Babel Support for German

Johannes Braams Bernd Raichle Walter Schmidt
Jürgen Spitzmüller*

v2.99c (2026/02/14)

Abstract

This manual documents babel language support for German as provided by the babel-german package. The package supports all major (standard) varieties of German (i. e., Austrian, Swiss, and German Standard German) in contemporary as well as in pre-1996 (i. e., 1901) spelling.

Contents

*Current maintainer. Please report issues via https://codeberg.org/jspitz/babel-german/issues.

1 Aims and Scope

The babel-german package documented in this manual provides the babel package with all language-specific strings, settings and commands needed for writing German texts (or German passages in texts). Furthermore, the package assures that the appropriate hyphenation patterns are used for these texts or text passages (see sec. 5 for details).

Since German is a pluricentric language with differing standard varieties (in Austria, Switzerland, and Germany), babel-german supports three varieties of Standard German.[1] Furthermore, since the spelling and hyphenation rules of German (in all these standard varieties) have been reformed in 1996 (and in subsequent years), the package provides support for two spelling and hyphenation variants of each standard variety, viz. the deprecated 1901 spelling and the current ('reformed') 1996 spelling.

The following section provides some information on the history of the package, and in particular on a major interface change as of version 2.99. If you have been a user of babel-german before that version, you are advised to read this. If you are new or started using babel-german later than that, and just interested how to use the package, you can jump directly to section 3.

2 Caveats on Language Naming

2.1 Language Names in babel-german: A Tangled Affair

The origins of this package reach well beyond the mentioned spelling reform.[2] This has led to a rather unfortunate situation. When the spelling reform happened, the terms *german* and *austrian*[3] have already been used in babel for a couple of years, but of

[1] Austria, Switzerland, and Germany are so-called *full centers* of Standard German, as they developed each their specific codex. German is also an official language in Liechtenstein and Luxembourg, in parts of Italy (South Tyrol/Alte Adige), and a legally acknowledged minority language in other regions in the world. However, these *half* and *quarter centers* do not have their own codizes; South Tyrol/Alte Adige employs Austrian standard, Liechtenstein Swiss standard, Luxembourg and Belgium orient towards German standard (cf. [1] for linguistic details).

[2] Support for pre-1996 German started as a re-implementation of the package `german.sty` (v. 2.5b), originally developed by Hubert Partl (cf. [8]) and later maintained by Bernd Raichle (cf. [9]). Support for current varieties of German (post-1996 orthography) emerged as a re-implementation of Walter Schmidt's (1998, cf. [11]) companion package to `german.sty`, `ngerman.sty`. The initial re-implementations for babel were done by Johannes Braams, the original author and then maintainer of babel, in 1990 (for pre-1996 conventions) and 1999 (for post-1996 conventions). In 2013, Jürgen Spitzmüller took over maintainership for the then orphaned language support files that have been outsourced from babel itself to the independent package, babel-german.

[3] *Austrian* is a rather clumsy and irritating shorthand for *Austrian [Standard] German*, which does not only imply that there is a completely separate 'Austrian' language, but also that in Austria, (Austrian Standard) German is the only official language (whereas, according to the Austrian constitution, there are seven more, the so-called 'acknowledged minority languages' [Burgenland] Croatian, Czech, Hungarian, Slovak, Slowenian, Romanes, and Austrian Sign Language).

course they adhered to the rules of pre-1996 spelling/hyphenation. Since these rules (both spelling and hyphenation) changed quite significantly with the reform, post-1996 German orthography could not be supported with the existing language support files. Adapting those to the reformed orthography was not an option, as this would have dropped support for the previous norms (and hence existing or future documents that employ pre-1996 orthography). It would also not have been socially acceptable since the spelling reform hit quite heavy resistance in the general public (cf. [7]), and many (LaTeX users certainly included) assumed, and hoped, the new rules would be withdrawn again rather sooner than later anyway.

Against the backdrop of this, post-1996 support had not been integrated into the existing language support files for German and Austrian German, but provided separately in additional ones (technically in completely separated, though collectively distributed packages) – incidentally almost three years after the reformed orthography has come into effect (albeit with a transition period of eight years). These additional support files were named *naustrian* and *ngerman* in order to distinguish them from the received ones (with the *n* obviously expanding to 'new', referring to the then common label, *neue Rechtschreibung* 'new orthography'). When support for the Swiss standard variety was added in 2013, the 'new orthography' was not so new anymore and widely accepted. Nonetheless, the naming convention was not touched and adopted for the new varieties, `swissgerman` (pre-1996) and `nswissgerman` (post-1996).

Fast forward even more, thirty years after the reformed rules have been implemented, the 1996 orthography and the heated debate it caused have long settled, the (no longer really) 'new' orthography is the common one in all German-writing countries. Pre-1996 orthography is only employed by a minority of writers as the main norm, but of course it is still needed for texts that have been written before the reform, or – more commonly – as a secondary variety if you quote from such older texts.

So, to be sure, in our days, most people expect to get current (that is, post-1996) standards when selecting `german` in babel. Some arguably are not even aware that there have been older orthographic standards. Yet with babel, one still needed to select `ngerman`, `naustrian`, or `nswissgerman` to get contemporary orthographic conventions and hyphenation rules in the year 2025! The terms you would intuitively use, on the other hand, loaded patterns and captions that are not what many users would expect, namely those adhering to pre-1996 norms.

Why hasn't this been changed once the 1996 orthography has settled? The main reason is *backwards compatibility*. A simple semantic switch (with `german` then suddenly meaning post-1996 orthography) would break with a central promise of LaTeX: LaTeX does not change the output of existing documents behind the back of their authors!

While this is a very good principle, sometimes breaking it might be warranted, since keeping things as they are causes more harm than it prevents from. And this has arguably become the case with the language names of babel-german: More and more people reported that they are irritated by the fact that `german` does not mean what they expect (namely, German according to current standards). It has been assumed that many even erroneuosly loaded pre-1996 patterns without noticing (and getting wrong hyphenations). Given all that, we have been urged to do the compatibility-breaking change, and at some point, we have finally been convinced to do it – but only since we found some strategies to do it in a way that will affect as less users as possible (although it will still definitely affect some). The next section will elaborate on the changes and strategies.

2.2 The New Language Naming Convention

New feature in v. 2.99!

With v. 2.99 of this package, a new and more appropriate naming scheme is introduced. As of v. 3.0, we will also change the (default) semantics of `german`. All of these are major changes which might break backwards compatibility (but only the change of `german` will do it in a way that affects *existing* documents as opposed to new documents sent to users of older versions of babel-german). The changes address several problems:

1. Ultimately, the confusion of `german` activating pre-1996 orthography shall be resolved. From v. 3.0 on, `german` will load contemporary (post-1996) patterns for German Standard German[4], except for documents where we assume it really means pre-1996 orthography, that is, documents also loading `ngerman`, `naustrian`, or `nswissgerman`.

   This will also have an effect on the internal language names, which are still defined in the file `language.dat` in the received way (meaning `\l@german` continues to denote pre-1996 patterns, `\l@ngerman` post-1996 patterns by default). If the semantics of `german` is changed, `\l@german` also has to be redefined. Hence, a new name for pre-1996 German is introduced, `\l@tgerman`, which will have a stable meaning independent of the naming convention, `\l@ngerman` continues to denote post-1996 patterns, `\l@german` might denote one or the other, depending on the option discussed next.

2. To adjust this for specific documents, we introduce an option where you can select whether `german` should still always load pre-1996 patterns (this will remain to be the default before v. 3.0), always post-1996 patterns notwithstanding parallel usages of `ngerman`, `naustrian`, or `nswissgerman`, or guess depending on whether these `n`-options are used or not (the default as of v. 3.0). See section 4 for details.

3. While we are at it, we introduce more appropriate terms (language options) for the selection of language varieties and deprecate some of the problematic ones together with the `n`-forms (but of course, those received options will continue to work, although they might encourage you to switch in a warning once the new scheme has settled).

4. These new language options also use the newer and better 'experimental' hyphenation patterns by default, whereas the received options continue to use the less accurate legacy patterns by default (see sec. 5). The option `german` uses the newer patterns whenever it is configured to refer to post-1996 orthography, legacy patterns otherwise.

Having unpacked all this rather intricate background, we now turn to the actual usage of the package.

3 Enabling German Support

In order to use the language support provided by babel-german, you need to load the babel package (via `\usepackage{babel}`) and pass one of the following language options either directly to babel (via `\usepackage[<options>]{babel}`) or to `\documentclass` (the

[4] This follows the received convention to imply *German* Standard German for `german`, but see section 4 why this is not so straightforward.

latter has the advantage that also other packages are informed of the option[5]). If you use multiple languages/varieties (including different regional or orthographic varieties of German), the one passed last is treated by babel as the main language of the document.

New feature in v. 2.99!

The behavior of some language varieties can be adjusted by language variety options. All of these can be set via the macro `\germansetup`, which takes a comma-separated list of options as its mandatory argument and is to be used in the document preamble after babel has been loaded (see sec. 3.4). Some options alternatively might be passed as a babel modifier, which might give a more granular setting, since `\germansetup` applies to all varieties that support a specific option, babel modifiers only to the variety that is being modified.[6]

The available language and language variety options are introduced in what follows.

3.1 Austrian Standard German

Austrian Standard German refers to the norms current in Austria and South Tyrol/Alte Adige. The available choices are:

New feature in v. 2.99!

- `german-at` or `german-austria` if you want contemporary (post-1996) patterns
- `german-at-1901` or `german-austria-1901` if you want pre-reform (pre-1996) patterns

New feature in v. 2.14!

Contemporary (post-1996) Austrian Standard German provides an additional feature that is enabled via the language variety option (`\germansetup` or babel modifier) `capsz` and disabled via `\germansetup` option `capsz=false` or babel modifier `nocapsz`:

- `capsz`: `\MakeUppercase{ß}`, `\MakeUppercase{"s}` and the `"S` shorthand expand to the capital eszett letter rather than ⟨SS⟩ (see sec. 7.1 for details).
- `nocapsz` (= default): `\MakeUppercase{ß}`, `\MakeUppercase{"s}` and the `"S` expand to ⟨SS⟩. Within `\germansetup`, use `capsz=false` instead.

The received options `austrian` and `naustrian` still work (the latter also with the `capsz` option), but you are discouraged to use them unless you share your document with co-authors who have an older version of babel-german installed, or if you use a package that does not (yet) understand the new options. Also note that these options use the legacy hyphenation patterns by default, whereas the recommended options use the newer and better 'experimental' patterns (see sec. 5).

3.2 German Standard German

German Standard German refers to the norms current in Germany, Luxembourg, and Belgium. The available choices are:

New feature in v. 2.99!

- `german-de` or `german-germany` if you want contemporary (post-1996) patterns
- `german-de-1901` or `german-germany-1901` if you want pre-reform (pre-1996) patterns

[5] Side note to package authors; babel-german inserts the respective legacy options to the class options list if new options are used. So the new options should also work with most packages that only rely on the received ones.

[6] If languages are loaded via babel option, modifiers are appended to the language name with a dot, e. g. `german-at.capsz`; if languages are loaded via `\documentclass` options, use additionally babel options of the form `modifiers.german-at=capsz`.

New feature in v. 2.14!

