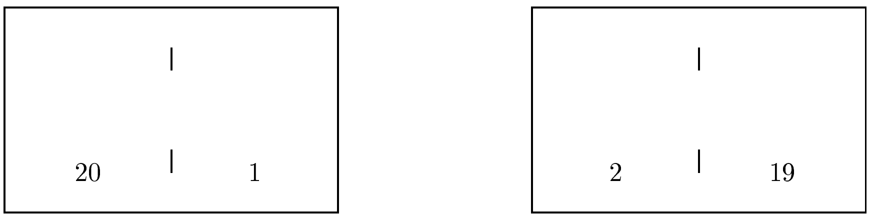
quire.tex

Macros for making booklets, printing double pages, and printing outlines and crop marks.
Version: 1.0 (19 January 1990) (doc: 10 Sep 1991)

A quire is a "set of folded sheets (as of a book) fitting one within another" [Webster's]. Such a set of sheets forms a little booklet. Larger books usually consist of a number of quires, with 20 to 40 pages each. The macros in quire.tex help you to turn your own texts into such booklets. (The macros can also be used for some other things, such as printing crop marks.) Quire.tex is designed to minimize the interference with other macro packages, so that, for instance, it can be used with plain TeX as well as with LaTeX. You do not need any special programs or special equipment to print quires, but it is convenient if you can print pages in "landscape mode." Whether you can print in landscape mode depends on your device driver and printer (most printers support this), it is outside the control of TeX. The reason for using landscape mode is that, because sheets are folded, two pages have to be printed next to each other on the same sheet. In "portrait mode" this allows only very narrow pages. Making quires is easier if you have a large version of TeX (i.e., one with lots of memory).

Before we explain what the macros do, let us emphasize what they do not do. Quire.tex does *not* change the layout of your pages, look at page numbers, print odd and even pages differently, or print small pages. These are all things that you have to take care of with other macros or style files. What quire.tex does is to take your pages, reorder them, print two per sheet, and draw outlines around them. As an example, assume we want to make a booklet consisting of five sheets folded together. With two pages next to each other, and with double sided printing, this gives us a booklet consisting of 20 pages. The first page of the five folded sheets is page 1. When we open the booklet we get page 2, printed on the back of the same sheet as page 1. Page 2 occupies the left-hand side of that sheet. The page on the right-hand side is not page 3, which is printed on the next sheet, but page 19. This reordering of pages is the main task of quire.tex. For this example, the following figure depicts two of the sheets generated by quire.tex (these are the last two sheets that are generated).



Once again, quire.tex does not look at the page numbers you print. What we call here "page 1" is simply what the first page of output would be if you were not using quire.tex. To make a booklet, the two double pages in the figure should be printed on the opposite sides of the same sheet of paper. There are two ways of doing that. One is to print all the sheets and then use a photocopier to make double sided prints. The second is to print all even sheets, walk to your printer and reinsert the stack of printed sheets upside down, then print all the odd sheets. Many device drivers can be instructed to print only even or odd sheets. But in case your installation does not support this, you can say `\shonly1` at the beginning of your file to instruct quire.tex to only generate odd sheets; `\shonly2` generates only even sheets; and `\shonly0`, which is the default, generates both.

Before you can use any of the macros, you have to `\input quire`*. Then you give values to some of the parameters (which are discussed later), and you can call, for instance, `\quire`. All of this should be done preceding the actual text of your file; in LaTeX all of this should be put in the preamble. If you are

* Although quire.tex can be used with LaTeX, we have made no attempts to LaTeXify the macros. So yes, in LaTeX you use the macros exactly as described here.

using LaTeX, you must say `\latexquire` somewhere in the preamble, regardless of what other macros you are using. The macro `\quire` is used to start typesetting of a quire. It takes one argument, the number of pages in the quire. Hence you have to TeX your document at least once before you use `\quire`, to determine the number of pages. That also helps to separate errors in your input from problems caused by quire.tex. Say your document consists of 19 pages, numbered 1…19. When you say

```
\quire{19}
```

TeX will produce 10 sheets. The first one is sheet 10 with pages 10 and 11, the second is sheet 9 with pages 12 and 9, etc. The last sheet has pages 20 and 1, where page 20 is blank.

This example illustrates that if there are not enough pages for a complete quire, TeX will insert empty pages at the end of your text. If, on the other hand, there are more pages than fit in a quire, TeX will just generate more than one quire. For instance, a 250-page book can be printed in 13 quires with `\quire{20}`. The mechanism through which TeX supplies extra blank pages at the end of your text is called TeX's "endgame." These extra pages are passed to TeX's output routine, which usually means that they will have headlines and footlines. You can bypass the output routine by saying `\endquire` at the end of your text, in which case the blank pages will really be blank pages (you can say `\endquire` even if you are not using `\quire`). In LaTeX you should *not* use `\endquire`, it is implied by the `\end{document}` that terminates your file. Sometimes you may want to start your text with an empty page. Here are two simple ways of generating an empty starting page (in plain TeX).

```
\line{}\vfill\eject % or:
\shipout\vbox{}
```

The method with `\shipout` bypasses the output routine.

