ENGLISH LINGUISTIC GUIDE

Despite the remarkable and irreversible changes
that have come upon the English language since the Anglo-Saxon period,
it has not yet reached a point of perfection and stability
such as we sometimes associate with Latin of the Golden Age,
the language of Virgil, Horace, and others.

— **DR. ROBERT BURCHFIELD** 'The English Language'
in The Oxford Guide to the English Language (1984)

What a patch-work has been our old saxon,
by the bitter frost that nipped its early budding,
and the constant habit of borrowing thence resulting,
the learned among us—as well as the unlearned,
though in very different ways—are constantly made to feel.
The English language as we have it now
is not so much a coherent growth as a disturbed organism.

— **PROF. JOHN STUART BLACKIE** Gaelic Self Taught (1910)

CONTENTS

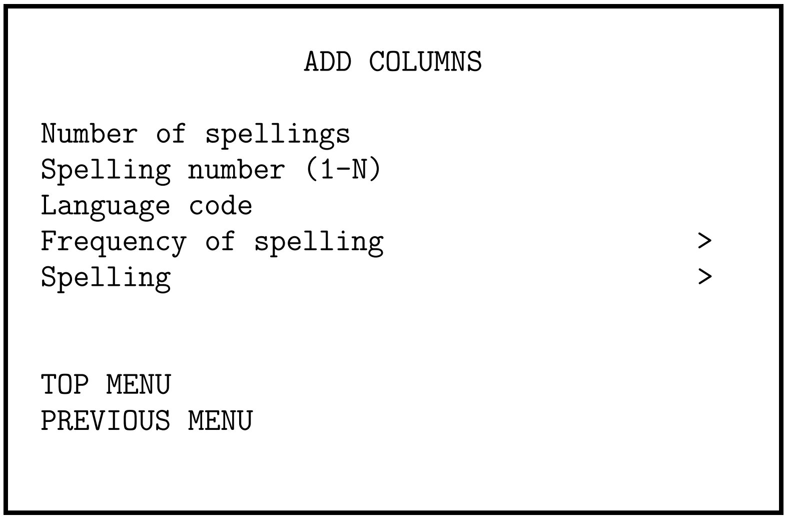
4 ENGLISH SYNTAX

5 ENGLISH FREQUENCY

1 ENGLISH ORTHOGRAPHY

Detailed and varied information is available on the orthographic forms of headwords and wordforms. 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)
Language code
Frequency of spelling                     >
Spelling                                  >


TOP MENU
PREVIOUS MENU
```

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 verb lemma *generalize*, this column has the value 2, which means there are two possible ways of spelling it. Unless you construct a restriction on your lexicon, *generalize* will occur twice: one row using the form *generalize*, the other using the form *generalise*.

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 an 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 or wordform, (according to the type of lexicon you are using), so this column uniquely identifies every spelling to be found for each lemma or wordform.

If you are using a lemma lexicon, the spelling variants are given in the form of headwords (with or without syllable markers). For example, the verb *generalize* has two spellings: one is the form *generalize*, and another is the form *generalise*. 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 *gen-er-al-ize* and spelling 2 the form *gen-er-al-ise*.

This means you can use the universal sequence number to identify a particular lemma (or wordform, depending on the type of lexicon you are using) and then the spelling number to identify the different individual spellings used for each lemma. Usually, no preference is indicated by the spelling numbers each variant has; *generalize* is as valid as *generalise*. However, the number 1 spelling is always an acceptable British from, and any American variants are always given a higher number. Thus the lemma *monologue* has the British form *monologue* as its number 1 spelling and the American form *monolog* as its number 2 spelling. (You can find out whether a spelling is British or American by using the ***OrthoStatus*** column described below.)

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 wordform), 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 *generalize* 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 headword. Without the restriction, FLEX returns two rows for *generalize*:

Spelling Number	**Headword**	**Pronunciation**
1	generalize	dZ.E.n.@.r.@.l.aI.z.
2	generalise	dZ.E.n.@.r.@.l.aI.z.
1	generalize	dZ.E.n.r.@.l.aI.z.
2	generalise	dZ.E.n.r.@.l.aI.z.

This is unnecessary, since you are interested only in the possible pronunciations. The extra row merely gives you a spelling variant while the pronunciations remain the same. When you include the restriction `OrthoNum␣=␣1`, however, only the rows with the number 1 spelling are included:

Spelling Number	**Headword**	**Pronunciation**
1	generalize	dZ.E.n.@.r.@.l.aI.z.
1	generalize	dZ.E.n.r.@.l.aI.z.

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 LANGUAGE CODE

For every different spelling there is a code which tells you whether it is an acceptable British form (B), or whether it is only ever an American form (A). This applies to lemmas and wordforms. A British spelling is always the first one given; that is, its spelling number is always `1`, while one which only occurs in American English is never the first form, and always has a spelling number greater than 1.

Spelling type	Status code	Example
British	B	*adviser*
American	A	*advisor*

Table 1: Orthographic status codes for English spellings

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

OrthoStatus `Status␣of␣spelling␣`
(OrthoStatusLemma)

1.4 FREQUENCY OF SPELLING

There are figures available which tell you how frequently each spelling of each lemma or wordform occurs in the COBUILD 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 for each wordform.

To arrive at these figures, a count has to be made of the number of times each string occurs in the current 17.9 million word version of the COBUILD corpus. This lets you see that the string *beauty* (for example) occurs 980 times, and that *truth* occurs 2279 times, and these figures are the spelling frequencies for the wordforms which are spelt that way.

Usually (but not always) this string frequency is the same as the wordform frequency, and it's then possible to formulate spelling frequencies for each lemma. This simply means adding together the frequencies for each wordform in each inflectional paradigm. Thus the spelling frequency for the lemma *beauty* is 63 (the spelling frequency for the wordform *beauties*) plus 980 (the spelling frequency for the wordform *beauty*), giving a total of 1043. Similarly the spelling frequency for the lemma *truth* is 126 (*truths*) plus 2279 (*truth*), giving a total of 2405.

On the occasions when the string frequency cannot be linked to just one lemma, an alternative plan of action is used. Take as an example the spelling *tender*, which can refer to four different lemmas: the first is the adjective meaning *soft* or *gentle*, the second is the verb meaning *offer* or *present*, the third is a noun meaning *financial estimate or proposition*, and the fourth meaning the *wagon* which comes behind a steam engine. The problem is that the string frequency can be assigned to any of these lemmas; it is always ambiguous. To overcome this problem, it is possible to check every occurrence of *tender* in the corpus and work out exactly how many belong to each of the four lemmas. 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. The string *tender* occurs 358 times, and after examining 100 occurrences of the word in the COBUILD corpus, 95 out of the hundred were *gentle*, 3 were the verb *offer*, 1 was the noun *financial proposition*, and 1 was the noun *wagon*. The ratio of one meaning to the other is thus $0.95 : 0.03 : 0.01 : 0.01$. The frequency of *tender* (that is, the adjective *gentle*) is then 358 multiplied by 0.95 – which is 340, while the frequency of the verb *offer*) is 358 multiplied by 0.03, which is 11. The noun *financial proposition* and the noun *wagon* share the frequency of 358 multiplied by 0.01, which rounded up comes to four.

However, the story is still not complete. Occasionally it is impossible to decide which lemma a particular spelling belongs to. The contraction *I'll* is one such word, since it can mean *I will* or *I shall*. 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:

CobSpellFreq
(CobSpellFreqLemma)

`Spelling␣frequency,␣COBUILD␣17.9m␣word␣corpus␣`

How accurate are the figures in the ***CobSpellFreq*** 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 *tender* (the adjective *gentle*), N is 358, n is 100, and p is 0.95, and the formula gives 13 as the deviation. This means that the true frequency for this form of *tender* is almost certain—95% certain at least—to lie between 327 and 353 .

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 *I'll* discussed above. While the frequency figures in such cases are arbitrary, the accompanying deviation figures are 100% accurate.

So while ***CobSpellFreq*** 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:

CobSpellDev `95%␣confidence␣deviation,␣COBUILD␣17.9m␣word␣`
(CobSpellDevLemma) `corpus␣`

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 ‘English Frequency’.

1.5 SPELLING

Before defining the specific spelling columns available with both of the English 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. Usually, only foreign words use such markers consistently in English – words like *vicuña* or *soupçon* or *débâcle*. 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 English 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: *à, â, ç, è, é, ê, ï, ô, ñ*, and *ü*. When you reach the `MODIFY␣CONVERSION␣`window, first select a column which contains transcriptions with diacritics, then type in the following string:

```
([\x20-\x7F]+␣
|\xE0%\x85|\xE2%\x83␣|\xE7%\x87|\xE8%\x8A␣
|\xE9%\x82|\xEA%\x88␣|\xEF%\x8B|\xF4%\x93␣
|\xF1%\xA4|\xFC%\x81)*␣
```

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 in an expression restriction or query.) 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 `\xF1␣`actually means `ñ`. In the MS-DOS coding set, the same `ñ␣`character is represented by the code `\xA4`. So to tell the pattern matcher to convert from a DIGITAL `ñ␣` to an MS-DOS `ñ`, you must type `\xF1%\xA4`.

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 *back* is given as *kcab*. 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 there is a short subsection dealing with corpus type transcriptions, then a longer subsection on the headword transcriptions available with a lemma lexicon, finishing up with a subsection on wordform transcriptions.

1.6.1 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 `Graphemic␣transcription␣`

1.6.2 TRANSCRIPTIONS FOR ENGLISH LEMMAS

The English lemma is always represented by the headword (as described in the *Introduction*, section 2.5). When you choose a column which contains orthographic transcriptions of headwords, it is as if you are choosing the bold-type headword in a dictionary. All the other columns in the database contain information specific to individual headwords, so the main function of the orthographic transcription is to identify any other information you look up – looking at a list of lemma frequency figures isn't meaningful unless you can see the lemmas they refer to. However, you may not always need to see the orthographic form of the headword: if you're looking for phonetic transcriptions with certain interesting syllable-final characteristics, say, you may not be interested in the orthographic headword – in which case you needn't keep it *on view*, and you might even want to miss it out of your lexicon altogether.

Described below are several different forms of orthographic transcription, and each form is assigned its own column. The first distinction you can make between them is whether or not syllable makers are included. Thereafter you can choose between back-to-front transcriptions, transcriptions with (or without) diacritics, transcriptions which consist only of lower case characters, and even transcriptions with the letters of the headword re-ordered alphabetically. Read the details, and then choose the columns which best help you to build useful lexicons.

1.6.2.1 SPELLINGS FOR ENGLISH HEADWORDS

There are six columns available which do not give any indication of the orthographic syllabification; they just deal with the letters in each headword.

```
                 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 has 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. So, the headword which represents the verbal family of inflections *walk, walks, walking* and *walked* is, quite simply, *walk*. (For information about which forms are used as headwords, see the *Introduction* section 2.5) The FLEX name and description of this column is as follows:

Head (HeadLemma) `Headword,␣without␣diacritics␣`

The second column contains the same transcriptions as the first, only the order of the letters is *reversed*. Thus the headword *walk* is given as *klaw*, and *increase* is given as *esaercni*. (However, with words like *repaper* you might not notice much of a difference. And proceed cautiously with the word *embargo*: the reverse transcription is *ograbme*.) The FLEX name and description of this column are as follows:

HeadRev `Headword,␣reversed␣`
(HeadRevLemma)

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. So, while the first column gives the plain form *cloissone*, this column includes the authentic French acute accent: *cloissoné*. Likewise *debacle* becomes *débâcle* and *deshabille* becomes *déshabillé*. The characteristics of diacritics are described in section 1.5.1 above. The FLEX name and description of this column are as follows:

HeadDia `Headword,␣diacritics␣`
(HeadDiaLemma)

The fourth column contains the same basic transcription as the first except that any upper case letters which occur are reduced to lower case, and any non-alphabetic characters are removed. Thus *Jeremiah* becomes *jeremiah* and *Sax* becomes *sax*.

Such a column is useful when you're trying to sort a list of words into true alphabetical order, as opposed to ASCII order. Each letter, whether upper or lower case, has a different ASCII number, and computers usually sort and order letters on the basis of these numbers. Because lower case letters all have higher numbers than upper case ones, the results of a sort program aren't always what you expect. However with this column, since all the characters are lower case, the problem doesn't arise. So, to make a file which contains a true alphabetical list of plain headwords, make a lexicon which consists of the plain headwords column ***Head*** and this column, and when you EXPORT it, put a `1␣` against this purely lower case column, and in this way alphabetic normality will be restored. The FLEX name and description of this column are as follows:

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

The most important feature of the fifth and last column is that the letters which make up each headword are sorted into alphabetical order. (This does *not* refer to the dictionary-like alphabetical order of headwords listed in the database; it's to do with the letters *within each word*). And in addition, any upper case characters are reduced to lower case, and non-alphabetic characters are removed. Thus, for example, *Jeremiah* becomes *aeehijmr*, and *dread* (perhaps confusingly) becomes *adder*. 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:

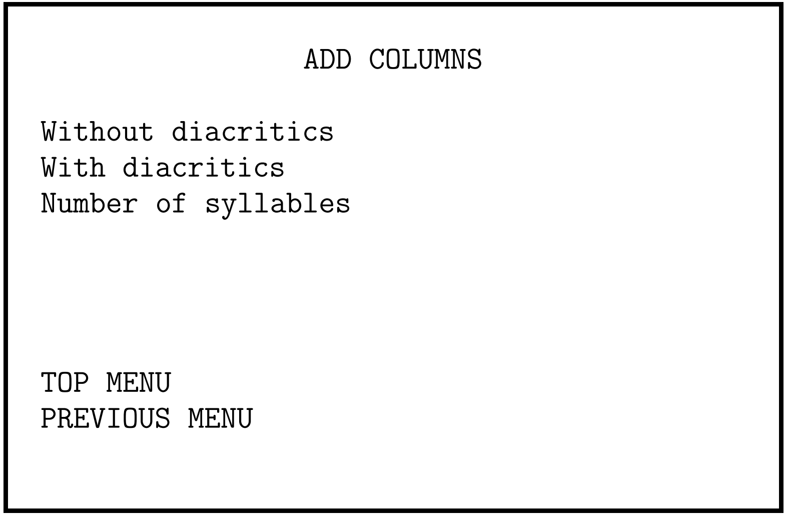
HeadLowSort `Headword,␣lowercase,␣alphabetical,␣sorted␣`
(HeadLowSortLemma)

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 *fo'c'sle* for example is 6. The FLEX name and description of this column are as follows:

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

1.6.2.2 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 *abandonment* is given as *a-ban-don-ment*. 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 *abandonment* would be *aban-don-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 *T-shirt*, 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-ban-don-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 TRANSCRIPTIONS FOR WORDFORMS

Wordforms are the words we use in everyday speech and writing. Elsewhere in the database, families of wordforms (inflectional paradigms) are represented by one form, the lemma. When you work with a wordform lexicon, all the wordforms are available as separate entries, not just one representative form. All the other columns in the database contain information specific to individual wordforms, so the main function of the orthographic transcription is to identify any other information you look up – looking at a list of syntactic class codes isn't very meaningful unless you can see the wordforms they refer to. 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'. Orthographic transcriptions of wordforms are available either with or without syllable markers. The next section deals with plain (unsyllabified) transcriptions, and the one after that deals with syllabified transcriptions.

1.6.3.1 SPELLINGS FOR PLAIN WORDFORMS

Described below are several different forms of orthographic transcription, and each form is assigned its own column. using the `ADD␣COLUMNS␣`menu shown below, you can choose between back-to-front transcriptions, transcriptions with (or without) diacritics, transcriptions which consist only of lower case characters, and even transcriptions with the letters of each wordform re-ordered alphabetically. Read the details below, and then choose the columns which best help you to build useful lexicons.

```
                  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.

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`. Thus the wordform *walks* is given as *sklaw*, and *increased* is given as *desaercni*. (However, with wordforms like *deified* you might not notice a big difference. And proceed cautiously with the word *desserts*, because in this column it has to be *stressed*.) 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 fourth column contains the same basic transcription as the first except that any upper case letters which occur are reduced to lower case. Thus *Peking* becomes *peking* and *Uranus* becomes *uranus*. In addition, any non-alphabetic characters (hyphens, apostrophes) are removed. Such a column is useful when you're trying to sort a list of words into true alphabetical order, as opposed to ASCII order. Each letter, whether upper or lower case, has a different ASCII number, and computers usually sort and order letters on the basis of these numbers. Because lower case letters all have higher numbers than upper case ones, the results of a sort program aren't always what you expect. However with this column, since all the characters are lower case, the problem doesn't arise. So, to make a file which contains a true alphabetical list of plain wordforms, make a lexicon which consists of the plain wordforms column ***Word*** and this column, and when you EXPORT it, put a `1␣` against this purely lower case column, and in this way alphabetic normality will be restored. The FLEX name and description of this column are as follows:

WordLow `Word,␣lowercase,␣alphabetical␣`

The most important feature of the fifth and last column is that the letters which make up each wordform are sorted into alphabetical order. (This does *not* refer to the dictionary-like alphabetical order of wordforms listed in the database; it's to do with the letters *within each word*). And in addition, any upper case characters are reduced to lower case. Thus *Peking* is given as *egiknp*, and *Uranus* as *anrsuu*. (Another example is the word *editorials*, whose sorted form is *adeiilorst*. Interestingly enough, *adeiilorst* is also the sorted form of the word *idolatries*.) 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:

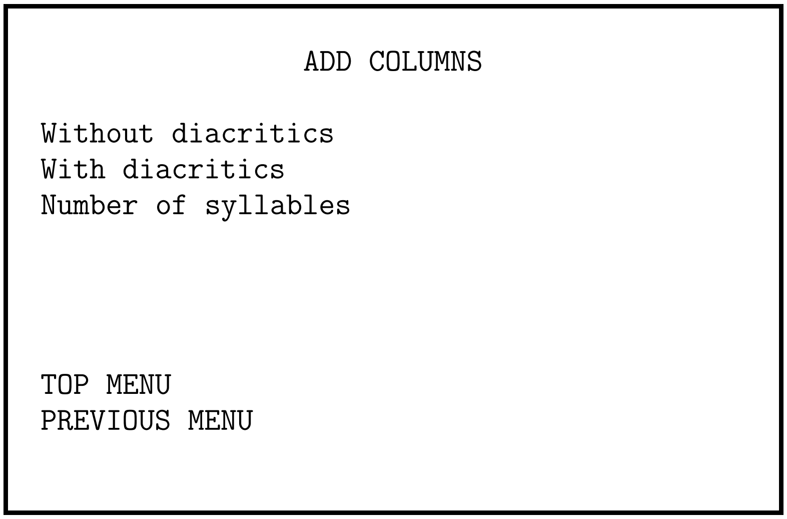
WordLowSort `Word,␣lowercase,␣alphabetical,␣sorted␣`

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 *shouldn't* for example is 8. The FLEX name and description of this column are as follows:

WordCnt `Word,␣number␣of␣letters␣`

1.6.3.2 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 *abandoning* is given as *a-ban-don-ing*. 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 (as explained in section 1.5.1) 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 *abandoning* is *aban-don-ing*. 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}␣
```

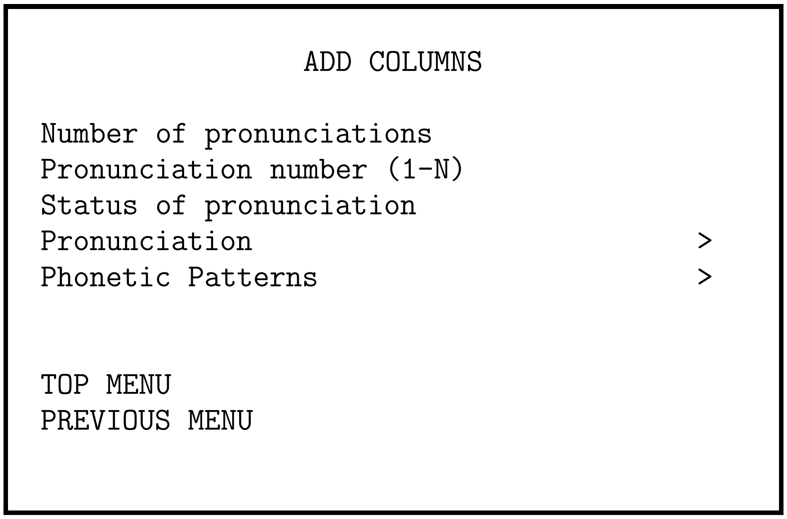
This basically means 'call the first character by the name *part1*. The second character may or may not be a hyphen. Any subsequent characters are called *part2*. Re-write the whole word as *part1* plus *part2*.' Only the parts of the word assigned to *part1* or *part2* are re-written, thus excluding 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.

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

WordSylCnt `Word,␣number␣of␣orthographic␣syllables␣`

2 ENGLISH PHONOLOGY

Phonetic transcriptions are available for each lemma and wordform in the database. They are specified in four different character sets, and variant pronunciations are also given whenever they occur. The transcriptions you choose can also include syllable markers or stress markers. Each pronunciation has a unique identification number, so that in conjunction with the lemma or wordform number, you can identify every single pronunciation in the database. Also available are 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 of the wordforms. The sections below deal with the information under the headings which correspond to those used in the ADD␣COLUMNS␣menus:



```
                    ADD COLUMNS

Number of pronunciations
Pronunciation number (1-N)
Status of pronunciation
Pronunciation                                   >
Phonetic Patterns                               >


TOP MENU
PREVIOUS MENU
```

Exactly the same columns are available for lemma lexicons and wordform lexicons, so all the phonetic column descriptions and definitions are equally valid for both lexicon types.

2.1 NUMBER OF PRONUNCIATIONS

This option in the ADD␣COLUMNS␣menu is a column which tells you how many ways each lemma or wordform (according to the type of lexicon you are using) can be pronounced. For the lemma *dexterous*, this column has the value 2, which means there are two possible ways of pronouncing it. Unless you construct a restriction on your lexicon, *dexterous* occurs twice: one row with `d.E.k.s.t.@.r.@.s.`, and another row with `d.E.k.s.t.r.@.s.␣`(these examples are from the ***PhonSAM*** column).

This column is particularly useful when you want to identify words which have pronunciation variants. To exclude from your lexicon all lemmas or wordforms which only have one possible pronunciation, containing instead those which can be pronounced in a number of ways, you can construct an expression restriction which simply states that the number of pronunciations must be greater than 1: `PhonCnt␣>␣1`.

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

PronCnt (PronCntLemma) `Number␣of␣Pronunciations␣`

2.2 PRONUNCIATION NUMBER

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

For example, the noun *dexterous* has two pronunciations: first `d.E.k.s.t.@.r.@.s.`, and second an alternative form `d.E.k.s.t.r.@.s.`. These have the pronunciation numbers 1 and 2 respectively. If you use a syllabified transcription in place of the plain transcription pronunciation 1 takes the form `dEk-st@-r@s␣`and pronunciation 2 the form `dEk-str@s`.

This means you can use the universal sequence number to identify a particular lemma or wordform and then the pronunciation number to identify the different individual pronunciations given for each lemma. Moreover, the pronunciation number allows you to identify quickly the ‘primary’ pronunciation (as laid down in the *English Pronouncing Dictionary* by Daniel Jones, A.C. Gimson and Susan Ramsaran) because such forms are always first in the list: for every lemma or wordform, the number 1 pronunciation is the ‘primary’ form. The classifications of variant pronunciations are dealt with fully in the next section on the 'status of pronunciation' column.

One important point to remember is that the pronunciation number can be used to eliminate unwanted rows from your lexicon. If you only want to see one pronunciation for each lemma (or whatever), you should make a restriction which states that only rows with a pronunciation number equal to 1 are to be included (in the form `PronNum␣=␣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 *dexterous* again, in a lexicon with three columns, one giving the pronunciation number, one giving the orthography of the headword, and one giving the pronunciation of the headword. Without the restriction, FLEX returns two rows for *dexterous*:

Pronunciation Number	**Headword**	**Pronunciation**
1	`dexterous`	`dEk-st@-r@s`
2	`dexterous`	`dEk-str@s`

You may find this unnecessary, needing to see only one 'default' pronunciation; the extra row merely gives you an extra pronunciation. When you include the restriction `PronNum␣ = 1`, however, only the row with the preferred spelling is included:

Pronunciation Number	**Headword**	**Pronunciation**
1	`dexterous`	`dEk-st@-r@s`

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

For some words, there are many variants given – *transitional* has forty possible pronunciations, all of which are given on a separate row in the database. If you are particularly interested in pronunciation variation of this kind, then do not add the `PronNum␣=␣1␣`restriction: that way you get to see all the variant forms. Otherwise, whenever you just want to use a simple phonetic transcription, always remember to insert it.

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

PronNum `Pronunciation␣ID␣number␣`
(PronNumLemma)

2.3 STATUS OF PRONUNCIATION

For every different pronunciation, there is a code which tells you whether it is a primary pronunciation (`P`), or a secondary pronunciation (`S`). This applies to both lemmas and wordforms. According to the *English Pronouncing Dictionary*, primary and secondary forms are all standard forms, but primary forms are heard more frequently.

The first pronunciation given for each word (that is, the pronunciation with ***PronNum*** 1) is always a *citation form*, the sort of pronunciation you are most likely to hear if you asked a speaker of 'standard' English to pronounce a particular word by itself. It is always given the status code `P␣` for 'primary' pronunciation. For example, the number one pronunciation of *transparent* is `tr{n-"sp{-r@nt`.

Sometimes *stylistic* variants of this first form are recorded, variants which indicate the highly frequent elision of sounds that occur in connected speech. Since the formal pronunciation of a word is quite rare, the stylistic variants are probably heard more often than the citation form. The second variant for *transparent* is `tr{n-"sp{-rn,t`, where the third syllable loses the schwa and the `n,␣`is a syllabic consonant. Frequent stylistic variants retain the status code `P␣`for 'primary pronunciation', but always have a pronunciation number which is greater than 1. Less frequent stylistic variants are considered secondary pronunciations, such as `tr@n-"sp{-r@nt␣`and `trn,-"sp{-r@nt`.

Of course there may be alternative, less frequent pronunciations of the primary citation form, alternatives which can't be attributed to the word being used in connected speech. Such differences (sometimes known as *speaker-to-speaker variations*) remain in all the stylistic variants each different citation form has. If you ask someone how to pronounce the word *transparent*, you might hear the primary form above, or instead a secondary form like `trA:n-"sp{-r@nt␣`or `tr{n-"spE@-r@nt`, the difference being the vowel used in the first and second syllables respectively. When the secondary form occurs in speech, it might be pronounced with some stylistic variation as `trA:n-"sp{-rn,t␣`or `tr@n-"spE@-r@nt`. No matter what the stylistic variations, the phoneme which distinguishes one citation form from another remains the same.

