embarcadero
Developer Camp

【B2】C++テクニカルセッション

「C++言語新機能を使おう！」

エンバカデロ・テクノロジーズ
エヴァンジェリスト　高橋智宏

embarcadero
Developer Camp

C++11
64bit

C++Builder XE3

bcc64

```
C:¥>bcc64 --version
Embarcadero C++ 6.50 for Win64 Copyright (c) 2012 Embarcadero Technologies, Inc.
Embarcadero Technologies Inc. bcc64 version 3.1 (32869.98b6e64.7f8e142) (based o
n LLVM 3.1svn)
Target: x86_64-pc-win32-elf
Thread model: posix


C:¥>bcc64 --help
Embarcadero C++ 6.50 for Win64 Copyright (c) 2012 Embarcadero Technologies, Inc.
OVERVIEW: bcc64 driver
USAGE: bcc64 [options] <inputs>

OPTIONS:
  -###                    このコンパイルで実行するコマンドを出力する
  --analyze               静的アナライザを実行する
  --help                  使用可能なオプションを表示する
  --migrate               移行ツールを実行する
  --relocatable-pch       再配置可能なプリコンパイル済みヘッダーを作成する
  --serialize-diagnostics <value>
                          コンパイラの診断情報をファイルにシリアル化する
  ...
```

bcc64

```
  -E                        プリプロセッサのみ実行する
  -ObjC++                   ソース入力ファイルを Objective-C++ の入力として扱う
  -ObjC                     ソース入力ファイルを Objective-C の入力として扱う
  -Qunused-arguments        使用されていないドライバ引数の警告を出力しない
  -S                        プリプロセスとコンパイルのステップのみ実行する
  -Wa,<arg>                 <arg> で指定されたコンマ区切り引数をアセンブラに渡す
  -Wl,<arg>                 <arg> で指定されたコンマ区切り引数をリンカに渡す
  -Wp,<arg>                 <arg> で指定されたコンマ区切り引数をプリプロセッサに渡す
  -Xanalyzer <arg>          <arg> を静的アナライザに渡す
  -Xassembler <arg>         <arg> をアセンブラに渡す
  -Xclang <arg>             <arg> を Clang コンパイラに渡す
  -Xlinker <arg>            <arg> をリンカに渡す
  -Xpreprocessor <arg>      <arg> をプリプロセッサに渡す
   -arcmt-migrate-emit-errors
                            たとえ移行ツールで ARC エラーを修正できても、そのエラーを出力する
  -arcmt-migrate-report-output <value>
                            plist レポートの出力パス
  -c                        プリプロセス、コンパイル、アセンブルのステップのみ実行する
  -emit-ast                 ソース入力用の Clang AST ファイルを出力する
  -emit-llvm                アセンブラ ファイルとオブジェクト ファイルに LLVM 表現を使用する
  -fcatch-undefined-behavior
                            未定義の動作の実行時チェックを生成する。
  -flimit-debug-info        生成されるデバッグ情報を制限してデバッグ バイナリのサイズを小さくする
  -fno-limit-debug-info     生成されるデバッグ情報を制限してデバッグ バイナリのサイズを小さくする
ことはしない
  …
```

