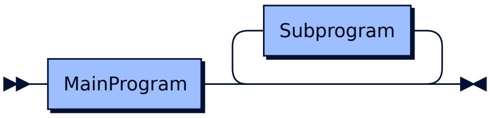
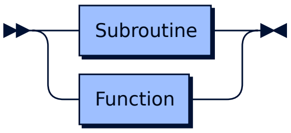
ExecutableProgram:



```
ExecutableProgram
         ::= MainProgram Subprogram*
```

no references

Subprogram:

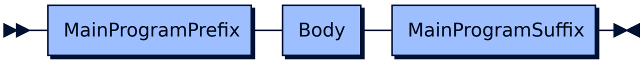


```
Subprogram
         ::= Subroutine
           | Function
```

referenced by:

- ExecutableProgram

MainProgram:

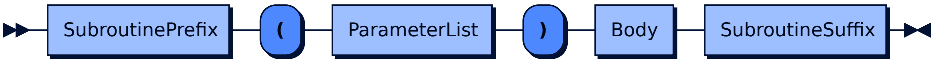


```
MainProgram
         ::= MainProgramPrefix Body MainProgramSuffix
```

referenced by:

- ExecutableProgram

Subroutine:

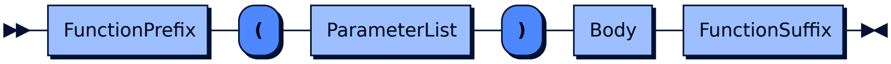


```
Subroutine
         ::= SubroutinePrefix '(' ParameterList ')' Body SubroutineSuffix
```

referenced by:

- Subprogram

Function:

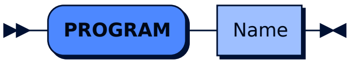


```
Function ::= FunctionPrefix '(' ParameterList ')' Body FunctionSuffix
```

referenced by:

- Subprogram

MainProgramPrefix:

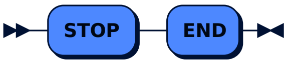


```
MainProgramPrefix
         ::= 'PROGRAM' Name
```

referenced by:

- MainProgram

MainProgramSuffix:

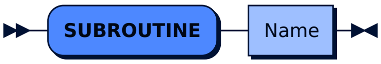


```
MainProgramSuffix
         ::= 'STOP' 'END'
```

referenced by:

- MainProgram

SubroutinePrefix:

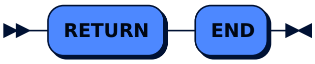


```
SubroutinePrefix
         ::= 'SUBROUTINE' Name
```

referenced by:

- Subroutine

SubroutineSuffix:

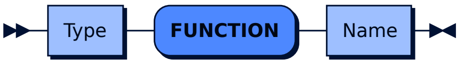


```
SubroutineSuffix
         ::= 'RETURN' 'END'
```

referenced by:

- Subroutine

FunctionPrefix:

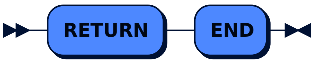


```
FunctionPrefix
         ::= Type 'FUNCTION' Name
```

referenced by:

- Function

FunctionSuffix:

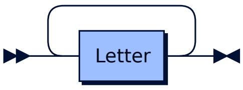


```
  FunctionSuffix
           ::= 'RETURN' 'END'
```

referenced by:

- Function

Name:

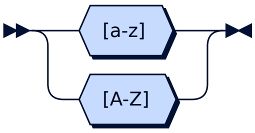


```
  Name     ::= Letter+
```

referenced by:

- CallStatement
- FunctionPrefix
- MainProgramPrefix
- SubroutinePrefix

Letter:

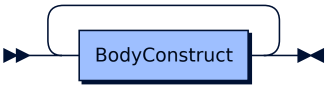


```
  Letter   ::= [a-zA-Z]
```

referenced by:

- Alphanumeric
- Identifier
- Name

Body:

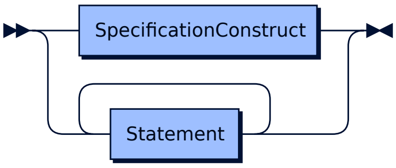


```
  Body     ::= BodyConstruct+
```

referenced by:

- Function
- MainProgram
- Subroutine

BodyConstruct:

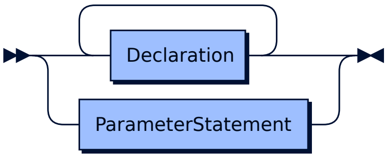


```
BodyConstruct
         ::= SpecificationConstruct
           | Statement+
```

referenced by:

- Body

SpecificationConstruct:

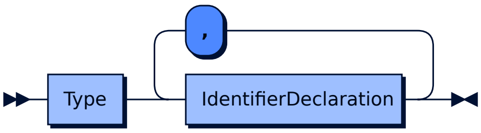


```
SpecificationConstruct
         ::= Declaration+
           | ParameterStatement
```

referenced by:

- BodyConstruct

Declaration:

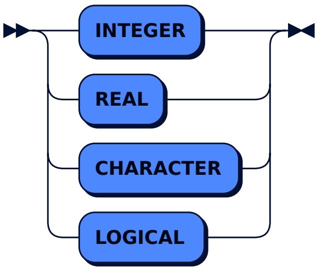


```
Declaration
         ::= Type IdentifierDeclaration ( ',' IdentifierDeclaration )*
```

referenced by:

- SpecificationConstruct

Type:

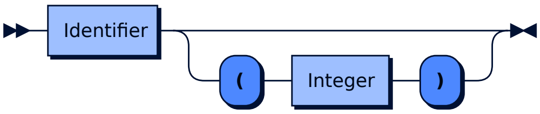


```
Type     ::= 'INTEGER'
           | 'REAL'
           | 'CHARACTER'
           | 'LOGICAL'
```

referenced by:

- Declaration
- FunctionPrefix

IdentifierDeclaration:

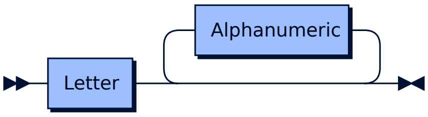


```
IdentifierDeclaration
         ::= Identifier ( '(' Integer ')' )?
```

referenced by:

- Declaration

Identifier:

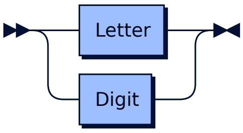


```
Identifier
         ::= Letter Alphanumeric*
```

referenced by:

- AssignmentStatement
- ConstantDefinition
- DoLoopControl
- IdentifierDeclaration
- Term

Alphanumeric:

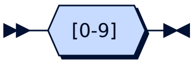


```
Alphanumeric
         ::= Letter
           | Digit
```

referenced by:

- Identifier

Digit:

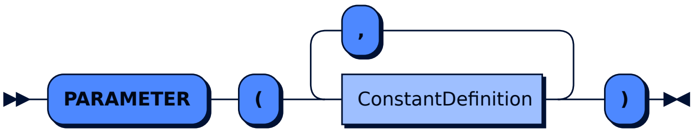


```
Digit    ::= [0-9]
```

referenced by:

- Alphanumeric
- Integer

ParameterStatement:

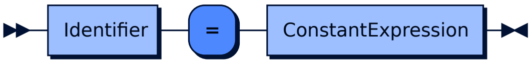


```
ParameterStatement
         ::= 'PARAMETER' '(' ConstantDefinition ( ',' ConstantDefinition )* ')'
```

referenced by:

- SpecificationConstruct

ConstantDefinition:

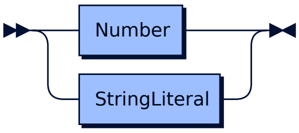


```
ConstantDefinition
         ::= Identifier '=' ConstantExpression
```

referenced by:

- ParameterStatement

ConstantExpression:

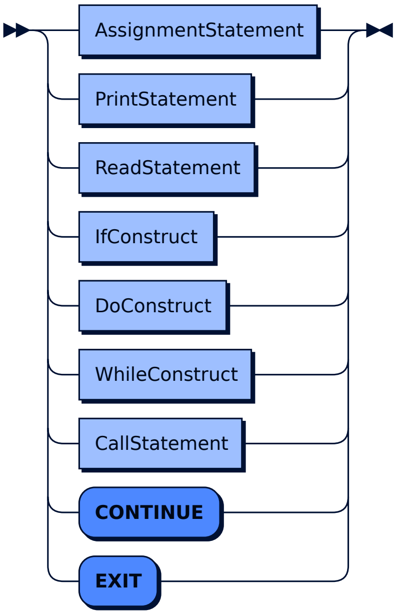


```
ConstantExpression
         ::= Number
           | StringLiteral
```

referenced by:

- ConstantDefinition

Statement:



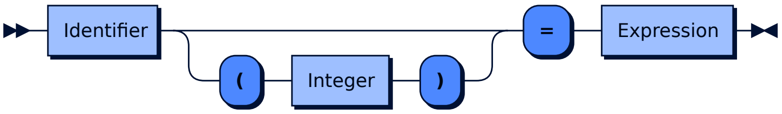
```
Statement
         ::= AssignmentStatement
           | PrintStatement
           | ReadStatement
           | IfConstruct
```

```
              | DoConstruct
              | WhileConstruct
              | CallStatement
              | 'CONTINUE'
              | 'EXIT'
```

referenced by:

- BodyConstruct
- EndDoStatement
- EndWhileStatement
- ThenConstruct

AssignmentStatement:

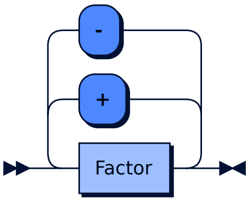


```
AssignmentStatement
         ::= Identifier ( '(' Integer ')' )? '=' Expression
```

referenced by:

- Statement

Expression:

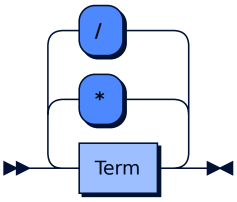


```
Expression
         ::= Factor ( ( '+' | '-' ) Factor )*
```

referenced by:

- AssignmentStatement
- DoLoopControl
- ElseConstruct
- ElseIfStatement
- LogicalExpression
- PrintItem
- Term

Factor:

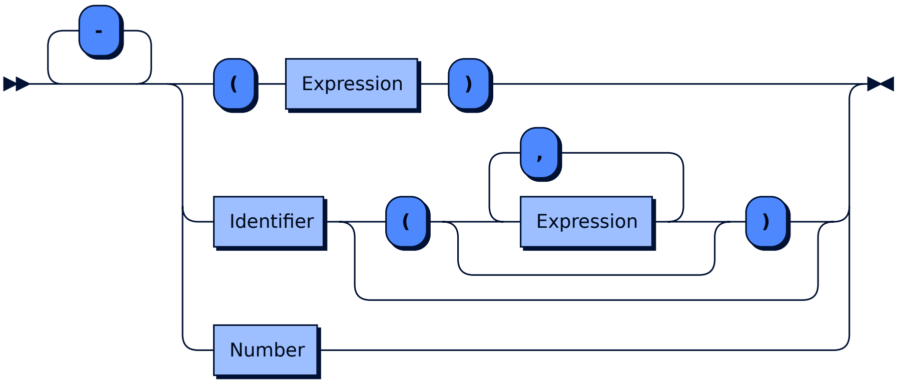


```
Factor   ::= Term ( ( '*' | '/' ) Term )*
```

referenced by:

- Expression

Term:

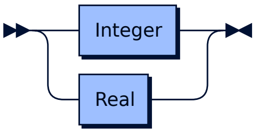


```
Term     ::= '-'* ( '(' Expression ')' | Identifier ( '(' ( Expression ( ',' Expression )* )? ')' )? | Number )
```

referenced by:

- Factor

Number:

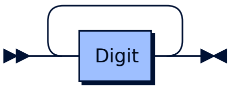


```
Number   ::= Integer
           | Real
```

referenced by:

- ConstantExpression
- Term

Integer:

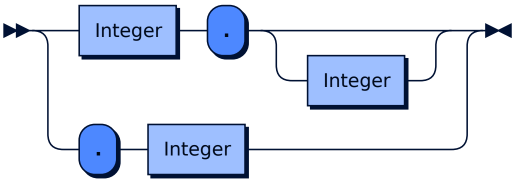


```
Integer  ::= Digit+
```

referenced by:

- AssignmentStatement
- IdentifierDeclaration
- Number
- Real

Real:

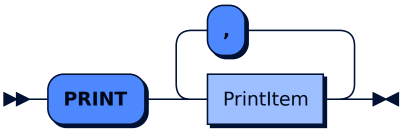


```
Real     ::= Integer '.' Integer?
           | '.' Integer
```

referenced by:

- Number

PrintStatement:

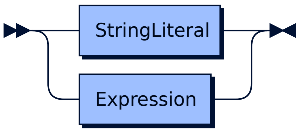


```
  PrintStatement
           ::= 'PRINT' PrintItem ( ',' PrintItem )*
```

referenced by:

- Statement

PrintItem:

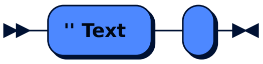


```
  PrintItem
           ::= StringLiteral
             | Expression
```

referenced by:

- PrintStatement

StringLiteral:

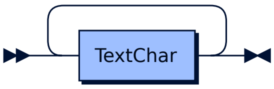


```
  StringLiteral
           ::= "'' Text " ''
```

referenced by:

- ConstantExpression
- PrintItem

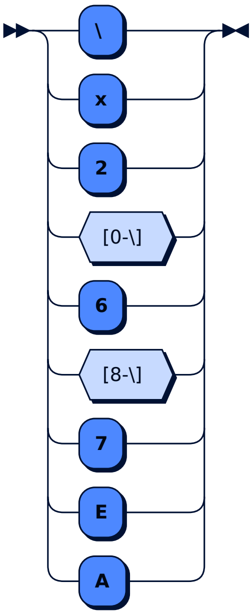
Text:



```
  Text     ::= TextChar+
```

no references

TextChar:

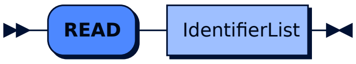


```
TextChar ::= [\x20-\68-\7EA]
```

referenced by:

- Text

ReadStatement:

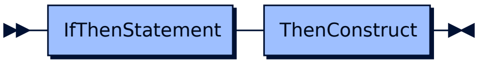


```
ReadStatement
         ::= 'READ' IdentifierList
```

referenced by:

- Statement

IfConstruct:

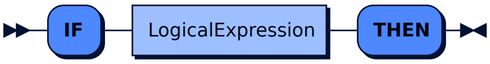


```
IfConstruct
         ::= IfThenStatement ThenConstruct
```

referenced by:

- Statement

IfThenStatement:

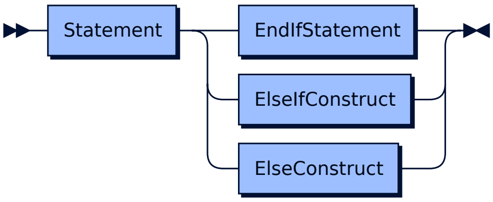


```
IfThenStatement
         ::= 'IF' LogicalExpression 'THEN'
```

referenced by:

- IfConstruct

ThenConstruct:

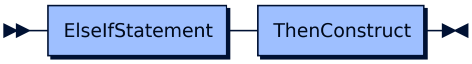


```
ThenConstruct
         ::= Statement ( EndIfStatement | ElseIfConstruct | ElseConstruct )
```

referenced by:

- ElseIfConstruct
- IfConstruct

ElseIfConstruct:

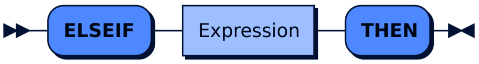


```
ElseIfConstruct
         ::= ElseIfStatement ThenConstruct
```

referenced by:

- ThenConstruct

ElseIfStatement:

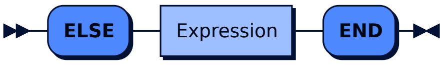


```
ElseIfStatement
         ::= 'ELSEIF' Expression 'THEN'
```

referenced by:

- ElseIfConstruct

ElseConstruct:

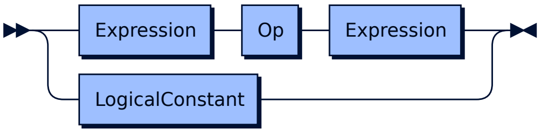


```
ElseConstruct
         ::= 'ELSE' Expression 'END'
```

referenced by:

- ThenConstruct

LogicalExpression:

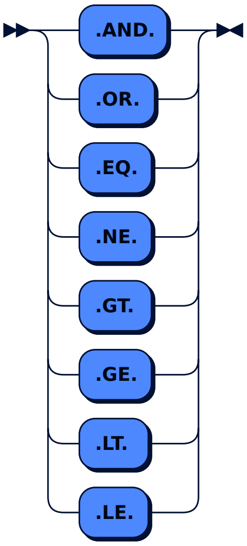


```
LogicalExpression
         ::= Expression Op Expression
           | LogicalConstant
```

referenced by:

- IfThenStatement
- WhileStatement

Op:

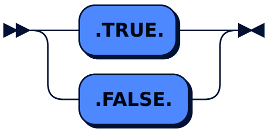


```
Op       ::= '.AND.'
           | '.OR.'
           | '.EQ.'
           | '.NE.'
           | '.GT.'
           | '.GE.'
           | '.LT.'
           | '.LE.'
```

referenced by:

- LogicalExpression

LogicalConstant:

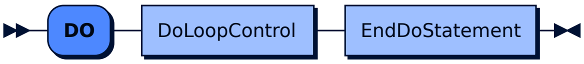


```
LogicalConstant
         ::= '.TRUE.'
           | '.FALSE.'
```

referenced by:

- LogicalExpression

DoConstruct:

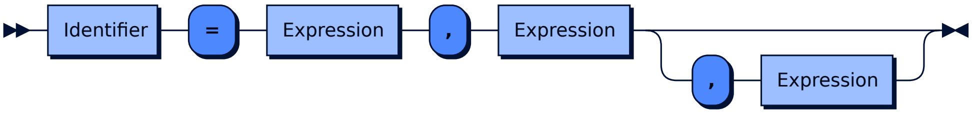


```
DoConstruct
         ::= 'DO' DoLoopControl EndDoStatement
```

referenced by:

- Statement

DoLoopControl:

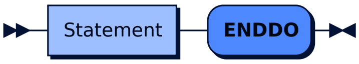


```
DoLoopControl
         ::= Identifier '=' Expression ',' Expression ( ',' Expression )?
```

referenced by:

- DoConstruct

EndDoStatement:

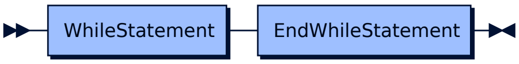


```
EndDoStatement
         ::= Statement 'ENDDO'
```

referenced by:

- DoConstruct

WhileConstruct:

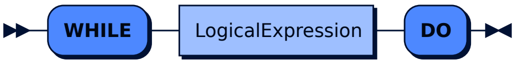


```
WhileConstruct
         ::= WhileStatement EndWhileStatement
```

referenced by:

- Statement

WhileStatement:

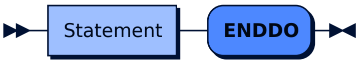


```
WhileStatement
         ::= 'WHILE' LogicalExpression 'DO'
```

referenced by:

- WhileConstruct

EndWhileStatement:

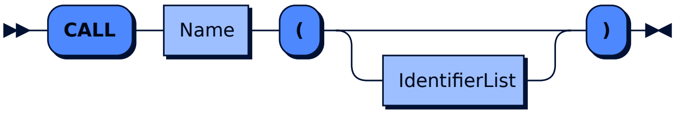


```
EndWhileStatement
         ::= Statement 'ENDDO'
```

referenced by:

- WhileConstruct

CallStatement:



```
CallStatement
         ::= 'CALL' Name '(' IdentifierList? ')'
```

referenced by:

- Statement

... generated by Railroad Diagram Generator