APPENDIX D

Syntax

An Extended Backus–Naur Form (EBNF) specification of the syntax of a programming language consists of a collection of rules or productions collectively called a "grammar" that describe the formation of sentences in the language. Each production consists of a non–terminal symbol and an EBNF expression separated by an equal sign and terminated with a period. The non–terminal symbol is a "meta–identifier" (a syntactic constant denoted by an English word), and the EBNF expression is its definition.

The EBNF expression is composed of zero or more terminal symbols, non–terminal symbols, and other metasymbols summarized in this table:

MetaSymbol	**Meaning**
=	is defined to be
|	alternatively
.	end of production
[*X*]	0 or 1 instance of *X*
{ *X* }	0 or more instances of *X*
(*X* | *Y*)	a grouping: either *X* or *Y*
"XYZ"	the terminal symbol XYZ
Meta–Identifier	the non–terminal symbol *MetaIdentifier*

As an example, EBNF can be used to define its own syntax.

```
Syntax          = { Production } .
Production      = NonTerminal "=" Expression "." .
Expression      = Term [ "|" Term ] .
Term            = Factor { Factor } .
Factor          = NonTerminal | Terminal | "(" Expression ")"
                  "[" Expression "]"  "{" Expression "}" .
Terminal        = “““” Character { Character } “”””  .
NonTerminal     = Letter { Letter | Digit } .
```