Secondary forms, whether citation forms or stylistic variants, are classified as such because they are thought to be used less commonly than primary forms. All secondary forms get the code `S`, and all have a pronunciation number greater than 1. Unlike primary forms, where the first form listed is automatically the citation form, secondary forms are not given in any sort of order. So, while the first secondary form you see listed in the database *might* be a citation form, it could just as well be one of many stylistic variants.

Pronunciation type	Status code	Pronunciation Number	Example *passenger*
Primary	P	1	"p{-sIn-dZ@r*
	P	2	"p{-sIn-Z@r*
Secondary	S	3	"p{-s@n-dZ@r*
	S	4	"p{-s@n-Z@r*
	S	5	"p{-sn,-dZ@r*
	S	6	"p{-sn,-Z@r*

Table 2: Pronunciation status codes for English

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

PronStatus
(PronStatusLemma)

`Status␣of␣pronunciation␣`

2.4 PHONETIC TRANSCRIPTIONS

When you begin selecting phonetic transcriptions from the CELEX databases, you have to choose from a wide array of options. First you must decide whether or not you want to use transcriptions which include syllable boundaries, and stress markers. Then you have to choose which of the four sets of computer phonetic codes best suits your task and your personal preferences. So, before defining each of the columns in turn, the section below describes the features of the four available sets of computer phonetic codes. In the last subsection, after the column definitions, there is a table of examples which gives some transcriptions from each of the transcription columns; it is especially useful for comparing the different types of transcription (plain, syllabified or stressed and syllabified).

2.4.1 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 tables over the next two pages list the basic set of segments for English. 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.

The first of the three IPA-like 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 SAM-PA implemented by CELEX has to include one change: the \␣character (ASCII code 92) representing the

IPA	example	SAM-PA	CELEX	CPA	DISC
p	**p**at	p	p	p	p
b	**b**ad	b	b	b	b
t	**t**ack	t	t	t	t
d	**d**ad	d	d	d	d
k	**c**ad	k	k	k	k
ɡ	**g**ame	g	g	g	g
ŋ	ba**ng**	N	N	N	N
m	**m**ad	m	m	m	m
n	**n**at	n	n	n	n
l	**l**ad	l	l	l	l
r	**r**at	r	r	r	r
f	**f**at	f	f	f	f
v	**v**at	v	v	v	v
θ	**th**in	T	T	T	T
ð	**th**en	D	D	D	D
s	**s**ap	s	s	s	s
z	**z**ap	z	z	z	z
ʃ	**sh**eep	S	S	S	S
ʒ	mea**s**ure	Z	Z	Z	Z
j	**y**ank	j	j	j	j
x	lo**ch**	x	x	x	x
h	**h**ad	h	h	h	h
w	**w**hy	w	w	w	w
ʧ	**ch**eap	tS	tS	T/	J
ʤ	**j**eep	dZ	dZ	J/	_
ŋ̩	baco**n**	N,	N,	N,	C
m̩	idealis**m**	m,	m,	m,	F
n̩	burde**n**	n,	n,	n,	H
l̩	dang**le**	l,	l,	l,	P
*	fathe**r**	r*	r*	r*	R
(possible linking ‘r’)					

Table 3: Computer phonetic codes for English consonants

‘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, 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

IPA	example	SAM-PA	CELEX	CPA	DISC
	p**i**t	`I`	`I`	`I`	`I`
	p**e**t	`E`	`E`	`E`	`E`
æ	p**a**t	`{`	`&`	`^/`	`{`
	p**u**tt	`V`	`V`	`^`	`V`
	p**o**t	`Q`	`O`	`O`	`Q`
ʊ	p**u**t	`U`	`U`	`U`	`U`
ə	**a**noth**er**	`@`	`@`	`@`	`@`
iː	b**ea**n	`i:`	`i:`	`i:`	`i`
ɑː	b**ar**n	`A:`	`A:`	`A:`	`#`
ɔː	b**or**n	`O:`	`O:`	`O:`	`$`
uː	b**oo**n	`u:`	`u:`	`u:`	`u`
ɜː	b**ur**n	`3:`	`3:`	`@:`	`3`
e	b**ay**	`eI`	`eI`	`e/`	`1`
a	b**uy**	`aI`	`aI`	`a/`	`2`
ɔ	b**oy**	`OI`	`OI`	`o/`	`4`
əʊ	n**o**	`@U`	`@U`	`O/`	`5`
aʊ	br**ow**	`aU`	`aU`	`A/`	`6`
ə	p**eer**	`I@`	`I@`	`I/`	`7`
ə	p**air**	`E@`	`E@`	`E/`	`8`
ʊə	p**oor**	`U@`	`U@`	`U/`	`9`
æ	t**i**mbre	`{~`	`&~`	`^/~`	`c`
ɑ̃ː	dét**en**te	`A~:`	`A~:`	`A~:`	`q`
æ̃ː	l**in**gerie	`{~:`	`&~:`	`^/~:`	`0`
˜ː	bouill**on**	`O~:`	`O~:`	`O~:`	`~`

Table 4: Computer phonetic codes for English vowels and diphthongs

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. One final point worth noting is that if instead of the standard sets of codes offered here you want to use a set of your own making, you can implement it by means of the pattern transducer available in the FLEX window `MODIFY␣CONVERSION`, and the DISC character set is probably the easiest set to convert.

2.4.2 PLAIN TRANSCRIPTIONS

The 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. 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. Also available with the plain transcriptions is a column which indicates how many phonemes each transcription contains.

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

PhonSAM
(PhonSAMLemma) `Unsyllabified,␣SAM-PA␣character␣set␣`

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

PhonCLX
(PhonCLXLemma) `Unsyllabified,␣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 is as follows:

PhonCPA
(PhonCPALemma) `Unsyllabified,␣CPA␣character␣set␣`

The last plain transcription column uses the DISC set. No delimiters, syllable markers or stress markers are included. The FLEX name and description of this column are as follows:

PhonDISC
(PhonDISCLemma) `Unsyllabified,␣DISC␣character␣set␣`

A count which tells you how many phonemes each headword you select contains is available. Since certain phonemes (long vowels and diphthongs) are sometimes given by two characters, this count is more sophisticated than merely the length of the string. Here are some examples: the lemma *farmhouse* has six phonemes in its transcription `['fa:m-haUs]`, and *ammeter* also has six `['&-mI-t@r*]`.

PhonCnt
(PhonCntLemma) `Number␣of␣phonemes␣`

2.4.3 SYLLABIFIED TRANSCRIPTIONS

The next set of transcriptions use the same basic transcriptions as the 'plain' set, but this time, they are given without any phoneme delimiters. Instead, the *syllables* which make up each word are shown in one of two ways. The first method is to use a hyphen (or, in the case of CPA, a full stop) to mark every syllable boundary within each word. The second method, available with the CELEX character set, is to enclose each syllable within square brackets.

The first syllabified transcription column uses the SAM-PA character set. It uses a hyphen as the syllable marker, and its FLEX name and description are as follows:

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

The next two syllabified transcription columns use the CELEX character set, and the first of these uses hyphens to mark every syllable boundary in each word. The FLEX name and description of this column are as follows:

PhonSylCLX
(PhonSylCLXLemma)
`Syllabified,␣CELEX␣character␣set␣`

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

PhonSylBCLX
(PhonSylBCLXLemma)
`Syllabified,␣CELEX␣character␣set␣(brackets)␣`

The fourth syllabified transcription column uses the CPA character set, and 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,␣CPA␣character␣set␣`

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

PhonSylDISC `Syllabified,␣DISC␣character␣set␣`
(PhonSylDISCLemma)

The last column in this set gives for each lemma or wordform a count which tells you the number of phonetic syllables in the word. For example, `['fa:m-haUs]␣`contains two syllables, and `['&-mI-t@r*]␣`contains three.

SylCnt `Number␣of␣phonetic␣syllables␣`
(SylCntLemma)

2.4.4 STRESSED AND SYLLABIFIED TRANSCRIPTIONS

The third set of columns gives transcriptions which are syllabified and also have primary and secondary stress markers. There are four such columns, each containing transcriptions in one of the four computer usable phonetic character sets described above.

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 (`"`) and points of secondary stress by means of the 'percent' character (`%`). These characters are placed immediately before a stressed syllable. The FLEX name and description of this column are as follows:

PhonStrsSAM `Syllabified,␣with␣stress␣marker,␣SAM-PA␣`
(PhonStrsSAMLemma) `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 (`'`) and the points of secondary stress with a 'double quote' (`"`).

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

The third column uses the CPA character set, including full stops to mark syllable boundaries, these transcriptions show points of primary stress with an inverted comma (') and the points of secondary stress with a 'double quote' (").

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

The fourth column uses the DISC character set, along with hyphens to mark syllable boundaries, these transcriptions show points of primary stress with an inverted comma (') and points of secondary stress with a 'double quote' (").

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

A *stress pattern* is available for each lemma or wordform, as well as a count of the number of syllables and phonemes each contains.

A *stress pattern* is a string which shows how each syllable is stressed in speech. Each syllable is represented by one numeric character: either `0`, `1␣`or `2`. `2␣`indicates that the syllable receives secondary stress, `1␣`indicates that it receives primary stress, and `0␣`that it does not receive primary or secondary stress. The examples below contrast the syllabified phonetic transcription which includes a primary stress marker with the stress pattern described here:

Example	Transcription	Stress pattern
biographic	`%baI-@U-"gr{-fIk`	`2010`
googly	`"gu:-glI`	`10`

Table 5: Example stress patterns

StrsPat (StrsPatLemma) `Stress␣pattern␣`

2.4.5 EXAMPLE TRANSCRIPTIONS

Column	Examples	
	excruciatingly	*oceanic*
Plain transcriptions		
PhonSAM	`I.k.s.k.r.u:.S.I.eI.t.I.N.l.I.`	`@U.S.I.{.n.I.k.`
PhonCLX	`I.k.s.k.r.u:.S.I.eI.t.I.N.l.I.`	`@U.S.I.&.n.I.k.`
PhonCPA	`I.k.s.k.r.u:.S.I.e/.t.I.N.l.I.`	`O/.S.I.^/.n.I.k.`
PhonDISC	`IkskruSI1tINlI`	`5SI{nIk`
Syllabified transcriptions		
PhonSylSAM	`Ik-skru:-SI-eI-tIN-lI`	`@U-SI-{-nIk`
PhonSylCLX	`Ik-skru:-SI-eI-tIN-lI`	`@U-SI-&-nIk`
PhonSylBCLX	`[Ik][skru:][SI][eI][tIN][lI]`	`[@U][SI][&][nIk]`
PhonSylCPA	`Ik.skru:.SI.e/.tIN.lI`	`O/.SI.^/.nIk`
PhonSylDISC	`Ik-skru-SI-1-tIN-lI`	`5-SI-{-nIk`
Syllabified and stressed transcriptions		
PhonStrsSAM	`Ik-"skru:-SI-eI-tIN-lI`	`%@U-SI-"{-nIk`
PhonStrsCLX	`Ik-'skru:-SI-eI-tIN-lI`	`"@U-SI-'&-nIk`
PhonStrsCPA	`Ik.'skru:.SI.e/.tIN.lI`	`"O/.SI.'^/.nIk`
PhonStrsDISC	`Ik-'skru-SI-1-tIN-lI`	`"5-SI-'{-nIk`

Table 6: Example English Phonetic Transcriptions

2.5 PHONETIC PATTERNS

Phonetic patterns here means CV patterns: the consonant and vowel patterns for the phonetic transcription (as opposed to the orthographic transcriptions) of any headword 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 patterns used here represent each *short vowel* as `V`, each *long vowel* and *diphthong* as `VV`, each *consonant* and *affricate* as `C`, and each syllabic consonant as `S`.

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

Example	Transcription	CV pattern	CV pattern with brackets
farmhouse	`'fa:m-haUs`	`CVVC-CVVC`	`[CVVC][CVVC]`
ammeter	`'&-mI-t@r*`	`V-CV-CVC`	`[V][CV][CVC]`

Table 7: Example CV patterns

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

PhonCV (PhonCVLemma) `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) `Phonetic␣CV␣pattern,␣with␣brackets␣`

3 ENGLISH MORPHOLOGY

Information on English Morphology 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 ENGLISH LEMMAS

The morphological analyses given for lemmas in the CELEX databases always use the *headword* form of the lemma, because this form (unlike Dutch) is usually the shortest in any inflectional paradigm, without any visible inflectional endings. However, when discussing English morphology, *stem* is the normal term used to describe this form, and so in this section *stem* is used instead of headword, just to fit in with common practice.

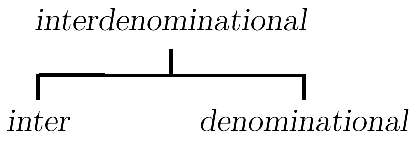
Before finding out details about each of the columns available, you should look at the sections below which try to give some explanation of the methods used to obtain the analyses given in the database. You will then know what CELEX means by terms such as *immediate segmentation*, *hierarchical segmentation, compound, derivation*, and *derivational compound*. 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

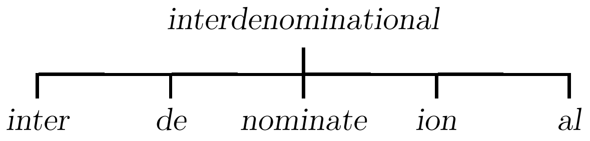
The first and most fundamental type of segmentation is *immediate segmentation*. This simply involves splitting a stem into its largest constituent parts. If you continue to carry out immediate segmentation until there is nothing left to segment, you arrive at the stem's *complete segmentation*. Depending 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, while the hierarchical segmentation shows the working.

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

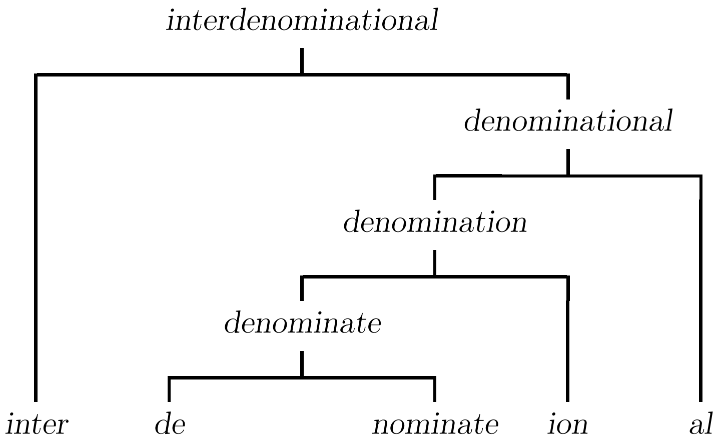
The first type of analysis 'Immediate segmentation' gives the affix *inter* plus the stem *denominational*:



The second type of analysis 'complete segmentation (flat)' shows you what you get if you keep applying immediate segmentation, namely the constituent morphemes of *interdenominational*: the affix *inter* plus the affix *de* plus the stem *nominate* plus the affix *ion* plus the affix *al*.



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.

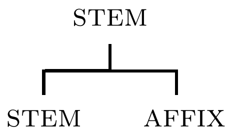
3.1.2 TYPES OF ANALYSES

When you attempt to split a stem into its biggest component parts, the result is always some combination of *stems* or *flections* and *affixes*. A flection—such as the *freezing* in *freezing point*—is treated the same as a stem, so that whenever an analysis involves a stem, you know that the stem could also be a flection. The most straightforward analysis 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 or affix plus at least one affix or one other stem, and can be termed either a *Derivation*, a *Compound*, a *Derivational Compound*, or a *Neo-classical Compound*. It is important to understand the differences between these four 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.

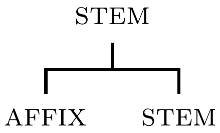
3.1.2.1 THE DERIVATION

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

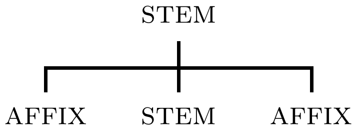
(i) a binary split into a stem or flection plus an affix (the word *careful* for example: *care + ful*).



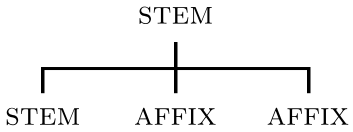
(ii) a binary split into an affix plus a stem or flection. For example, the word *barometer* is analysed as *baro* + *meter*.



(iii) a triform split into an affix, a stem or flection, and an affix (the word *extracurricular* for example: *extra* + *curriculum* + *ar*).



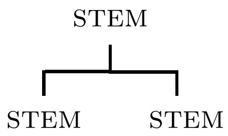
(iv) a triform split into a stem or flection, an affix and another affix. Such words can be derivations of inflected forms – the word *falteringly*, for example, is analysed as *falter* + *ing* + *ly*, which is a stem plus an inflectional affix plus a derivational affix. Alternatively, they can be lexicalised forms of inflected derivations like *countrified*, analysed as *country* + *ify* + *ed*, which is a stem plus a derivational affix plus an inflectional affix. This sort of analysis is only appropriate when the stem and the affix which immediately follows it don't together form a lemma, because otherwise—as with the word *inflationary*—the immediate analysis would be like type (i) above, a stem plus an affix (*inflation* + *ary*).



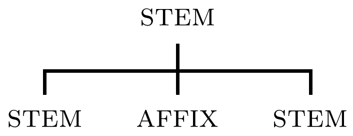
3.1.2.2 THE COMPOUND

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

(i) a binary split into two stems (the word *nameplate* for example: *name* + *plate*).



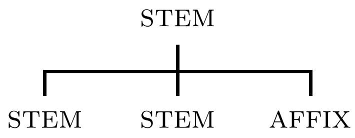
(ii) a triform split into a stem or flection, an affix (simply a ‘link’ morpheme), and a stem or flection (the word *bandsman* for example: *band* + *s* + *man*).



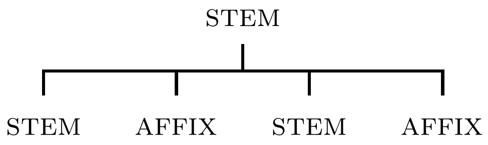
Words which consist of more than two stems aren’t analysed as compounds, since they normally have the structure of a phrase or sentence. So headwords like *nevertheless*, *Australian Rules football* and *be-all-and-end-all* don’t get a morphological analysis.

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 normally takes the form of a triform split into a stem or flection, another stem or flection, and an affix (the word *icebreaker* for example: *ice* + *break* + *er*).

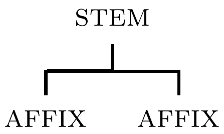


A couple of words can be analysed as a quaternary split into a stem, an affix, a stem, and an affix (the word *brinksmanship*, for example, is analysed as *brink* + *s* + *man* + *ship* and *whippersnapper* is analysed as *whip* + *er* + *snap* + *er*). However this is a very rare form of analysis.



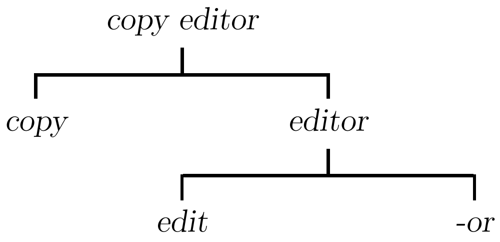
3.1.2.4 THE NEO-CLASSICAL COMPOUND

A neo-classical compound is a word which appears to be made up of two affixes, neither of which can occur as a word in its own right, like *aerodrome* (*aero* + *drome*) or *neurology* (*neuro* + *ology*). The affixes which combine to form this type of compound are generally known as *combining forms*.

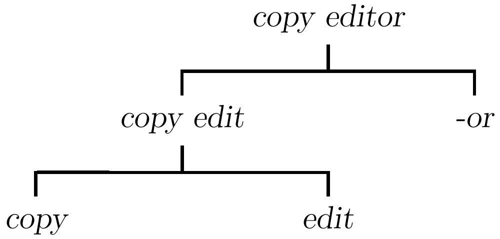


3.1.2.5 THE NOUN-VERB-AFFIX COMPOUND

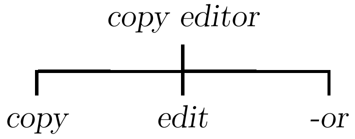
Problems sometimes arise with the analysis of words which look like derivational compounds. The general definition of a derivational compound is normally sufficient, but when the second stem is a verbal form, things become more complicated. A stem which comprises a noun plus a verb plus an affix can normally be considered a derivational compound, but some people may want to treat it as an ordinary compound or derivation. 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 *copy editor* is such a 'problem' compound. If you consider it to be an ordinary compound (the stem *copy* plus the stem *editor)*, its complete hierarchical tree looks like this:



If you consider it to be an ordinary derivation (the stem *copy-edit* plus the affix *-or)*, 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 *copy* plus the stem *edit* plus the affix *-or*, which gives the full hierarchical tree a different appearance:



3.1.3 HOW TO ASSIGN AN ANALYSIS

When you're faced with a headword that needs to be analysed, how do you work out the correct analysis? How did the people at CELEX who carried out the morphological analysis by hand arrive at the answers contained in the database? In particular, in the case of noun-plus-verb-plus-affix words, how did they decide which of the analysis types discussed in the previous section were appropriate?

To illustrate the principles used in analysing the information, there are two diagrams, given as Tables 8 and 9 below. The first illustrates the general strategy adopted for each headword, and the second deals with the special problems that arise with noun-plus-verb-plus-affix words. In both diagrams abbreviations are used: S means *stem*, and A means *affix*, making it easy to refer back to the sections above which define derivations, compounds, and derivational compounds in terms of stems and affixes. When an analysis is *acceptable*, it means that the component parts identified are current stems or affixes, and that the word can be defined as a derivation, a compound, or a derivational compound according to the definitions given in sections 3.1.2.1–3.1.2.3 above. An acceptable stem is one which appears in the *Collins English Dictionary* without being marked as ‘obsolete’ or ‘archaic’.

Following the first diagram, analysis starts with an attempt to see if the word under scrutiny is just the same as an already existing word with a different word class. The word *railroad*, for example, can be used as a verb, and it is said to come from the corresponding noun *railroad*. This phenomenon is called *conversion* or *zero derivation*, since there is no difference in the form of the two words even though they have a different word class. Conversion is explained in full under section 3.1.4 ‘Status and Language codes’. If conversion has occurred, the analysis need go no further: in ***MorphStatus*** the word gets the code Z, and ***NVAffComp*** and its subordinate columns ***Der, Comp,*** and ***DerComp*** are all set to N.

If the word is not a conversion, then the next step is to check whether it fits with the definition of a *derivation* given in section 3.1.2.1 above. For example, the word *calculator* is analysed as the stem *calculate* with the suffix *-or*, *encircle* is analysed as the prefix *en-* with the stem *circle*, and *unflappable* as the prefix *un-* plus the stem *flap* plus the suffix *-able*. In all three cases, the word is classified as a *derivation*: in ***MorphStatus*** the word gets the code C␣to indicate that it is *complex*, and ***NVAffComp*** and its subordinate columns ***Der, Comp,*** and ***DerComp*** are all set to N.

If the word turns out not to be a derivation, then the next stage is to see if it fits with the definition of a *compound* given

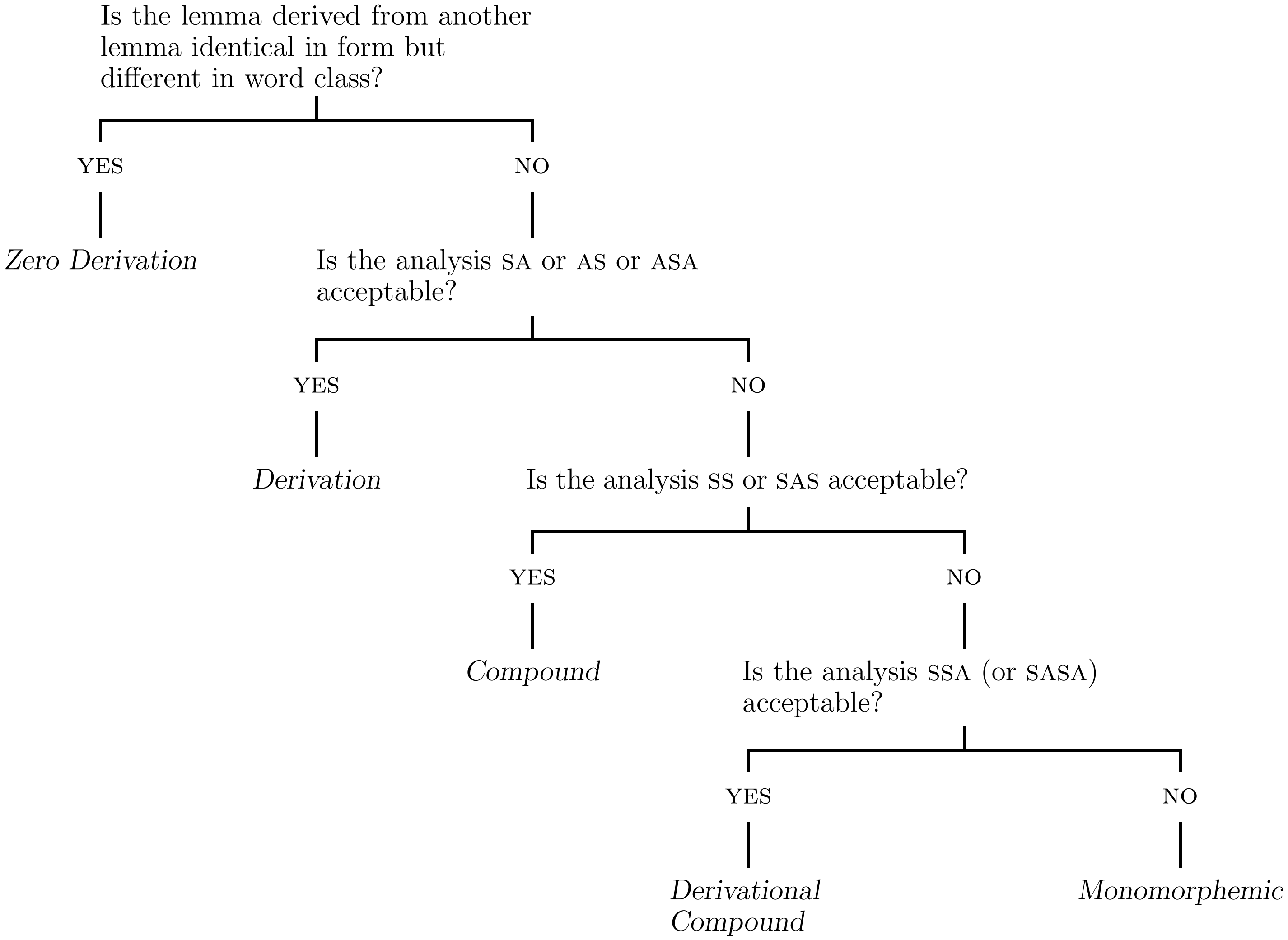

Is the lemma derived from another lemma identical in form but different in word class?
YES
NO
Zero Derivation
Is the analysis SA or AS or ASA acceptable?
YES
NO
Derivation
Is the analysis SS or SAS acceptable?
YES
NO
Compound
Is the analysis SSA (or SASA) acceptable?
YES
NO
Derivational Compound
Monomorphemic

Table 8: How to carry out morphological analysis

in section 3.1.2.2 above. For example, the noun *keyboard* is analysed as the stem *key* plus the stem *board*, and *groundsman* as the stem *ground* plus the infix -*s*- plus the stem *man*. In both cases, the word is classified as a *compound*: under the ***MorphStatus*** column it gets the code `C␣`to indicate that it is *complex*, and ***NVAffComp*** and its subordinate columns ***Der, Comp,*** and ***DerComp*** are all set to `N`.

