Introduction

The interface definitions have a long history. They started out in as TEX commands, something `\start` ... `\stop` with embedded specifications. When ConTeXt became larger and XML showed up the definitions were converted into XML and instead of putting the definitions in the source files they moved to one file: `cont-en.xml`.

When at some point the number of commands not covered grew and the covered ones lagged behind reality, Wolfgang started to systematically collect all the information needed to make a more complete set of definitions. In the process we enhanced the supported syntax variants and added more methods to share common definitions. The current set of files describes all commands (even those not really meant for users). Because the definitions are also used to generate files for editors like SciTE, there are some tools that operate on the XML file. If needed one can still generate the large files (one per interface with merged definitions).

Overviews

The files describing the interface can be recognized by the prefix `i-` and suffix `xml`. We don't explain the syntax here as those files give enough examples of usage.

`i-context`	the main file (it loads other files)
`i-common-definitions`	common definitions that save time and space when defining others
`i-common-*`	files loaded by the common definition file
`i-*`	the setups organized by functionality

There are a couple of styles that implement the rendering of the interface commands (traditionally called setups):

`x-setups-basics`	loading of definitions and rendering of compact of extensive interface commands
`x-setups-overview`	generate a document with all commands using the large combined definition file
`x-setups-generate`	generate a document with all commands using the individual files but generate the combined file in the process
`x-setups-proofing`	used for direct rendering of a file where commands are defined

The proofing only works when there is the following line in a definition file:

```
<?context-directive job ctxfile x-setups.ctx ?>
```

In that case running the `context` command on the file will render the defined commands.

```
context i-backend.xml
```

If you want the combined XML file(s), you need to call:

```
context x-setups-generate.mkiv
context x-setups-generate.mkiv --interface=nl --result=setup-nl
```

For each relevant interface. If you don't want that, and save quite some disk space, you can use:

```
context x-setups-overview.mkiv
context x-setups-overview.mkiv --interface=nl --result=setup-nl
```

Instead of these commands you can also do this:

```
context --extra=setups --overview
context --extra=setups --overview --save
context --extra=setups --overview --interface=nl
context --extra=setups framed
```

Use in manuals

todo

Keeping up

We try to keep up with additions in ConTeXt but it might be that we forget some. If you run into issues when processing, can't find what should be there, or find a discrepancy in a manual (like the beginners manual) you can contact us.

Wolfgang Schuster
Ton Otten
Hans Hagen