means ignoring the '*vet* is an adjective' analysis) and that whenever problem compounds occur, you want to see the derivational compound form.

Whether you choose the single ***MorphNum*** restriction or the two ***Def*** and ***DerComp*** restrictions, the effects on your lexicon are the same. The resulting example lexicon looks like this:

Stem	MorphNum	DerComp	Comp	Def	Imm
`hartvervetting`	1	Y	N	Y	`hart+vervet+ing`

If you want one analysis, and if in the case of problem compounds you want that one analysis to be an ordinary compound rather than a derivational compound, all you have to do is add two restrictions. First, ask for a default analysis by saying `Def = Y`; this omits the non-preferred analyses like the '*vet* is an adjective' option. Then specify that you want any problem compounds to be given as ordinary compounds by adding the restriction `Comp = Y`. This is what the example lexicon then looks like:

Stem	MorphNum	DerComp	Comp	Def	Imm
`hartvervetting`	3	N	Y	Y	`hart+vervetting`

These explanations may appear complicated, but by reading them, you can get to know the important restrictions that you can use to extract the types of analysis you really want.

3.2.1 IMMEDIATE SEGMENTATION

Immediate segmentation is the least detailed form of analysis offered here. It doesn't give you a full analysis, right down to all the smallest elements a stem contains; rather it is a simple, one-level breakdown of a stem into its next biggest elements. So, while complete segmentation is equivalent to a full analytical tree, immediate analysis can be thought of as a close look at a particular level.

There are six columns which present the immediate segmentation of stems to you. The first gives the orthography of the analysed elements. The next three give more general codings, so that using the FLEX options `SHOW` and `QUERY`, you can look for stems which have a particular form: a preposition plus a noun, say, or a stem plus a stem plus an affix, and so on. The last two indicate when stem allomorphy or affix substitution occurs in the immediate analysis of a stem.

In the first column, you get the orthography of the first-level elements themselves, each separated by a `+` sign. Diacritical markers are not included. Thus the stem *bijrijder* is shown as `bij+rijd+er`, in accordance with the various rules discussed in section 3.1.2.4 . Note that each element is given in the form of a stem or an affix, even when the original word doesn't use that particular form. Thus the stem *aansprakelijk* is analysed as *aanspreek + elijk*, where *aansprak* is re-written in the form of the stem *aanspreek*. The FLEX name and description of this column are as follows:

Imm `Immediate segmentation`
(ImmLemma)

The second column is like the first, except that where the first column gives you the orthography of each element, this column gives you the word class of each element. Single letter labels are used to represent the syntactic class of each element – which is unlike many of the syntactic codes used in other parts of the database. The use of a single character means that there is no possibility of a code becoming ambiguous, since each character is unique. Table 6 shows you the labels used in this column:

Word Class	Label
Noun	`N`
Adjective	`A`
Quantifier/Numeral	`Q`
Verb	`V`
Article	`D`
Pronoun	`O`
Adverb	`B`
Preposition	`P`
Conjunction	`C`
Interjection	`I`
Abbreviation	`X`
Affix	`x`

Table 6: Word class labels (immediate segmentation)

Using these codes, the stem *bijrijder* is given the code `PVx`, indicating that it is made up of a preposition, a verb, and an affix. The word *nettenknoopster* has the code `NxVx`. The FLEX name and description of the column that gives you these codes are as follows:

ImmClass
(ImmClassLemma)

`Immediate segmentation, word class labels`

The third column provides more detailed information about the syntactic categorization of verbal stems. The basic codes used are exactly the same as the `ImmClass` column, except that instead of the `V` code to represent a verb, any one of a number of codes is given. Table 7 shows you these codes, along with their meaning.

Verbal sub-category	Label
Intransitive	`1`
Transitive	`2`
Intransitive & transitive	`3`
Reflexive	`4`
Intransitive & reflexive	`5`
Transitive & reflexive	`6`
Intransitive, transitive & reflexive	`7`
Non-lexical verb	`0`

Table 7: One-character verbal subclass labels

In this column, the word *nettenknoopster* has the code `Nx2x`. It is exactly the same as the code in the previous column, except that the `V` is replaced by the number `2`, indicating in more detail what sort of verb it is.

The FLEX name and description of this column are as follows:

ImmSubcat `Immediate segmentation, subcat labels`
(ImmSubcatLemma)

The fourth immediate segmentation column simply tells you whether the elements identified are stems or affixes. Upper case `S` indicates a stem, upper case `A` indicates an affix. Thus the stem *nettenknoopster* is represented as `SASA`. The FLEX name and description of this column are as follows:

ImmSA `Immediate segmentation, stem/affix labels`
(ImmSALemma)

The fifth immediate segmentation column concerns stem allomorphy. Within words, stems sometimes take a form different from their generally accepted stem form. When morphological analysis is noted down, any resulting stems are given their normal stem form, because that is the most appropriate form which occurs in Dutch. An example is the word *aansprakelijk*, which comprises the stem *aanspreek* and the affix *elijk*: note the difference between *aansprak* and *aanspreek*, where the one element is spelt two different ways. This is stem allomorphy. This column indicates whether or not stem allomorphy occurs in its immediate segmentation. The code `Y` means that it does occur, the code `N` that it does not. The FLEX name and description for this column are as follows:

ImmAllo `Stem allomorphy, top level`
(ImmAlloLemma)

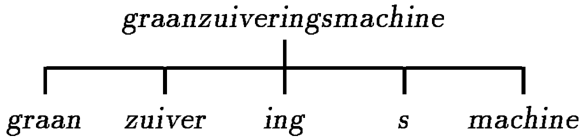
The last of the six immediate segmentation columns marks those stems whose morphological analysis involves *affix substitution*. This is the process whereby an affix replaces part of a stem when that stem and the affix join to form another stem. For example, *immigratie* is analysed as the stem *immigreer* and the affix *-atie*: the affix *eer* has disappeared, and the new affix *-atie* has taken the place of the old one. So, this column gives `Y` for yes if the immediate analysis of the stem involves affix substitution, or `N` for no if it does not.

The FLEX column name and description of this column are as follows:

Subst* *(SubstLemma) `Affix substitution, top level`

3.2.2 COMPLETE SEGMENTATION (FLAT)

Complete segmentation is 'complete' in the sense that it identifies all the morphemes a stem contains. This is in contrast to immediate segmentation, which only picks out the next two (sometimes three or four) morphological elements. The complete segmentation discussed in this section is also *flat*, which means that you can see what the constituent morphemes are without knowing the details of the full morphological analysis which has been carried out. When you draw a morphological 'tree diagram', this information gives the outermost branches only; you cannot analyse any further, and you cannot see the intermediate levels. So, when you want to see the complete, flat, segmentation of *graanzuiveringsmachine* for example, you get this sort of information:



There are three columns with complete segmentation (flat) information. The first contains the morphemes themselves. The second contains the word class of each morpheme, and the third simply states whether each morpheme is a stem or an affix. The last two columns are useful when you're looking for a stem with a particular combination of morphemes: using the FLEX **SHOW** and **QUERY** options, you can hunt out stems which are made up of a noun plus an affix plus a noun, say, or all the stems which contain at least three other stems.

The first column gives you each stem split into its morphemes by + signs. Thus the stem *graanzuiveringsmachine* is written in the following way:

```
graan+zuiver+ing+s+machine
```

No diacritics are included. The FLEX name and description of this column are as follows:

Flat* *(FlatLemma) `Flat segmentation`

The second column uses single-letter codes to represent the word class of each morpheme.

Word Class	Label
Noun	`N`
Adjective	`A`
Quantifier/Numeral	`Q`
Verb	`V`
Article	`D`
Pronoun	`O`
Adverb	`B`
Preposition	`P`
Conjunction	`C`
Interjection	`I`
Abbreviation	`X`
Affix	`x`

Table 8: Word class labels (flat segmentation)

Using these codes, the stem *graanzuiveringsmachine* is given as `NVxxN`. The FLEX name and description of the column are as follows:

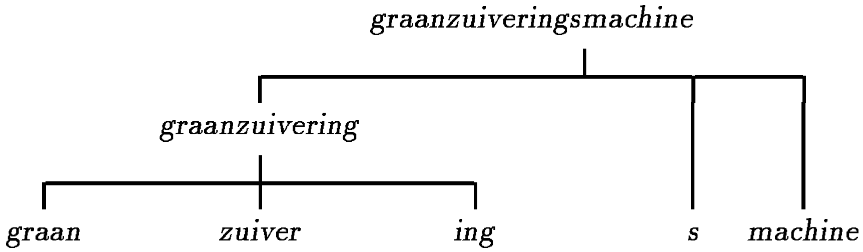
FlatClass* *(FlatClassLemma) `Flat segmentation, word class labels`

The last column simply indicates whether each morpheme is a stem or an affix. Upper case `S` means Stem, and upper case `A` means Affix. The full code for *graanzuiveringsmachine* is thus `SSAAS`. The FLEX name and description of this column are as follows:

FlatSA* *(FlatSALemma) `Flat segmentation, stem/affix labels`

3.2.3 COMPLETE SEGMENTATION (HIERARCHICAL)

Complete, hierarchical segmentation gives the most detailed analysis available for each stem. It is called *hierarchical* because it can cover several different levels: it is arrived at after immediate analysis has been carried out on every stem that can be identified within a larger stem. With this information, you can draw a complete morphological 'tree diagram', from the root to the outermost branches, with every intermediate branch fully represented. So, for the stem *graanzuiveringsmachine*, you can get the following morphological analysis:



There are five columns which give information about the full segmentations of stems. Three of them give the hierarchical segmentations themselves. The simplest of these tells you what the constituent morphemes of the stem are, indicating with algebra-like brackets the structure of the 'tree'. Also available are similar bracket notations which supply a word class label alongside each morpheme on each level, or the word class without the morpheme itself. The remaining two columns indicate whether stem allomorphy or affix substitution has occurred anywhere in the full hierarchical analysis.

The first column provides all the information you need to draw a tree diagram like the one above – that is, the constituent morphemes of a stem each delimited by a comma and enclosed in brackets which indicate its complete morphological structure. The stem *graanzuiveringsmachine* thus looks like this:

```
(((graan),(zuiver),(ing)),(s),(machine))
```

Each identifiable stem or affix is enclosed by a pair of brackets, beginning with the brackets round the full original stem. Then there is a pair of brackets round each of the three elements of the derivational compound *graanzuivering*, and finally a pair of brackets round each of the five constituent morphemes.

The FLEX name and description of the column which contains morphological analyses in this form are as follows:

Struc **(*StrucLemma*)** `Structured segmentation`

The next two columns use extra labels to indicate the word class of each segment. They are given between square brackets to the right of each closing round bracket, so that every segment on every level within the original stem has a word class code. The word class codes used are as follows:

Word Class	Label
Noun	N
Adjective	A
Quantifier/Numeral	Q
Verb	V
Article	D
Pronoun	O
Adverb	B
Preposition	P
Conjunction	C
Interjection	I
Abbreviation	X

Table 9: Word class labels (complete segmentation)

The codes used for affixes are combinations of these word class labels. The stem *graanzuiveringsmachine* can be represented as follows:

```
(((graan)[N],(zuiver)[V],(ing)[N|NV.])[N],(s)[N|N.N], (machine)[N])[N]
```

This example illustrates the special form affix codes take. There are two elements in each affix code which are separated by a vertical bar |. In front of the vertical bar is a single code which is the word class of the stem which the affix in question helps to form. After the vertical bar comes a combination of single letter codes which indicate the word class of each element within the stem formed, and the position of the affix itself is given by a dot.

In the *graanzuiveringsmachine* example above, the code given alongside the affix *ing* is `[N|NV.]`. The `N` before the bar means that the affix *ing* helps to form a stem which is a noun *(graanzuivering)*. The `NV.` after the bar means that the segmentation of the noun *graanzuivering* is noun plus verb plus affix. These detailed codes can help you to identify the way affixes are used, and to get lists of stems which contain affixes used in particular contexts: the fact that the second part of the *ing* code is `NV.` helps you to see at once that this affix helps to form a derivational compound, in conjunction with a noun and a verb.

Sometimes a pair of affixes can only be used together, as in the word *geboomte* – the word *boomte* does not exist and the word *geboom* does not exist. In such cases, `x` marks the other part of the affix, and denotes that the affixes must occur in combination with each other: so-called *split affixes*. The code for the *ge-* of *geboomte* is thus `[N|.Nx]`, and the code for the *-te* is `[N|xN.]`.

So, this column is particularly useful for two things. First, you get the word class of each stem in the segmentation alongside the orthographic representations of individual morphemes. Second, you get detailed information about each affix each stem contains. The FLEX name and description of this column are as follows:

StrucLab `Structured segmentation, word class labels`
(StrucLabLemma)

The next column shows the hierarchical structure of each stem by means of round brackets and commas, and the full word class labels between square brackets, just as with the previous column. The only difference is that in this column the orthographic representation of the constituent stems and affixes is missed out altogether. Thus the stem *graanzuiveringsmachine* gets the following representation:

```
((()[N],()[V],()[N|NV.])[N],()[N|N.N],()[N])[N]
```

This column again helps you to search for stems which have a particular morphological structure and particular combinations of syntactic elements. The FLEX name and description of this column are as follows:

StrucBrackLab `Structured segmentation, word class labels only`
(StrucBrackLabLemma)

The fourth hierarchical segmentation column deals with stem allomorphy. Within words, stems sometimes take a form different from their generally accepted stem form. When a morphological analysis is noted down, the resulting stems are given their normal stem orthography, because that is the most appropriate form which occurs in Dutch. An example is the word *aansprakelijk*, which comprises the stem *aanspreek* and the affix *elijk*: note the difference between *aansprak* and *aanspreek*, where the one element is spelt two different ways. This is stem allomorphy. This column indicates whether or not stem allomorphy occurs at any point in a stem's complete hierarchical segmentation. The code `Y` means that it does occur, the code `N` that it does not. The FLEX name and description for this column are as follows:

StrucAllo `Stem allomorphy, any level`
(StrucAlloLemma)

The fifth and last column marks stems whose morphological analysis involves *affix substitution*. This is the process whereby an affix replaces part of a stem when that stem and the affix join to form another stem. For example, *immigratie* is analysed as the stem *immigreer* plus the affix *-atie*: the affix *eer* has disappeared, and the new affix *-atie* has taken the place of the old one. So, this column gives `Y` for yes if the complete analysis of the stem involves affix substitution, or `N` for no if it does not. The FLEX name and description of this column are as follows:

StrucSubst `Affix substitution, any level`
(StrucSubstLemma)

3.3 OTHER CODES

The remaining three columns give counts of various sorts: the number of *components* (i.e. stems and affixes) in the immediate analysis of each stem, the number of *morphemes* each stem contains, and the number of *levels* involved in the complete hierarchical analysis of each stem.

The first of these columns is the simple count of the number of components each stem contains. The normal figure is two; words are generally split into two parts each time one level of morphological analysis takes place. Sometimes three components can be identified: derivational compounds are usually analysed as a stem plus a stem plus an affix, as are normal compounds which are joined with *-s-*, the special 'link morpheme'. Derivational compounds occasionally contain four elements, stem plus affix plus stem plus affix. And of course, monomorphemic words only contain one component. Any stems which have not yet received an adequate morphological analysis (for the reasons given in section 3.1.3) get the number 0.

Some examples: the number of components in the stem *aansprakelijkheidsverzekering* is three *(aansprakelijkheid + s + verzekering)*, and for the stem *weerborstel* it is two *(weer + borstel)*.

The FLEX name and description of this column are as follows:

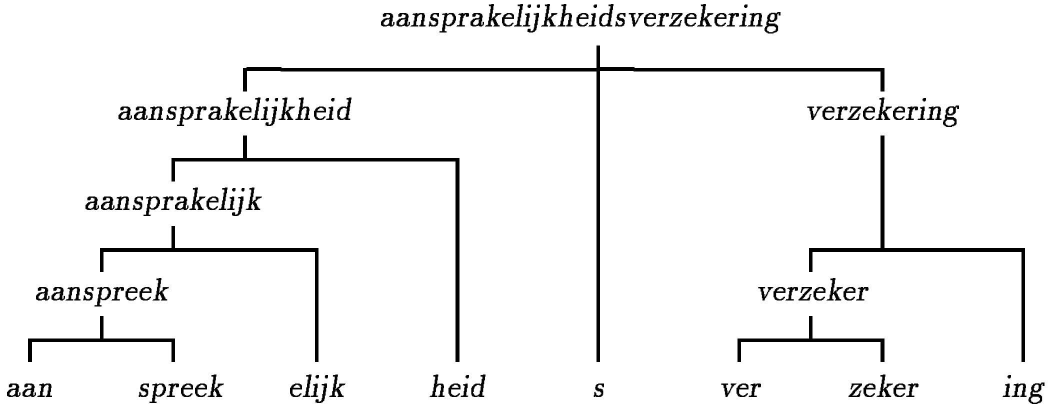
CompCnt `Number of morphological components`
(CompCntLemma)

The second column gives you the number of morphemes in each stem. For words without a morphological analysis, the number given is zero. The number of morphemes in the stem *aansprakelijkheidsverzekering* for example is eight, while for *weerborstel* it is two.

The FLEX name and description of this column are as follows:

MorphCnt `Number of morphemes`
(MorphCntLemma)

The last of the three columns gives a count of the number of levels in the complete hierarchical segmentation described above, which is best illustrated by means of a tree diagram:



Including the stem at the top, the diagram covers five lines: this is the *number of levels* the stem has. It is the number of times you can carry on doing immediate analysis when you analyse a particular stem in full. Do not confuse it with the number of all the immediate analyses required to arrive at the complete hierarchical segmentation (which for *aansprakelijkheidsverzekering* is six); any one *level* of analysis may include more than one immediate segmentation. Monomorphemic stems always get the number 1, while stems without analysis (for reasons explained in section 3.1.3) get the number 0.

The FLEX column name and description of this column are as follows:

LevelCnt (LevelCntLemma) `Number of morphological levels`

3.4 MORPHOLOGY OF DUTCH WORDFORMS

There are two types of morphology information available for the 400,000 wordforms given in the CELEX database: first, information about the lemma which underlies each family of wordforms, and second, a simple identification of the inflectional features which are specific to each wordform, either in the form of seventeen 'yes/no' feature columns or one column with feature identification codes.

Dictionaries present their lexical information under bold-type headwords, which are used instead of listing every individual inflected form separately. Such a form is often called the *canonical form*, since it represents a full canon of inflections. Thus the word *eet* is understood as referring not only to the form *eet* itself, but also the forms *eten, gegeten, at*, and *aten*. To print full details about every inflected form separately would result in a lot of needless repetition and enormous books which no one could lift from the bookshelf. However, for many applications, lemma information has to be listed for each individual wordform, and in a CELEX lexicon of type wordform, you can do just that when you include certain 'morphological' columns. This is done by providing a link between the wordform information and the lemma information. When you choose the option `Lemma information` from the `ADD COLUMNS` menu, you are in fact being allowed into the lemma information by the back door. You can now look up information specific to a particular wordform in your lexicon, and at the same time see general information which is common to all the other forms in the same inflectional paradigm. One particularly useful type of lemma information you can use in your wordform lexicon is the syntactic information, which can give the word class of any wordform you are looking at. There is also an important distinction which you may be able to draw upon with the frequency information. The wordform lexicon gives you an INL frequency figure specific to each wordform, while the lemma information available lets you see the sum frequency for all the inflectional forms in the same paradigm, a figure referred to as the *lemma frequency*.

