

Search:

Go

Not logged in

Reference

<fstream>

filebuf

seekoff

register

log in

C++

Information

Tutorials

Reference

Articles

Forum

Reference

C library:

Containers:

Input/Output:

<fstream>

<iomanip>

<ios>

<iosfwd>

<iostream>

<istream>

<ostream>

<sstream>

<streambuf>

Multi-threading:

Other:

<fstream>

class templates:

basic_filebuf

basic_fstream

basic_istream

basic_ofstream

classes:

filebuf

fstream

istream

ofstream

wfilebuf

wfstream

wistream

wofstream

filebuf

filebuf::filebuf

filebuf::~filebuf

public members:

filebuf::close

filebuf::is_open

filebuf::open

filebuf::operator=

filebuf::swap

protected virtual members:

filebuf::imbue

filebuf::overflow

filebuf::pbackfail

filebuf::seekoff

filebuf::seekpos

filebuf::setbuf

filebuf::showmanyc

filebuf::sync

filebuf::uflow

filebuf::underflow

non-member overloads:

swap (filebuf)

Pisql Features

Download the 30 day trail

version for PL/SQL IDE!

protected virtual member function

std::filebuf::seekoff

<fstream>

```
streampos seekoff (streamoff off, ios_base::seekdir way,
                  ios_base::openmode which = ios_base::in | ios_base::out);
```

Set internal position to relative position

Sets a new position for the internal position pointers specified by parameter *which*. This position is calculated as an offset of *off* characters relative to a the origin specified by *way*.

Unless *off* is zero and *way* is *cur*, the function also writes any unwritten characters in the *intermediate output buffer* to the file (also calling *unshift* using the proper facet, when needed).

When successful, this function has the same effects as the equivalent call to *fseek* (see *fseek* for details and limitations, such as on files open in text mode).

This virtual function is called by the public member *streambuf::pubseekoff*.

Parameters

off

Offset value, relative to the way parameter.
streamoff is a signed integral type.

way

Object of type *ios_base::seekdir*, indicating the origin from which the offset is applied. It may take any of the following constant values:

value	offset is relative to...
<i>ios_base::beg</i>	beginning of the file
<i>ios_base::cur</i>	current position of either the <i>input position</i> or the <i>output position</i> , depending on argument <i>which</i> .
<i>ios_base::end</i>	end of the file

which

Determines which of the *internal position pointers* is affected: the *input position*, the *output position*, or both. It is an object of type *ios_base::openmode* that, for this function, may take any combination of the following significant constant values:

value	position pointer affected
<i>ios_base::in</i>	Modifies the <i>input position</i> , and its corresponding <i>get pointer</i> (<i>gptr</i>)
<i>ios_base::out</i>	Modifies the <i>output position</i> and its corresponding <i>put pointer</i> (<i>pptr</i>)

Both positions may be selected simultaneously, but if both are selected when *way* is *ios_base::cur*, the function fails.

Return Value

On success, it returns the new absolute position the *internal position pointer* points to after the call, if representable as a value of type *streampos*.

On failure, or if the above is not possible, the function returns *streampos(streamoff(-1))*.

streampos is a positioning type that can be converted to/from integral types (an *fpos* type).

Data races

Modifies the *filebuf* object.

Concurrent access to the same *file stream buffer* object may introduce data races.

Exception safety

Basic guarantee: if an exception is thrown, the *file stream buffer* is in a valid state.

See also

filebuf::seekpos Set position pointer to absolute position (*protected virtual member function*)

streambuf::pubseekoff Set internal position pointer to relative position (*public member function*)