NAME

Parsers::YYLexer

SYNOPSIS

use Parseres::YYLexer;

use Parsers::YYLexer qw(:all);

DESCRIPTION

YYLexer class provides the following methods:

new, GetYYLex, Next, Peek, SetupYYTabFile, StringifyYYLexer, YYLex

Parsers::YYLexer class is derived from Parsers::Lexer base class, which provides all the underlying lexer functionality. YYLexer class is designed to be used with yyparse code generated by running byacc on a parsers defined using parser definition ParserName.yy file.

YYTabFile containing mapping of token labels to integers must be explicitly specified by the caller. This file is processed during new method invocation and mapping of token labels to integers is loaded in a hash to be used later by YYLex method to return token number and text pairs to the parser.

METHODS

new

```
$YYLexer = new Parsers::YYLexer($Input,  @YYLexerTokensSpec);
```

Using specified *Input* and *YYLexerTokensSpec*, new method generates a new YYLexer and returns a reference to newly created YYLexer object.

Examples:

```
# Tokens specifications supplied by the caller. It's an array containing references
# to arrays with each containing TokenLabel and TokenMatchRegex pair along with
# an option reference to code to be executed after a matched.
#
@LexerTokensSpec = (
    [ 'LETTER', qr/[a-zA-Z]/ ],
    [ 'NUMBER', qr/\d+/ ],
    [ 'SPACE', qr/[ ]*/,
        sub { my($This, $TokenLabel, $MatchedText) = @_; return ''; }
    ],
    [ 'NEWLINE', qr/(?:\r\n|\r|\n)/,
        sub { my($This, $TokenLabel, $MatchedText) = @_;  return "\n"; }
    ],
    [ 'CHAR', qr/./ ]
);

# Input string...
$InputText = 'y = 3 + 4';

$YLexer = new Parsers::YYLexer($InputText,  @LexerTokensSpec);

# Setup default token table file...
$YYTabFile = "Parsers/SimpleCalcParser.tab.ph";
$This->SetupYYTabFile($YYTabFile);

# Process input stream...
($TokenNumber, $TokenText) = $YYLexer->Lex();
print "TokenNumber: $TokenNumber; TokenText: $TokenText\n";

# Input file...
$InputFile = "Input.txt";
open INPUTFILE, "$InputFile" or die "Couldn't open $InputFile: $!\n";
$Lexer = new Parsers::YYLexer(\*INPUTFILE, @LexerTokensSpec);

# Input file iterator...
$InputFile = "TestSimpleCalcParser.txt";
open INPUTFILE, "$InputFile" or die "Couldn't open $InputFile: $!\n";
$InputIterator = sub { return <INPUTFILE>; };
$Lexer = new Parsers::YYLexer($InputIterator, @LexerTokensSpec);
```

```
# Usage with code generated by byacc from a parser definition
# file SimpleCalcParser.yy...

$InputText = "3 + 4 +6\nx=3\ny=5\nx+y\nx+z\n";

$YYLexer = new Parsers::YYLexer($InputText,@LexerTokensSpec);

$YYLex = $YYLexer->GetYYLex();

$YYTabFile = "Parsers/SimpleCalcParser.tab.ph";
$YYLexer->SetupYYTabFile($YYTabFile);

$Debug = 0;
$SimpleCalcParser = new Parsers::SimpleCalcParser($YYLex,
                        \&Parsers::SimpleCalcParser::yyerror, $Debug);

$Value = $SimpleCalcParser->yyparse();
print "Value = " . (defined($Value) ? "$Value" : "Undefined") . "\n";
```

GetYYLex

```
$YYLex = $YYLexer->GetYYLex();
```

Returns a curried verson of YYLexer as YYLex: yyparse in parser generated by byacc expects it to call without passing any argument for the *YYLexer* object.

Next

```
($TokenNumber, $TokenText) = $YYLexer->Next();
```

Returns next available TokenNumber and any matched TokenText from input stream by removing it from the input stream. Token number and text of zero corresponds to end of input (EOI).

Peek

```
($TokenNumber, $TokenText) = $YYLexer->Peek();
```

Returns next available TokenNumber and any matched TokenText from input stream by simply looking ahead and without removing it from the input stream. Token number and text of zero corresponds to end of input (EOI).

SetupYYTabFile

```
$YYLexer = $YYLexer->SetupYYTabFile($YYTabFile);
```

Processes token labels to integers data map in specified *YYTabFile* and returns *YYLexer*.

Notes:

```
. YYTabFile must be a complete path or available through @INC path in the
  same directory where this package is located.
. Name of YYTabFile might start with any valid sub directory name in @INC
  For example, "Parsers/<YYTablFile>" implies the tab file in parsers sub
  directory under MayaChemTools lib directory as it would be already in @INC
  path.
. YYTabFile must be explicitly set by the caller. The default YYTabFile name,
  y.tab.ph, generated by byacc is not used implicitly to avoid confusion
  among multiple distinct instances of YYLexer.
. YYTabFile is generated by byacc during its usage with -d options and
  contains mapping of token codes to token names/labels. YYLexer used this
  file to map token labels to token codes before returning token code and
  value pair back to yyparse function used by byacc.
. User defined token numbers start from 257
. Token number for any user defined token EOI is mapped to its value before
  default token number of 0 for EOI.

The format of YYTabFile generated by byacc during generation of parser code in
Perl code is:

... ...
$NUMBER=257;
$ADDOP=258;
```

```
$SUBOP=259;
... ..
```

YYLex

```
($TokenNumber, $TokenText) = $YYLexer->YYLex();
($TokenNumber, $TokenText) = $YYLexer->YYLex($Mode);
```

Returns available TokenNumber and any matched TokenText from input stream by either removing it from the input stream or by simply looking ahead and without removing it from the input stream. Token number and text of zero corresponds to end of input (EOI).

Possible *Mode* values: *Peek*, *Next*. Default: *Next*.

YYLex is designed to be used with yyparse code generated by running byacc on a parsers defined using parser definition ParserName.yy file.

Notes:

. Token label and value pairs returned by Lexer from by base class, which can't be mapped to token labels specified in YYTabFile are ignored.
. Token text of length 1 returned by Lexer from base class without a corresponding explicit token label, which can't be mapped to a token number using Perl ord function, is ignored.

StringifyYYLexer

```
$YYLexerString = $YYLexer->StringifyYYLexer();
```

Returns a string containing information about *YYLexer* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

Lexer.pm, SimpleCalcYYLexer.pm, SimpleCalcParser.yy

COPYRIGHT