

This Way

ConTEXt magazine #1101 MKIV
July 2011

Project Structure
Hans Hagen
PRAGMA ADE

A regular document has a simple structure. When we talk about structure here, we only refer to the overall document structure.

```
% style specification

\starttext
  % the document content
\stoptext
```

For practical reasons we delay initial font loading till the first `\starttext` so that one can overload the defaults. This means that when no `bodyfont` is specified, and is not given, there will be hardly any visible output.

An example of a more elaborate structure is the following:

```
\environment environment-1
\environment environment-2

\startproduct product-1

  \component component-1.tex
  \component component-2.mkiv
  \component component-3.cld

  \component component-1
  \component component-2

\stopproduct
```

Here we have a specific product, made up out of components and using a few environment files that specify the style. By default we assume `tex` files, but you can be specific and use known suffixes. A less abstract example is the following:

```
\environment my-fonts
\environment my-style
\environment my-abbreviations
\environment my-urls

\startproduct manual

  \component titlepage
  \component contents

  \component chapter-1
  \component chapter-2
  \component chapter-3

  \component index

\stopproduct
```

You can process components and products independently but be aware that you won't get cross document (or chapter) references then.

There is one more level: projects.

```
\environment my-fonts
\environment my-style
\environment my-abbreviations
\environment my-urls

\startproject documentation

  \product manual
  \product faqs

\stopproject
```

This means that we can also define the manual as follows:

```
\project documentation

\startproduct manual

  \component titlepage
  \component contents

  \component chapter-1
  \component chapter-2
  \component chapter-3

  \component index

\stopproduct
```

Environments are only loaded once and when you run a component or product that refers to environments or when environments are picked up from an encapsulating structure you need to be aware of the order of loading.

The names given after the start command are not that important but the names after the simple commands refer to filenames, so in the next case there need to be a file called `index.tex`:

```
\component index
```

Equally valid is:

```
\component [index]
```

Subpaths are also permitted:

```
\component manual/index
```

The meaning of the mentioned commands is not frozen but adapts itself to the current situation. A file can be processed many times, only once or never. The following table shows what will happen when:

	<code>\component</code>	<code>\environment</code>	<code>\product</code>	<code>\project</code>
<code>\start .. \stopcomponent</code>	many	once	none	once
<code>\start .. \stopenvironment</code>	none	once	none	none
<code>\start .. \stopproduct</code>	many	once	many	once
<code>\start .. \stopproject</code>	none	once	none	none
<code>\start .. \stoptext</code>	many	once	many	once

When you load an environment or component, you can specify it to be a LUA file by using the `lua` or `cld` suffix. In that case the file will be loaded in the right way. From the table you can deduce that the following is also valid:

```
\environment mystyle
```

```
\starttext
% the content
\stoptext
```

combined with:

```
\startenvironment mystyle
% the definitions
\stopenvironment
```

This is about the simplest structure that you can use that still gives a bit of abstraction.

In addition to files in a project structure, you can load predefined modules.

```
\usemodule [mathml]
```

or more specific:

```
\usemodule [x] [mathml]
```

Which limits the lookup to the `x` namespace. The first match quits the search and the order of lookups is: `mkvi`, `mkiv`, `tex`, `cld`, `lua`. It follows that modules can be LUA files.

When you use structure in the files you will find an overview in the log file. This looks as follows:

```
system          > structure > start used structure

used structure  > text: product-1
used structure  > environment: environment-1
used structure  > environment: environment-2
used structure  > product: product-1
```

```

used structure > component: component-1
used structure > component: component-2
used structure > component: component-1
used structure > component: component-2

system > structure > stop used structure

```

Some basic logging on the console can be enabled with:

```
\enabletrackers[system.jobfiles]
```

A new command pair is the following:

```

\startdocument[settings]
  structured content
\stopdocument

```

The settings are key/value pairs and the values can be retrieved using:

```
\documentvariable{key}
```

You can set before and after parameters and by default these are set up as follows:

```

\setvariables
[document]
[before=\directsetup{document:start},
after=\directsetup{document:stop}]

```

You can for instance define these setups to generate a title page (using document variables) and a colophon page. In the future more functionality might be added.

