

Structorizer

User Manual

Table of Contents

1. Preface.....	2
1.1. Getting Started.....	2
2. User Interface.....	3
2.1. The Menu.....	3
2.1.1. File.....	3
2.1.2. Edit.....	3
2.1.3. Diagram.....	4
2.1.4. Preferences.....	4
2.1.5. Help.....	5
3. Code Parsing.....	5
4. LaTeX – StrukTeX.....	5
5. Index.....	6

1. Preface

Structorizer has been written by Robert Fisch.

The purpose of this program is to create Nassi-Schneiderman Diagrams. The project's homepage is located at <http://structorizer.fisch.lu> .

This User Manual has been written by David Morais. If you have found any typos or got anything else to report just send an email to narutodc@hotmail.com .

1.1. Getting Started

Structorizer is quite simple to use. I will shortly explain the Interface and then concentrate on Structorizer's features. I will explain the features using the Menu because it's easier for me to explain the features using the menu. I used the default theme called *Metal* on the screenshots.

I used the italic style for filenames and names of packages and bold for path addresses in the Menu.

2. User Interface

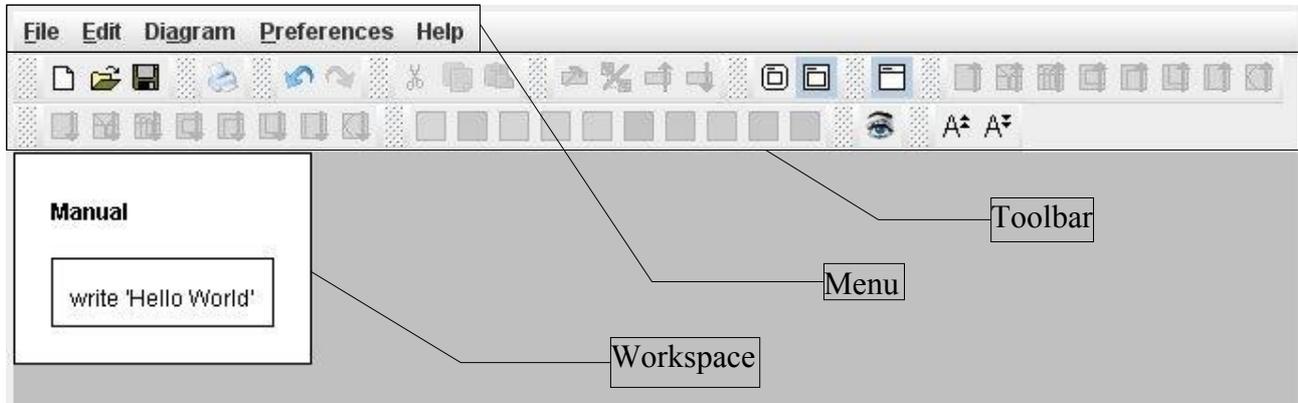


Illustration 1: The User Interface

The UI (User Interface) is quite minimalist and simple to use. It is built of a Toolbar that has gotten shortcuts to the features, the Menu which contains everything the Toolbar does and even some more features and the Work Area which is where you create your NSD.

2.1. The Menu

As I said earlier, I will explain the the most advanced features in a different chapter, but most of the Menu Items are self explaining.

2.1.1. File

This Menu Item mostly contains Features you already know such as *Open*, *Save*, *Print* and *Quit*.

I will however explain *Import* and *Export*.

In *Import* you can select a pascal source code file to be converted into a NSD.

In *Export* you can take your existing NSD and convert it into an jpeg Picture, convert it into a pascal source code file or save it as a LaTeX file using the package called *struktex*.

2.1.2. Edit

Another Menu Item you probably know for what you use it. It's items like *Undo* / *Redo*, *Copy* and *Paste*.

2.1.3. Diagram

When you want to add a block into your NSD then you click on an existing block, and click on **Diagram/Add** and then you select if you want to add the block before the one you selected or after the block you selected, and then you select which type of block you want to add.

You can also move up and down blocks, just select the block you want to move and click on **Diagram/move up** or **Diagram/move down**.

There are two types of NSD Designs you can choose between, the *Main* and the *Sub*, the difference between *Main* and *Sub* is that *Sub* got rounded edges.

With the Checkbox *Boxed Diagrams?* You can select if you want to have a border around the NSD or not.

When you add an block into your NSD, you can also add a comment, which you can show or hide with the Checkbox *Show Comments?*.

A recent add into Structorizer is the Variable highlighting, it shows you the variables in a color.

You can enable or disable it with the Checkbox *Highlight variables?*.

2.1.4. Preferences

You can select the font size in **Preferences/Font**.

When you select a Block you can paint it's background to a different color than white, you can change those default colors in **Preferences/Colors**.

Sometimes you may need to change the default labels for Blocks like *if*, *repeat until*, *for* and *while* for example when you create your NSD in a different language. You can change those default labels in **Preferences/Structures**.

Perhaps you have already noticed it, but Structorizer is able to parse Pascal source code into and to parse a NSD into pascal source code. You can change the options in **Preferences/Code Parser**

If you are used to the versions of Structorizer on Windows or Linux, you might be confused with this new theme called *Metal*, but you can change it to a few different themes in **Preferences/Look & Feel**.

2.1.5. Help

Here you can find the Changelog and the Credits.

3. Code Parsing

A very interesting feature is the Code parser. A parser is basically a program that determines the grammatical structure of a programming language (Pascal in this case). You can select a pascal source code file in **File/Import/Pascal code**.

However, when you need to write your NSD's in another language, like French for example, you need to translate the English keywords (like *While*, *for ... do*) into French keywords. This can be done in the Parser Preferences, when using the parser with a for loop, the *pre* Textbox is the word *for* and the Post Textbox would be *do*. In French those keywords are called *pour* and *à*, so you change the TextBox content.

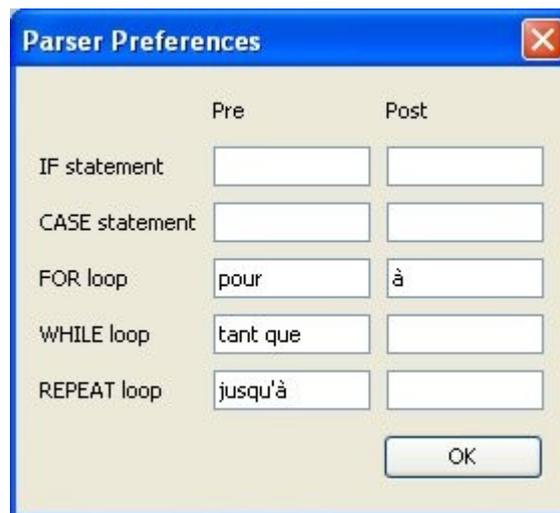


Illustration 2: My parser Preferences set to French NSD

4. LaTeX – StrukTeX

You can save a NSD for LaTeX by clicking on **File/Export/StrukTeX Code**. You need to have the StrukTeX package installed to be able to create NSD's in LaTeX.

If you don't know how to use StrukTeX refer to the program's documentation. You can get the package and the documentation at <http://ctan.tug.org/tex-archive/macros/latex/contrib/struktex>

5. Index

Alphabetical Index

Boxed Diagrams?.....	4
Code Parsing.....	5
Export.....	3
Highlight variables?.....	4
Import.....	3
LaTeX – StrukTeX.....	5
Show Comments?.....	4
Structures.....	4
User Interface.....	3