

# nanicolle for herbarium specimen labels<sup>\*</sup>

杨宇昌 (Yuchang YANG)<sup>✉</sup>



July 8th, 2020      ver. 2.02

Herbarium specimens are plant material well preserved as samples of plant populations. The plant material itself is insufficient to reflect all important information of the population, so it is required to prepare a fully recorded *collecting label* along with the material. Plant taxonomists may study a herbarium specimen and determine which species the population belongs to, and such comments are presented on *identifying labels* and then pasted on the specimen sheet.

nanicolle is a L<sup>A</sup>T<sub>E</sub>X document class for typesetting collecting and identifying labels for herbarium specimens, in Chinese style or in western style using English. Labels mentioned hereinafter are by default in western style, which uses a really different layout from the Chinese version (see the Chinese manual for details). Collecting and identifying labels can be typeset by macros `\collect` and `\identify` (NB: both lower case!) respectively. The output file can be printed on A4 papers (297 × 210 mm).

Documents in this class can only be compiled with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X.

nanicolle is distributed under the terms of L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL) 1.3c<sup>1</sup>. It depends on package collection C<sub>T</sub><sub>E</sub>X as well as packages including calc, color, geometry, graphicx, listofitems, multicol<sup>2</sup>, xstring, etc.

## Contents

<b>1</b>	<b>Structure of documents in nanicolle class</b>	<b>2</b>
<b>2</b>	<b>The macro <code>\collect</code> for collecting labels</b>	<b>2</b>
<b>3</b>	<b>The macro <code>\identify</code> for identifying labels</b>	<b>4</b>
<b>4</b>	<b>Other issues</b>	<b>5</b>
4.1	Store original data in a spreadsheet software . . . . .	5
4.2	Set the printer correctly . . . . .	5
<b>5</b>	<b>Change history</b>	<b>5</b>
<b>6</b>	<b>A full example using nanicolle</b>	<b>5</b>

---

<sup>\*</sup>Github repository: <https://github.com/Mikumikunisiteageru/nanicolle>.

<sup>✉</sup>Email address: [yang.yc.allium@gmail.com](mailto:yang.yc.allium@gmail.com).

<sup>1</sup>Details of the license are available on <http://www.latex-project.org/lppl.txt>.

<sup>2</sup>Since applying nanicolle document class leads to indirect use of multicol package, if one wishes to employ nanicolle for commercial use, he/she may be subject to moral obligation of multicol (see the notice in its source file for details).

# 1 Structure of documents in nanicolle class

A document in the nanicolle class should be a plain text file with the extension `.tex`. The document generally should consist of the five following parts:

1. Document class loader, i.e. `\documentclass[⟨option list⟩]{nanicolle}`. *⟨option⟩*s separated by comma `,` in the *⟨option list⟩* control the behavior of the document. For example, `nomap` suppresses the map in the collecting labels, and `autoduplicate` repeats the collecting labels according to the *⟨duplicate count⟩* (vide infra). When no *⟨option⟩* is specified, one can simply write `\documentclass{nanicolle}` instead.
2. Optional `\heading{⟨heading⟩}` and `\subheading{⟨subheading⟩}` in the preamble to customize content of the headings on each collecting label. Both the length of *⟨heading⟩* and *⟨subheading⟩* should not exceed the line width. One can make a one-line heading by skipping the `\subheading`. If both `\heading` and `\subheading` are skipped, no heading will occur on the collecting labels.
3. `\begin{document}`, as the name implies.
4. Lines starting with either `\collect` or `\identify` to typeset collecting or identifying labels respectively. Syntaxes of these macros will be declared in Section 2 and Section 3.
5. `\end{document}`.

# 2 The macro \collect for collecting labels

The syntax of the macro `\collect` is

```
\collect → ⟨record number⟩ → ⟨collectors⟩ → ⟨collecting number⟩
          → ⟨collecting date⟩ → ⟨family⟩ → ⟨vernacular name⟩ → ⟨scientific name⟩
          → ⟨photo number⟩ → ⟨duplicate count⟩ → ⟨location⟩ → ⟨longitude⟩
          → ⟨latitude⟩ → ⟨altitude⟩ → ⟨habitat⟩ → ⟨life form⟩ → ⟨height⟩
          → ⟨diameter at breast height⟩ → ⟨note⟩
```

where `→` denotes a horizontal tab (U+0009, the character that the tab key inputs). Each `\collect` macro followed by its parameters must exclusively occupy a single line without comment sign, and the line should begin immediately with the macro. Parameters can be left empty (some cannot), but even so the tabs separating them should never be omitted. The requirements of each parameter of `\collect` are listed as follows.

