Aves Documentation

It is now necessary to consolidate all the documentation in to two multipart manuals

1. Hardware Manual - Covering the design of all of the Aves models
2. Software Manual - Covering the design, use and implementation of the kingfisher programmin language and tools

The following section provides guidelines. to follow to produce consistent documentation

Documentation Structure Guidelines

The Kingfisher documentation format follows a structured approach with specific sections and formatting requirements.

Opening Section

The document begins with an introduction and smoothly transitions to the main subject matter, written in a narrative style.

Constructor documentation follows on directly after the opening section using a two-column table format, sorted alphabetically by Constructor with a bold title as shown below.

All narrative should be kept ahead of the tables and examples, the exception being the use of [NOTE] if required after a table or code.

Multiple Tables and Examples

In some circumstances additional subheadings are required, but this should follow the same patterm with no lower than a level 4 heading. that is a section heading with narrative followed byt table(s) and examples.

Table 1. Constructor Definitions

Constructor	**Description**
Constructor name with stack signature	Description

Methods and definition follow on directly from constructors using a similar style to constructors.

Table 2. Available Methods

Operation	**Description**
Operation name with stack signature	Detailed explanation

Code examples follow on from available operations in bold titles, unbreakable source code blocks. Examples should be the last item in section.

Basic Usage Example

```
code goes here
```

Section Hierarchy

Section headings should follow these levels: * Level 1: = * Level 2: == * Level 3: === * Level 4: ==== * Beyond level 4: Use titles

Writing Style

The documentation should:

- Use British English spelling and conventions. Continue to use program rather than programme in the context of computer software
- Employ narrative form rather than bullet points where possible
- Avoid short, colon-terminated sentences
- Maintain a professional and comprehensive tone throughout

Kingfisher Programming Language

located in ~/projects/kingfisher/doc

- kpl-ref-doc.adoc = Abstract, Foreward, Contents
  - part1.adoc = Getting Started
  - part2.adoc = Language Fundamentals
  - part3.adoc = Assembly Language and System Fundamentals
  - part4.adoc = Runtime Features
  - part5.adoc = kingfisher development
  - part6.adoc = Reference Sections

Table of Contents

8. Stack-Based Programming Fundamentals Understanding Postfix Notation
9. Number System
10. Error Management Error Types and Handling
11. Stack Signatures Type Categories
12. The Parameter Stack Stack Signatures Stack Operators
13. Arithmetic Operators Bitwise Operators General Operators
14. Built-in Types Scalar Types
15. Constants Declaring Constants
16. Variables Variable Construction Variable Operations
17. Collections
18. Arrays Array Slices
19. Strings String Slices
20. StrArrays StrArrSlices
21. Definitions

II: Language Fundamentals

1. Program Organisation Bootstrap Vocabularies and Chains Modules Aliases Scope and Lifetime Rules Error Handling
2. Type Definitions and Linked Methods Type Definitions Linked Methods Type Field Vocabulary Datasets
3. Control Flow
4. Boolean Operations Conditional Operations Error Handling Branching Iterators and Range

III: Assembly Language and System Fundamentals

1. Assembly Language Design Philosophy Assembly Language Core Instruction Set Support Assembly Directives Assembler Macro Processing
2. System Data Structures
3. Interrupts Interrupt Handling System

IV: Runtime Features

1. Introduction
2. I/O Subsystem Console I/O Stream Definitions File I/O Operating System Management
3. System Primitives
4. Memory Management Dictionary Operations Heap Operations

V: Kingfisher Development

1. The Software Development Lifecycle (SDLC) Development Environment Talon IDE Example Application Another Example Application

VI: Reference Sections

1. Historical Context
2. System References Error Message Reference
3. Technical Background System Architecture139 Development References Perch CLI Reference Talon IDE Reference
4. Fundamental System Structures Dictionary Entries Type System The Dictionary Dictionary Concepts Dictionary Reference Dictionary Management Dictionary Structure Memory Management Compile-time Features

Aves Hardware: Reference manual

located in ~/projects/aves/doc

aves.adoc → one large document

WinCUPL PLD Files

- aves-io-1.pld
- aves-mem-1.pld
- aves-mem-2.pld
- aves-video-1.pld
- aves-video-2.pld

ASB Protocol

- Foundational Influences → introduction.adoc
- Physical Layer → physical_layer.adoc
- Data Link Layer → data_link.adoc
- Frame Format → frame_format.adoc
- Bus Protocol → bus_protocol.adoc
- Transport Protocol → transport.adoc
- Performance Analysis → performance.adoc

Design Notes

- CRTC Design Notes → crtc-timing.adoc
- VIC Design Notes → vic-timing.adoc