All the lemma information has already been defined elsewhere in this linguistic guide, so there is no point in repeating it all here. All that needs to be pointed out is that the column names used in a real lemma lexicon differ from those used in the lemma information option in the morphology of wordforms. When a FLEX column name and description are defined in the course of lemma lexicon text, the column name given in brackets is the name of the column when it is used as part of a wordforms lexicon. Usually this name is identical to the lemma lexicon name, except that the word *lemma* is added to the end.

ExampleName (ExampleNameLemma)

`The column names used for lemma information in a Wordforms lexicon are given in brackets, as this Example Name shows.`

All the other details and definitions remain the same in both cases. So, when you're looking for the columns of lemma information provided with a wordforms lexicon under morphology, just go back to the original lemma information: it's all there.

3.4.1 INFLECTIONAL FEATURES

There are twenty special columns available only with a lexicon of type wordforms. Each one corresponds to a particular inflectional attribute which a wordform can have. There can only be one of two codes in each column: `Y` for 'yes, this wordform has this attribute', or `N` for 'no, this wordform does not have this attribute'. These columns are therefore useful for constructing restrictions on your lexicons, restrictions which need not be 'on view': it's unlikely that you will want to look at the contents of these columns with the `SHOW` option. (If, on the other hand, you want to have a label which lets you see at a glance all the inflectional features each wordform has, then you should use the 'type of flection' codes described in the next section.)

An example. To make a lexicon which gives you all the wordforms in the database with the exception of the 'separated' forms of verbs, you have to include at least two columns in the wordforms lexicon you create, namely a column which gives the orthographic representations you prefer, and ***Sepa*** (which is amongst the twenty columns described below). You must then construct a restriction for your lexicon which states that ***Sepa*** must be equal to `N`. You can then format your lexicon to make sure that ***Sepa*** is not 'on view': that way, when you `SHOW` or `EXPORT` your lexicon, you just get the list of words you require without the list of `N`'s. To this basic lexicon, you can of course add any other columns you require, either the orthographic and frequency information specific to each wordform, or the general lemma information—particularly syntax—which is available through the 'Morphology of Dutch wordforms' options.

The first inflectional features column marks those wordforms which have two separate parts, even though they 'belong' to a stem or headword which is a single unit. Forms like *gaf door*, *heffen op* and *liepen achterna* have the positive `Y` code, even though their headwords are *doorgeven*, *opheffen*, and *achternalopen*. The FLEX name and description of this column are as follows:

Sepa `Separated wordform`

The second column indicates whether a wordform is a singular form of any sort. Mostly this means verbal forms such as *koop* or *doop om*, or nouns such as *mopneus*. The FLEX name and description of this column are as follows:

Sing `Inflectional feature: singular`

The third column indicates whether a wordform is a plural inflection of any sort. Mostly this means verbal forms such as *kopen* or *dopen om*, or nouns such as *mopneuzen*. Also included are adjectival forms which are used as nouns, such as *welmenenden* or *neo-klassieken*. And there are also a few pronouns like *haren* and *uwen*. The FLEX column name and description of this column are as follows:

Plu `Inflectional feature: plural`

The fourth column marks wordforms (or more specifically nouns) which are diminutives. Forms such as *fopperijtje* or *jaguartje* get the code `Y` to indicate that they are diminutives, and all other forms get the code `N` to indicate that they are not. There are also a few forms which look like diminutives, but still get the `N` code: the proper noun *Aafje*, for example. The FLEX name and description of this column are as follows:

Dim `Inflectional feature: diminutive`

The fifth column marks those wordforms in the database which are genitives. Wordforms like the determiner *des* (as in *de dag des oordeels*) or the noun *christendoms* get the code `Y`, as do the adjectival genitive forms like *moois*. All the other non-genitive forms get the code `N`. The FLEX name and description for this column are as follows:

Gen `Inflectional feature: genitive`

The sixth column marks those few wordforms in Dutch which are datives. Wordforms like *gronde* (as in the phrase *iemand te gronde richten*), or the noun *huize* get the code `Y`, while all the other non-dative forms get the code `N`. The FLEX column name and description of this column are as follows:

Dat `Inflectional feature: dative`

The seventh column marks all the wordforms which are positive forms – that is, not comparative or superlative forms like *beter* and *best*, but plain adjectival forms like *goed* or *vaak*. Thus adjectives like *centraal* and *centrale* or *lubberig* and *lubberige* get the code `Y`, while all other forms get the code `N`. The FLEX name and description of this column are as follows:

Pos `Inflectional feature: positive`

The eighth column marks all the wordforms which are comparative forms, almost always adjectives. Wordforms such as *beter* or *losbolligere* or *talentrijker* thus get the code `Y`, while all other non-comparative forms get the code `N`. There is also a small number of comparative adverbs which get the `Y` code, such as *vaker*. The FLEX name and description of this column are as follows:

Comp `Inflectional feature: comparative`

The ninth column marks all superlative forms, so that wordforms such as *best* or *centraalst* get the code `Y`, and every other form gets the code `N`. There is also a small number of superlative adverbs which get the `Y` code, such as *vaakst*. The FLEX column name and description of this column are as follows:

Sup `Inflectional feature: superlative`

The tenth column marks all those wordforms which have the inflectional *-e* ending. There are a number of reasons why this ending may be added. Adjectives which qualify indefinite non-neuter nouns get this ending, as in 'een *leuke* mop', and this applies to comparative forms too, such as 'een *aardigere* dame'. Most past or present participles can get this ending when they are used adjectivally (as in 'het *geadopteerde* kind') or in a substantivized form ('het *geadopteerde*'). In fact, many other words can get such an ending: pronouns (as in '*onze* kinderen' or 'dat zijn de *onze*'), numerals ('de *tweede* jongen'), quantifiers ('*vele* meisjes'), and so on. So if for any of these reasons a wordform ends with the inflectional *-e*, then it gets the code `Y` in this column, and all the other wordforms get the code `N`. The FLEX name and description of this column are as follows:

Suff_e `Inflectional feature: with suffix -e`

The eleventh column marks the form of the verb usually known as the infinitive. It is used as a headword in the CELEX databases, and in most dictionaries. For most verbs, the ending is *-en*: *hebben* or *dimmen*, for example. Some other verbs have slightly different infinitives, such as *zijn* or *staan*. Any wordform which is an infinitive gets a `Y` code in this column; all the others get the code `N`. The FLEX column name and description for this column are as follows:

Inf `Inflectional feature: infinitive`

The twelfth column marks any participles, past tense or present tense. Present participles are normally formed by adding *-end* to the stem of the verb, with the exception of some irregular verbs. Past participles of 'weak' verbs add the prefix *ge-* and the suffix *-en*, *-t* or *-d* to the stem, and they are used in the formation of the perfect tense: 'ik heb twee jaren in Zoetermeer *gewerkt*'. The past participle of a 'strong' verb is usually the same except that a vowel change may also occur within the stem itself: 'ik heb te veel *gedronken*'. Most past participles can also be used adjectivally, as in 'het *gekafte* boek'. Any wordforms which are participles get the code `Y`, and all the rest get the code `N`. The FLEX name and description of this column are as follows:

Part `Inflectional feature: participle`

The thirteenth column identifies any present tense forms, including the present participles mentioned under ***Part***. Thus verb forms like *weck*, *wecken* and *weckende* get the code `Y`, while all other forms (including infinitives, which are marked in a different column) get the code `N`. The FLEX name and description of this column are as follows:

Pres `Inflectional feature: present tense`

The fourteenth column identifies any past tense forms, including the past participles mentioned under ***Part***. In the simple past tense, regular 'weak' verbs add *-te*, *-ten* or *-de*, *-den* to the stem, as in 'ik *werkte*' or 'wij *hoorden*'. There are many other 'strong' verbs, which often just change a vowel sound in the stem, as in 'ik *schreef* een boek'. All past tense forms get the code Y, while all other forms (including infinitives, which are marked in a different column) get the code N. The FLEX name and description of this column are as follows:

Past `Inflectional feature: past tense`

The fifteenth column marks first person singular forms of verbs. Only present tense forms are marked since the past tense forms for first, second, and third person are always the same, and they are marked in a different column. For most verbs, the first person form is the shortest form of the verb and is usually taken as the stem of the verb. So, all first person singular forms, like 'ik *wed*' or *dien op*, are given the code Y, and every other form gets the code N. The FLEX column name and description of this column are as follows:

Sin1 `Inflectional feature: 1st person verb`

The sixteenth column marks second person singular forms of verbs. Only present tense forms are marked since the past tense forms for first, second, and third person are always the same, and they are marked in a different column. For most verbs, the second person form consists of the stem plus the suffix *-t*. When you ask a question with such a form, however, the *-t* disappears, leaving only the stem. So, all second person forms like 'jij *sappelt*' or '*loop* je?' are given the code Y, and every other form gets the code N. The FLEX column name and description of this column are as follows:

Sin2 `Inflectional feature: 2nd person verb`

The seventeenth column marks those second person singular verb forms which are used in questions or in other specific syntactic contexts. Usually, these forms are the same as the stem. Every form like '*loop* je naar huis?' gets the code Y, while non-interrogative second person forms like 'jij *loopt*', get the code N. The FLEX name and description for this column are as follows:

Inv Inflectional feature: inversed

The eighteenth column identifies third person singular forms of the verb. Only present tense forms are marked since the past tense forms for first, second, and third person are always the same, and they are marked in a different column. For most verbs, the third person form consists of the stem plus the suffix *-t*. Thus forms like 'hij *blijft weg*' or 'Gilbert *calligrafeert*' get the code Y while every other form gets the code N. The FLEX name and description for this column are as follows:

Sin3 Inflectional feature: 3rd person verb

The nineteenth column marks the one form in the database which can *only* be used as an imperative. To save you looking it up, here it is: *wees*. Usually, the imperative takes the same form as the present tense first person singular form – which means, of course, that the imperative form is never unique for any verb except *zijn*. So, the only wordform that gets the code Y in this column is *wees*; every other wordform gets the code N. The FLEX name and description for this column are as follows:

Imp Inflectional feature: imperative only

The twentieth and last column marks the three forms in the database which are subjunctive forms, and they are *grave*, *zegge*, and *zij*. They are included because they were found during the disambiguation of the INL text corpus: subjunctive forms are not usually included in the database. These three forms have the code Y in this column, while every other wordform gets the code N. The FLEX name and description of this column are as follows:

Sub Inflectional feature: subjunctive only

3.4.2 TYPE OF FLECTION

In the 'Inflectional Features' section above, twenty different inflectional features are distinguished, and assigned to twenty separate 'yes/no' columns. The same information is also available in one single column, using combinations of single-letter codes to show all the features each wordform has. The 'yes/no' columns are useful for constructing restrictions on your lexicon, whereas the 'type of flection' column described here provides you with a label that identifies at a glance all the features each wordform has. Table 10 below sets out the single-letter codes.

Inflectional feature	Label	'yes/no' column name
Separated wordform	`s`	***Sepa***
Singular	`e`	***Sing***
Plural	`m`	***Plu***
Diminutive	`d`	***Dim***
Genitive	`G`	***Gen***
Dative	`D`	***Dat***
Positive	`P`	***Pos***
Comparative	`C`	***Comp***
Superlative	`S`	***Sup***
With suffix -e	`E`	***Suff_e***
Infinitive	`i`	***Inf***
Participle	`p`	***Part***
Present tense	`t`	***Pres***
Past tense	`v`	***Past***
1st person verb	`1`	***Sin1***
2nd person verb	`2`	***Sin2***
Inversed	`I`	***Inv***
3rd person verb	`3`	***Sin3***
Imperative only	`g`	***Imp***
Subjunctive only	`a`	***Sub***
Headword form (not nouns, verbs or adjectives)	`X`	

Table 10: Type of flection labels

For a full definition of these flection types, read the details given for the appropriate 'yes/no' columns in section 3.4.1 above. However, note that there is one type of flection label which does not correspond to a 'yes/no' column. The `X` label identifies many forms not covered by the other labels, including adverbs like *eerlang*, numerals like *tachtigduizend* or conjunctions like *waar*. These forms are always the same as those used as the headword form of the lemma (thus the very few inflected adverbial forms do not get the code `X`). No nouns, verbs or adjectives ever get the code `X`.

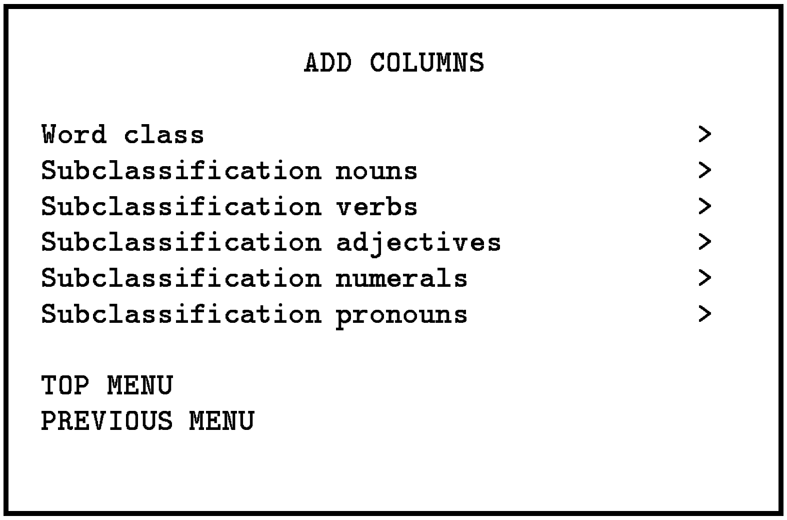
Each wordform may have more than one code attached to it. Thus the wordform *geabonneerde* has the code `pvE`: `p` means it is a participle, `v` means that it is a past tense form, and `E` means that it has an inflectional -*e* suffix.

The FLEX name and description of this column are as follows:

FlectType `Type of flection`

4 DUTCH SYNTAX

Syntactic information is available for lemma lexicons. It consists of syntactic codes which describe all the lemmas in the database. A general word class code is available, as well as more detailed codes on nouns, verbs, adjectives, numerals and pronouns. Diagram 6 in Appendix I 'Syntax of Dutch Lemmas' gives an overview of the syntactic information offered to you in the `ADD COLUMNS` menus:



```
                  ADD COLUMNS

Word class                                  >
Subclassification nouns                     >
Subclassification verbs                     >
Subclassification adjectives                >
Subclassification numerals                  >
Subclassification pronouns                  >

TOP MENU
PREVIOUS MENU
```

If you want to use syntactic information of this type in conjunction with a wordforms lexicon (perhaps you want to know the word class of your wordforms), then you should use the 'lemma information' columns available with the morphological columns for wordforms. Since the syntactic category of a wordform is always the same as the lemma it belongs to, there is no need to provide extra, unnecessary syntactic columns for wordforms. The special link with lemma information means you can get access to all sorts of general information about the lemmas which represent each wordform.

However on occasions there are wordforms whose categorizations are different from those given for their lemma. Diminutive forms are always neuter ('het') words, whereas the normal form of the word may not be – thus the gender code for *tafeltje* indicates that it is a masculine ('de') word since its lemma is *de tafel*. Likewise although the infinitive form of a verb can be used as a noun ('*vloeken* is aangeleerd') it is always classified as a verb. Such differences are specific to certain wordforms, and because they usually work according to well-known rules, the details need not be given in the database.

4.0.1 SYNTACTIC CODES: LETTERS OR NUMBERS

For most of the classifications described below, there are two ways of representing each syntactic code. You can choose whether to use numbers (`Numeric codes`) or shortened verbal codes (`Labels`). An adverb, for example, is represented by the number `7` or the letters `ADV`. No matter which type of codes you decide to use, the *information* remains the same; only the *representation* changes.

Numeric codes use single digits to represent syntactic subclassifications. If ever you see a lemma with more than one digit, it means that more than one of the syntactic categories can apply to it. Thus the verb *dagen*, for example, has the subclassification code 14: the 1 means 'this can be a lexical verb', and the 4 means 'this can be an impersonal verb'. With the exception of the general word class column, the number 0 (that is, zero) means that the correct subcategorization is unknown, and a null value (that is, no value at all) means that the particular subcategorization is not appropriate for the lemma in question.

Labels are made up of letters or short abbreviations. When a lemma fits more than one subcategory, the appropriate labels are separated by a / character. Thus the verb *dagen* is given the subclassification label `zelfst./onpers` This means that the lemma can be a lexical verb or an impersonal verb. With the exception of the general word class column, a question mark (`?`) means that the correct subcategorization is not known, and a null value means that the particular subcategorization is not appropriate for the lemma in question.

4.1 WORD CLASS

The word class code is a simple way to identify the syntactic class of every lemma in the database. Ten basic categories—set out in Table 11 below—are distinguished, and you can identify them using either of the two forms described in section 4.0.1 above. Note that there are no null values in these columns: one of the categories listed is applied to every lemma.

Only the first of the ten categories *Expression* may need some explanation. This refers to words which are *only* used in combination with certain other words to make up a particular phrase, and which cannot be properly classified with the other word class codes. An example is the word *voorschijn*, which is only ever part of the expression *te voorschijn*.

The definitions of the two word class columns are given below, followed by Table 11 which sets out the meaning of each code with examples. If you want syntactic codes in the form of numbers, choose the column with this FLEX name and description:

ClassNum (ClassNumLemma) `Word class, numeric`

If you want syntactic codes in the form of short verbal symbols, choose the column with this FLEX name and description:

Class (ClassLemma) `Word class, labels`

Word Class	Columns		Example
	ClassNum	***Class***	
Expression	0	EXP	*voorschijn*
Noun	1	N	*bank*
Adjective	2	A	*groot*
Quantifier/Numeral	3	NUM	*meer, zes*
Verb	4	V	*verlaat*
Article	5	ART	*het*
Pronoun	6	PRON	*ik*
Adverb	7	ADV	*fijntjes*
Preposition	8	PREP	*over*
Conjunction	9	C	*maar*
Interjection	10	I	*asjemenou*

Table 11: Word class codes

4.2 SUBCLASSIFICATION NOUNS

One important distinction between nouns in Dutch is *gender*. Using the information described here, you can find out either the full gender of any noun, or the simple gender (the simple gender is a straight choice between *de* and *het*). In addition, *proper nouns* (names of various sorts) are further subclassified.

4.2.1 NOUNS: FULL GENDER

There are three basic genders in Dutch: *male, female,* and *neutral.* In addition to these three, CELEX also identifies those female nouns which can be treated as male, and nouns whose gender depends on the context in which they are used. This makes five basic 'genders', which are represented by a set of numeric codes and a set of labels (as described in section 4.0.1 above). Table 12 below gives the meanings represented by both sets of codes along with some examples:

Gender	Columns		Example
	GendNum	***Gend***	
male	1	`m.`	*wijn*
female	2	`v.`	*ruimte*
neutral	3	`o.`	*vertrek*
female (male)	4	`v.(m.)`	*bierhal*
context	5	`m.-v.`	*gelovige*

Table 12: Nouns: full gender codes

Sometimes these codes are used in combination, to show that the gender information is more complicated. The word *kameelzadel*, for example, has the codes 13 and *m. of o.* to indicate that it is male or neutral. This convention is often used in dictionaries (like the *Van Dale Groot Woordenboek der Nederlandse Taal* or the *Woordenlijst van de Nederlandse taal*). No preference is implied with this type of combination; either gender is equally valid.