If the word still hasn't been classified, then the last stage is to see whether the word is a *derivational compound*, as defined in section 3.1.2.3 above. For example, the adjective *barefaced* is analysed as the stem *bare* plus the stem *face* plus the affix -ed. The word is therefore classified as a *derivational compound*: under ***MorphStatus*** it gets the code `C␣`to indicate that it is *complex*, and ***NVAffComp*** and its subordinate columns ***Der, Comp,*** and ***DerComp*** are all set to `N`.

It is possible that the word might not fit into any of the above categories: this makes the word *monomorphemic*, and the 'analysis' is simply the word itself – *chair*, for example, or *llama*. Under ***MorphStatus*** the code `M␣`is given, and ***NVAffComp*** and its subordinate columns ***Der, Comp,*** and ***DerComp*** are all set to `N`. In other cases where no analysis can be carried out, the code under ***MorphStatus*** indicates why. You can read about these codes in section 3.1.4 'Status and language codes'.

3.1.3.1 THE NOUN-VERB-AFFIX COMPOUND

The general scheme explained above is enough to arrive at an analysis in most cases. However, difficulties in applying the system arise when you start considering so-called *noun-verb-affix compounds* – those words which contain a verbal element, which aren't conversions, and which could be analysed as a nominal stem plus a verbal stem plus an affix. Examples of such words are *stockholder* and *copy-editor*. This type of compound is characterised by a `Y␣`in the ***NVAffComp*** column. As the diagram below shows, just because they *could* be analysed in such a way, it doesn't mean they necessarily *should* be. *Stockholder* is both a compound and a derivational compound, and *copy-editor* is a derivation, a compound and a derivational compound. The approach outlined below is designed to keep as many morphologists as possible happy with the information available in the database: it's possible to choose for yourself whether to restrict your lexicon to just one type of analysis, or to permit them all, according to your own requirements.

The first step is to see whether the ***NVAffComp*** word you are dealing with can be classified as a derivation, in accordance with the definition in section 3.1.2.1 above. Take the word *dive-bomber* as an example – it can be analysed as the stem *dive-bomb* plus the affix *-er*. The verb *dive-bomb* is accepted as legitimate because it occurs as such in the *Collins English Dictionary* (CED). So this word does meet the definition of a derivation, and thus gets the code `Y␣`in the ***Der*** column. Another example is the word *bricklayer*: since a verb *bricklay* doesn't exist (according to the CED) it can't be analysed as a stem plus an affix, and so it gets the code `N␣`in the ***Der*** column.

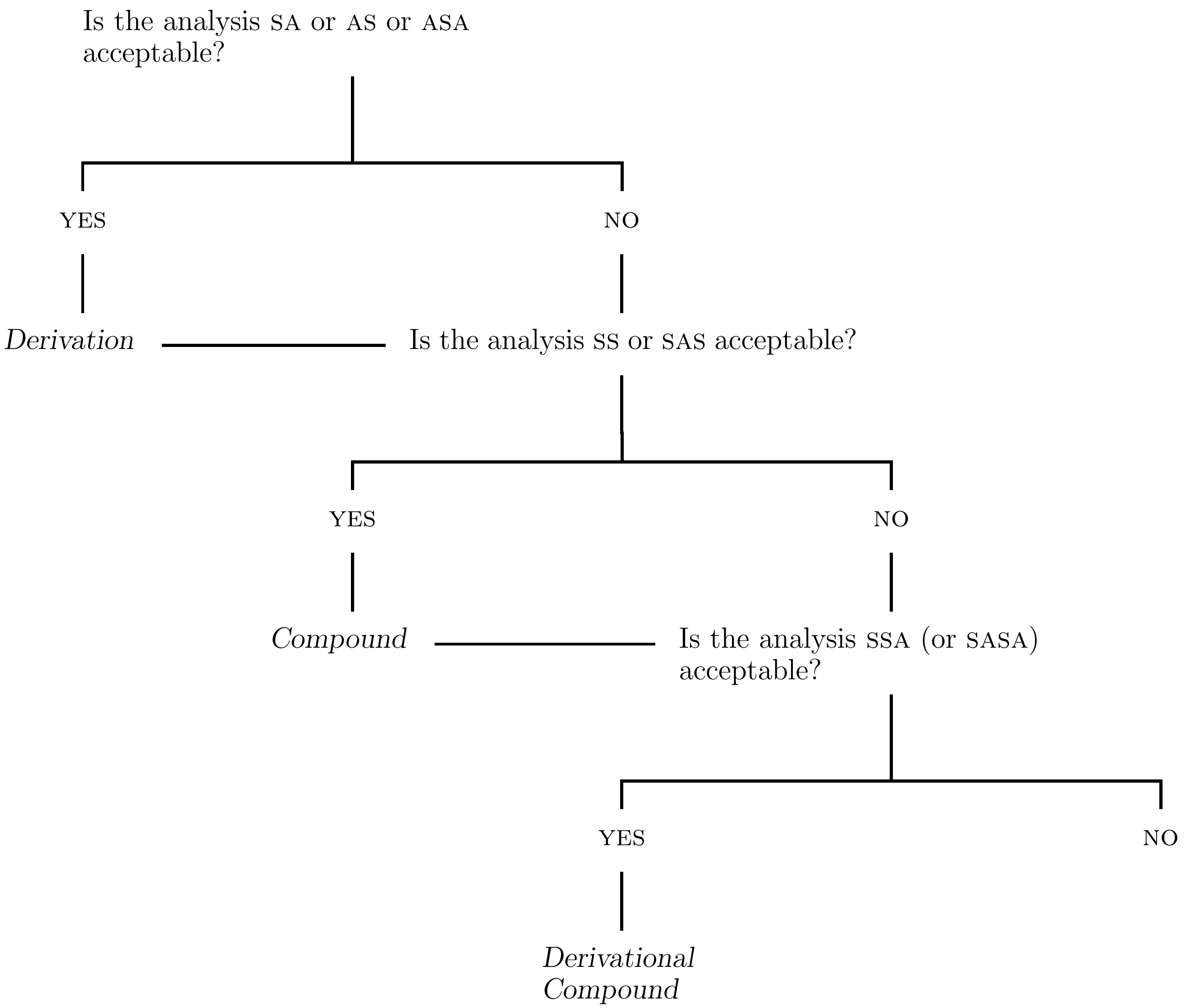

Is the analysis SA or AS or ASA acceptable?
YES
NO
Derivation
Is the analysis SS or SAS acceptable?
YES
NO
Compound
Is the analysis SSA (or SASA) acceptable?
YES
NO
Derivational Compound

Table 9: Dealing with noun-verb-affix compound analyses

The next stage is to see whether the ***NVAffComp*** word in question can be classified as a compound, as defined in section 3.1.2.2 above. Again, the word *dive-bomber* meets the definition: it can be analysed as the stem *dive* plus the stem *bomber* and it is a particular sort of bomber. It can therefore be called a compound, and it gets the code `Y␣`in the ***Comp*** column. Applying the same rules to *bricklayer* produces the opposite result: a *bricklayer* isn't really a particular sort of *layer*, since (according to the CED) *layer* doesn't mean 'someone who lays things'. So, *bricklayer* gets the code `N␣`in the ***Comp*** column to show that it isn't a compound.

The last stage is to decide whether the word is a derivational compound, as defined in section 3.1.2.3 above. This time *dive-bomber* does not qualify. Even though it can have the structure stem (*dive*) plus stem *bomb* plus affix (-*er*), the noun *dive* cannot be the object of the verb *bomb* – you can't talk about 'bombing a dive'. Since *dive* isn't any sort of obligatory complement to the verb *bomb*, *dive-bomber* is not a derivational compound, and therefore gets the code `N␣` in the ***DerComp*** column. Applying the rules to *bricklayer* again produces the opposite result. It can have the structure stem (*brick*) plus stem (*lay*) plus affix (-*er*), and *brick* is the object of the verb *lay*; it is possible to talk about 'laying bricks'. So since this time *bricks* is some sort of complement to the verb *lay*, *bricklayer* gets the code `Y␣`in ***DerComp*** column to show that it is a derivational compound.

To illustrate this process further, more examples are given in the table below.

Word	Classifications			
	NVAffComp	***Der***	***Comp***	***DerComp***
typesetter	Y	Y	N	Y
dive-bomber	Y	Y	Y	N
copy-editor	Y	Y	Y	Y
stockholder	Y	N	Y	Y
churchgoer	Y	N	N	Y
cub reporter	Y	N	Y	N

Table 10: Example noun-verb-affix compound analyses

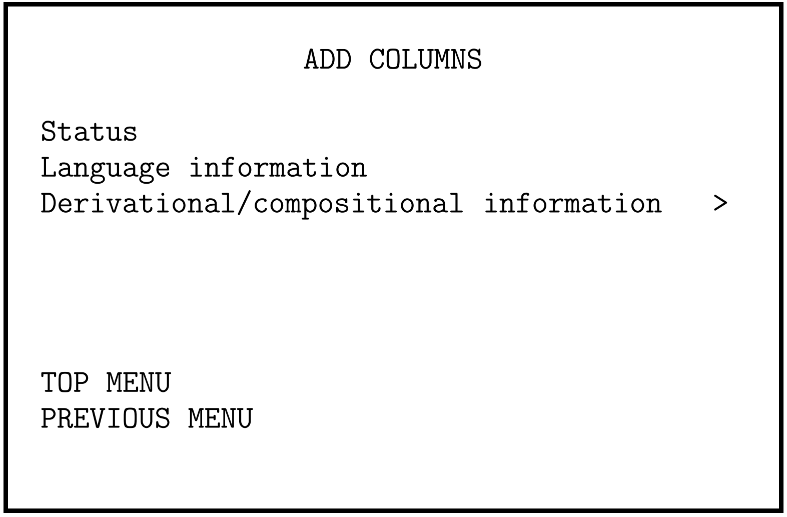
It shows six examples and the codes each one gets in ***NVAffComp, Der, Comp***, and ***DerComp***, as a quick way of showing how the words are classified in the ***NVAffComp*** analysis scheme. In the database, however, each separate *analysis* gets a separate row, and if you looked up analyses for these six words you would get something like this:

Stem	MorphNum	NVAffComp	Der	Comp	DerComp	Def	Imm
typesetter	1	Y	Y	N	N	Y	typeset+er
typesetter	2	Y	N	N	Y	Y	type+set+er
dive-bomber	1	Y	Y	N	N	Y	dive-bomb+er
dive-bomber	2	Y	N	Y	N	Y	dive+bomber
copy-editor	1	Y	Y	N	N	Y	copy-edit+or
copy-editor	2	Y	N	Y	N	Y	copy+editor
copy-editor	3	Y	N	N	Y	Y	copy+edit+or
stockholder	1	Y	N	Y	N	Y	stock+holder
stockholder	2	Y	N	N	Y	Y	stock+hold+er
churchgoer	1	Y	N	N	Y	Y	church+go+er
cub reporter	1	Y	N	Y	N	Y	cub+reporter

If you've followed in full the explanation of how CELEX carried out its morphological analysis of English words, most of this example lexicon should be clear. The columns it contains, along with other columns are described and defined in the sections that follow. Using the columns available 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 derivation, 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.

3.1.4 STATUS AND LANGUAGE CODES

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



```
                    ADD COLUMNS

 Status
 Language information
 Derivational/compositional information   >





 TOP MENU
 PREVIOUS MENU
```

Before dealing with the various derivational/compositional information columns, which form the bulk of the available morphological information, the first two columns are dealt with here.

The first column simply tells you by means of a single code whether each stem is morphologically simple, morphologically complex, or a conversion, or why it is as yet unanalysed. The table below shows the codes that are used, and it is followed by a description of each of the eight codes. Just before the description concludes with the column definition, there is a diagram which illustrates the strategies CELEX used to determine a status code for each stem.

Status	Code	Example
Morphological analysis available:		
Morphologically complex	`C`	*sandbank*
Monomorphemic	`M`	*camel*
Conversion (Zero Derivation)	`Z`	*abandon*
Contracted form	`F`	*I've*
Morphological analysis unavailable:		
Morphology irrelevant	`I`	*meow*
Morphology obscure	`O`	*dedicate*
Morphology may include a ‘root’	`R`	*imprimatur*
Morphology undetermined	`U`	*hinterland*

Table 11: 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.

If a stem appears to be derived from another stem which is identical in form but different in word class, it gets the code `Z␣` for 'zero derivation' or conversion. The noun *delinquent*, for example, can be said to derive from the adjective *delinquent*. Normally derivations from one word class to another are clearly marked by means of an affix – *sheepish* is an adjective derived from the noun *sheep*, for example. But conversions, on the other hand, are not so marked: it's as if an affix containing nothing had been added to the original stem.

Naturally enough, when conversion occurs, it's not immediately obvious which stem is the original and which is the derivative. In analysing these words, CELEX adopted a strategy for determining the *direction* of the conversion (that is, if a verb has been converted into a noun, the derivation is *in the direction* of the noun). Table 12 indicates the normal direction of conversion. Conversion in the opposite direction is also possible provided that it is specified in the *Shorter Oxford English Dictionary* (SOED).

Default direction	Example
VERB—NOUN	*paint*
ADJECTIVE—NOUN	*parallel*
ADJECTIVE—ADVERB	*pretty*
ADJECTIVE—VERB	*pale*
PREPOSITION—ADVERB	*past*

Table 12: Direction of conversions

The status code `F␣` indicates that the 'analysis' given is in fact a contraction. The single contraction *'d* can represent *had*, *would* or *did*, and the complex contraction *he's* represents *he* *is* or *he has*. Each contracted form gets its own row in the database and the status code F.

In the case of monomorphemic stems, complex stems, conversion stems, and contractions of stems, morphological analyses are provided in the various segmentation columns. However there remains a large number of stems which have no analysis, and in such cases, codes indicating the reasons for the lack of analysis are given in this column, and these codes and reasons are explained below.

First of all, sometimes even attempting morphological analysis is not appropriate for a particular stem. Usually this is true when the stem is an exclamation or an interjection of some sort (*gosh*, *prithee* or *meow*, for example), or when it is a proper noun – *Spooner* and *Germany* aren't analysed. In addition, those few words which seem to have taken on the structure of a short sentence (or at least consist of three or more stems), like *nowadays* or *whodunit*, don't get an analysis. So, whenever 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.

Some stems are recognizable recent loanwords which have achieved some sort of currency in English – words like *virtuoso* or *pretzel* or *mazurka*. Since providing analyses for such stems would, in many cases, mean delving into the morphology of languages not covered by CELEX, they simply receive the code U␣for 'undetermined'. The languages loanwords originate from are shown in the next column, ***Lang***.

On other occasions, an analysis seems possible, but cannot be fully explained. The stem *tabby*, for instance, appears to consist of the productive suffix -*y* plus what might be another stem *tab*. However *tab* bears no immediate relation to the adjective *tabby*, so that *tabby* gets the code O␣to indicate that the morphological analysis is 'obscure'.

In most cases morphological analysis is carried out on a *synchronic* basis: the stems or affixes which make up a word must occur in modern, current English, regardless of the historical origins they might have. On many occasions, however, an etymological root could explain the morphology of a stem which would otherwise be unanalysable. The stem *patrimony*, for example, appears to be made up of a Latin prefix *patri-* and what may be a Latin suffix *-mony*. Stems like this, which could be analysed on the basis of the historical root of its constituent parts, are given the code `R␣`for 'root'.

If you want to understand this coding system more fully, then you can examine Table 13 , which is a diagram that sets out a scheme for arriving at appropriate morphological status codes. Starting with a stem whose morphology is relevant (that is, it doesn't belong with those stems that have the code `I`), you can work out the code it should have by following the diagram through. This strategy is the one actually used by CELEX to determine the correct codes.

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 `Morphological␣status␣`
(MorphStatusLemma)

The second column contains codes that identify the particular geographical origins of lemmas, including the foreign languages from which some of the lemmas have been borrowed. The headword *conquistador* thus gets the code `S␣`to indicate that it is of Spanish origin, and the verb *waffle* gets the code `B␣`because it's reckoned to be a peculiarly British usage. Whenever a lemma doesn't have a code, it isn't a recent borrowing from another language, and it doesn't have associations with one regional variety of English. Table 14 below sets out all the codes used and their meanings.

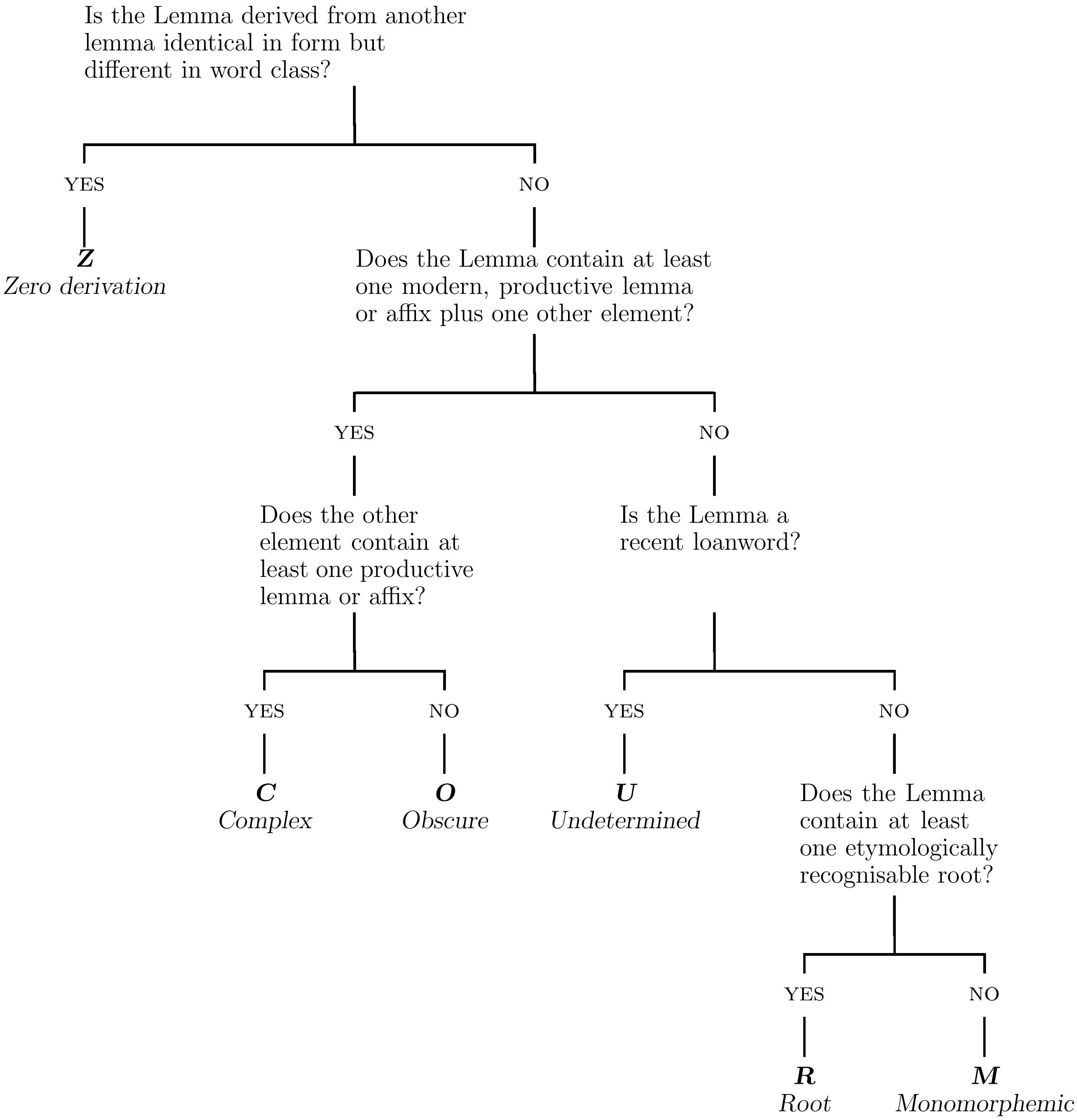

Is the Lemma derived from another lemma identical in form but different in word class?
YES
NO
Z
Zero derivation
Does the Lemma contain at least one modern, productive lemma or affix plus one other element?
YES
NO
Does the other element contain at least one productive lemma or affix?
Is the Lemma a recent loanword?
YES
NO
YES
NO
C
Complex
O
Obscure
U
Undetermined
Does the Lemma contain at least one etymologically recognisable root?
YES
NO
R
Root
M
Monomorphemic

*Table 13: How to assign **MorphStatus** codes*

Language Code	Meaning	Example
`A`	American English	*billfold*
`F`	French	*patisserie*
`B`	British English	*divvy*
`D`	German	*sauerkraut*
`G`	Greek	*eureka*
`I`	Italian	*cicerone*
`L`	Latin	*emeritus*
`S`	Spanish	*siesta*

Table 14: Language codes for English headwords

The FLEX name and description of the column which contains these codes is as follows:

