data_seg

Specifies the data segment where initialized variables are stored in the .obj file.

Syntax

```
#pragma data_seg( [ [ { push | pop }, ] [ identifier, ] ] [ "segment-name" [, "segment-
class" ] )
```

Remarks

The meaning of the terms *segment* and *section* are interchangeable in this topic.

OBJ files can be viewed with the dumpbin application. The default segment in the .obj file for initialized variables is .data. Variables that are uninitialized are considered to be initialized to zero and are stored in .bss.

data_seg with no parameters resets the segment to .data.

push(optional)
> Puts a record on the internal compiler stack. A **push** can have an *identifier* and *segment-name*.

pop (optional)
> Removes a record from the top of the internal compiler stack.

identifier (optional)
> When used with **push**, assigns a name to the record on the internal compiler stack. When used with **pop**, pops records off the internal stack until *identifier* is removed; if *identifier* is not found on the internal stack, nothing is popped.

> *identifier* enables multiple records to be popped with a single **pop** command.

"segment-name"(optional)
> The name of a segment. When used with **pop**, the stack is popped and *segment-name* becomes the active segment name.

"segment-class" (optional)
> Included for compatibility with C++ prior to version 2.0. It is ignored.

Example

```cpp
// pragma_directive_data_seg.cpp
int h = 1;                      // stored in .data
int i = 0;                      // stored in .bss
#pragma data_seg(".my_data1")
int j = 1;                      // stored in "my_data1"

#pragma data_seg(push, stack1, ".my_data2")
int l = 2;                      // stored in "my_data2"

#pragma data_seg(pop, stack1)   // pop stack1 off the stack
int m = 3;                      // stored in "stack_data1"

int main() {
}
```

Data allocated using **data_seg** does not retain any information about its location.

See /SECTION for a list of names you should not use when creating a section.

You can also specify sections for const variables (const_seg), uninitialized data (bss_seg), and functions (code_seg).

See Also

Pragma Directives and the __Pragma Keyword