bcScript Specification

v 0.1

Contents

0 **Introduction**  
  
 0.1 Design Philosophy

0.2 Structure  
 0.2.1 Analysis  
 0.2.1.1 Lexer  
 0.2.1.2 Parser  
 0.2.2 Intermediate Generation  
 0.2.2.1 Bytecode  
 0.2.3 Execution  
 0.2.3.1 Virtual Machine

0.3

1 **Language Specification**

1.1 Declaration  
 1.1.1 Function  
 1.1.2 Variable  
 1.1.3 Type  
 1.1.3.1 Common Types   
 1.1.4 Compound Type  
 1.1.4.1   
 1.1.5 Namespaces

1.2 Assignment  
 1.2.1 Variable

1.3 Control Flow  
 1.3.1 If Statements  
 1.3.2 Loop Statements  
 1.3.3 Switch Statements

1.4 Data Types  
 1.4.1 Core Types  
 1.4.2 Dynamic Types  
 1.4.3 User Defined Types

2 **API Specification**

3

**0 Introduction  
 0.1 Design Philosophy**

bcScript is essentially a C clone with some dyanamic data types and additional control structures, providing a means to easily pass data between application and script. The API and accompanying Virtual Machine provide an enviroment for the execution of scripts and bytecode, with attempts to keep the design simplistic for newcomers, whilst providing some degree of power and speed to experienced coders. Constantly finding myself requiring a better way to handle the data in my data in programs,

**0.2 Structure**

The library provides two main functions; the conversion of bcScript code to Bytecode, and the execution of that Bytecode. Within the conversion process, the original source code is checked for errors then

if statement

loop statement