Sometimes two codes can be given for a lemma, one of which is the normal, preferred usage, while the other is an acceptable but secondary usage. When this happens, no numeric code is given for the second usage. A verbal label does include the second form by using the word `en` after the first code in the combination. For instance, the word *bezwaarplicht* has the label *m. en v.* which means 'this is a male word, though it may be treated as a female word'. In contrast, the corresponding numeric code is simply `1`. So, remember that `en` implies that the first gender code is preferred to the second, while in contrast `of` implies that both codes are equally acceptable. The verbal labels thus provide slightly more information than the numeric codes, because when a numeric code has more than one number, either of the codes is equally valid; preferences are never implied by numeric codes.

The FLEX names and descriptions of these two gender code columns are as follows:

GendNum (GendNumLemma) `For nouns: gender, numeric`

Gend (GendLemma) `For nouns: gender, labels`

4.2.2 DE/HET DISTINCTION

Increasingly, the male/female distinction in Dutch appears to be fading away. The fundamental distinction is now de/het, where *de* means 'not neutral' and *het* means 'neutral'. There are two columns available which indicate whether your selected noun is a 'de-word' or a 'het-word', or whether it can be either. As described in section 4.0.1, you can choose numeric codes or letter codes, according to your requirements.

de/het?	Columns		Example
	DeHetNum	***DeHet***	
De (non-neutral)	`1`	`de`	*deur*
Het (neutral)	`2`	`het`	*schip*
De or het	`12`	`de/het`	*jolijt*

Table 13: De/het distinction codes

The FLEX names and descriptions of these two de/het distinction code columns are as follows:

DeHetNum
(DeHetNumLemma) `For nouns: de/het distinction, numeric`

DeHet
(DeHetLemma) `For nouns: de/het distinction, labels`

4.2.3 PROPER NOUNS

A proper noun is a name of some kind, and usually begins with a capital letter. CELEX distinguishes four types of proper nouns, and Table 14 defines these four types and gives examples:

Proper Noun	Columns		Example
	PropNum	***Num***	
Geographical names	`1`	`geo.`	*Belfast*
Names of people	`2`	`pers.`	*Wilma*
Company or brand names	`3`	`merk.`	*Droste*
Other	`4`	`over.`	*Teleac*

Table 14: Proper noun codes

The last code `over.` covers all sorts of things – examples include *Interpol*, the (non-commercial) international police organization, and *Basic*, the popular Beginners' All-Purpose Instruction Code.

Sometimes more than one code is appropriate. *Stein*, for example, is the name of a town in Limburg, but it can also be a first name. Thus the correct code in numeric form is `12`, or in label form `geo./pers.`

The two columns available with information on proper nouns contain codes in numeric forms or as labels (as described in section 4.0.1), and their FLEX names and descriptions are as follows:

PropNum
(PropNumLemma) `For nouns: proper noun, numeric`

Prop
(PropLemma) `For nouns: proper noun, labels`

4.3 SUBCLASSIFICATION VERBS

When the simple word class code isn't detailed enough, further information on verbs is available here. You can find out which verbs take *hebben* as their auxiliary verb, which take *zijn*, and which can take either *hebben* or *zijn*. In addition, different types of verbs are distinguished and coded – copulas, impersonal verbs, and ordinary lexical verbs, for example. Lexical verbs are further subclassified into transitive, intransitive, and reflexive. As with all the syntactic information, both numeric codes and verbal labels (see section 4.0.1) are provided for each subclassification.

4.3.1 PERFECT TENSE (HEBBEN/ZIJN)

When the perfect tense occurs in Dutch, one of two *auxiliary* verbs is linked with a main verb. (In the sentence *ik heb gerikkekikt*, for example, the main verb *rikkekikken* is supported by the auxiliary verb *hebben*.) To find out whether the verb you have selected takes *hebben* or *zijn* in the perfect tense, include in your lexicon one of the columns described here. Table 15 below sets out the simple codes used in the two columns available. When either *hebben* or *zijn* can be used in conjunction with a particular verb, the codes for each auxiliary are combined to make a two-digit code.

Auxiliary	Columns		Example
	AuxNum	***Aux***	
hebben	1	`hebben`	*doen*
zijn	2	`zijn`	*groeien*
hebben or zijn	12	`hebben/zijn`	*volgen*

Table 15: Perfect tense auxiliary verb codes

The FLEX names and descriptions of these two columns are as follows:

AuxNum ***(AuxNumLemma)*** `For verbs, auxiliary verb, numeric`

Aux ***(AuxLemma)*** `For verbs, auxiliary verb, labels`

4.3.2 SUBCLASSES

To distinguish further between all the verbs in the database, four subclassification codes are given in the two columns described here. The first category is *lexical verb*, which is a normal 'content word' verb; it is used in a sentence primarily for the meaning it conveys, rather than fulfilling a purely grammatical or structural role. The second category, *auxiliary verb*, is the opposite: it is used in a sentence to indicate *tense* rather than the *action* implied by the main lexical verb. The third category, *copula*, is also a function word: it usually links a subject to a complement. An example is the sentence 'ben jij dat?', where the copula verb *zijn* links the subject *jij* to a complement *dat*. The fourth category, *impersonal verbs*, refers to those verbs which cannot have a referential subject; *het regent*, for example.

Often, a particular verb may get more than one code: the verb *regenen* is classified as a lexical verb *and* an impersonal verb, and thus has the numeric code `14` and the label `zelfst./onpers.` Other verbs may require a different combination of the four basic codes.

Table 16 sets out the basic codes used, and after that, the FLEX names and descriptions for the two columns are given.

Subclass	Columns		Example
	SubClassVNum	***SubClassV***	
Lexical verb	1	`zelfst.`	*afwassen*
Auxiliary verb	2	`hulp.`	*hebben*
Copula	3	`koppel.`	*lijken*
Impersonal verb	4	`onpers.`	*regenen*

Table 16: Verb subclass codes

SubClassVNum* *(SubClassVNumLemma) `For verbs, subclasses, numeric`

SubClassV* *(SubClassVLemma) `For verbs, subclasses, labels`

4.3.3 SUBCATEGORIZATION LEXICAL VERBS

Lexical verbs—those verbs which are used because of the meaning they convey rather than the grammatical function they perform—can be further subcategorized into three different groups. Those groups are *transitive* (the verb takes a direct object), *intransitive* (the verb does not take a direct object), and *reflexive* (the verb can or must be used along with a reflexive pronoun, so that the pronoun and the subject of a sentence refer to the same entity, e.g. 'soms voel ik mij helemaal niet').

Two sorts of codes are available, either numeric codes or labels (as described in section 4.0.1 above).

As with most of the syntactic codes, combinations are sometimes possible. The stem *stoot*, for example, is a verb which can be transitive, intransitive, and reflexive. Thus it has the numeric code `123`, and the label `intrans./trans./wederk.`

Lexical verb subcategorization	Columns		Example
	SubCatNum	***SubCat***	
Intransitive	`1`	`intrans.`	*vallen*
Transitive	`2`	`trans.`	*kopen*
Reflexive	`3`	`wederk.`	*vergissen*

Table 17: Lexical verb subcategorization codes

The FLEX names and descriptions of these two columns are as follows:

SubCatNum (SubCatNumLemma) `For verbs, subcategorization, numeric`

SubCat (SubCatLemma) `For verbs, subcategorization, labels`

4.4 SUBCLASSIFICATION ADJECTIVES

Although most adjectives in Dutch can also function as adverbs, some do not. There are two columns available which indicate by means of numeric codes or labels (see section 4.0.1 above) whether or not the adjective you select can be an adverb.

If the adjective can also be used as an adverb (like the word *goed*, for example), the numeric code is `2`, and the label is `adv`. If the adjective cannot be an adverb (like the word *zwartzijden*), the numeric code is `1`, and the label is `nonadv`.

The FLEX names and descriptions of these two columns are as follows:

AdvNum* *(AdvNumLemma) `For adjectives, adverbial usage, numeric`

Adv* *(AdvLemma) `For adjectives, adverbial usage, labels`

4.5 SUBCLASSIFICATION NUMERALS

The general term *numerals* covers quantifiers (such as *meer* or *veel*) and also words which relate directly to numeric values. These 'numeric-value words' can be subdivided into *cardinal* numerals (for example *zeventien* or *vijfduizendzevenhonderddrieennegentig*), and *ordinal* numerals (for example *zeventiende* or *vijfduizendzevenhonderdzevenennegentigste*).

The two columns defined here let you distinguish between cardinal and ordinal numerals by means of numeric codes and labels (as described in section 4.0.1 above). If the 'numeric-value word' represents a cardinal numeral, the numeric code is `1`, and the label `hoofd`. If the 'numeric-word value' represents an ordinal numeral, the numeric code is `2`, and the label is `rang`.

The FLEX names and descriptions of these two columns are as follows:

CardOrdNum* *(CardOrdNumLemma) `For numerals, cardinal/ordinal, numeric`

CardOrd* *(CardOrdLemma) `For numerals, cardinal/ordinal, labels`

4.6 SUBCLASSIFICATION PRONOUNS

There are over one hundred and thirty pronouns given in the N3.1 database, and most of them can be sub-classified in accordance with the codes given in Table 18 (below). The usual numeric codes and labels are available.

Whenever more than one code applies to a particular pronoun, multiple codes are given. For example, the word *wie* can be a relative pronoun, an interrogative pronoun, and an indefinite pronoun. The correct numeric code is therefore 458. The correct labels are formed by joining up all the appropriate labels, and separating them with a slash, so that the label is **betr./vraag./onbep.**

Pronoun subclass	Columns		Example
	SubClassPNum	***SubClassP***	
Personal	1	`pers.`	*jij*
Demonstrative	2	`aanw.`	*deze*
Possessive	3	`bez.`	*ons*
Relative	4	`betr.`	*die*
Interrogative	5	`vraag.`	*welk*
Reflexive	6	`wknd.`	*zich*
Reciprocal	7	`wkg.`	*elkaar*
Indefinite	8	`onbep.`	*weinig*
Exclamatory	9	`uitr.`	*wat*

Table 18: Pronoun subclassification codes

The FLEX names and descriptions of these two columns are as follows:

SubClassPNum (SubClassPNumLemma) For pronouns, subclasses, numeric

SubClassP (SubClassPLemma) For pronouns, subclasses, labels

5 DUTCH FREQUENCY

Frequency information (that is, details of how often words occur in Dutch) is available both for lemmas and wordforms. It is taken from the INL corpus. Work on the INL corpus is being carried out at the *Instituut voor Nederlandse Lexicologie* in Leiden, and at the time information was extracted from it for CELEX, it contained 42,380,000 words, all taken from written sources of every kind.

To understand the figures given in the various frequency columns, it's worth pausing to consider the way in which the frequency information taken from the INL corpus has been processed. The very first stage is to find out how many times each word occurs in the corpus, an easy task for a computer (it takes a week or so to read through the 42-million word corpus). The resulting figures are raw 'string' counts – that is, they indicate how many times each separate group of letters occurs in the corpus, taking no account of the different meanings that can be applied to each group. Types which only occur once, or which only occur in one text are ignored. You can see the remaining raw counts in an INL corpus types lexicon when you select the ***Freq*** column. The string *naaldbossen* for example occurs 11 times, and *aal* occurs 100 times. To develop this basic string count into a more helpful word count, the strings must be identified either as wordforms which can be linked to a particular lemma, or as other things not represented in the database, such as personal names, foreign words, and words which the INL's Optical Character Recognition machine misread (for example, the letters *ri* were occasionally mistaken for the single character *n*).

Sometimes this identification process is straightforward – the string *naaldbossen*, which occurs 11 times in the corpus, is always the plural wordform of the noun lemma *naaldbos*. So in this case the raw string frequency of the string *naaldbossen* is also the frequency of the wordform *naaldbossen*, and so in the wordform lexicon ***inl*** column it is also given as 11.

Once you know the frequencies of the wordforms associated with a particular lemma, working out a frequency figure for the lemma as a whole is straightforward – all you have to do is add up the appropriate wordform frequencies. In this way the frequency of the noun lemma *naaldbos* is 11 (the frequency of the wordform *naaldbossen*) plus 14 (the frequency of the wordform *naaldbos*). So the frequency of the lemma *naalbos* is thus 25, and this is the figure given in the lemma lexicon ***inl*** column.

On other occasions, though, the story is more complicated. Not all words can be assigned to just one lemma, and the string *aal*, which occurs 100 times in the corpus, is an example. This string can be linked to three lemmas which occur in the database: one means *eel* (the fish), another means *beer* (the alcoholic drink), and the last means *liquid manure*. It can also be linked to the surname *Aal* or to a dialect word, neither of which occur in the database.

The only way to sort out the individual frequencies of each of these strings is to look at the way they are used in the corpus, a process known as *disambiguation*. It's possible to carry out this task quickly by computer program, but at present the results of such programs can never be wholly accurate. For this reason, CELEX chose to disambiguate by hand, which means that someone reads each occurrence of each ambiguous form in the corpus, and notes the lemma to which it belongs. While such an approach is both costly and time-consuming, it does produce results which are more dependable and accurate. For *aal*, it seems that 63 of the occurrences mean *eel*, and 2 mean *liquid manure*, which means that there were no occurrences of *aal* meaning *beer* in the INL 42 million word corpus. These are the three figures given in the wordform lexicon ***inl*** column for the three different *aal* wordforms. Of the remaining 35 occurrences of the string *aal*, 28 were identified as being the surname, 2 couldn't be disambiguated, and the rest were typing errors. This information is not given in the database since it doesn't relate directly to any of the lemmas or wordforms available.

Again, once the wordform frequencies have been clarified, working out the lemma frequencies is straightforward. For the three lemmas with the form *aal*, the lemma frequencies are 73 (meaning *eel*), 2 (meaning *liquid manure*), and 0 (meaning *beer*). The figure 73 is arrived at after you take into account the frequency of the other wordform associated with the lemma meaning *eel* – its plural form *alen*, which occurs 8 times in the corpus. These lemma frequency figures are given in the lemma lexicon ***inl*** column, and in the same column to be found with the 'lemma information' given for wordforms.

When strings occur very frequently in the corpus, the work required to disambiguate each case by hand can be daunting. It may also be unnecessary, since an intelligent estimate coupled with an indication of how far that estimate is accurate should usually be enough. So, whenever ambiguous words occur more than 100 times in the corpus, not all the occurrences are checked individually. Instead, one hundred occurrences of the string are taken at random from the corpus and then analysed. In this way it's possible to formulate a ratio which indicates the proportions of the various interpretations, and this ratio can then be applied to the real figures to see an estimate of how the fully disambiguated figures would look.

As an example, take the string *roer*. Its basic corpus string frequency is 511. It can be a verb, either the fist or second person singular form, meaning *stir*, or a noun meaning *rudder*, or a part of the expression *in rep en roer*. Here is a lexicon which shows these wordforms with their word class and frequency :

Word	**Class**	**INL**
roer	EXP	93
roer	N	382
roer	V	5
roer	V	5

To calculate these figures, a 100 occurrences of the string *roer* were taken from the corpus and disambiguated by hand. It turned out that 2 of the occurrences belonged to the verbal lemma, 18 to the expression lemma, and 74 to the noun lemma. 5 were identified as a personal name. The remaining 1 could not be disambiguated, meaning that effectively the sample consisted of 99 occurrences, not 100. So to estimate the real frequency of the wordform belonging to the expression lemma, divide the number of times it occurred in the sample by the total number of successfully disambiguated forms, and then multiply the result by the original string frequency: $\frac{18}{99} \times 511 = 93$. Repeating this procedure gives 382 for the noun lemma, and 10 for the verbal lemma. This latter figure is divided equally for the two possible wordforms, 5 for the first person singular form, and 5 for the second person singular form. This is the usual way of sorting out ambiguous verbal flections, since disambiguating every verbal form by hand is a task which would involve a great deal of work yielding results of interest to only a few.

For most items in the database, the frequency figures are accurate. However, when estimates have to be made on the basis of a hundred examples, then deviation figures have to be calculated, to let you see just how accurate the estimates are. This formula gives the required deviation figure:

$$N \times 1.96 \times \sqrt{\frac{p\,(1-p)}{n} \times \frac{N-n}{N-1}}$$

where N is the frequency of the string as a whole, n is the number of items which could be disambiguated in the random 100-item sample, and p is the ratio figure for the item when it belongs to one particular lemma. Thus for the noun wordform *roer*, N is 511, n is 99, and p is 0.7474. The formula gives 39 as the deviation. This means that the true frequency for this form of *roer* is almost certain—at least 95% certain—to lie between 472 and 550.

This deviation information is in the ***InlDev*** columns, available for both wordforms and lemmas. You can include the column alongside the frequency figures themselves to see how accurate they are:

Word	ClassLemma	INL	INLDev
roer	EX	93	35
roer	N	382	39
roer	V	5	25
roer	V	5	25

Whenever the deviation is greater than the frequency itself, then you know for sure that some sort of arbitrary approximation has been carried out. This happened for the verbal forms of *roer*, as you can see in the table above.

Working out deviation figures for a lemma involves adding together the frequencies of its disambiguated wordforms. And once again, whenever the resulting deviation figure is equal to or greater than the frequency itself, you know that some arbitrary 'disambiguation' has been necessary.

One final point to note here is that some frequency information is available with the orthographic columns. This relates directly to the different spellings that wordforms, headwords, or stems can have. It does not affect the frequency information given here, which deals with each form as a whole regardless of how it can be spelt. For instance, *actief* can also be spelt *aktief*, and the lemma frequency given alongside each of them is the same: 2622. The spelling frequency on the other hand shows that the spelling *actief* occurs 2201 times while the spelling *aktief* occurs 421. For more details about this extra layer of disambiguation, read the appropriate subsection under 'Dutch Orthography'.

5.1 FREQUENCY INFORMATION FOR LEMMAS, WORDFORMS AND ABBREVIATIONS

Now that the background details have been explained, the individual column names and descriptions can be formally defined. For lemmas, wordforms and abbreviations, there are four columns available which express the INL frequency figures in various ways. For lemma lexicons there is an additional column which indicates whether a particular lemma is represented in one of the paper dictionaries used when CELEX began building the database.

The first column gives the plain INL frequency count for each lemma or wordform. The figure given in the lemma version of the column for *componist* is 598, which means that out of the 42,380,000 words in the corpus, 598 are the word *componist* in some form or other. The figures given in the wordform version of this column reveal how frequently each of the possible forms occur: for *componist* the figure is 514, and for *componisten* it is 84. The FLEX name and description of this column are as follows:

INL
(INLLemma) `INL frequency`

The second column indicates how accurate the frequencies in the previous column are by providing a deviation figure for each lemma or wordform, calculated according to the methods described in the previous section. If a word has been fully disambiguated without the need for any estimates, the figure is 0. When some estimation has been required, the figure will be greater than zero. If the figure should ever be equal to or greater than the frequency it qualifies, then you know that full disambiguation was not possible. The figure given for the lemma *objektief* is 26, and when you use it in conjunction with the INL frequency figure of 1494, it indicates that you can be almost certain (95% certain) that *objektief* occurs in one form or another somewhere between 1468 and 1520 times. The FLEX name and description of this column are as follows:

InlDev
(InlDevLemma) `INL frequency deviation`

The next column contains the same frequency figures as the first column, except that they have been scaled down to a range of 1 to 1,000,000 instead of the usual 1 to 42,380,000. This is done by dividing the normal INL frequency for each word by the number of words in the whole corpus, and then multiplying the answer by 1,000,000. The end result is a set of figures which are probably easier to understand: it makes greater sense to say that the word *gangmaker* is one in a million than it does to say that it's 51 words out of 42,380,000. And since other well-known text corpora—such as the *London-Oslo-Bergen* (LOB) and *Brown* corpora of English—are also based on a count of one million, this scale provides the opportunity for interesting comparisons to be made. However as you might expect, some detail is lost in the scaling-down process: the words *gaafheid* and *geboortedatum*, which have the 42 million word lemma frequencies of 23 and 62 respectively, both share the same 1 million word frequency of 1.