Let us turn to the parameters that determine the output of quire.tex. You have to take care that you indeed generate small pages. The page size is normally determined by `\hsize` and `\vsize`, but depending on the macro package you are using, other parameters may also be important. However, quire.tex does not look at the values of these parameters, they only serve to determine the size of the text area (i.e., the printed part of the page). In the following, "page" refers to the small pages you print, "sheet" refers to the actual sheets of paper. We start with the parameters related to pages, which are all dimen parameters. First there are `\htotal` and `\vtotal`, which should be equal to the horizontal and vertical size of the small pages you want. For instance, a pocket book has `\htotal` ≈ 10.5 cm, `\vtotal` ≈ 17 cm. Then there are `\horigin` and `\vorigin` which describe the "natural" offset of the text area from the top left corner of the page. These offsets are added to the TeX offsets `\hoffset` and `\voffset`. You may wonder why there are four parameters to describe the offsets, TeX itself only has the latter two. The reason is that, unfortunately, TeX has built in extra offsets of 1 inch. That may be right for American "letter" size paper, but on smaller pages it is generally too much. To make scaling of these offsets possible we had to introduce two extra parameters for them.

The names of parameters related to sheets all start with "`sh`"; all but one are dimen parameters. Quire.tex draws a number of lines to denote where the actual pages are on the sheet. As in the previous figure, lines are drawn around pairs of pages. The width is `\shoutline`, so you can make them disappear with `\shoutline=0pt`. The two pages are separated by two "staples," with width `\shstaplewidth` and length `\shstaplelength`. You can make them disappear by making the width `0pt`; setting the length to `0.5\vtotal` will give you a solid line instead of two staples. The outlines are drawn exactly around the pages, the staples on top of the two pages. The widths of these lines does not influence the width of the pages. Finally, quire.tex draws "crop marks" of width `\shcrop`. Crop marks are lines that indicate the outlines and center of the pages, without actually touching the pages. When you are cutting the paper, crop marks are sometimes more convenient than outlines. The crop marks are drawn so that, if they were extended, they would just be on the page. So they fall within the outlines that are drawn, which is easily observable if you make the widths excessive, say 1 cm.

The distance of the top of the sheet to the top of the pages is `\shvoffset` plus 1 inch. The pages are centered horizontally, therefore there is no parameter for horizontal offset. Under the pages, a footline is printed. The footline is determined by token register `\shfootline`. By default the sheet number and the numbers of the two pages are printed (more about page numbers later), as well as the filename. The next paragraph explains the remaining four parameters, but if you are lucky you don't have to set them yourself:

They describe the printer you are using, so they need only be determined once. Your local installation may have macros that set these parameters automatically. Local macros are described at the end of this manual section.

In order to center the pages, `\shhtotal` must be set to the horizontal size of the sheets. The size is usually not a nice number, like 8.5 inch (216 mm), but more likely something like 220.1 mm. It is important to get this size right, since double sided printing requires properly centered pages. You probably have to experiment a bit with double sided pages to get this right, because often the printer introduces small inaccuracies. If a photocopier is used to turn single sided print into double sided print, you may need the parameters `\shhcorrection` and `\shvcorrection`. These denote extra offsets which are applied to odd numbered sheets only, to compensate for shifts introduced by "smart" copiers. `\shhcorrection` can sometimes be used to compensate for a wrong value of `\shhtotal`, but the author recommends that `\shhtotal` is set properly for the printer, and that the correction offsets are only used for copiers. If you take a stack of sheets and fold them, the outer sheets will have a longer way to go around the bend than the inner ones. In an attempt to compensate for that, the pages on each sheet are shifted outwards with respect to the pages on the next inner sheet, by an amount equal to `\shthickness`. Naturally, this parameter should have a rather small value.

Consider printing a quire of 20 pages. The first sheet that can be printed is sheet number 10 with pages 10 and 11. That means that TeX has to store 10 complete pages in memory. If your TeX version is small, there may not be enough memory for this, resulting in an error message

```
TeX capacity exceeded, sorry.
```

Luckily, there are some ways to reduce the number of pages that have to be stored. We already mentioned the use of `\shonly` to restrict output to odd or even sheets only. Doing that halves the number of pages that have to be stored. If that is not enough, you can use the macro `\makequire` instead of `\quire`. `\makequire` takes three arguments. The first argument is the same as for `\quire`, namely the number of pages per quire. Let s and r be the second and third arguments, respectively. Let n be the number of a sheet in the first quire. Then that sheet is only printed if $s \leq n < s + r$ (and similarly for following quires). This means that at most r pages need to be stored (or about half that if you also use `\shonly`). As an example, our 20 page quire can be printed in three runs with, respectively,

```
\makequire{20}{1}{4}
\makequire{20}{5}{4}
\makequire{20}{9}{4}
```

The runs result in sheets 4...1, 8...5, and 10...9. In the last run, the range r is larger than the actual number of sheets, but that causes no problems. Also, sheet s need not exist. In fact, `\quire{20}` is just an abbreviation for `\makequire{20}{0}{1000}`.