Like Austrian Standard German, contemporary (post-1996) German Standard German optionally supports the capital eszett letter. The feature is enabled via the language variety option (`\germansetup` or babel modifier) `capsz` and disabled via babel modifier `nocapsz` or `\germansetup` option `capsz=false`:

- `capsz`: `\MakeUppercase{ß}`, `\MakeUppercase{"s}` and the `"S` shorthand expand to the capital eszett letter rather than ⟨SS⟩ (see sec. 7.1 for details).
- `nocapsz` (= default): `\MakeUppercase{ß}`, `\MakeUppercase{"s}` and the `"S` expand to ⟨SS⟩. Within `\germansetup`, use `capsz=false` instead.

The received options `german` and `ngerman` still work (the latter also with the `capsz` option), but you are discouraged to use them unless you share your document with co-authors who have an older version of babel-german installed, or if you use a package that does not (yet) understand the new options.

While `german` seems the obvious choice to typeset German, it is in many ways ambiguous due to its terminological tradition in babel (as well as in other packages such as `german.sty`) and due to the fact that there are multiple parallel standards (see sec.4). So it is always advisable to use a more precise option such as `german-de`.

Also, unless `glottonyms=contemporary` or `glottonyms=auto` is set (see section 4), `german` will still load pre-reform (pre-1996) and legacy hyphenation patterns (this will change as of babel-german v. 3.0), `ngerman` also uses the legacy hyphenation patterns by default. The options recommended above, by contrast, use the newer and better 'experimental' patterns (see sec. 5).

3.3 Swiss Standard German

Swiss Standard German refers to the norms current in Switzerland and Liechtenstein. The available choices are:

New feature in v. 2.99!

- `german-ch` or `german-switzerland` if you want contemporary (post-1996) patterns
- `german-ch-1901` or `german-switzerland-1901` if you want pre-reform (pre-1996) patterns

New feature in v. 2.10!

Swiss Standard German (both pre-and post-1996) provides an additional feature that is enabled via the language variety option (`\germansetup` or babel modifier) `toss`:

- `toss`: the shorthands `"s` and `"z` will expand to ⟨ss⟩ rather than ⟨ß⟩ (see sec. 7.2 for details).

The received options `swissgerman` and `nswissgerman` still work (also with the `toss` option), but you are discouraged to use them unless you share your document with co-authors who have an older version of babel-german installed, or if you use a package that does not (yet) understand the new options. Also note that `nswissgerman` uses the legacy hyphenation patterns by default, whereas the recommended options use the newer and better 'experimental' patterns (see sec. 5).

3.4 Options

The behavior of babel-german can be configured by a number of options. These are all set in the preamble via the command `\germansetup{<options>}` which takes a comma-separated list of key-value options. These are:

- `capsz=true|false` determines whether capital ⟨ß⟩ is output as a capital eszett letter or as ⟨SS⟩. Only available for Austrian and German 1996 German. See sec. 7.1. Default: `false`.
- `gendermark=<symbol>` the symbol used for marking gender forms as entered via the babel shorthand `"x` (see sec. 6). Default: ∗.
- `glottonyms=auto|contemporary|legacy` determines how the language option `german` is interpreted (1901 or 1996 orthography). See sec. 4. Default: `legacy`.
- `toss=true|false` determines whether the shorthands `"s` and `"z` will expand to ⟨ss⟩ or than ⟨ß⟩. Only available for Swiss Standard German. See sec. 7.2. Default: `false`.
- `hyphenrules=latest|legacy|<date>` determines which hyphenation patterns are used. See sec. 5. Default: `legacy` for `austrian`, `german` with 1901 meaning (see sec. 4), `swissgerman`, `naustrian`, `ngerman` and `nswissgerman`, `latest` for all other language options.

4 Configuring the Meaning of `german`

New feature in v. 2.99!

In section 2, we have elaborated on the intricate situation with how pre- and post-1996 spelling variants have been traditionally named in babel-german. Meanwhile, babel-german has introduced more appropriate names, but still the situation with the ambiguity of `german` is challenging.[7]

Since German is a pluricentric language (cf. [3]), a label such as `german` is of course inherently ambiguous (does it mean Austrian, German, or Swiss Standard German? The answer arguably depends on where you are located[8]). Hence, it is better to use less ambiguous terms such as `german-de` or `german-germany`.

Having said this, we understand that an option `german` which produces sensible results is expected in the context of babel and its (often rather awkward) language option terminology. And in this context, the traditional meaning of `german` (referring to the pre-1996 German Standard German) obviously causes irritation. Whichever national variety of German babel users might expect when using `german`, they most probably would expect *contemporary* standards – at least if they are not already familiar with the received babel-german terminology.

In order to mitigate this for users who employ the `german` option rather than the more precise alternatives, and considering both novice and experienced users of babel-german, an option `glottonyms` is provided (*glottonym* means 'language name'). It has to be set via `\germansetup` and offers the following choices:

- `glottomyms=legacy`: `german` always enables pre-1996 spelling, as it has been traditionally the case in babel-german. It also uses the legacy hyphenation patterns.

[7] Arguable, this also applies to `austrian` and `swissgerman`, but these names are discouraged anyway and will not change semantics.

[8] On a more global scale, it arguably also depends on politics and historical hegemonies. Which variety a generic language name assumes as default is neither depending on the 'origin' of the language or whether the language name is associated with the name of a country (cf. `english` which assumes American, not British Standard English, as default), nor the number of speakers (cf. `spanish` which assumes Standard Spanish in Spain, not in Mexico, as default, or `portuges` [sic!] which assumes Standard Portuguese in Portugal, not in Brazil). Yet it is also not by coincidence. The selection of a default (and this also accounts for norm authorities such as the ISO or the IETF) is mainly, and inherently, political, ideological, and power-driven, notwithstanding the involved actors often stating that it is not.

Use this if you need to maintain 100 % backwards compatibility. This is also the current default.

- `glottomyms=contemporary`: `german` always enables post-1996 spelling, which breaks with traditional package conventions and might break documents that use those. It also uses the 'experimental' hyphenation patterns by default. Use with care!

- `glottomyms=auto`: `german` *as a rule* enables post-1996 spelling (and newer hyphenation patterns). However, as soon as `ngerman`, `naustrian` or `nswissgerman` are also used in the same document, we assume that `german` is supposed to refer to the pre-1996 variant instead and hence make `german` enable pre-1996 German Standard German (and legacy patterns). This should handle gracefully most contemporary uses of `german`, although it will break documents using only `german` with the traditional meaning (for which you should use `glottomyms=legacy`). This choice will be the new default as of babel-german v. 3.0.

Note, however, that the semantics is only changed here for babel-german itself. If you use a third-party package which relies on the `legacy` semantics, you need to stick with this and report to the maintainer of that package.

5 Hyphenation Patterns

For German, different hyphenation patterns are available. Which of these a given document employs does not only depend on the varieties it uses, but also on the TEX engine and on the language options you employ. For most varieties and options, there are multiple options to select from. This is elaborated in what follows.

Hyphenation patterns for pre- and post-1996 German orthography have been available with TEX distributions for a long time (currently, these are shipped in form of the `dehypht` [= traditional] and `dehyphn` [= new] files). These established patterns, however, have many flaws: they are hard to maintain and improve since the sources are not available and not much is known about their construction, since they do not work well with loans, some compounds, and technical terms and often do not hyphenate where they could, and – most gravely – since they might produce wrong hyphenations (e.g., *Mordopfer* instead of *Mord-opfer*). The patterns for post-1996 orthography are even worse: they have only been *manually* adapted to the new rules and intended to be just some intermediate solution right from the start (cf. [11]).

Therefore, a group of experienced germanophone LATEX users (including the author of the legacy `dehyphn` patterns) took on the challenge and developed completely new patterns that do much better, the so-called 'experimental' new hyphenation patterns of German, distributed in the dehyph-exptl package [6]. As opposed to the established patterns, the new ones undergo constant improvement. The price for this is that hyphenation and thus the typeset document is subject to change with, and only due to, pattern updates. However, the new patterns are around and used since 2008, they have largely stabilized and are really no longer 'experimental'.

Modern engines (i. e., `xetex` and `luatex`) who require utf8-encoded patterns have already embraced those new patterns long ago, i. e., they are activated on these engines by default (cf. [10]). The classic TEX engines (`tex`/`pdftex`) have been more reluctant and continue to use the old patterns by default. The reason for this are the TEX quality standards already mentioned in sec. 2.1: refrain, if ever possible, from changing the output of user's documents in the wake of software updates. However, even there, there was

an exception: with pre-1996 Swiss Standard German, the classic engines use the 'experimental' patterns by default since when Swiss German was introduced, the 'experimental' patterns have already been available.

In babel-german, we take the introduction of the new language options in v. 2.99 as an opportunity to default to the latest 'experimental' patterns with these options. For `german`, with `glottonyms=legacy` or `glottonyms=auto` and the use of an `n`-option (i. e., if it refers to pre-1996 orthography), the established (legacy) patterns will continue to be used. With the newer meaning (`glottonyms=contemporary` or `glottonyms=auto` without the use of an `n`-option), babel-german also defaults to the latest 'experimental' patterns. The other legacy options (`austrian`, `naustrian`, `ngerman`, and `nswissgerman`) will continue to default to the established (legacy) patterns. This way, we assure existing documents will not change their hyphenation behind your back.

In all these cases (except for pre-1996 Swiss Standard German where no 'legacy' patterns exist), however, you can opt-in to a different setting. If you want to use the experimental patterns also with the legacy language options, use

```
\germansetup{hyphenrules=latest}
```

in the document preamble after babel has been loaded. This will activate the experimental hyphenation patterns in their most recent version for all used varieties of German. Reversely,

```
\germansetup{hyphenrules=legacy}
```

will switch to the established patterns for all used varieties of German.

The dehyph-exptl package also allows to load patterns of a given (fixed) date instead of the latest ones, e.g. `2024-02-28`. Therewith, you can prevent future changes in hyphenation due to package updates. The feature is also supported by babel-german: Simply pass the date to `hyphenrules` (e. g., `hyphenrules=2024-02-28`). Of course, you need to assure patterns of this date exist in your tree. Cf. [6] for details, also for ways to set specific patterns to specific varieties of German only.

6 Shorthands and Convenience Macros

For all varieties of German, the character `"` is made active in order to provide some shorthand macros.

Some of these shorthands address peculiarities of pre-1996 German spelling with which you do not need to bother if you adhere to contemporary orthography:

1. the so-called *Dreikonsonanten-Regel* ('three consonant rule') which required specific handling of specific compounds in hyphenation[9], and

[9] The three consonant rule (cf. [4, R 204]) prescribed that one of three identical consonants had to be omitted if a vowel followed the three consonants (i. e., you wrote *Schiffahrt*, not *Schifffahrt*, *schnellebig*, not *schnelllebig*, *wetturnen*, not *wettturnen*). If the word was hyphenated at this position, however, the third consonant needed to reappear (*Schiff-fahrt*, *schnell-lebig*, *wett-turnen*); the shorthands `"f`, `"l`, `"t` etc. account for that. With the 1996 reform, the rule has been taken out of force (cf. [5, R 136]). Now, all consonants are always written (some lexicalized exceptions are *dennoch* and *Mittag*, but these get no additional consonant on hyphenation either: *den-noch*, *Mit-tag*). Note also that ⟨s⟩ (as in *Kongresssaal*, if ⟨ss⟩ is used as an alternative to ⟨ß⟩ or in Swiss writing) has always been excluded from this rule (cf. [4, R 204]), which is why no shorthand for that case is needed.

