

NAME

XML::Parser::Style::Stream – Stream style for XML::Parser

SYNOPSIS

```

use XML::Parser;
my $p = XML::Parser->new(Style => 'Stream', Pkg => 'MySubs');
$p->parsefile('foo.xml');

{
    package MySubs;

    sub StartTag {
        my ($e, $name) = @_;
        # do something with start tags
    }

    sub EndTag {
        my ($e, $name) = @_;
        # do something with end tags
    }

    sub Characters {
        my ($e, $data) = @_;
        # do something with text nodes
    }
}

```

DESCRIPTION

This style uses the Pkg option to find subs in a given package to call for each event. If none of the subs that this style looks for is there, then the effect of parsing with this style is to print a canonical copy of the document without comments or declarations. All the subs receive as their 1st parameter the Expat instance for the document they're parsing.

It looks for the following routines:

- StartDocument

Called at the start of the parse .

- StartTag

Called for every start tag with a second parameter of the element type. The \$_[0] variable will contain a copy of the tag and the \$_[1] variable will contain attribute values supplied for that element.

- EndTag

Called for every end tag with a second parameter of the element type. The \$_[0] variable will contain a copy of the end tag.

- Text

Called just before start or end tags with accumulated non-markup text in the \$_[0] variable.

- PI

Called for processing instructions. The \$_[0] variable will contain a copy of the PI and the target and data are sent as 2nd and 3rd parameters respectively.

- EndDocument

Called at conclusion of the parse.