InlMln
(InlMlnLemma) `INL frequency (1,000,000)`

For those whose work requires a further transformation of the figures (psycholinguists working with stimulus response times for example), a column containing logarithmic values is available. The effect of the logarithmic scale is to emphasize the importance of lower frequency words in a way that the usual linear scale does not. For example, the difference between two words, one of frequency 2 and the other of frequency 1, becomes much greater than the difference between two words of frequency 2002 and 2001. (For the first pair of words, the difference is 0.30103, while for the second pair the difference is a mere 0.000217.) This confirms mathematically what we know intuitively: because there are so many words with a low frequency, the differences between them are that much more significant. With a high frequency word, a difference of one or two isn't very significant.

The values given are the base 10 logarithms of each `INL frequency (1,000,000)` described above. In place of a scale from 1 to 1,000,000, then, the resulting logarithmic values in this column range from zero ($\log_{10}1$) to 6 ($\log_{10}1{,}000{,}000$). And when a word has a normal frequency of zero, the logarithmic value is also given as zero. This is mathematically inaccurate ($\log_x 0$ doesn't exist), but—at least in this context—relatively unimportant: any word with a logarithmic frequency of 0 occurs at the very most only 63 times in the full INL 42 million word corpus, and is consequently only of interest to those concerned with the more esoteric branches of lexicography. The thing to remember is that only words which have an INL 1,000,000 frequency value of two or more (or, if you prefer, only words which occur 64 or more times in the INL corpus) have a logarithmic value greater than zero.

InlLog `INL frequency, logarithmic`
(InlLogLemma)

The last column to be described is available only with lemma lexicons, or with the lemma information available for wordforms lexicons under the morphological information. It simply notes whether or not a particular lemma is given in any one of the dictionary sources used when the CELEX database was first built. If the value given is `Y`, then the lemma can be found in the *Van Dale Groot Woordenboek van Hedendaags Nederlands*, the *Woordenlijst van de Nederlandse Taal*, or Uit den Boogaart's *Woordfrequenties* book. If the value given is `N`, you know that although the word is not in any of these dictionary sources, it occurs frequently enough in the INL corpus to merit inclusion in the database.

Dict `Dictionary Source`
(DictLemma)

5.2 FREQUENCY INFORMATION FOR INL CORPUS TYPES

The frequency information given in INL corpus types lexicons consists of the raw string counts from which all the other frequency figures for lemmas, wordforms and abbreviations are derived. Also available are figures which indicate how many of the texts which make the corpus contain each type, as well as some basic classifications for types which are not to be found amongst the wordforms and abbreviations given in the CELEX database. If you are not already familiar with the terms *token* and *type*, then check the glossary and the first part of the manual, the *Introduction*, in the section 'Lexicon types'.

The first column is the basic 'string' count which tells you how many times each type occurs in the INL corpus, which contains 42,380,000 tokens. For example, *de* occurs 2,323,977 times, *nederland* 8560 times, and *gatenkaas* 7 times. The FLEX name and description of this column are as follows;

Freq `Frequency`

The second column tells you in how many corpus texts each type occurs. The INL corpus is made up of 835 different texts, mostly recent books. For a type to be included in the CELEX list of corpus types, it must occur in more than one text. For example, *de* occurs in 835 different texts (in fact it occurs in every text in the corpus), *nederland* in 481, and *gatenkaas* in 6. The FLEX name and description of this column are as follows:

Disp `Dispersion`

The third column simply states whether or not a type is represented as a wordform (or an abbreviation) in the CELEX databases. The code `N` means 'no, the type is not indeterminate' – that is, it can be found in the main CELEX databases. The code `Y` means 'yes, the type *is* indeterminate', and it is not in the main CELEX databases. The types *de* and *nederland* both have the code `N`, while the type *gatenkaas* has the code `Y`. The FLEX name and description of this column are as follows:

Indet `Indeterminate`

The last column classifies all corpus types which occur in the type list but not in the main CELEX databases – or to put it another way, these codes classify all those corpus types which have the ***Indet*** code `Y`. This table sets out the codes and their meanings:

Category	Code	Example
Dutch	`D`	*gatenkaas*
Proper noun	`P`	*edinburgh*
Abbreviation	`A`	*cda*
Non-Dutch word	`N`	*seulement*
Erroneous word	`E`	*belangnjkheid*
Other	`O`	*verkitten*

Table 19: Status codes for indeterminate corpus types

When you use these codes, you should be aware that they are by no means completely consistent or accurate, because they were quickly classified without checking their original context in the corpus. The FLEX name and description of this column are as follows:

Status `Status`

1 TREE DIAGRAMS & COLUMN DESCRIPTIONS

This appendix is divided into sections corresponding to the lexicon types currently available. Each section begins with a set of tree diagrams which give you an overview of the columns you can choose when you select a particular type of lexicon, and then there are technical details about each of those columns – the type of the column, its minimum and maximum values and lengths, the number of null values it contains, and the characters used in each column. These details are particularly useful when you export a file from FLEX.

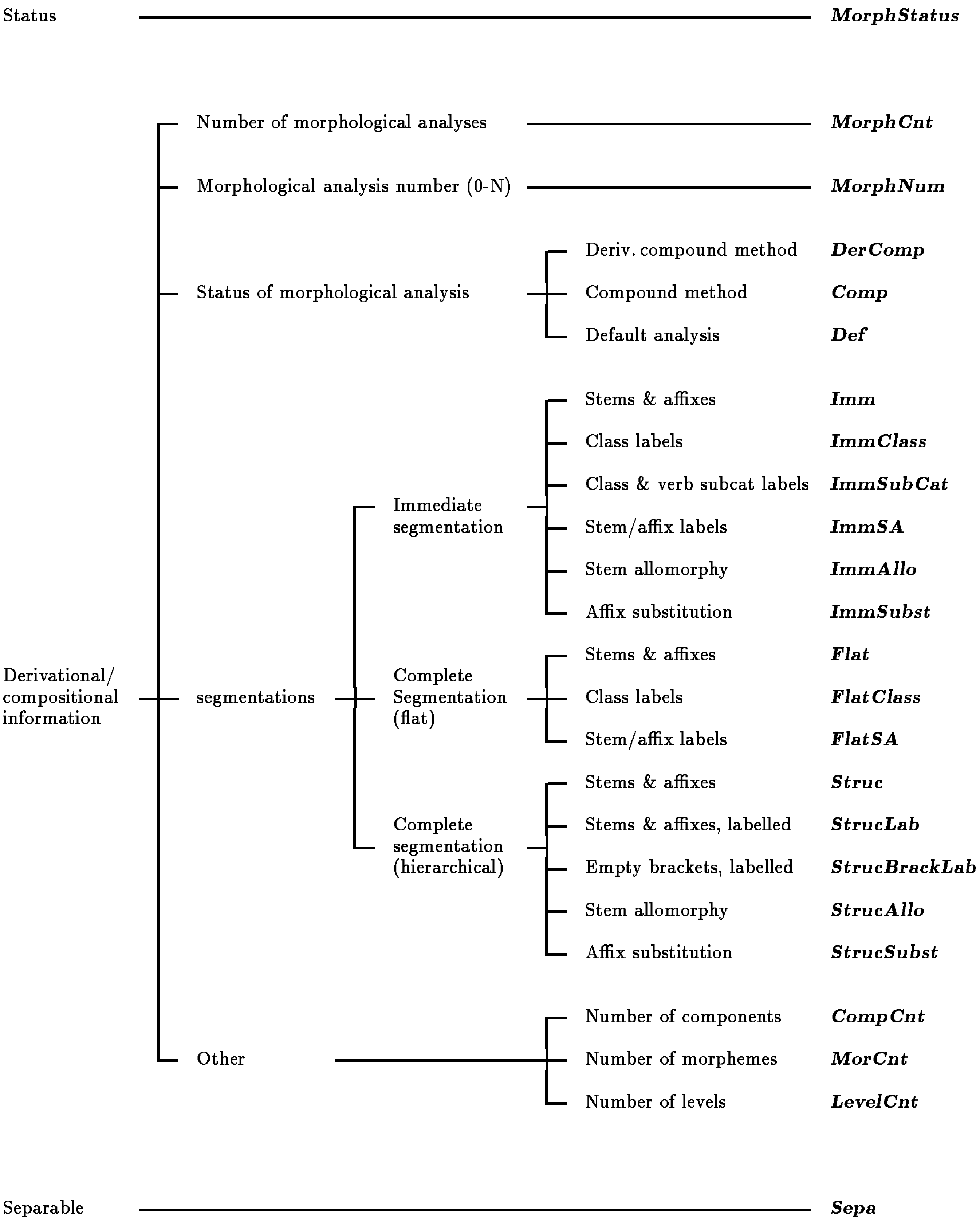
Whenever a new version of the database is released, the corresponding section in this appendix will also be replaced with the relevant diagrams and technical details. Always remember to check the name and lexicon number when you're using this appendix: you can see which lexicon type and version you are dealing with by reading the title of each diagram or the line at the top of each right-hand page.

1 ORTHOGRAPHY OF DUTCH LEMMAS (N31)

Number of spellings			***OrthoCnt***
Spelling number (1-N)			***OrthoNum***
Status of spelling			***OrthoStatus***
Frequency of spelling	INL frequency 42m		***InlSpellFreq***
	INL 95% confidence deviation 42m		***InlSpellDev***
Spelling	Headwords	Without diacritics	***Head***
		Without diacritics, reversed	***HeadRev***
		With diacritics	***HeadDia***
		Purely lowercase alphabetical	***HeadLow***
		Purely lowercase alphabetical, sorted	***HeadLowSort***
		Number of letters	***HeadCnt***
	Headwords syllabified	Without diacritics	***HeadSyl***
		With diacritics	***HeadSylDia***
		Number of syllables	***HeadSylCnt***
	Stems	Without diacritics	***Stem***
		Without diacritics, reversed	***StemRev***
		With diacritics	***StemDia***
		Number of letters	***StemCnt***
	Stems syllabified	Without diacritics	***StemSyl***
		With diacritics	***StemSylDia***
		Number of syllables	***StemSylCnt***
	Abstract stems	Without diacritics	***AbStem***
		With diacritics	***AbStemDia***
		Number of letters	***AbStemCnt***

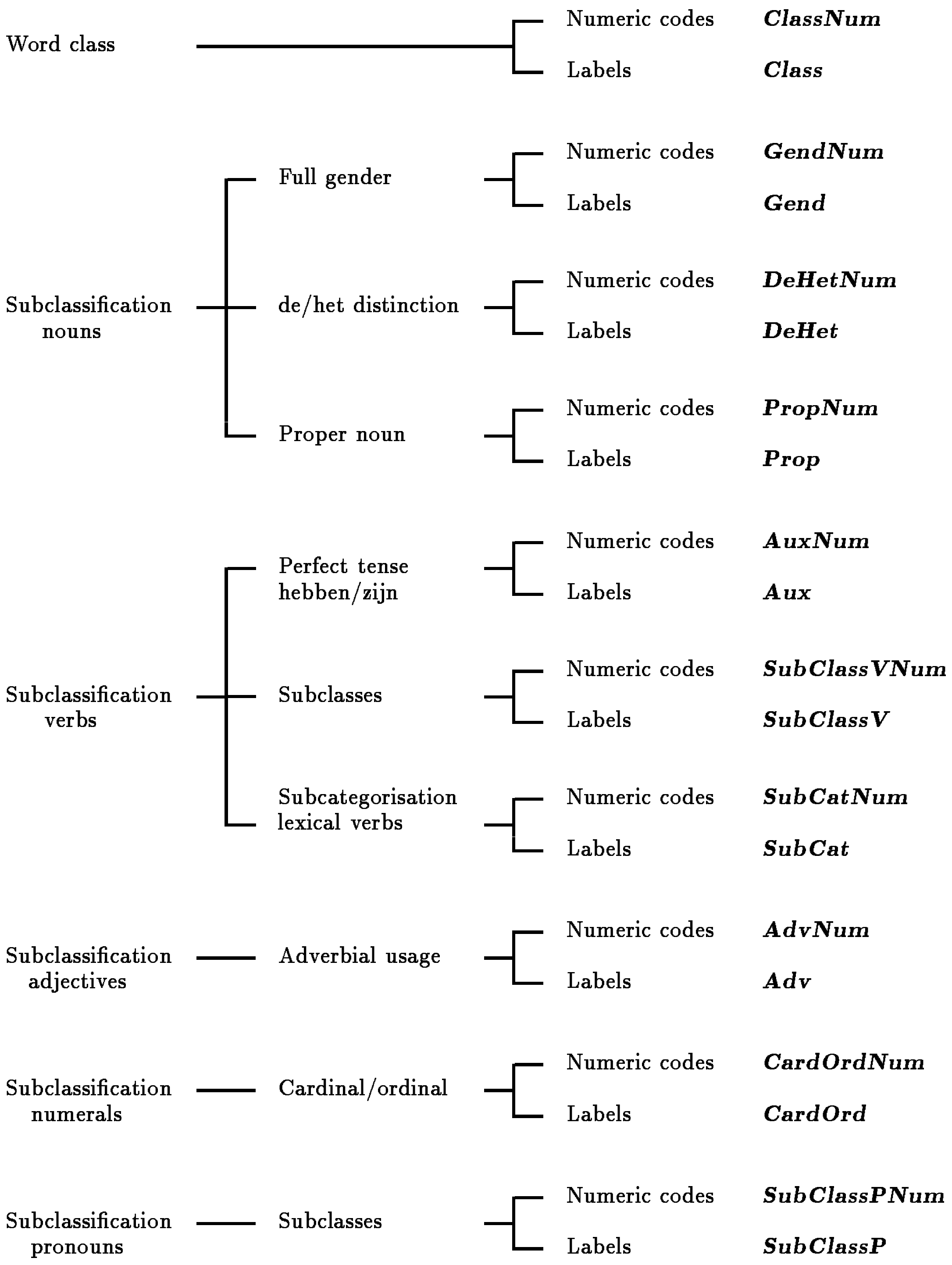
Phonetic transcriptions	Headwords plain	SAM-PA char set	***PhonSAM***
		CELEX char set	***PhonCLX***
		CPA char set	***PhonCPA***
		DISC char set	***PhonDISC***
		Number of phonemes	***PhonCnt***
	Headwords syllabified	SAM-PA char set	***PhonSylSAM***
		CELEX char set	***PhonSylCLX***
		CELEX char set, brackets	***PhonSylBCLX***
		CPA char set	***PhonSylCPA***
		DISC char set	***PhonSylDISC***
		Number of syllables	***SylCnt***
	Headwords syllabified with stress	SAM-PA char set	***PhonStrsSAM***
		CELEX char set	***PhonStrsCLX***
		CPA char set	***PhonStrsCPA***
		DISC char set	***PhonStrsDISC***
		Stress pattern	***StrsPat***
	Stems plain	SAM-PA char set	***PhonStSAM***
		CELEX char set	***PhonStCLX***
		CPA char set	***PhonStCPA***
		DISC char set	***PhonStDISC***
		Number of phonemes	***PhonStCnt***
	Stems syllabified	SAM-PA char set	***PhonSylStSAM***
		CELEX char set	***PhonSylStCLX***
		CELEX char set, brackets	***PhonSylStBCLX***
		CPA char set	***PhonSylStCPA***
		DISC char set	***PhonSylStDISC***
		Number of syllables	***StSylCnt***
	Stems syllabified with stress	SAM-PA char set	***PhonStrsStSAM***
		CELEX char set	***PhonStrsStCLX***
		CPA char set	***PhonStrsStCPA***
		DISC char set	***PhonStrsStDISC***
		Stress pattern	***StStrsPat***
Phonetic patterns	Headwords syllabified	CV pattern	***PhonCV***
		CV pattern, brackets	***PhonCVBr***
	Stems syllabified	CV pattern	***PhonStCV***
		CV pattern, brackets	***PhonStCVBr***
Phonological stem representations		CELEX char set	***PhonolCLX***
		CPA char set	***PhonolCPA***

3 MORPHOLOGY OF DUTCH LEMMAS (N31)



- Status — *MorphStatus*
- Derivational/ compositional information
  - Number of morphological analyses — *MorphCnt*
  - Morphological analysis number (0-N) — *MorphNum*
  - Status of morphological analysis
    - Deriv. compound method — *DerComp*
    - Compound method — *Comp*
    - Default analysis — *Def*
  - segmentations
    - Immediate segmentation
      - Stems & affixes — *Imm*
      - Class labels — *ImmClass*
      - Class & verb subcat labels — *ImmSubCat*
      - Stem/affix labels — *ImmSA*
      - Stem allomorphy — *ImmAllo*
      - Affix substitution — *ImmSubst*
    - Complete Segmentation (flat)
      - Stems & affixes — *Flat*
      - Class labels — *FlatClass*
      - Stem/affix labels — *FlatSA*
    - Complete segmentation (hierarchical)
      - Stems & affixes — *Struc*
      - Stems & affixes, labelled — *StrucLab*
      - Empty brackets, labelled — *StrucBrackLab*
      - Stem allomorphy — *StrucAllo*
      - Affix substitution — *StrucSubst*
  - Other
    - Number of components — *CompCnt*
    - Number of morphemes — *MorCnt*
    - Number of levels — *LevelCnt*
- Separable — *Sepa*

4 SYNTAX OF DUTCH LEMMAS (N31)



5 FREQUENCY OF DUTCH LEMMAS (N31)

INL frequency 42m	***Inl***
INL 95% confidence deviation 42m	***InlDev***
INL frequency 1m	***InlMln***
INL frequency, logarithmic	***InlLog***
Dictionary entry	***Dict***

AbStem Abstract stem

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	zymotisch	Maximum length:	34
Characters:	’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

AbStemCnt Abstract stem, number of letters

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	33	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

AbStemDia Abstract stem, diacritics

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	überhaupt	Maximum length:	34
Characters:	’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü		

Adv For adjectives: adverbial usage, labels

Type:	character	Null values:	122412
Minimum value:	?	Minimum length:	1
Maximum value:	nonadv	Maximum length:	6
Characters:	? a d n o v		

AdvNum For adjectives: adverbial usage, numeric

Type:	character	Null values:	122412
Minimum value:	0	Minimum length:	1
Maximum value:	2	Maximum length:	1
Characters:	0 1 2		

Aux For verbs: auxiliary verb, labels

Type:	character	Null values:	125232
Minimum value:	?	Minimum length:	1
Maximum value:	zijn	Maximum length:	11
Characters:	/ ? b e h i j n z		

AuxNum For verbs: auxiliary verb, numeric

Type:	character	Null values:	125232
Minimum value:	0	Minimum length:	1
Maximum value:	2	Maximum length:	2
Characters:	0 1 2		

CardOrd For numerals: cardinal/ordinal, labels

Type:	character	Null values:	137140
Minimum value:	?	Minimum length:	1
Maximum value:	rang	Maximum length:	5
Characters:	? a d f g h n o r		

CardOrdNum For numerals: cardinal/ordinal, numeric

Type:	character	Null values:	137140
Minimum value:	0	Minimum length:	1
Maximum value:	2	Maximum length:	1
Characters:	0 1 2		

Class Word class, labels

Type:	character	Null values:	0
Minimum value:	A	Minimum length:	1
Maximum value:	V	Maximum length:	4
Characters:	A C D E I M N O P R T U V X		

ClassNum Word class, numeric

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	10	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