2. the hyphenation of the digraph[10] ⟨ck⟩ as ⟨k-k⟩ (e. g., *Bäcker*, *Bäk-ker*), which has been dropped with the reform in favor of shifting the whole digraph to the next line (*Bä-cker*).

Other shorthands are provided for frequently used special characters as well as for better control of hyphenation, line breaks, and ligatures, and are useful for all varieties of German.

Table 1 provides an overview of the shorthands that are provided by babel-german for all its variants. Table 2 lists some babel macros for quotation marks that might be used as an alternative to the quotation mark shorthands listed above.

7 Variety-Specific Options

7.1 Capital Eszett Letter

New feature in v. 2.14!

In 2008, a capital variant of the letter ⟨ß⟩ has been included to the Unicode standard (U+1E9E), and in 2017, the capital eszett letter has been acknowledged in German orthography as a valid alternative to ⟨SS⟩ in uppercase writing of ⟨ß⟩. The letter differs from its miniscule counterpart in that it is usually wider to match the form of uppercase letters.

If you want to use this letter, you can do so by using the variety option `capsz`, which is supported for the contemporary (post-1996) Austrian (cf. sec. 3.1) and German (cf. sec. 3.2) varieties. If you pass the option to `germansetup`, i. e.,

```
\germansetup{capsz}
```

it will apply to both those varieties. If you want a more granular setting, use a babel modifier instead (see sec. 3). As the eszett letter is not common in Swiss German writing in general, the option is not supported there. Similarly, the pre-1996 varieties do not support the letter either.

The option causes both `\MakeUppercase` and the `"S` shorthand (but not `"Z`) to use the capital eszett letter. Note that this requires a font which actually contains the glyph (otherwise, you still get ⟨SS⟩) and LaTeX kernel 2023/06 at least.

New feature in v. 2.15!

Note that you can also set the casing via `\babelprovide[casing=eszett]{<lang>}`. This is adhered to. If you want to disable such a global setting, you can do so by means of the `\germansetup` option `capsz=false` or babel modifier `nocapsz`.

7.2 Handling of "s and "z in Swiss German

New feature in v. 2.10!

In Swiss (and Liechtensteinian) German writing, the use of ⟨ß⟩ is rather uncommon. Swiss writers would normally use ⟨ss⟩ where German or Austrian writers use the ⟨ß⟩ character (e. g., *Buße* vs. *Busse* 'penance'). When texts (or names) from other German speaking areas are quoted, however, the spelling and hence the ⟨ß⟩ is often maintained (particularly in scholarly writing where the spelling of quoted text is not supposed to be touched).

We assume that Swiss writers will normally input ⟨ss⟩ directly when they mean ⟨ss⟩, and we assume furthermore that the ⟨ß⟩-related shorthands `"s` and `"z` are useful also for Swiss writers when they actually need ⟨ß⟩, the more so since the ⟨ß⟩ is not as directly

[10] In graphematics, the term *digraph* denotes two characters that make a functional pair (which means, depending on the theoretical assumptions: they represent a single sound or they are semantically distinctive).

Table 1: Shorthands provided by babel-german

	`"a`	Umlaut ⟨ä⟩ (shorthand for `\"a`). Similar shorthands are available for all other lower- and uppercase vowels (umlauts: `"a`, `"o`, `"u`, `"A`, `"O`, `"U`; tremata: `"e`, `"i`, `"E`, `"I`).
	`"s`	German ⟨ß⟩ (shorthand for `\ss`); but cf. sec. 7.2 for specifics with Swiss Standard German.
	`"z`	German ⟨ß⟩ (shorthand for `\ss`). The difference to `"s` is the uppercase version; again, cf. sec. 7.2 for specifics with Swiss Standard German.
	`"ck`	⟨ck⟩, hyphenated as ⟨k-k⟩ in pre-1996 variants. Behaves like ordinary *ck* in post-1996.
	`"ff`	⟨ff⟩, hyphenated as ⟨ff-f⟩ in pre-1996 variants; outputs ⟨fff⟩ in post-1996 variants; this is also implemented for ⟨l⟩, ⟨m⟩, ⟨n⟩, ⟨p⟩, ⟨r⟩ and ⟨t⟩. Please refer to footnote 9 for why this does not include ⟨s⟩.
	`"S`	`\uppercase{"s}`, typeset as ⟨SS⟩ – ⟨ß⟩ is traditonally written as ⟨SS⟩ (or, in 1901 spelling, also optionally ⟨SZ⟩, see below) in uppercase writing; cf. sec. 7.1 if you prefer a capital eszett.
	`"Z`	`\uppercase{"z}`, typeset as ⟨SZ⟩. In 1901 spelling, ⟨ß⟩ could also be written as ⟨SZ⟩ instead of ⟨SS⟩ in uppercase writing. Note that, with reformed orthography, the ⟨SZ⟩ variant has been deprecated in favour of ⟨SS⟩ only.
	`"|`	Disable ligature at this position (e. g., at morpheme boundaries, as in `Auf"|lage`).
	`"-`	An additional breakpoint that does still allow for hyphenation at the breakpoints preset in the hyphenation patterns (as opposed to `\-`).
	`"=`	An explicit hyphen with a breakpoint, allowing for hyphenation at the other points preset in the hyphenation patterns (as opposed to plain `-`); useful for long compounds such as `IT"=Dienstleisterinnen`.
	`"~`	An explicit hyphen without a breakpoint. Useful for cases where the hyphen should stick at the following syllable, e. g., `bergauf und "~ab .`
	`""`	A breakpoint that does not output a hyphen if the line break is performed (consider parenthetical extensions as in `(pseudo"~)""wissenschaftlich`).
New feature in v. 2.9!	`"/`	A slash that allows for a linebreak. As opposed to `\slash`, hyphenation at the breakpoints preset in the hyphenation patterns is still allowed.
New feature in v. 2.14!	`"*`	An asterisk which assures the word can still be hyphenated at its defined breakpoints. Useful if you want to employ gender-sensitive writing (‘gender star’). Similar shorthands are available for the alternative gender-sensitive spellings, `":` and `"_`.
New feature in v. 2.14!	`"x`	Inserts a gender mark which assures the word can still be hyphenated at its defined breakpoints. This is predefined to ∗ but can be globally redefined via `\germansetup{gendermark=<symbol>}`.
	`"‘`	German left double quotes ⟨„⟩.
	`"’`	German right double quotes ⟨“⟩.
	`"<`	French/Swiss left double quotes ⟨«⟩.
	`">`	French/Swiss right double quotes ⟨»⟩.

accessible on Swiss keyboards as it is on German and Austrian ones. On the other hand, there might be occasions where writers want to transfer a text from German or Austrian Standard into Swiss Standard German and adapt the spelling on the fly, i. e., transform all ⟨ß⟩ into ⟨ss⟩.

For this special case, we provide an option to make the ⟨ß⟩-related shorthands `"s` and `"z` expand to the respective digraphs[11] ⟨ss⟩ and ⟨sz⟩ rather than to ⟨ß⟩. This is not the default behavior with `german-ch` and `german-ch-1901` since, as mentioned, there are situations when the ⟨ß⟩ is (and has to be) used in Swiss writing, and normally, no shorthand is needed to input (or output) two simple ⟨s⟩ characters. You can opt-in (and

[11] See footnote 10 for what this means.

Table 2: Alternative commands for quotation marks (provided by babel)

`\glqq`	German left double quotes ⟨„⟩.
`\grqq`	German right double quotes ⟨“⟩.
`\glq`	German left single quotes ⟨‚⟩.
`\grq`	German right single quotes ⟨‘⟩.
`\flqq`	French/Swiss left double quotes ⟨«⟩.
`\frqq`	French/Swiss right double quotes ⟨»⟩.
`\flq`	French/Swiss left single quotes ⟨‹⟩.
`\frq`	French/Swiss right single quotes ⟨›⟩.
`\dq`	The straight quotation mark character ⟨"⟩.

out) digraphical expansion of `"s` and `"z` on a global and local level:

- To globally switch on the digraphical expansion, use the `\germansetup` option or the babel modifier `toss` (read: ‘to ⟨ss⟩’) with `german-ch`, `german-ch-1901` or its aliases (see sec. 3). The former will apply to all Swiss German varieties, the latter only to the language option that is being modified.
- To switch on the digraphical expansion only locally, you can use the boolean switch `\tosstrue`. Likewise, `\tossfalse` switches off (both locally and globally set) digraphical expansion.

All these changes result in the following deviant behavior of two shorthands:

`"s`	expands to digraph ⟨ss⟩
`"z`	expands to digraph ⟨sz⟩

One further note related to the use of ⟨ss⟩ in pre-1996 Swiss Standard German. As opposed to other consonantial letters, the ⟨s⟩ is excluded from the three consonant rule (*Dreikonsonantenregel*) of traditional (i. e., 1901) German spelling (cf. footnote 9). This is why we don’t provide a shorthand for the ⟨sss⟩ case.

8 Captions, Extras, and Dates

The caption strings (such as “figure”) are defined in the macros `\captions<language>`, where `<language>` resolves to the current language option. With the new names, however, captions macros of traditional names are still defined and inherited. So redefining `\captionsngerman` and `\captionsgerman-de` will have the same effect with the option `german-de`. Vice versa, however, redefining `\captionsgerman-de` will *not* have any effect if the alias `german-germany` is used. The same applies to `\extras<language>` and `\noextras<language>` which hold extra language settings.

The recommended way to change or add caption strings is the babel macro

```
\setlocalecaption{<language-name>}{<caption-name>}{<string>}
```

where `<language-name>` is the respective language option, `<caption-name>` the respective caption macro (without preceding backslash and trailing `name`), and `<string>` the string to which it should expand. E.g.,

```
\setlocalecaption{german-de}{ref}{Bibliografie}
```

The predefined caption names are (the macros have an appended `name`):

- `preface`: *Vorwort*
- `ref`: *Literatur*
- `abstract`: *Zusammenfassung*
- `bib`: *Literaturverzeichnis*
- `chapter`: *Kapitel*
- `appendix`: *Anhang*
- `contents`: *Inhaltsverzeichnis*
- `listfigure`: *Abbildungsverzeichnis*
- `listtable`: *Tabellenverzeichnis*
- `index`: *Index*
- `figure`: *Abbildung*
- `table`: *Tabelle*
- `part`: *Teil*
- `encl`: *Anlage(n)* (de) or *Beilage(n)* (ch and at)
- `cc`: *Verteiler*
- `headto`: *An*
- `page`: *Seite*
- `see`: *siehe*
- `also`: *siehe auch*
- `proof`: *Beweis*
- `glossary`: *Glossar*

If you want to change captions of all German varieties at once, you can use the `<language-name>` `allgerman` with the `\setlocalecaption` macro (`\captionsallgerman` is inherited by all varieties). To redefine all Austrian or Swiss varieties, respectively, use `allatgerman` and `allchgerman`.

