\starttext

ConTeXt Meeting 2021
Wolfgang Schuster

Why do we need \starttext?

- Plain T_EX has only the **\end** command (or better **\bye** to flush remaining objects) to finish a document.

- CONT_EXT use the **\stoptext** command to finish a document but it requires also **\starttext** to mark the begin of a document.

- Why do we have to tell CONT_EXT where the document starts when plain T_EX has no need for this?

Minimal example without \starttext

```
\centerline{\bf A SHORT STORY}
\vskip 6pt
\centerline{\sl by A. U. Thor}
\vskip .5cm
Once upon a time, in a distant
  galaxy called Ööç,
there lived a computer
named R.~J. Drofnats.

Mr.~Drofnats—or "R. J.," as
he preferred to be called—
was happiest when he was at work
typesetting beautiful documents.

\bye
```

```
                    1

           A SHORT STORY

          by A. U. Thor
```

Minimal example with \starttext

```
\starttext

\centerline{\bf A SHORT STORY}
\vskip 6pt
\centerline{\sl by A. U. Thor}
\vskip .5cm
Once upon a time, in a distant
  galaxy called Ööç,
there lived a computer
named R.~J. Drofnats.

Mr.~Drofnats—or "R. J.," as
he preferred to be called—
was happiest when he was at work
typesetting beautiful documents.

\stoptext
```

Why are the examples different?

◇ Plain T_EX preloads its default fonts in the format file and when you process a file these fonts are used for the text.

◇ CONT_EXT doesn't postpones the loading of the default fonts until `\starttext` and since the command was never used no fonts are loaded.

◇ Only the `\hbox` command uses a fallback font in case no main font is used which leads to certain string being visible in the PDF.

How can this be fixed?

- ◇ One way to get the text back in the output is to load a font at the start of the document with **\setupbodyfont**.

- ◇ The better and correct way is to put **\starttext** at the beginning of the document which loads the default font in the case that no other font hasn't been loaded.

- ◇ There isn't even any other method besides adding **\starttext** because CONT$_E$XT loads and enables a few other setting besides the default font at this point and omitting **\starttext** can have side effects.

What happens when we put \starttext in \starttext?

- ◇ The **\starttext** environment can be nested as much as needed and CONTEXT keeps track of the current level.

- ◇ The **\stoptext** at the outermost level is always used to finish the document.

```
\starttext

    \starttext
     .  .  .
    \stoptext

\stoptext
```

What about \startTEXpage?

When you use **\startTEXpage** for a single page there is no need to use **\starttext** in the document even though the command is needed to produce text in the output.

```
\startTEXpage

 .  .  .

\stopTEXpage
```

There isn't even **\stoptext** in the document to finish it.

How \startTEXpage works

The reason why you can create a document which has only **\startTEXpage** as delimiters is that the command add **\starttext** at the begin of the environment and **\stoptext** at the end of it.

A simplified version of the definition for the **\startTEXpage** looks like this.

```
\def\startTEXpage
  {\starttext
      . . . }


\def\stopTEXpage
    { . . .
      \stoptext}
```

Using multiple \startTEXpage environments in a document

To use multiple **\startTEXpage** environment in a single document put **\starttext** at the begin of the document and CONT_EXT continues until it reaches **\stoptext** at the same level as the first **\starttext**.

```
\starttext

\startTEXpage
 . . .
\stopTEXpage

\stoptext
```

What is the difference between \starttext and \startdocument?

◇ Like **\startTEXpage** the **\startdocument** environment puts **\starttext** at the definition of the command.

◇ An additional feature are hooks to flush content (e.g. book cover or title page) at the begin and end of the document.

◇ Another feature is set metadata for the PDF to set the *title*, *author* and *keywords*.

◇ The **\startdocument** environment can be used in combinations with other mechanism like the project structure.

Organizing files in small and large documents

Dependant on the size of a document there are different ways how the content and style can be organized.
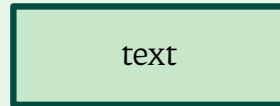
There are four different ways to structure ways to organize them:

1. Single file,

2. separate files for content and style,

3. product and component files,

4. project file with product and component files.

Standalone documents

For short documents it is often enough to put all settings at the begin of the document before **\starttext**.

The document consists only of a single file:

```
text
```

Separation of content of style

Once the document gets longer or more settings are needed it useful to separate the content and style of the document.

All settings are put in an environment file which is then loaded from the file with the content.

text	→	environment

Using product files

When a document gets even larger (e.g. book) it is useful to split the document into smaller files where each chapter resides in a separate file.

