To: J3
From: Malcolm Cohen
Subject: Editorial replacement for GET_COMMAND_ARGUMENT.
Date: 2012/02/08

1. Introduction

This document shows the results of editing GET_COMMAND_ARGUMENT according to 12-127.

13.7.66 GET_COMMAND_ARGUMENT (NUMBER [, VALUE, LENGTH, STATUS])

Description. Program invocation argument.

Class. Subroutine.

Arguments.

NUMBER shall be a default integer scalar. It is an INTENT (IN) argument that specifies the number of the command argument that the other arguments give information about.

Command argument 0 always exists, and is the command name by which the program was invoked if the processor has such a concept; otherwise, the value of command argument 0 is processor dependent. The remaining command arguments are numbered consecutively from 1 to the argument count in an order determined by the processor.

VALUE (optional) shall be a default character scalar. It is an INTENT (OUT) argument. If the command argument specified by NUMBER exists, its value is assigned to VALUE; otherwise, VALUE is assigned all blanks.

LENGTH (optional) shall be a default integer scalar. It is an INTENT (OUT) argument. If the command argument specified by NUMBER exists, its significant length is assigned to LENGTH; otherwise, LENGTH is assigned the value 0. It is processor dependent whether the significant length includes trailing blanks. This length does not consider any possible truncation or padding in assigning the command argument value to the VALUE argument; in fact the VALUE argument need not even be present.

STATUS (optional) shall be a default integer scalar. It is an INTENT (OUT) argument. If NUMBER is less than zero or greater than the argument count that would be returned by COMMAND_ARGUMENT_COUNT, or command argument retrieval fails, STATUS is assigned a processor-dependent positive value. Otherwise, if VALUE is present and has a length less than the significant length of the specified command argument, it is assigned the value -1. Otherwise it is assigned the value 0.

Example.

```
PROGRAM echo
  INTEGER :: i
  CHARACTER :: command*32, arg*128
  CALL get_command_argument(0, command)
  WRITE (*,*) "Command name is: ", command
  DO i = 1, command_argument_count()
    CALL get_command_argument(i, arg)
    WRITE (*,*) "Argument ", i, " is ", arg
  END DO
END PROGRAM echo
```