As opposed to this, `\extras<language>`, `\noextras<language>` and `\date<language>` are appended by `\addto`. Note, however, that the extension via `\addto` is not straightforward in the case of language options containing hyphens (such as `german-de`) since the hyphen has a different catcode normally (e.g., in document preambles) and hence terminates command parsing. So you either need to resort to the legacy names (such as `ngerman`), control command parsing via

```
\expandafter\addto\csname <macro>\endcsname{<code>}
```

or use the babel-german helper macro `\addtocs{<macro>}{<code>}`, which does the latter under the hood, taking as first argument the macro name without backslash. E.g.,

```
\addtocs{extrasgerman-de}{\bbl@nonfrenchspacing}
```

Packages that want to change extras and noextras of all German varieties at once might append code to the internal macros `\@extrasgerman` and `\@noextrasgerman` that are inherited by all varieties.

For date redefinitions, packages should redefine the internal macros `\date@german@at`, `\date@german@ch`, and `\date@german@de` that hold the definitions for the respective regions and are being inherited in the respective varieties.

9 Implementation

9.1 General Settings

The file `babel-german.def` holds the common code for all varieties of German. In this file, which is inputted by all `*.ldf` files of babel-german, the main work is done.

We define some helper macros that help us to identify later on whether we use an option that conforms with the internal language naming.

```
\def\bbl@opt@german{german}
\def\bbl@opt@swissgerman{swissgerman}
\def\bbl@opt@ngerman{ngerman}
```

Also, we define helpers to identify the region

```
\def\bbl@german@region@at{AT}
\def\bbl@german@region@ch{CH}
\def\bbl@german@region@de{DE}
```

… and legacy hyphenation patterns:

```
\def\bbl@german@legacy@patterns{legacy}
```

\ifbbl@german@newterms
\ifbbl@german@maybe@newterms
\tosstrue
\tossfalse
\capsztrue
\capszfalse
\@bbl@german@at@capsztrue
\@bbl@german@at@capszfalse
\@bbl@german@ge@capsztrue
\@bbl@german@ge@capszfalse
\bbl@german@patterns@oldterms
\bbl@german@patterns@newterms

We provide (key-val) variety options. To this end, we first define some booleans and macros to store the settings.

```
\newif\ifbbl@german@newterms
\bbl@german@newtermsfalse
\newif\ifbbl@german@maybe@newterms
\bbl@german@maybe@newtermsfalse
\newif\iftoss
\tossfalse
\newif\ifcapsz
\capszfalse
\newif\if@bbl@german@at@capsz
\@bbl@german@at@capszfalse
\newif\if@bbl@german@ge@capsz
\@bbl@german@ge@capszfalse
\providecommand*\bbl@german@patterns@oldterms{legacy}
\providecommand*\bbl@german@patterns@newterms{latest}
```

\mkgender
\mkngender

`\mkgender` holds the string to which the `"x` shorthand resolves. `\mkngender` is just for backwards compatibilty (has been used in previous versions for 1996 orthography).

```
\def\mkgender{*}
\AtBeginDocument{%
   \ifx\mkngender\@undefined\else
       \let\mkgender\mkngender
   \fi
}
```

Now, the actual option definitions that set the booleans and macros:

```
\ExplSyntaxOn
\DeclareKeys[bblgerman]
 {
   % glottonyms=<legacy|contemporary|auto>
   glottonyms.choice:,
   % a. legacy
   glottonyms / legacy.code:n =
```

```
        { \bbl@german@newtermsfalse
          \bbl@german@maybe@newtermsfalse },
    % b. contemporary
    glottonyms / contemporary.code:n =
        { \bbl@german@newtermstrue
          \bbl@german@maybe@newtermsfalse },
    % c. auto
    glottonyms / auto.code:n =
        { \bbl@german@newtermsfalse
          \bbl@german@maybe@newtermstrue },
    glottonyms.default:n = { auto },
    % toss={true|false}
    toss.legacy_if_set:n = toss,
    % capsz={true|false}
    capsz.code:n =
        {
         \str_if_eq:nnTF { #1 } { true }
         {
            \capsztrue
            \@bbl@german@at@capsztrue
            \@bbl@german@ge@capsztrue
         }{
            \capszfalse
            \@bbl@german@at@capszfalse
            \@bbl@german@ge@capszfalse
         }
       },
    capsz.default:n = { true },
    % hyphenrules={version}
    hyphenrules.code:n =
     {
       \def\bbl@german@patterns@oldterms{#1}
       \def\bbl@german@patterns@newterms{#1}
       \def\bbl@german@xptl@patterns{}
     },
    gendermark.store = \mkgender,
}
\ExplSyntaxOff
```

`\germansetup` Provide a command to set macros. Assure it can only be used in preamble.

```
\providecommand*{\germansetup}[1]{%
  \SetKeys[bblgerman]{#1}%
}
\@onlypreamble\germansetup
```

We check for the existence of the required hyphenation patterns, issue a warning (and fall back to the null language) if it is unknown. If required we set `\l@<langoption>` as a 'dialect' of the hyphenation language.

`\l@tgerman` Since `\l@german` is ambiguous depending on the setting of `glottonyms`, we define `\l@tgerman` which always represents 1901 patterns:

```
\ifx\l@tgerman\@undefined
  \ifx\l@german\@undefined\else
        \let\l@tgerman\l@german
```

```
  \fi
\fi
```

Now, as this has been set, handle the `glottonyms` option if we are within `german`:

```
\ifx\CurrentOption\bbl@opt@german
  \AddToHook{begindocument/before}{%
```

First, if we have `glottonyms=auto`, check whether we have an n-variety that forces legacy semantics:

```
    \ifbbl@german@maybe@newterms
       \@ifundefined{bbl@german@force@legacy}{%
          \bbl@german@newtermstrue
          \PackageInfo{babel-german}{Using legacy glottonyms\MessageBreak
                       'german' denotes pre-1996 spelling.}%
       }{%
          \PackageInfo{babel-german}{Using contemporary glottonyms\MessageBreak
                       'german' denotes post-1996 spelling.}%
       }%
    \fi
```

We know now if `german` means 1901 or 1996, so set the hyphenation patterns if needed. Here, we also consider the `hyphenrules` option for `german`. The following is either `glottonyms=contemporary` or `glottonyms=auto` without n-variety:

```
    \ifbbl@german@newterms
```

If we do not find legacy `ngerman` patterns, warn and fall back to null language:

```
      \ifx\l@ngerman\@undefined
          \@nopatterns{German (current orthography),
                       falling back to 1901 orthography!}%
      \else
```

If `hyphenrules` have not been set, use `ngerman`:

```
          \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns
              \adddialect\l@german\l@ngerman
```

otherwise, use what is requested:

```
          \else
              \expandafter\adddialect\csname l@german\expandafter
                \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname
          \fi
```

and record that we use `german` in the modern sense:

```
          \addto\extrasgerman{\bbl@german@tradspellingfalse}%
      \fi
    \else
```

The following is either `glottonyms=legacy` or `glottonyms=auto` with n-variety. Here, we only set patterns if requested via `hyphenrules`:

```
       \ifx\bbl@german@patterns@oldterms\bbl@german@legacy@patterns\else
           \expandafter\adddialect\csname l@german\expandafter
               \endcsname\csname l@german-x-\bbl@german@patterns@oldterms\endcsname
       \fi
    \fi
  }
\fi
```

For the following, we need to temporarily give the hyphen ‘letter’ catcode as our options might consist of hyphens:

```
\edef\bbl@german@save@hyphen@catcode{\the\catcode‘\-}
\catcode‘\-=11
```

We begin with region DE, first 1901 spelling:

```
\ifx\bbl@german@region\bbl@german@region@de
  \ifbbl@german@tradspelling
      \ifx\l@tgerman\@undefined
        \@nopatterns{German (1901 orthography)}
        \adddialect\l@german0
        \adddialect\l@tgerman0
      \fi
    \ifx\CurrentOption\bbl@opt@german\else
       \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@tgerman
    \fi
```

Then DE-1996:

```
  \else% 1996 spelling
    \ifx\l@ngerman\@undefined
      \@nopatterns{German (current orthography)}
      \adddialect\l@ngerman0
    \fi
    \ifx\CurrentOption\bbl@opt@ngerman\else
       \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@ngerman
    \fi
  \fi
\fi
```

For AT-1901, we set `<langopt>` as a dialect of `german`, since the Austrian variety uses the same hyphenation patterns as Germany’s Standard German (both in pre- and post-1996 spelling).

If no German patterns are found, we issue a warning and fall back to null language.

```
\ifx\bbl@german@region\bbl@german@region@at
  \ifbbl@german@tradspelling
      \ifx\l@tgerman\@undefined
        \@nopatterns{German (1901 orthography), needed by Austrian (1901 orthography)}
        \expandafter\adddialect\csname l@\CurrentOption\endcsname0
      \else
        \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@tgerman
     \fi
```

Same for AT-1996, but as a dialect of `ngerman`:

```
  \else% 1996 spelling
    \ifx\l@ngerman\@undefined
      \@nopatterns{German (current orthography), needed by Austrian (current orthography)}
        \expandafter\adddialect\csname l@\CurrentOption\endcsname0
    \else
       \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@ngerman
    \fi
  \fi
\fi
```

For the pre-1996 Swiss variety, we attempt to load the specific `swissgerman` hyphenation patterns and fall back to `german` if those are not available. If no patterns are found, we issue a warning and go for null language.

```
\ifx\bbl@german@region\bbl@german@region@ch
  \ifbbl@german@tradspelling
     \ifx\l@swissgerman\@undefined
        \ifx\l@tgerman\@undefined
          \@nopatterns{Swiss Standard German (1901 orthography) and German (1901 orthography)}
          \expandafter\adddialect\csname l@\CurrentOption\endcsname0
        \else
          \@nopatterns{Swiss Standard German (1901 orthography),
                       falling back to German (1901 orthography)}
          \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@tgerman
        \fi
     \else
       \ifx\CurrentOption\bbl@opt@swissgerman\else
         \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@swissgerman
       \fi
     \fi
```

Post-1996 Swiss German uses `ngerman` hyphenation patterns, so try those:

```
  \else% 1996 spelling
    \ifx\l@ngerman\@undefined
      \@nopatterns{German (current orthography),
                   needed by Swiss Standard German (current orthography)}
       \expandafter\adddialect\csname l@\CurrentOption\endcsname0
    \else
       \expandafter\adddialect\csname l@\CurrentOption\endcsname\l@ngerman
    \fi
  \fi
\fi
```

Now we can reset the hyphen catcode to the saved value:

```
\catcode`\-=\bbl@german@save@hyphen@catcode
```

`\addtocs` Since the hyphen has catcode 12 normally and hence terminates a command name, we provide a helper command to easily append captions, extras etc. for the language names with hyphen:

```
\providecommand*\addtocs[2]{\expandafter\addto\csname #1\endcsname{#2}}
```

With the option `hyphenrules`, we load experimental hyphenation patterns (package dehyph-exptl). The following passes the respective code for a given variety to a hook that is being executed at document begin (when we know the setting of `hyphenrules`). We do not handle `german` here, as this is already done in the code that also considers `glottonyms`. Also, 1901 Swiss German already uses exptl patterns, so we ignore this:

```
\ActivateGenericHook{babel/german/patterns}
```

