

C

Generated by Doxygen 1.9.1



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 data_methods Struct Reference	5
3.2 file_handler Struct Reference	5
3.3 file_methods Struct Reference	5
3.4 list_handler Struct Reference	6
3.5 list_methods Struct Reference	6
3.5.1 Member Data Documentation	6
3.5.1.1 peekn	6
3.6 object_methods Struct Reference	7
3.7 string_methods Struct Reference	7
3.8 string_pointer_methods Struct Reference	8
3.9 string_reference Struct Reference	8
3.10 user_methods Struct Reference	8
3.11 vm_type Struct Reference	8
<b>4 File Documentation</b>	<b>9</b>
4.1 src/pointer/types/data/v1/data_v1.h File Reference	9
4.1.1 Detailed Description	10
4.2 src/pointer/types/file/v1/file_v1.h File Reference	10
4.2.1 Detailed Description	11
4.2.2 Variable Documentation	11
4.2.2.1 file_methods_definition	11
4.3 src/pointer/types/list/v1/list_v1.h File Reference	12
4.3.1 Detailed Description	12
4.4 src/pointer/types/object/v1/object_v1.h File Reference	13
4.4.1 Detailed Description	13
4.5 src/pointer/types/string/v1/string_v1.h File Reference	14
4.5.1 Detailed Description	14
4.6 src/pointer/types/string_pointer/v1/string_pointer_v1.h File Reference	15
4.6.1 Detailed Description	15
4.7 src/pointer/types/user/v1/user_v1.h File Reference	15
4.7.1 Detailed Description	16
<b>Index</b>	<b>17</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">data_methods</a>	5
<a href="#">file_handler</a>	5
<a href="#">file_methods</a>	5
<a href="#">list_handler</a>	6
<a href="#">list_methods</a>	6
<a href="#">object_methods</a>	7
<a href="#">string_methods</a>	7
<a href="#">string_pointer_methods</a>	8
<a href="#">string_reference</a>	8
<a href="#">user_methods</a>	8
<a href="#">vm_type</a>	8



## Chapter 2

# File Index

### 2.1 File List

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

src/pointer/types/ <b>types.h</b> . . . . .	??
src/pointer/types/data/v1/ <a href="#">data_v1.h</a>	
C API / data . . . . .	9
src/pointer/types/file/v1/ <a href="#">file_v1.h</a>	
C API / file . . . . .	10
src/pointer/types/list/v1/ <a href="#">list_v1.h</a>	
C API / list . . . . .	12
src/pointer/types/object/v1/ <a href="#">object_v1.h</a>	
C API / object . . . . .	13
src/pointer/types/string/v1/ <a href="#">string_v1.h</a>	
C API / string . . . . .	14
src/pointer/types/string_pointer/v1/ <a href="#">string_pointer_v1.h</a>	
C API / string pointer . . . . .	15
src/pointer/types/user/v1/ <a href="#">user_v1.h</a>	
C API / user . . . . .	15
src/pointer/types/v1/ <b>types_v1.h</b> . . . . .	??





## Chapter 3

# Class Documentation

### 3.1 data\_methods Struct Reference

#### Public Attributes

- u64(\* **alloc** )(u64 size)
- void(\* **free** )(u64 ptr)
- void(\*(\* **unsafe** )(u64 ptr)
- u64(\* **size** )(u64 ptr)

The documentation for this struct was generated from the following file:

- src/pointer/types/data/v1/[data\\_v1.h](#)

### 3.2 file\_handler Struct Reference

#### Public Attributes

- FILE \* **file**

The documentation for this struct was generated from the following file:

- src/pointer/types/file/v1/[file\\_v1.c](#)

### 3.3 file\_methods Struct Reference

#### Public Attributes

- u64(\* **alloc** )(u64 file\_path\_ptr, u64 mode\_ptr)
- void(\* **free** )(u64 ptr)
- u64(\* **data** )(u64 ptr)

The documentation for this struct was generated from the following file:

- src/pointer/types/file/v1/[file\\_v1.h](#)

## 3.4 list\_handler Struct Reference

### Public Attributes

- u64 **size**
- struct list\_data \* **list**

The documentation for this struct was generated from the following files:

- src/pointer/types/list/v1/list\_v1.c
- src/pointer/types/string/v1/string\_v1.c

## 3.5 list\_methods Struct Reference

### Public Attributes

- u64(\* **alloc** )(void)
- void(\* **free** )(u64 ptr)
- u64(\* **peek** )(u64 list\_ptr)
- u64(\* **peekn** )(u64 list\_ptr, u64 nelements)
- u64(\* **pop** )(u64 list\_ptr)
- u64(\* **popn** )(u64 list\_ptr, u64 nelements)
- void(\* **push** )(u64 list\_ptr, u64 ptr)
- u64(\* **size** )(u64 ptr)
- void(\* **release** )(u64 ptr)

### 3.5.1 Member Data Documentation

#### 3.5.1.1 peekn

```
u64(* list_methods::peekn) (u64 list_ptr, u64 nelements)
```

returns a list of picked items

#### Parameters

<i>list_ptr</i>	a list to get items from
<i>nelements</i>	number of elements to peek from the list

#### Returns

returns new list which contains n elements from the list in reverse order

The documentation for this struct was generated from the following file:

- [src/pointer/types/list/v1/list\\_v1.h](#)

## 3.6 object\_methods Struct Reference

### Public Attributes

- u64(\* **alloc** )(u64 size)
- void(\* **free** )(u64 ptr)
- void(\* **unsafe** )(u64 ptr)
- u64(\* **load** )(const void \*data, u64 size)
- u64(\* **size** )(u64 ptr)

The documentation for this struct was generated from the following file:

- [src/pointer/types/object/v1/object\\_v1.h](#)

## 3.7 string\_methods Struct Reference

### Public Attributes

- void(\* **free** )(u64 ptr)
- u64(\* **copy** )(u64 ptr)
- void(\* **strcpy** )(u64 dest, u64 src)
- void(\* **strcat** )(u64 dest, u64 src)
- u64(\* **strchr** )(u64 src\_ptr, u64 match\_ptr)
- u64(\* **strchr** )(u64 src\_ptr, u64 match\_ptr)
- u64(\* **match** )(u64 src\_ptr, u64 match\_ptr)
- u64(\* **offset** )(u64 src\_ptr, u64 match\_ptr)
- u64(\* **load** )(const char \*data)
- void(\* **put\_char** )(u64 ptr, char value)
- char(\* **unsafe** )(u64 ptr)
- u64(\* **size** )(u64 ptr)
- u64(\* **lessthan** )(u64 src, u64 dest)
- u64(\* **greaterthan** )(u64 src, u64 dest)
- u64(\* **equals** )(u64 src, u64 dest)
- u64(\* **compare** )(u64 src, u64 dest)
- u64(\* **left** )(u64 src, u64 offset)
- u64(\* **strncpy** )(u64 src, u64 nbytes)
- u64(\* **left\_strncpy** )(u64 src, u64 offset)
- u64(\* **right** )(u64 src, u64 offset)
- u64(\* **move\_left** )(u64 src, u64 nbytes)
- u64(\* **move\_right** )(u64 src, u64 nbytes)
- u64(\* **strcmp** )(u64 src, u64 dest)

The documentation for this struct was generated from the following file:

- [src/pointer/types/string/v1/string\\_v1.h](#)

## 3.8 string\_pointer\_methods Struct Reference

### Public Attributes

- void(\* **free** )(u64 ptr)

The documentation for this struct was generated from the following file:

- src/pointer/types/string\_pointer/v1/[string\\_pointer\\_v1.h](#)

## 3.9 string\_reference Struct Reference

### Public Attributes

- u64 **address**
- u64 **offset**

The documentation for this struct was generated from the following file:

- src/pointer/types/string\_pointer/v1/[string\\_pointer\\_v1.h](#)

