1. This is a sample block of text designed to test indexing with the glossaries-extra·⌄ package. To test only \index· and theindex· environment, just use testidx·⌄ instead of testidx-glossaries·⌄. Alternatively, use \tstidxtoidx· to switch back to the original testidx·⌄ behaviour. This sample text is provided to help test the layout· of the index· (or glossary·) and the ·indexing applications, such as makeindex·⌄ or xindy·⌄, that may be used with the glossaries-extra·⌄ package. This example· text is just filler· (produced using \testidx· provided by the testidx·⌄ package) to pad out the document with instances of \index· interspersed throughout. You can find out more information from the testidx·⌄ user manual, which can be accessed using the texdoc·⌄ application.

2. The testidx-glossaries·⌄ package uses \gls· (or \glspl·) unless no text should be displayed when indexing occurs (in which case just \glsadd· is used). This test package doesn't make any modifications to \gls· (or \glspl·), but there are some minor changes to entry definitions and command use to allow this ·dummy text to test various cases. By default \gls· performs the indexing before the ·link text to prevent any unwanted interference caused by the whatsit· introduced by the indexing. However, it's possible that you may want the indexing to occur after the ·link text, even though it creates a whatsit·. This ·dummy text is designed to use a mixture of pre- and post-indexing to investigate the results (particularly with phrases· spanning a ·page break). The glossaries-extra·⌄ package (as from version 1.14) provides a new key "wrgloss" (for commands like \gls·), which may take the values "before" or "after". If this key is available, it will be used for the post-indexing examples·, otherwise the post-indexing will be performed manually using \glsadd· after the ·link text (which is displayed using \glshyperlink·). The testidx-glossaries·⌄ package just loads the base glossaries·⌄ package by default. Use the extra⌄︾ package option to also load glossaries-extra·⌄. The ·visual effects in this ·dummy text are provided to illustrate where the indexing occurs. The testidx-glossaries·⌄ package doesn't attempt to produce a well-designed index or glossary. Its purpose· is to help you test· your chosen design·, which is easier to do with a relatively small test document·, than with a large book·. The ·dummy text is intended to produce an index· that is at least three pages long to allow you to test the page headers and footers in a two-sided document. The "mcolindexgroup" style is set by default (rather than the "list" style) as that style more closely represents the style used in indexes. If you use the desc⌄︾ option, the style will instead be set to "indexgroup". You can change the style using \setglossarystyle·. By default the "tree" styles (including "mcolindexgroup") display the name field in bold in the glossary·. This is governed by the command \glstreenamefmt·, which may be redefined as appropriate. You can hide the visual effects with the hidemarks⌄︾ package option.

3. The actual place where the indexing occurs in this ·dummy text is marked with the symbol · (\tstidxmarker)· if there is no range· or ·cross-reference. The word· or phrase· adjacent to this marker· is the text being indexed.[1] A sub-entry is indicated with the symbol ⌄ (\tstidxsubmarker)· and a sub-sub-entry is indicated with the symbol ︾ (\tstidxsubsubmarker)·. If an encap· value is provided, both the text· and the marker· are typeset in the argument· of the corresponding command. (The text occurring in the document is also typeset within the argument of \tstidxtext·. The default value is to use a dark grey, but since the default values for the predefined· encaps used in this text all change the colour, the dark grey will only apply where the encap hasn't been set.) There are three encap· values used throughout this ·dummy text (unless you've used the notestencaps⌄︾ package option): tstidxencapi·⌄, tstidxencapii·⌄ and tstidxencapiii·⌄. (The default values use \textcolor·, so you might want to use the hidelinks⌄︾ option if you want to use the hyperref·⌄ package.) If you use the noseekey⌄︾

---

[1] The indexing may occur before or after the word· or phrase· (·link text) being indexed in this ·dummy text, but there's no space between the marker· and the term being indexed. Incidentally, this footnote· text was produced using the command \tstidxfootnote·, which you can redefined as required. (It defaults to just \footnote·.)