source code of this document

```

% language=uk

% author      : Hans Hagen
% copyright   : PRAGMA ADE & ConTeXt Development Team
% license     : Creative Commons Attribution ShareAlike 4.0 International
% reference   : pragma-ade.nl | contextgarden.net | texlive (related) distributions
% origin      : the ConTeXt distribution
%
% comment     : Because this manual is distributed with TeX distributions it comes with a rather
%              liberal license. We try to adapt these documents to upgrades in the (sub)systems
%              that they describe. Using parts of the content otherwise can therefore conflict
%              with existing functionality and we cannot be held responsible for that. Many of
%              the manuals contain characteristic graphics and personal notes or examples that
%              make no sense when used out-of-context.

\usemodule[mag-01,abr-02,job-01]

\startbuffer[abstract]
  For a long time already \CONTEXT\ provides a way to organize your document(s)
  in a structure that permits processing of components. This mechanism has been
  upgraded a bit in \MKIV\ and here we will summarize the status quo.
\stopbuffer

\startdocument
  [title={Project Structure},
   author=Hans Hagen,
   affiliation=PRAGMA ADE,
   date=July 2011,
   number=1101 \MKIV]

A regular document has a simple structure. When we talk about structure here, we
only refer to the overall document structure.

\startscite[tex]
% style specification

\starttext
  % the document content
\stoptext
\stopscite

For practical reasons we delay initial font loading till the first \type
{\starttext} so that one can overload the defaults. This means that when no
bodyfont is specified, and {\starttext} is not given, there will be hardly any
visible output.

An example of a more elaborate structure is the following:

% \enabletrackers[context.trace]

\startscite[tex]
\environment environment-1
\environment environment-2

\startproduct product-1

  \component component-1.tex
  \component component-2.mkiv
  \component component-3.cld

  \component component-1

```

source code of this document

```
\component component-2
\stopproduct
\stopscite
```

Here we have a specific product, made up out of components and using a few environment files that specify the style. By default we assume tex files, but you can be specific and use known suffixes. A less abstract example is the following:

```
\startscite[tex]
\environment my-fonts
\environment my-style
\environment my-abbreviations
\environment my-urls

\startproduct manual

\component titlepage
\component contents

\component chapter-1
\component chapter-2
\component chapter-3

\component index

\stopproduct
\stopscite
```

You can process components and products independently but be aware that you won't get cross document (or chapter) references then.

There is one more level: projects.

```
\startscite[tex]
\environment my-fonts
\environment my-style
\environment my-abbreviations
\environment my-urls

\startproject documentation

\product manual
\product faqs

\stopproject
\stopscite
```

This means that we can also define the manual as follows:

```
\startscite[tex]
\project documentation

\startproduct manual

\component titlepage
\component contents

\component chapter-1
\component chapter-2
\component chapter-3

\component index
```

source code of this document

```
\stopproduct
\stopscite
```

Environments are only loaded once and when you run a component or product that refers to environments or when environments are picked up from an encapsulating structure you need to be aware of the order of loading.

The names given after the start command are not that important but the names after the simple commands refer to filenames, so in the next case there need to be a file called `\type {index.tex}`:

```
\startscite[tex]
\component index
\stopscite
```

Equally valid is:

```
\startscite[tex]
\component[index]
\stopscite
```

Subpaths are also permitted:

```
\startscite[tex]
\component manual/index
\stopscite
```

The meaning of the mentioned commands is not frozen but adapts itself to the current situation. A file can be processed many times, only once or never. The following table shows what will happen when:

```
\ctxlua{moduledata.jobs.showprocessors()}
```

When you load an environment or component, you can specify it to be a `\LUA\` file by using the `\type {lua}` or `\type {cld}` suffix. In that case the file will be loaded in the right way. From the table you can deduce that the following is also valid:

```
\startscite[tex]
\environment mystyle

\starttext
% the content
\stoptext
\stopscite
```

combined with:

```
\startscite[tex]
\startenvironment mystyle
% the definitions
\stopenvironment
\stopscite
```

This is about the simplest structure that you can use that still gives a bit of abstraction.

In addition to files in a project structure, you can load predefined modules.

```
\startscite[tex]
\usemodule[mathml]
\stopscite
```


source code of this document

or more specific:

```
\startscite[teX]
\usemodule[X][mathml]
\stopscite
```

Which limits the lookup to the `\type {x}` namespace. The first match quits the search and the order of lookups is: `\type {mkvi}`, `\type {mkiv}`, `\type {tex}`, `\type {cld}`, `\type {lua}`. It follows that modules can be `\LUA\` files.

