fox-listings

Listings language definition for COSYScript

Eremey Valetov
https://github.com/evvaletov

March 21, 2026 v1.4

1 Introduction

The `fox-listings` package provides a language definition for the `listings` package to typeset source code in COSYScript, the programming language of COSY INFINITY, a beam dynamics code using high-order differential-algebraic (DA) transfer maps and methods, developed at Michigan State University. COSYScript is compiled and executed by the FOXY subsystem; source files use the `.fox` extension. This package uses the name `FOX` for the `listings` language identifier and style prefixes.

The package defines six keyword groups that can be independently styled, supports nested {...} comments and single-quoted string literals, and provides two ready-made styles.

2 Usage

Load the package after `listings`:

```
\usepackage{fox-listings}
```

The package automatically loads `listings` and `xcolor`.

2.1 Color style

```
\begin{lstlisting}[style=FOXcolor]
  ...
\end{lstlisting}
```

2.2 Monochrome style

```
\begin{lstlisting}[style=FOXmono]
  ...
\end{lstlisting}
```

2.3 Language only (custom style)

```
\begin{lstlisting}[language=FOX]
  ...
\end{lstlisting}
```

2.4 Inline code

```
\lstinline[language=FOX]{VARIABLE X 1 ;}
```

3 Keyword groups

Keywords are split into six groups so that each category can be styled independently. Groups 1–3 cover the general-purpose language (control flow, math, DA algebra), group 4 covers the beam-physics command set, group 5 covers the graphics subsystem, and group 6 covers built-in constants and global variables. The `FOXcolor` style assigns a distinct color to each group; the `FOXmono` style bolds groups 1, 2, and 6, and leaves the rest unstyled.

All keywords have been cross-checked against the pristine COSY INFINITY v10 source (`cosy.fox` procedures, functions, and global variables) and the FOXY compiler intrinsic tables (`foxy.f`). Group 3 also includes five COSYFFAG-specific configuration procedures (`CONFIG_SET`, etc.) that are not in the pristine distribution.

Group	Category	Examples
1	Control flow, declarations	`PROCEDURE`, `IF`, `VARIABLE`, `WRITE`
2	Intrinsic functions	`SIN`, `SQRT`, `ABS`, `CONS`, `DA`
3	Intrinsic procedures	`DAINI`, `VELSET`, `CONFIG_SET`
4	Beam physics	`OV`, `MQ`, `CR`, `FR`, `ER`
5	Graphics	`GRMOVE`, `GRDRAW`, `GREPS`, `PP`
6	Constants/globals	`PI`, `CLIGHT`, `MAP`, `RAY`

4 Examples

4.1 Color style (`FOXcolor`)

```
INCLUDE 'COSY' ;

PROCEDURE GREET A B ;
   VARIABLE C 1 ;
   C := A + B ;
   WRITE 6 'Sum =' C ;
ENDPROCEDURE ;

PROCEDURE RUN ;
   VARIABLE X 1 ;
   VARIABLE Y 1 ;
   VARIABLE B 1 ;

   {Compute and display a value}
   X := SIN(0.5) ;
   Y := SQRT(X) + 1 ;
   WRITE 6 'Result:' Y ;

   GREET 3 4 ;

   OV 3 2 0 ;
   RP 10000 1.00728 1 ;
   B := 0.01*CONS(CHIM)*0.05 ;
   UM ;
   MQ 0.5 B 0.05 ;
   DL 24 ;
   MQ 1 -B 0.05 ;
   DL 24 ;
   MQ 0.5 B 0.05 ;
   CR ;
ENDPROCEDURE ;
RUN ; END ;
```

4.2 Monochrome style (FOXmono)

```
INCLUDE 'COSY' ;

PROCEDURE RUN ;
   VARIABLE KF 1 ; VARIABLE KD 1 ; VARIABLE OBJ 1 ;

   OV 1 2 0 ;
   RP 10000 1.00728 1 ;
   KF :=  0.01*CONS(CHIM)*0.05 ;
   KD := -0.01*CONS(CHIM)*0.05 ;

   FIT KF KD ;
      UM ;
      MQ 0.5 KF 0.05 ;
      DL 24 ;
      MQ 1 KD 0.05 ;
      DL 24 ;
      MQ 0.5 KF 0.05 ;
      OBJ := (ME(1,2))^2 + (ME(3,4))^2 ;
   ENDFIT 1E-10 100 1 OBJ ;

   WRITE 6 'Matched KF:' KF ;
   WRITE 6 'Matched KD:' KD ;
ENDPROCEDURE ;
RUN ; END ;
```

4.3 Tune extraction

```
INCLUDE 'COSY' ;

PROCEDURE COMPUTE_TUNES ;
   VARIABLE MU 100 2 ;
   VARIABLE B 1 ;
   B := 0.01*CONS(CHIM)*0.05 ;

   {Build a FODO cell transfer map}
   UM ;
   MQ 0.5 B 0.05 ;
   DL 24 ;
   MQ 1 -B 0.05 ;
   DL 24 ;
   MQ 0.5 B 0.05 ;

   {Extract tunes}
   TP MU ;
   WRITE 6 'Tunes: '&SF(CONS(MU(1)),'(F8.5)')&'  '&SF(CONS(MU(2)),'(F8.5)') ;
ENDPROCEDURE ;

PROCEDURE RUN ;
   OV 3 2 0 ;
   RP 10000 1.00728 1 ;
   COMPUTE_TUNES ;
ENDPROCEDURE ;
RUN ; END ;
```

4.4 Including an external file

Use `\lstinputlisting` to typeset an external `.fox` source file:

```
\lstinputlisting[style=FOXcolor, caption={Simulation program},
  firstline=1, lastline=30]{EEFFAGsim.fox}
```

5 Known limitations

The `listings` package does not highlight in-code numbers for user-defined languages. Fortran-style `D`-exponent notation (e.g., `1.5D-3`) and standard decimal literals (`0.5`, `1E-3`) are rendered in the base style.

The `listings` tokenizer has no scope or context awareness, so a user variable that shares its name with a keyword (e.g., `OV`, `MQ`, `CR`) will be highlighted as a keyword.

6 License