The *product* is used to put all files together in a single document and it contains all the strcuture blocks to separate the document into section blocks, place register or add covers pages.
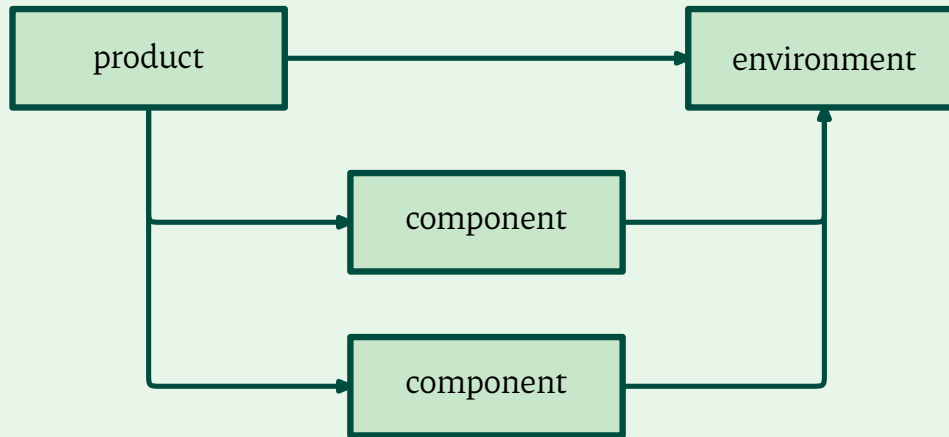
A feature of *components* is that each file is a document of its own and you can produce a PDF for each individual file.

Product structure

The *product* file loads all the *component* file and the *environment* file.

When only a single *component* is processed the component file loads the *environment* file.

Using project file

In some cases loading the *environment* file from the *product* is no longer the preferred method.
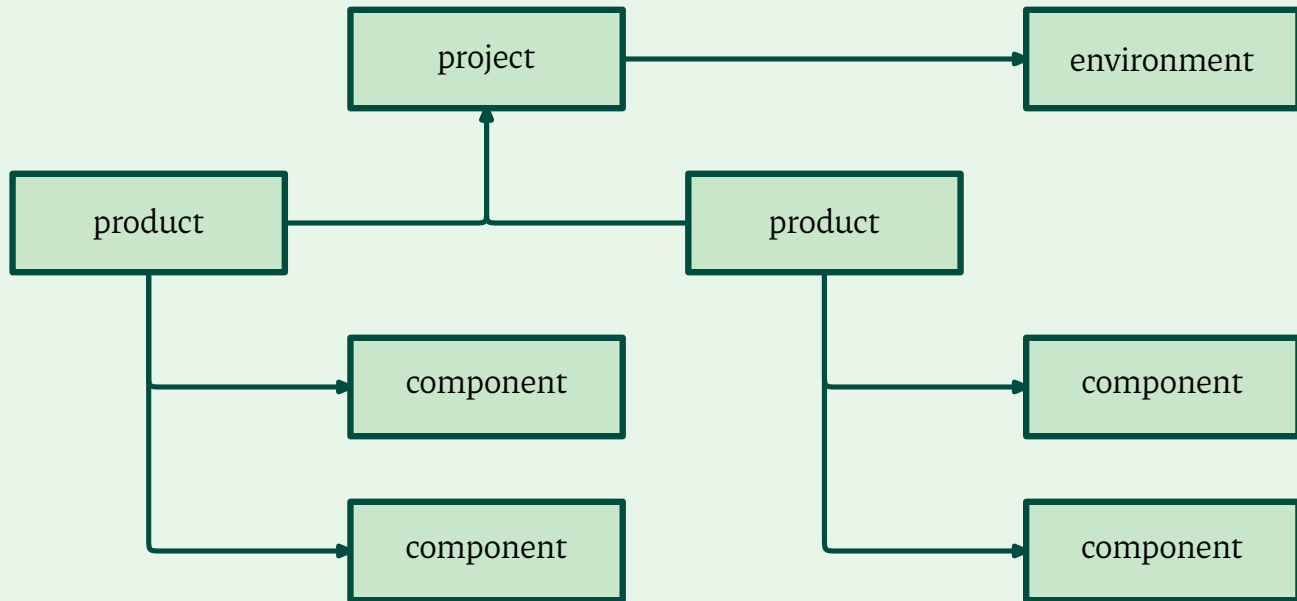
⋄ The *environment* file shared by multiple files (e.g. for a magazine).

⋄ The style has been split over multiple file to make it easier to handle.

The *environment* file(s) are now loaded from the *project* file and the *product* and *component* files load the *project* file.

Project structure

The structure remains the same as with *products* but the files load now the *project* file for the document style.

Should I load an *environment* before or after \startproduct?

When *product* and *component* files are used the environment can either be loaded before or after the corresponding start command.

In many case there is no difference between

```
\environment docment-style

\startproduct *
```

and

```
\startproduct *

\environment docment-style
```

Loading the *environment* before \startproduct

When an *environment* file is loaded before the start-command of the content file and there is a custom font setting in the *environment*, no default fonts are loaded.

Loading the *environment* after \startproduct

When an *environment* is loaded after the start-command of the content file, CONTEXT always loads its default fonts before custom fonts in the style are enabled.

A feature which can be used in this case are the *product* and *component* modes. When the main file is the *product* its mode is active in the *environment* but when only a single *component* is processed its mode is active. This can be used to change the style of section titles or other elements.

```
\startmode[*component]
   . . .
\stopmode
```

\stoptext