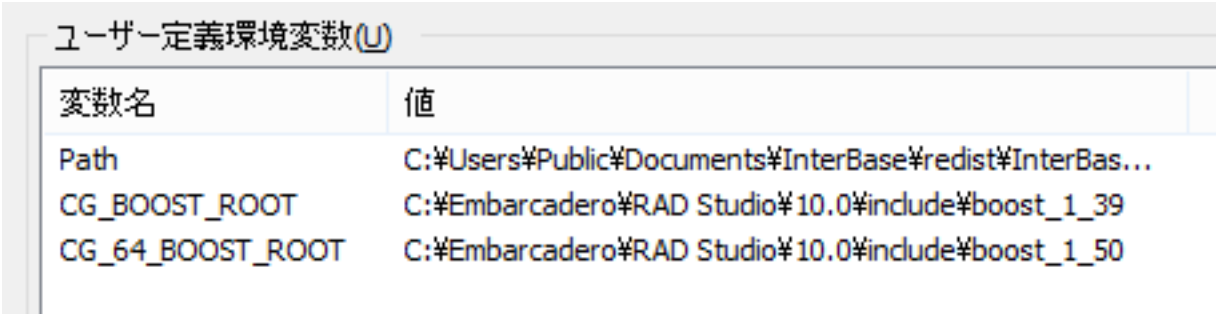
bcc64

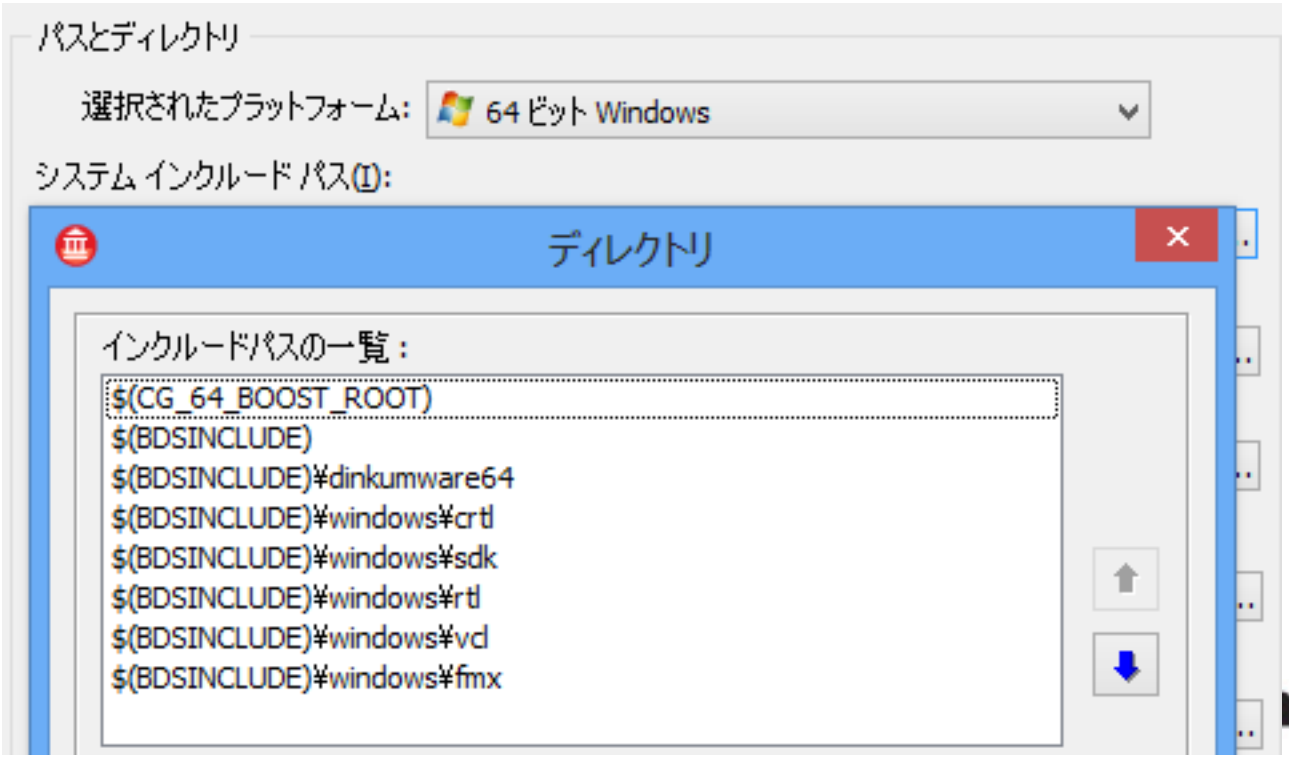
```
-ftrap-function=<value> トラップ命令ではなく指定された関数の呼び出しを発行する
-gcc-toolchain <value>  指定されたディレクトリの gcc ツール チェーンを使用する
-n <directory>          中間ファイルの出力ディレクトリを指定する
-objcmt-migrate-literals
                        最新の Objective-C リテラルへの移行を可能にする
-objcmt-migrate-subscripting
                        最新の Objective-C 添字指定への移行を可能にする
-o <file>               出力を <file> に書き込む
-pipe                   可能であれば、コマンド間にパイプを使用する
-print-file-name=<file> <file> というライブラリの完全パスを出力する
-print-libgcc-file-name "libgcc.a" のライブラリ パスを出力する
-print-prog-name=<name> <name> というプログラムの完全パスを出力する
-print-search-dirs      ライブラリやプログラムの検索に使用されたパスを出力する
-q                      コンパイラ識別バナーを表示しない
-rewrite-legacy-objc    既存の Objective-C ソースを C++ に書き換える
-rewrite-objc           Objective-C ソースを C++ に書き換える
-save-temps             コンパイルの中間結果を保存する
-target <value>         指定されたターゲットのコードを生成する
-time                   個々のコマンドの時間を計る
-t<value>               Borland ターゲット実行可能ファイルを指定する
-verify                 検証ツールを使って出力を検証する。
-v                      実行するコマンドを表示し、詳細出力を使用する
-working-directory <value>
                        指定されたディレクトリからの相対ファイル パスを解決する
-x <language>           後続の入力ファイルを <language> タイプの入力として扱う
```

