Hans Hagen & Ton Otten (with thanks to Karel H Wesseling for the translation)



#### abstract

Within the T<sub>E</sub>X community there is a widely used database for bibliographic references, BIBTEX, but not for figures. To manage figures ConT<sub>E</sub>Xt now supports a figures database. The database is setup in XML and converted to an interactive PDF figure library featuring ordered displays and a search mechanism. From the library, figures can be included easily in ConT<sub>E</sub>Xt documents as long as both the PDF and the XML files remain present.

### Structure

. . .

Figures are described and maintained in an XML type file. Each figure is encapsulated in a figure element and the assembly of all figure elements is encapsulated in the main database element figurelibrary:

```
<figurelibrary language="en">
```

```
</figurelibrary>
```

The figurelibrary element has a language attribute to tell ConTEXt which language is used in the included texts. Only one language can be specified for each figure database. The default language is english and need not be specified.

The database has comment fields to describe shared characteristics of the figures. They are encapsulated in a description element. Several sub-elements or fields in the element are supported and each field can be omitted, left empty or filled with many lines of text. The three dots ... signify this possibility.

```
<description>
<organization>...</organization>
<project>...</project>
<product>...</product>
<comment>...</comment>
</description>
```

Each figure, as described in the figure element, has four fields of which two (file and label) are mandatory and must be completed, and two are optionally filled:

```
<figure>
<file>filename.png</file>
<label>symbolic name</label>
</figure>
```

If a figure is available in several formats, such as PNG and JPG, and you don't supply the file's picture format in the file extension, ConTEXt locates the most appropriate one automatically, given that the pdfTEX compiler can handle PDF, PNG and JPG formats and METAPOST. To keep the database general purpose we advise you, however, to supply the file extension together with the file name as shown.

If the figure is not yet available, a placeholder can be substituted for the future file name by replacing the file by a dummy sub-element:

```
<figure>
<dummy width="4cm" height="3cm">to be made</dummy>
<label>mandatory symbolic name</label>
</figure>
```

Note the positions of the <> brackets. The label element associates a symbolic name with the figure, to be used later to recall the figure from the database.

## Usage

The figure library is a PDF file generated from the XML specification file, say myfigs.xml, by processing it by  $T_EXexec$  (a PERL script in the Con $T_EXt$  distribution available from Pragma ADE):

```
texexec --pdf --use=fig-make myfigs.xml
```

The first option (-pdf) tells  $T_EXexec$  to generate a PDF file. The second option (-use) invokes the style file fig-make<sup>1</sup> that comes with the Con $T_EXt$  distribution.  $T_EXexec$  invokes Con $T_EXt$  which calls PDFTEX with this style file.

The resulting PDF file receives the name myfigs.pdf. It is an interactive figure library for a PDF viewer such as ACROBAT or GHOSTVIEW. Up front in the library file are the individual figures shown at their natural size. At the rear in the file is a summary with the descriptions, a table of contents and an index. Figure I below shows the figures of the example given later.



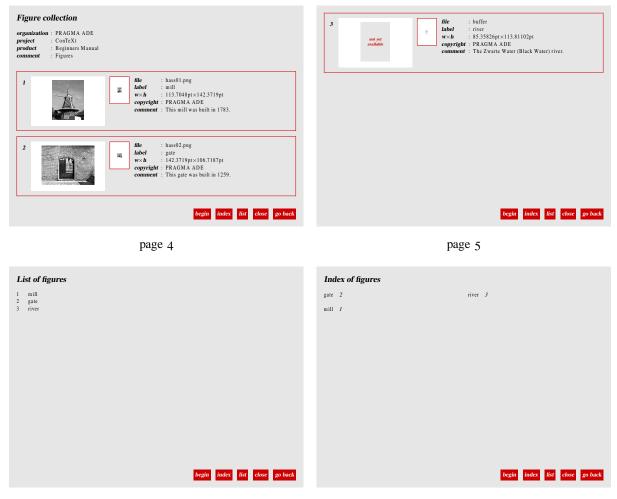
page I page 2 page 3 Figure 1 The first three pages of the figure library file.