Lang (LangLemma) `Language␣information␣`

3.2 DERIVATIONAL/COMPOSITIONAL INFORMATION

```
                   ADD COLUMNS

Number of morphological analyses
Analysis 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, *backfire* has one analysis, *flashbulb* has two, and *treasurer* 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 the adjective lemma *flashbulb* has two rows: one has the ***MorphNum*** 1, the other has the ***MorphNum*** 2 or a stem. description of this column are as follows:

MorphNum
(MorphNumLemma) `Morphological␣analysis␣number␣(0-N)␣`

3.2.1 ANALYSIS TYPE CODES

Under the ‘status of morphological analysis’ option there are five ‘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 noun-verb-affix compound. (A noun-verb-affix compound, as discussed in section 3.1.2.5 , can correctly be analysed as a derivation, a compound, or a derivational compound.) The five columns in question are called ***NVAffComp, Der, Comp, DerComp***, and ***Def***.

Whenever ***NVAffComp*** contains a `Y`, you know that ’yes, this row contains a stem which is considered a noun-verb-affix compound, and which therefore might be analysed in three different ways’. And naturally whenever it contains an `N`, you know that the row contains a stem which is *not* considered a noun-verb-affix compound. The FLEX name and description of this column are as follows:

NVAffComp
(NVAffCompLemma) `Noun-verb-affix␣compound␣`

Whenever ***Der*** contains a `Y`, you know that ‘Yes, this row contains a noun-verb-affix compound which is analysed as a derivation’. And whenever it contains an `N`, you know that the row contains a noun-verb-affix compound which is *not* analysed as a derivation or a stem which is not of the noun-verb-affix compound type. The FLEX name and description of this column are as follows:

Der
(DerLemma) `Derivation␣analysis␣`

Whenever ***Comp*** contains a `Y`, you know that ‘yes, this row contains a noun-verb-affix compound which is analysed as a compound’. And again, `N␣`means that the row contains a noun-verb-affix compound which *isn’t* analysed as a compound or a stem which is not of the noun-verb-affix compound type. The FLEX name and description of this column are as follows:

Comp
(CompLemma) `Compound␣analysis␣`

Likewise whenever ***DerComp*** contains a `Y`, you know that ‘yes, this row contains a noun-verb-affix compound which is analysed as a derivational compound’. And naturally, `N␣` means that the noun-verb-affix compound *isn’t* analysed as a derivational compound or that it is a stem which is not of the noun-verb-affix compound type. The FLEX name and description of this column are as follows:

DerComp
(DerCompLemma) `Derivational␣compound␣analysis␣`

If a stem has more than one analysis, it’s sometimes helpful to be able to identify one which is the best or most useful, or at least to discard unwanted alternatives. Whenever ***Def*** contains a `Y`, you know that 'yes, this row contains a default analysis', and when it contains an `N`, you know that the row contains another, non-default analysis.

Since there are three types of analyses which can be assigned to a complex stem, there might also be up to three default analyses for one word: a default derivation analysis, a default compound analysis, and a default derivational compound analysis. (Of course, not many words are eligible for three default analyses.) While morphological analysis was being carried out, rules were formulated to determine which analyses should take precedence over the others, and these rules are explained in Table 15 below.

The left-hand column gives the problem which those doing the analysis came up against, the central column shows which of the two possible analyses should be the default (or 'take precedence'), and the right-hand column illustrates the principle with an example. The first part of the table shows the preferential order for derivations, and the second part shows the preferential order for compounds. A part for derivational compounds isn't necessary since they are only analysed in one way.

If, despite the range of analyses available, you only want just *one* default analysis, then you can get it by making a restriction on ***MorphNum***: `MorphNum␣=␣1`. The first analysis for a lemma is always a default analysis. Analyses which are derivations take precedence over compounds, and likewise compounds take precedence over derivational compounds.

Using this column in conjunction with the three preceeding columns, you can construct restrictions which select or omit the analyses you specify. The FLEX name and description of this column are as follows:

Def (DefLemma) `Default␣analysis␣`

To illustrate how you can use these columns, imagine that you have chosen ***Imm*** and ***ImmClass*** as the form of morphological analysis you want to see. ***Imm*** shows the analysis, and ***ImmClass*** shows the word class of the analysed parts (these columns, and the other columns containing the same analyses in different forms, are described in the sections

Option	Solution	Example
Preferential order for the analysis of derivations:		
stem + affix or affix + stem	stem + affix takes precedence over affix + stem	*disavowal* is analysed first as *disavow* + *-al*, then as *dis-* + *avowal*.
verb ending in *-ate, -ete, -ote* or *-ute* + the affix *-ion* or verb not ending in *-ate, -ete, -ote* or *-ute* + an affix like *-tion*	The verb with the higher frequency in the COBUILD type list takes precedence.	*annunciation* is analysed first as the verb *announce* plus the affix *-iation*, and then as the verb *annunciate* plus the affix *-ion*.
adjective + suffix *-ly* or adjective + suffix *-ally*	The adjective with the higher frequency in the COBUILD type list takes precedence.	*problematically* is analysed first as *problematic* + *-ally*, and second as *problematical* plus *-ly*.
verb + suffix or noun + suffix	Verb takes precedence when the suffix is *-able, -er, -or* or *-ure*; noun takes precedence when the suffix is *-ery, -ism, -ist, -ous, -some* or *-y*.	*comfortable* is analysed first as the verb *comfort* plus the suffix *-able*, and second as the noun *comfort* and the suffix *-able*. *chatty* is analysed first as the noun *chat* plus the suffix *-y*, and second as the verb *chat* plus the suffix *-y*.
verb + suffix *-age* or noun + suffix *-age*	When the word denotes action or an instance of a phenomenon, then the verb takes precedence; when the word denotes a measure or collection of something, then the noun takes precedence.	*leakage* is analysed first as the verb *leak* + the suffix *-age*, and second as the noun *leak* + the affix *-age*.
prefix *a-* + noun or prefix *a-* + verb	Verb takes precedence.	*aglow* is analysed first as the prefix *a-* plus the verb *glow*, and second as prefix *a-* plus the noun *glow*.
Preferential order for the analysis of compounds:		
verb + noun or noun + noun	Verb + noun takes precedence.	*checkpoint* is analysed first as the verb *check* plus the noun *point*, and second as the noun *check* plus the noun *point*.
noun + noun or noun + verb	Noun + noun takes precedence.	*windfall* is analysed first as the noun *wind* plus the noun *fall*, and second as the noun *wind* plus the verb *fall*.

Table 15: How to order multiple analyses of compounds and derivations

following this one). Then say that you are interested in two stems *dive-bomber*, which has three different analyses, and *typesetter*, which has two. Both words are noun-verb-affix type words which may be derivations or compounds or derivational compounds, and this accounts for four of the analyses given. However, for the compound analysis of *dive-bomber*, the stem *dive* is analysed as a verb but can also be thought of as a noun, which gives an extra analysis.

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

If you want to see all possible segmentations, then you don't need to add extra restrictions. As the ***MorphCnt*** column indicates, there are three analyses given for *dive-bomber* and two for *typesetter*, so this is what the unrestricted example lexicon looks like:

Stem	**MorphNum**	**NVAffComp**	**Der**	**Comp**	**DerComp**	**Def**	**Imm**	**ImmClass**
`dive-bomber`	`1`	`Y`	`Y`	`N`	`N`	`Y`	`dive-bomb+er`	`Vx`
`dive-bomber`	`2`	`Y`	`N`	`Y`	`N`	`Y`	`dive+bomber`	`VN`
`dive-bomber`	`3`	`Y`	`N`	`Y`	`N`	`N`	`dive+bomber`	`NN`
`typesetter`	`1`	`Y`	`Y`	`N`	`N`	`Y`	`typeset+er`	`Vx`
`typesetter`	`2`	`Y`	`N`	`N`	`Y`	`Y`	`type+set+er`	`NVx`

Derivations take precedence over compounds, so for both words the first row, with analysis number 1, contains the derivation and gets `Y␣` under ***Der***. And since each word has only one possible derivation analysis, both are also default analyses, and therefore get `Y␣` under ***Def*** too. The `N␣`under ***Comp*** and ***DerComp*** confirm that they are not compounds or derivational compounds.

Compounds take precedence over derivational compounds, so for *dive-bomber* the next two rows contain the two compound analyses, with analysis numbers (***MorphNum***) 2 and 3. Both get the code `Y␣` under ***Comp***. Since verb + noun compounds take precedence over noun + noun compounds, the verb + noun analysis is a default analysis: it gets 2 as its ***MorphNum***, and `Y␣`under ***Def***. The noun + noun analysis gets 3 as its ***MorphNum***, and `N␣`under ***Def***. The `N␣`codes under ***Der*** and ***DerComp*** confirm that neither of these analyses is a derivation or a derivational compound.

The last row in the lexicon gives the derivational compound analysis of *typesetter*, with `Y␣`under ***DerComp***. Since it is the only possible derivational compound analysis, it is also a default analysis, and therefore gets `Y␣`under ***Def*** too. The `N␣`under ***Der*** and ***Comp*** confirm that it is not a derivation or a compound.

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

Stem	MorphNum	NVAffComp	Der	Comp	DerComp	Def	Imm	ImmClass
dive-bomber	2	Y	N	Y	N	Y	dive+bomber	VN
dive-bomber	3	Y	N	Y	N	N	dive+bomber	NN

In the same way, if you want to examine derivational compound analyses, and leave out all the other analyses, you should add an ‘expression’ restriction to your lexicon which states that `DerComp␣=␣Y`. In the example lexicon, this restriction produces the following result:

Stem	MorphNum	NVAffComp	Der	Comp	DerComp	Def	Imm	ImmClass
typesetter	2	Y	N	N	Y	Y	type+set+er	NVx

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.

Sometimes there may be more than one default analysis. If you want to see just the default analysis of each compound, you should use these two restrictions: `Def␣=␣Y␣`and `Comp␣=␣Y`. In the example lexicon, this means that the non-preferred noun + noun analysis is left out:

Stem	MorphNum	NVAffComp	Der	Comp	DerComp	Def	Imm	ImmClass
dive-bomber	2	Y	N	Y	N	Y	dive+bomber	VN

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 analyses you really want.

3.2.2 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 ten 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 remaining six deal with particular features which sometimes occur in morphological analysis: stem allomorphy, affix substitution, opacity, derivational transformation, infixation and reversion.

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 *nameplate* is shown as `name+plate`, in accordance with the various rules discussed in section 3.1.1 . Note that each element is given in the form of a headword or an affix, even when the original word doesn't use that particular form. Thus the stem *liturgical* is analysed as *liturgy+ical*, where *liturg* is rewritten in the normal form of the stem *liturgy*. 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, leaving out any `+␣`signs. 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 16 shows you the labels used in this column:

Word Class	Label
Noun	`N`
Adjective	`A`
Numeral	`Q`
Verb	`V`
Article	`D`
Pronoun	`O`
Adverb	`B`
Preposition	`P`
Conjunction	`C`
Interjection	`I`
Single contraction	`S`
Complex contraction	`T`
Affix	`x`

Table 16: Word class labels (immediate segmentation)

Using these codes, *nameplate* is given the code `NN`, to indicate that it is made up of two nouns (a compound), and *emigration* has the code `Vx␣`to indicate that it is made up of a verb and an affix (a derivation). 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 17 shows you these codes, along with their meaning.

Verbal sub-category	Label
Intransitive	`1`
Transitive	`2`
Intransitive & transitive	`3`
Unmarked for transitivity	`0`

Table 17: One-character verbal subclass labels

In this column, the word *emigration* has the code `1x`. It is exactly the same as the code in the previous column, except that the `V␣`is replaced by the number `1`, indicating in more detail what sort of verb it is.

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

ImmSubCat
(ImmSubCatLemma) `Immediate␣segmentation,␣subcat␣labels␣`

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, and upper case `F␣`indicates a flectional form of a stem. Thus *emigration* is represented as `SA`, and *bagpipes* as `SF`. The FLEX name and description of this column are as follows:

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

The fifth immediate segmentation column concerns stem allomorphy. Within a word, a stem sometimes takes a form different from the one used when it is written down as a word in its own right. When morphological analysis is noted down, any resulting stems are given their normal stem form, because it's easiest to understand. An example is the word *abundant*, which comprises the stem *abound* and the affix *ant*. Note the difference between what appears in the original word (*abund*) and its regular stem form (*abound*): each has the same meaning; the only difference between them is their spelling. This is an example of *derivational* stem allomorphy, since a new word has been derived by linking a different form of stem to an affix.

Another sort of stem allomorphy sometimes occurs with conversions – that is, words which change their word class without the addition of an affix (*sleep* is both a noun and a verb, for example). When conversion occurs, and the form of the stem seems to have altered, the process can be termed *conversion with allomorphy*. The verb *halve* is an instance of conversion with allomorphy, since it is a conversion from the noun form *half*. Thus *half* and *halve* are considered *allomorphs*: two different forms or representations of the same stem. There are three types of conversion with allomorphy. The first is the voicing of the final consonant with the addition of a final *-e*: thus the verb *thieve* is considered to be a conversion of the noun *thief*. The second is the same process in reverse – the removal of a final *-e*, and the devoicing of the last consonant: thus the noun *belief* is a conversion of the verb *believe*. The third is the change in spelling from final *s* to *c*: the noun *practice* can thus be thought of as a zero-derivation from the verb *practise*.

The next type of stem allomorphy is *flectional allomorphy*, a relatively rare type. When the irregular past tense of a verb is used as an adjective, both are said to derive from the infinitive form, so that the adjective *drunken* comes from the verb *drink*. The same is true for past participle forms: the adjective *born* thus derives from the verb *bear*.

There are two other categories which are dealt with under stem allomorphy even though they're not really instances of stem allomorphy – *clippings* and *blends*. Clippings are shortened forms of words which do not change word class. For example, *phone* is a simple clipping of *telephone*. Sometimes a clipping consists of more than one morpheme – *vibes* is a clipping of *vibraphone* which contains the stem *vibraphone* and the affix *-s*, and *hanky* is a diminutive form consisting of the stem *handkerchief* and the affix *-y*.

A blend is a word which is made up of two stems, at least one of which may be shortened. The word *smog* is made up of the stems *smoke* and *fog*, and *paratrooper* consists of the stems *parachute* and *trooper*. Note that the definition of a blend only allows for stems, not affixes.

The table below summarizes the five types of allomorphy and shows the codes used to identify them in the ***ImmAllo*** column.

Stem Allomorphy	Code	Example
Blend	`B`	*breathalyse*
Clipping	`C`	*phone*
Derivational	`D`	*clarify*
Flectional	`F`	*born*
Conversion	`Z`	*belief*

Table 18: Stem allomorphy codes

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

ImmAllo `Stem␣allomorphy,␣top␣level␣`
(ImmAlloLemma)

The sixth immediate segmentation column marks stems with a morphological analysis involving *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, *active* is analysed as the stem *action* and the affix *-ive*; the affix *-ion* has disappeared, and the new affix *-ive* 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:

ImmSubst `Affix␣substitution,␣top␣level␣`
(ImmSubstLemma)

The seventh column identifies those words whose analysis is *opaque* – that is, words made up of morphemes which are recognisable, but where the meaning of the head element isn't reflected in the meaning of the full word. An example of this is *accordion*: it appears to be made up of the verbal stem *accord* (the head element) and the affix *-ion*. Since the semantic link between *accord* and *accordion* is far from obvious, the analysis is marked as being opaque, and it gets a `Y␣`in this column. Words whose analyses are morphologically and semantically clear get the code `N`. The FLEX name and description of this column are as follows:

ImmOpac `Opacity,␣top␣level␣`
(ImmOpacLemma)

The eighth immediate segmentation column gives simple expressions to illustrate any orthographic alterations the analysis of a word involves. A morpheme boundary is marked by a `#`, and letters removed from either side of a morpheme are prefixed by a `-`, and letters which are added are prefixed by a `+`. Letters which do not change are considered part of a morpheme, and not shown. A simple example is `#␣`- this is the pattern for the word *unable*, since it consists of the affix *un-* and the stem *able*, and neither morpheme alters. On the other hand, *undersized* is given as `#-e#`, since nothing happens to the first morpheme *under-*, the final *e* of the second morpheme *size* is removed, and nothing happens to the last morpheme *-ed*. The FLEX name and description of the columns that contain these expressions are as follows:

TransDer
(TransDerLemma) `Derivational␣transformation,␣top␣level␣`

The ninth column indicates which stems have an immediate analysis involving derivation by means of an infix. Usually, derivational affixes are added to the beginning or end of a stem, but in some cases the affix is inserted into a multi-word, as in derivations from verb-and-particle combinations like *hanger-on* from *hang on* and *looker-on* from *look on*. Stems marked for this type of infixation get the code `Y␣`in this column, all other analyses get the code `N`.

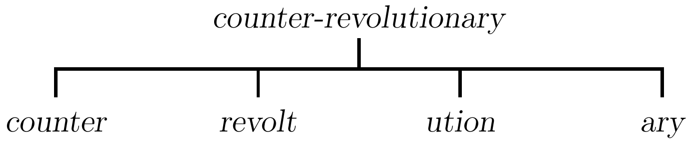
ImmInfix
(ImmInfixLemma) `Infixation,␣top␣level␣`

The last immediate segmentation column deals with stems analysed as conversions from multi-words which have undergone reversion of their parts in the process of conversion. For instance, the noun *downpour* is considered a conversion of the verb *pour down*, and the adjective *off-putting* is derived from the verb *put off* via its flection *putting off*. Whenever a stem is analysed in this way, this column yields a `Y␣`code. In all other cases, an `N␣`code is given.

ImmRevers
(ImmReversLemma) `Reversion,␣top␣level␣`

3.2.3 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) 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 *counter-revolutionary* 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 *counter-revolutionary* is written in the following way:

`counter+revolt+ution+ary␣`

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
Numeral	Q
Verb	V
Article	D
Pronoun	O
Adverb	B
Preposition	P
Conjunction	C
Interjection	I
Single contraction	S
Complex contraction	T
Affix	x

Table 19: Word class labels (flat segmentation)

Using these codes, the stem *counter-revolutionary* is given as `xVxx`. The FLEX name and description of the column are as follows:

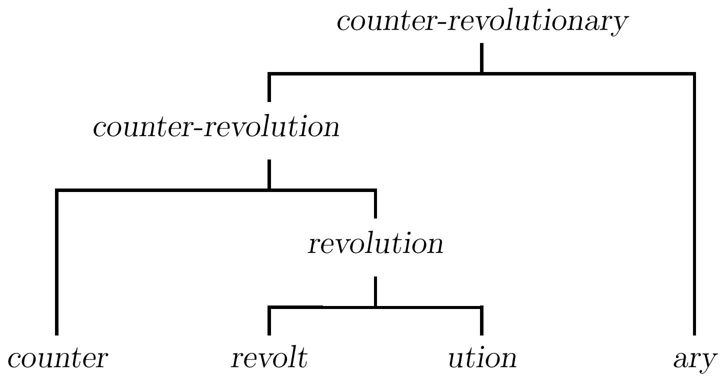
FlatClass (FlatClassLemma) `Flat␣segmentation,␣word␣class␣labels␣`

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

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

3.2.4 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 *counter-revolutionary*, you can get the following morphological analysis:



There are six 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 element on each level, or the word class without the spelling of the element 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 *counter-revolutionary* thus looks like this:

```
(((counter),((revolt),(ution))),(ary))␣
```

Each identifiable stem or affix is enclosed by a pair of brackets, beginning with the brackets round the full original stem. Then there are brackets round the stem *counter-revolution*, and subsequently round the stem *revolution*. Finally there are brackets round each of the four 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`
Numeral	`Q`
Verb	`V`
Article	`D`
Pronoun	`O`
Adverb	`B`
Preposition	`P`
Conjunction	`C`
Interjection	`I`
Single contraction	`S`
Complex contraction	`T`

Table 20: Word class labels (complete segmentation)

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

```
(((counter)[N|.N],((revolt)[V],(ution)[N|V.])[N])[N],(ary)[A|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 *counter-revolutionary* example above, the code given alongside the affix *counter* is `[N|.N]`. The `N␣`before the bar means that the affix *counter* helps to form a stem which is a noun (*counter-revolution*). The `.N␣`after the bar means that the segmentation of the noun *counter-revolution* is affix plus noun. 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 *counter* code is `.N␣`helps you to see at once that this affix helps to form a derivation in conjunction with a noun.

Sometimes a pair of affixes can only be used together, as in the word *aerodrome* – the word *aero* does not exist and the word *drome* does not exist. In such cases, `x␣`marks the other affix, and denotes that the affixes must occur in combination with each other: so-called *combining forms*. The code for the *aero-* of *aerodrome* is thus `[N|.x]`, and the code for the *-drome* is `[N|x.]`.

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

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 *counter-revolutionary* gets the following representation:

```
((()[N|.N],(()[V],()[N|V.])[N])[N],()[A|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
(StrucBrackLabLemma)
`Structured␣segmentation,␣word␣class␣labels␣only␣`

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. An example is the word *inedible*, which comprises the affix *in-*, the stem *eat* and the affix *-ible*: note the difference between *ed* and *eat*, where the one element is spelt two different ways. This is stem allomorphy. If stem allomorphy occurs at any point in a stem's complete hierarchical segmentation, a code is given in this column to show what sort of stem allomorphy occurs. The table below shows the codes, and you can read more about what each code means in section 3.2.2 above – they are the same codes used in the ***ImmAllo*** column.

Stem Allomorphy	Code	Example
Blend	`B`	*breathalyse*
Clipping	`C`	*phone*
Derivational	`D`	*clarify*
Flectional	`F`	*born*
Conversion	`Z`	*belief*

Table 21: Stem allomorphy codes

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

StrucAllo
(StrucAlloLemma)

`Stem␣allomorphy,␣any␣level␣`

The fifth hierarchical segmentation column marks stems with a morphological analysis involving *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, *melodic* is analysed as the stem *melody* plus the affix *-ic*; the affix *-y* has disappeared, and the new affix *-ic* 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
(StrucSubstLemma)

`Affix␣substitution,␣any␣level␣`

The sixth and last hierarchical segmentation column identifies those words whose analysis is completely or partly *opaque* – that is, words made up of morphemes which are recognisable, but where the meaning of the head element isn't reflected in the meaning of the full word. An example of this is *ladykiller*: it appears to be made up of the noun stem *lady* and the noun stem *killer* (which can subsequently be analysed as *kill* plus -*er*). Since the meaning of the head element *killer* doesn't relate directly to the meaning of the full word, the analysis is marked as being opaque, and it gets a `Y␣`in this column. Words whose analyses are morphologically and semantically clear get the code `N`. The FLEX name and description of this column are as follows:

StrucOpac `Opacity,␣any␣level␣`
(StrucOpacLemma)

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* a stem contains after complete segmentation, 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 a special 'link morpheme' (-*a*-, -*o*-, or -*s*-). And of course, monomorphemic words only contain one component. Any stems which cannot receive an adequate morphological analysis (for the reasons given in section 3.1.4) get the number 0.

Some examples: in the stem *counter-revolutionary*, the number of components is two (the stem *counter-revolution* and the affix -*ary*), and for *law-breaker* it is three (the stem *law*, the stem *break*, and the affix -*er*.

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

CompCnt `Number␣of␣morphological␣components␣`
(CompCntLemma)

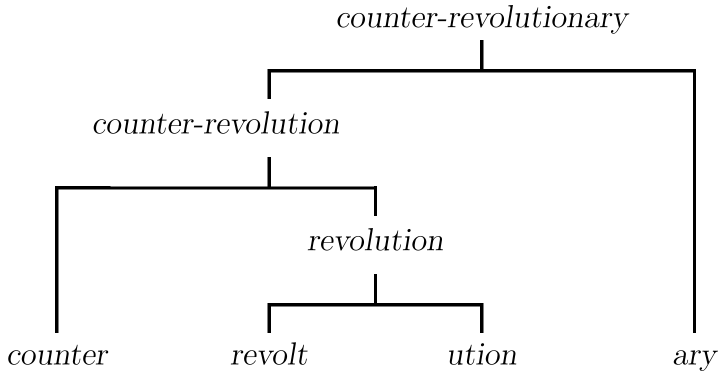
The second column gives you the number of morphemes in each stem. For words without a morphological analysis, the number given is zero. The number of morphemes in the stem *counter-revolutionary* for example is four, while for *lawbreaker* it is three.

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

MorCnt
(MorCntLemma)

```
Number␣of␣morphemes␣
```

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 another look at the tree diagram illustrating the analysis of *counter-revolutionary*:



Including the stem at the top, the diagram covers four 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; 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 3.1.4) 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 ENGLISH WORDFORMS

There are two types of morphology information available for the 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 thirteen '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 *eat* is understood as referring not only to the form *eat* itself, but also the forms *eats*, *eating, ate*, and *eaten*. 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 a COBUILD 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 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 thirteen 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 first person, present tense verb forms in the database, you have to include at least three columns in the wordforms lexicon you create, namely a column which gives the orthographic representations you prefer, along with ***Pres*** and ***Sin1*** (which are amongst the thirteen columns described below). You must then construct two restrictions for your lexicon, one stating that ***Pres*** must be equal to `Y`, and another stating that ***Sin1*** must be equal to `Y`. You can then format your lexicon to make sure that ***Pres*** and ***Sin1*** are not ‘on view’: that way, when you `SHOW␣`or `EXPORT␣`your lexicon, you just get the list of words you require without two lists of `Y`’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 English wordforms’ options.

The first inflectional features column indicates whether a wordform is a singular form of any sort. This means past and present tense verb forms such as *hibernated* or *babbles*, or nouns such as *sagacity*. The FLEX name and description of this column are as follows:

Sing `Inflectional␣feature:␣singular␣`

The second column indicates whether a wordform is a plural inflection of any sort. This means past and present tense verb forms such as *hibernate* or *submerged*, or nouns such as *jocularities*. The FLEX column name and description of this column are as follows:

Plu `Inflectional␣feature:␣plural␣`

The third column marks all the wordforms which are positive forms – that is, not comparative or superlative forms like *better* and *best*, but plain adjectival forms like *good* or *often*. Thus adjectives like *goofy* and *idiomatic* or adverbs like *seldom* and *idiomatically* 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 fourth column marks all the wordforms which are comparative forms, almost always adjectives. Wordforms such as *better* or *angrier* or *cannier* 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 *further*. The FLEX name and description of this column are as follows:

Comp `Inflectional␣feature:␣comparative␣`

The fifth column marks all superlative forms, so that wordforms such as *best* or *angriest* 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 *furthest*. The FLEX name and description of this column are as follows:

Sup `Inflectional␣feature:␣superlative␣`

The sixth 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. Words like *waffle* or *have* or *eat*, which can be used with the particle *to* in front of them, are infinitives. 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 seventh column marks any participles, past tense or present tense. Present participles are normally formed by adding *-ing* to the stem of the verb, with the exception of some irregular verbs. Past participles add a suffix ending in *-d* to the stem, and they are used in the formation of the perfect tense: 'I've *lived* in Nijmegen for four years'. Again, many irregular verbs don't match this rule (*gone* is the past participle of *go*, for example). Most past participles can also be used adjectivally, as in 'the *panelled* walls'. 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 eighth column identifies any present tense forms, including the present participles mentioned under ***Part***. Thus verb forms like *gleam*, *gleams* and *gleaming* 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 ninth column identifies any past tense forms, including the past participles mentioned under ***Part***. Thus forms like *occupied* and *elicited* 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 tenth column marks first person singular forms of verbs, whether present tense or past tense. So, all first person singular forms, like 'I *go*' or 'I *finished off*', 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 eleventh column marks second person singular forms of verbs, whether present tense or past tense forms. For most verbs, the second person form is the same as the first person form, but some irregular verbs are exceptions. So all second person forms like 'you *are*' or 'you *shout*' 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 twelfth column identifies third person singular forms of verbs, whether present tense or present tense forms. For most verbs, the third person present tense form consists of the stem plus the suffix *-s*. Thus forms like 'he *stood up*' or 'Gilbert *acts*' 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 thirteenth and last column marks rare forms – normally forms which have become outdated like *brethren*, *shouldst*, or *wert*. Such 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:

Rare `Inflectional␣feature:␣Rare␣form␣`

3.4.2 TYPE OF FLECTION

In the 'Inflectional Features' section above, thirteen different inflectional features are distinguished, and assigned to thirteen 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 22 below sets out all the combinations of single-letter codes that occur.

Inflectional feature	Label	'yes/no' column name
Singular	`S`	***Sing***
Plural	`P`	***Plu***
Positive	`b`	***Pos***
Comparative	`c`	***Comp***
Superlative	`s`	***Sup***
Infinitive	`i`	***Inf***
Participle	`p`	***Part***
Present tense	`e`	***Pres***
Past tense	`a`	***Past***
1st person verb	`1`	***Sin1***
2nd person verb	`2`	***Sin2***
3rd person verb	`3`	***Sin3***
Rare form	`r`	***Rare***
Headword form (not nouns, verbs adjectives or adverbs)	`X`	

Table 22: 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 prepositions like *among* or *less*, pronouns like *that* or *hers*, conjunctions like *immediately* or *that*, numerals like *fifth* or *thousand*, contracted forms like *I'll* or *hadn't*, and interjections like *phew* or *amen*. 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, adjectives or adverbs ever get the code `X`.

Each wordform may have more than one code attached to it. Thus the wordform *boasted* has the code `a3S`: `a␣`means it is a past tense form, `3␣`means that it is a third person form, and `S␣`means that it is singular.

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

FlectType `Type␣of␣flection␣`

3.4.3 INFLECTIONAL TRANSFORMATION

The last column shows how the orthographic form of a stem is altered when a flection is formed. Each string of letters in the stem is shown by the symbol `@`, so the first person present tense form of the verb whose stem is *abide* is simply given as `@`. Any blanks or hyphens in the stem are shown as a blank, so *abide by* is shown as `@ @`. Letters removed from the front or back of a string are prefixed by a minus sign `-`, and letters added are prefixed by a plus sign `+`, so *abiding by* is given as `@-e+ing␣@`: first the final *-e* of *abide* is removed, and then the suffix *-ing* is added. This formalism is an unambiguous way of showing the inflectional transformations that occur in the orthographic formation of wordforms.

Whenever the inflectional transformation is irregular (as with the past tense forms of the verb *sing* for example – *sang* and *sung*) no transformation is given; the field remains empty.

The tables below show all the lettergroups represented in the database which can be subtracted from or added to a headword to make a wordform.

Lettergroups removed from a headword				
e	ey	f	fe	y

Table 23: Inflectional transformation codes (letters removed)

Lettergroups added to a headword				
bed	ber	best	bing	d
ded	der	dest	ding	ed
er	es	est	ged	ger
gest	ging	ied	ier	ies
iest	ing	ked	king	led
ler	lest	ling	med	mer
mest	ming	ned	ner	nest
ning	ped	ping	r	red
ring	s	sed	ses	sing
st	ted	ter	test	ting
ved	ving	ves	zed	zes
zing				

Table 24: Inflectional transformation codes (letters added)

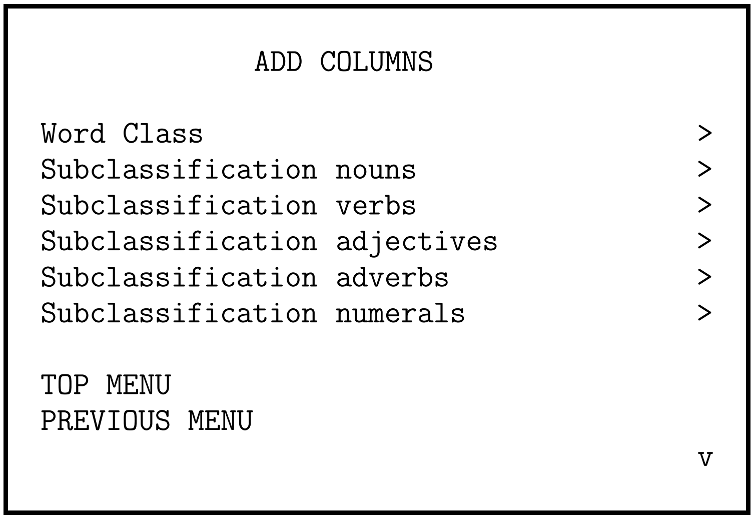
The FLEX name and description of this column are as follows:

TransInfl
(TransInflLemma)

`Inflectional␣transformation␣`

4 ENGLISH SYNTAX

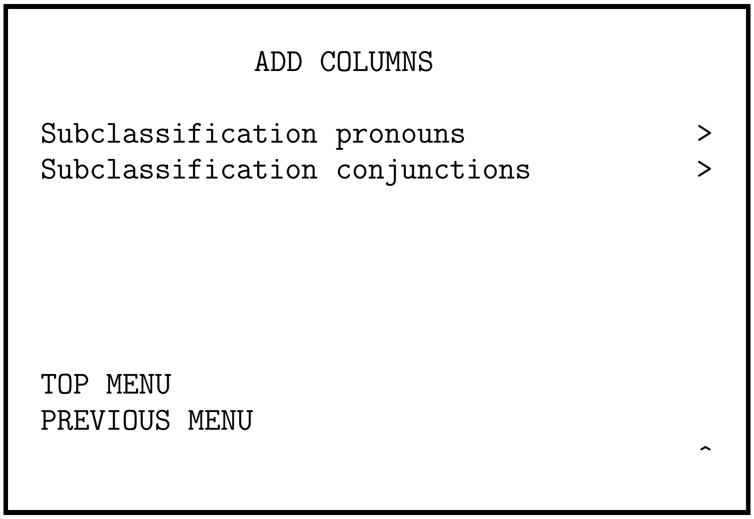
Syntactic information is available for lexicons of type lemma, or with the lemma information presented with *Morphology of English Wordforms*. The subsections which make up this section correspond to the seven subclassification options FLEX offers you when you ask for syntactic information under the ADD COLUMNS window:



```
               ADD COLUMNS

Word Class                               >
Subclassification nouns                  >
Subclassification verbs                  >
Subclassification adjectives             >
Subclassification adverbs                >
Subclassification numerals               >

TOP MENU
PREVIOUS MENU
                                         v
```



```
               ADD COLUMNS

Subclassification pronouns               >
Subclassification conjunctions           >






TOP MENU
PREVIOUS MENU
                                         ˆ
```

First and foremost, as shown in the two ADD COLUMNS menus, there are basic word class codes available for all the lemmas in the database, in the form of numbers or labels. Then there is a wealth of subclassification information which supplements the basic word class codes: for each of the more important classes of lemma, there are a number of columns which indicate whether or not a certain lemma has a particular feature. For example, you can quickly see that the verb *filch* is transitive and the verb *fizzle* is not when you check the entries for these words in the ***Trans_V*** column. An explanation of the format used for the subclassification columns is given in section 4.2 .

4.1 WORD CLASS CODES – LETTERS OR NUMBERS?

There are two ways of representing the syntactic code of each lemma. You can choose for yourself 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 code you decide to use, the *information* remains the same; only the *format* changes. Examples are provided in the table below.

Word class codes are a simple way of identifying the syntactic class of every lemma in the database. In addition, they also identify the word class of every *wordform*, since all the wordforms which make up the inflectional paradigm of any given lemma naturally share the same word class. So if you want a wordform lexicon which gives you the word class of any wordform, you have to use the lemma information columns which are available under *Morphology of English Wordforms*.

Twelve basic categories—set out in Table 25 below—are distinguished, and you can use them in either of the two forms described here. Only the last two classifications given may need some explanation: they deal with *contractions*, those words which are made up of shortened forms of other words. For example, *isn't* is a contraction of *is* and *not*, and *d'you* is a contraction of *do* and *you*. Two types of contraction are identified. A *simple contraction* is a shortened form in isolation: *'ve* and *'re* are both simple contractions. A *complex contraction* is one form which contains two words, one of which might be shortened: *I've* and *couldn't* are both complex contractions.

Word Class	Columns		Example
	ClassNum	***Class***	
Noun	1	`N`	*garrison*
Adjective	2	`A`	*biblical*
Numeral	3	`NUM`	*twentieth*
Verb	4	`V`	*chortle*
Article	5	`ART`	*the*
Pronoun	6	`PRON`	*mine*
Adverb	7	`ADV`	*sheepishly*
Preposition	8	`PREP`	*through*
Conjunction	9	`C`	*whereas*
Interjection	10	`I`	*alleluia*
Single contraction	11	`SCON`	*'re*
Complex contraction	12	`CCON`	*you're*
Letter	13	`LET`	*r*
Abbreviation	14	`ABB`	*NHS*
Infinitival 'to'	15	`TO`	*to*

Table 25: English word class codes

If you want word class codes in the form of numbers, then choose the column which has the following FLEX name and description:

ClassNum (ClassNumLemma) `Word␣class,␣numeric␣`

If you want word class codes in the form of short verbal symbols, choose the column which has the following FLEX name and description:

Class (ClassLemma) `Word␣class,␣labels␣`

4.2 SUBCLASSIFICATION – Y OR N

All the remaining syntactic columns come under a general heading of *subclassifications*. Each of these subclassification columns represents a particular syntactic attribute that a lemma might have. The values contained in each column can only be one of two characters: `Y␣`or `N`. Using these columns you can carry out quick checks on the qualities of any lemma. Is the conjunction *whereas* a coordinating conjunction? Look up its value in ***Cor_C*** and you discover that `N`, no, it is not.

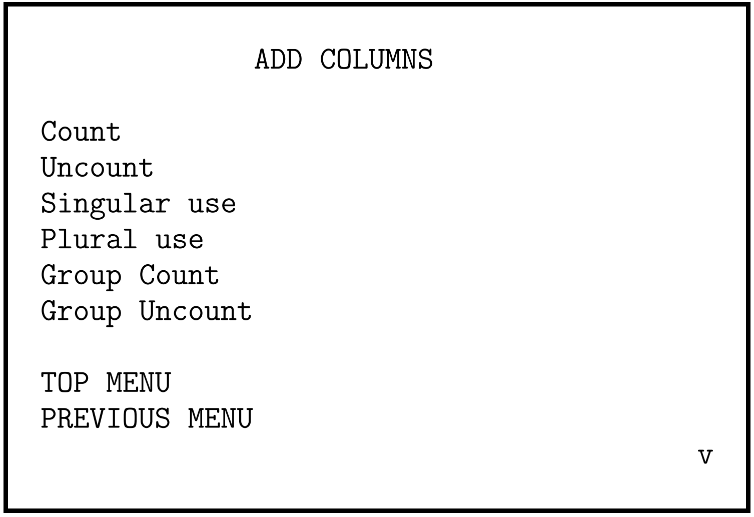
Is the noun *brother* a vocative form? Check its value in the column ***Voc_N***, and you discover that `Y`, yes, it is (or rather, yes, it can be used in this way).

Of course it's possible to do more with these columns than simple checks on individual lemmas. You can use them to define the contents of your lexicon by constructing a restriction with one or more of them. If you want (say) a lexicon which only contains entries that can be ditransitive verbs, then you should first include the column ***Ditrans_V*** in your lexicon, then construct a restriction which states that `Ditrans␣V␣=␣Y`. Moreover you can construct any number of restrictions in this way, to make the contents of your lexicon even more closely defined.

Remember, though, that these `Y / N␣`answers are not exclusive answers. The verb *write* can be both transitive (*She writes very long letters*) and intransitive (*What does he do for a living? He writes!*). So under the columns ***Trans_V*** and ***Intrans_V***, *write* gets the code `Y`. It follows that if you want a list of verbs which are *always* intransitive, you must construct two restrictions which state first that `Trans␣V␣=␣N␣`and second that `Intrans␣V␣=␣Y`. To get the best from these subclassification codes, then, you should read the descriptions given below carefully, work out exactly how to get what you want using restrictions, and then use FLEX to build carefully planned lexicons.

4.3 SUBCLASSIFICATION NOUNS

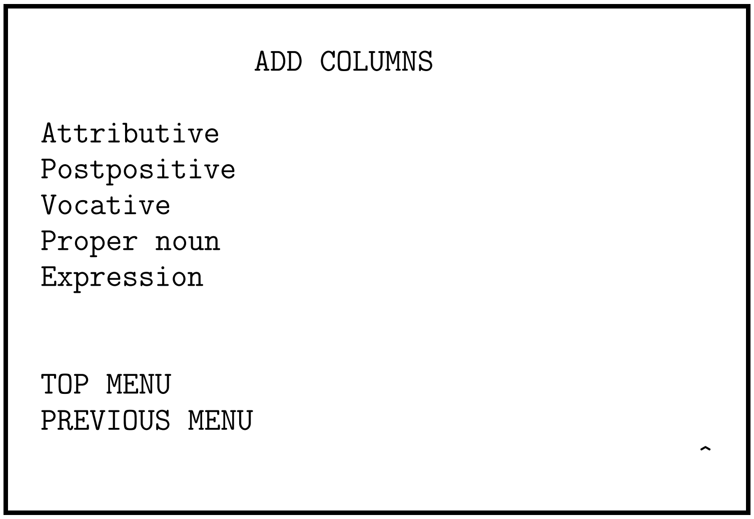
There are eleven possible attributes which a noun can have, so two ADD COLUMNS menus are needed to cover them all. (The `v␣`and `^␣`symbols in the bottom right hand corner of either screen indicate that you should use the NEXT and PREV keys to move from one part of the menu to the other.)



```
                ADD COLUMNS

Count
Uncount
Singular use
Plural use
Group Count
Group Uncount

TOP MENU
PREVIOUS MENU
                                        v
```



```
                ADD COLUMNS

Attributive
Postpositive
Vocative
Proper noun
Expression


TOP MENU
PREVIOUS MENU
                                        ^
```

What follows now is a concise description of each of these ten attributes, along with the column names and descriptions used in FLEX.

The first column answers the question ‘is this lemma a *count noun*?’ Such nouns can be treated as individual units, and thus counted; you can talk about *seventy-six trombones*, but not **two musics*. Thus the noun lemma *trombone* gets the code `Y␣`because it is a count noun, while the noun lemma *music* gets the code `N␣`because it isn’t a count noun. And of course every lemma with a word class other than noun automatically gets the code `N`. The FLEX name and description of this column are as follows:

C_N
(C_NLemma) `For␣nouns,␣countable␣`

The second column a mirror image of the first: it answers the question ‘is this lemma an *uncountable noun*?’ Uncountable nouns are those nouns which are continuous entities; they are not individual units which can occur in both singular and plural forms. You cannot talk about *a dozen handwritings*, but it is quite permissible to talk about *twelve pens*. The lemma *handwriting* thus gets the code `Y`, while the lemma *pen* gets the code `N`. Another term used for *uncountable noun* is *mass noun*.

Note, however, that some nouns appear to be both countable *and* uncountable. The noun *space* is an uncountable noun when used in a phrase like *Space – the Final Frontier*, but countable when used in a phrase like *The Netherlands – flat, open spaces*. This is one example of a lemma which gets a `Y␣` in the count *and* the uncount column.

Any noun lemma which can be uncountable gets the code `Y␣` in this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Unc_N
(Unc_NLemma)

`For␣nouns,␣uncountable␣`

The third column answers the question ‘does this lemma only ever occur in the singular form?’ This refers to what are sometimes called *singularia tantum*. The lemma *monopoly*, when used in a phrase like *they think they have a monopoly on the truth*, can only be singular, and it therefore gets the code `Y␣`in this column. (Note however that in other circumstances, *monopoly* does occur in the plural: the phrase *private or state monopolies*, for example.) Any noun lemma that can occur in a singular-only form gets the code `Y␣`in this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Sing_N
(Sing_NLemma)

`For␣nouns,␣singular␣use␣`

The fourth column answers the question ‘does this lemma ever occur in a plural-only form?’ This refers to what are often called *pluralia tantum*. The noun lemma *people* occurs in election-time phrases like *the people have spoken* where it means the general population of a country or region. It only occurs with a plural verb form, and therefore gets the code `Y␣` in this column. Any other lemmas which can never be plural-only nouns get the code `N`. The FLEX name and description for this column are as follows:

Plu_N
(Plu_NLemma)

`For␣nouns,␣plural␣use␣`

The fifth and sixth column deal with *collective* nouns: those nouns which refer to groups of people or things (the lemma *majority* for example). Usually, all such lemmas can take a plural or a singular form of a verb: *the majority is in favour* is as acceptable as *the majority are in favour*. However, only in certain cases does the inflectional paradigm of the lemma include a plural form. The lemma *mankind*, for example, can take a plural or a singular verb, but it has no plural form **mankinds*. In contrast, the lemma *crew* can take a plural or a singular form of the verb *and* it has a plural form *crews*.

The fifth column answers the question ‘is this lemma a collective noun that has a singular *and* a plural form?’ The lemma *government* gets the code `Y`, because it’s always possible to talk about a particular (say the French) *government* as well as several (say the European) *governments*. The lemma *mankind* gets the code `N␣`since it never has a plural form under any circumstances. Other lemmas which are neither collective nor nouns also get the code `N`. The FLEX name and description of this column are as follows:

GrC_N
(GrC_NLemma)

`For␣nouns,␣group␣countable␣`

The sixth column answers the question ‘Is this lemma a collective noun that only has a singular form, and *not* a plural form? The lemma *populace* has the code `Y␣`because it is never possible to use the form **populaces*, and for the same reason the lemma *mankind* also gets a `Y␣`code. The FLEX name and description of this column are as follows:

GrUnc_N
(GrUnc_NLemma)

`For␣nouns,␣group␣uncountable␣`

The seventh column answers the question ‘can this lemma be used attributively?’ This refers to nouns like *machine* which occurs in phrases like *machine translation*: the first word says something about the word that follows it. So in this column, *machine* gets the code `Y␣`since it can be used attributively. A lemma like *gadgetry*, on the other hand, gets the code `N`, because it cannot be used attributively. The FLEX name and description of this column are as follows:

Attr_N
(Attr_NLemma)
`For␣nouns,␣attributive␣`

The eighth column answers the question ‘can this lemma ever be used in a postpositive way?’ Here *postpositive* refers to nouns which come after another noun and which qualify that other noun. An example is *proof*, which occurs in phrases like *Bushmills whiskey is forty-two percent proof*. The FLEX name and description of this column are as follows:

PostPos_N
(PostPos_NLemma)
`For␣nouns,␣postpositive␣`

The ninth column answers the question ‘is this lemma used to address people or things?’ This usually refers to nouns like *chicken*, which can be used when you are speaking directly to a hen (as in the well-known song *chick chick chick chick chicken, lay a little egg for me*), or when you are speaking to someone you consider to be a coward (*Chicken! Come back and fight!*). The noun *chicken* thus gets the code `Y␣` because it can be used as a vocative, whereas a noun *daffodil* gets the code `N␣`because normally (the more flowery poets excepted) it cannot be used as a vocative. The FLEX name and description of this column are as follows:

Voc_N
(Voc_NLemma)
`For␣nouns,␣vocative␣`

The tenth column answers the question ‘is this lemma used as a proper noun?’ Proper nouns are the names of people or places, so that lemmas such as *Arthur* or *York* get the code `Y`, while lemmas which aren’t proper nouns get the code `N`. Most of the proper nouns in the database are included because they form a necessary part of the morphological analyses provided in the ‘Morphology of English lemmas’ columns. The FLEX name and description of this column are as follows:

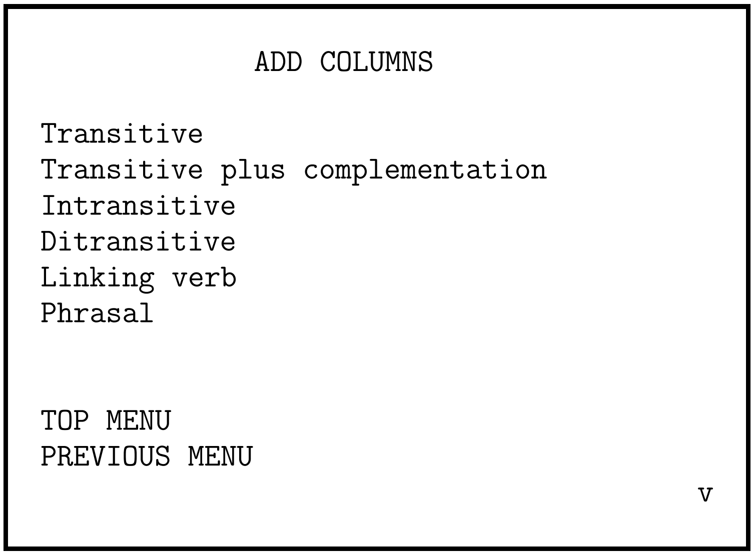
Proper_N
(Proper_NLemma)
`For␣nouns,␣proper␣noun␣`

The eleventh and last column answers the question 'is this noun lemma only ever used in combination with certain other words to make up a particular phrase?' An example is the word *loggerheads*, which only ever occurs in the phrase *at loggerheads*, and *brunt* which only ever occurs in the phrase *bear the brunt of*. Such nouns get the code `Y`, while all other words get the code `N`. The FLEX name and description of this column are as follows:

Exp_N
(Exp_NLemma) `For␣nouns,␣expression␣`

4.4 SUBCLASSIFICATION VERBS

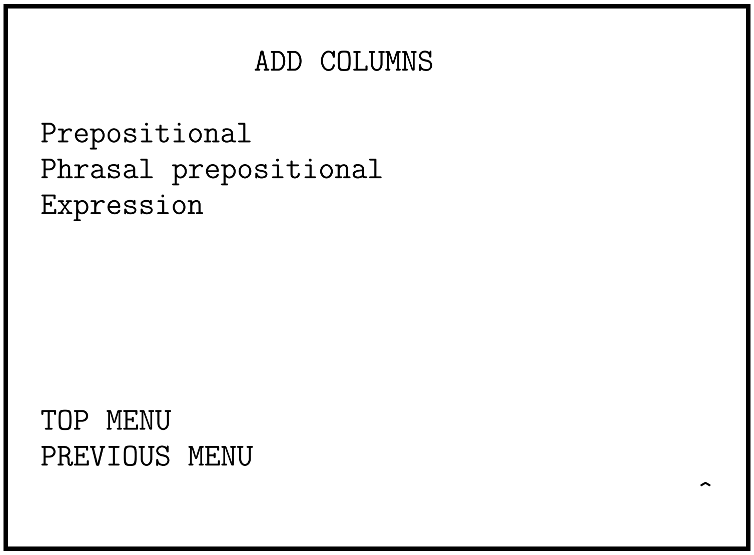
There are nine verbal subclassification columns available, and they all take the form of `Y` / `N`␣answers to questions (as described in section 4.2 above). They are presented to you in two ADD COLUMNS windows:



```
               ADD COLUMNS

Transitive
Transitive plus complementation
Intransitive
Ditransitive
Linking verb
Phrasal