First 1901 variants:

```
\ifbbl@german@tradspelling
    \ifx\bbl@german@region\bbl@german@region@ch\else
       \def\tmpa{austrian}
       \ifx\CurrentOption\tmpa
           \AddToHook{babel/german/patterns}{%
             \ifx\bbl@german@patterns@oldterms\bbl@german@legacy@patterns\else
```

```
              \expandafter\adddialect\csname l@austrian\expandafter
              \endcsname\csname l@german-x-\bbl@german@patterns@oldterms\endcsname\fi}
      \fi
      \def\tmpa{german-at-1901}
      \ifx\CurrentOption\tmpa
          \AddToHook{babel/german/patterns}{%
            \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
              \expandafter\adddialect\csname l@german-at-1901\expandafter
              \endcsname\csname l@german-x-\bbl@german@patterns@newterms\endcsname\fi}
      \fi
      \def\tmpa{german-austria-1901}
      \ifx\CurrentOption\tmpa
          \AddToHook{babel/german/patterns}{%
            \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
              \expandafter\adddialect\csname l@german-austria-1901\expandafter
              \endcsname\csname l@german-x-\bbl@german@patterns@newterms\endcsname\fi}
      \fi
      \def\tmpa{german-de-1901}
      \ifx\CurrentOption\tmpa
          \AddToHook{babel/german/patterns}{%
            \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
              \expandafter\adddialect\csname l@german-de-1901\expandafter
              \endcsname\csname l@german-x-\bbl@german@patterns@newterms\endcsname\fi}
      \fi
      \def\tmpa{german-germany-1901}
      \ifx\CurrentOption\tmpa
          \AddToHook{babel/german/patterns}{%
            \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
              \expandafter\adddialect\csname l@german-germany-1901\expandafter
              \endcsname\csname l@german-x-\bbl@german@patterns@newterms\endcsname\fi}
      \fi
    \fi
\else
```

Then 1996 variants:

```
  \ifx\CurrentOption\bbl@opt@ngerman
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@oldterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@ngerman\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@oldterms\endcsname\fi}
  \fi
  \def\tmpa{german-at}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-at\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{naustrian}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@oldterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@naustrian\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@oldterms\endcsname\fi}
  \fi
```

```
  \def\tmpa{german-austria}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-austria\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{german-ch}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-ch\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{german-switzerland}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-switzerland\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{german-de}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-de\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{german-germany}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@newterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@german-germany\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@newterms\endcsname\fi}
  \fi
  \def\tmpa{nswissgerman}
  \ifx\CurrentOption\tmpa
      \AddToHook{babel/german/patterns}{%
        \ifx\bbl@german@patterns@oldterms\bbl@german@legacy@patterns\else
          \expandafter\adddialect\csname l@nswissgerman\expandafter
          \endcsname\csname l@ngerman-x-\bbl@german@patterns@oldterms\endcsname\fi}
  \fi
\fi
```

Execute the hook (once) before document begin if `hyphenrules` are requested:

```
\AddToHook{begindocument/before}{%
  \ifdefined\bbl@german@xptl@patterns
    \UseOneTimeHook{babel/german/patterns}%
  \fi
}
```

9.2 Language-Specific Strings (Captions)

The next step consists of defining macros that provide language specific strings and settings.

`\@captionsgerman` The macro `\@captionsgerman` defines all strings used in the four standard document classes provided with LaTeX for German. This is an internal macro that is inherited and modified by the following macros for the respective language varieties.

```
\@namedef{@captionsgerman}{%
  \def\prefacename{Vorwort}%
  \def\refname{Literatur}%
  \def\abstractname{Zusammenfassung}%
  \def\bibname{Literaturverzeichnis}%
  \def\chaptername{Kapitel}%
  \def\appendixname{Anhang}%
  \def\contentsname{Inhaltsverzeichnis}%
  \def\listfigurename{Abbildungsverzeichnis}%
  \def\listtablename{Tabellenverzeichnis}%
  \def\indexname{Index}%
  \def\figurename{Abbildung}%
  \def\tablename{Tabelle}%
  \def\partname{Teil}%
  \def\enclname{Anlage(n)}%
  \def\ccname{Verteiler}%
  \def\headtoname{An}%
  \def\pagename{Seite}%
  \def\seename{siehe}%
  \def\alsoname{siehe auch}%
  \def\proofname{Beweis}%
  \def\glossaryname{Glossar}%
}
```

`\captionsallgerman` The macro `\captionsallgerman` is a more accessible intermediate copy of `\@captionsgerman`.

```
\@namedef{captionsallgerman}{%
    \@nameuse{@captionsgerman}%
}
```

`\captionsallatgerman` The macro `\captionsallatgerman` redefines the variants common in AT and is inherited by all Austrian varieties.

```
\@namedef{captionsallatgerman}{%
    \@nameuse{captionsallgerman}%
    \def\enclname{Beilage(n)}%
}
```

`\captionsallchgerman` The macro `\captionsallchgerman` redefines the variants common in CH and is inherited by all Swiss varieties (currently identical to AT).

```
\@namedef{captionsallchgerman}{%
    \@nameuse{captionsallgerman}%
    \def\enclname{Beilage(n)}%
}
```

`\captionsgerman` The macro `\captionsgerman` is identical to `\captionsallgerman`, but only defined if `german`, `german-de-1901` or `german-germany-1901` are requested.

```
\ifx\CurrentOption\bbl@opt@german
  \@namedef{captionsgerman}{%
    \@nameuse{captionsallgerman}%
}
```

\captionsgerman-de-1901
\captionsgerman-germany-1901

For `german-de-1901` and `german-germany-1901`, we define both `\captionsgerman` and `\captionsgerman-de-1901` or `\captionsgerman-germany-1901`, respectively, which import the former.

```
\else
  \ifx\bbl@german@region\bbl@german@region@de
    \ifbbl@german@tradspelling
       \@namedef{captionsgerman}{%
          \@nameuse{captionsallgerman}%
       }
       \@namedef{captions\CurrentOption}{%
          \@nameuse{captionsgerman}%
       }
    \fi
  \fi
\fi
```

\captionsngerman

The macro `\captionsngerman` is identical to `\captionsallgerman`, but only defined if `ngerman`, `german-de` or `german-germany` is requested.

```
\ifx\CurrentOption\bbl@opt@ngerman
  \@namedef{captionsngerman}{%
      \@nameuse{captionsallgerman}%
}
```

\captionsgerman-de
\captionsgerman-germany

For `german-de` and `german-germany`, we define both `\captionsngerman` and `\captionsgerman-de` or `\captionsgerman-germany`, respectively, which import the former.

```
\else
  \ifx\bbl@german@region\bbl@german@region@de
     \ifbbl@german@tradspelling\else
        \@namedef{captionsngerman}{%
           \@nameuse{captionsallgerman}%
        }
        \@namedef{captions\CurrentOption}{%
           \@nameuse{captionsngerman}%
        }
     \fi
  \fi
\fi
```

\captionsaustrian
\captionsnaustrian
\captionsgerman-at-1901
\captionsgerman-at
\captionsgerman-austria-1901
\captionsgerman-austria

The Austrian `\caption*s` build on `\captionsallgerman`, but redefine some strings following Austrian conventions (for the respective variants, cf. [1]). They are only defined if an Austrian variety is requested.

```
\ifx\bbl@german@region\bbl@german@region@at
 \ifbbl@german@tradspelling
    \def\bbl@tmpa{austrian}
    \ifx\CurrentOption\bbl@tmpa
       \@namedef{captions\CurrentOption}{%
         \@nameuse{captionsallatgerman}%
       }
```

```
    \else
      \@namedef{captionsaustrian}{%
        \@nameuse{captionsallatgerman}%
      }
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsaustrian}%
      }
    \fi
  \else
    \def\bbl@tmpa{naustrian}
    \ifx\CurrentOption\bbl@tmpa
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsallatgerman}%
      }
    \else
      \@namedef{captionsnaustrian}{%
        \@nameuse{captionsallatgerman}%
      }
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsnaustrian}%
      }
    \fi
  \fi
\fi
```

\captionsswissgerman
\captionsnswissgerman
\captionsgerman-ch-1901
\captionsgerman-ch
\captionsgerman-switzerland-1901
\captionsgerman-switzerland

The Swiss \caption∗s build on \captionsallgerman, but redefine some strings following Swiss conventions (for the respective variants, cf. [1]). They are only defined if a Swiss German variety is requested.

```
\ifx\bbl@german@region\bbl@german@region@ch
 \ifbbl@german@tradspelling
   \ifx\CurrentOption\bbl@opt@swissgeman
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsallchgerman}%
      }
    \else
      \@namedef{captionsswissgeman}{%
        \@nameuse{captionsallchgerman}%
      }
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsswissgeman}%
      }
    \fi
  \else
    \def\bbl@tmpa{nswissgeman}
    \ifx\CurrentOption\bbl@tmpa
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsallchgerman}%
      }
    \else
      \@namedef{captionsnswissgeman}{%
        \@nameuse{captionsallchgerman}%
      }
      \@namedef{captions\CurrentOption}{%
        \@nameuse{captionsnswissgeman}%
```

```
      }
    \fi
  \fi
\fi
```

9.3 Date Localizations

\month@german

The macro \month@german defines German month names for all varieties.

```
\def\month@german{\ifcase\month\or
  Januar\or Februar\or M\"arz\or April\or Mai\or Juni\or
  Juli\or August\or September\or Oktober\or November\or Dezember\fi}
```

\date@german@at
\date@german@ch
\date@german@de

We define some internal macros with common settings for each region. From these, only Austrian differs in the naming of January (*Jänner*):

```
\@namedef{date@german@at}{\def\today{\number\day.~\ifnum1=\month
    J\"anner\else \month@german\fi \space\number\year}}
\@namedef{date@german@ch}{\def\today{\number\day.~\month@german
      \space\number\year}}
\@namedef{date@german@de}{\def\today{\number\day.~\month@german
      \space\number\year}}
```

\dateaustrian
\datenaustrian
\dategerman-at-1901
\dategerman-at
\dategerman-austria-1901
\dategerman-austria

The Austrian \date* macros redefine the command \today to produce Austrian versions of the German dates (with the specific naming of January which differs from the other German varieties). The macro is only defined if an Austrian variety is requested.

```
\ifx\bbl@german@region\bbl@german@region@at
   \@namedef{date\CurrentOption}{\@nameuse{date@german@at}}
\else
```

\dateswissgerman
\datenswissgerman
\dategerman-ch-1901
\dategerman-ch
\dategerman-switzerland-1901
\dategerman-switzerland

The other \date* macros redefine the command \today to produce the respective dates for Swiss and German Standard German. They are all identical, both for all Swiss varieties:

```
    \ifx\bbl@german@region\bbl@german@region@ch
      \@namedef{date\CurrentOption}{\@nameuse{date@german@ch}}
    \else
```

\dategerman
\datengerman
\dategerman-de-1901
\dategerman-de
\dategerman-germany-1901
\dategerman-germany

as well as for all German varieties:

```
      \@namedef{date\CurrentOption}{\@nameuse{date@german@de}}
    \fi
\fi
```

9.4 Extras

The \extras* macros will perform all the extra definitions needed for the respective variety. The \noextras* macros are used to cancel the actions of \extras*.

