

C++14 LWG.2193

explicitなデフォルトコンストラクタをなくす

---

高橋 晶(Akira Takahashi)

[faithandbrave@longgate.co.jp](mailto:faithandbrave@longgate.co.jp)

2014/06/27(金) WG21 C++14 DISレビュー会議

# まえがき

---

- この資料は、C++14に取り入れられる予定の変更、LWG (Library Working Group)のIssue 2194のレビューです。
  - 2193. Default constructors for standard library containers are explicit
    - <http://www.open-std.org/jtc1/sc22/wg21/docs/lwg-active.html#2193>
-

# 概要

---

- `explicit Container(Allocator alloc = Allocator());`
  - となっているデフォルトコンストラクタを、  
以下のように分割する：
  - `Container() : Container(Allocator()) {}`
  - `explicit Container(Allocator alloc);`
-

# 元々の問題

---

- 以下のコードが不適格になっていた：

```
std::set<int> s = {}; // エラー！explicitが付いている
```

- このコードはClang 3.0でコンパイルエラーになっていた。
  - この問題を解決するために、前述した解決策が提示されたが、実際にはこの問題はEWG (Evolution Working Group)側で処理された。
  - そのためこのIssueでの解決策は、気持ち悪さの解消程度。
-

# 变更对象

---

- `basic_string`
  - `deque`
  - `forward_list`
  - `list`
  - `vector`
  - `(multi)map`
  - `(multi)set`
  - `unordered_(multi)map`
  - `unordered_(multi)set`
-

# ABIが壊れないか調査した

---

- オーバーロードの追加は、ABIを壊さない。
  - デフォルト引数の追加・削除は、ABIを壊さない。
  - よって、この変更はABIを壊さない。
  - C++11でstd::vectorのresize()メンバ関数に以下の変更が入ったが、その変更はABIを壊さなかった。
    - `void resize(size_type sz, T c = T());`
    - `void resize(size_type sz);`
    - `void resize(size_type sz, const T& c);`
-

# 参照

---

- ABI Policy and Guidelines - libstdc++
    - <http://gcc.gnu.org/onlinedocs/libstdc++/manual/abi.html>
  - Cxx11AbiCompatibility - GCC Wiki
    - <http://gcc.gnu.org/wiki/Cxx11AbiCompatibility>
  - Thiago Macieira: Binary compatibility for library developers
    - [https://github.com/boostcon/cppnow\\_presentations\\_2013/blob/master/tue/binary\\_compat\\_for\\_cpp\\_devs.pdf?raw=true](https://github.com/boostcon/cppnow_presentations_2013/blob/master/tue/binary_compat_for_cpp_devs.pdf?raw=true)
-