When you use structure in the files you will find an overview in the log file. This looks as follows:

```
\starttyping
system          > structure > start used structure

used structure  > text: product-1
used structure  >  environment: environment-1
used structure  >  environment: environment-2
used structure  >  product: product-1
used structure  >    component: component-1
used structure  >    component: component-2
used structure  >    component: component-1
used structure  >    component: component-2

system          > structure > stop used structure
\stoptyping
```

Some basic logging on the console can be enabled with:

```
\startscite[teX]
\enabletrackers[system.jobfiles]
\stopscite
```

A new command pair is the following:

```
\starttyping
\startdocument[settings]
  structured content
\stopdocument
\stoptyping
```

The settings are key||value pairs and the values can be retrieved using:

```
\starttyping
\documentvariable{key}
\stoptyping
```

You can set `\type {before}` and `\type {after}` parameters and by default these are set up as follows:

```
\starttyping
\setvariables
  [document]
  [before=\directsetup{document:start},
  after=\directsetup{document:stop}]
\stoptyping
```

You can for instance define these setups to generate a title page (using document variables) and a colophon page. In the future more functionality might be added.

```
\stopdocument
```

the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million (19.5% of the population).

There are a number of reasons why the number of people aged 65 and over has increased. One of the main reasons is that people are living longer. The life expectancy at birth in the UK is now 77 years for men and 81 years for women. This is a significant increase from the 1950s, when life expectancy at birth was 71 years for men and 75 years for women. The increase in life expectancy is due to a number of factors, including improvements in diet, lifestyle, and medical care.

Another reason why the number of people aged 65 and over has increased is that people are having children later in life. This is due to a number of factors, including the fact that women are now having children at an older age, and the fact that men are now having children at an older age.

There are a number of implications of the increase in the number of people aged 65 and over. One of the main implications is that there will be a need for more social care services. This is because people aged 65 and over are more likely to need help with everyday tasks, such as shopping, cooking, and cleaning.

Another implication is that there will be a need for more housing for older people. This is because many older people live in overcrowded and unsuitable housing, and they need to move to more suitable accommodation.

There are a number of ways in which the government can address the implications of the increase in the number of people aged 65 and over. One way is to invest in social care services, so that older people can receive the help they need with everyday tasks.

Another way is to invest in housing for older people, so that they can move to more suitable accommodation. This could be done by building new housing for older people, or by providing grants to help older people to buy or rent more suitable housing.

There are a number of other ways in which the government can address the implications of the increase in the number of people aged 65 and over. For example, the government could invest in health care services, so that older people can stay healthy and independent for as long as possible.

The increase in the number of people aged 65 and over is a significant demographic change, and it has a number of implications for society. The government needs to take action to address these implications, so that older people can live well in the 21st century.

The following table shows the number of people aged 65 and over in the UK, by year, from 1990 to 2010. The number of people aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2010.

The following table shows the life expectancy at birth in the UK, by year, from 1950 to 2010. Life expectancy at birth has increased from 71 years for men and 75 years for women in 1950 to 77 years for men and 81 years for women in 2010.

The following table shows the number of children born in the UK, by year, from 1950 to 2010. The number of children born in the UK has decreased from 3.5 million in 1950 to 1.5 million in 2010.

The following table shows the number of people aged 65 and over in the UK, by year, from 1990 to 2010. The number of people aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2010.

The following table shows the life expectancy at birth in the UK, by year, from 1950 to 2010. Life expectancy at birth has increased from 71 years for men and 75 years for women in 1950 to 77 years for men and 81 years for women in 2010.

The following table shows the number of children born in the UK, by year, from 1950 to 2010. The number of children born in the UK has decreased from 3.5 million in 1950 to 1.5 million in 2010.

The following table shows the number of people aged 65 and over in the UK, by year, from 1990 to 2010. The number of people aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2010.

The following table shows the life expectancy at birth in the UK, by year, from 1950 to 2010. Life expectancy at birth has increased from 71 years for men and 75 years for women in 1950 to 77 years for men and 81 years for women in 2010.

The following table shows the number of children born in the UK, by year, from 1950 to 2010. The number of children born in the UK has decreased from 3.5 million in 1950 to 1.5 million in 2010.