

# FortranUtilities

1.0

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Modules Index</b>	<b>3</b>
2.1	Modules List . . . . .	3
<b>3</b>	<b>Module Documentation</b>	<b>5</b>
3.1	prec_m Module Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	strings_m Module Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Function/Subroutine Documentation . . . . .	6
3.2.2.1	char2int() . . . . .	7
3.2.2.2	char2real() . . . . .	7
3.2.2.3	count_digits_integer() . . . . .	8
3.2.2.4	endswith() . . . . .	8
3.2.2.5	int2char() . . . . .	9
3.2.2.6	int2char00000() . . . . .	9
3.2.2.7	splitstr() . . . . .	10
3.2.2.8	startswith() . . . . .	10
	<b>Index</b>	<b>13</b>



# Chapter 1

## Main Page

FortranUtilities is a collection of simple functions and subroutines for Fortran programs.

Only functions to manipulate strings are included so far. See file refman.pdf in folder doc for specifications of the different functions. More functions for different tasks will be included in the future.

### Downloading.

The latest version of this library can be obtained from the GitHub repository located [here](#).

### Compilation

1. Check that you have cmake version 3.10 or newer, and a Fortran compiler compatible with Fortran 2008.
2. Download the files and enter directory FortranUtilities. Then create a build directory and enter that directory:

```
mkdir build
cd build
```

3. Type:

```
cmake ..
```

Optionally you can select an install directory by typing:

```
cmake .. -DCMAKE_INSTALL_PREFIX=/installation/path/
```

4. Compile:

```
make
```

5. Install:

```
make install
```

A static and and a shared library are created. Test programs for both libraries are created too.

## Usage

To use the library in your programs you first need to use the module of interest, as in the following example:

```
PROGRAM test
USE strings_m
WRITE(*,*) int2char(5)
END PROGRAM test
```

Then you have to link to the library when compiling. For example:

1. To use the static library:

```
gfortran program.f90 -lFortranUtilitiesStatic -L/path/to/the/library/folder -I/path/to/include/folder
```

2. To use the shared library:

```
gfortran program.f90 -lFortranUtilities -L/path/to/the/library/folder -I/path/to/include/folder
```

## Contact

Developed by Emilio Castro.

Create a Issue in GitHub if you have any suggestion, comment, enhancement, bug, etc.

## License

See LICENSE file for more information on using and distributing these files.

## Chapter 2

# Modules Index

### 2.1 Modules List

Here is a list of all documented modules with brief descriptions:

<a href="#">prec_m</a>	Precision parameters to use in Fortran programs . . . . .	<a href="#">5</a>
<a href="#">strings_m</a>	Useful tools to manipulate strings in Fortran programs . . . . .	<a href="#">6</a>





## Chapter 3

# Module Documentation

### 3.1 prec\_m Module Reference

Precision parameters to use in Fortran programs.

#### Variables

- integer, parameter `sp` = REAL32  
*Kind parameter to specify a real type with a storage size of 32 bits (default).*
- integer, parameter `dp` = REAL64  
*Kind parameter to specify a real type with a storage size of 64 bits.*
- integer, parameter `qp` = REAL128  
*Kind parameter to specify a real type with a storage size of 128 bits.*
- integer, parameter `i8` = INT8  
*Kind parameter to specify an integer type with a storage size of 8 bits.*
- integer, parameter `i16` = INT16  
*Kind parameter to specify an integer type with a storage size of 16 bits.*
- integer, parameter `i32` = INT32  
*Kind parameter to specify an integer type with a storage size of 32 bits (default).*
- integer, parameter `i64` = INT64  
*Kind parameter to specify an integer type with a storage size of 64 bits.*

#### 3.1.1 Detailed Description

Precision parameters to use in Fortran programs.

#### Author

Emilio Castro.

#### Version

1.0.

#### Copyright

See LICENSE file that comes with this distribution.

## 3.2 strings\_m Module Reference

Useful tools to manipulate strings in Fortran programs.

### Functions/Subroutines

- character(len=:) function, allocatable, public [splitstr](#) (str, delimiter, fieldNumber)  
*Splits a string and returns the portion selected by the user.*
- logical function, public [startswith](#) (str, substr)  
*Checks if a string starts with a given substring.*
- logical function, public [endswith](#) (str, substr)  
*Checks if a string ends with a given substring.*
- character(len=:) function, allocatable, public [int2char](#) (integ)  
*Converts an integer variable into a character variable. Useful to open files named sequentially.*
- character(len=:) function, allocatable, public [int2char00000](#) (integ, total\_length)  
*Converts an integer variable into a character variable, filling with leading zeros up to the limit imposed by the user. Useful to open files named sequentially with leading zeros in the name.*
- integer function, public [count\\_digits\\_integer](#) (integ)  
*Counts the number of digits of an integer, including the - sign in case it is a negative value.*
- integer function, public [char2int](#) (charac)  
*Converts a character string into an integer.*
- real(kind=8) function, public [char2real](#) (charac)  
*Converts a character string into a real with double precision.*

### Variables

- integer, parameter [exit\\_error\\_code](#) = 10  
*Error code issued by all functions in module Strings\_M.*

### 3.2.1 Detailed Description

Useful tools to manipulate strings in Fortran programs.