1. *⟨record number⟩*: Only for the convenience of data organizing, not printed on collecting label.
2. *⟨collectors⟩*: Names of the persons or the team who collected the specimen. When there were more than one collectors, all their names should be listed here. When a team was involved, it is strongly suggested to list its members' names in parentheses after the team name. *⟨collectors⟩* cannot be empty.
3. *⟨collecting number⟩*: Serial number indexing the specimen collection of the first component of *⟨collectors⟩*. Traditionally it is suggested to apply sequences of increasing integers starting from 1 to *⟨collecting number⟩*.
4. *⟨collecting date⟩*: Date when the specimen was collected, better expressed in arabic numerals in the formula *⟨year⟩.⟨month⟩.⟨date⟩*. Parameter *⟨collecting date⟩* cannot be empty.

5. *<family>*: Preliminary scientific name (in Latin) of the family.
6. *<vernacular name>*: Preliminary vernacular name of the species, in local language. Not printed on collecting label.
7. *<scientific name>*: Preliminary scientific name (in Latin) of the species, better with no author citation especially when uncertain. Unless empty, *<scientific name>* follows the formula *<generic part><specific part><infraspecific part>*.

In the formula above, there are two possible patterns for the *<generic part>*:

- i. *<genus name>*; ii. *×<genus name>*.

*<specific part>* has nine possible patterns:

- i. *␣sp.*; ii. *␣sp.␣nov.*; iii. *␣<species epithet>*; iv. *␣×<species epithet>*;
- v. *␣aff.␣<species epithet>*; vi. *␣aff.␣×<species epithet>*;
- vii. *␣cf.␣<species epithet>*; viii. *␣cf.␣×<species epithet>*;
- ix. *␣'<cultispecies name>'*.

where *␣* denotes a normal space (U+0020). *<infraspecific part>* can be not empty when and only when *<specific part>* fits its pattern iii or iv, at this time having four possible patterns:

- i. *␣subsp.␣<subspecific epithet>*; ii. *␣var.␣<varietal epithet>*;
- iii. *␣f.␣<form epithet>*; iv. *␣'<cultivar name>'*.

Control sequences like *\textit* manually designating font style are unavailable in *<scientific name>*.

8. *<photo number>*: Only for the convenience of data organizing, not printed on collecting label.
9. *<duplicate count>*: Count of specimen duplicates with the same *<collecting number>*, expressed in arabic numerals; not printed on collecting label. When *autoduplicate* is loaded as an *<option>* of the document class, each *\collect* macro automatically makes *<duplicate count>* duplicate collecting labels.
10. *<location>*: Location where the specimen was collected, expressed in natural way, providing as much detailed information as possible, including country, province, city, town, etc., and the specific locality (probably with respect to some landmarks), so that other researchers can locate the population. *<location>* cannot be empty.
11. *<longitude>*: Longitude value of the *<location>*, a decimal floating number in degree (without unit), positive for east, negative for west. Sexagesimal expression (in degree, minute, and second) are not acceptable.
12. *<latitude>*: Latitude value of the *<location>*, a decimal floating number in degree (without unit), positive for north, negative for south. Sexagesimal expression (in degree, minute, and second) are not acceptable.
13. *<altitude>*: Altitude value of the *<location>*, in meter (without unit), positive or possibly negative.
14. *<habitat>*: Living habitat of the population in the wild, e.g. *slopes*, *forest margins*, *streamsides*; or *cultivated* for those in garden or arboretum.
15. *<life form>*: Life form of typical individuals in the population, e.g. *tree*, *shrub*, *vine*.
16. *<height>*: Height of typical individuals in the population, in meter (without unit).

17. `<diameter at breast height>`: Diameter at breast height (DBH) of typical individuals in the population, in centimeter (without unit), only applying to trees or large shrubs.
18. `<note>`: Other valuable information that is no longer observable on herbarium specimens, in aspects of morphology (e.g. color & smell of different parts, texture of the bark), ecology (e.g. richness, pollinator species), or ethnobotany (e.g. local usages). Different from other parameters of `\collect`, `<note>` is a complete sentence (unless empty), so that the leading letter of the first word should be capitalized, and a punctuation (usually period) is required at the end.

By default, when preparing a collecting label, `nanicolle` typesets a map below the main body of the label, illustrating the position of the coordinates, given that the `<longitude>` lies between 73°E and 136°E, and the `<latitude>` lies between 17°N and 54°N. One can load a `nomap` `<option>` into the document class (Section 1) to suppress the typesetting of maps. It is also possible to redefine the geographic range of the maps.