First, the character " is declared active for all German varieties. This is done once, later on its definition may vary.

```
\initiate@active@char{"}
```

\@extrasgerman The macro \@extrasgerman holds all the default extras setting. This is an internal macro that is inherited and modified by the following macros for the respective language varieties.

```
\@namedef{@extrasgerman}{%
```

First, we load the shorthands defined below and activate the " character

```
  \languageshorthands{german}%
  \bbl@activate{"}%
```

In order for TEX to be able to hyphenate German words which contain 'ß' (in the OT1 position ^^Y), we furthermore have to give the character a nonzero \lccode (see Appendix H, the TEXbook).

```
  \babel@savevariable{\lccode25}%
  \lccode25=25%
```

The umlaut accent macro \" is changed to lower the umlaut dots. The redefinition is done with the help of \umlautlow.

```
  \babel@save\"\umlautlow
```

For German texts, we finally need to assure that \frenchspacing is turned on.

```
  \bbl@frenchspacing
}
```

Depending on the option with which the language definition file has been loaded, a respective \extras* macro is defined. Each of those is identical: it simply inherits \@extrasgerman. However, the traditional names (german, ngerman, austrian, naustrian, swissgerman, and nswissgerman) are used as an intermediate layer, so redefining those will also redefine the newer aliases.

\extrasgerman First, the legacy extras macro for pre-1996 German German:

```
\ifx\CurrentOption\bbl@opt@german
  \@namedef{extrasgerman}{%
      \@nameuse{@extrasgerman}%
  }
\else
```

\extrasngerman Then, the legacy extras macro for post-1996 German German:

```
  \ifx\CurrentOption\bbl@opt@ngerman
    \@namedef{extrasngerman}{%
        \@nameuse{@extrasgerman}%
    }
  \else
```

\extrasgerman-de-1901 Now newer alias names for pre-1996 German German:

\extrasgerman-germany-1901

```
    \ifx\bbl@german@region\bbl@german@region@de
       \ifbbl@german@tradspelling
          \@namedef{extrasgerman}{%
             \@nameuse{@extrasgerman}%
          }
          \@namedef{extras\CurrentOption}{%
             \@nameuse{extrasgerman}%
          }
       \else
```

\extrasgerman-de
\extrasgerman-germany
and post-1996 German German:

```
        \@namedef{extrasngerman}{%
           \@nameuse{@extrasgerman}%
        }
        \@namedef{extras\CurrentOption}{%
           \@nameuse{extrasngerman}%
        }
      \fi
    \fi
  \fi
\fi
```

\extrasaustrian
Same for Autrian: first, the legacy `extras` macro for pre-1996 Austrian German:

```
\def\bbl@tmpa{austrian}
\def\bbl@tmpb{naustrian}
\ifx\CurrentOption\bbl@tmpa
  \@namedef{extrasaustrian}{%
     \@nameuse{@extrasgerman}%
  }
\else
```

\extrasnaustrian
Then, the legacy `extras` macro for post-1996 Austrian German:

```
  \ifx\CurrentOption\bbl@tmpb
    \@namedef{extrasnaustrian}{%
      \@nameuse{@extrasgerman}%
    }
  \else
```

\extrasgerman-at-1901
\extrasgerman-austria-1901
Now newer alias names for pre-1996 Austrian German:

```
    \ifx\bbl@german@region\bbl@german@region@at
      \ifbbl@german@tradspelling
        \@namedef{extrasaustrian}{%
           \@nameuse{@extrasgerman}%
        }
        \@namedef{extras\CurrentOption}{%
           \@nameuse{extrasaustrian}%
        }
      \else
```

\extrasgerman-at
\extrasgerman-austria
Then, the newer `extras` macro for post-1996 Austrian German:

```
        \@namedef{extrasnaustrian}{%
           \@nameuse{@extrasgerman}%
        }
        \@namedef{extras\CurrentOption}{%
           \@nameuse{extrasnaustrian}%
        }
      \fi
    \fi
  \fi
\fi
```

\extrasswissgerman
Finally, same for Swiss German; first, the legacy `extras` macros for pre-1996 Swiss German:

```
\ifx\CurrentOption\bbl@opt@swissgerman
  \@namedef{extrasswissgerman}{%
      \@nameuse{@extrasgerman}%
  }
\else
```

`\extrasnswissgerman` Then, the legacy `extras` macro for post-1996 Swiss German:

```
  \def\bbl@tmpa{nswissgerman}
  \ifx\CurrentOption\bbl@tmpa
    \@namedef{extrasnswissgerman}{%
       \@nameuse{@extrasgerman}%
    }
  \else
```

`\extrasgerman-ch-1901` Now newer alias names for pre-1996 Swiss German:

`\extrasgerman-switzerland-1901`

```
     \ifx\bbl@german@region\bbl@german@region@ch
        \ifbbl@german@tradspelling
           \@namedef{extrasswissgerman}{%
              \@nameuse{@extrasgerman}%
           }
           \@namedef{extras\CurrentOption}{%
              \@nameuse{extrasswissgerman}%
           }
        \else
```

`\extrasgerman-ch` Then, the newer `extras` macro for post-1996 Swiss German:

`\extrasgerman-switzerland`

```
           \@namedef{extrasnswissgerman}{%
              \@nameuse{@extrasgerman}%
           }
           \@namedef{extras\CurrentOption}{%
              \@nameuse{extrasnswissgerman}%
           }
        \fi
     \fi
  \fi
\fi
```

Register spelling state:

```
\ifbbl@german@tradspelling
    \expandafter\addto\csname extras\CurrentOption\endcsname{%
        \bbl@german@tradspellingtrue}
\else
    \expandafter\addto\csname extras\CurrentOption\endcsname{%
        \bbl@german@tradspellingfalse}
\fi
```

`toss` For Swiss Standard German, we allow optionally to expand the ⟨ß⟩-related shorthands the Swiss way, i. e. as ⟨ss⟩ (globally, if the modifier or variety option `toss` is used or locally if `\tosstrue`).

```
\newif\ifbbl@toss\bbl@tossfalse
\def\bbl@tmpa{german-ch-1901}
```

First, query the modifiers for 1901 Swiss German:

```
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@swissgerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\def\bbl@tmpa{german-switzerland-1901}
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@swissgerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\ifx\bbl@mod@swissgerman\@undefined\else
  \@expandtwoargs\in@{,toss,}{,\bbl@mod@swissgerman,}
  \ifin@
    \tosstrue
  \fi
\fi
```

`\ntosstrue` `\ntossfalse` Now to 1996 Swiss German. For backwards compatibility reasons, we also still provide `\ntosstrue` which had been promoted in earlier versions of babel-german.

```
\newif\ifntoss\ntossfalse
\newif\ifbbl@ntoss\bbl@ntossfalse
\def\bbl@tmpa{german-ch}
```

Again, query the modifiers for 1996 Swiss German:

```
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@nswissgerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\def\bbl@tmpa{german-switzerland}
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@nswissgerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\ifx\bbl@mod@nswissgerman\@undefined\else
  \@expandtwoargs\in@{,toss,}{,\bbl@mod@nswissgerman,}
  \ifin@
    \tosstrue
  \fi
\fi
```

Now set `extras<lang>` for Swiss German (1901 and 1996) to consider toss setting. Also set toss at document begin if one of these is main language. This all needs to be done at document begin when we have the options set:

```
\AtBeginDocument{%
  \edef\tmpa{\localeinfo*{language.tag.bcp47}}%
  \edef\tmpb{de}%
  \ifx\tmpa\tmpb
      \edef\tmpa{\localeinfo*{region.tag.bcp47}}%
      \ifx\tmpa\bbl@german@region@ch
         \ifntoss
            \bbl@tosstrue
         \else
            \iftoss
                \bbl@tosstrue
            \else
                \bbl@tossfalse
            \fi
         \fi
```

```
      \fi
  \fi
\ifdefined\extrasswissgerman
  \addto\extrasswissgerman{%
    \iftoss\bbl@tosstrue\else\bbl@tossfalse\fi}%
\fi
\ifdefined\extrasnswissgerman
    \addto\extrasnswissgerman{%
        \ifntoss
           \bbl@tosstrue
        \else
            \iftoss
               \bbl@tosstrue
           \else
               \bbl@tossfalse
           \fi
         \fi
    }%
  \fi
}
```

capsz For German and Austrian Standard German, we allow optionally to uppercase ⟨ß⟩ with the capital eszett letter rather as ⟨SS⟩ if the font provides the glyph (if the modifier or variety option `capsz` is used).

```
\newif\ifnocapsz\nocapszfalse
\newif\ifbbl@capsz\bbl@capszfalse
```

Save current casing, since it needs to be reset afterwards (this is important particularly if casing had been altered externally, e.g. via `\babelprovide`).

```
\ifdefined\casing@german
  \let\save@casing@german\casing@german
\else
  \xdef\save@casing@german{de}
\fi
\ifdefined\casing@ngerman
  \let\save@casing@ngerman\casing@ngerman
\else
  \xdef\save@casing@ngerman{de}
\fi
\ifdefined\casing@naustrian
  \let\save@casing@naustrian\casing@naustrian
\else
  \xdef\save@casing@naustrian{de}
\fi
```

Now query the modifiers for 1996 German:

```
\def\bbl@tmpa{german-de}
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@ngerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\def\bbl@tmpa{german-germany}
\ifx\CurrentOption\bbl@tmpa
  \expandafter\let\expandafter\bbl@mod@ngerman\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\ifx\bbl@mod@ngerman\@undefined\else
```

```
  \@expandtwoargs\in@{,capsz,}{,\bbl@mod@ngerman,}
  \ifin@
    \capsztrue
  \fi
  \@expandtwoargs\in@{,nocapsz,}{,\bbl@mod@ngerman,}
  \ifin@
    \nocapsztrue
  \fi
\fi
```

and 1996 Austrian:

```
\newif\if@bbl@german@naustrian
\@bbl@german@naustrianfalse
\def\bbl@tmpa{german-at}
\ifx\CurrentOption\bbl@tmpa
  \@bbl@german@naustriantrue
  \expandafter\let\expandafter\bbl@mod@naustrian\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\def\bbl@tmpa{german-austria}
\ifx\CurrentOption\bbl@tmpa
  \@bbl@german@naustriantrue
  \expandafter\let\expandafter\bbl@mod@naustrian\csname bbl@mod@\bbl@tmpa\endcsname
\fi
\ifx\bbl@mod@naustrian\@undefined\else
  \@expandtwoargs\in@{,capsz,}{,\bbl@mod@naustrian,}
  \ifin@
    \@bbl@german@at@capsztrue
  \fi
  \@expandtwoargs\in@{,nocapsz,}{,\bbl@mod@naustrian,}
  \ifin@
    \nocapsztrue
  \fi
\fi
```

We also do it for `german` for the case of it meaning 1996:

```
\ifx\bbl@mod@german\@undefined\else
  \@expandtwoargs\in@{,capsz,}{,\bbl@mod@german,}
  \ifin@
    \@bbl@german@ge@capsztrue
  \fi
  \@expandtwoargs\in@{,nocapsz,}{,\bbl@mod@german,}
  \ifin@
    \nocapsztrue
  \fi
\fi
```