ilink64

```
C:¥>ilink64
Turbo Incremental Link 0.20 Copyright (c) 1997-2012 Embarcadero Technologies, Inc.
Syntax: ILINK objfiles, exefile, mapfile, libfiles, deffile, resfiles
@xxxx indicates use response file xxxx
General Options:                          -Af:nnnn Specify file alignment
  -C       Clear state before linking     -Ao:nnnn Specify object alignment
  -wxxx    Warning control                -ax      Specify application type
  -Enn     Max number of errors           -b:xxxx  Specify image base addr
  -r       Verbose linking                -Txx     Specify output file type
  -q       Suppress banner                -H:xxxx  Specify heap reserve size
  -c       Case sensitive linking         -Hc:xxxx Specify heap commit size
  -v       Full debug information         -S:xxxx  Specify stack reserve size
  -Gn      No state files                 -Sc:xxxx Specify stack commit size
  -Gi      Generate import library        -Vd.d    Specify Windows version
  -GD      Generate .DRC file             -Dstring Set image description
Map File Control:                         -Vd.d    Specify subsystem version
  -M       Map with mangled names         -Ud.d    Specify image user version
  -m       Map file with publics          -GC      Specify image comment str
  -s       Detailed segment map           -GF      Set image flags
  -x       No map                         -Gl      Static package
Paths:                                    -Gpd     Design time only package
  -I       Intermediate output dir        -Gpr     Runtime only package
  -L       Specify library search paths   -GS      Set section flags
  -j       Specify object search paths    -Gt      Fast TLS
Image Control:                            -Gz      Do image checksum
  -d       Delay load a .DLL              -Rr      Replace resources
```

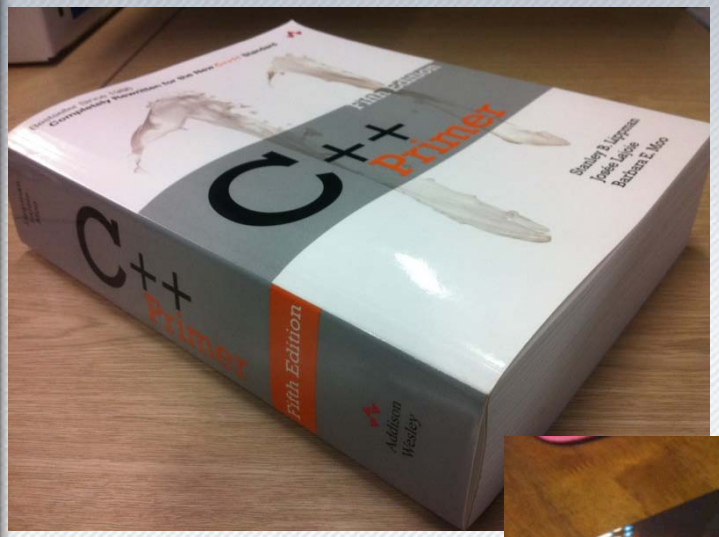
Dinkumware 5.3(64bit), Boost 1.50(64bit)

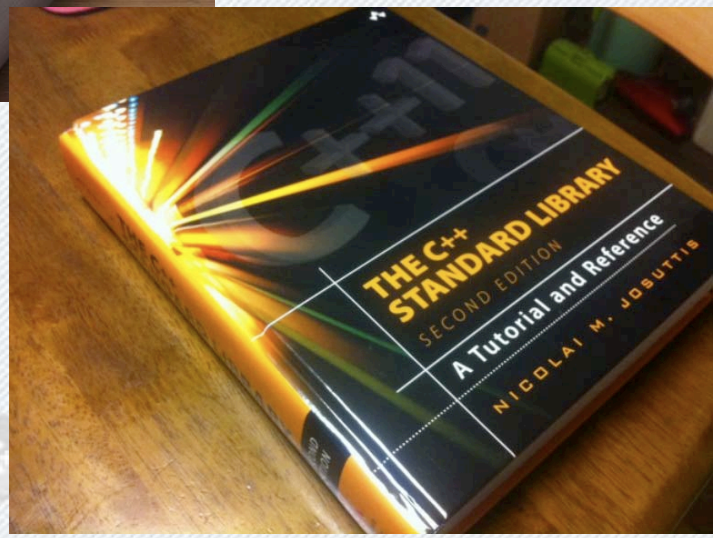




embarcadero
Developer Camp

新しい
言語機能&ライブラリ





データ型, 文字列リテラル

```
short a0;        // 16bit
int a1;          // 32bit
long a2;         // 32bit
long long a3;    // 64bit
float a4;        // 32bit
double a5;       // 64bit
long double a6; // Win64:64bit, Win32:80bit

Extended a7; // Win64:64bit, Win32:80bit
Currency a8; // ?
```

```
ShowMessage(L"Hello ¥¥ こんにちは ¥" 你好");

ShowMessage(LR"(Hello ¥ こんにちは " 你好)");

ShowMessage(R"delm((a b c))delm");
```