TOP MENU
PREVIOUS MENU
                                        v
```



```
               ADD COLUMNS

Prepositional
Phrasal prepositional
Expression





TOP MENU
PREVIOUS MENU
                                        ˆ
```

The first of the nine columns answers the question ‘is this a verb which can (sometimes) take a direct object?’ The verb lemma *crash*, for example, gets the code `Y`, because you can say things like *he crashed the car*. So does the verb *admit*, because you can say things like *he admitted that he was wrong*, where the direct object is a clause. The verb lemma *cycle* gets the code `N␣`because you can’t say things like **he cycled the bike*. The FLEX name and description of this column are as follows:

Trans_V
(Trans_VLemma)

`For␣verbs,␣transitive␣`

The second column answers the question ‘is this a verb which has an object complement?’ In a phrase like *the jury found him guilty*, there is a direct object *him* plus a complement which relates to that direct object *guilty*. These object complements can take the form of a noun phrase (*they had made him chairman*), or an adjective phrase (*so he thought it odd*), or a prepositional phrase (*when they threw him into jail*), or an adverb phrase (*and kept him there*) or a clause (*since it caused him to be embarrassed*). Verbal lemmas which can take such complements get the code `Y␣`in this column; all other lemmas get the code `N`. The FLEX name and description of this column is as follows:

TransComp_V
(TransComp_VLemma)

`For␣verbs,␣transitive␣plus␣complementation␣`

The third verbal subclassification column answers the question ‘is this a verb which (sometimes) cannot take a direct object?’ The verb *alight* for example can never take a direct object – *he got the bus and alighted at the City Hall*. The verb *leave* can occur with or without a direct object – *she left a will* or *she left at ten o’clock*. Both verbs thus get the code `Y␣`in this column. The verb *modify* gets the code `N`, however, since it always takes a direct object: you cannot say **he modified*, but you can say *he modified his opinion on the matter*. The FLEX name and description of this column are as follows:

Intrans_V
(Intrans_VLemma)

`For␣verbs,␣intransitive␣`

The fourth column answers the question ‘is this a verb which can be ditransitive?’ Here *ditransitive* refers to verbs which can take two objects, one direct object plus one indirect object. The verb *envy*, for example, gets the code `Y␣`in this column, since you can say *he envied his colleagues their success.* So does *tell* because you can say things like *she told him she would keep in touch.* Verbs like *dance*, on the other hand, which cannot take two objects, and all other non-verbal lemmas, get the code `N`. The FLEX name and description of this column are as follows:

Ditrans_V
(Ditrans_VLemma) `For␣verbs,␣ditransitive␣`

The fifth verbal subclassification column answers the question ‘is this lemma ever a linking verb?’ The copula *be* is a linking verb – it links a subject *I* with a complement which describes that subject *a doctor* in a sentence like *I am a doctor.* These *subject complements* can take the form of a noun phrase (*she is an intelligent woman*), an adjective phrase (*she looks worried*), a prepositional phrase (*she lives in Cork*), an adverb phrase (*how did she end up there?*) or a clause (*her main intention is to move somewhere else*). Verbs which can have such subject complements get the code `Y␣`in this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Link_V
(Link_VLemma) `For␣verbs,␣linking␣verb␣`

The sixth column answers the question ‘is this lemma a phrasal verb?’ A phrasal verb is a verb which is linked to a particular adverb, such as *speak out* or *run away.* Often these combinations acquire a specific, idiomatic meaning. Phrasal verbs in the database get the code `Y␣`if they are marked as such in volume one of the *Oxford Dictionary of Current Idiomatic English.* All other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Phr_V
(Phr_VLemma) `For␣verbs,␣phrasal␣verb␣`

The seventh column answers the question ‘is this lemma a prepositional verb?’ A prepositional verb is one which is linked to a particular preposition, such as *minister to* or *consist of*. Prepositional verbs in the database get the code `Y␣`if they are marked as such in the *Oxford Dictionary of Current Idiomatic English*. All other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Prep_V
(Prep_VLemma) `For␣verbs,␣prepositional␣verb␣`

The eighth column answers the question ‘is this lemma a phrasal prepositional verb?’ A phrasal prepositional verb is, naturally, one which is linked to a particular adverb and also to a particular preposition, such as *walk away with* or *cry out against*. Phrasal prepositional verbs in the database are current phrasal prepositional verbs: only those which are in use nowadays are given, on the basis of their inclusion in the *Oxford Dictionary of Current Idiomatic English*. All phrasal prepositional verbs get the code `Y`, and all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

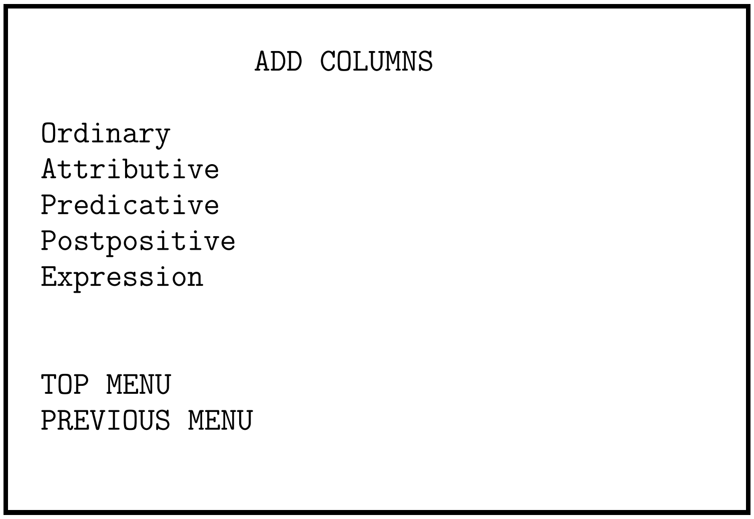
PhrPrep_V
(PhrPrep_VLemma) `For␣verbs,␣phrasal␣prepositional␣verb␣`

The ninth and last column answers the question ‘is this verb lemma only ever used in combination with certain other words to make up a particular phrase?’ An example is the verb *toe*, which only ever occurs in the phrase *toe the line*, and *bell* which only ever occurs in the phrase *bell the cat*. Such verbs get the code `Y`, while all other words get the code `N`. The FLEX name and description of this column are as follows:

Exp_V
(Exp_VLemma) `For␣verbs,␣expression␣`

4.5 SUBCLASSIFICATION ADJECTIVES

There are five attributes of adjectives covered in this version of the database, which are shown in the ADD COLUMNS window below. Each of these attributes or subclassifications has its own column in the database, and it always contains the code `Y␣`or `N␣`(this simple coding system is explained in section 4.2 above). Each of the five attributes and their database columns are explained below.



The first adjective subclassification column is a simple one: it answers the question 'is this lemma an ordinary adjective?', where *ordinary* means that it can be used both attributively ('the new book') and predicatively ('the book is new'). Thus adjectives such as *new* and *elementary* get the code `Y␣`because they are ordinary adjectives, while *actual* and *ablaze* get the code `N␣`because you cannot say '*the reason is actual' or '*the ablaze house'. The FLEX name and description of this column are as follows:

Ord_A (Ord_ALemma) `For␣adjectives,␣ordinary␣`

The second column gives an answer to the question 'is this lemma an adjective which in some contexts can only be used attributively?' Here, *attributive* means those adjectives which always come before the noun or phrase, as in *sheer nonsense*, where *sheer* cannot come after the noun it qualifies. The FLEX name and description of this column are as follows:

Attr_A (Attr_ALemma) `For␣adjectives,␣attributive␣`

The third column answers the question 'is this lemma an adjective which in some contexts can only be used predicatively?' Here *predicatively* refers to adjectives like *awake* which can only qualify a noun when linked to it by a verb – *the cat is awake*, for example. The FLEX name and description of this column are as follows:

Pred_A
(Pred_ALemma)
`For␣adjectives,␣predicative␣`

The fourth column answers the question ‘can this adjectival lemma ever be used in a postpositive way?’ Here *postpositive* refers to lemmas like the adjective *everlasting*, which occurs in the phrase *life everlasting*: here it comes *after* the noun it modifies, whereas it is more normal in English for a modifier to come *before* the noun it qualifies (*everlasting life* is also acceptable). So *everlasting* gets the code `Y␣`in this column, while adjectives like *durable*, which never occur postpositively, get the code `N`.

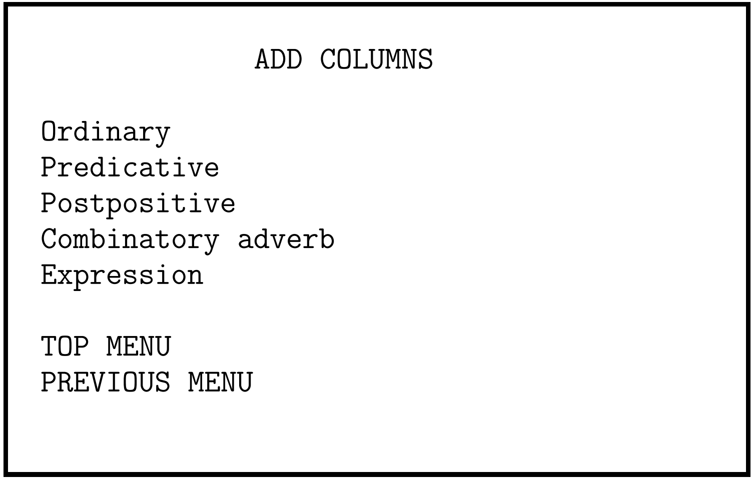
PostPos_A
(PostPos_ALemma)
`For␣adjectives,␣postpositive␣`

The fifth and last column answers the question ‘is this adjective lemma only ever used in combination with certain other words to make up a particular phrase?’ An example is the adjective *bated*, which only ever occurs in the phrase *with bated breath*, and the adjective *put* which only ever occurs in the phrase *stay put*. Such adjectives get the code `Y`, while all other words get the code `N`. The FLEX name and description of this column are as follows:

Exp_A
(Exp_ALemma)
`For␣adjectives,␣expression␣`

4.6 SUBCLASSIFICATION ADVERBS

There are five adverb subclassification columns available, and they all take the form of `Y / N␣`answers to questions (as described in section 4.2 above). They are presented to you in this ADD COLUMNS window:



The first of these five columns answers the question ‘is this lemma an ordinary adverb’, where *ordinary* simply means that it doesn’t necessarily have any special subclassification features; it’s just an adverb which can modify a verb or an adjective. Examples include *generously* which occurs in a phrase like *people have given very generously* or *a generously illustrated book*. The FLEX name and description of this column are as follows:

Ord_ADV (Ord_ADVLemma) `For␣adverbs,␣ordinary␣`

The second of the five adverb subclassification column answers the question ‘is this lemma an adverb which can usually only be used predicatively?’ Adverbs which get the code `Y␣` here can be distinguished from ordinary adverbs on the basis of their predicative use with the verb *be*: it is possible to say *the boat is adrift* but not **the boat is quickly*. Thus *adrift* is a predicative adverb, while *quickly* is not. Typically many predicative adverbs begin with the letter *a* – *around*, *astern* or *awry* for example. Many don’t however: *high*, *inland* and *downtown*, amongst others. All predicative adverbs get the code `Y␣`in this column, and all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Pred_ADV (Pred_ADVLemma) `For␣adverbs,␣predicative␣`

The third adverb subclassification column answers the question ‘is this lemma an adverb that can be used postpositively?’ Here *postpositive* means ‘after a noun’ as with *offside* in the phrase *several yards offside* or *apart* in the phrase *a race apart*. Adverbs which can be used in this way get the code `Y␣`in this column; all other adverbs get the code `N`. The FLEX name and description for this column are as follows:

PostPos_ADV
(PostPos_ADVLemma) `For␣adverbs,␣postpositive␣`

The fourth of the five adverb subclassification columns answers the question 'is this lemma an adverb which can be used in combination with a preposition or another adverb?' An example is the adverb *clean*: you can say *the sledgehammer broke clean through the door* – here an adverb combines with a preposition. Another example is the adverb *all*: you can say *they left the dog all alone* – here an adverb combines with another adverb. Adverbs which can combine in one of these two ways get the code `Y`; all other lemmas get the code `N`. The FLEX column name and description of this column are as follows:

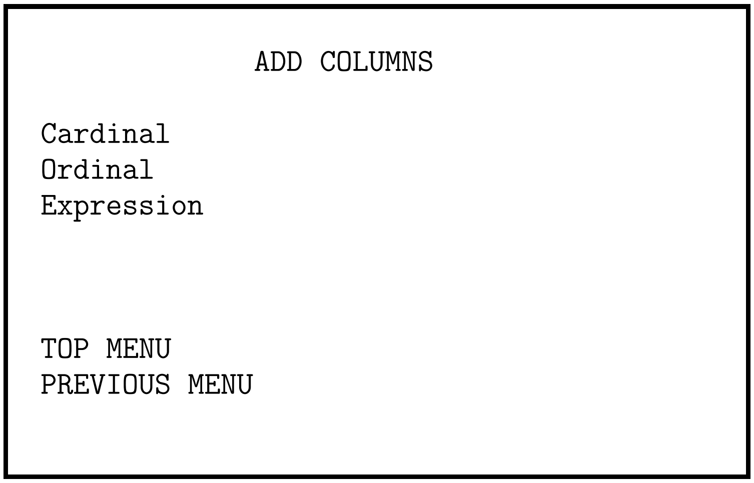
Comb_ADV
(Comb_ADVLemma) `For␣adverbs,␣combinatory␣adverb␣`

The last of the five adverb subclassification columns answers the question 'is this adverb lemma only ever used in combination with certain other words to make up a particular phrase?' An example is the adverb *amok*, which only ever occurs in the phrase *run amok*, and the adverb *screamingly* which only ever occurs in the phrase *screamingly funny*. Such adverbs get the code `Y`, while all other words get the code `N`. The FLEX name and description of this column are as follows:

Exp_ADV
(Exp_ADVLemma) `For␣adverbs,␣expression␣`

4.7 SUBCLASSIFICATION NUMERALS

The numerals given in the database are sufficient to let you spell any number in full: it's possible to reconstruct the orthography of number 51,094,386 (for example) using numerals from the database. As well as the numbers themselves, a few extra terms such as *score* or *umpteenth* are also given. There are two subclassification columns available for the numerals specified in the database, and they both take the form of `Y/N␣`answers to questions (as explained in section 4.2). In addition, there is a column which identifies those numerals used in certain expressions. All three columns are presented to you in this ADD COLUMNS window:



The first numeral subclassification column answers the question 'is this lemma a cardinal number?' Cardinal numbers—like *three* or *seven* or *twelve*—are the most important forms of numbers, and they simply indicate quantity rather than rank order. Any lemma in the database which is a cardinal number gets the code `Y␣`in this column, and every other lemma gets the code `N`. The FLEX name and description of this column are as follows:

Card_NUM (Card_NUMLemma) `For␣numerals,␣cardinal␣`

The second numeral subclassification column answers the question 'is this lemma an ordinal number?' In contrast to cardinal numbers, ordinals—like *third* or *seventh* or *twelfth*—indicate quantity *and* rank order. Any lemma in the database which is an ordinal number gets the code `Y␣`in this column, and every other lemma gets the code `N`. The FLEX name and description of this column are as follows:

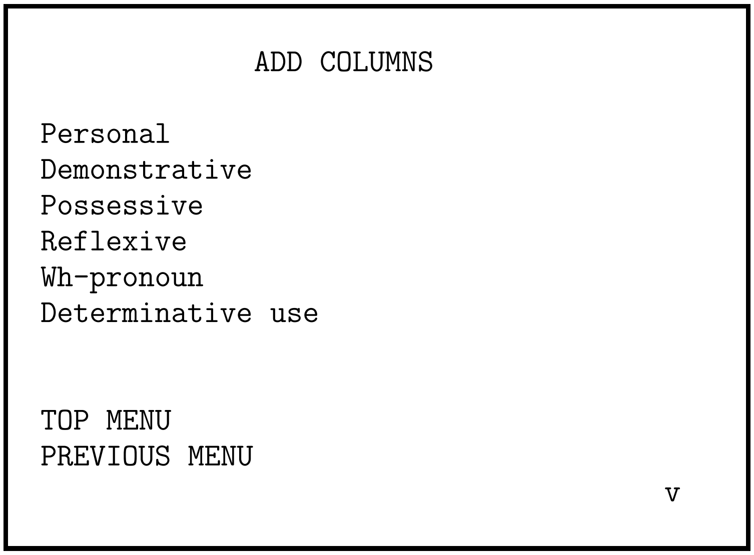
Ord_NUM (Ord_NUMLemma) `For␣numerals,␣ordinal␣`

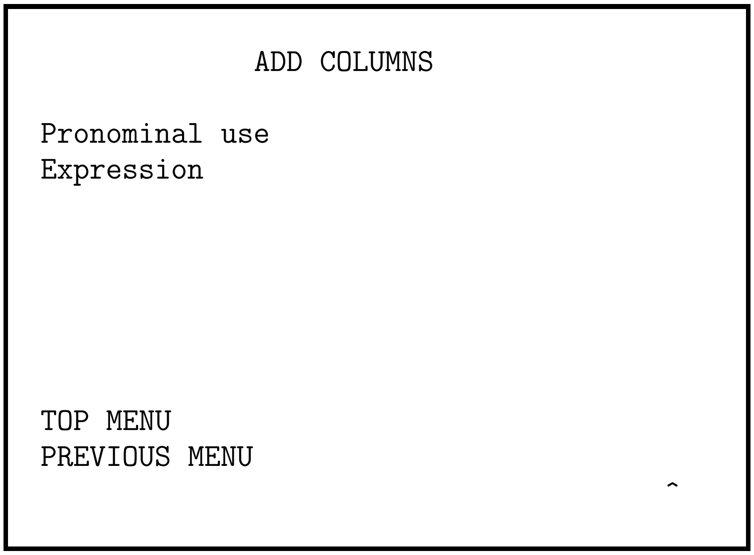
The last numeral subclassification column answers the question 'is this numeral ever used in combination with certain other words to make up a particular phrase?' An example is *ninety-nine*, which occurs in the phrase *ninety-nine times out of a hundred*, and *sixty-four* which occurs in the phrase *sixty-four thousand dollar question*. Such numerals get the code `Y`, while all other words get the code `N`. The FLEX name and description of this column are as follows:

Exp_NUM (Exp_NUMLemma) `For␣numerals,␣expression␣`

4.8 SUBCLASSIFICATION PRONOUNS

There are a total of eight pronoun subclassification columns available, and they all take the form of `Y/N␣` answers to questions (as explained in section 4.2). They are presented to you in these ADD COLUMNS windows:





The first of the seven pronoun subclassification columns answers the question 'is this lemma a personal pronoun?' Pronouns which refer directly to people or things are personal pronouns. This can include subject pronouns, such as the masculine third person singular form *he*, and object pronouns, such as the third person plural form *them*. These lemmas thus get the code `Y`; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Pers_PRON
(Pers_PRONLemma) `For␣pronouns,␣personal␣`

The second of these columns answers the question 'is this lemma a demonstrative pronoun?'. The lemmas *this*, *that*, *these* and *those*, as used in phrases like *that dress* or *this sceptr'd isle*, get the code `Y␣`under this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Dem_PRON
(Dem_PRONLemma) `For␣pronouns,␣demonstrative␣`

The third column answers the question 'is this lemma a possessive pronoun?' Lemmas like *her* or *hers*, (as in *her decision is hers and hers alone*, for example) which can indicate ownership of some sort, both get the code `Y`, while all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Poss_PRON
(Poss_PRONLemma) `For␣pronouns,␣possessive␣`

The fourth pronoun subclassification column answers the question 'is this lemma a reflexive pronoun?' Lemmas like *yourself* in *give yourself a holiday* or *ourselves* in *we saw it for ourselves* get the code `Y␣`in this column; all other lemmas get the code `N`. The FLEX name and description for this column are as follows:

Refl_PRON
(Refl_PRONLemma) `For␣pronouns,␣reflexive␣`

The fifth pronoun subclassification column answers the question 'is this lemma a wh-pronoun?' Such pronouns mostly begin with the letters *wh-* and can be used as relative pronouns (*an expert is one who knows more and more about less* *and less*) or as interrogative pronouns (*who do you love?*). All wh-pronouns, such as *who, whither, whence, whosoever, that, howsoever* and so on get the code `Y␣`under this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Wh_PRON
(Wh_PRONLemma)

`For␣pronouns,␣wh-pronoun␣`

The sixth pronoun subclassification column answers the question 'can this pronoun lemma be used as a determiner?' A determiner helps to clarify which particular noun or noun phrase you are referring to. For example, you might talk about *their* cat in order to differentiate it from *your* cat or *another* cat. As the name suggests, such words help determine what you're talking about. Any lemma that can be used as a determiner gets the code `Y␣`in this column, and every other lemma gets the code `N`. The FLEX name and description of this column are as follows:

Det_PRON
(Det_PRONLemma)

`For␣pronouns,␣determinative␣use␣`

The seventh pronoun subclassification column answers the question 'can this pronoun lemma be used pronominally?' where *pronominally* means 'instead of a noun or noun phrase *and* independent of any other noun'. For example, *other* can be used in a phrase like *choose the other*, or *mine* in a phrase like *you can't; it's mine* or *mine is better* – so both these pronouns can be used pronominally. In contrast, *my* cannot be used pronominally; it can only occur in phrases like *my book*. All pronoun lemmas which can be used pronominally get the code `Y␣`in this column, and all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Pron_PRON
(Pron_PRONLemma)

`For␣pronouns,␣pronominal␣use␣`

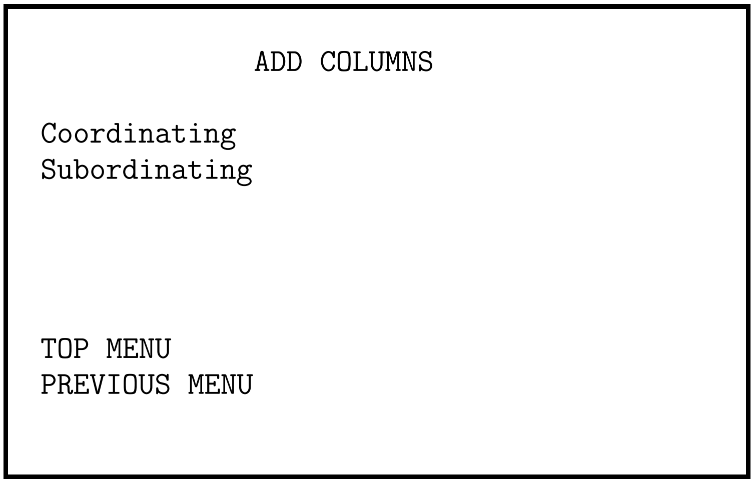
The eighth and last pronoun subclassification columns answers the question 'is this pronoun lemma only ever used in combination with certain other words to make up a particular phrase?' Examples of this are few, but one is *aught*, which occurs in the phrase *for aught I know* or *for aught I care*.

Such pronouns get the code Y, while all other words get the code N. The FLEX name and description of this column are as follows:

Exp_PRON
(Exp_PRONLemma) For␣adverbs,␣expression␣

4.9 SUBCLASSIFICATION CONJUNCTIONS

There are two conjunction subclassification columns available, and they both take the form of Y / N␣answers to questions (as described in section 4.2 above). They are presented to you in this ADD COLUMNS window:



```
                  ADD COLUMNS

 Coordinating
 Subordinating





 TOP MENU
 PREVIOUS MENU
```

The first column answers the question ‘is this lemma a coordinating conjunction?’ A conjunction like *and* is a coordinating conjunction; it can link two (or more) clauses together in such a way that they remain of equal value to each other (or to put it another way, one clause of the new sentence is not subordinate the other). The other co-ordinating conjunctions are *but* and *or*, and these three lemmas get the code Y␣under this column. All other lemmas get the code N. The FLEX name and description of this column are as follows:

Cor_C
(Cor_CLemma) For␣conjunctions,␣coordinating␣

The second of the two conjunction subclassification columns answers the question ‘is this conjunction a subordinating conjunction?’ A conjunction like *because* is a subordinating conjunction; it can link two clauses together in such a way that one clause becomes dependent on the other, as in a sentence like *he's reading a book because the television is broken*. Here *because* acts as a subordinating conjunction; the link between the two clauses is more complex than the use of a co-ordinating conjunction like *and* would imply. All conjunction lemmas which are subordinating conjunctions get the code `Y␣` in this column; all other lemmas get the code `N`. The FLEX name and description of this column are as follows:

Sub_C
(Sub_CLemma) `For␣conjunctions,␣subordinating␣`

5 ENGLISH FREQUENCY

The frequency information given in the database (that is, details of how often words occur in English) is available both for lemmas and wordforms. It is taken from the COBUILD corpus of the University of Birmingham, which in the early 1991 version extracted for and corrected by CELEX contained about 17.9 million words, taken from written sources of many kinds, and some spoken sources as well. Frequency figures are available for lemmas and for wordforms.

The starting point for calculating frequency information is the COBUILD 17.9 million word corpus: a count is made of the number of times each string occurs. This task is easy for a computer, which can quickly make a count of all the words that appear in the 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 or word classes that can be applied to each group. You can see the remaining raw counts in a COBUILD corpus types lexicon when you select the ***Freq*** column. The string count of *families* for example is 1649, while for *bank* it is 2381. 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 mistyped or misread by an Optical Character Recognition machine.

Sometimes this identification process is straightforward – the string *millstones* is only ever the plural wordform of the noun lemma *millstone*. So in this case the raw string frequency of the string *millstones* is also the frequency of the wordform *millstones*, and so in the wordform lexicon ***Cob*** column it gets the same frequency as the string.

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 *millstone* is the frequency of the wordform *millstones* plus the frequency of the wordform *millstone*. The frequency of the lemma *millstone* is the total of the two, and this is the figure given in the lemma lexicon ***Cob*** column.

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 *jumper*, it seems that 18 of the occurrences mean *item of clothing*, and 30 mean *someone who jumps*. These are the two figures given in the wordform lexicon ***Cob*** column for the two different *jumper* wordforms. Sometimes not all occurrences refer to wordforms in the database. Some may be proper nouns (surnames, for example) or typing errors, and some simply can't be disambiguated. For example *jumper* occurs 23 times in relation to a person's name. Such 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 two lemmas with the form *jumper*, the lemma frequencies are 23 (meaning *clothing*), and 47 (meaning *someone who jumps*), giving a total of 70. These lemma frequency figures are given in the lemma lexicon ***Cob*** 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 *bank*. Its basic corpus string frequency is 2381. It can either be a singular noun, or an instance of a verb, the first or second person singular form, the plural form, or the infinitive. Here is a lexicon which shows these wordforms with their word class and frequency:

Word	**Class**	**Cob**
bank	N	2310
bank	V	18
bank	V	18
bank	V	18
bank	V	18

To calculate these figures, a 100 occurrences of the string *bank* were taken from the corpus and disambiguated by hand. It turned out that 3 of the occurrences belonged to the verbal lemma, and 97 to the noun lemma. So to estimate the real frequency of the wordform belonging to the noun 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: $\frac{97}{100} \times 2381 = 2310$. Repeating this procedure gives 71 for the verb lemma. This latter figure is divided equally for the four possible wordforms: 18 each for the first person singular form, the second person singular form, the plural form and the infinitive. 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 *bank*, N is 2381, n is 100, and p is 0.97. The formula gives 78 as the deviation. This means that the true frequency for this form of *bank* is almost certain—at least 95% certain—to lie between 2232 and 2388.

Word	ClassLemma	Cob	CobDev
bank	N	2310	78
bank	V	18	78
bank	V	18	78
bank	V	18	78
bank	V	18	78

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 *bank*, 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 or headwords 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, *realize* can also be spelt *realise*, and the lemma frequency given alongside each of them is the same: 1384. The spelling frequency on the other hand shows that the spelling *realize* occurs 913 times while the spelling *realise* occurs 471. For more details about this extra layer of disambiguation, read the appropriate subsection under 'English Orthography'.

5.1 FREQUENCY INFORMATION FOR LEMMAS AND WORDFORMS

Now that the background details have been explained, the individual column names and descriptions can be formally defined. For both lemmas and wordforms, there are four columns available which express the COBUILD frequency figures in various ways.

The first column gives the plain COBUILD frequency count for each lemma or wordform. The figure given in the lemma version of the column for *collate* is 18, which means that out of the 17,900,000 words in the corpus, 18 are the word *collate* 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 *collate* the figure is 4, for *collates* it is 1. There are also 11 occurrences of *collated*, and 2 occurrences of *collating*. The FLEX name and description of this column are as follows:

Cob
(CobLemma)

```
COBUILD␣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 verb lemma *shine* (in the sense of 'be bright' or 'direct the light') is 33, and when you use it in conjunction with the COBUILD frequency figure of 567, it indicates that you can be almost certain (95% certain) that *shine* occurs in one form or another somewhere between 534 and 600 times. The FLEX name and description of this column are as follows:

CobDev
(CobDevLemma)

```
COBUILD␣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 17,900,000. This is done by dividing the normal COBUILD 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 *magnificent* is twenty in a million than it does to say that it's 351 words out of 17,900,000. 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 *barbecue* and *babysitter*, which have the 17.9 million word lemma frequencies of 26 and 9 respectively, both share the same 1 million word frequency of 1.

CobMln
(CobMlnLemma)

`COBUILD␣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 `COBUILD␣frequency␣(1,000,000)␣` described above. In place of a scale from 1 to 1,000,000, 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 26 times in the full COBUILD 17.9 million word corpus. The thing to remember is that only words which have a COBUILD 1,000,000 frequency value of two or more (or, if you prefer, only words which occur 27 or more times in the COBUILD corpus) have a logarithmic value greater than zero.

CobLog `COBUILD␣frequency,␣logarithmic␣`
(CobLogLemma)

5.1.1 FREQUENCY INFORMATION FROM WRITTEN AND SPOKEN SOURCES

About 16,600,000 words in the COBUILD corpus make up written texts, and the remaining 1,300,000 words make up spoken texts. In a sense, then, there are two other corpora you can use, one which deals with written texts only and one with spoken texts only. You can choose for yourself whether you wish to use either written or spoken figures in place of the full figures explained in the preceeding sections. The methods used in working out the figures given are the same as those described in the previous section.

The columns available for written and spoken corpus frequencies are roughly the same as those for the full corpus, with the exception of the deviation figures – they are not re-calculated for the written and spoken texts. Instead, you can use the figures given for the full corpus, though remember that when you apply them to frequencies for the written and spoken corpora, the range of error is actually larger than would otherwise be.

5.1.2 WRITTEN CORPUS INFORMATION

There are three columns which contain frequency information for the written sources in the COBUILD corpus. The figure given in the lemma version of the column for *memory* is 1524, which means that out of the 16,600,000 words in the corpus, 1524 are the word *memory* 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 *memory* the figure is 1092, and for *memories* it is 432. The FLEX name and description of this column are as follows:

CobW `COBUILD␣written␣frequency␣16.6m␣`
(CobWLemma)

The next column contains the same frequency figures as ***CobW***, except that they have been scaled down to a range of 1 to 1,000,000 instead of the usual 1 to 16,600,000. This is done by dividing the normal COBUILD written frequency for each word by the number of words in the written corpus (about 16,600,000), 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 a word is one in a million than it does to say that it's 22 words out of 16,600,000. However as you might expect, some detail is lost in the scaling-down process: all words which have 16.6 million word lemma frequencies of between 9 and 24 share the same 1 million word frequency of 1.

CobWMln
(CobWMlnLemma)

```
COBUILD␣written␣frequency␣(1,000,000)␣
```

The third and last written corpus column contains the base 10 logarithms of each ***CobWMln***, for the reasons described above in connection with the full corpus. 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 24 times in the COBUILD 16.6 million written word corpus. The thing to remember is that only words which have a ***CobWMln*** frequency value of two or more (or, if you prefer, only words which occur 25 or more times in the COBUILD corpus) have a logarithmic value greater than zero.

CobWLog
(CobWLogLemma)

