

1540-0701 (S) The limit on nested ...



Alexander-Vinokur 4 posts since

Jun 22, 2009

Hi,

uname -aAIX ilibm016 3 5 0004F43AD400

/usr/vacpp/bin/xlC_r -qversionIBM XL C/C++ Enterprise Edition V8.0 for AIX

Version: **08.00.0000. 0000**

Here is some program

- prog.cpp ---

```
template <int N>
int foo()
{
return (foo<N-1>() + 2);
}

template <>
int foo<0>()
{
return 0;
}

int main()
{
int value = foo<300>();
return value;
}
```

Compilation:

/usr/vacpp/bin/xlC_r -q64 -qwarn64 prog.cpp "prog.cpp", line 2.5: 1540-0701 (S) The limit on nested templateinstantiations has been exceeded while instantiating "int foo<250>()".

=====

Question-1. Is there any compilation option or pragma that enables the user to increase the template-related recursion deep/limit?

Question-2. Does any xlC's description contains the exact value of

1540-0701 (S) The limit on nested ...

that limit?

Thanks.

Alex Vinokur Tags: aix, c++, templates



[DaveyC](#) 22 posts since

Jan 20, 2009 1. **Re: 1540-0701 (S) The limit on nested template** Jun 23, 2009 7:24 AM

Works fine on the z/OS v1.10 compiler. I thought they all shared the same front end? BTW, good luck. Template meta-programming is for the hard core! Wouldn't want to support it...



[Michael_Wong](#) 19 posts since

Sep 30, 2008 2. **Re: 1540-0701 (S) The limit on nested template** Jun 23, 2009 7:38 AM

Hi, xIC++ (AIX/Linux) V10.1 has template depth control, starting with V9. But V8 does not have that option.

-qtemplatedepth (C++ only)

Category

Template control

Pragma equivalent

None.

Purpose

Specifies the maximum number of recursively instantiated template specializations that will be processed by the compiler.

Syntax

Read syntax diagramSkip visual syntax diagram

-q--templatedepth--==--number-----><

Defaults

-qtemplatedepth=300

Parameters

number

The maximum number of recursive template instantiations. The number can be a value between 1 and INT_MAX. If your code attempts to recursively instantiate more templates than number, compilation halts and an error message is issued. If you specify an invalid value, the default value of 300 is used.

Usage

Note that setting this option to a high value can potentially cause an out-of-memory error due to the complexity and amount of code generated.

Predefined macros

None.

Examples

To allow the following code in myprogram.cpp to be compiled successfully:

```
template <int n> void foo() {  
foo<n-1>();  
}  
  
template void foo<0>() {}  
  
int main() {  
foo<400>();  
}
```

Enter:

```
xlc++ myprogram.cpp -qtemplatedepth=400
```

Related information

* Using C++ templates .

Parent topic: Individual option descriptions

[Provide feedback](#)