

ŠäferT_EX: Source Code Esthetics for Automated Typesetters

Frank-Rene Schaefer
(private research)
<frank_r_schaefer@gmx.net>

February 26, 2005

Abstract

In 2003, the first attempts towards ŠäferT_EX were made, targeting to create a text processing with the goal of optimizing its ease of use, i.e. the beauty of its code appearance. The macro based T_EX system still lives in niches of typesetting experts, computer scientist, and engineers who are willing to learn the overhead required to use the system - for the sake of high quality typesetting. ŠäferT_EX faced the challenge to provide an interface language that does only differ minimally from a normal human edited text, while the compiler itself extracts the commands it requires.

Despite to simple wrappers programs, ŠäferT_EX is a real three phases compiler, consisting of a lexical analyzer, a parser, and a code generator. In the last year the system mainly underwent internal changes that allowed to maintain this structure, while allowing a rather unusually simple and transparent programming syntax. As a consequence, the system now reached a robustness, so that it can be used by a wider audience.

This presentation shall give the reader an overview over the system of ŠäferT_EX and demonstrate its abilities with a example application. Also, the fundamental ideas which allowed to maintain the classical three phases compiler structure are introduced. With the start of this conference the system can be downloaded at safertex.sourceforge.net.

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

ŞäferTeX

Source Code Esthetics

Frank-René Schäfer

March 9, 2005 / EuroTeX-2005

Frank-René Schäfer

ŞäferTeX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŞäferTeX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 Parsing
 - The ŞäferTeX Engine
 - Achievements (Brief)
- 4 Summary

Frank-René Schäfer

ŞäferTeX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

The Goals

Optimized Code Appearance

- **Avoiding Visual Noise**
 - 'Natural Appearance' of Commands
 - Unicode Based Input
- **Three Phases Compiler**
 - Lexical Analyzer
 - Parser
 - Code Generator (for TeX)
- **Organization of Variables**
 - Symbol Table Mechanism (OO Support)
 - XML Interfaces

Frank-René Schäfer

ŞäferTeX

Goals
Paradigms

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŞäferTeX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 Parsing
 - The ŞäferTeX Engine
 - Achievements (Brief)
- 4 Summary

Frank-René Schäfer

ŞäferTeX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration
Goals
Paradigms

Paradigms of ŠaferT_EX

Means to Improve Code Appearance II

Example

```
\section{Voilà, une section}
--- \item character sequences that do something
--- either start
--- \item with a backslash, or
--- \item with a double non-letter character
--- \item und nun ein wieder Gliederungspunkt in der
--- \item \*(u)bergeordneten Liste.

\begin{quote}
We should not produce \#100 pieces/week in order to get 10\%
more profit, if the adaption process costs us \$25,000.
\end{quote}
```

- Automatic Scope/Group Detection
- Indentation as Scope Delimiter

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration
Goals
Paradigms

Paradigms of ŠaferT_EX

Means to Improve Code Appearance

Example

```
\section{Voilà, une section}
\begin{itemize}
\item character sequences that do something
either start
\begin{enumerate}
\item with a backslash, or
\item with a double non-letter character
\end{enumerate}
\item und nun ein wieder Gliederungspunkt in der
\item \*(u)bergeordneten Liste.
\end{itemize}

\begin{quote}
We should not produce \#100 pieces/week in order to get 10\%
more profit, if the adaption process costs us \$25,000.
\end{quote}
```

- Automatic Scope/Group Detection

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration
Processing
The Country Hut Solution

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŠaferT_EX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 Parsing
 - The ŠaferT_EX Engine
 - Achievements (Brief)
- 4 Summary

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration
Goals
Paradigms

Paradigms of ŠaferT_EX

Means to Improve Code Appearance III

Example

```
(*) Voilà, une section
-- character sequences that do something
-- either start
-- ## with a backslash, or
-- ## with a double non-letter character
-- und nun ein wieder Gliederungspunkt in der
-- übergeordneten Liste.

\quote:
We should not produce \#100 pieces/week in order to get 10 %
more profit, if the adaption process costs us $25,000.
```

- Automatic Scope/Group Detection
- Indentation as Scope Delimiter
- Intuitive Treatment of Characters

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŠaferT_EX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 Parsing
 - The ŠaferT_EX Engine
 - Achievements (Brief)
- 4 Summary

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis Process

Example

Imagine a little more complicated case with many items, nested lists, tables, quotes and so on. Isn't this much more complicated? How can ...

-- This is verily the first item in this example.

And for clarity, this item even has a paragraph - just to show a little more of the problems that may appear.

-- Well, this item here is just another one, but it has some sub items as can be seen below:

```
## The higher indentation tells us that this item belongs to a nested
group.
## The 'sharp' sign tells us that the item list, well it is an
enumeration.
But now, let us have a table environment, because this is now
real fun.
\table: Tabular which shows some things just for demonstration.
Product    & Price $/pp    && Amount ($)
Shoes      10,21             5    @@
Socks      1,23              2    @@
Gloves     3,00              2    --
-- Suddenly, we are in the first item list again ...
```

And eventually, all the item lists are closed and we are reading ...

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

\quote: Der Apfel fällt nicht weit ...

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

\quote: Der Apfel fällt nicht weit ...

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

`\quote: Der Apfel fällt nicht weit ...`

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis III

The Country Hut Solution

`\quote: Der Apfel fällt nicht weit ...`

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

The ŠaferT_EX Engine
Achievements (Brief)

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŠaferT_EX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 Parsing
 - The ŠaferT_EX Engine
 - Achievements (Brief)
- 4 Summary

Frank-René Schäfer
ŠaferT_EX

Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Processing
The Country Hut Solution

Lexical Analysis

Achievements on Lexical Analysis

- 1 Object Oriented Lexical Analysis (-> 'quex')
- 2 Country Hut Solution
(Framework for indentation based lexical analysis)
- 3 Platform for the '3 Paradigms of ŠaferT_EX'
- 4 Classical Token Stream as Output

Frank-René Schäfer
ŠaferT_EX

The ŠaferT_EX Engine
Achievements (Brief)
Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Outline

- 1 Introduction
 - Optimized Code Appearance
 - Paradigms of ŠaferT_EX
- 2 Lexical Analysis
 - Processing
 - The Country Hut Solution
- 3 **Parsing**
 - The ŠaferT_EX Engine
 - **Achievements (Brief)**
- 4 Summary

The ŠaferT_EX Engine
Achievements (Brief)
Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Parsing

The ŠaferT_EX Engine

ŠaferT_EX
Frank-René Schäfer
Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Summary

- Development of tools for lexical analysis: **Quex** (Object oriented pattern recognition)
- Development of tools for parser generation: **Lemon++** and **Quakk**.
- Clarity of code and Ease of programming)
- Stable platform for the '3 Paradigms of ŠaferT_EX'
 - Automatic Scope/Group Detection
 - Indentation as Scope Delimiter
 - Intuitive Treatment of Characters
- **First official release of ŠaferT_EX 0.0.0 at sourceforge.net**

ŠaferT_EX
Frank-René Schäfer
Introduction
Lexical Analysis
Parsing
Summary
Demonstration

Parsing

Problems

- 1 Proliferation of rules
- 2 Confusion by **Precedence rules**
- 3 Malformation of parse trees

Solutions

- 1 Improvements to parser generator (-> 'Lemon++' and 'quakk')
- 2 Grammar clarity by separation of rules and rule action
- 3 Streaming ability of operational classes
- 4 Automatic C++ code generation for non-terminal tokens
- 5 Avoidance of Precedence
- 6 Automatic XML code generation for parse tree