package option, a cross-referenced entry (using `\glssee`· or `\glsxtrindexseealso`·) is identified using the marker· ˆ (`\tstidxseemarker`)· and the cross-referenced information is displayed as a ·marginal note by default, with the term being indexed followed by the cross-reference·. For example, lyuk· has the marker · to show that the word "lyuk" was indexed. If the `noseekey`⌄⌄ package option is set, this is followed by the marker ˆ to show that a cross-reference to digraph· has also been indexed (with the details shown in the margin). This document has used the `seekey`⌄⌄ package option, which means that `\glssee`· and `\glsxtrindexseealso`· aren't used in the ·dummy text. Instead the "see" and "seealso" keys provided in the entry definition were used to perform the cross-reference· at the time the entry was defined, and so no markers or marginal notes are displayed. A sub-level cross-reference· is identified with the marker· ˆ⌄ (`\tstidxsubseemarker`)· and the marginal note displays the main term followed by the sub-term (separated by the symbol ▷). The marker· used for the start of a range is ⌈ (`\tstidxopenmarker`)· and the marker· used for the end of a range is ⌋ (`\tstidxclosemarker`)· unless the entry is a sub-level, in which case the marker· for the start of the range is ⌈ (`\tstidxopensubmarker`)· and the marker· used for the end of a range is ⌋ (`\tstidxclosesubmarker`)·, or for a sub-sub-level ⌈ (`\tstidxopensubsubmarker`)· and ⌋ (`\tstidxclosesubsubmarker`)·. There are no tests for any further sub-levels. Although `xindy`·⌄ allows more than three levels (`makeindex`·⌄ doesn't), it's somewhat excessive· to go below a sub-sub-level. You'll have to add your own tests for anything deeper. Watch out for lonely⌄ sub-items.

4. Here's an example· of the start of a range⌈ but remember that a range must also have an end, so make sure that block· 9 has been included in this ·dummy text, which closes this example·. When the `verbose`⌄⌄ option is used with just `testidx`·⌄, the indexing information will show the argument· being passed to `\tstindex`· (which will cause ·overfull lines). Since the `testidx-glossaries`·⌄ package doesn't use `\tstindex`·, the `verbose`⌄⌄ package option will instead write information to the transcript file showing the label, name field, sort field, text field, parent field and see field for each entry as it's defined. (The parent and see fields may be empty). If the `tex`⌄⌄ package option is used, the sorting may take a long time (since TeX is being used to sort the entries). In this case, the `verbose`⌄⌄ package option will also write information while it's sorting the entries, so that you know TeX is still doing something and hasn't got stuck. This document has the `noverbose`⌄⌄ package option set.

5. Now that the preliminaries have been dispensed with in the previous paragraphs·, we can get on to some serious waffle· to act as filler· text because this really needs some padding· in order to get a decent sized index· with lots of locations. I did consider using just plain old ·lorem ipsum (like the `lipsum`·⌄ package), but it gets a bit boring after a while, and it's easier to check the indexing has been performed successfully if you can understand the text. Of course, this doesn't help those who don't know any English, but at least they're no worse off than they would have been with random gibberish·, at least, I hope not. In other words, if I could just clarify· what I'm trying to say here, in a confidential· and not too roundabout· fashion· — ·between you, me and the gatepost — is please don't consider this to be an illustration· of my stunning wit·, eloquence· and ·way with words because I'm shamelessly contravening the ·creative writing adage· (or possibly motto·) of ·cut to the chase, remove excessive verbiage· and ·get to the point. I shall take care to hide this drivel· from my ·creative writing tutor· and fellow writers, so ·keep mum and don't grass· on me because that just won't be fair, and it might distress· them to a certain extent. I shall not be assailed· by indecision and will aspire· to find the ·mot juste. Where was I? Oh, yes, padding·. I'm trying to make this paragraph· quite long, not because I have any pretensions of being the next James Joyce·⌄ and competing with *Ulysses*·⌄, but because one of the things we need to check for is what happens with paragraphs that span a ·page break. (If you're feeling particularly daring, try out the starred version of `\testidx`·, although some of the blocks, such as block· 6, have some sneaky paragraph· breaks that won't be suppressed.) TeX·'s asynchronous