型のエリアス, 初期化構文, auto型, Rang for

```
using MYINT = int;
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    MYINT v[] = {0,1,2,3};

    vector<vector<int>> vv = {{1,2},{3,4}};
}
```

```
MYINT v[] = {0,1,2,3};
for(auto& c : v) {
  c++;
}

vector<vector<int>> vv = {{1,2},{3,4}};
for(auto a : vv) {
  for(const auto b : a) {
    //b++;
  }
}
```

```
vector<int> cv = {1,2,3};
auto citr = cv.cbegin(); // vecot<int>::const_iterator
//*citr = 99;
```

ilitializer_list パラメータ

```
#include <initializer_list>
using namespace std;
void varParamsFunc(initializer_list<string> argv)
{
  for(auto p : argv) {
    ShowMessage(AnsiString(p.c_str()));
  }
}

void __fastcall TForm1::Button5Click(TObject *Sender)
{
  varParamsFunc({"abc","def"});
}
```

auto + 戻り値型 + decltype

```
static int ar[10];

int (*testFunc(int i))[10]
{
  return &ar;
}

auto trailFunc(int i) -> int(*)[10]
{
  return &ar;
}

template <typename T>
auto MyAddFunc(T a, T b) -> decltype(a)
{
  return a + b;
}

auto a = testFunc(0);
auto b = trailFunc(0);
auto c = MyAddFunc(1, 2);
```

コンストラクタのデリゲート

```
class MyBook {
private:
  string isbn;
  double price;
public:
  MyBook() : MyBook("", 0.0) { }
  MyBook(string s) : MyBook(s, 0.0) { }
  MyBook(string s, double p) { isbn = s; price = p; }
};

void __fastcall TForm1::Button7Click(TObject *Sender)
{
  MyBook b("123-456");
}
```

ラムダ式

```
[capture list] (parameter list) -> return type { function body }
```

※パラメータリストと戻り値型は省略可能
※return文が無ければ戻り値はvoid
※戻り値型は decltype(return文の型)

```
void __fastcall TForm1::Button8Click(TObject *Sender)
{
  auto f = [Sender] { return Sender->ClassName(); };
  auto a = f();
  ShowMessage(a);
}
```

ラムダ式(キャプチャーリスト)

```
□int v1 = 40;
  auto f = [v1] { return v1; };
  v1 = 99;
  auto v2 = f();
  ShowMessage(IntToStr(v2));
```

```
  int v1 = 40;
  auto f = [&v1] { return v1; };
  v1 = 99;
  auto v2 = f();
  ShowMessage(IntToStr(v2));
```

```
void __fastcall TForm1::Button11Click(TObject *Sender)
{
  auto f = [=] { return Sender->ClassName(); };
  auto a = f();
  ShowMessage(a);
}
```

スマートポインタ(shared_ptr, unique_ptr)

```
#include <memory>
void __fastcall TForm1::Button12Click(TObject *Sender)
{
  shared_ptr<int> sp1(new int(40));

  auto sp2 = make_shared<int>(40);
  if(sp2) {
    ShowMessage(IntToStr(*sp2));
  }

  auto sp3 = make_shared<vector<int>>();
}
```

```
void __fastcall TForm1::Button13Click(TObject *Sender)
{
  unique_ptr<string> up1(new string("test"));

  //unique_ptr<int> up2 = up1;
  unique_ptr<string> up2(up1.release());
}
```

配列とnewと初期化

```
void __fastcall TForm1::Button14Click(TObject *Sender)
{
  int* p1 = new int[10];
  int* p2 = new int[10]();
  …

  string* p3 = new string[10]{"ab","cd","ef"};
  int* p4 = new int[10]{1,2,3,4,5};
  …
}
```

function型

```
int MyAddFuncPtr(int a, int b) { return a + b; }

struct MyDivFuncObj {
  int operator()(int x, int d) { return x / d; }
};

#include <functional>
void __fastcall TForm1::Button15Click(TObject *Sender)
{
  function<int(int, int)> f1 = MyAddFuncPtr;
  function<int(int, int)> f2 = MyDivFuncObj();
  function<int(int, int)> f3 = [](int i, int j) { return i * j; };

  auto a1 = f1(4, 2); // 6
  auto a2 = f2(4, 2); // 2
  auto a3 = f3(4, 2); // 8
}
```

