TextUtilPkg Package User Guide

User Guide for Release 2024.11

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 TextUtilPkg Overview

TextUtilPkg provides basic string handling and reading utilities.

2 Character Tests

TextUtilPkg provides the following character test functions:

```
function IsUpper  (constant Char : character ) return boolean ;
function IsLower  (constant Char : character ) return boolean ;
function IsWhiteSpace (constant Char : character ) return boolean ;
function IsHex (constant Char : character ) return boolean ;
function IsHexOrStdLogic (constant Char : character ) return boolean ;
function IsNumber (constant Char : character ) return boolean ;
function IsNumber (Name : string ) return boolean ;
function isstd_logic (constant Char : character ) return boolean ;
```

3 Text Case Conversions

TextUtilPkg provides the following text conversion test functions:

```
function to_lower (constant Char : character ) return character ;
function to_lower (constant Str : string ) return string ;
function to_upper (constant Char : character ) return character ;
function to_upper (constant Str : string ) return string ;
```

4 Space and CRLF handling

TextUtilPkg provides the following functions that remove Space or CRLF:

```
procedure RemoveSpace (variable S : inout string ; variable Len : InOut integer) ;
-- Only needed for non-compliant simulators -- X
procedure RemoveCrLf (S : string ; variable Len : out integer) ;
function  RemoveCrLf( S : string ) return string ;
```

5 Array of Std_logic to Hex

TextUtilPkg provides the following functions that convert from an array of std_logic to hex. A string of 4 Meta characters, such as U, X, Z, W, and – print as that character. If there is a mixture of Meta characters and other characters a ? is printed.

```
function to_hxstring ( A : std_ulogic_vector) return string ;
function to_hxstring ( A : unsigned) return string ;
function to_hxstring ( A : signed) return string ;
```

6 To_String_Max

Convert integer to string. If the value is integer'high or integer'low, return that instead of the string version of the integer.

```
function to_string_max ( I : integer) return string ;
```

7 Justify

Justify a string value. If Fill character specified use that as the fill character, otherwise, use space.

```
type AlignType is (RIGHT, LEFT, CENTER) ;

function Justify (
  S       : string ;
  Amount  : natural ;
  Align   : AlignType := LEFT
) return string ;

function Justify (
  S       : string ;
  Fill    : character ;
  Amount  : natural ;
  Align   : AlignType := LEFT
) return string ;
```

8 File Exists

FileExists returns true if a file exists.

```
impure function FileExists(FileName : string) return boolean ;
```

9 Read Utilities

9.1 GetLine

GetLine reads lines from a file. Optionally will skip over empty lines. Returns EndOfFile as true if there are no more lines in the file.

```
procedure GetLine(
  File     FileId           :       text ;
  variable Buf              : inout line ;
  variable LineCount        : inout integer ;
  variable EndOfFile        : out   boolean ;
  constant IgnoreEmptyLines : in    boolean
) ;
```

9.2 Skip White Space

SkipWhiteSpace discards leading whitespace characters by reading them.

```
procedure SkipWhiteSpace (
  variable L     : InOut line ;
  variable Empty : out   boolean
) ;
procedure SkipWhiteSpace (variable L : InOut line) ;
```

9.3 Empty Line Checks

IsWhiteSpaceOrEmpty returns Empty as true if the line is empty or only contains white space. Is a procedure due to VHDL limitations (before 2019).

```
    procedure IsWhiteSpaceOrEmpty (
      variable L                : InOut  line ;
      variable Empty            : Out    boolean
    ) ;
```

EmptyOrCommentLine indicates whether a line is empty or whether the read process is currently in the middle of a multiple line comment.

```
    procedure EmptyOrCommentLine (
      variable L                : InOut  line ;
      variable Empty            : InOut  boolean ;
      variable MultiLineComment : inout  boolean
    ) ;
```

9.4 Finding a Delimiter

ReadUntilFindDelimiterOrEOL read a Name until a delimiter or EOL is found.

```
      procedure ReadUntilDelimiterOrEOL(
        variable L         : InOut line ;
        variable Name      : InOut line ;
        constant Delimiter : In    character ;
        variable ReadValid : Out   boolean
      ) ;
```

FindDelimiter read the next character. If it is the specified delimiter, return Found as TRUE. If the delimiter is not SPACE, then SkipWhiteSpace before searching for the delimiter.

```
      procedure FindDelimiter(
        variable L                : InOut line ;
        constant Delimiter        : In    character ;
        variable Found            : Out   boolean
      ) ;
```

9.5 ReadHexToken

ReadHexToken reads hex values upto Result'length. Less is OK. Does not skip white space.

```
      procedure ReadHexToken (
        variable L      : InOut line ;
        variable Result : Out   std_logic_vector ;
        variable StrLen : Out   integer
      ) ;
```

9.6 ReadBinaryToken

ReadBinaryToken reads binary values upto Result'length. Less is OK. Does not skip white space.

```
procedure ReadBinaryToken (
  variable L      : InOut line ;
  variable Result : Out   std_logic_vector ;
  variable StrLen : Out   integer
) ;
```

9.7 Sread_C

Sread_c is an sread procedure for Xilinx tools.

```
procedure sread_c (
  variable L          : InOut line ;
  variable Name       : Out   string ;
  variable NameLength : Out   integer
) ;
```

10 Compiling TextUtilPkg and Friends

See OSVVM_release_notes.pdf for the current compilation directions. Rather than referencing individual packages, we recommend using the context declaration:

```
library OSVVM ;
  context osvvm.OsvvmContext ;
```

11 About TextUtilPkg

TextUtilPkg was developed and is maintained by Jim Lewis of SynthWorks VHDL Training. Prior to its release to OSVVM it was used in SynthWorks' VHDL classes.

Please support our effort in supporting the OSVVM library of packages by purchasing your VHDL training from SynthWorks.

TextUtilPkg is released under the Apache open source license. It is free (both to download and use - there are no license fees). You can download it from osvvm.org or from our development area on GitHub (https://github.com/OSVVM/OSVVM).

If you add features to the package, please donate them back under the same license as candidates to be added to the standard version of the package. If you need features, be sure to contact us. I blog about the packages at http://www.synthworks.com/blog. We also support the OSVVM user community and blogs through http://www.osvvm.org.

Find any innovative usage for the package? Let us know, you can blog about it at osvvm.org.

12 Future Work

TextUtilPkg.vhd is a work in progress and will be updated from time to time.

Caution, undocumented items are experimental and may be removed in a future version.

13 About the Author - Jim Lewis

Jim Lewis, the founder of SynthWorks, has thirty plus years of design, teaching, and problem solving experience. In addition to working as a Principal Trainer for SynthWorks, Mr Lewis has done ASIC and FPGA design, custom model development, and consulting.

Mr. Lewis is chair of the IEEE 1076 VHDL Working Group (VASG) and is the primary developer of the Open Source VHDL Verification Methodology (OSVVM.org) packages. Neither of these activities generate revenue. Please support our volunteer efforts by buying your VHDL training from SynthWorks.

If you find bugs these packages or would like to request enhancements, you can reach me at jim@synthworks.com.