Comp Compound analysis method

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

CompCnt Number of morphological components

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	6	Maximum length:	1
Characters:	0 1 2 3 4 5 6		

Def Default analysis

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

DeHet For nouns: "de/het" distinction, labels

Type:	character	Null values:	34524
Minimum value:	?	Minimum length:	1
Maximum value:	het	Maximum length:	6
Characters:	/ ? d e h t		

DeHetNum For nouns: "de/het" distinction, numeric

Type:	character	Null values:	34524
Minimum value:	0	Minimum length:	1
Maximum value:	2	Maximum length:	2
Characters:	0 1 2		

DerComp Derivational compound analysis method

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Dict Dictionary entry

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Flat Flat segmentation

Type:	character	Null values:	26269
Minimum value:	’m	Minimum length:	1
Maximum value:	zwoord	Maximum length:	44
Characters:	’ + - . A B C D E F G H I J K L M N O P R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

FlatClass Flat segmentation, word class labels

Type:	character	Null values:	26269
Minimum value:	A	Minimum length:	1
Maximum value:	xxxVx	Maximum length:	9
Characters:	A B C D E I N O P Q V X x		

FlatSA Flat segmentation, stem/affix labels

Type:	character	Null values:	26269
Minimum value:	AAAS	Minimum length:	1
Maximum value:	SSSSSA	Maximum length:	9
Characters:	A S		

Gend For nouns: gender, labels

Type:	character	Null values:	34524
Minimum value:	?	Minimum length:	1
Maximum value:	v.(m.) of o.	Maximum length:	14
Characters:	␣ ( ) - . ? e f m n o v		

GendNum For nouns: gender, numeric

Type:	character	Null values:	34524
Minimum value:	0	Minimum length:	1
Maximum value:	5	Maximum length:	2
Characters:	0 1 2 3 4 5		

Head Headword

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	zymotisch	Maximum length:	34
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

HeadCnt Headword, number of letters

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	33	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

HeadDia Headword, diacritics

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	überhaupt	Maximum length:	34
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü		

HeadLow Headword, lowercase, alphabetical

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	zymotisch	Maximum length:	33
Characters:	a b c d e f g h i j k l m n o p q r s t u v w x y z		

HeadLowSort Headword, lowercase, alphabetical, sorted

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	z	Maximum length:	33
Characters:	a b c d e f g h i j k l m n o p q r s t u v w x y z		

HeadRev Headword, reversed

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	zzibwohs	Maximum length:	34
Characters:	’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

HeadSyl Headword, syllabified

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	zy-mo-tisch	Maximum length:	44
Characters:	’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

HeadSylCnt Headword, number of orthographic syllables

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	11	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

HeadSylDia Headword, syllabified, diacritics

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	ü-ber-haupt	Maximum length:	44
Characters:	’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü		

IdNum Lemma number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	124136	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

Imm Immediate segmentation

Type:	character	Null values:	26269
Minimum value:	’m	Minimum length:	1
Maximum value:	zwoord	Maximum length:	36
Characters:	’ + - . A B C D E F G H I J K L M N O P R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

ImmAllo Stem allomorphy, top level

Type:	character	Null values:	26269
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

ImmClass Immediate segmentation, word class labels

Type:	character	Null values:	26269
Minimum value:	A	Minimum length:	1
Maximum value:	xxN	Maximum length:	6
Characters:	A B C D E I N O P Q V X x		

ImmSA Immediate segmentation, stem/affix labels

Type:	character	Null values:	26269
Minimum value:	AAS	Minimum length:	1
Maximum value:	SSSS	Maximum length:	6
Characters:	A S		

ImmSubcat Immediate segmentation, subcat labels

Type:	character	Null values:	26269
Minimum value:	0A	Minimum length:	1
Maximum value:	xxN	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 A B C D E I N O P Q V X x		

ImmSubst Affix substitution, top level

Type:	character	Null values:	26269
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Inl INL frequency

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	2344641	Maximum length:	7
Characters:	0 1 2 3 4 5 6 7 8 9		

InlDev INL frequency deviation

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	108718	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

InlLog INL frequency, logarithmic

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	4.7429	Maximum length:	6
Characters:	. 0 1 2 3 4 5 6 7 8 9		

InlMln INL frequency (1,000,000)

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	55324	Maximum length:	5
Characters:	0 1 2 3 4 5 6 7 8 9		

InlSpellDev INL spelling frequency deviation

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	108718	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

InlSpellFreq INL spelling frequency

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	2344641	Maximum length:	7
Characters:	0 1 2 3 4 5 6 7 8 9		

LevCnt Number of morphological levels

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	6	Maximum length:	1
Characters:	0 1 2 3 4 5 6		

MorCnt Number of morphemes

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	9	Maximum length:	1
Characters:	0 1 2 3 4 5 6 7 8 9		

MorphCnt Number of morphological analyses

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	8	Maximum length:	1
Characters:	0 1 2 3 4 6 8		

MorphNum Morphological analysis number

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	8	Maximum length:	1
Characters:	0 1 2 3 4 5 6 7 8		

MorphStatus Morphological status

Type:	character	Null values:	0
Minimum value:	C	Minimum length:	1
Maximum value:	U	Maximum length:	1
Characters:	C F I M U		

OrthoCnt Number of spellings

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 3 4		

OrthoNum Spelling number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 3 4		

OrthoStatus Status of spelling

Type:	character	Null values:	0
Minimum value:	I	Minimum length:	1
Maximum value:	P	Maximum length:	1
Characters:	I N P		

PhonCLX Phon. headword, CELEX charset

Type:	character	Null values:	2940
Minimum value:	&:.d.e:.m.	Minimum length:	2
Maximum value:	z.y:.r.z.u:.t.	Maximum length:	68
Characters:	& . : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonCnt Headword, number of phonemes

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	31	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

PhonCPA Phon. headword, CPA charset

Type:	character	Null values:	2940
Minimum value:	@.	Minimum length:	2
Maximum value:	z.y:.r.z.u:.t.	Maximum length:	68
Characters:	. / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonCV Headword, phon. CV pattern

Type:	character	Null values:	2940
Minimum value:	C	Minimum length:	1
Maximum value:	VVCCC-VVC-CVV-CVCC	Maximum length:	48
Characters:	- C V		

PhonCVBr Headword, phon. CV pattern, with brackets

Type:	character	Null values:	2940
Minimum value:	[CCCVCC]	Minimum length:	3
Maximum value:	[V][CVC][CCVC]	Maximum length:	59
Characters:	C V [ ]		

PhonDISC Phon. headword, DISC charset

Type:	character	Null values:	2940
Minimum value:	)d@kwisin@	Minimum length:	1
Maximum value:	}xt@nt	Maximum length:	31
Characters:	! ( ) * < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonolCLX Phonological deep structure, CELEX charset

Type:	character	Null values:	68472
Minimum value:	&:-ka:-lIpt	Minimum length:	1
Maximum value:	zy:r+IN	Maximum length:	38
Characters:	# & + - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonolCPA Phonological deep structure, CPA charset

Type:	character	Null values:	68472
Minimum value:	@m	Minimum length:	1
Maximum value:	zy:r+IN	Maximum length:	38
Characters:	# + . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonSAM `Phon. headword, SAM-PA charset`

Type:	`character`	Null values:	`2940`
Minimum value:	`/:.v.r.@.`	Minimum length:	`2`
Maximum value:	`}.x.t.@.n.t.`	Maximum length:	`68`
Characters:	`. / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }`		

PhonStCLX `Phon. stem, CELEX charset`

Type:	`character`	Null values:	`2940`
Minimum value:	`&:.d.e:.m.`	Minimum length:	`2`
Maximum value:	`z.y:.r.z.u:.t.`	Maximum length:	`68`
Characters:	`& . : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z`		

PhonStCnt `Stem, number of phonemes`

Type:	`numeric`	Null values:	`0`
Minimum value:	`0`	Minimum length:	`1`
Maximum value:	`31`	Maximum length:	`2`
Characters:	`0 1 2 3 4 5 6 7 8 9`		

PhonStCPA `Phon. stem, CPA charset`

Type:	`character`	Null values:	`2940`
Minimum value:	`@.`	Minimum length:	`2`
Maximum value:	`z.y:.r.z.u:.t.`	Maximum length:	`68`
Characters:	`. / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z`		

PhonStCV Stem, phon. CV pattern

Type:	character	Null values:	2940
Minimum value:	C	Minimum length:	1
Maximum value:	VVCCC-VVC-CVV-CVCC	Maximum length:	48
Characters:	- C V		

PhonStCVBr Stem, phon. CV pattern, with brackets

Type:	character	Null values:	2940
Minimum value:	[CCCVCC]	Minimum length:	3
Maximum value:	[V][CVC][CCVC]	Maximum length:	59
Characters:	C V [ ]		

PhonStDISC Phon. stem, DISC charset

Type:	character	Null values:	2940
Minimum value:	)d@kwisin@	Minimum length:	1
Maximum value:	}xt@nt	Maximum length:	31
Characters:	! ( ) * < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonStrsCLX Syll. phon. headword, with stress, CELEX charset

Type:	character	Null values:	2940
Minimum value:	&:-’de:m	Minimum length:	2
Maximum value:	zwa:rt-’fOr-m@x	Maximum length:	49
Characters:	& ’ - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonStrsCPA `Syll. phon. headword, with stress, CPA charset`

Type:	`character`	Null values:	2940
Minimum value:	`'@`	Minimum length:	2
Maximum value:	`zy/l.'va:r.d@x`	Maximum length:	49

Characters: `' . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z`

PhonStrsDISC `Syll. phon. headword, with stress, DISC charset`

Type:	`character`	Null values:	2940
Minimum value:	`')-d@-kwi-si-n@`	Minimum length:	2
Maximum value:	`}r-zy-'li-n@`	Maximum length:	43

Characters: `! ' ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }`

PhonStrsSAM `Syll. phon. headword, with stress, SAM-PA charset`

Type:	`character`	Null values:	2940
Minimum value:	`"/:-vr@`	Minimum length:	2
Maximum value:	`}r-zy:-"li:-n@`	Maximum length:	49

Characters: `" - / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }`

PhonStrsStCLX `Syll. phon. stem, with stress, CELEX charset`

Type:	`character`	Null values:	2940
Minimum value:	`&:-'de:m`	Minimum length:	2
Maximum value:	`zwa:rt-'fOr-m@x`	Maximum length:	49

Characters: `& ' - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z`

PhonStrsStCPA Syll. phon. stem, with stress, CPA charset

Type:	character	Null values:	2940
Minimum value:	'@	Minimum length:	2
Maximum value:	zy/l.'va:r.d@x	Maximum length:	49
Characters:	' . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonStrsStDISC Syll. phon. stem, with stress, DISC charset

Type:	character	Null values:	2940
Minimum value:	')-d@-kwi-si-n@	Minimum length:	2
Maximum value:	}r-zy-'li-n@	Maximum length:	43
Characters:	! ' ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonStrsStSAM Syll. phon. stem, with stress, SAM-PA charset

Type:	character	Null values:	2940
Minimum value:	"/:-vr@	Minimum length:	2
Maximum value:	}r-zy:-"li:-n@	Maximum length:	49
Characters:	" - / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonStSAM Phon. stem, SAM-PA charset

Type:	character	Null values:	2940
Minimum value:	/:.v.r.@.	Minimum length:	2
Maximum value:	}.x.t.@.n.t.	Maximum length:	68
Characters:	. / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSylBCLX Syll. phon. headword, CELEX charset (brackets)

Type:	character	Null values:	2940
Minimum value:	[&:][de:m]	Minimum length:	3
Maximum value:	[zy:r][zu:t]	Maximum length:	59
Characters:	& : @ A E G I N O S U Z [ ] a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylCLX Syll. phon. headword, CELEX charset

Type:	character	Null values:	2940
Minimum value:	&:-de:m	Minimum length:	1
Maximum value:	zy:r-zu:t	Maximum length:	47
Characters:	& - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylCPA Syll. phon. headword, CPA charset

Type:	character	Null values:	2940
Minimum value:	@	Minimum length:	1
Maximum value:	zy:r.zu:t	Maximum length:	47
Characters:	. / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonSylDISC Syll. phon. headword, DISC charset

Type:	character	Null values:	2940
Minimum value:	)-d@-kwi-si-n@	Minimum length:	1
Maximum value:	}x-t@nt	Maximum length:	41
Characters:	! ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSylSAM Syll. phon. headword, SAM-PA charset

Type:	character	Null values:	2940
Minimum value:	/:-vr@	Minimum length:	1
Maximum value:	}x-t@nt	Maximum length:	47
Characters:	- / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSylStBCLX Syll. phon. stem, CELEX charset (brackets)

Type:	character	Null values:	2940
Minimum value:	[&:][de:m]	Minimum length:	3
Maximum value:	[zy:r][zu:t]	Maximum length:	59
Characters:	& : @ A E G I N O S U Z [ ] a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylStCLX Syll. phon. stem, CELEX charset

Type:	character	Null values:	2940
Minimum value:	&:-de:m	Minimum length:	1
Maximum value:	zy:r-zu:t	Maximum length:	47
Characters:	& - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylStCPA Syll. phon. stem, CPA charset

Type:	character	Null values:	2940
Minimum value:	@	Minimum length:	1
Maximum value:	zy:r.zu:t	Maximum length:	47
Characters:	. / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonSylStDISC Syll. phon. stem, DISC charset

Type:	`character`	Null values:	2940
Minimum value:	`)-d@-kwi-si-n@`	Minimum length:	1
Maximum value:	`}x-t@nt`	Maximum length:	41

Characters: `! ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }`

PhonSylStSAM Syll. phon. stem, SAM-PA charset

Type:	`character`	Null values:	2940
Minimum value:	`/:-vr@`	Minimum length:	1
Maximum value:	`}x-t@nt`	Maximum length:	47

Characters: `- / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }`

Prop For nouns: proper noun, labels

Type:	`character`	Null values:	134389
Minimum value:	`geo.`	Minimum length:	4
Maximum value:	`pers.`	Maximum length:	10

Characters: `. / e g k m o p r s v`

PropNum For nouns: proper noun, numeric

Type:	`character`	Null values:	134389
Minimum value:	`1`	Minimum length:	1
Maximum value:	`4`	Maximum length:	2

Characters: `1 2 3 4`

Sepa Separable

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Stem Stem

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	zymotisch	Maximum length:	34
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

StemCnt Stem, number of letters

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	33	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

StemDia Stem, diacritics

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	überhaupt	Maximum length:	34
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü		

StemRev Stem, reversed

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	zzibwohs	Maximum length:	34
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

StemSyl Stem, syllabified

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	zy-mo-tisch	Maximum length:	44
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

StemSylCnt Stem, number of orthographic syllables

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	11	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

StemSylDia Stem, syllabified, diacritics

Type:	character	Null values:	0
Minimum value:	'm	Minimum length:	1
Maximum value:	ü-ber-haupt	Maximum length:	44
Characters:	' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü		

StrsPat Headword, stress pattern

Type:	character	Null values:	2940
Minimum value:	000000001	Minimum length:	1
Maximum value:	110000	Maximum length:	11
Characters:	0 1		

Struc Structured segmentation

Type:	character	Null values:	26269
Minimum value:	(’m)	Minimum length:	3
Maximum value:	(zwoord)	Maximum length:	72
Characters:	’ ( ) , - . A B C D E F G H I J K L M N O P R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

StrucAllo Stem allomorphy, any level

Type:	character	Null values:	26269
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

StrucBrackLab Structured segmentation, word class labels only

Type:	character	Null values:	26269
Minimum value:	(((((()[B],()[N])[N], ()[A|N.])[A], ()[N|A.])[N], ()[N|N.N], ()[N])[N]	Minimum length:	5
Maximum value:	()[V]	Maximum length:	110
Characters:	( ) , . A B C D E I N O P Q V X [ ] x |		

StrucLab `Structured segmentation, word class labels`

Type:	`character`	Null values:	26269
Minimum value:	`('m)[O]`	Minimum length:	6
Maximum value:	`(zwoord)[N]`	Maximum length:	139
Characters:	`' ( ) , - . A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ ] a b c d e f g h i j k l m n o p q r s t u v w x y z |`		

StrucSubst `Affix substitution, any level`

Type:	`character`	Null values:	26269
Minimum value:	`N`	Minimum length:	1
Maximum value:	`Y`	Maximum length:	1
Characters:	`N Y`		

StStrsPat `Stem, stress pattern`

Type:	`character`	Null values:	2940
Minimum value:	000000001	Minimum length:	1
Maximum value:	110000	Maximum length:	11
Characters:	0 1		

StSylCnt `Stem, number of phonetic syllables`

Type:	`numeric`	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	11	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

SubCat For verbs: subcategorisation, labels

Type:	character	Null values:	124855
Minimum value:	?	Minimum length:	1
Maximum value:	wederk.	Maximum length:	23
Characters:	. / ? a d e i k n r s t w		

SubCatNum For verbs: subcategorisation, numeric

Type:	character	Null values:	124855
Minimum value:	0	Minimum length:	1
Maximum value:	3	Maximum length:	3
Characters:	0 1 2 3		

SubClassP For pronouns: subclasses, labels

Type:	character	Null values:	137289
Minimum value:	?	Minimum length:	1
Maximum value:	wknd.	Maximum length:	19
Characters:	. / ? a b d e g i k n o p r s t u v w z		

SubClassPNum For pronouns: subclasses, numeric

Type:	character	Null values:	137289
Minimum value:	0	Minimum length:	1
Maximum value:	924	Maximum length:	3
Characters:	0 1 2 3 4 5 6 7 8 9		

SubClassV For verbs: subclasses, labels

Type:	character	Null values:	124855
Minimum value:	onpers.	Minimum length:	7
Maximum value:	zelfst./onpers.	Maximum length:	21
Characters:	. / e f h k l n o p r s t u z		

SubClassVNum For verbs: subclasses, numeric

Type:	character	Null values:	124855
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	3
Characters:	1 2 3 4		

