Coding standards for The Meta-Environment

Jurgen Vinju

2007-12-04 16:55:04 +0100 (Tue, 04 Dec 2007)

Table of Contents

Introduction .................................................................................................................... 1
Coding style in C and Java ............................................................................................ 1
Coding style in Makefile.am and configure.{ac,in} ....................................................... 2
Coding style in T-script (ToolBus) ............................................................................... 2
Coding style in SDF .................................................................................................... 2
Coding style in ASF+SDF ........................................................................................... 2

Introduction

This document represents an ongoing effort to document our coding standards. The descriptions are to be short and clear. This information is not complete, please help.

Coding style in C and Java

• "Code Complete" by Steve McConnel influences our basic guidelines, please read it.

• We always use javaCase for identifiers such as function names, method names and variable names. In javaCase, you start with a lowercase letter, and separate words by starting new words with uppercase letters.

  • Example: myIdentifierForExamplePurposes

  • Exception: all API in C start with a prefix start in all uppercase and end with an underscore.
    Example: PT_makeTreeAppl

  • Types or classes start with capitals, and continue with javaCase. There might be a prefix, which is all capitals and ends with an underscore again.

  • Examples: ATermList, StackNode, PT_Tree

• We never have global variables in C or Java public interfaces. Always use getters and setters. Global variables are sometimes used, but not exported in any interface.

• Use -Wall -Werror for gcc

• In C all function and variables local to a file must be declared static.

• In C all functions that do not change the value of a pointer argument must declare it const.

  • Example: void printMessage(const char* message);

• Options to pass to indent:

  • --indent-level2 --braces-on-if-line --no-space-after-function-call-names
Coding style in Makefile.am and configure. {ac,in}

- Only Makefile.in files are instantiated at configure time by the AC_OUTPUT macro, all other build/substitute/instantiation is to be done by automake targets (using gcc or sed for example).
- Do not use the name configure.in anymore, use configure.ac

Coding style in T-script (ToolBus)

- Do not use process calls with return variables that are not used
- Remove all unused variables in lets
- Use process calls sooner than snd-msg/rec-msg for procedure execution

Coding style in SDF

- Group productions on the non-terminal on the right-hand side of productions
- Do not duplicate productions between priority sections and normal syntax sections
- Fix all warnings mentioned by the SDF checker
- Non-terminals (sorts) start with a capital letter, and continue with javaCase

Coding style in ASF+SDF

- Hide all variable definitions in SDF (hiddens)
- Group all equations for a single outermost function in an equations section
- Put the default equations last
- Write test equations for non-trivial functions
- Fix all warnings mentioned by the ASF checker
- import basic/Whitespace and basic/Comments only in hiddens sections to prevent contamination of other modules with arbitrary LAYOUT definitions