## 3.10 user\_methods Struct Reference

### Public Attributes

- u64(\* **alloc** )(void)
- void(\* **free** )(u64 ptr)

The documentation for this struct was generated from the following file:

- src/pointer/types/user/v1/[user\\_v1.h](#)

## 3.11 vm\_type Struct Reference

### Public Attributes

- void(\* **free** )(struct pointer \*ptr)

The documentation for this struct was generated from the following file:

- src/pointer/types/types.h

## Chapter 4

# File Documentation

### 4.1 src/pointer/types/data/v1/data\_v1.h File Reference

C API / data.

```
#include "std/common.h"
```

Include dependency graph for data\_v1.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [data\\_methods](#)

## Variables

- const struct [data\\_methods](#) **data\_methods\_definition**

### 4.1.1 Detailed Description

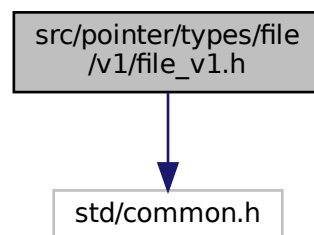
C API / data.

## 4.2 src/pointer/types/file/v1/file\_v1.h File Reference

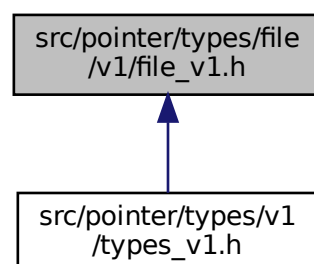
C API / file.

```
#include "std/common.h"
```

Include dependency graph for file\_v1.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [file\\_methods](#)

## Variables

- const struct [file\\_methods](#) [file\\_methods\\_definition](#)

### 4.2.1 Detailed Description

C API / file.



```
struct file_methods {  
    u64 (*alloc)(u64 file_path_ptr, u64 mode_ptr);  
    void (*free)(u64 ptr);  
    u64 (*data)(u64 ptr);  
};
```

### 4.2.2 Variable Documentation

#### 4.2.2.1 file\_methods\_definition

```
const struct file_methods file_methods_definition [extern]
```

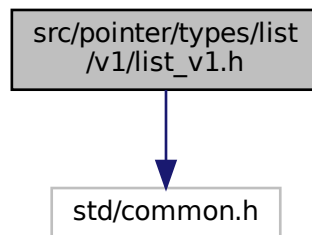
definition (initialization) of [file\\_methods](#) structure

### 4.3 src/pointer/types/list/v1/list\_v1.h File Reference

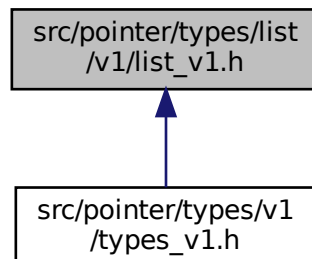
C API / list.

```
#include "std/common.h"
```

Include dependency graph for list\_v1.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- struct [list\\_methods](#)

#### Variables

- const struct [list\\_methods](#) [list\\_methods\\_definition](#)

#### 4.3.1 Detailed Description

C API / list.

structure for organizing [list\\_methods](#) functions

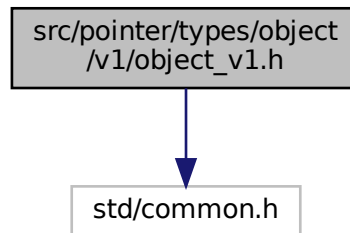


## 4.4 src/pointer/types/object/v1/object\_v1.h File Reference

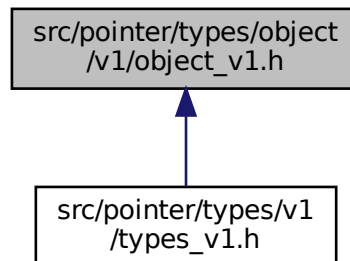
C API / object.

```
#include "std/common.h"
```

Include dependency graph for object\_v1.h:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [object\\_methods](#)

### Variables

- const struct [object\\_methods](#) [object\\_methods\\_definition](#)

#### 4.4.1 Detailed Description

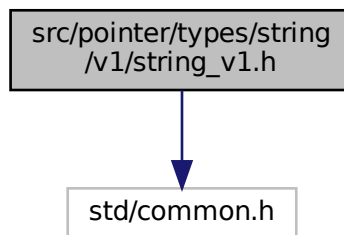
C API / object.