·output routine can cause things to go a bit ·out of whack, so lengthy paragraphs in this example· document increase the chances of testing for these occurrences. Whether or not this particular paragraph· actually spans a ·page break does, of course, depend on various things including your document properties⌄, such as the ·page dimensions, ·font family and ·font size. If it turns out that this paragraph· has spanned a ·page break, you might want to check the terms indexed here to make sure they have the correct page numbers listed. Something else that you might want to check, while you're at it, is what's happened to the location list· for the word "paragraph·", as I've used different encap· values for it in various places in this example· paragraph·. If you are using `makeindex`·⌄, you might notice some warnings about ·multiple encaps, and the ·page number may be duplicated in the location list. If you are using `xindy`·⌄, then it will discard duplicate page numbers and give preference to the first defined attribute in whatever `xindy`·⌄ module you're using. However, be careful if a range· overlaps a different encap·. Remember that there's a difference between an index· and a concordance·. If you just index pertinent places, there's less likelihood of conflicting encaps. This is the end of a paragraph· that was written to deliberately upset `makeindex`·⌄. Mean, aren't I?

6. On the subject of `xindy`·⌄, if you want to use it with this example· document, you'll need to add the encap· values used in this ·dummy text as allowed attributes. (Don't forget you'll also need the `xindy`⌄⌄ package option when you load `testidx-glossaries`·⌄.) With the `glossaries`·⌄ package, this means using `\GlsAddXdyAttribute`· with the encap name as the argument. For example

```
\GlsAddXdyAttribute{tstidxencapi}
\GlsAddXdyAttribute{tstidxencapii}
\GlsAddXdyAttribute{tstidxencapiii}
```

This sets up allowed encap values and how they should be formatted. The ordering of the allowed attributes· here gives the `tstidxencapi`·⌄ encap precedence in the event of a ·multiple encaps clash, since it's the first one in the list. Check out the difference between using `xindy`·⌄ and `makeindex`·⌄ on this document. Also check the difference between using `makeglossaries`·⌄ and `makeglossaries-lite`·⌄ to build this document. The page separator⌄ is given by `\delimN`·, and the range separator⌄ is given by `\delimR`·. On the subject of location lists, the word passim· (meaning "here and there") is sometimes used to tidy up ragged lists. For example, the locations "1, 3, 4, 6, 7" may look neater as "1-7 passim", which indicates references are scattered (here and there) throughout that range. The `bib2gls`·⌄ application may be used with the `glossaries-extra`·⌄ package instead of `makeindex`·⌄ or `xindy`·⌄. Use the `bib2gls`⌄⌄ package option to set up this dummy text to use it. Note that this option doesn't make use of the prefix commands `\tstidxindexmarkerprefix`· or `\tstidxindexmathsymprefix`·. The `prefix`⌄⌄ package option will instead alter the way the bib files are loaded to produce a similar grouping.

7. Computer algorithms can sometimes have difficulties with localisation·. They can be tripped up by ·input encoding issues and digraphs· (such as the Welsh ll· digraph in llan·, the Dutch ij· digraph in lijnbus· and ijsvrij·, and the dz· digraph in the Hungarian dzéta· and Polish dzwon·) or trigraphs· (such as the Hungarian dzs· trigraph in dzsóker· and dzsungel·), so this paragraph· is designed to provide some examples for testing various Latin alphabets. If you enable both UTF-8 (either with `inputenc`·⌄ or using XeLaTeX/LuaLaTeX) and the `digraphs`⌄⌄ option, then the examples above will use the "ll", "ij" and "dz" glyphs· (*if supported*) for the digraphs· (but not for "dzs", which is a trigraph·). Remember that you'll also need a font that supports those glyphs. Other digraphs include the Welsh dd·, ff·, and ng·, the Hungarian ly· (in lyuk· mentioned earlier) and the Polish cz·, but these don't use glyphs in the sort value. Now for some more nonsense· text to pad the index. We, the élite· who discovered the æsthetic· delights of