#### Author

Emilio Castro.

#### Version

1.0.

#### Copyright

See LICENSE file that comes with this distribution.

### 3.2.2 Function/Subroutine Documentation

### 3.2.2.1 char2int()

```
integer function, public strings_m::char2int (  
    character(len=*), intent(in) charac )
```

Converts a character string into an integer.

#### Author

Emilio Castro.

#### Version

1.0.

#### Parameters

<i>charac</i>	String to convert to integer.
---------------	-------------------------------

#### Returns

Integer containing the number of the input string.

### 3.2.2.2 char2real()

```
real(kind=8) function, public strings_m::char2real (  
    character(len=*), intent(in) charac )
```

Converts a character string into a real with double precision.

#### Author

Emilio Castro.

#### Version

1.0.

#### Parameters

<i>charac</i>	String to convert to double precision real.
---------------	---

#### Returns

Double preicision containing the number of the input string.

### 3.2.2.3 `count_digits_integer()`

```
integer function, public strings_m::count_digits_integer (
    integer, value integ )
```

Counts the number of digits of an integer, including the - sign in case it is a negative value.

#### Author

Emilio Castro.

#### Version

1.0.

#### Parameters

<i>integ</i>	Integer number whose digits are to be counted.
--------------	--

#### Returns

The number of digits of the input number.

### 3.2.2.4 `endswith()`

```
logical function, public strings_m::endswith (
    character(len=*), intent(in) str,
    character(len=*), intent(in) substr )
```

Checks if a string ends with a given substring.

#### Author

Emilio Castro.

#### Version

1.0.

#### Parameters

<i>str</i>	String that the user wants to check how it ends.
<i>substr</i>	Substring to search to check if str ends with it.

**Returns**

True if the string ends with the substring and False otherwise. If substr is empty it returns True.

**3.2.2.5 int2char()**

```
character(len=:) function, allocatable, public strings_m::int2char (
    integer, intent(in) integ )
```

Converts an integer variable into a character variable. Useful to open files named sequentially.

**Author**

Emilio Castro.

**Version**

1.0.

**Parameters**

<i>integ</i>	Integer number to convert.
--------------	----------------------------

**Returns**

Character containing the number.

**3.2.2.6 int2char00000()**

```
character(len=:) function, allocatable, public strings_m::int2char00000 (
    integer, intent(in) integ,
    integer, intent(in) total_length )
```

Converts an integer variable into a character variable, filling with leading zeros up to the limit imposed by the user. Useful to open files named sequentially with leading zeros in the name.

**Author**

Emilio Castro.

**Version**

1.0.

**Parameters**

<i>integ</i>	Integer number to convert. This number MUST be positive.
<i>total_length</i>	Number of digits to use, including zeros.

**Returns**

Character containing the number.

**3.2.2.7 splitstr()**

```
character(len=:) function, allocatable, public strings_m::splitstr (  
    character(len=*), intent(in) str,  
    character(len=*), intent(in), optional delimiter,  
    integer, intent(in), optional fieldNumber )
```

Splits a string and returns the portion selected by the user.

**Author**

Emilio Castro.

**Version**

1.0.

**Parameters**

<i>str</i>	String that the user wants to split.
<i>delimiter</i>	String that the users wants to use as a delimiter for splitting. Optional parameter. Default is Space.
<i>fieldNumber</i>	Integer indicating which of the divisions to return. Optional parameter. Default is the first part obtained.

**Returns**

A string with the selected part of str. If the fieldNumber does not exists or if the delimiter does not exists it returns an empty string.

**3.2.2.8 startswith()**

```
logical function, public strings_m::startswith (  
    character(len=*), intent(in) str,  
    character(len=*), intent(in) substr )
```

Checks if a string starts with a given substring.

**Author**

Emilio Castro.

**Version**

1.0.

**Parameters**

<i>str</i>	String that the user wants to check how it starts.
<i>substr</i>	Substring to search to check if str starts with it.

**Returns**

True if the string starts with the substring and False otherwise. If substr is empty it returns True.





# Index

- char2int
  - strings\_m, [6](#)
- char2real
  - strings\_m, [7](#)
- count\_digits\_integer
  - strings\_m, [7](#)
- endswith
  - strings\_m, [8](#)
- int2char
  - strings\_m, [9](#)
- int2char00000
  - strings\_m, [9](#)
- prec\_m, [5](#)
- splitstr
  - strings\_m, [10](#)
- startswith
  - strings\_m, [10](#)
- strings\_m, [6](#)
  - char2int, [6](#)
  - char2real, [7](#)
  - count\_digits\_integer, [7](#)
  - endswith, [8](#)
  - int2char, [9](#)
  - int2char00000, [9](#)
  - splitstr, [10](#)
  - startswith, [10](#)