Figure 2 shows the summary. In the summary the figures are shown first at a reduced height, second at their size with respect to A4 paper size, third with their descriptions. At screen page bottom is a navigation menu to reach the table of contents or the alphabetically ordered index. Click on a figure to switch between full size and reduced size. Leave the viewer by clicking on close.

Instead of showing the figure relative to A4 paper size other sizes can be chosen with the TEXexec -mode command line option:

texexec --pdf --use=fig-make --mode=letter

I. fig-make is a synonym for x-fig-01



## page 6

page 7

Figure 2 The overview, table of contents and index of the figures that are present in the file.

The choices for -mode are A4, letter and compact, of which A4 is default.

| mode    | meaning                         |
|---------|---------------------------------|
| letter  | map the preview on letter size  |
| compact | use an alternative presentation |

## An XML example

The XML database specification file for the example shown above in figures 1 and 2 is:

```
<figurelibrary language="en">
```

```
<description>
<organization>PRAGMA ADE</organization>
```

MAPS

### Figures

```
<project>ConTeXt</project>
    <product>Beginners Manual</product>
    <comment>Figures</comment>
  </description>
 <figure>
   <file>hass01.png</file>
    <label>mill</label>
    <copyright>PRAGMA ADE</copyright>
    <comment>This mill was built in 1783</comment>
  </figure>
  <figure>
    <file>hass02.png</file>
    <label>gate</label>
    <copyright>PRAGMA ADE</copyright>
    <comment>This gate was built in 1259</comment>
  </figure>
 <figure>
    <file>buffer</file>
   <label>river</label>
    <copyright>PRAGMA ADE</copyright>
    <comment>The Zwarte Water (Black Water) river</comment>
  </figure>
</figurelibrary>
```

# A T<sub>E</sub>X example

To apply the database to a ConT<sub>E</sub>Xt document the figure library processing module must be specified:

\usemodule[fig-base]

Next specify which database to use:

```
\usefigurebase[myfigs]
```

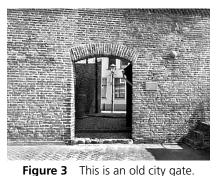
From this point, having both the XML and the PDF files available, you may place figures in your document file to be extracted from the (one or more) PDF libraries just specified:

```
\placefigure
{A very old mill.}
{\externalfigure[mill][width=4cm]}
```

Another example using more options of the \placefigure command is:

```
\placefigure
 [left] [fig:the gate] {This is an old city gate.}
 {\externalfigure[gate][height=4cm]}
```

Voorjaar 2002



the result of which is shown in figure 3. A figure may also be included by its page number in the figure database:

```
\placefigure
{This is an old city gate.}
{\externalfigure[myfigs][page=2,
width=.3\textwidth]}
```

This method is not recommended for a final document since updating the myfigs figure database may change the page number of a figure thus resulting in a document with errors.

Finally, it continues to be possible to load a figure directly from disk:

\externalfigure[myfigure][width=4cm]

or

```
\useexternalfigure[my not too large figure][myfigure][width=4cm]
\externalfigure[my not too large figure]
```

When the figure base module is loaded, ConTEXt will first try to locate myfigs in the XML database. This file, therefore, must be kept and not deleted after construction of the PDF file. When the figure cannot be found there, ConTEXt will search for a file with the name myfigure which may include a file name extension.

Apart from the already mentioned  $T_EXexec \mod option$  there are no other options that can be used.

Non-ConTEXt users can access the database by means of an additional file. For this, you need to run TEXexec once more:

```
texexec --pdf --use=fig-fake yourfile.xml
```

The resulting file, yourfile.fig, can be loaded in your document in the normal way. The figures can then be accessed with:

\getfigurefile{label}
\getfigurepage{label}

Given that you have also loaded supp-pdf.tex, you can now include the figure file with:

\includefigurefile width 10cm height 4cm {label}

Don't use this method in ConTEXt.

# Documentation

ConT<sub>E</sub>Xt has many more options to include graphics in your document, including scaling, clipping, positioning, subtitling, grouping and general manipulating. These techniques and more a found in the ConT<sub>E</sub>Xt and MetaFun manuals available from our web site: www.pragma-ade.com.