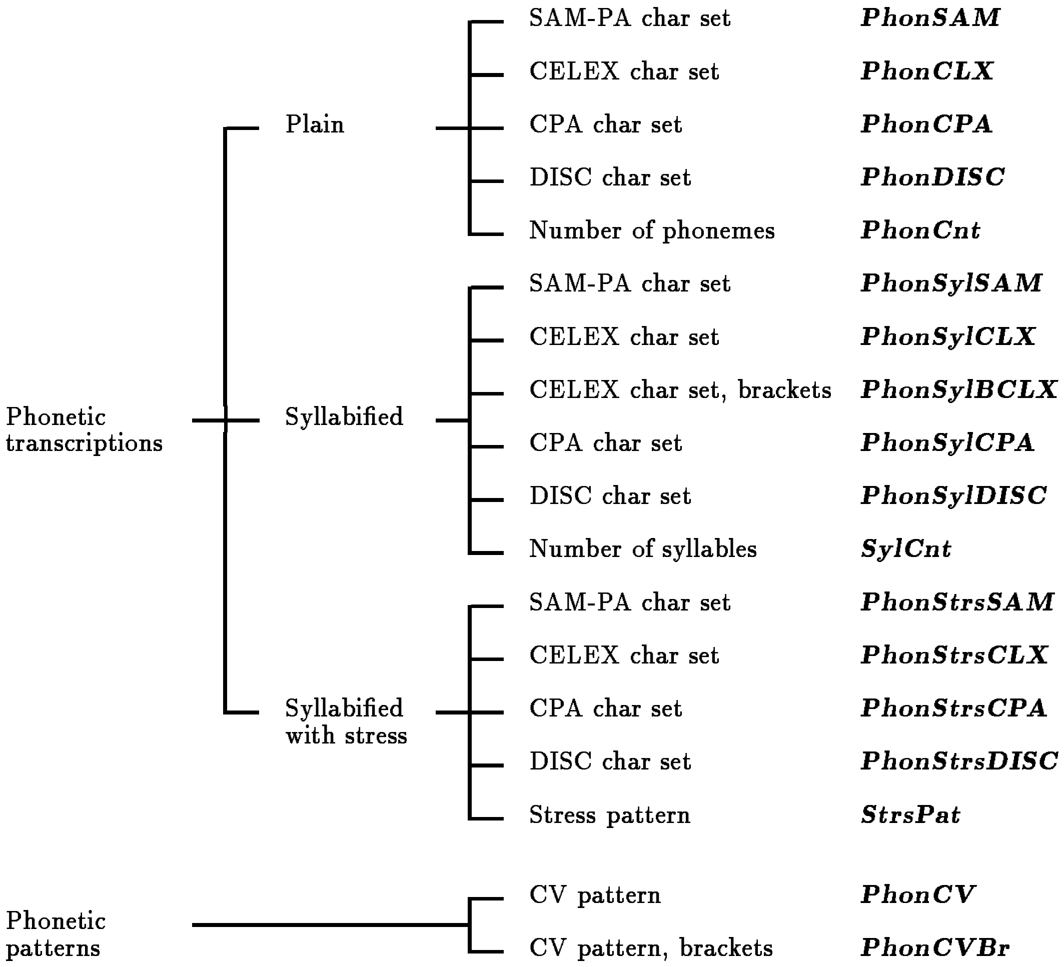
SylCnt Headword, number of phonetic syllables

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	11	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

6 ORTHOGRAPHY OF DUTCH WORDFORMS (N31)

Number of spellings			***OrthoCnt***
Spelling number (1-N)			***OrthoNum***
Status of spelling			***OrthoStatus***
Frequency of spelling	INL frequency 42m		***InlSpellFreq***
	INL 95% confidence deviation 42m		***InlSpellDev***
Spelling	Plain	Without diacritics	***Word***
		Without diacritics, reversed	***WordRev***
		With diacritics	***WordDia***
		Purely lowercase alphabetical	***WordLow***
		Purely lowercase alphabetical, sorted	***WordLowSort***
		Number of letters	***WordCnt***
	Syllabified	Without diacritics	***WordSyl***
		With diacritics	***WordSylDia***
		Number of syllables	***WordSylCnt***

7 PHONOLOGY OF DUTCH WORDFORMS (N31)



Phonetic transcriptions	Plain	SAM-PA char set	***PhonSAM***
		CELEX char set	***PhonCLX***
		CPA char set	***PhonCPA***
		DISC char set	***PhonDISC***
		Number of phonemes	***PhonCnt***
	Syllabified	SAM-PA char set	***PhonSylSAM***
		CELEX char set	***PhonSylCLX***
		CELEX char set, brackets	***PhonSylBCLX***
		CPA char set	***PhonSylCPA***
		DISC char set	***PhonSylDISC***
		Number of syllables	***SylCnt***
	Syllabified with stress	SAM-PA char set	***PhonStrsSAM***
		CELEX char set	***PhonStrsCLX***
		CPA char set	***PhonStrsCPA***
		DISC char set	***PhonStrsDISC***
		Stress pattern	***StrsPat***
Phonetic patterns		CV pattern	***PhonCV***
		CV pattern, brackets	***PhonCVBr***

8 MORPHOLOGY OF DUTCH WORDFORMS (N31)

Lemma information	Numeric id	***IDNumLemma***
	Orthography	**ORTHOGRAPHY OF DUTCH LEMMAS**
	Phonology	**PHONOLOGY OF DUTCH LEMMAS**
	Morphology	**MORPHOLOGY OF DUTCH LEMMAS**
	Syntax	**SYNTAX OF DUTCH LEMMAS**
	Frequency	**FREQUENCY OF DUTCH LEMMAS**

(See the information
in these diagrams for
the available columns)

Inflectional features	Separated	***Sepa***
	Singular	***Sing***
	Plural	***Plu***
	Diminutive	***Dim***
	Genitive	***Gen***
	Dative	***Dat***
	Positive	***Pos***
	Comparative	***Comp***
	Superlative	***Sup***
	With suffix 'e'	***Suff_e***
	Infinitive	***Inf***
	Participle	***Part***
	Present tense	***Pres***
	Past tense	***Past***
	1st person verb	***Sin1***
	2nd person verb	***Sin2***
	Inversed	***Inv***
	3rd person verb	***Sin3***
	Imperative only	***Imp***
	Subjunctive only	***Sub***
Type of flection		***FlectType***

9 FREQUENCY OF DUTCH WORDFORMS (N31)

INL frequency 42m	***Inl***
INL 95% confidence deviation 42m	***InlDev***
INL frequency 1m	***InlMln***
INL frequency, logarithmic	***InlLog***

Comp `Inflectional feature: comparative`

Type:	`character`	Null values:	`0`
Minimum value:	`N`	Minimum length:	`1`
Maximum value:	`Y`	Maximum length:	`1`
Characters:	`N Y`		

Dat `Inflectional feature: dative`

Type:	`character`	Null values:	`0`
Minimum value:	`N`	Minimum length:	`1`
Maximum value:	`Y`	Maximum length:	`1`
Characters:	`N Y`		

Dim `Inflectional feature: diminutive`

Type:	`character`	Null values:	`0`
Minimum value:	`N`	Minimum length:	`1`
Maximum value:	`Y`	Maximum length:	`1`
Characters:	`N Y`		

FlectType `Type of flection`

Type:	`character`	Null values:	`0`
Minimum value:	`C`	Minimum length:	`1`
Maximum value:	`vms`	Maximum length:	`5`
Characters:	`1 2 3 C D E G I P S X a d e g i m p s t v`		

Gen `Inflectional feature: genitive`

Type:	`character`	Null values:	`0`
Minimum value:	`N`	Minimum length:	`1`
Maximum value:	`Y`	Maximum length:	`1`
Characters:	`N Y`		

IdNum Word number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	381292	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

Imp Inflectional feature: imperative only

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Inf Inflectional feature: infinitive

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Inl INL frequency

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	2323977	Maximum length:	7
Characters:	0 1 2 3 4 5 6 7 8 9		

InlDev INL frequency deviation

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	108718	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

InlLog INL frequency, logarithmic

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	4.7391	Maximum length:	6
Characters:	. 0 1 2 3 4 5 6 7 8 9		

InlMln INL frequency (1,000,000)

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	54837	Maximum length:	5
Characters:	0 1 2 3 4 5 6 7 8 9		

InlSpellDev INL spelling frequency deviation

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	108718	Maximum length:	6
Characters:	0 1 2 3 4 5 6 7 8 9		

InlSpellFreq INL spelling frequency

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	2323977	Maximum length:	7
Characters:	0 1 2 3 4 5 6 7 8 9		

Inv Inflectional feature: inversed

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

OrthoCnt Number of spellings

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 3 4		

OrthoNum Spelling number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 3 4		

OrthoStatus Status of spelling

Type:	character	Null values:	0
Minimum value:	I	Minimum length:	1
Maximum value:	P	Maximum length:	1
Characters:	I N P		

Part Inflectional feature: participle

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Past Inflectional feature: past tense

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

PhonCLX Phon. wordform, CELEX charset

Type:	character	Null values:	2944
Minimum value:	&:.d.e:.m.	Minimum length:	2
Maximum value:	z.y:.r.z.u:.t.s.t.@.	Maximum length:	70
Characters:	␣ & . : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonCnt Wordform, number of phonemes

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	32	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

PhonCPA Phon. wordform, CPA charset

Type:	character	Null values:	2944
Minimum value:	@.	Minimum length:	2
Maximum value:	z.y:.r.z.u:.t.s.t.@.	Maximum length:	70
Characters:	␣ . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonCV Wordform, phon. CV pattern

Type:	character	Null values:	2944
Minimum value:	C	Minimum length:	1
Maximum value:	VVCCCC	Maximum length:	50
Characters:	␣ - C V		

PhonCVBr Wordform, phon. CV pattern, with brackets

Type:	character	Null values:	2944
Minimum value:	[CCCVCCCC]	Minimum length:	3
Maximum value:	[V][CVC][CCVC]	Maximum length:	62
Characters:	␣ C V [ ]		

PhonDISC Phon. wordform, DISC charset

Type:	character	Null values:	2944
Minimum value:	)d@kwisin@	Minimum length:	1
Maximum value:	}xt@nt	Maximum length:	32
Characters:	␣ ! ( ) * < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSAM Phon. wordform, SAM-PA charset

Type:	character	Null values:	2944
Minimum value:	/:.v.r.@.	Minimum length:	2
Maximum value:	}.x.t.@.n.t.	Maximum length:	70
Characters:	␣ . / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonStrsCLX Syll. phon. wordform, with stress, CELEX charset

Type:	character	Null values:	2944
Minimum value:	&:-’de:-m@	Minimum length:	2
Maximum value:	zwa:rt-’fOr-m@xst	Maximum length:	51
Characters:	␣ & ’ - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonStrsCPA Syll. phon. wordform, with stress, CPA charset

Type:	character	Null values:	2944
Minimum value:	’@	Minimum length:	2
Maximum value:	zy/l.’va:r.d@x	Maximum length:	51
Characters:	␣ ’ . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonStrsDISC Syll. phon. wordform, with stress, DISC charset

Type:	character	Null values:	2944
Minimum value:	’)-d@-kwi-si-n@	Minimum length:	2
Maximum value:	}r-zy-’li-n@	Maximum length:	45
Characters:	␣ ! ’ ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonStrsSAM Syll. phon. wordform, with stress, SAM-PA charset

Type:	character	Null values:	2944
Minimum value:	"/:-vr@	Minimum length:	2
Maximum value:	}r-zy:-"li:-n@	Maximum length:	51
Characters:	␣ " - / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSylBCLX Syll. phon. wordform, CELEX charset (brackets)

Type:	character	Null values:	2944
Minimum value:	[&:][de:][m@]	Minimum length:	3
Maximum value:	[zy:rt]	Maximum length:	62
Characters:	␣ & : @ A E G I N O S U Z [ ] a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylCLX Syll. phon. wordform, CELEX charset

Type:	character	Null values:	2944
Minimum value:	&:-de:-m@	Minimum length:	1
Maximum value:	zy:rt	Maximum length:	49
Characters:	␣ & - : @ A E G I N O S U Z a b d e f g h i j k l m n o p r s t u v w x y z		

PhonSylCPA Syll. phon. wordform, CPA charset

Type:	character	Null values:	2944
Minimum value:	@	Minimum length:	1
Maximum value:	zy:rt	Maximum length:	49
Characters:	␣ . / : @ A E G I J N O Q S Y Z a b d e f g h i j k l m n o p q r s t u v w x y z		

PhonSylDISC Syll. phon. wordform, DISC charset

Type:	character	Null values:	2944
Minimum value:	)-d@-kwi-si-n@	Minimum length:	1
Maximum value:	}x-t@nt	Maximum length:	43
Characters:	␣ ! ( ) * - < @ A E G I K L M N O S Z _ a b d e f g h i j k l m n o p r s t u v w x y z | }		

PhonSylSAM Syll. phon. wordform, SAM-PA charset

Type:	character	Null values:	2944
Minimum value:	/:-vr@	Minimum length:	1
Maximum value:	}x-t@nt	Maximum length:	49
Characters:	␣ - / : @ A E G I N O Q S Z a b d e f g h i j k l m n o p r s t u v w x y z | }		

Plu Inflectional feature: plural

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Pos Inflectional feature: positive

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Pres Inflectional feature: present tense

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sepa Separated wordform

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sin1 Inflectional feature: 1st person verb

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sin2 Inflectional feature: 2nd person verb

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sin3 Inflectional feature: 3rd person verb

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sing Inflectional feature: singular

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

StrsPat Wordform, stress pattern

Type:	character	Null values:	2944
Minimum value:	000000001	Minimum length:	1
Maximum value:	110000	Maximum length:	12
Characters:	␣ 0 1		

Sub Inflectional feature: subjunctive only

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Suff_e Inflectional feature: with suffix "e"

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Sup Inflectional feature: superlative

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

SylCnt Wordform, number of phonetic syllables

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	12	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

Word Word

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	zymotische	Maximum length:	36
Characters:	␣ ’ - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

WordCnt Word, number of letters

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	35	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

WordDia Word, diacritics

Type:	character	Null values:	0
Minimum value:	’m	Minimum length:	1
Maximum value:	überhaupt	Maximum length:	36

Characters: ␣ ’ - A B C D E F G H I J K L M N O P Q R S T
U V W X Y Z a b c d e f g h i j k l m n o p q
r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï
ñ ó ô ö û ü

WordLow Word, lowercase, alphabetical

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	zymotische	Maximum length:	35

Characters: a b c d e f g h i j k l m n o p q r s t u v w
x y z

WordLowSort Word, lowercase, alphabetical, sorted

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	z	Maximum length:	35

Characters: a b c d e f g h i j k l m n o p q r s t u v w
x y z

WordRev Word, reversed

Type:	character	Null values:	0
Minimum value:	a	Minimum length:	1
Maximum value:	zzibwohs	Maximum length:	36

Characters: ␣ ’ - A B C D E F G H I J K L M N O P Q R S T
U V W X Y Z a b c d e f g h i j k l m n o p q
r s t u v w x y z

WordSyl `Word, syllabified`

Type:	`character`	Null values:	0
Minimum value:	`'m`	Minimum length:	1
Maximum value:	`zy-mo-tisch`	Maximum length:	47
Characters:	`␣ ' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z`		

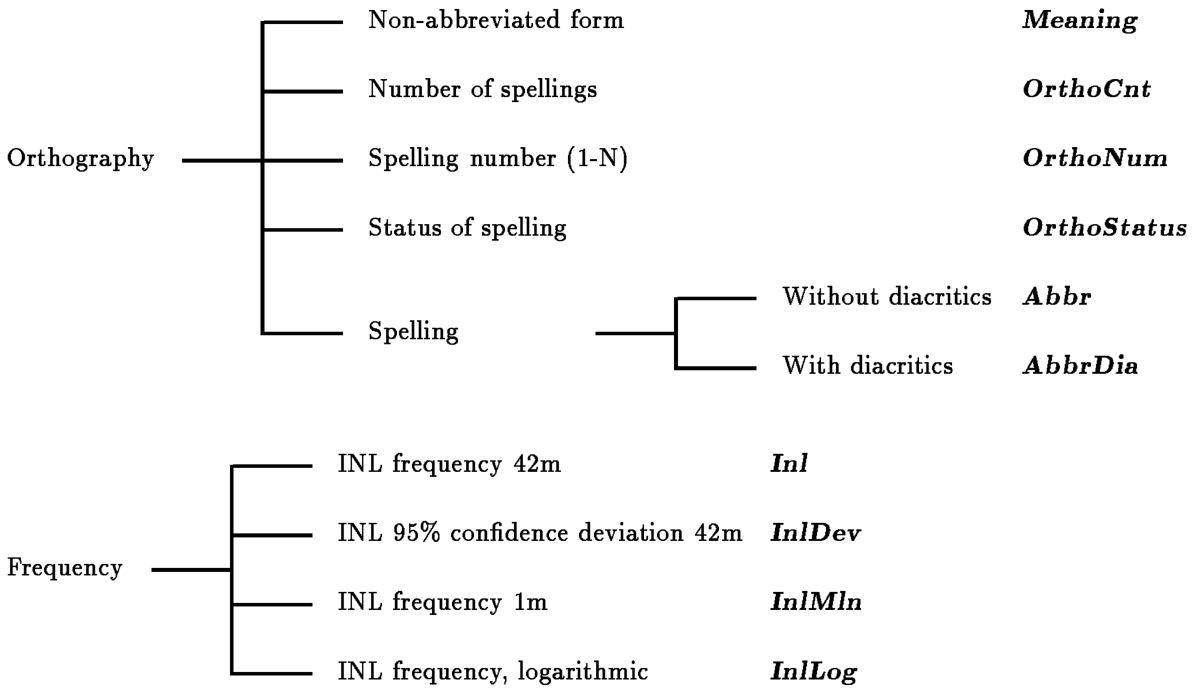
WordSylCnt `Word, number of orthographic syllables`

Type:	`numeric`	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	12	Maximum length:	2
Characters:	`0 1 2 3 4 5 6 7 8 9`		

WordSylDia `Word, syllabified, diacritics`

Type:	`character`	Null values:	0
Minimum value:	`'m`	Minimum length:	1
Maximum value:	`ü-ber-haupt`	Maximum length:	47
Characters:	`␣ ' - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z É Ü à á â ä å ç è é ê ë î ï ñ ó ô ö û ü`		

10 DUTCH ABBREVIATIONS (N31)



Abbr Abbreviations

Type:	character	Null values:	0
Minimum value:	A	Minimum length:	1
Maximum value:	zog.	Maximum length:	13
Characters:	␣ - . A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

AbbrDia Abbreviations, diacritics

Type:	character	Null values:	0
Minimum value:	A	Minimum length:	1
Maximum value:	zog.	Maximum length:	13
Characters:	␣ - . A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z ° ö		

IdNum Abbreviation number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	1315	Maximum length:	4
Characters:	0 1 2 3 4 5 6 7 8 9		

Inl INL frequency

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	2881	Maximum length:	4
Characters:	0 1 2 3 4 5 6 7 8 9		

InlDev INL frequency deviation

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	35179	Maximum length:	5
Characters:	0 1 2 3 4 5 6 7 8 9		

InlLog INL frequency, logarithmic

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	1.8325	Maximum length:	6
Characters:	. 0 1 2 3 4 5 6 7 8 9		

InlMln INL frequency (1,000,000)

Type:	numeric	Null values:	0
Minimum value:	0	Minimum length:	1
Maximum value:	68	Maximum length:	2
Characters:	0 1 2 3 4 5 6 7 8 9		

Meaning Non-abbreviated form

Type:	character	Null values:	0
Minimum value:	’t is te zeggen	Minimum length:	3
Maximum value:	zuster	Maximum length:	78
Characters:	␣ ’ ( ) , - A B C D E F G H I J K L M N O P R S T U V W Z a b c d e f g h i j k l m n o p q r s t u v w x y z		

OrthoCnt Number of spellings

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 4		

OrthoNum Spelling number

Type:	numeric	Null values:	0
Minimum value:	1	Minimum length:	1
Maximum value:	4	Maximum length:	1
Characters:	1 2 3 4		

OrthoStatus Status of spelling

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	P	Maximum length:	1
Characters:	N P		

11 DUTCH INL CORPUS TYPES (N31)

Orthography	Graphemic transcription	***Type***
Frequency	Absolute frequency	***Freq***
	Dispersion	***Disp***
	Indeterminate	***Indet***
	Provisional status	***Status***

Disp Dispersion

Type:	numeric	Null values:	0
Minimum value:	2	Minimum length:	1
Maximum value:	835	Maximum length:	3
Characters:	0 1 2 3 4 5 6 7 8 9		

Freq Frequency

Type:	numeric	Null values:	0
Minimum value:	2	Minimum length:	1
Maximum value:	2323977	Maximum length:	7
Characters:	0 1 2 3 4 5 6 7 8 9		

Indet Indeterminate

Type:	character	Null values:	0
Minimum value:	N	Minimum length:	1
Maximum value:	Y	Maximum length:	1
Characters:	N Y		

Status Status

Type:	character	Null values:	128428
Minimum value:	A	Minimum length:	1
Maximum value:	P	Maximum length:	1
Characters:	A D E N O P		

Type Corpus type spelling

Type:	character	Null values:	0
Minimum value:	-dat	Minimum length:	1
Maximum value:	zzw	Maximum length:	35
Characters:	" & ' ( ) , - . / 0 1 2 3 4 5 6 7 8 9 : = > ? @ a b c d e f g h i j k l m n o p q r s t u v w x y z		

2 COMPUTER PHONETIC CHARACTER CODES

The tables in this appendix exemplify the DISC character set in full. DISC is the character set which gives a single, unique code to each phonetic segment in the standard sounds systems of Dutch, English, and German. Here *segment* means consonant, affricate, syllabic consonant, short vowel, long vowel, diphthong or nasalized vowel.