TEX·, must not become blasé· about being the protégé· of the great Donald Knuth·⌄. It may stagger the clientèle· of the commercial world· to discover our résumé· (after foraging for it in our natty ·attaché case) while we sample a vol-au-vent· or two· at the soirée· in the ·pied-à-terre with the delightful phœnix·-themed décor·, daft· oak· ·dado rail and færie· façade· that has stunned many an æthereal· débutante· sporting a ·bergère hat, but it would be naïve· to fall for such a fœtid· cliché·. This paragraph· is in a state of déshabillé·. Like a sculpture· of Venus· in a négligée·, it's transparently obvious that this paragraph· is provided for the sole purpose of ogling· ·extended Latin characters and testing how `xindy`·⌄ and `makeindex`·⌄ compare. Time for a quick trip to the café· for an anæmic· ·cup of tea (to recover· from our travels) with Anders Jonas Ångström·⌄ (but don't let it scald your œsophagus·) and then off to find a zoo· in Östergötland·⌄, so we can get to the end of the alphabet·. (We may even see an adventurous aardvark· or a lucky llama· or a rhinoceros· eating rhubarb·.) Perhaps then we should go over to Ängelholm·⌄ and head off across the Øresund·⌄ bridge and resume· our search for some more examples. We'll go on a ·whistle-stop tour around Tårnby·⌄, Rødovre·⌄, Næstved·⌄ and Ølstykke-Stenløse·⌄. Afterwards, we'll fly· to Poland·⌄ (possibly in an aeroplane·—if passengers would like to look out of their window·, they'll see we're passing over Aßlar·⌄ and Bad Gottleuba-Berggießhübel·⌄) and then we'll say "cześć·" to Łódź·⌄, and visit Świętokrzyskie·⌄ (that one will cause a problem for certain font encodings because of the ogonek· and will be omitted if you use the default OT1⌄ encoding, but not if you use the `fontenc`·⌄ package with, for example, the `T1`⌄︾ option), Żory·⌄, Żelechów·⌄, Łobez·⌄, Głogów·⌄ (not to be confused with Glasgow·⌄), Ćmielów·⌄, Ścinawa·⌄ and Świdnica·⌄. Then let's sail· to Iceland·⌄ (possibly in a ship·) and visit the lakes of Iceland·⌄, such as Ölvesvatn·⌄, Úlfsvatn·⌄, Ánavatn·⌄, Másvatn·⌄, Þríhyrningsvatn·⌄ (that one starts with a thorn (þ)·), Sigríðarstaðavatn·⌄ (that one has an eth (ð)·—those last two will also be omitted if you use the default OT1· font encoding·), Grænavatn·⌄, Árneslón·⌄ and Íshólsvatn·⌄. If you are using this with `xindy`·⌄ and UTF-8·, try this out with a different language option, for example `-L swedish (xindy)`·⌄︾ or `-L danish (xindy)`·⌄︾ or `-L german-duden (xindy)`·⌄︾ or `-L german-din5007 (xindy)`·⌄︾ or `-L polish (xindy)`·⌄︾ or `-L icelandic (xindy)`·⌄︾. If you want to use `makeindex`·⌄ instead of `xindy`·⌄, then the package option `german`⌄︾ or `ngerman`⌄︾ will allow you to use `makeindex`·⌄'s `-g (makeindex)`·⌄︾ option. You currently have the `sanitize`⌄︾ option on. This means that the words containing UTF-8· characters will first be sanitized before being passed to `\tstindex`·, which will allow you to test how well the ·indexing application sorts UTF-8· characters. If you'd rather test how `\index`· writes these characters to the file read by the indexing application, use the `nosanitize`⌄︾ option instead. This may cause the UTF-8· characters to be written in terms of `\IeC`·. The `nosanitize`⌄︾ option doesn't have an effect when used with the `bib2gls`⌄︾ package option, since the sort value isn't written to an external file.

8. Don't forget there's also a ·number group, so let's have some numbers. *the Hitchhiker's Guide to the Galaxy*·⌄ has of course propelled the number 42· to stardom, as the answer to life·, the universe· and everything. We usually deal in base 10·, but sometimes base 16· is useful to programmers, and computers prefer base 2· (and 2· is the only even· prime number·). A century· in cricket· means 100· runs, and in the calendar· 100· years. If you're using `xindy`·⌄, the "glsnumbers" letter group (with the title "Numbers") will automatically be added to your `.xdy` file unless you switch off this feature with the `noglsnumbers`⌄︾ package option. Whilst we're on the subject of numbers, let's try out some equations.