```
COBUILD␣written␣frequency,␣logarithmic␣
```

5.1.3 SPOKEN CORPUS INFORMATION

There are three columns which contain frequency information for the spoken sources in the COBUILD corpus. The figure given in the lemma version of the column for *memory* is 60, which means that out of the approximately 1,300,000 words in the corpus, 60 are the word *memory* 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 *memory* the figure is 49, and for *memories* it is 11. The FLEX name and description of this column are as follows:

CobS
(CobSLemma) `COBUILD␣spoken␣frequency␣1.3m␣`

The next column contains the same frequency figures as ***CobS***, except that they have been scaled down to a range of 1 to 1,000,000 instead of the usual 1 to 1,300,000. This is done by dividing the normal COBUILD spoken frequency for each word by the number of words in the spoken corpus, and then multiplying the answer by 1,000,000.

CobSMln
(CobSMlnLemma) `COBUILD␣spoken␣frequency␣(1,000,000)␣`

The third and last spoken corpus column contains the base 10 logarithms of each ***CobSMln*** frequency, for the reasons described above in connection with the full corpus. In place of a scale from 1 to 1,000,000, 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 once in the COBUILD 1.3 million spoken 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 ***CobSMln*** frequency value of two or more (or, if you prefer, only words which occur two or more times in the COBUILD spoken corpus) have a logarithmic value greater than zero.

CobSLog
(CobSLogLemma) `COBUILD␣spoken␣frequency,␣logarithmic␣`

5.2 FREQUENCY INFORMATION FOR COBUILD CORPUS TYPES

The frequency information given in COBUILD 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 for the spoken and written texts in the corpus, as well as for British English and American English 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 simply lists the orthographic forms of all types as they occur in the COBUILD corpus. The FLEX name and description of this column are as follows:

Type `Graphemic␣transcription␣`

The second column is the basic ‘string’ count which tells you how many times each type occurs in the COBUILD corpus, which contains about 17,900,000 tokens. The FLEX name and description of this column are as follows:

Freq `Absolute␣frequency␣`

5.3 FREQUENCY INFORMATION FOR COBUILD WRITTEN CORPUS TYPES

There are four columns which contain raw string counts from the written texts in the COBUILD corpus. The first contains the frequencies of all types which occur more than once in all the written texts.

FreqW `COBUILD␣written␣frequency,␣16.6m␣`

The next column contains raw string counts from the written texts that are normal British usages, as opposed to American English or some other brand of English.

FreqWB `COBUILD␣written␣frequency,␣British␣English␣`

The third column contains raw string counts from the written texts that are normal American usages, as opposed to British English or some other brand of English.

FreqWA `COBUILD␣written␣frequency,␣American␣English␣`

The fourth column contains raw string counts from the written texts that are not normal British or American usages, but are instead from an unidentified brand of English.

FreqWU `COBUILD␣written␣frequency,␣undetermined␣origin␣`

5.4 FREQUENCY INFORMATION FOR COBUILD SPOKEN CORPUS TYPES

There are three columns which contain raw string counts from the spoken texts in the COBUILD corpus. About 1.3 million words were transcribed from recorded conversations and included in the corpus. None of conversations transcribed involved American English, so no separate figures for American English spoken types are available.

The first column contains the frequencies of all types which occur more than once in the spoken texts.

FreqS `COBUILD␣spoken␣frequency,␣1.3m␣`

The next column contains raw string counts from the written texts that are marked as normal British usages.

FreqSB `COBUILD␣spoken␣frequency,␣British␣English␣`

The third column contains raw string counts from the written texts that are not normal British usages, but are instead from an unidentified brand of English.

FreqSU `COBUILD␣spoken␣frequency,␣undetermined␣origin␣`

1 TREE DIAGRAMS & COLUMN DESCRIPTIONS

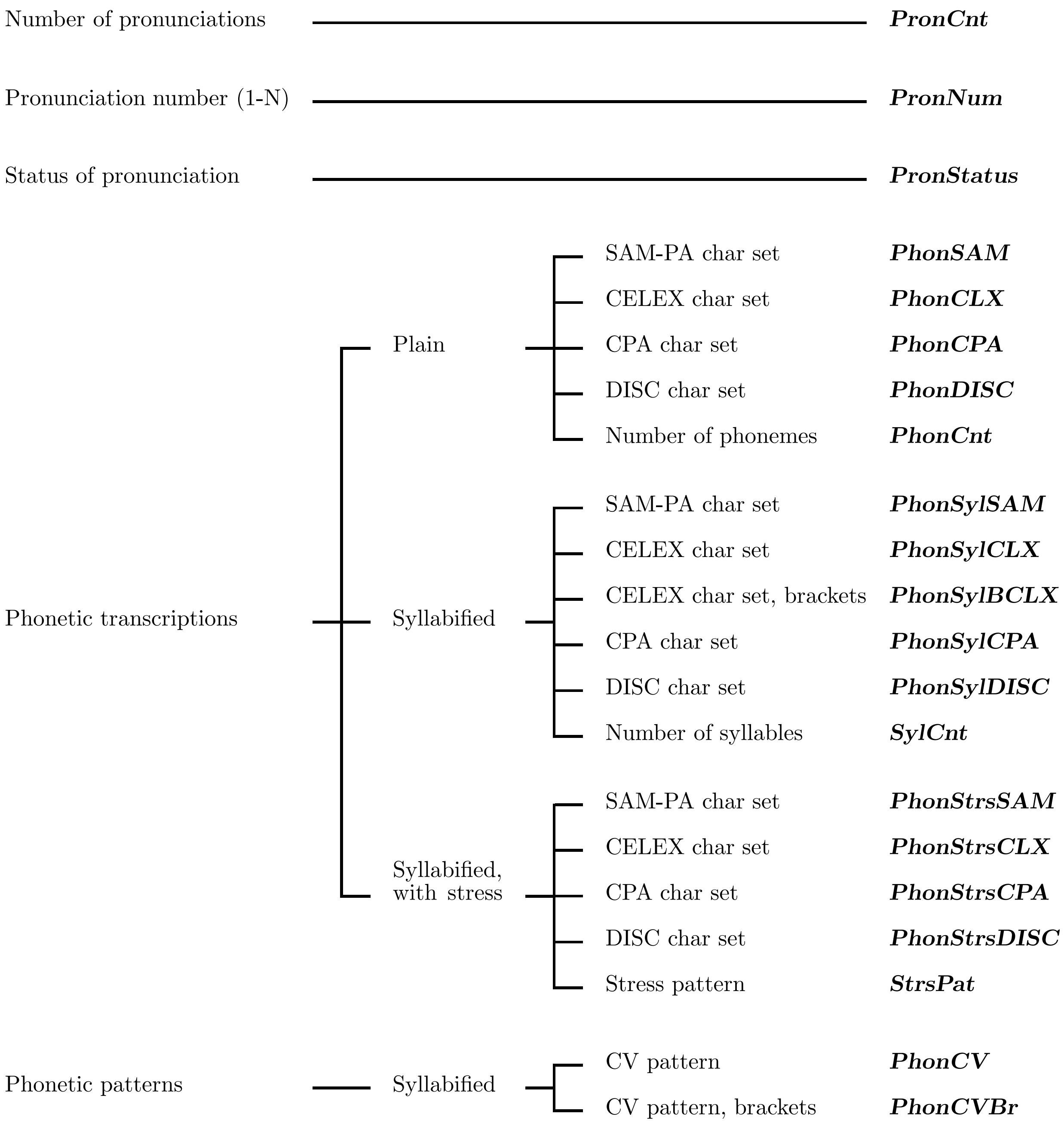
This appendix is divided into sections corresponding to the lexicon types currently available. Each section begins with a set of tree diagrams which give you an overview of the columns you can choose when you select a particular type of lexicon, and then there are technical details about each of those 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 ENGLISH LEMMAS (E25)

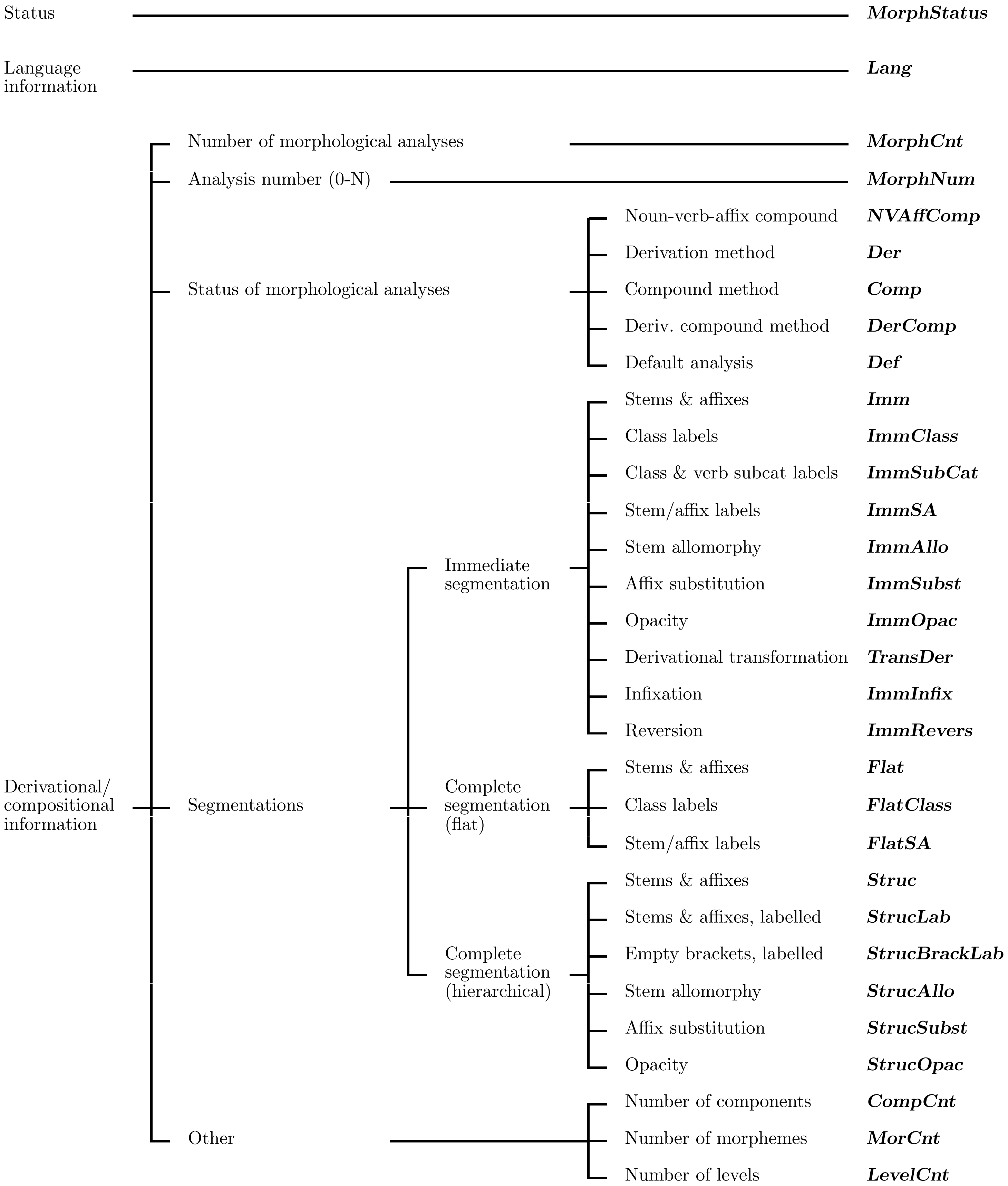
Number of spellings			***OrthoCnt***
Spelling number (1-N)			***OrthoNum***
Language code			***OrthoStatus***
Frequency of spelling	COBUILD frequency 17.9m		***CobSpellFreq***
	COBUILD 95% confidence deviation 17.9m		***CobSpellDev***
Spelling	Plain	Without diacritics	***Head***
		Without diacritics, reversed	***HeadRev***
		With diacritics	***HeadDia***
		Purely lowercase alphabetical	***HeadLow***
		Purely lowercase alphabetical, sorted	***HeadLowSort***
		Number of letters	***HeadCnt***
	Syllabified	Without diacritics	***HeadSyl***
		With diacritics	***HeadSylDia***
		Number of syllables	***HeadSylCnt***

2 PHONOLOGY OF ENGLISH LEMMAS (E25)



Number of pronunciations			***PronCnt***
Pronunciation number (1-N)			***PronNum***
Status of pronunciation			***PronStatus***
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	Syllabified	CV pattern	***PhonCV***
		CV pattern, brackets	***PhonCVBr***

3 MORPHOLOGY OF ENGLISH LEMMAS (E25)



- Status — ***MorphStatus***
- Language information — ***Lang***
- Derivational/ compositional information
  - Number of morphological analyses — ***MorphCnt***
  - Analysis number (0-N) — ***MorphNum***
  - Status of morphological analyses
    - Noun-verb-affix compound — ***NVAffComp***
    - Derivation method — ***Der***
    - Compound method — ***Comp***
    - Deriv. compound method — ***DerComp***
    - 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***
      - Opacity — ***ImmOpac***
      - Derivational transformation — ***TransDer***
      - Infixation — ***ImmInfix***
      - Reversion — ***ImmRevers***
    - 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***
      - Opacity — ***StrucOpac***
  - Other
    - Number of components — ***CompCnt***
    - Number of morphemes — ***MorCnt***
    - Number of levels — ***LevelCnt***

4 SYNTAX OF ENGLISH LEMMAS (NOUNS, VERBS) (E25)

Word class	Numeric codes	***ClassNum***
	Labels	***Class***
Subclassification nouns	Count	***C_N***
	Uncount	***Unc_N***
	Singular use	***Sing_N***
	Plural use	***Plu_N***
	Group count	***GrC_N***
	Group uncount	***GrUnC_N***
	Attributive	***Attr_N***
	Postpositive	***PostPos_N***
	Vocative	***Voc_N***
	Proper noun	***Proper_N***
	Expression	***Exp_N***
Subclassification verbs	Transitive	***Trans_V***
	Transitive & complementation	***TransComp_V***
	Intransitive	***Intrans_V***
	Ditransitive	***Ditrans_V***
	Linking verb	***Link_V***
	Phrasal	***Phr_V***
	Prepositional	***Prep_V***
	Phrasal prepositional	***PhrPrep_V***
	Expression	***Exp_V***

16 SYNTAX OF ENGLISH LEMMAS (ADJECTIVES, ADVERBS, NUMERALS, PRONOUNS, CONJUNCTIONS) (E25)

Subclassification adjectives	Ordinary	***Ord_A***
	Attributive	***Attr_A***
	Predicative	***Pred_A***
	Postpositive	***PostPos_A***
	Group uncount	***GrUnc_A***
	Expression	***Exp_A***
Subclassification adverbs	Ordinary	***Ord_ADV***
	Predicative	***Pred_ADV***
	Postpositive	***PostPos_ADV***
	Combinatory	***Comb_ADV***
	Expression	***Exp_ADV***
Subclassification numerals	Cardinal	***Card_NUM***
	Ordinal	***Ord_NUM***
	Expression	***Exp_NUM***
Subclassification pronouns	Personal	***Pers_PRON***
	Demonstrative	***Dem_PRON***
	Possessive	***Poss_PRON***
	Reflexive	***Refl_PRON***
	Wh-pronoun	***Wh_PRON***
	Determinative use	***Det_PRON***
	Pronominal use	***Pron_PRON***
	Expression	***Exp_PRON***
Subclassification conjunctions	Coordinating	***Cor_C***
	Subordinating	***Sub_C***

17 FREQUENCY OF ENGLISH LEMMAS (E25)

COBUILD all sources	COBUILD frequency 17.9m	***Cob***
	COBUILD 95% confidence deviation 17.9m	***CobDev***
	COBUILD frequency 1m	***CobMln***
	COBUILD frequency, logarithmic	***CobLog***
COBUILD written sources	COBUILD written frequency 16.6m	***CobW***
	COBUILD written frequency 1m	***CobWMln***
	COBUILD written frequency, logarithmic	***CobWLog***
COBUILD spoken sources	COBUILD spoken frequency 1.3m	***CobS***
	COBUILD spoken frequency 1m	***CobSMln***
	COBUILD spoken frequency, logarithmic	***CobSLog***

Attr_A For␣adjectives,␣attributive␣

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

Attr_N For␣nouns,␣attributive␣

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

C_N For␣nouns,␣countable␣

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

Card_NUM For␣numerals,␣cardinal␣

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

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␣S␣T␣U␣V␣		

ClassNum Word␣class,␣numeric␣

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␣		

Cob COBUILD␣frequency␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	587096␣	Maximum length:	6␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

CobDev COBUILD␣frequency␣deviation␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	546772␣	Maximum length:	6␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

CobLog COBUILD␣frequency,␣logarithmic␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	4.5149␣	Maximum length:	6␣
Characters:	. 0 1 2 3 4 5 6 7 8 9␣		

CobMln COBUILD␣frequency␣(1,000,000)␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	32727␣	Maximum length:	5␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

CobS `COBUILD␣spoken␣frequency␣1.3m␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`40718␣`	Maximum length:	`5␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSLog `COBUILD␣spoken␣frequency,␣logarithmic␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`4.4993␣`	Maximum length:	`6␣`
Characters:	`. 0 1 2 3 4 5 6 7 8 9␣`		

CobSMln `COBUILD␣spoken␣frequency␣(1,000,000)␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`31571␣`	Maximum length:	`5␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSpellDev `COBUILD␣spelling␣frequency␣deviation␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546772␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSpellFreq `COBUILD␣spelling␣frequency␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`587096␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobW `COBUILD␣written␣frequency␣16.6m␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546378␣`	Maximum length:	`6␣`
Characters:	`0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

CobWLog `COBUILD␣written␣frequency,␣logarithmic␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`4.5161␣`	Maximum length:	`6␣`
Characters:	`.␣0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

CobWMln `COBUILD␣written␣frequency␣(1,000,000)␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`32817␣`	Maximum length:	`5␣`
Characters:	`0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

Comb_ADV `For␣adverbs,␣combinatory␣`

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

Comp `Compound␣analysis␣`

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:	`4␣`	Maximum length:	`1␣`
Characters:	`0␣1␣2␣3␣4␣`		

Cor_C `For␣conjunctions,␣coordinating␣`

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

Dem_PRON `For␣pronouns,␣demonstrative␣`

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

Der `Derivation␣analysis␣`

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

DerComp `Derivational␣compound␣analysis␣`

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

Det_PRON For␣pronouns,␣determinative␣use␣

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

Ditrans_V For␣verbs,␣ditransitive␣

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

Exp_A For␣adjectives,␣expression␣

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

Exp_ADV For␣adverbs,␣expression␣

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

Exp_N For␣nouns,␣expression␣

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

Exp_NUM For␣numerals,␣expression␣

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

Exp_PRON For␣pronouns,␣expression␣

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

Exp_V For␣verbs,␣expression␣

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:	10809␣
Minimum value:	April␣	Minimum length:	1␣
Maximum value:	zoom␣	Maximum length:	29␣
Characters:	’␣+␣A␣B␣D␣E␣F␣G␣H␣I␣J␣L␣M␣O␣P␣Q␣S␣T␣U␣V␣W␣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:	`10809␣`
Minimum value:	`A␣`	Minimum length:	`1␣`
Maximum value:	`xxxx␣`	Maximum length:	`6␣`
Characters:	`A␣B␣C␣D␣I␣N␣O␣P␣Q␣S␣T␣V␣x␣`		

FlatSA `Flat␣segmentation,␣stem/affix␣labels␣`

Type:	`character␣`	Null values:	`10809␣`
Minimum value:	`AA␣`	Minimum length:	`1␣`
Maximum value:	`SSSA␣`	Maximum length:	`6␣`
Characters:	`A␣F␣S␣`		

GrC_N `For␣nouns,␣group␣countable␣`

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

GrUnc_N `For␣nouns,␣group␣uncountable␣`

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

Head Headword␣

Type:	character␣	Null values:	0␣
Minimum value:	’d␣	Minimum length:	1␣
Maximum value:	zoophyte␣	Maximum length:	20␣
Characters:	’␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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:	20␣	Maximum length:	2␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

HeadDia Headword,␣diacritics␣

Type:	character␣	Null values:	0␣
Minimum value:	’d␣	Minimum length:	1␣
Maximum value:	épée␣	Maximum length:	20␣
Characters:	’␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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␣à␣â␣ç␣è␣é␣ê␣ï␣ñ␣ô␣ü␣		

HeadLow Headword,␣lowercase,␣alphabetical␣

Type:	character␣	Null values:	0␣
Minimum value:	a␣	Minimum length:	1␣
Maximum value:	zoophyte␣	Maximum length:	20␣
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:	ttuu␣	Maximum length:	20␣
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:	'oht␣	Minimum length:	1␣
Maximum value:	zzuf␣	Maximum length:	20␣
Characters:	'␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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:	'd␣	Minimum length:	1␣
Maximum value:	zoom␣	Maximum length:	27␣
Characters:	'␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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:	8␣	Maximum length:	1␣
Characters:	1␣2␣3␣4␣5␣6␣7␣8␣		

HeadSylDia Headword,␣syllabified,␣diacritics␣

Type:	character␣	Null values:	0␣
Minimum value:	’d␣	Minimum length:	1␣
Maximum value:	é-pée␣	Maximum length:	27␣
Characters:	’␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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␣à␣â␣ç␣è␣é␣ê␣ï␣ñ␣ô␣ü␣		

IdNum Lemma␣number␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	31295␣	Maximum length:	5␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

Imm Immediate␣segmentation␣

Type:	character␣	Null values:	10809␣
Minimum value:	April␣	Minimum length:	1␣
Maximum value:	zoom␣	Maximum length:	22␣
Characters:	’␣+␣A␣B␣D␣E␣F␣G␣H␣I␣J␣L␣M␣O␣P␣Q␣S␣T␣U␣V␣W␣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:	10809␣
Minimum value:	B␣	Minimum length:	1␣
Maximum value:	Z␣	Maximum length:	1␣
Characters:	B␣C␣D␣F␣N␣Z␣		

ImmClass Immediate␣segmentation,␣word␣class␣labels␣

Type:	character␣	Null values:	10809␣
Minimum value:	A␣	Minimum length:	1␣
Maximum value:	xxx␣	Maximum length:	4␣
Characters:	A␣B␣C␣D␣I␣N␣O␣P␣Q␣S␣T␣V␣x␣		

ImmInfix Infixation,␣top␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	N␣	Maximum length:	1␣
Characters:	N␣		

ImmOpac Opacity,␣top␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	Y␣	Maximum length:	1␣
Characters:	N␣Y␣		

ImmRevers Reversion,␣top␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	Y␣	Maximum length:	1␣
Characters:	N␣Y␣		

ImmSA Immediate␣segmentation,␣stem/affix␣labels␣

Type:	character␣	Null values:	10809␣
Minimum value:	AA␣	Minimum length:	1␣
Maximum value:	SSA␣	Maximum length:	4␣
Characters:	A␣F␣S␣		

ImmSubCat Immediate␣segmentation,␣subcat␣labels␣

Type:	character␣	Null values:	10809␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	xxx␣	Maximum length:	4␣
Characters:	0␣1␣2␣3␣A␣B␣C␣D␣I␣N␣O␣P␣Q␣S␣T␣x␣		

ImmSubst Affix␣substitution,␣top␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	Y␣	Maximum length:	1␣
Characters:	N␣Y␣		

Intrans_V For␣verbs,␣intransitive␣

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

Lang Language␣information␣

Type:	character␣	Null values:	59173␣
Minimum value:	A␣	Minimum length:	1␣
Maximum value:	S␣	Maximum length:	1␣
Characters:	A␣B␣D␣F␣G␣I␣L␣S␣		

LevelCnt Number␣of␣morphological␣levels␣

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

Link_V `For␣verbs,␣linking␣verb␣`

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

MorCnt `Number␣of␣morphemes␣`

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␣`		

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 5 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:	`Z␣`	Maximum length:	`1␣`
Characters:	`C F I M O R U Z␣`		

NVAffComp Noun-verb-affix␣compound␣

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

Ord_A For␣adjectives,␣ordinary␣

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

Ord_ADV For␣adverbs,␣ordinary␣

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

Ord_NUM For␣numerals,␣ordinal␣

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:	5␣	Maximum length:	1␣
Characters:	1␣2␣3␣4␣5␣		

OrthoNum Spelling␣number␣

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

OrthoStatus Status␣of␣spelling␣

Type:	character␣	Null values:	0␣
Minimum value:	A␣	Minimum length:	1␣
Maximum value:	B␣	Maximum length:	1␣
Characters:	A␣B␣		

Pers_PRON For␣pronouns,␣personal␣

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

PhonCLX Phon.␣headword,␣CELEX␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	&.N.S.@.s.␣	Minimum length:	2␣
Maximum value:	z.u:.m.␣	Maximum length:	38␣
Characters:	&␣*␣,␣.␣3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonCnt Headword,␣number␣of␣phonemes␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	19␣	Maximum length:	2␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

PhonCPA Phon.␣headword,␣CPA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	@.␣	Minimum length:	2␣
Maximum value:	z.u:.m.␣	Maximum length:	38␣
Characters:	*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonCV Headword,␣phon.␣CV␣pattern␣

Type:	character␣	Null values:	0␣
Minimum value:	C␣	Minimum length:	1␣
Maximum value:	VVCC-CVCC␣	Maximum length:	26␣
Characters:	- C␣S␣V␣		

PhonCVBr Headword,␣phon.␣CV␣pattern,␣with␣brackets␣

Type:	character␣	Null values:	0␣
Minimum value:	[CCCVCCC]␣	Minimum length:	3␣
Maximum value:	[V][VC]␣	Maximum length:	35␣
Characters:	C␣S␣V␣[␣]␣		

PhonDISC Phon.␣headword,␣DISC␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	#␣	Minimum length:	1␣
Maximum value:	~t~t␣	Maximum length:	19␣
Characters:	#␣$␣0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣@␣C␣D␣E␣F␣H␣I␣J␣N␣P␣Q␣ R␣S␣T␣U␣V␣Z␣_␣b␣c␣d␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣q␣r␣s␣ t␣u␣v␣w␣x␣z␣{␣~␣		

PhonSAM Phon.␣headword,␣SAM-PA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	3:.T.␣	Minimum length:	2␣
Maximum value:	{~.n.Z.eI.n.j.u:.␣	Maximum length:	38␣
Characters:	*␣,␣.␣3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"&-b@-’lI-S@-nIst␣	Minimum length:	2␣
Maximum value:	zU-’O-l@-dZIst␣	Maximum length:	29␣
Characters:	"␣&␣’␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"@.’mO.r@l␣	Minimum length:	2␣
Maximum value:	zU.’O.l@.J/Ist␣	Maximum length:	30␣
Characters:	"␣’␣*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"#-’mEn␣	Minimum length:	2␣
Maximum value:	~n-’t~t␣	Maximum length:	29␣
Characters:	"␣#␣$␣’␣- 0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣@␣C␣D␣E␣F␣H␣I␣J␣ N␣P␣Q␣R␣S␣T␣U␣V␣Z␣_␣b␣c␣d␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣ q␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"3:-TI␣	Minimum length:	2␣
Maximum value:	{z-"bE-stQs␣	Maximum length:	29␣
Characters:	"␣%␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

PhonSylCLX Syll.␣phon.␣headword,␣CELEX␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	&-SI␣	Minimum length:	1␣
Maximum value:	zu:m␣	Maximum length:	26␣
Characters:	&␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	[&N][S@s]␣	Minimum length:	3␣
Maximum value:	[zu:m]␣	Maximum length:	35␣
Characters:	&␣*␣,␣3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣[␣]␣a␣b␣d␣e␣ f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonSylCPA Syll.␣phon.␣headword,␣CPA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	@␣	Minimum length:	1␣
Maximum value:	zu:m␣	Maximum length:	27␣
Characters:	*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonSylDISC Syll.␣phon.␣headword,␣DISC␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	#␣	Minimum length:	1␣
Maximum value:	~n-t~t␣	Maximum length:	26␣
Characters:	#␣$␣- 0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣@␣C␣D␣E␣F␣H␣I␣J␣N␣P␣ Q␣R␣S␣T␣U␣V␣Z␣_␣b␣c␣d␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣q␣r␣ s␣t␣u␣v␣w␣x␣z␣{␣~␣		

PhonSylSAM Syll.␣phon.␣headword,␣SAM-PA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	3:-TI␣	Minimum length:	1␣