Now set `extras<lang>` for 1996 Austrian and German to consider caps setting. Also set caps at document begin if one of these is main language:

```
\AtBeginDocument{%
   \iflanguage{ngerman}{%
       \edef\tmpa{\localeinfo*{region.tag.bcp47}}%
       \ifx\tmpa\bbl@german@region@ch\else
          \ifcapsz\bbl@capsztrue\bbl@csarg\xdef{casing@\languagename}{de-x-eszett}\fi
       \fi
   }{%
```

```
      \ifbbl@german@newterms
        \edef\tmpa{\localename}%
        \ifx\tmpa\bbl@opt@german
          \ifcapsz\bbl@capsztrue\bbl@csarg\xdef{casing@\languagename}{de-x-eszett}\fi
        \fi
      \fi
  }%
  \ifdefined\extrasngerman
    \addto\extrasngerman{%
       \ifcapsz\bbl@capsztrue\bbl@csarg\xdef{casing@ngerman}{de-x-eszett}%
        \else\ifnocapsz\bbl@csarg\xdef{casing@ngerman}{de}\fi\bbl@capszfalse\fi}%
  \fi
  \ifbbl@german@newterms
     \ifdefined\extrasgerman
         \addto\extrasgerman{%
           \if@bbl@german@ge@capsz\bbl@capsztrue\bbl@csarg\xdef{casing@german}{de-x-eszett}%
            \else\ifnocapsz\bbl@csarg\xdef{casing@german}{de}\fi\bbl@capszfalse\fi}%
       \fi
  \fi
}
\if@bbl@german@naustrian
  \AtBeginDocument{%
     \addto\extrasnaustrian{%
         \if@bbl@german@at@capsz\bbl@capsztrue\bbl@csarg\xdef{casing@naustrian}{de-x-eszett}%
         \else\ifnocapsz\bbl@csarg\xdef{casing@naustrian}{de}\fi\bbl@capszfalse\fi}%
  }
\fi
```

\@noextrasgerman The macro \@noextrasgerman holds all the default noextras setting. This is an internal macro that is inherited and modified by the following macros for the respective language varieties.

```
\@namedef{@noextrasgerman}{%
```

First, we deactivate the " character and thus turn the shorthands off again outside of the respective variety:

```
  \bbl@deactivate{"}%
```

Also, undo redefinition of umlaut accent macro \" to lower the umlaut dots,

```
  \umlauthigh
```

and turn off \frenchspacing:

```
  \bbl@nonfrenchspacing
}
```

Depending on the option with which the language definition file has been loaded, a respective \noextras* macro is defined. Each of those is identical: it simply inherits \@noextrasgerman. However, the traditional names (german, ngerman, austrian, naustrian, swissgerman, and nswissgerman) are used as an intermediate layer, so redefining those will also redefine the newer aliases.

\noextrasgerman First, the legacy noextras macro for pre-1996 German German:

```
\ifx\CurrentOption\bbl@opt@german
  \@namedef{noextrasgerman}{%
     \@nameuse{@noextrasgerman}%
  }
```

```
\else
```

\noextrasngerman Then, the legacy `noextras` macro for post-1996 German German:

```
  \ifx\CurrentOption\bbl@opt@ngerman
    \@namedef{noextrasngerman}{%
        \@nameuse{@noextrasgerman}%
    }
  \else
```

\noextrasgerman-de-1901 Now newer alias names for pre-1996 German German:
\noextrasgerman-germany-1901

```
    \ifx\bbl@german@region\bbl@german@region@de
       \ifbbl@german@tradspelling
          \@namedef{noextrasgerman}{%
             \@nameuse{@noextrasgerman}%
          }
          \@namedef{noextras\CurrentOption}{%
             \@nameuse{noextrasgerman}%
          }
       \else
```

\noextrasgerman-de and post-1996 German German:
\noextrasgerman-germany

```
         \@namedef{noextrasngerman}{%
            \@nameuse{@noextrasgerman}%
         }
         \@namedef{noextras\CurrentOption}{%
            \@nameuse{noextrasngerman}%
         }
       \fi
    \fi
  \fi
\fi
```

Now deactivate casing if needed:

```
\ifdefined\noextrasgerman
  \if@bbl@german@ge@capsz
    \addto\noextrasgerman{%
        \bbl@capszfalse\bbl@csarg\xdef{casing@german}{\save@casing@german}}
  \fi
\fi
\ifdefined\noextrasngerman
  \ifbbl@capsz
    \addto\noextrasngerman{%
        \bbl@capszfalse\bbl@csarg\xdef{casing@ngerman}{\save@casing@ngerman}}
  \fi
\fi
```

\noextrasaustrian Same for Autrian: first, the legacy `noextras` macro for pre-1996 Austrian German:

```
\def\bbl@tmpa{austrian}
\def\bbl@tmpb{naustrian}
\ifx\CurrentOption\bbl@tmpa
  \@namedef{noextrasaustrian}{%
      \@nameuse{@noextrasgerman}%
  }
\else
```

\noextrasnaustrian Then, the legacy noextras macro for post-1996 Austrian German:

```
\ifx\CurrentOption\bbl@tmpb
  \@namedef{noextrasnaustrian}{%
    \@nameuse{@noextrasgerman}%
  }
\else
```

\noextrasgerman-at-1901 Now newer alias names for pre-1996 Austrian German:

\noextrasgerman-austria-1901

```
   \ifx\bbl@german@region\bbl@german@region@at
      \ifbbl@german@tradspelling
         \@namedef{noextrasaustrian}{%
            \@nameuse{@noextrasgerman}%
         }
         \@namedef{noextras\CurrentOption}{%
            \@nameuse{noextrasaustrian}%
         }
      \else
```

\noextrasgerman-at Then, the newer noextras macro for post-1996 Austrian German:

\noextrasgerman-austria

```
         \@namedef{noextrasnaustrian}{%
            \@nameuse{@noextrasgerman}%
         }
         \@namedef{noextras\CurrentOption}{%
            \@nameuse{noextrasnaustrian}%
         }
      \fi
   \fi
\fi
\fi
```

Also de-activate casing if needed:

```
\if@bbl@german@naustrian
  \if@bbl@german@at@capsz
    \addto\noextrasnaustrian{%
        \bbl@capszfalse\bbl@csarg\xdef{casing@naustrian}{\save@casing@naustrian}}
  \fi
\fi
```

\noextrasswissgerman Finally, same for Swiss German; first, the legacy noextras macros for pre-1996 Swiss German:

```
\ifx\CurrentOption\bbl@opt@swissgerman
  \@namedef{noextrasswissgerman}{%
     \@nameuse{@noextrasgerman}%
  }
\else
```

\noextrasnswissgerman Then, the legacy noextras macro for post-1996 Swiss German:

```
  \def\bbl@tmpa{nswissgerman}
  \ifx\CurrentOption\bbl@tmpa
    \@namedef{noextrasnswissgerman}{%
       \@nameuse{@noextrasgerman}%
    }
  \else
```

\noextrasgerman-ch-1901 Now newer alias names for pre-1996 Swiss German:

\noextrasgerman-switzerland-1901

```
    \ifx\bbl@german@region\bbl@german@region@ch
       \ifbbl@german@tradspelling
          \@namedef{noextrasswissgerman}{%
             \@nameuse{@noextrasgerman}%
          }
          \@namedef{noextras\CurrentOption}{%
             \@nameuse{noextrasswissgerman}%
          }
       \else
```

\noextrasgerman-ch Then, the newer `noextras` macro for post-1996 Swiss German:

\noextrasgerman-switzerland

```
          \@namedef{noextrasnswissgerman}{%
             \@nameuse{@noextrasgerman}%
          }
          \@namedef{noextras\CurrentOption}{%
             \@nameuse{noextrasnswissgerman}%
          }
       \fi
    \fi
  \fi
\fi
```

For the Swiss varieties, we need to deactivate `\toss`.

```
\ifx\bbl@german@region\bbl@german@region@ch
  \expandafter\addto\csname noextras\CurrentOption\endcsname{%
    \bbl@tossfalse}
\fi
```

The German hyphenation patterns can be used with `\lefthyphenmin` and `\righthyphenmin` set to 2.

```
\providehyphenmins{\CurrentOption}{\tw@\tw@}
```

9.5 Active Characters, Macros, and Shorthands

The following code is necessary because we need an extra active character. This character is then used as indicated in table 1.

In order to be able to define the function of `"`, we first define a couple of 'support' macros.

\dq We save the original double quotation mark character in `\dq` to keep it available, the math accent `\"` can now be typed as `"`.

Furthermore, we define some helper macros for contextual ⟨ß⟩ handling.

```
\begingroup \catcode`\"12
\def\x{\endgroup
  \def\dq{"}
  \def\@SS{\mathchar"7019 }
  \def\bbl@ss{\ifbbl@toss ss\else\textormath{\ss}{\@SS{}}\fi}
  \def\bbl@SS{{\ifbbl@capsz\MakeUppercase{\ss}\else SS\fi}}
  \def\bbl@sz{\ifbbl@toss sz\else\textormath{\ss}{\@SS{}}\fi}
  \def\bbl@SZ{SZ}
}
\x
```

Since we need to add special cases for hyperref which needs hyperref's `\texorpdfstring`, we provide a dummy command for the case that hyperref is not loaded.

```
\providecommand\texorpdfstring[2]{#1}
```

`\bbl@german@allowhyphenationbefore` `\bbl@german@allowhyphenationafter` We also define two helper commands to allow hyphenation before and after a character as defined in shorthands. These are similar to babel's `\bbl@allowhyphens` but differentiate the position:

```
\def\bbl@german@allowhyphenationbefore{\ifvmode\else\nobreak\fi}
\def\bbl@german@allowhyphenationafter{\nobreak\hskip\z@skip}
```

Now we can define the doublequote shorthands: the umlauts,

```
\declare@shorthand{german}{"a}{\textormath{\"{a}}{\ddot a}}
\declare@shorthand{german}{"o}{\textormath{\"{o}}{\ddot o}}
\declare@shorthand{german}{"u}{\textormath{\"{u}}{\ddot u}}
\declare@shorthand{german}{"A}{\textormath{\"{A}}{\ddot A}}
\declare@shorthand{german}{"O}{\textormath{\"{O}}{\ddot O}}
\declare@shorthand{german}{"U}{\textormath{\"{U}}{\ddot U}}
```

tremata,

```
\declare@shorthand{german}{"e}{\textormath{\"{e}}{\ddot e}}
\declare@shorthand{german}{"E}{\textormath{\"{E}}{\ddot E}}
\declare@shorthand{german}{"i}{\textormath{\"{\i}}%
                               {\ddot\imath}}
\declare@shorthand{german}{"I}{\textormath{\"{I}}{\ddot I}}
```

German ⟨ß⟩,

```
\declare@shorthand{german}{"s}{\bbl@ss}
\declare@shorthand{german}{"S}{\bbl@SS}
\declare@shorthand{german}{"z}{\bbl@sz}
\declare@shorthand{german}{"Z}{\bbl@SZ}
```

German and French/Swiss quotation marks,

```
\declare@shorthand{german}{"'}{\glqq}
\declare@shorthand{german}{"'}{\grqq}
\declare@shorthand{german}{"<}{\flqq}
\declare@shorthand{german}{">}{\frqq}
```

`\bbl@german@disc` and discretionary commands. Here we discriminate contemporary (post-1996) German from pre-1996 German (due to the hyphenation specifics). In the macro, `#1` is what is output for 1901 spelling in unhyphenated context (incl. math), `#2` is printed before the hyphen in hyphenated context. `#3` is printed in 1996 spelling in all contexts.

```
\def\bbl@german@disc#1#2#3{%
  \ifbbl@german@tradspelling
```