$$\cdot E = mc^2 \tag{1}$$

Since this document is using `amsmath`·⌄, let's try out the `align`· environment:

$$\cdot f(\vec{x}) = \alpha\cdot \sum\cdot_{i}^{\cdot n} x_i + \beta\cdot \sum_{i}^{n} x_i^2 + \gamma\cdot \tag{2}$$

$$\frac{\partial\cdot f}{\partial x_j} = \alpha + 2\beta x_j \tag{3}$$

(Note how the subscripts and superscripts can be affected by material inserted between the symbol and the sub- and superscripts. With the `glossaries`·⌄ package, this can be avoided by using the final optional argument of commands like `\gls`· to bring the scripts into the ·link text, but not if the scripts also include `\gls`·, as the superscript does here.) If this document hadn't loaded the `amsmath`·⌄ package, we would have had to use the `eqnarray`· environment instead. Incidentally, that's just a regular partial derivative symbol ∂·. Not to be confused with the spin-weighted partial derivative ð·, which can be displayed here as this document has loaded the `amssymb`·⌄ package. Now I've been a bit fancy here and used multiple bib files to define the glossary entries. The maths symbols are defined in their own separate bib file. This document has used the `prefix`⌄⌄ package option, which means that `\tstidxmakegloss`· is defined to perform separate sorts for each resource block, which enables the maths symbols to appear in their own group. If you haven't used `\tstidxmakegloss`· then the above won't apply. If you don't use the `bib2gls`⌄⌄ package option, then the `prefix`⌄⌄ package will instead insert `>` in front of the sort key so I can get `xindy`·⌄ to create a special group for the maths symbols. Here's the code you can add to your document to implement it:

```
\GlsAddLetterGroup{Maths}{:prefixes (">") :before "glsnumbers"}
```

I've done something similar with the markers· where I've used < as the prefix·.

9. Let's re-cover· old ground and talk about ranges again. This is the end of the range⌋ example· from block· 4. There's not much else to say about this block· really. It's quite boring, isn't it? However, you'll need it if you've included block 4. Unless you're testing for a mismatched range, of course. That might be quite interesting, possibly, but I'm not going to ·hold my breath.

10. Now this is going to be hard to believe — in fact I'm totally gobsmacked· and utterly astounded· — but I've discovered that we're still missing some ·letter groups, and I've run out of anything quaint· to say, so I'm going to have to yatter· for a while longer, which will probably make you yawn· and fall asleep·. What shall we talk about? My quirky· badinage· is about to expire·. How about a story·? Here's one I made up for my friend Paulo Cereda·⌄ in TeX·.SE chat because he likes ducks and is the creator of `arara`·⌄. So, are you sitting comfortably? Then let's begin·. By the way, before I forget·, it's called *Sir Quackalot and the Golden Arara*·⌄ and is the first story in *the Adventures of Sir Quackalot*·⌄. It's a tale· of adventure· and derring-do·. The hero· of the story· is Sir Quackalot·⌄, in case you can't tell from the title·. ·Once upon a time, a long time ago in a far away land·, there lived a knight·. He was handsome·, he was bold·, he was brave·. He was — a duck·. His name· was Sir Quackalot·⌄. One day the Fairy Goose·⌄ appeared. "Brave knight·," she said. "A terrible plight· has fallen on the land·. The evil OgRe· has stolen the Golden Arara·. Only you can save it." (That's a reference· to TeX·'s ·output routine, if you missed it.) "It will be a perilous· quest·, but find the Mighty Helm of Knuth· and the Legendary Sword· xor· to aid you." (Ask David Carlisle·⌄ about the xor· reference·.)

11. So Sir Quackalot·⌄ set out on his quest·. (This is the continuation· from the previous block·, for any newcomers· who have just turned up.) He soon arrived at the Bog of Eternal Glossaries· (that's a reference to my `glossaries`·⌄ package, and it's also a nod· to the Bog of Eternal Stench·⌄ in *Labyrinth*·⌄), in the centre· (or center· for those of you ·across the pond)

of which was suspended the Mighty Helm of Knuth·, but Sir Quackalot·⌄ was learned in the lore· of installing `Perl`·⌄ and was able to leap upon the magical· raft· `makeglossaries`·⌄ and steer his way through the external· ·indexing applications and their many arguments. (That's supposed to be a pun·, but it's ·bad form to explain the joke·, and it wasn't even particularly witty·. Incidentally, Joseph Wright·⌄ makes a cameo· at this point with the exclamation· "fetchez la vache!" but you'll have to ask Paulo Cereda·⌄ what that's all about. It wouldn't surprise me if it had something to do with ·Monty Python.)