Maximum value:	{~n-ZeI-nju:␣	Maximum length:	26␣
Characters:	*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

Phr_V For␣verbs,␣phrasal␣verb␣

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

PhrPrep_V `For␣verbs,␣phrasal␣prepositional␣verb␣`

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

Plu_N `For␣nouns,␣plural␣use␣`

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

Poss_PRON `For␣pronouns,␣possessive␣`

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

PostPos_A `For␣adjectives,␣postpositive␣`

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

PostPos_ADV `For␣adverbs,␣postpositive␣`

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

PostPos_N For␣nouns,␣postpositive␣

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

Pred_A For␣adjectives,␣predicative␣

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

Pred_ADV For␣adverbs,␣predicative␣

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

Def Default␣analysis␣

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

Prep_V For␣verb,␣prepositional␣verb␣

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

Pron_PRON `For␣pronouns,␣pronominal␣use␣`

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

PronCnt `Number␣of␣pronunciations␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`1␣`	Minimum length:	`1␣`
Maximum value:	`40␣`	Maximum length:	`2␣`
Characters:	`0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

PronNum `Pronunciation␣number␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`1␣`	Minimum length:	`1␣`
Maximum value:	`40␣`	Maximum length:	`2␣`
Characters:	`0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

PronStatus `Status␣of␣pronunciation␣`

Type:	`character␣`	Null values:	`0␣`
Minimum value:	`P␣`	Minimum length:	`1␣`
Maximum value:	`S␣`	Maximum length:	`1␣`
Characters:	`P␣S␣`		

Proper_N `For␣nouns,␣proper␣noun␣`

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

Refl_PRON `For␣pronouns,␣reflexive␣`

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

Sing_N `For␣nouns,␣singular␣use␣`

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

StrsPat `Headword,␣stress␣pattern␣`

Type:	character␣	Null values:	0␣
Minimum value:	000100␣	Minimum length:	1␣
Maximum value:	2102␣	Maximum length:	8␣
Characters:	0␣1␣2␣		

Struc `Structured␣segmentation␣`

Type:	character␣	Null values:	10809␣
Minimum value:	((((((confide),(ent)),␣ (ence)),((trick)))))␣	Minimum length:	3␣
Maximum value:	(zoom)␣	Maximum length:	51␣
Characters:	’␣(␣)␣,␣A␣B␣D␣E␣F␣G␣H␣I␣J␣L␣M␣O␣P␣Q␣S␣T␣U␣V␣ W␣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:	10809␣
Minimum value:	B␣	Minimum length:	1␣
Maximum value:	Z␣	Maximum length:	2␣
Characters:	B␣C␣D␣F␣N␣Z␣		

StrucBrackLab Structured␣segmentation,␣word␣class␣labels␣only␣

Type:	character␣	Null values:	10809␣
Minimum value:	((␣)[V])[N]␣	Minimum length:	5␣
Maximum value:	()[V]␣	Maximum length:	75␣
Characters:	(␣)␣,␣.␣A␣B␣C␣D␣I␣N␣O␣P␣Q␣S␣T␣V␣[␣]␣x␣|␣		

StrucLab Structured␣segmentation,␣word␣class␣labels␣

Type:	character␣	Null values:	10809␣
Minimum value:	((((((confide)[V],(ent)␣ [A|V.])[A],(ence)␣ [N|A.])[N],␣ ((trick)[V])[N])[N])␣ [N])[V]␣	Minimum length:	6␣
Maximum value:	(zoom)[V]␣	Maximum length:	99␣
Characters:	’␣(␣)␣,␣.␣A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣L␣M␣N␣O␣P␣Q␣S␣ T␣U␣V␣W␣[␣]␣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␣|␣		

StrucOpac Opacity,␣any␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	Y␣	Maximum length:	1␣
Characters:	N␣Y␣		

StrucSubst Affix␣substitution,␣any␣level␣

Type:	character␣	Null values:	10809␣
Minimum value:	N␣	Minimum length:	1␣
Maximum value:	Y␣	Maximum length:	1␣
Characters:	N␣Y␣		

Sub_C For␣conjunctions,␣subordinating␣

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

SylCnt Headword,␣number␣of␣phonetic␣syllables␣

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

Trans_V For␣verbs,␣transitive␣

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

TransComp_V For␣verbs,␣transitive␣plus␣complementation␣

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

TransDer Derivational␣transformation,␣top␣level␣

Type:	character␣	Null values:	25887␣
Minimum value:	#␣	Minimum length:	1␣
Maximum value:	-yupon+i#␣	Maximum length:	20␣
Characters:	#␣+␣- a␣b␣c␣d␣e␣f␣g␣h␣i␣k␣l␣m␣n␣o␣p␣q␣r␣s␣t␣u␣ v␣w␣x␣y␣z␣		

Unc_N For␣nouns,␣uncountable␣

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

Voc_N For␣nouns,␣vocative␣

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

Wh_PRON For␣pronouns,␣wh-pronoun␣

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

18 ORTHOGRAPHY OF ENGLISH WORDFORMS (E25)

Number of spellings			***OrthoCnt***
Spelling number (1-N)			***OrthoNum***
Language code			***OrthoStatus***
Frequency of spelling	COBUILD frequency 17.9m		***CobSpellFreq***
	COBUILD 95% confidence deviation 17.9m		***CobSpellDev***
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***

19 PHONOLOGY OF ENGLISH WORDFORMS (E25)

Number of pronunciations			***PronCnt***
Pronunciation number (1-N)			***PronNum***
Status of pronunciation			***PronStatus***
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	Syllabified	CV pattern	***PhonCV***
		CV pattern, brackets	***PhonCVBr***

20 MORPHOLOGY OF ENGLISH WORDFORMS (E25)

Lemma information	Numeric id	***IDNumLemma***
	Orthography	**ORTHOGRAPHY OF ENGLISH LEMMAS**
	Phonology	**PHONOLOGY OF ENGLISH LEMMAS**
	Morphology	**MORPHOLOGY OF ENGLISH LEMMAS**
	Syntax	**SYNTAX OF ENGLISH LEMMAS**
	Frequency	**FREQUENCY OF ENGLISH LEMMAS**
		(See the information in these diagrams for the available columns)
Inflectional features	Singular	***Sing***
	Plural	***Plu***
	Positive	***Pos***
	Comparative	***Comp***
	Superlative	***Sup***
	Infinitive	***Inf***
	Participle	***Part***
	Present tense	***Pres***
	Past tense	***Past***
	1st person verb	***Sin1***
	2nd person verb	***Sin2***
	3rd person verb	***Sin3***
	Rare form	***Rare***
Type of flection		***FlectType***
Inflectional Transformation		***TransInfl***

21 FREQUENCY OF ENGLISH WORDFORMS (E25)

COBUILD all sources	COBUILD frequency 17.9m	***Cob***
	COBUILD 95% confidence deviation 17.9m	***CobDev***
	COBUILD frequency 1m	***CobMln***
	COBUILD frequency, logarithmic	***CobLog***
COBUILD written sources	COBUILD written frequency 16.6m	***CobW***
	COBUILD written frequency 1m	***CobWMln***
	COBUILD written frequency, logarithmic	***CobWLog***
COBUILD spoken sources	COBUILD spoken frequency 1.3m	***CobS***
	COBUILD spoken frequency 1m	***CobSMln***
	COBUILD spoken frequency, logarithmic	***CobSLog***

Cob `COBUILD␣frequency␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546774␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobDev `COBUILD␣frequency␣deviation␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546772␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobLog `COBUILD␣frequency,␣logarithmic␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`4.484␣`	Maximum length:	`6␣`
Characters:	`. 0 1 2 3 4 5 6 7 8 9␣`		

CobMln `COBUILD␣frequency␣(1,000,000)␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`30479␣`	Maximum length:	`5␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobS `COBUILD␣spoken␣frequency␣1.3m␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`36388␣`	Maximum length:	`5␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSLog `COBUILD␣spoken␣frequency,␣logarithmic␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`4.4505␣`	Maximum length:	`6␣`
Characters:	`. 0 1 2 3 4 5 6 7 8 9␣`		

CobSMln `COBUILD␣spoken␣frequency␣(1,000,000)␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`28214␣`	Maximum length:	`5␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSpellDev `COBUILD␣spelling␣frequency␣deviation␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546773␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobSpellFreq `COBUILD␣spelling␣frequency␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`546774␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobW `COBUILD␣written␣frequency␣17.4m␣`

Type:	`numeric␣`	Null values:	`0␣`
Minimum value:	`0␣`	Minimum length:	`1␣`
Maximum value:	`516413␣`	Maximum length:	`6␣`
Characters:	`0 1 2 3 4 5 6 7 8 9␣`		

CobWLog COBUILD␣written␣frequency,␣logarithmic␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	4.4916␣	Maximum length:	6␣
Characters:	.␣0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

CobWMln COBUILD␣written␣frequency␣(1,000,000)␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	31017␣	Maximum length:	5␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

Comp Inflectional␣feature:␣comparative␣

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:	P␣	Minimum length:	1␣
Maximum value:	s␣	Maximum length:	4␣
Characters:	1␣2␣3␣P␣S␣X␣a␣b␣c␣e␣i␣p␣r␣s␣		

IdNum Word␣number␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	100669␣	Maximum length:	6␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

IdNumLemma Lemma␣number␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	31295␣	Maximum length:	5␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

Inf Inflectional␣feature:␣infinitive␣

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:	5␣	Maximum length:	1␣
Characters:	1␣2␣3␣4␣5␣		

OrthoNum Spelling␣number␣

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

OrthoStatus Status␣of␣spelling␣

Type:	character␣	Null values:	0␣
Minimum value:	A␣	Minimum length:	1␣
Maximum value:	B␣	Maximum length:	1␣
Characters:	A␣B␣		

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:	0␣
Minimum value:	&.N.S.@.s.␣	Minimum length:	2␣
Maximum value:	z.u:.z.␣	Maximum length:	38␣
Characters:	&␣*␣,␣.␣3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonCnt Wordform,␣number␣of␣phonemes␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	19␣	Maximum length:	2␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

PhonCPA Phon.␣wordform,␣CPA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	@.␣	Minimum length:	2␣
Maximum value:	z.u:.z.␣	Maximum length:	38␣
Characters:	*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonCV Wordform,␣phon.␣CV␣pattern␣

Type:	character␣	Null values:	0␣
Minimum value:	C␣	Minimum length:	1␣
Maximum value:	VVCCC␣	Maximum length:	27␣
Characters:	- C␣S␣V␣		

PhonCVBr Wordform,␣phon.␣CV␣pattern,␣with␣brackets␣

Type:	character␣	Null values:	0␣
Minimum value:	[CCCC]␣	Minimum length:	3␣
Maximum value:	[V][VC]␣	Maximum length:	35␣
Characters:	C␣S␣V␣[␣]␣		

PhonDISC Phon.␣wordform,␣DISC␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	#␣	Minimum length:	1␣
Maximum value:	~t~ts␣	Maximum length:	19␣
Characters:	# $ 0 1 2 3 4 5 6 7 8 9 @ C D E F H I J N P Q␣ R S T U V Z _ b c d f g h i j k l m n p q r s␣ t u v w x z { ~␣		

PhonSAM Phon.␣wordform,␣SAM-PA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	3:.D.z.␣	Minimum length:	2␣
Maximum value:	{~.n.Z.eI.n.j.u:.z.␣	Maximum length:	38␣
Characters:	*␣,␣.␣3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"&-b@-'lI-S@-nIst␣	Minimum length:	2␣
Maximum value:	zU-'O-l@-dZIsts␣	Maximum length:	29␣
Characters:	"␣&␣'␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"@.'mO.r@l␣	Minimum length:	2␣
Maximum value:	zU.'O.l@.J/Ists␣	Maximum length:	30␣
Characters:	"␣'␣*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"#-'mEn␣	Minimum length:	2␣
Maximum value:	~n-'t~ts␣	Maximum length:	29␣
Characters:	"␣#␣$␣'␣- 0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣@␣C␣D␣E␣F␣H␣I␣J␣ N␣P␣Q␣R␣S␣T␣U␣V␣Z␣_␣b␣c␣d␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣ q␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	"3:-TI␣	Minimum length:	2␣
Maximum value:	{z-"bE-stQs␣	Maximum length:	29␣
Characters:	"␣%␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣ e␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣{␣~␣		

PhonSylCLX Syll.␣phon.␣wordform,␣CELEX␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	&-SI␣	Minimum length:	1␣
Maximum value:	zu:z␣	Maximum length:	27␣
Characters:	&␣*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

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

Type:	character␣	Null values:	0␣
Minimum value:	[&N][S@s]␣	Minimum length:	3␣
Maximum value:	[zu:z]␣	Maximum length:	36␣
Characters:	&␣*␣,␣3␣:␣@␣A␣D␣E␣I␣N␣O␣S␣T␣U␣V␣Z␣[␣]␣a␣b␣d␣e␣ f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonSylCPA Syll.␣phon.␣wordform,␣CPA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	@␣	Minimum length:	1␣
Maximum value:	zu:z␣	Maximum length:	28␣
Characters:	*␣,␣.␣/␣:␣@␣A␣D␣E␣I␣J␣N␣O␣S␣T␣U␣Z␣^␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣o␣p␣r␣s␣t␣u␣v␣w␣x␣z␣~␣		

PhonSylDISC Syll.␣phon.␣wordform,␣DISC␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	#␣	Minimum length:	1␣
Maximum value:	~n-t~ts␣	Maximum length:	26␣
Characters:	#␣$␣- 0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣@␣C␣D␣E␣F␣H␣I␣J␣N␣P␣ Q␣R␣S␣T␣U␣V␣Z␣_␣b␣c␣d␣f␣g␣h␣i␣j␣k␣l␣m␣n␣p␣q␣r␣ s␣t␣u␣v␣w␣x␣z␣{␣~␣		

PhonSylSAM Syll.␣phon.␣wordform,␣SAM-PA␣charset␣

Type:	character␣	Null values:	0␣
Minimum value:	3:-TI␣	Minimum length:	1␣
Maximum value:	{~n-ZeI-nju:z␣	Maximum length:	27␣
Characters:	*␣,␣- 3␣:␣@␣A␣D␣E␣I␣N␣O␣Q␣S␣T␣U␣V␣Z␣a␣b␣d␣e␣f␣ g␣h␣i␣j␣k␣l␣m␣n␣p␣r␣s␣t␣u␣v␣w␣x␣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␣		

PronCnt Number␣of␣pronunciations␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	40␣	Maximum length:	2␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

PronNum Pronunciation␣number␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	40␣	Maximum length:	2␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

PronStatus Status␣of␣pronunciation␣

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

Rare Inflectional␣feature:␣rare␣form␣

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:	`0␣`
Minimum value:	`000100␣`	Minimum length:	`1␣`
Maximum value:	`2102␣`	Maximum length:	`8␣`
Characters:	`0␣1␣2␣`		

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:	1␣	Minimum length:	1␣
Maximum value:	8␣	Maximum length:	1␣
Characters:	1␣2␣3␣4␣5␣6␣7␣8␣		

TransInfl Inflectional␣transformation␣

Type:	character␣	Null values:	3368␣
Minimum value:	@␣	Minimum length:	1␣
Maximum value:	@-y+iest␣	Maximum length:	11␣
Characters:	+␣- @␣b␣d␣e␣f␣g␣i␣k␣l␣m␣n␣p␣r␣s␣t␣v␣y␣z␣		

Word Word␣

Type:	character␣	Null values:	0␣
Minimum value:	'd␣	Minimum length:	1␣
Maximum value:	zoos␣	Maximum length:	20␣
Characters:	' - A B C D E F G H I J K L M N O P Q R S T␣ U V W Y 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:	`20␣`	Maximum length:	`2␣`
Characters:	`0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣`		

WordDia `Word,␣diacritics␣`

Type:	`character␣`	Null values:	`0␣`
Minimum value:	`’d␣`	Minimum length:	`1␣`
Maximum value:	`épées␣`	Maximum length:	`20␣`
Characters:	`  ’␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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␣à␣â␣ç␣è␣é␣ê␣ï␣ñ␣ô␣ü␣`		

WordLow `Word,␣lowercase,␣alphabetical␣`

Type:	`character␣`	Null values:	`0␣`
Minimum value:	`a␣`	Minimum length:	`1␣`
Maximum value:	`zoos␣`	Maximum length:	`20␣`
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:	`ttuu␣`	Maximum length:	`20␣`
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:	'oht␣	Minimum length:	1␣
Maximum value:	zzuf␣	Maximum length:	20␣
Characters:	'␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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:	'd␣	Minimum length:	1␣
Maximum value:	zoos␣	Maximum length:	27␣
Characters:	'␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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:	8␣	Maximum length:	1␣
Characters:	1␣2␣3␣4␣5␣6␣7␣8␣		

WordSylDia Word,␣syllabified,␣diacritics␣

Type:	character␣	Null values:	0␣
Minimum value:	'd␣	Minimum length:	1␣
Maximum value:	é-pées␣	Maximum length:	27␣
Characters:	'␣- A␣B␣C␣D␣E␣F␣G␣H␣I␣J␣K␣L␣M␣N␣O␣P␣Q␣R␣S␣T␣ U␣V␣W␣Y␣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␣à␣â␣ç␣è␣é␣ê␣ï␣ñ␣ô␣ü␣		

22 ENGLISH COBUILD CORPUS TYPES (E25)

Orthography	Graphemic transcription	***Type***
COBUILD all sources	Absolute frequency	***Freq***
COBUILD written sources	Written frequency	***FreqW***
	British English, written frequency	***FreqWB***
	American English, written frequency	***FreqWA***
	Undetermined origin, written frequency	***FreqWU***
COBUILD spoken sources	Spoken frequency	***FreqS***
	British English, spoken frequency	***FreqSB***
	Undetermined origin, spoken frequency	***FreqSU***

Freq Absolute␣frequency␣

Type:	numeric␣	Null values:	0␣
Minimum value:	1␣	Minimum length:	1␣
Maximum value:	1093546␣	Maximum length:	7␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

FreqS COBUILD␣spoken␣frequency,␣1.3m␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	60721␣	Maximum length:	5␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

FreqSB COBUILD␣spoken␣frequency,␣British␣English␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	38299␣	Maximum length:	5␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

FreqSU COBUILD␣spoken␣frequency,␣undetermined␣origin␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	22422␣	Maximum length:	5␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

FreqW COBUILD␣written␣frequency,␣17.4m␣

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	1032825␣	Maximum length:	7␣
Characters:	0 1 2 3 4 5 6 7 8 9␣		

FreqWA `COBUILD␣written␣frequency,␣American␣English␣`

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	25899␣	Maximum length:	5␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

FreqWB `COBUILD␣written␣frequency,␣British␣English␣`

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	970099␣	Maximum length:	6␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

FreqWU `COBUILD␣written␣frequency,␣undetermined␣origin␣`

Type:	numeric␣	Null values:	0␣
Minimum value:	0␣	Minimum length:	1␣
Maximum value:	36827␣	Maximum length:	5␣
Characters:	0␣1␣2␣3␣4␣5␣6␣7␣8␣9␣		

Type `Graphemic␣transcription␣`

Type:	character␣	Null values:	0␣
Minimum value:	0-40mm␣	Minimum length:	1␣
Maximum value:	zzzzzzrrrrr␣	Maximum length:	72␣
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	**p**ut	**p**at	**P**akt	p	p	p	p
b	**b**ad	**b**ad	**B**ad	b	b	b	b
t	**t**ak	**t**ack	**T**ag	t	t	t	t
d	**d**ak	**d**ad	**d**ann	d	d	d	d
k	**k**at	**c**ad	**k**alt	k	k	k	k
ɡ	**g**oal	**g**ame	**G**ast	g	g	g	g
ŋ	la**ng**	ba**ng**	Kla**ng**	N	N	N	N
m	**m**at	**m**ad	**M**aß	m	m	m	m
n	**n**at	**n**at	**N**aht	n	n	n	n
l	**l**at	**l**ad	**L**ast	l	l	l	l
r, ʀ	**r**at, late**r**	**r**at	**R**atte	r	r	r	r
f	**f**iets	**f**at	**f**alsch	f	f	f	f
v	**v**at	**v**at	**W**elt	v	v	v	v
θ		**th**in		T	T	T	T
ð		**th**en		D	D	D	D
s	**s**ap	**s**ap	Ga**s**	s	s	s	s
z	**z**at	**z**ap	**S**uppe	z	z	z	z
ʃ	**sj**aal	**sh**eep	**Sch**iff	S	S	S	S
ʒ	rava**g**e	mea**s**ure	**G**enie	Z	Z	Z	Z
j	**j**as	**y**ank	**J**acke	j	j	j	j
x, ç	li**ch**t, **g**aat	lo**ch**	Ba**ch**, i**ch**	x	x	x	x
ɣ	re**g**en			G	G	G	G
h	**h**ad	**h**ad	**H**and	h	h	h	h
w		**w**hy	**w**aterproof	w	w	w	w
ʋ	**w**at			w	w	w	w
pf			**Pf**erd	pf	pf	pf	+
ts			**Z**ahl	ts	ts	C/	=
ʧ		**ch**eap	Ma**tsch**	tS	tS	T/	J
ʤ	**jazz**	**j**eep	**G**in	dZ	dZ	J/	_
ŋ̩		baco**n**		N,	N,	N,	C
m̩		idealis**m**		m,	m,	m,	F
n̩		burde**n**		n,	n,	n,	H
l̩		dang**l**e		l,	l,	l,	P
*		fathe**r** *(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	l**ie**p	b**ea**n	L**ie**d	i:	i:	i:	i
iːː	anal**y**se			i::	i::	i::	!
ɑː		b**ar**n	Adv**a**ntage	A:	A:	A:	#
aː	l**aa**t		kl**a**r	a:	a:	a:	a
ɔː		b**or**n	**A**llroundman	O:	O:	O:	$
uː	b**oe**k	b**oo**n	H**u**t	u:	u:	u:	u
ɜː		b**ur**n	Teamw**or**k	3:	3:	@:	3
yː	b**uu**t		f**ü**r	y:	y:	y:	y
yːː	centrif**u**ge			y::	y::	y::	(
ː	sc**e**ne		K**ä**se	E:	E:	E:	)
œː	fr**eu**le			/:	U:	Q:	*
ː	z**o**ne			Q:	O:	o:	<
eː	l**ee**g		M**e**hl	e:	e:	e:	e
øː	d**eu**k		M**ö**bel	|:	&:	q:	|
oː	b**oo**m		B**oo**t	o:	o:	o:	o
e		b**ay**	N**a**tive	eI	eI	e/	1
a		b**uy**	Sh**y**lock	aI	aI	a/	2
ɔ		b**oy**	Playb**oy**	OI	OI	o/	4
əʊ		n**o**		@U	@U	O/	5
aʊ		br**ow**	Allr**ou**ndsportler	aU	aU	A/	6
ə		p**ee**r		I@	I@	I/	7
ə		p**air**		E@	E@	E/	8
ʊə		p**oo**r		U@	U@	U/	9
i	w**ij**s			EI	EI	y/	K
œy	h**ui**s			/I	UI	q/	L
ɑu	k**ou**d			Au	AU	A/	M
ai			w**ei**t	ai	ai	a/	W
au			H**au**t	au	au	A/	B
ɔy			fr**eu**t	Oy	Oy	o/	X

Disc Computer Phonetic Codes
long vowels and diphthongs

IPA	Dutch	English	German	SAM-PA	CELEX	CPA	DISC
	l**i**p	p**i**t	M**i**tte	`I`	`I`	`I`	`I`
ʏ			Pf**ü**tze	`Y`	`Y`	`Y`	`Y`
	l**e**g	p**e**t	B**e**tt	`E`	`E`	`E`	`E`
œ			G**ö**tter	`/`	`Q`	`Q`	`/`
æ		p**a**t	R**a**gtime	`{`	`&`	`^/`	`{`
a			hat	`a`	`a`	`a`	`&`
ɑ	l**a**t		K**a**lev**ala**	`A`	`A`	`A`	`A`
		p**o**t		`Q`	`O`	`O`	`Q`
		p**u**tt	Pl**u**mpudding	`V`	`V`	`^`	`V`
ɔ	b**o**m		Gl**o**cke	`O`	`O`	`O`	`O`
ʊ		p**u**t	P**u**lt	`U`	`U`	`U`	`U`
ʉ	p**u**t			`}`	`U`	`Y/`	`}`
ə	g**e**lijk	**a**noth**er**	B**e**ginn	`@`	`@`	`@`	`@`
œ̃ː			Par**fum**	`/~:`	`Q~:`	`Q~:`	`^`
æ̃		t**im**bre	**im**promptu	`{~`	`&~`	`^/~`	`c`
ɑ̃ː		dét**en**te	Dét**en**te	`A~:`	`A~:`	`A~:`	`q`
æ̃ː		l**in**gerie	Bass**in**	`{~:`	`&~:`	`^/~:`	`0`
˜ː		bouill**on**	Affr**on**t	`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	c 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	β 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 ***Co****llins* ***B****irmingham* ***U****niversity* ***I****nternational* ***L****anguage* ***D****atabase.* 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 ***I**nstituut voor **N**ederlandse **L**exicologie*, 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.