

Visual C++ 2015の紹介 (C++11/14的に)

H.27/09/30

Extra

In 歌舞伎座.tech 番外編「C++11/14コア言語」出版記念



自己紹介

- **Egtra**
 - **C++プログラマ(?)**
 - **Visual C++ (MSVC) を使う**



自己紹介

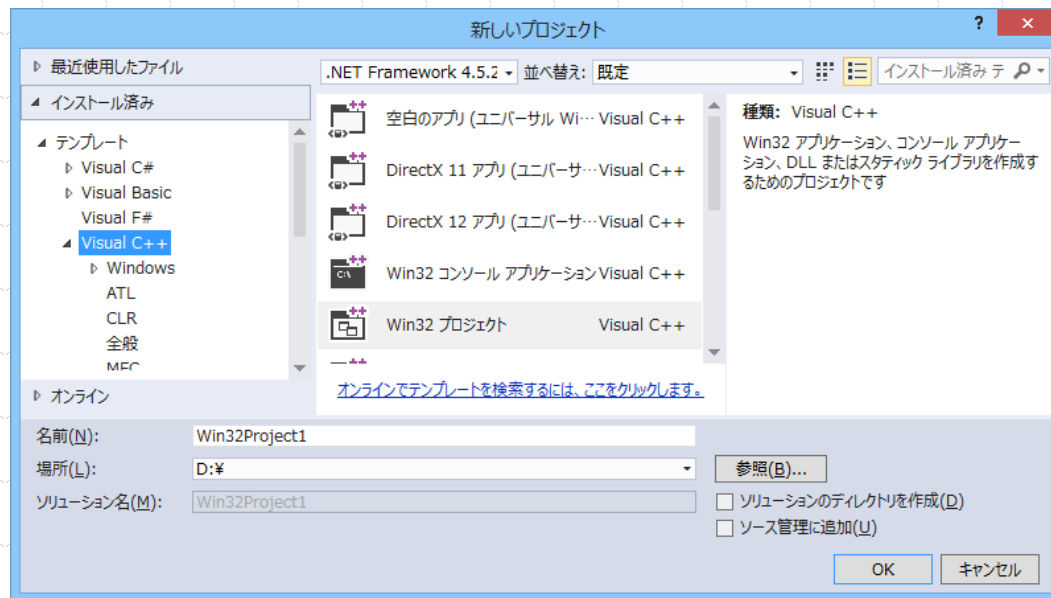
• C++ポケリ

- 今年(2015年)6月: 改定新版



Visual C++とは何か

- Microsoftの開発環境・コンパイラ
- MSVCなどとも呼ばれる



Visual C++とは何か

江添曰く

江添 Ryou Ezoë
@EzoëRyou

 Follow

ようやく、やっと、うちの会社がMSVC6.0からMSVC2013に移行する気になったらしい。だが、古いライブラリに依存しているので、新環境に適応できるか、非常に疑問だ。 — MSVCはC++コンパイラーではない。 ask.fm/a/c01dickd

 View translation

RETWEETS

11

FAVORITES

4



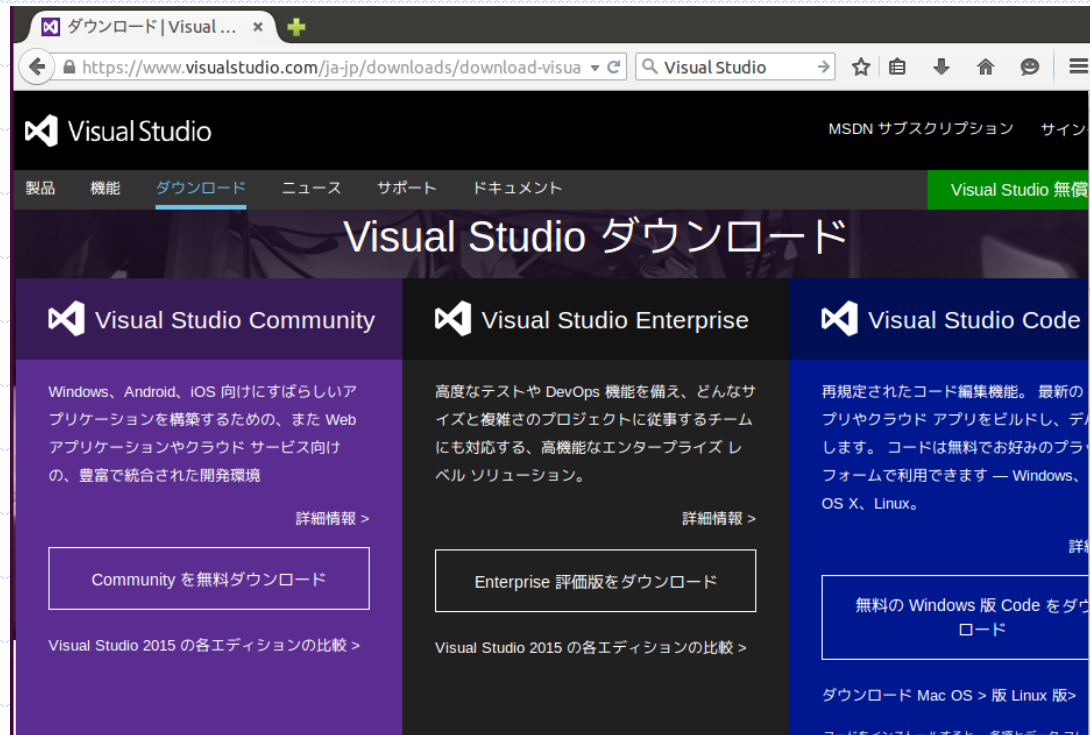
11:38 PM - 27 Mar 2015

江添曰く

MSVCは
C++コンパイラー
ではない。

Visual Studio 2015

- 2015年7月: RTMがリリース



Visual C++ 2015

- **だいぶ実装されてきた**
 - **Cpprefjp基準で**
 - **コンパイラの実装状況**

<http://cpprefjp.github.io/implementation-status.html>



Visual C++ 2015

- **C++11: ↓以外はできた**
 - Expression SFINAE
 - C99プリプロセッサ
- **C++14: ぼちぼち**
 - 変数テンプレートなどがまだ

C++14: 関数の戻り値型推論

```
template<typename T, typename U>  
auto f(T&& x, U&& y)  
{  
    return std::forward<T>(x)  
        + std::forward<U>(y);  
}
```

C++14: 汎用ラムダキャプチャ

```
std::promise<int> p;  
auto f = [q = std::move(p)] {  
    q.set_value(1);  
};
```

C++14: ジェネリックラムダ

```
auto f = [](auto x, auto y) {  
    return x + y;  
};  
auto i = f(1, 2);  
auto s = f("abc"s, "def");
```

C++14: その他実装されたもの

- **二進数リテラル**
- **[[deprecated]]属性**
- **数値リテラルの区切り**
 - **123'456'789**
- **サイズ指定デアロケーション**

まとめ

- **Visual C++ 2015は、
それなりにC++11/14風の
コードを書ける。**

まとめ (2)

- **Visual C++はC++コンパイラではないので、私はC++プログラマではない。**

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 License](https://creativecommons.org/licenses/by-sa/4.0/).