## 4.5 src/pointer/types/string/v1/string\_v1.h File Reference

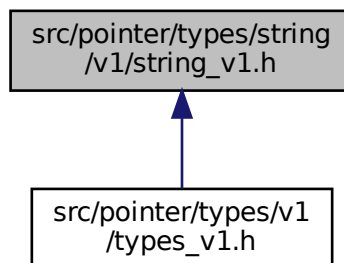
C API / string.

```
#include "std/common.h"
```

Include dependency graph for string\_v1.h:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [string\\_methods](#)

### Variables

- const struct [string\\_methods](#) **string\_methods\_definition**

### 4.5.1 Detailed Description

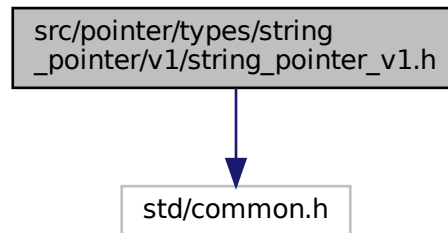
C API / string.

## 4.6 src/pointer/types/string\_pointer/v1/string\_pointer\_v1.h File Reference

C API / string pointer.

```
#include "std/common.h"
```

Include dependency graph for string\_pointer\_v1.h:



### Classes

- struct [string\\_pointer\\_methods](#)
- struct [string\\_reference](#)

### Variables

- const struct [string\\_pointer\\_methods](#) [string\\_pointer\\_methods\\_definition](#)

#### 4.6.1 Detailed Description

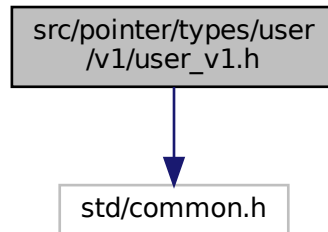
C API / string pointer.

## 4.7 src/pointer/types/user/v1/user\_v1.h File Reference

C API / user.

```
#include "std/common.h"
```

Include dependency graph for user\_v1.h:



## Classes

- struct [user\\_methods](#)

## Variables

- const struct [user\\_methods](#) `user_methods_definition`

### 4.7.1 Detailed Description

C API / user.

# Index

[data\\_methods](#), [5](#)

[file\\_handler](#), [5](#)

[file\\_methods](#), [5](#)

[file\\_methods\\_definition](#)

[file\\_v1.h](#), [11](#)

[file\\_v1.h](#)

[file\\_methods\\_definition](#), [11](#)

[list\\_handler](#), [6](#)

[list\\_methods](#), [6](#)

[peekn](#), [6](#)

[object\\_methods](#), [7](#)

[peekn](#)

[list\\_methods](#), [6](#)

[src/pointer/types/data/v1/data\\_v1.h](#), [9](#)

[src/pointer/types/file/v1/file\\_v1.h](#), [10](#)

[src/pointer/types/list/v1/list\\_v1.h](#), [12](#)

[src/pointer/types/object/v1/object\\_v1.h](#), [13](#)

[src/pointer/types/string/v1/string\\_v1.h](#), [14](#)

[src/pointer/types/string\\_pointer/v1/string\\_pointer\\_v1.h](#),  
[15](#)

[src/pointer/types/user/v1/user\\_v1.h](#), [15](#)

[string\\_methods](#), [7](#)

[string\\_pointer\\_methods](#), [8](#)

[string\\_reference](#), [8](#)

[user\\_methods](#), [8](#)

[vm\\_type](#), [8](#)