For previewing it is often inconvenient to use `\quire`, because of the reordering of pages. But you may still want to see how your text fits on the smaller pages. There are two macros which you can use instead of `\quire`. `\qtwopages` prints two pages per sheet, just like quire, but does not reorder the pages. All parameters mentioned before remain valid, except for `\shthickness`. Of course, you can also use this if you are not previewing a quire. For instance, if you have appropriately scaled fonts and you print in landscape mode, the following preamble will print your file two pages per sheet.

```
\input quire
\mag=500 % works in LaTeX too
\qtwopages
```

Finally, there is `\qonepage`, which prints just one page per sheet as usual. It does however draw outlines (no staples) and crop marks, and `\shonly` can also be used.

We claimed that quire.tex would work together with most other macro packages. But there is one type of macro that doesn't go well with quires, and usually you will have to switch them off when you are typesetting a quire. TeX's primitives for file handling, such as `\write`, are only performed when a page is actually shipped out. That is often used for the creation of an index or a contents page, in order to get the page numbers right. But quire.tex prints two pages per sheet, and a sheet has only one "page number," so that such methods will most likely get all the page numbers wrong. Worse, the pages are shipped out in an unusual order, so that for instance a `\write` may take place before the file has been opened. Therefore, you should avoid generation of an index or contents page when you are using `\quire`. In LaTeX this is done by inserting `\nofiles` in the preamble. (For all but the final run you can use `\qonepage`, which does not change the page numbers.) Of course, if you want your booklet to start with a contents page, then that page has to be generated before the final TeX run, since you cannot just rearrange the printed pages.

In `\shfootline`, `\pageno` refers to the sheet number (if you are using plain TeX's definitions); `\firstpageno` is the number of the first of the two pages (i.e., the page that was finished first, that can be either the left-hand or the right-hand page), and `\secondpageno` is the number of the other page. These three numbers are also shipped to the `dvi` file, as `\count0`, `\count1`, and `\count2`, respectively. When a page is finished but not yet printed on a sheet, the page number is displayed on your terminal between parentheses, as in `[(1)]`. When sheet number 7 with pages 11 and 14 is finished, `[7.11.14]` is displayed.

Quire.tex does its job by redefining `\shipout`. Since `\shipout` is a TeX primitive, such redefinition is not recommended; normally one would redefine the output routine instead. However, important parts of the typesetting of a text are often done in the output routine, which may well be different for different pages of a book. Since quire.tex is supposed to handle the whole book, and since it only handles complete pages without changing their layout, it seemed that in this particular case redefinition of `\shipout` was justified. The new `\shipout` can be used in the same way as the original one, except that you should not try to ship out a void box (which is a pretty meaningless thing to do anyway). Doing that will most likely result in a, seemingly unrelated, error message somewhere later in the text. The dangers of using `\openout`, `\closeout` and `\write`, without `\immediate`, were already explained. Like these file operations, `\special` works with “whatsits” which are handled when the page is shipped out. But as the argument of `\special` is expanded immediately, rather than when the page is shipped out, this is less likely to cause problems.

Quire.tex uses mostly TeX's primitives, but also a few macros defined in plain.tex. The latter are `\newbox`, `\newcount`, `\newdimen`, `\newif`, `\newtoks`, `\nointerlineskip`, and `\offinterlineskip`, none of which are likely to be redefined by other macros. As mentioned before, quire.tex relies on TeX's endgame to generate complete quires. Although the endgame is built in, a really devious macro package may disable it. This is in fact exactly what happens in LaTeX's `\end` command, which is why we need `\latexquire` (which inserts an implicit `\endquire` at the end of the text).

1. Printer definitions. It is convenient if the parameters that depend (only) on the printer are predefined. As an example, here are some of the local definitions for the author's installation. When someone wants to use the printer called “maser,” all he or she has to do to set up those parameters is to say `\Pmaser` or `\Pmaserlandscape`. Note that this does not put any information in the `dvi` file, the user is still responsible for printing on the correct output device. There are similar definitions for other available printers. The next subsection lists the local definitions for your installation, if any.

```
\def\Pmaser
   {\shhtotal=220.02true mm
    \shthickness=0.21true mm
    \message{maser: flip paper so that short sides remain in place, %
    align closest right-hand corner in tray.}%
   }

\def\Pmaserlandscape
   {\shhtotal=281true mm
    \shthickness=0.21true mm
    \message{maser landscape: ...}
   }
```

2. Local Caltech definitions. The following macros will set the appropriate printer parameters: `\Pmaser`, `\Pmaserlandscape`, `\Plaser`, `\Plaserlandscape`, `\Phaser`, `\Phaserlandscape`, `\canon` (for the copier). For double-sided printing on maser, flip the pages so that the short edges remain in place. Align the closest right-hand corner in the tray. But for printing in landscape mode, flip the pages so that the long edges remain in place. On haser, you should rotate the pages 180°, but keep the text side up. You have to reorder the pages (or print the second time with `gtex -t`). Haser seems to have problems keeping the paper straight, which makes double-sided printing hard.