

# Phase 1: Parser General Changes

|                                       |          |
|---------------------------------------|----------|
| <b>Keywords</b>                       | <b>2</b> |
| Removing Keywords                     | 2        |
| Adding Keywords                       | 2        |
| Adding string predeclared identifiers | 3        |
| <b>Character classes</b>              | <b>3</b> |
| Added new input character classes     | 3        |
| <b>Syntax Tokens</b>                  | <b>4</b> |
| Deletion of PT Syntax Tokens          | 4        |
| Insertion of Quby Syntax Tokens       | 4        |
| Insertion of Quby Syntax Symbols      | 5        |
| <b>String Literals</b>                | <b>6</b> |
| Deletion of PT Alternative Comments   | 6        |
| Insert Quby Comments                  | 7        |

## Keywords

### Removing Keywords

**Where:** In stdIdentifiers, in the keywords section

**How:** Removed not until program const procedure begin repeat

**Why:** These keywords are no longer recognized as keywords in Quby (example: Pascal repeat is now Quby do, so we must remove repeat)

### Adding Keywords

**Where:** In stdIdentifiers, in the keywords section

**How:** Added using val def break when module unless elsif

stdidentifiers

```
using
val
def
break
when
```

```
module
unless
elsif
```

**Why:** These are new keywords that must be added to Quby to support its functional changes from PT Pascal (example: Quby has a cascade elsif statement, so we must add elsif keyword)

Adding string predeclared identifiers

**Where:** In stdIdentifiers, in predeclared identifiers, in the predeclared type names section

**How:** Removed char, added string

Alter:

```
% Predeclared type names
integer
char
boolean
text
```

To:

```
% Predeclared type names
integer
string
boolean
text
```

**Why:** Quby supports varying length strings which were not supported in PT Pascal

## Character classes

Added new input character classes

**Where:** In scan.ssl and parser.pt input tokens

**How:** Added new classes and changed lQuote to double quotes

```
lQuote          '"'
lPercent         '%'
lHash            '#'
lExclamation     '!'
lQuestion        '?'
```

```
lDollar      '$'
```

Updated character class map in parser.pt

```
lHash = 28;  
lExclamation = 29;  
lQuestion = 30;  
lDollar = 31;
```

**Why:** Quby uses new character classes for %, #, !, ?, \$ and double quote instead of single.

## Syntax Tokens

### Deletion of PT Syntax Tokens

**Where:** In the scan.ssl file in the output section and the parser.pt file under output tokens

**How:** The old syntax tokens of not, until, program, const, procedure, begin, and repeat were deleted from the file (i.e pNot, pUntill, etc.)

**Why:** These syntax tokens no longer apply to the Quby programming language. The remainder of the syntax tokens are still applicable in Quby so they were left

### Insertion of Quby Syntax Tokens

**Where:** In the scan.ssl file in the output section and the parser.pt file under output tokens

**How:** The new syntax tokens of using, val, def, unless, elsif, break, when and module were inserted. The order of the insertion is the same as in the stdIdentifiers.

scan.ssl

```
pUsing  
pVal  
pDef  
pBreak  
pWhen  
pModule  
pUnless  
pElsif
```

parser.pt

```
pUsing = 16;  
pVal = 17;  
pDef = 18;  
pBreak = 19;
```

```
pWhen = 20;  
pModule = 21;  
pUnless = 22;  
pElsif = 23;
```

**Why:** These syntax tokens are needed in the Quby language and define what the keywords are that will be used in commands

### Insertion of Quby Syntax Symbols

**Where:** In the scan.ssl file output section and the parser.pt file under the output tokens

**How:** New tokens were added for the syntax symbols not, question, dollar, and hash. They were defined later scan case statement in the scan.ssl file. Further modifications to the case statement were made to alter the equals. Colon equals was changed to just colon and on the equals case, a separate case for double equals was added.

scan.ssl

```
pPercent  
pHash  
pExclamation  
pQuestion  
pDollar  
pDoubleEquals
```

parser.pt

```
pPercent = 47;  
pHash = 48;  
pExclamation = 49;  
pQuestion = 50;  
pDollar = 51;  
pDoubleEquals = 52;
```

scan.ssl

```
| '%' :  
    @Comment  
| '#' :  
    .pHash  
| '!' :  
    .pExclamation  
| '?' :
```

```

        .pQuestion
    | '$':
        .pDollar

```

```

| '=':
    [
        | '=':
            .pDoubleEquals
        | INewLine, INewPage:
            .pEquals
        >
        | '*':
            .pEquals
        >
    ]

```

**Why:** The new tokens were needed in order for the new symbols to be able to be referenced. This change allows the new Quby symbols to be referenced and the token to be outputted when the scanner reads in the symbol.

## String Literals

**Replace** PT single quotes with Quby double quotes for String Literals

**Where:** In the scan.ssl file under input tokens and in parser.pt under the Initialization procedure

**How:** PT References indicating that a string literal is identified with a single quote (IQuote parameter) is now replaced with a Quby double quote. IQuote value is tied to quote, which is then referenced throughout the two files to identify string literals.

```

charClassMap[ord('\"')] := IQuote;
quote = '\"';
IQuote = '\"

```

**Why:** PT uses single quotes to identify string literals, while Quby uses double quotes.

## Comments

### Deletion of PT Alternative Comments

**Where:** In the scan.ssl file in the rules section

**How:** Instead of allowing the compiler to recognize left parenthesis as a possible start to an alternative comment, it now only recognizes it as a left parenthesis. Additionally, the call function that identifies alternative comments is removed.

Remove:

```

AlternateComment :
    {
        | '*' :
        [
            | ')' :
            >
            | * :
        ]
        | lNewLine:
        .pNewLine
        | lEndFile:
        #eCommentEOF
        .pEndFile
        >
        | * :
        ?
    }
};

```

Alter:

```

    | '(' :
    [
        | '*' :
        @AlternateComment
        | * :
        .pLeftParen
    ]

```

To:

```

    | '(' :
    .pLeftParen

```

**Why:** Alternative Comments are not used in Quby

## Insert Quby Comments

**Where:** In the scan.ssl file in the rules section in the Comment function

**How:** The choice that indicates an end of comment with a right brace is now removed. New Quby comments are ended with a new line. Insert a stop at the new line choice in the Comment function. Replace the left bracket indicator for a PT comment with a percent symbol to indicate a Quby comment.

Alter:

```

    | '}' :
    >
    | lNewLine:
    .pNewLine
    | lEndFile:
    % comment must be closed before EOF

```

```
#eCommentEOF
.pEndFile
>
| *:          % comment may contain any other character
?
```

To:

```
| lNewLine:
    .pNewLine
    >
    | lEndFile:          % comment must be closed before EOF
    #eCommentEOF
    .pEndFile
    >
    | *:          % comment may contain any other character
    ?
```

**Why:** Quby does not use brackets as identifiers for comments, instead it uses percent symbols.