For pre-1996 spelling, we apply ck->k-k hyphenation for `"ck` and `"CK`, or the three-consonant rule (e.g., ll -> ll-l) for the other relevant shorthands. Therefore, `#2` is output if a hyphenation follows, otherwise `#1`:

```
    \textormath{%
      \bbl@german@allowhyphenationbefore\discretionary{#2-}{}{#1}%
      \bbl@german@allowhyphenationafter
```

No hyphenation in math, so unconditionally go for `#1`:

```
    }{#1}% math
  \else
```

For post-1996 spelling, we simply output ⟨c⟩ or ⟨C⟩ for `"c` and `"C`, or the two consonants in all contexts (passed as `#3`):

```
    #3%
  \fi
}
```

And here are the actual shorthands for these 1901 specifics:

```
\declare@shorthand{german}{"c}{\bbl@german@disc{c}{k}{c}}
\declare@shorthand{german}{"C}{\bbl@german@disc{C}{K}{C}}
\declare@shorthand{german}{"f}{\bbl@german@disc{f}{ff}{ff}}
\declare@shorthand{german}{"F}{\bbl@german@disc{F}{FF}{FF}}
\declare@shorthand{german}{"l}{\bbl@german@disc{l}{ll}{ll}}
\declare@shorthand{german}{"L}{\bbl@german@disc{L}{LL}{LL}}
\declare@shorthand{german}{"m}{\bbl@german@disc{m}{mm}{mm}}
\declare@shorthand{german}{"M}{\bbl@german@disc{M}{MM}{MM}}
\declare@shorthand{german}{"n}{\bbl@german@disc{n}{nn}{nn}}
\declare@shorthand{german}{"N}{\bbl@german@disc{N}{NN}{NN}}
\declare@shorthand{german}{"p}{\bbl@german@disc{p}{pp}{pp}}
\declare@shorthand{german}{"P}{\bbl@german@disc{P}{PP}{PP}}
\declare@shorthand{german}{"r}{\bbl@german@disc{r}{rr}{rr}}
\declare@shorthand{german}{"R}{\bbl@german@disc{R}{RR}{RR}}
\declare@shorthand{german}{"t}{\bbl@german@disc{t}{tt}{tt}}
\declare@shorthand{german}{"T}{\bbl@german@disc{T}{TT}{TT}}
```

Furthermore, and for contemporary orthography as well, we define some additional useful shorthands (hyphenation, line breaking and ligature control):

```
\declare@shorthand{german}{"-}{%
    \bbl@german@allowhyphenationbefore\-\bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"|}{%
  \texorpdfstring{%
    \textormath{% text
      \bbl@german@allowhyphenationbefore\discretionary{-}{}{\kern.03em}%
      \bbl@german@allowhyphenationafter
    }{}% math
  }{}% PDF string
}
\declare@shorthand{german}{""}{%
  \bbl@german@allowhyphenationbefore\discretionary{}{}{}%
  \bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"~}{%
  \texorpdfstring{%
    \textormath{% text
      \bbl@german@allowhyphenationbefore\mbox{-}%
      \bbl@german@allowhyphenationafter
    }{-}% math
  }{-}% PDF string
}
\declare@shorthand{german}{"=}{%
   \bbl@german@allowhyphenationbefore-\bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"/}{%
  \bbl@german@allowhyphenationbefore/\discretionary{}{}{}%
```

```
  \bbl@german@allowhyphenationafter
}
```

and some shorthands to support gender-sensitive spelling:

```
\declare@shorthand{german}{":}{%
  \bbl@german@allowhyphenationbefore:\bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"*}{%
  \bbl@german@allowhyphenationbefore*\bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"_}{%
  \bbl@german@allowhyphenationbefore\_\bbl@german@allowhyphenationafter
}
\declare@shorthand{german}{"x}{%
  \bbl@german@allowhyphenationbefore\mkgender\bbl@german@allowhyphenationafter
}
```

9.6 Compatibility of External Packages

`\mdqon` `\mdqoff` `\ck`

We define a couple of commands for reasons of compatibility with `german.sty` and `ngerman.sty`.

```
\def\mdqon{\shorthandon{"}}
\def\mdqoff{\shorthandoff{"}}
\def\ck{%
  \ifbbl@german@tradspelling
    \bbl@german@allowhyphenationbefore\discretionary{k-}{k}{ck}%
    \bbl@german@allowhyphenationafter
  \else
    ck%
  \fi
}
```

`\bbl@mk@class@alias`

For external packages that rely on legacy option names, we provide a method to transmit those (in addition to newer ones) in the global options list.

```
\def\bbl@mk@class@alias#1{%
  \def\bbl@class@alias{#1}%
  \def\bbl@tmp@classoptionslist{}%
  \bbl@foreach\@raw@classoptionslist{%
    \def\bbl@tmpa{##1}%
    \ifx\CurrentOption\bbl@tmpa
      \edef\bbl@tmp@classoptionslist{%
        \bbl@tmp@classoptionslist\zap@space\bbl@class@alias,##1 \@empty,}%
    \else
      \edef\bbl@tmp@classoptionslist{%
        \bbl@tmp@classoptionslist\zap@space##1 \@empty,}%
    \fi
  }%
  \let\@raw@classoptionslist\bbl@tmp@classoptionslist
  \def\bbl@tmp@classoptionslist{}%
  \bbl@foreach\@classoptionslist{%
     \def\bbl@tmpa{##1}%
     \ifx\CurrentOption\bbl@tmpa
        \edef\bbl@tmp@classoptionslist{%
          \bbl@tmp@classoptionslist\zap@space\bbl@class@alias,##1 \@empty,}%
```

```
    \else
       \edef\bbl@tmp@classoptionslist{%
         \bbl@tmp@classoptionslist\zap@space##1 \@empty,}%
    \fi
  }%
  \let\@classoptionslist\bbl@tmp@classoptionslist
```

For biblatex, we also adopt `\bbl@main@language` locally:

```
  \AddToHook{package/biblatex/after}{%
    \let\bbl@german@mkautolangbabel\blx@mkautolangbabel
    \def\blx@mkautolangbabel{%
      \let\bbl@main@language\bbl@class@alias
      \bbl@german@mkautolangbabel
    }%
  }%
}
```

The macro `\ldf@finish` takes care of looking for a configuration file, setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```
\ldf@finish\CurrentOption
```

9.7 Portmanteau `*.ldf` Files

Babel expects a ⟨*lang*⟩`.ldf` file for each ⟨*lang*⟩. So we create portmanteau ldf files for

- `german.ldf`
- `german-de.ldf`
- `german-germany.ldf`
- `german-de-1901.ldf`
- `german-germany-1901.ldf`
- `german-at.ldf`
- `german-austria.ldf`
- `german-at-1901.ldf`
- `german-austria-1901.ldf`
- `german-ch.ldf`
- `german-switzerland.ldf`
- `german-ch-1901.ldf`
- `german-switzerland-1901.ldf`

and the deprecated

- `austrian`
- `ngerman`
- `swissgerman`
- `nswissgerman`

All these files themselves load `babel-german.def`, which does the real work, with the appropriate option.

With `ngerman`, `naustrian`, and `nswissgerman`, we force german to 1901 with `glottonyms=auto`. This is simply determined by the existence of the following macro:

```
\def\bbl@german@force@legacy{}
```

With the newer options, we load exptl hyphenation patterns by default. This is determined by the existence of the following macro:

```
\def\bbl@german@xptl@patterns{}
```

The macro `\LdfInit` takes care of preventing that each `*.ldf` file is loaded more than once with the same option, checking the category code of the `@` sign, etc.

```
\LdfInit\CurrentOption{captions\CurrentOption}
```

Track whether we have 1901 spelling:

```
\newif\ifbbl@german@tradspelling
```

Set spelling and region params. First `german`, `germanb`, `german-de-1901` or `german-germany-1901`:

```
\bbl@german@tradspellingtrue
\def\bbl@german@region{DE}
```

Now, `ngerman`, `ngermanb`, `german-de` or `german-germany`:

```
\bbl@german@tradspellingfalse
\def\bbl@german@region{DE}
```

Now, `austrian`, `german-at-1901` or `german-austria-1901`:

```
\bbl@german@tradspellingtrue
\def\bbl@german@region{AT}
```

Now, `naustrian`, `german-at` or `german-austria`:

```
\bbl@german@tradspellingfalse
\def\bbl@german@region{AT}
```

Now, `swissgerman`, `german-ch-1901` or `german-switzerland-1901`:

```
\bbl@german@tradspellingtrue
\def\bbl@german@region{CH}
```

And finally, `nswissgerman`, `german-ch` or `german-switzerland`:

```
\bbl@german@tradspellingfalse
\def\bbl@german@region{CH}
```

Now load the common file;

```
\input babel-german.def\relax
```

Finally, set legacy class options if needed:
german-at-1901 and german-austria-1901,

```
\bbl@mk@class@alias{austrian}
```

german-at and german-austria,

```
\bbl@mk@class@alias{naustrian}
```

german-ch-1901 and german-switzerland-1901,

```
\bbl@mk@class@alias{swissgerman}
```

german-ch and german-switzerland,

```
\bbl@mk@class@alias{nswissgerman}
```

german-de-1901 and german-germany-1901,

```
\bbl@mk@class@alias{german}
```

as well as german-de and german-germany

993 `\bbl@mk@class@alias{ngerman}`

That's it! Fertig.

Change History

References

[1] Ammon, Ulrich, Hans Bickel & Alexandra Lenz. 2026. *Variantenwörterbuch des Deutschen: Die Standardsprache in Österreich, der Schweiz, Deutschland, Liechtenstein, Luxemburg, Ostbelgien und Südtirol sowie Rumänien, Namibia und Mennonitensiedlungen*. 2nd ed. Berlin, Boston: De Gruyter.

[2] Braams, Johannes and Bezos, Javier: *Babel*. http://www.ctan.org/pkg/babel.

[3] Clyne, Michael. 1991. German as a pluricentric language. In *Pluricentric Languages: Differing Norms in Different Nations*, ed. by Michael Clyne, 117–148. Berlin & Boston: De Gruyter Mouton.

[4] Dudenredaktion. 1991. *Duden: Die deutsche Rechtschreibung*. 20th ed. Mannheim: Bibliographisches Institut & F. A. Brockhaus.

[5] Dudenredaktion. 1996. *Duden: Die deutsche Rechtschreibung*. 21th ed. Mannheim: Bibliographisches Institut & F. A. Brockhaus.

[6] Deutschsprachige Trennmustermannschaft: *dehyph-exptl – Experimental hyphenation patterns for the German language*. https://ctan.org/pkg/dehyph-exptl.

[7] Johnson, Sally. 2005. *Spelling trouble: Language, ideology and the reform of German orthography*. Clevedon: Multilingual Matters.

[8] Partl, Hubert. 1988. *German TEX*, *TUGboat* 9(1): 70–72.

[9] Raichle, Bernd: *German*. http://www.ctan.org/pkg/german.

[10] Pégourié-Gonnard, Manuel et al.: *hyph-utf8 – Hyphenation patterns expressed in UTF-8*. https://ctan.org/pkg/hyph-utf8.

[11] Schmidt, Walter. 1998. TEX und die neue deutsche Rechtschreibung. *Die TEXnische Komödie* 10(2): 35–37.