Each table gives the IPA characters at the far left hand side, and the corresponding DISC characters on the far right hand side. In between come examples (where they occur) of words in Dutch, English, and German which exemplify the segments in question, and the code or codes used to represent those segments in the other character coding sets available: SAM-PA, CELEX, and CPA. This means you can use this appendix both as a full overview of DISC and a check on every phonetic character code used in the CELEX databases. If you just want to see the codes used for one particular language, then you should consult the Phonology section of the appropriate Linguistic Guide; you can also find general descriptions of the character sets there.

IPA	Dutch	English	German	SAM-PA	CELEX	CPA	DISC
p	put	pat	Pakt	p	p	p	p
b	bad	bad	Bad	b	b	b	b
t	tak	tack	Tag	t	t	t	t
d	dak	dad	dann	d	d	d	d
k	kat	cad	kalt	k	k	k	k
ɡ	goal	game	Gast	g	g	g	g
ŋ	lang	bang	Klang	N	N	N	N
m	mat	mad	Maß	m	m	m	m
n	nat	nat	Naht	n	n	n	n
l	lat	lad	Last	l	l	l	l
r, R	rat, later	rat	Ratte	r	r	r	r
f	fiets	fat	falsch	f	f	f	f
v	vat	vat	Welt	v	v	v	v
θ		thin		T	T	T	T
ð		then		D	D	D	D
s	sap	sap	Gas	s	s	s	s
z	zat	zap	Suppe	z	z	z	z
ʃ	sjaal	sheep	Schiff	S	S	S	S
ʒ	ravage	measure	Genie	Z	Z	Z	Z
j	jas	yank	Jacke	j	j	j	j
x, ç	licht, gaat	loch	Bach, ich	x	x	x	x
ɣ	regen			G	G	G	G
h	had	had	Hand	h	h	h	h
w		why	waterproof	w	w	w	w
ʋ	wat			w	w	w	w
pf			Pferd	pf	pf	pf	+
ts			Zahl	ts	ts	C/	=
ʧ		cheap	Matsch	tS	tS	T/	J
ʤ	jazz	jeep	Gin	dZ	dZ	J/	_
ŋ̩		bacon		N,	N,	N,	C
m̩		idealism		m,	m,	m,	F
n̩		burden		n,	n,	n,	H
l̩		dangle		l,	l,	l,	P
*		father *(linking 'r')*		r*	r*	r*	R

Disc Computer Phonetic Codes
consonants, affricates and syllabic consonants

IPA	Dutch	English	German	SAM-PA	CELEX	CPA	DISC
iː	liep	bean	Lied	i:	i:	i:	i
iːː	analyse			i::	i::	i::	!
ɑː		barn	Advantage	A:	A:	A:	#
aː	laat		klar	a:	a:	a:	a
ɔː		born	Allroundman	O:	O:	O:	$
uː	boek	boon	Hut	u:	u:	u:	u
ɜː		burn	Teamwork	3:	3:	@:	3
yː	buut		für	y:	y:	y:	y
yːː	centrifuge			y::	y::	y::	(
ɛː	scene		Käse	E:	E:	E:	)
œː	freule			/:	U:	Q:	*
ɒː	zone			Q:	O:	o:	<
eː	leeg		Mehl	e:	e:	e:	e
øː	deuk		Möbel	|:	&:	q:	|
oː	boom		Boot	o:	o:	o:	o
eɪ		bay	Native	eI	eI	e/	1
aɪ		buy	Shylock	aI	aI	a/	2
ɔɪ		boy	Playboy	OI	OI	o/	4
əʊ		no		@U	@U	O/	5
aʊ		brow	Allroundsportler	aU	aU	A/	6
ɪə		peer		I@	I@	I/	7
ɛə		pair		E@	E@	E/	8
ʊə		poor		U@	U@	U/	9
ɛi	wijs			EI	EI	y/	K
œy	huis			/I	UI	q/	L
ɑu	koud			Au	AU	A/	M
ai			weit	ai	ai	a/	W
au			Haut	au	au	A/	B
ɔy			freut	Oy	Oy	o/	X

Disc Computer Phonetic Codes
long vowels and diphthongs

IPA	Dutch	English	German	SAM-PA	CELEX	CPA	DISC
ɪ	lip	pit	Mitte	I	I	I	I
ʏ			Pfütze	Y	Y	Y	Y
ɛ	leg	pet	Bett	E	E	E	E
œ			Götter	/	Q	Q	/
æ		pat	Ragtime	{	&	^/	{
a			hat	a	a	a	&
ɑ	lat		Kalevala	A	A	A	A
ɒ		pot		Q	O	O	Q
ʌ		putt	Plumpudding	V	V	^	V
ɔ	bom		Glocke	O	O	O	O
ʊ		put	Pult	U	U	U	U
ʉ	put			}	U	Y/	}
ə	gelijk	another	Beginn	@	@	@	@
œ̃ː			Parfum	/˜:	Q˜:	Q˜:	^
æ̃		timbre	impromptu	{˜	&˜	^/˜	c
ɑ̃ː		détente	Détente	A˜:	A˜:	A˜:	q
æ̃ː		lingerie	Bassin	{˜:	&˜:	^/˜:	0
ɒ̃ː		bouillon	Affront	O˜:	O˜:	O˜:	˜

Disc Computer Phonetic Codes
SHORT VOWELS AND NASALIZED VOWELS

IPA	Description	SAM-PA	CELEX	CPA	DISC
ː	length marker	:	:	:	
-	syllable marker	-	-	.	-
ˈ	primary stress	"	'	'	'
ˌ	secondary stress	%	"	"	"
˜	nasalization	˜	˜	˜	
	examples:	A˜:	A˜:	A˜:	

Disc Computer Phonetic Codes
LENGTH, STRESS, SYLLABLE AND NASALIZATION MARKERS

3 ASCII AND EIGHT-BIT CHARACTER CODES

The two tables which follow show full details of the seven and eight bit character codes used by CELEX on its DIGITAL VAX/VMS computer systems. They are particularly useful when you need to transfer data to or from the CELEX machine: you can find out which codes must be converted. The first table shows the basic characters in use – they are the standard seven bit ASCII codes, and most ASCII terminals and printers should reproduce these characters as shown. The second table shows the eight bit codes which DIGITAL VT200 and VT300-series terminals can reproduce; these are the codes which provide the diacritic characters available in some columns in the CELEX databases.

Most of the printable seven and eight bit codes conform to the standard character set known as ISO 8859-1 (Latin Alphabet No. 1) or ECMA-94. There are some exceptions, however. The ISO 8859-1 (decimal) characters 160, 164, 166, 172, 173, 174, 175, 184, 190, 208, 222, 240, 254, and 255 are not implemented in the DIGITAL set, and 168, 215, and 247 each produce a character other than the ISO 8859-1 recommended one.

For details about each character, consult the DIGITAL VMS General User Guide, Volume 2A *Guide to using VMS* (VMS version 5.0, April 1988), pages A-6—A-11.

	0	1	2	3	4	5	6	7	
0	NUL 0 0 0	SOH 1 1 1	STX 2 2 2	ETX 3 3 3	EOT 4 4 4	ENQ 5 5 5	ACK 6 6 6	BEL 7 7 7	0
	BS 10 8 8	HT 11 9 9	LF 12 10 A	VT 13 11 B	FF 14 12 C	CR 15 13 D	SO 16 14 E	SI 17 15 F	
1	DLE 20 16 10	DC1 21 17 11	DC2 22 18 12	DC3 23 19 13	DC4 24 20 14	NAK 25 21 15	SYN 26 22 16	ETB 27 23 17	1
	CAN 30 24 18	EM 31 25 19	SUB 32 26 1A	ESC 33 27 1B	FS 34 28 1C	GS 35 29 1D	RS 36 30 1E	US 37 31 1F	
2	SP 40 32 20	! 41 33 21	” 42 34 22	# 43 35 23	$ 44 36 24	% 45 37 25	& 46 38 26	’ 47 39 27	2
	(50 40 28	) 51 41 29	* 52 42 2A	+ 53 43 2B	, 54 44 2C	− 55 45 2D	. 56 46 2E	/ 57 47 2F	
3	0 60 48 30	1 61 49 31	2 62 50 32	3 63 51 33	4 64 52 34	5 65 53 35	6 66 54 36	7 67 55 37	3
	8 70 56 38	9 71 57 39	: 72 58 3A	; 73 59 3B	< 74 60 3C	= 75 61 3D	> 76 62 3E	? 77 63 3F	
4	@ 100 64 40	A 101 65 41	B 102 66 42	C 103 67 43	D 104 68 44	E 105 69 45	F 106 70 46	G 107 71 47	4
	H 110 72 48	I 111 73 49	J 112 74 4A	K 113 75 4B	L 114 76 4C	M 115 77 4D	N 116 78 4E	O 117 79 4F	
5	P 120 80 50	Q 121 81 51	R 122 82 52	S 123 83 53	T 124 84 54	U 125 85 55	V 126 86 56	W 127 87 57	5
	X 130 88 58	Y 131 89 59	Z 132 90 5A	[133 91 5B	\ 134 92 5C	] 135 93 5D	ˆ 136 94 5E	— 137 95 5F	
6	‘ 140 96 60	a 141 97 61	b 142 98 62	c 143 99 63	d 144 100 64	e 145 101 65	f 146 102 66	g 147 103 67	6
	h 150 104 68	i 151 105 69	j 152 106 6A	k 153 107 6B	l 154 108 6C	m 155 109 6D	n 156 110 6E	o 157 111 6F	
7	p 160 112 70	q 161 113 71	r 162 114 72	s 163 115 73	t 164 116 74	u 165 117 75	v 166 118 76	w 167 119 77	7
	x 170 120 78	y 171 121 79	z 172 122 7A	{ 173 123 7B	| 174 124 7C	} 175 125 7D	˜ 176 126 7E	DEL 177 127 7F	
	8	9	A	B	C	D	E	F	

Character	O 117 79 4F	Octal Decimal Hexadecimal

DIGITAL/CELEX SEVEN-BIT ASCII CODES

	0	1	2	3	4	5	6	7	
8	200 128 80	201 129 81	202 130 82	203 131 83	IND 204 132 84	NEL 205 133 85	SSA 206 134 86	ESA 207 135 87	8
	HTS 210 136 88	HTJ 211 137 89	VTS 212 138 8A	PLD 213 139 8B	PLU 214 140 8C	RI 215 141 8D	SS2 216 142 8E	SS3 217 143 8F	
9	DCS 220 144 90	PU1 221 145 91	PU2 222 146 92	STS 223 147 93	CCH 224 148 94	MW 225 149 95	SPA 226 150 96	EPA 227 151 97	9
	230 152 98	231 153 99	232 154 9A	CSI 233 155 9B	ST 234 156 9C	OSC 235 157 9D	PM 236 158 9E	APC 237 159 9F	
A	240 160 A0	¡ 241 161 A1	¢ 242 162 A2	£ 243 163 A3	244 164 A4	¥ 245 165 A5	246 166 A6	§ 247 167 A7	A
	¤ 250 168 A8	© 251 169 A9	ª 252 170 AA	« 253 171 AB	254 172 AC	255 173 AD	256 174 AE	257 175 AF	
B	° 260 176 B0	± 261 177 B1	² 262 178 B2	³ 263 179 B3	264 180 B4	µ 265 181 B5	¶ 266 182 B6	· 267 183 B7	B
	270 184 B8	¹ 271 185 B9	º 272 186 BA	» 273 187 BB	¼ 274 188 BC	½ 275 189 BD	276 190 BE	¿ 277 191 BF	
C	À 300 192 C0	Á 301 193 C1	Â 302 194 C2	Ã 303 195 C3	Ä 304 196 C4	Å 305 197 C5	Æ 306 198 C6	Ç 307 199 C7	C
	È 310 200 C8	É 311 201 C9	Ê 312 202 CA	Ë 313 203 CB	Ì 314 204 CC	Í 315 205 CD	Î 316 206 CE	Ï 317 207 CF	
D	320 208 D0	Ñ 321 209 D1	Ò 322 210 D2	Ó 323 211 D3	Ô 324 212 D4	Õ 325 213 D5	Ö 326 214 D6	Œ 327 215 D7	D
	Ø 330 216 D8	Ù 331 217 D9	Ú 332 218 DA	Û 333 219 DB	Ü 334 220 DC	Ÿ 335 221 DD	336 222 DE	ß 337 223 DF	
E	à 340 224 E0	á 341 225 E1	â 342 226 E2	ã 343 227 E3	ä 344 228 E4	å 345 229 E5	æ 346 230 E6	ç 347 231 E7	E
	è 350 232 E8	é 351 233 E9	ê 352 234 EA	ë 353 235 EB	ì 354 236 EC	í 355 237 ED	î 356 238 EE	ï 357 239 EF	
F	360 240 F0	ñ 361 241 F1	ò 362 242 F2	ó 363 243 F3	ô 364 244 F4	õ 365 245 F5	ö 366 246 F6	œ 367 247 F7	F
	ø 370 248 F8	ù 371 249 F9	ú 372 250 FA	û 373 251 FB	ü 374 252 FC	ÿ 375 253 FD	376 254 FE	377 255 FF	
	8	9	A	B	C	D	E	F	

Character	Ö	326 214 D6	Octal Decimal Hexadecimal

DIGITAL/CELEX EIGHT-BIT CODES

4 GLOSSARY

ABBREVIATION A term which refers to the shortened form of a normal word or phrase which can be used when the word or phrase itself is thought to be too long or unwieldy. A special LEXICON TYPE which contains nothing but abbreviations is available. An abbreviation can take one of three general forms, of which the most common is the *contraction*, where particular letters (often vowels) are removed from the word (thus *Gld.* is an abbreviation of *Gelderland*). Another form is the *acronym*, where the initial letters of each constituent word in a phrase are joined to make a new word (thus *FIFA* is an acronym of *__F__édération __I__nternationale de __F__ootball __A__ssociation*). Finally there is also *truncation*, where a number of letters is removed from the end of a word (thus the chemical symbol for *Argon* is *Ar*).

ABSTRACT STEM This term refers to an alternative orthographic form of the STEM. When a STEM ends in '*-s*' or '*-f*', and when, in any of its related WORDFORMS, that '*-f*' becomes a '*v*' (the verb *leven: ik leef, wij leven)* or that '*s*' becomes a '*z*' (the noun *kaas:* singular *kaas*, plural *kazen*), then the abstract stem is given with the endings '*-v*' and '*-z*' respectively (thus *leev* and *kaaz* instead of the normal stems *leef* and *kaas*). All other abstract stems have the same form as the normal STEM.

AFFIX SUBSTITUTION This refers to the process by which an affix replaces part of a lemma when the affix and the lemma combine to make a new lemma. An example is the English lemma *fatuity*, where the headword *fatuous* can be said to lose the affix *-ous* and gain the affix *-ity*.

ALPHABETIC KEYS This refers to the letters—as opposed to the numbers and various other symbols—that are on your keyboard, twenty-six upper case and twenty-six lower case characters. When you are working in FLEX, you can use them to move to the nearest menu option which begins with the letter you press.

AND OPERATOR The logical connective combining two RESTRICTIONS (or groups of RESTRICTIONS) *x* and *y* in such a way that a ROW is included in the LEXICON only if both *x* and *y* are true for that ROW; otherwise the ROW is not included in the LEXICON.

ASCII The seven-bit binary number coding system used to represent alphabetic, numeric, punctuation and other characters in some types of computers. The letters stand for *American Standard Code for Information Interchange.*

BACKTRACK KEY This is a FLEX term which refers to the key you press to reverse back down the menu path you have just come along. You generally use the backtrack key to leave a menu window you do not wish to use, and to return to the previous window.

BAR This refers to the way FLEX indicates which option you are choosing in a particular window. Usually bold text, underlined text or reverse video text is used to differentiate the current option (the one you will get if you press return) from the others.

BATCH MODE This allows you, or FLEX working for you, to submit certain commands to the computer which it carries out as a separate, non-interactive job. FLEX uses batch mode for certain types of job, because in this way they are executed while the computer is not being used for smaller jobs or more important jobs.

BELL This refers to the noise your TERMINAL can make to attract your attention. In FLEX, the bell normally sounds when a new WINDOW appears, or when a message is displayed.

CANCELLED This refers to the status of a FLEX job which either you or FLEX has asked to be stopped. When you cancel a job, the computer stops working on it, and ignores any results already achieved by that job.

CLASS LABELS A simple coding system used to indicate the syntactic class of a word: *n* means a noun, *a* means an adjective, and so on. They can be used instead of a numeric coding system, or typing the syntactic class in full.

COBUILD This is an acronym for ***Collins Birmingham University International Language Database***. In 1987, COBUILD published the *Collins Cobuild English Language Dictionary*, which is based on analysis of their large CORPUS of modern English. The FREQUENCY information in the CELEX English DATABASE was taken from this corpus which at the time contained 17,900,000 words.

COLUMN A database term which refers to the storage of one particular type of information: a column can contain a specific sort of words, or codes, or analyses.

COMPLETE SEGMENTATION This means the full derivational morphological analysis of a lemma into all its constituent morphemes.

COMPLETED This is a FLEX term which indicates that a job working in BATCH MODE has now finished successfully.

COPY This is a FLEX term which refers to the creation of a new LEXICON by using the definitions (i.e. the COLUMNS and RESTRICTIONS) already specified for a different LEXICON. The LEXICON you copy can be your own, or, if you have a GRANT, someone else's.

CORPUS A sizeable collection of words, usually written texts, which can be used and processed by computers. Three text corpora were used to provide CELEX's FREQUENCY information: the INL, COBUILD, and EINDHOVEN corpora. They all contain modern-day texts drawn from diverse printed sources, such as recently-published books, newspapers and magazines, and sometimes, though to a much lesser extent, transcriptions made from recordings of speech.

CORPUS TOKEN A term which refers to the units distinguished during the DISAMBIGUATION by computer of a text CORPUS used to provide FREQUENCY information. A corpus token is any string containing *at least* one ALPHABETIC character, *along with* zero or more ALPHANUMERIC characters. The INL CORPUS contained 43,549,704 tokens, and the COBUILD CORPUS contained 17,900,000 tokens.

CORPUS TYPE A term which refers to a CORPUS TOKEN that occurs one or more times in a CORPUS. During the process of DISAMBIGUATION, the occurrence CORPUS TOKENS can be quantified. Whenever a new CORPUS TOKEN is discovered, it is also noted as a corpus type, and thereafter any re-occurrences counted to give the FREQUENCY count of the type. The type which accounts for the greatest number of tokens in the INL CORPUS is *de*; it occurs 2,440,897 times.

CPA A computer phonetic alphabet developed the Ruhr Universität Bochum. The letters stand for *Computer Phonetic Alphabet*.

CURSOR An indicator, usually a small flashing box or a line, used to indicate where the next character will appear or, in FLEX, to mark the current menu option, usually in conjunction with a BAR.