finalクラス, override

```
class MyFinalClass final {
};

class MyChildClass : MyFinalClass {
};
```

```
class MyParent {
public:
  virtual void f1() {}
  virtual void f2(int x) {}
  void f3() {}
};

class MyChild : public MyParent {
public:
  void f1() override {}
  //void f2() override {}
  //void f3() override {}
  //void f4() override {}
};
```

tuple型

```
#include <tuple>
void __fastcall TForm1::Button16Click(TObject *Sender)
{
  tuple<string, double> book1("123-456", 99.9); // contructor
  tuple<string, double> book2{"123-456", 99.9}; // initializer

  // tuple<const char*, double>
  auto book3 = make_tuple("123-456", 99.9);
  auto isbn3 = get<0>(book3);
  auto price3 = get<1>(book3);

  // tuple<string, double>
  auto book4 = make_tuple(string("123-456"), 99.9);
  auto isbn4 = get<0>(book4);
  auto price4 = get<1>(book4);
}
```

乱数

```
#include <random>
void __fastcall TForm1::Button17Click(TObject *Sender)
{
  default_random_engine e1(time(0)); // unsigned int
  for(int i = 0; i < 3; i++) {
    ListBox2->Items->Add( UIntToStr((unsigned int)e1()) );
  }

  uniform_int_distribution<unsigned long> u(0, 9);
  for(int i = 0; i < 3; i++) {
    ListBox2->Items->Add( IntToStr((int)u(e1)) );
  }

  mt19937_64 e2(time(0)); // unsigned long long(64bit)
  for(int i = 0; i < 3; i++) {
    ListBox2->Items->Add( UIntToStr((unsigned __int64)e2()) );
  }
}
```

embarcadero
Developer Camp

クイズ!!

最も適切な構文は？

```
#include <memory>
void test() {
   std::unique_ptr<int>   p1(new int[10]); //(イ)
   std::unique_ptr<int[]> p2(new int[10]); //(ロ)
   std::shared_ptr<int>   p3(new int[10]); //(ハ)
   std::shared_ptr<int[]> p4(new int[10]); //(ニ)
}
```

正解は…

（イ）　（ロ）

（ハ）　（ニ）

(イ) 適切な delete [] が呼ばれない

```
unique_ptr<int> p1(new int[10]);
```

```
unique_ptr<int,default_delete<int[]>> p1(new int[10],
                                         default_delete<int[]>());
```

```
unique_ptr<int,void(*)(int*)> p1(new int[10],
                                 [](int* p){ delete [] p; });
```

```
unique_ptr<int,function<void(int*)>> p1(new int[10],
                                        [](int* p){ delete [] p; });
```

```
auto d = [](int* p){ delete [] p; };
unique_ptr<int,decltype(d)> p1(new int[10], d);
```

(ロ) 完全に正しい

```
unique_ptr<int[]> p2(new int[10]);
```

delete [] が正しく呼び出される

(ハ)適切な delete [] が呼ばれない

```
shared_ptr<int> p3(new int[10]);
```

```
shared_ptr<int> p3(new int[10], std::default_delete<int[]>());
```

```
shared_ptr<int> p3b(new int[10], [](int* p){ delete [] p; });
```

(二) コンパイルエラー

```
shared_ptr<int[]> p4(new int[10]);
```

コンパイルできません

embarcadero
Developer Camp

Boost 1.50

配列用スマートポインタ

```
#include <boost/scoped_array.hpp>
void __fastcall TForm1::Button19Click(TObject *Sender)
{
  boost::scoped_array<int> sp1(new int[10]);
}
```

```
#include <boost/shared_array.hpp>
void __fastcall TForm1::Button19Click(TObject *Sender)
{
  boost::shared_array<int> sp2(new int[10]);
}
```

Boost 1.50 : Local Functions

http://www.boost.org/doc/libs/1_50_0/libs/local_function/doc/html/index.html

```
#include <boost/local_function.hpp>
void __fastcall TForm1::Button20Click(TObject *Sender)
{
  int sum = 0;
  int factor = 10;

  void BOOST_LOCAL_FUNCTION(bind& sum, const bind factor, int num) {
    sum += factor * num;
  } BOOST_LOCAL_FUNCTION_NAME(add)

  add(1);
  int nums[] = {2, 3};
  for_each(nums, nums + 2, add);

  ShowMessage(IntToStr(sum));
}
```

embarcadero
Developer Camp

Q & A