std::atexit

Defined in header <cstdlib>		

`int atexit( /*c-atexit-handler*/* func );`
`int atexit( /*atexit-handler*/* func );`
	(1)	(until C++11)

`int atexit( /*c-atexit-handler*/* func ) noexcept;`
`int atexit( /*atexit-handler*/* func ) noexcept;`
		(since C++11)

`extern "C++" using /*atexit-handler*/ = void(); // exposition-only`
`extern "C" using /*c-atexit-handler*/ = void(); // exposition-only`
	(2)	

Registers the function pointed to by `func` to be called on normal program termination (via `std::exit()` or returning from the `main` function)

The functions will be called during the destruction of the static objects, in reverse order: if A was registered before B, then the call to B is made before the call to A. Same applies to the ordering between static object constructors and the calls to `atexit`: see `std::exit`	(until C++11)
The functions may be called concurrently with the destruction of the objects with static storage duration and with each other, maintaining the guarantee that if registration of A was sequenced-before the registration of B, then the call to B is sequenced-before the call to A, same applies to the sequencing between static object constructors and calls to `atexit`: see `std::exit`	(since C++11)

The same function may be registered more than once.

If a function exits via an exception, `std::terminate` is called.

`atexit` is thread-safe: calling the function from several threads does not induce a data race.

The implementation is guaranteed to support the registration of at least 32 functions. The exact limit is implementation-defined.

Parameters

func - pointer to a function to be called on normal program termination

Return value

`0` if the registration succeeds, nonzero value otherwise.

Notes

The two overloads are distinct because the types of the parameter `func` are distinct (language linkage is part of its type)

Example

Run this code

```
#include <iostream>
#include <cstdlib>

void atexit_handler_1()
{
    std::cout << "at exit #1\n";
}

void atexit_handler_2()
{
    std::cout << "at exit #2\n";
}

int main()
{
    const int result_1 = std::atexit(atexit_handler_1);
    const int result_2 = std::atexit(atexit_handler_2);

    if ((result_1 != 0) || (result_2 != 0)) {
        std::cerr << "Registration failed\n";
        return EXIT_FAILURE;
    }
 
    std::cout << "returning from main\n";
    return EXIT_SUCCESS;
}
```

Output:

```
returning from main
at exit #2
at exit #1
```

See also

abort	causes abnormal program termination (without cleaning up) (function)
exit	causes normal program termination with cleaning up (function)
quick_exit (C++11)	causes quick program termination without completely cleaning up (function)
at_quick_exit (C++11)	registers a function to be called on std::quick_exit invocation (function)
C documentation for **atexit**	

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/utility/program/atexit&oldid=117939"