### 3 The macro `\identify` for identifying labels

The syntax of the macro `\identify` is

```
\identify → <record number> → <scientific name> → <vernacular name>  

→ <identifier> → <identifier code> → <identifying date> → <comment>
```

Just as `\collect`, each `\identify` macro with its parameters must exclusively occupy a single line. Parameters can be left empty unless specialized, but the tabs separating them cannot be omitted. The requirements of each parameter of `\identify` are listed as follows.

1. `<record number>`: Only for the convenience of data organizing, not printed on identifying label.
2. `<scientific name>`: Scientific name with author citation of the species that the identification yielded, following the formula `<generic part>``<specific part>``<infraspecific part>``<author citation>`. As that of `\collect`, macros for manually manipulating font style are unavailable here either. `<scientific name>` cannot be empty.
3. `<vernacular name>`: Common name associated with `<scientific name>` of the species.
4. `<identifier>`: Full name of the identifier.
5. `<identifier code>`: Standard form (taxonomic) of the name of the identifier. `<identifier>` and `<identifier code>` cannot be both empty, while it is suggested to leave either of them empty in a record.
6. `<identifying date>`: Date when the specimen was identified, following the same restriction as `<collecting date>` for `\collect`. `<identifying date>` cannot be empty.
7. `<comment>`: Comment about the identification worth mention. Different from other parameters of `\identify`, `<comment>` is a complete sentence (unless empty), so that the leading letter of the first word should be capitalized, and a punctuation (usually period) is required at the end.

The `\identify` macro has no `<duplication count>` parameter, so identifying label will not be automatically repeated. When repeating is wanted, one has to repeat the lines with `\identify` manually.

## 4 Other issues

### 4.1 Store original data in a spreadsheet software

Using tabs `→` as delimiters between parameters is not the convention of  $\text{\LaTeX}$ . This special rule for delimiters was designed to make `nanicolle` able to read the plain text lines from a spreadsheet software<sup>3</sup>. When some rows of cells are pasted from a spreadsheet software to plain text environment, it is automatically converted to TSV (Tab-Separated Values) format — rows/lines are separated by end-of-line character(s), and cells within a row/line are separated by tab. This mechanism allows users to establish a database for collecting or identifying records in a spreadsheet software (as Table 1). When one wants to print labels according to certain records, he/she can simply copy the involved rows from the database, paste them in a  $\text{\LaTeX}$  source file, and then call `nanicolle` to deal with them. Databases can also contain extra information after the parameters required, which will be ignored by `\collect` or `\identify` and will not affect the output.

macro	record number	collectors	collecting number	...
<code>\collect</code>	1	Foo, Bar	3141	...
<code>\collect</code>	2	Foo, Bar	3142	...
<code>\collect</code>	3	Foo, Bar	3143	...

Table 1: A sample of database for collecting records in a spreadsheet database

### 4.2 Set the printer correctly

Before the PDF file from `nanicolle` is sent to a printer, it is necessary to do some settings. When printing an A4-sized PDF file on to an A4-sized paper with a home printer, the file is usually shrunk a little bit smaller to fit into the printable range. If so, since `nanicolle` uses a four-column landscape layout, the outer columns would be some broader than the inner ones. To avoid unbalance, one can select to print “at actual size”, “at absolute size”, or make the scale “100%”.

## 5 Change history

`nanicolle` was originally designed for making Chinese collecting labels and identifying labels, with its first version completed on 2016/8/3 (ver. 1.01). Later on 2017/10/22 (ver. 1.07), the typesetting of western style collecting labels was carried out for an international plant expedition, and that was the first version with maps. The macro for collection labels in western style had been temporarily hidden since 2019.4.28 (ver. 2.00), until rewritten and republished on 2020.7.8 (ver. 2.02). For more details, please refer to the Chinese manual `nanicolle-zh.PDF`.

## 6 A full example using `nanicolle`

The following is a full example file using document class `nanicolle`. It can be found as `nanicolle-ex-en.tex` in the package. To display it more clearly, the `\hspace` denotation for space is no longer used in this example. Actual lines correspond with line numbers in the left. An actual line may be so long that it is wrapped here, just as in text editors, but remember, these wrapped parts in fact belong to a single line, as there is no end-of-line character in between.