12. Anyway, where were we? Oh, yes. He (that's Sir Quackalot·⌄ we're talking about, if you've only just joined us) snatched up the Mighty Helm of Knuth· and escaped from the perilous· bog·. Soon he came to the Dread Vale of the Editors·, guarded at either end by the ever-quarrelling leviathans· `Emacs`·⌄ and `Vi`·⌄. As he approached the vale·, Emacs uttered the ·magic incantation that sent forth the butterflies of chaos·. (I know "doom·" is more appropriate but, as is ·common knowledge, chaos· is a butterfly· motif·.)

13. But Sir Quackalot·⌄ was protected by the Mighty Helm of Knuth· and raced past into the vale·, where he found the Legendary Sword· xor· in the centre of the great `longtable`·⌄. (Ooh·, I've started a sentence· with a conjunction·. How naughty· is that?) With a great leap· and a bound·, Sir Quackalot·⌄ plucked out the sword· and headed for the far end of the vale·. Up pounced `Vi`·⌄ and belched forth a myriad· of clones· that bore down on Sir Quackalot·⌄. But, brandishing the sword· xor·, Sir Quackalot·⌄ sliced them down. (There's some repetition· there, but hopefully no one's noticed. There's even more coming up in the next block·.)

14. Sir Quackalot·⌄ escaped from the Dread Vale of the Editors· and set off up the path that led to the evil OgRe·'s lair·. As Sir Quackalot·⌄ approached, there was a fearful roar·, and the OgRe· pounced on Sir Quackalot·⌄. The brave knight· raised his powerful sword· xor· and brought it down on the OgRe·, destroying him. Sir Quackalot·⌄ rescued the Golden Arara· and the land· was once more restored to peace· and harmony· and paragraphs· were able to span· ·page breaks without fear·. The End·. Don't miss the next thrilling· adventure· *Sir Quackalot and the Hyper Lake of Doom*·⌄ where our seaborne· ·intrepid hero meets a quixotic· seal· with a zither· (a zealous· fan· of *The Third Man*·⌄), a youthful· ·sea lion with a magic· yo-yo·, and a wily· wombat· warrior· with a laser-guided· ·sealant gun. Can they defeat the villainous·, zany· zoologist· sailing a xebec· across the sea· bearing canisters of xenon·, xylem· and xylene·? Oh, zounds·! He's ashore· wearing a ·zoot suit and smoking a zucchini· whilst playing a xylophone·. As one ·anonymous reviewer said, it's as exhilarating· as watching a yuppie· eating a ·yule log soaked in ·yoghurt. Hmm, yummy·—or yuck·, depending on your tastes. Don't forget to check· you have some chalk· so we can write ddisgynedig· and ddyrchafedig· in Nghaerdydd·⌄ (over in Nghymru·⌄) because I want a few more words with digraphs·, and then we can take a ffotograff· of Ffestiniog·⌄.

15. Oh, did I tell you about the ·vice-president who was a Viking· in a vignette·? No? Well, I can't quite remember the story· myself, but it had something to do with a ·vice admiral with a ·Victoria plum and a ·viceroy with a ·Victoria sponge, or was it ·vice versa? The ·vice chancellor preferred vichyssoise·. For ·letter ordering use the `-l` (`makeindex`)·⌄⌄ option with `makeindex`·⌄ or the `ord/letorder` module with `xindy`·⌄ (`-M ord/letorder` (`xindy`)·⌄⌄). If you omit this, the default ·word ordering is used. The ordering in the *Compact Oxford English Dictionary*·⌄ (third edition, revised) for these words are: vice admiral, vice chancellor, vice-president, viceregal·, viceroy, vice versa. Quick quiz·. Can you get `makeindex`·⌄ or `xindy`·⌄ to reproduce that order?

16. This is the final block· of dummy text provided by the `testidx`·⌄ package. This block contains the close of a range· that was started in block 1. Fun, wasn't it?

Glossary

Maths | Markers | Numbers | A | B | C | D | Dd | Dz | Dzs | Đ | E | F | Ff | G | H | I | IJ | J | K | L | Ll | M | N | Ng | O | P | Q | R | S | T | U | V | W | X | Y | Z | Æ | Œ | Þ | Ø | Ł