CV PATTERNS A CV pattern is a re-written orthographic, phonetic or phonological transcription in which, generally speaking, any vowels or diphthongs are replaced by the letter V, and consonants by the letter C.

DATABASE A database is a collection of information stored in computer files in such a way as to make the retrieval of that information quicker and more flexible.

DATANET-1 This is the name of the main public PSDN in the Netherlands. At present, all SURFnet nodes are DATANET-1 nodes, since SURFnet uses DATANET-1

DBMS These letters stand for *database management system*, which is computer software designed to facilitate the use and development of a DATABASE. CELEX and FLEX use the relational DBMS marketed by the ORACLE company.

DELIMITER This refers to a character or group of characters used in a FILE to indicate the beginning or end of every FIELD.

DERIVATIONAL/COMPOSITIONAL SEGMENTATION This is the the type of morphological analysis which identifies the constituent LEMMAS, affixes and morphemes in a lemma, as opposed to inflectional analysis which deals with the WORDFORMS each lemma takes.

DIACRITICS The markers used in conjunction with regular orthographic characters to indicate some difference in pronunciation or stress, as with the German ümlaut, the French ácute, and the Czech háček.

DISAMBIGUATION This term refers to the process by which the FREQUENCY of words in a large text CORPUS can be established, either by computer, or people, or both. The process tries to link each word in the corpus (that is, each string consisting of one alphanumeric charcter plus at least one alphabetic character, with a space on either side) with a LEMMA. If a string occurs more than once, and if such a link can be made, then the word is considered to be a WORDFORM, and the number of times the link was made is the FREQUENCY of that WORDFORM.

DISC This is the name of the CELEX computer phonetic alphabet which uses one unique, distinct character for each vowel, long vowel, diphthong, consonant and affricate. Although not elegant in appearance, it is useful for computer processing.

DRAFT A FLEX term which refers to the VERSION of a LEXICON. It indicates that its definition is stored by FLEX, and that when you use the LEXICON, the information is extracted from the main CELEX database using that definition. Contrast with FIXED.

DTE These letters stand for *data terminal equipment*, which, for most CELEX users, normally just means 'computer'.

ETHERNET A special communications set-up for a LAN which allows different sorts of computers and other devices to be linked without central control from any one computer.

EXECUTING This is a FLEX term which indicates that a job is currently being carried out in BATCH MODE.

EXPORT This is a FLEX term which refers to the process of making a normal VAX/VMS file from the contents of a LEXICON.

EXPRESSION This is a FLEX term which refers to the right-hand part of a RESTRICTION; that is, the part which contains some number, word, or WILD CARD. A column name is linked to an expression by means of an OPERATOR.

FIELD In FLEX, this refers to that part of a window where information from the database appears. In a VAX/VMS FILE, it refers to a specific part of a line which is used for a particular sort of information.

FILE A collection of data stored for computer use, and arranged in a way which is significant to the user.

FINITE FORMS This refers to those flections which can occur in their own right in a main clause or sentence, and which indicate differences in tense and person for example *ik beweg, ik bewog, wij bewegen, wij bewogen.*

FIXED A FLEX term which refers to the VERSION of a LEXICON. A FIXED LEXICON is a separate, independent database which contains information originally taken from the central CELEX databases, and when you use it, the information is extracted from this database rather than the central CELEX database. Contrast with DRAFT.

FIXED FORMAT FILE This is a FILE whose FIELDS are always a fixed number of characters wide, regardless of the width of the data each field contains.

FLAT SEGMENTATION This is one type of derivational/compositional morphological analysis. It reduces a lemma directly to its constituent morphemes, without showing any of the intermediate levels of analysis you get. Contrast with HIERARCHICAL SEGMENTATION.

FLEX$EXP This is a LOGICAL NAME which refers to the DIRECTORY of your CELEX account which is set aside specifically for FILES which are extracted from FLEX using the EXPORT facility.

FREQUENCY The number of times a CORPUS TYPE occurs in a particular CORPUS. For example, the WORDFORM *radio* has an INL frequency of 2394, as counted in the 43,549,704 word INL CORPUS. This figure can also be expressed proportionally (i.e. the frequency expected per million words) or logarithmically. To arrive at a figure for the frequency of LEMMAs, the frequencies of its inflectional forms (that is, its WORDFORMS) are added together.

FULLY SYLLABIFIED This refers to orthographic transcriptions which have a syllable marker whenever a syllable boundary occurs within a word, including single-letter syllables which occur at the beginning or end of a word. Contrast with PARTIALLY SYLLABIFIED.

GATEWAY This refers to the point of interconnection between two different communications networks. Often users are not aware they are using GATEWAYS; occasionally, though, you may first have to connect to a GATEWAY before being able to use the other network.

GRANT This is a FLEX term that indicates whether one or more particular FLEX users, or every FLEX user, can COPY a LEXICON created by you.

GRAPHEMIC This is the adjective used to denote characters which occur in normal Dutch, English or German orthography. It is used to distinguish phonetic or phonological transcriptions, which use specifically phonetic character alphabets, from transcriptions which are written or typed using the roman alphabet.

HEADWORD A term which refers to one of the two forms a LEMMA is given in the CELEX databases. It corresponds to the traditional lexicographic headword to be found in dictionaries. In Dutch, German, and English the forms used always resemble words that occur naturally in the language, rather than abstract forms. Thus in Dutch, the headword of a noun is its singular form. (For a definitive list of the forms used, consult Appendix IV). Contrast with STEM.

HELP KEY This is a FLEX term that refers to the key you press to receive on-line advice on how to use FLEX as you are working with it.

HIDDEN This is a FLEX term which refers to the COLUMNS displayed using the SHOW option. If your LEXICON contains so many COLUMNS that not all of them can be displayed at once on screen, then you can indicate that certain columns should temporarily be missed out of the display, so that you can see other columns of more interest. The missed out columns are called HIDDEN columns.

HIERARCHICAL SEGMENTATION This is one type of derivational or compositional morphological analysis. It reduces a lemma directly to its constituent morphemes, showing all the intermediate levels of analysis involved in arriving at all the morphemes. Contrast with FLAT SEGMENTATION.

IMMEDIATE SEGMENTATION This is one type of derivational or compositional morphological analysis. It reduces a lemma to its next biggest components – other lemmas, affixes or morphemes. To arrive at COMPLETE SEGMENTATION, IMMEDIATE SEGMENTATION may have to be carried out several times.

INDEX This is a DATABASE term which refers to COLUMNS whose contents are indexed in a way conceptually identical to the indexing of book. Information from COLUMNS with an index can be looked up more quickly by the DBMS.

INL This is the normal abbreviation for *Instituut voor Nederlandse Lexicologie*, the Dutch Lexicography Institute in Leiden. They are developing a large text CORPUS of modern written Dutch, and the FREQUENCY information contained in the CELEX Dutch database was extracted from this CORPUS when it contained over 43 million words. It is still being extended, and now contains more than 45 million words.

INTEGRITY A term which refers to the protection of information stored in a DATABASE when it can be altered by two or more sources. A DATABASE maintains its integrity so long as only one source can alter the data at any one time. If two people try to alter the same data at the same time, the resulting information is no longer consistent, and the integrity of the DATABASE is lost.

INTERVAL This is a FLEX option which allows you to specify a particular set of consecutive ROWS in your LEXICON for EXPORT

IPA This letters stand for *International Phonetic Alphabet*, the set of written characters approved for phonetic transcription by the International Phonetic Association.

ISO These letters stand for the *International Standards Organization*, the Swiss-based organization which is involved in developing and coordinating worldwide standards.

LAN These letters stand for *local area network*, and refer to a communications network which links a number of computers over a relatively small area, such as a factory plant or university.

LAT These letters stand for *local area transport*, and refer to the protocols a DEC terminal server uses to communicate with computers using VAX/VMS over an ETHERNET.

LANGUAGE CODES These are codes used in the English database to provide background information about some lemmas, such as the national origin words loaned from other languages and whether certain lemmas are more likely to be British or American English.

LEMMA A term intended to signify the abstract notion which underlies a family of inflected forms, so that, for example, *walk* could be the lemma underlying the verbal forms *walk, walks, walked,* and *walking.* In the CELEX databases, lemmas are distinguished on the basis of (1) the *pronunciation,* (2) the *syntactic class,* (3) the *morphological structure,* (4) the *orthographic form* of their various WORDFORMS, as well as (5) the full *inflectional paradigm* of the lemma. **No explicit consideration of meaning is involved**, so in the CELEX databases, the lemmas of any two (or more) words which differ in meaning but which otherwise are identical in each of these five ways are reduced to *one* lemma. In principle, any convenient form could be used to represent a lemma: an abstract form, or even a number. In practice, CELEX uses two forms: the HEADWORD and the STEM.

LEVEL This refers to any one analytical step in morphological analysis. COMPLETE SEGMENTATION is finished when every possible level of analysis has been carried out.

LEXICON This term refers to a subset of one of the CELEX databases which you can define for yourself using FLEX. Rather than using the entire database at all times, you specify certain COLUMNS and delimit their contents using RESTRICTIONS to form a coherent subset of information drawn from the central database.

LEXICON TYPE This is a FLEX term that indicates which of the central CELEX databases the information in a LEXICON is drawn from. Each of the central databases has as its main subject one type of canonical form, such as Dutch lemmas or English wordforms. The type of canonical form is then used to indicate the type of LEXICON.

LISP This is a high level programming language often used in artificial intelligence work. In particular, it uses a special brackets notation for its input and output data.

LOCKED This means that FLEX is currently working on your LEXICON, and that in order to protect its INTEGRITY, you cannot do any more work with it until the job FLEX is doing has finished.

LOGICAL COMBINATION This is a FLEX term which refers to the way RESTRICTIONS or groups of RESTRICTIONS linked by brackets work together to delimit the contents of a LEXICON, by means of the AND OPERATOR, the OR OPERATOR, and the NOT OPERATOR.

LOGICAL NAME A VAX/VMS term which refers to a specific DIRECTORY in your account. It is used as part of a FILE name to help you to remember where it is, and the computer to know how to find it or store it.

LOGIN This refers to the way you identify yourself to the computer before beginning any work. You normally have to give the name of your account and a password.

LOGOUT This refers to the way you indicate to the computer that you want to stop working. On the CELEX machine, you simply type `logout`.

MAIL This a FLEX term and a VMS term. In FLEX, it refers to the main menu option MAIL, which allows you to communicate with other FLEX users purely within FLEX; it does not link in with the other national or international networks. In VMS, there is a more comprehensive mail facility which allows you to send messages to other CELEX computer users, as well as users on other computers via DECnet or DATANET-1.

MENU This is a FLEX term which refers to the boxes displayed on your screen from which you can choose an option that allows you to continue with your work, or a particular type of information. Compare WINDOW.

MESSAGE LINE This is a FLEX term which refers to the line immediately above the bottom line of the screen. It displays instructions, error messages and other information to help you as you use FLEX, and whenever CELEX computer system messages are sent to your terminal, they are also displayed here.

MODEM This is an an acronym for the words *modulator and demodulator.* It is a machine which converts the characters from your computer (a *digital* bit stream) into a form (an *analog* signal) that can be transmitted along a telephone line; this is *modulation.* It can also convert the analog signal received down a telephone line back into the digital bit stream used in your computer; this is *demodulation.* Thus you can use telephone lines to work interactively with a computer that might be located hundreds of miles away, provided that you have a terminal and a modem, and the remote computer is also linked to a modem.

NEXT KEY This is a FLEX term which refers to the key you press to display more information in a WINDOW or MENU.

NOT OPERATOR The logical connective applied to one RESTRICTION or group of RESTRICTIONS *z* in such a way that a ROW is included in the LEXICON if *z* is untrue. If *z* is true, the ROW is not included in the LEXICON.

ON VIEW This is a FLEX term which is important for columns that are used in the construction of RESTRICTIONS. If a column is ON VIEW, you can see it when you display your lexicon using the SHOW or EXPORT options. If it is *not* ON VIEW, you never see it, but it still works in any restriction you have made with it. All columns are ON VIEW by default; you can change this in the EDIT RESTRICTIONS menu.

OPERATING SYSTEM This refers to the software which you use specifically to control a computer or a computer system. The commands you type to start a program running or to give a DIRECTORY listing are OPERATING SYSTEM commands. The CELEX computers use VAX/VMS.

OPERATOR This is a FLEX term which refers to the simple mathematical relation symbols that you can use in RESTRICTIONS.

OR OPERATOR The logical connective combining two RESTRICTIONS (or groups of RESTRICTIONS) *x* and *y* in such a way that a ROW is included in the LEXICON if (i) either *x* or *y* is true for that ROW, or (ii) both *x* and *y* are true for that ROW; otherwise the ROW is not included in the LEXICON.

PAD These letters stand for *packet assembler/disassembler*, a device (or program) which gathers individual characters that you send from your TERMINAL or computer and puts them into groups (that is, *packets*) which can then be sent across a PSDN to some other computer. Likewise when packets come back to your computer across the PSDN, the PAD splits them up into individual characters again, ready for display on your terminal.

PAGE This is a FLEX term that refers to data displayed in the SHOW window: there is room for ten lines of information on screen, and one PAGE is equal to these ten lines.

PARTIALLY SYLLABIFIED This refers to orthographic transcriptions which indicate each syllable boundary within a word by means of a hyphen, with the exception of syllables at the beginning or end of the word which consist of only one letter; such syllables are not marked. Compare with FULLY SYLLABIFIED.

PENDING This is a FLEX term which means that a BATCH JOB cannot be executed by the computer at the moment, usually because other BATCH JOBS are being carried out. A job which is PENDING will eventually be carried out, however, unless you CANCEL it.

PREV KEY This is a FLEX term which refers to the key you press to re-display old information that you have already seen in the WINDOW or MENU you are currently working in.

PSDN These letters stand for *packet switching data network*, which is a wide area network that can control the rapid transmission of packets of data (possibly prepared by a PAD, for example) between different points in the network. PSDNs enable you to work interactively on a computer which is located hundreds of miles away. In the Netherlands, the public X25 PSDN is called DATANET-1, and it is currently used in the implementation of SURFnet.

PSI These letters stand for *packetnet system interface*, the VAX/VMS software product that enables VAX computers to link up with PSDNs. It performs the function of a PAD.

PSS These letters stand for *packet switch stream*, the name of the British X25 PSDN.

QUERY This is a FLEX term that refers to the SHOW menu option that allows you to look at a particular part of your lexicon. It does not permanently alter your lexicon.

REDRAW This is a FLEX term that refers to the key which you press to redisplay all the FLEX information currently displayed on screen. It allows you to correct any badly-drawn lines or get rid of unwanted messages or stray characters.

RESTRICTION This is a FLEX term which refers to a simple logical statement you formulate to specify in detail the information to be included in your lexicon, with reference to the contents of the COLUMNS already in your lexicon.

ROW A database term which refers to the storage of different types of information which refer to one word: each row contains an orthographic transcription, a phonetic transcription, a morphological analysis, a syntactic code and a frequency count (and more besides) for each word.

SAM-PA These letters stand for *Speech Assessment Methods Phonetic Alphabet*. SAM is an Esprit (European Community funded) project. The development of the phonetic alphabet was co-ordinated by John Wells with the intention of it becoming the standard European computer phonetic alphabet.

SEGMENTATION This is a term which refers to the process of morphological analysis of words into their constituent lemmas, affixes and morphemes.

SQL*PLUS This is the name of the standard DBMS produced by the ORACLE company. It is DBMS used by CELEX, when you work with FLEX, you are using a system which generates SQL*PLUS code to access the CELEX databases.

STATUS LINE This is a FLEX term which refers to the very bottom line of the screen. It displays your FLEX username, the name of the lexicon you have selected, and version number of the FLEX program you are using.

STEM A term which refers to one of the two forms a LEMMA is given in the CELEX Dutch database, and the term used in place of HEADWORD in English morphology. It is that part of a LEMMA's inflectional paradigm which is common to all the inflected forms, separate from the inflectional affixes themselves. Usually, it is identical to the HEADWORD *except* for Dutch verbs, where it takes the form of the first person singular, present tense (but see also ABSTRACT STEM). In English morphology, a STEM is a HEADWORD, or sometimes a flectional form of a headword.

STRESS PATTERN This refers to special strings of numbers, each of which represents one phonetic syllable and indicates how that syllable is stressed. A zero always means 'unstressed'; a '1' indicates 'stressed' in stress patterns for Dutch words and 'primary stress' for English words; '2' indicates 'secondary stress' for English words.

SURFNET This is the Dutch national academic computer network which provides electronic mail facilities and logins to computers all over the Netherlands. At present, it uses DATANET-1 to carry out its work.

SYLLABIC CONSONANT This term refers to a consonant which by itself or with other consonants forms a distinct syllable in the pronunciation of a word, without the presence of a vowel. The final *-l* in the word *bottle* can be realised as a SYLLABIC CONSONANT.

TERMINAL This is a device which can accept data from and transmit data to a computer. For most people a TERMINAL is a *visual display unit* (VDU for short), which consists of a television-like screen to display data received, and a keyboard to transmit data, including OPERATING SYSTEM commands. There are many types of TERMINAL, all with their own specific control codes and capabilities.

TERMINAL EMULATOR This is a type of software which allows your personal computer or TERMINAL to behave and respond like another sort of terminal.

TERMINAL SERVER A device that connects TERMINALS (and modems and printers) to an ETHERNET.

VAX/VMS The trademark used by the Digital Electronic Corporation (DEC) to identify the OPERATING SYSTEM used on their VAX series computers. VAX stands for *Virtual Addressing eXtension*, and VMS stands for *Virtual Memory System*.

VERSION This is a FLEX term that refers to the way your lexicon is stored. If it is a *draft* lexicon, only the definition is stored, and when you use it, the data it requires is looked up in the main CELEX database. If it is a *fixed* lexicon, it is a separate, probably much smaller database which is quicker and easier used.

VT100 This refers to a standard DEC type of TERMINAL. Users who have such a TERMINAL, or who have a TERMINAL EMULATOR which can imitate such a terminal, should be able to log into CELEX and use FLEX with no problems.

VT220 This refers to a standard DEC type of TERMINAL which is newer than the VT100. It is the default TERMINAL type for CELEX and FLEX.

WAN These letters stand for *wide area network* and refer to a communications network which links a number of computers over a relatively large area. Sometimes these networks cover entire nations (such as SURFnet in the Netherlands) or even larger areas (such as EARN, the European academic network).

WILDCARD This refers to the % and _ characters which can be used in a RESTRICTION or QUERY to indicate respectively 'any character or group of characters' and 'any single character'.

WINDOW This is a FLEX term which refers to the boxes shown on your screen which contain either MENU options, data drawn from the database, or other relevant FLEX information. A window which contains options is almost always called simply a MENU.

WORDFORM A term which is synonymous with *word* in the general sense. Wordforms are the units occurring in natural language, which, when written, are bounded on either side by a space, and which can be associated with a LEMMA. (However some English and Dutch wordforms include spaces – *swimming pool*, for example, or *Nederlandse Spoorwegen)*. They are the INFLECTED FORMS in regular use, as opposed to LEMMAS, STEMS, and HEADWORDS which are convenient, but abstract, representations of complete families of wordforms).

X25 This refers to the standard protocols recommended by the *Comité Consultatif International Télégraphique et Téléphonique* for equipment operating within a PSDN.

X29 This refers to the standard procedures recommended by the *Comité Consultatif International Télégraphique et Téléphonique* for the exchange of user data and the required control information between your terminal and a remote PAD over a PSDN.