---

<sup>3</sup>Microsoft Office Excel is an instance of spreadsheet software.

```

1 \documentclass[autoduplicate]{nanicolle}
2 \begin{document}
3 \collect→1997→Yuchang Yang (杨宇昌)→5731→2018.5.8
   →Caprifoliaceae→苦糖果→Lonicera fragrantissima subsp.
   standishii→7609→1→between Dongjiamen Village (董家门村)
   and Dongnao (洞恼), Guantao Town (管陶乡), Wu'an City
   (武安市), Hebei Province (河北省), China→113.759512
   →36.951612→1356.0→meadow thickets on slopes→shrub→3→
   →Ripe fruit orange-red, tasting sweet with minimal bitter.
4 \collect→1545→Sino-Nepal Joint Plant Expedition (Haining Qin,
   Prabin Bhandari, Tirtha Raj Pandey, Bijay Raj Subedee,
   Yuchang Yang, Shuren Zhang)→601→2017.9.18→Fagaceae→
   →Quercus glauca→-→2→Talkot, Bajhang District, Nepal→
   →→1700→forests→tree→10→15
   →Fruiting. Associated with \textit{Rhododendron arboreum}
   and \textit{Lyonia ovalifolia}.
5 \identify→392→Allium atrosanguineum var. tibeticum (Regel) G.
   H. Zhu & Turland→藏葱 (Z\`ang C\`ong)→Yuchang Yang→
   →2018.10.7→
6 \identify→176→Acer davidii subsp. grosseri (Pax) P. C. de
   Jong→→Yuchang Yang→→2018.4.19→
7 \identify→230→Erysimum ×cheiri (L.) Crantz→→Yuchang Yang
   →→2018.5.17→
8 \identify→590→Koenigia alpina (All.) T. M. Schust. & Reveal→
   →Yuchang Yang→→2019.4.13→
9 \end{document}

```

Locate to the path of the example file in a command line window, then type and execute `xelatex nanicolle-ex-en`. After the compilation, the output PDF file (see Figure 1) `nanicolle-ex-en.PDF` can be found in the same path.

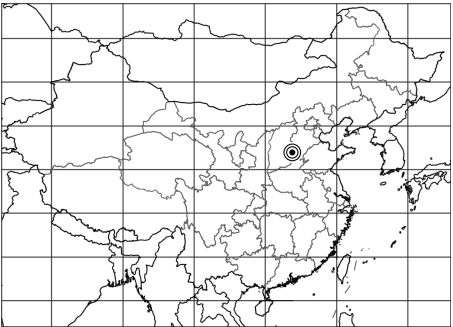
<p>CAPRIFOLIACEAE</p> <p><i>Lonicera fragrantissima</i> subsp. <i>standishii</i></p> <p>Yuchang Yang (杨宇昌) # 5731</p> <p>2018.5.8; between Dongjiamen Village (董家门村) and Dongnao (洞埡), Guantao Town (管陶乡), Wu'an City (武安市), Hebei Province (河北省), China, 113.759512°E, 36.951612°N, 1356.0 m a.s.l.; meadow thickets on slopes.</p> <p>Shrub, 3 m tall. Ripe fruit orange-red, tasting sweet with minimal bitter.</p>	<p>FAGACEAE</p> <p><i>Quercus glauca</i></p> <p>Sino-Nepal Joint Plant Expedition (Haining Qin, Prabin Bhandari, Tirtha Raj Pandey, Bijay Raj Subedee, Yuchang Yang, Shuren Zhang) # 601</p> <p>2017.9.18; Talkot, Bajhang District, Nepal, 1700 m a.s.l.; forests.</p> <p>Tree, 10 m tall, DBH 15 cm. Fruiting. Associated with <i>Rhododendron arboreum</i> and <i>Lyonia ovalifolia</i>.</p>
	<p><b><i>Allium atosanguineum</i> var. <i>tibeticum</i> (Regel) G. H. Zhu et Turland</b> 藏葱 (Zàng Cōng)</p> <p>Yuchang Yang 2018.10.7</p>
<p>FAGACEAE</p> <p><i>Quercus glauca</i></p> <p>Sino-Nepal Joint Plant Expedition (Haining Qin, Prabin Bhandari, Tirtha Raj Pandey, Bijay Raj Subedee, Yuchang Yang, Shuren Zhang) # 601</p> <p>2017.9.18; Talkot, Bajhang District, Nepal, 1700 m a.s.l.; forests.</p> <p>Tree, 10 m tall, DBH 15 cm. Fruiting. Associated with <i>Rhododendron arboreum</i> and <i>Lyonia ovalifolia</i>.</p>	<p><b><i>Acer davidii</i> subsp. <i>grosseri</i> (Pax) P. C. de Jong</b></p> <p>Yuchang Yang 2018.4.19</p>
	<p><b><i>Erysimum</i> × <i>cheiri</i> (L.) Crantz</b></p> <p>Yuchang Yang 2018.5.17</p>
	<p><b><i>Koenigia alpina</i> (All.) T. M. Schust. et Reveal</b></p> <p>Yuchang Yang 2019.4.13</p>

Figure 1: Left two columns in the sample PDF file nanicolle-ex-en.pdf