



BORLAND® DEVELOPER CAMP

テクニカルセッション

オープンソースを利用した 3層C/Sシステムの構築方法

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。



BORLAND® DEVELOPER CAMP

第2回 ボーランド デベロッパー キャンプ

講師紹介

- 高橋智宏
- 1973年生まれ、京都大学 法学部卒
- エバンジェリスト 兼 コンサルタント 兼 トレーナー...
- 学生の時購入したTurboC++2ndからの熱狂的なボーランドファン
- 参加しているメーリングリストやコミュニティ
 - JBuilder ML, C++Builder ML, Delphi ML, C# ML, CORBA ML 等...
 - ミクシィ
 - http://mixi.jp/show_friend.pl?id=208738
- 「Java読書会」を運営
 - <http://www.javareading.com/bof/>

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。

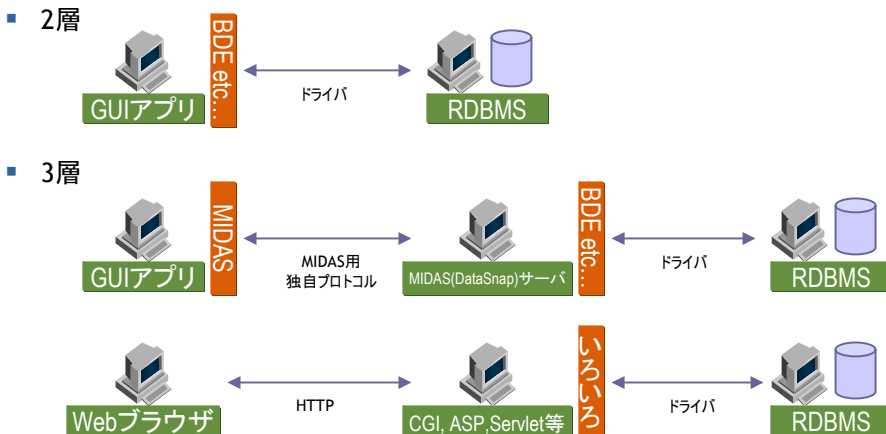
アジェンダ

- 2層 から 3層へ
- CORBA(IIOP)の概要
- 作成するシステムの概要
- C++用オープンソースCORBA実装
 - C++Builderで作るCORBAサーバ
 - MySQL5.0.x と オープンソースdbExpressドライバ
 - C++Builderで作るCORBAクライアント
- Java用オープンソースCORBA実装で作るクライアント
- Delphi(Win32)用オープンソースCORBA実装で作るクライアント
- .NET用オープンソースCORBA実装で作るクライアント
- Delphi(Win32)用オープンソースCORBA実装で作るCORBAサーバ
 - InstantObjects(OPF, O/Rマッパー) + dbExpress + MySQL5.0.x

Borland®

Copyright © 2006, Borland Software Corporation. 本スライドの一部または全部の複製を禁じます。

2層 から 3層へ

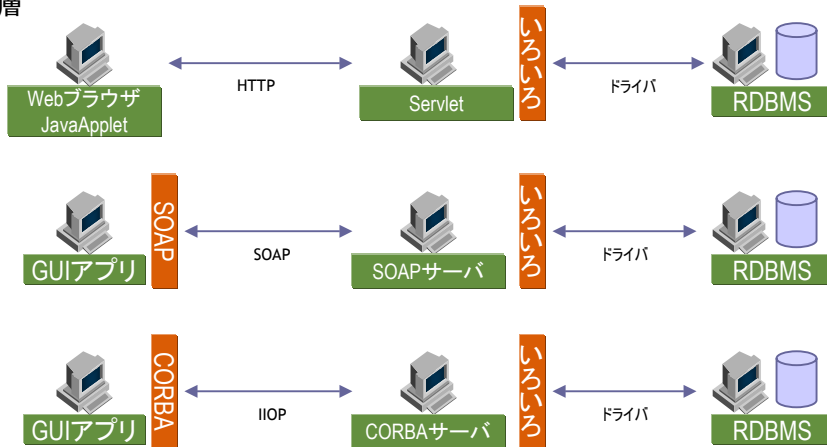


Borland®

Copyright © 2006, Borland Software Corporation. 本スライドの一部または全部の複製を禁じます。

2層 から 3層へ (続き)

■ 3層

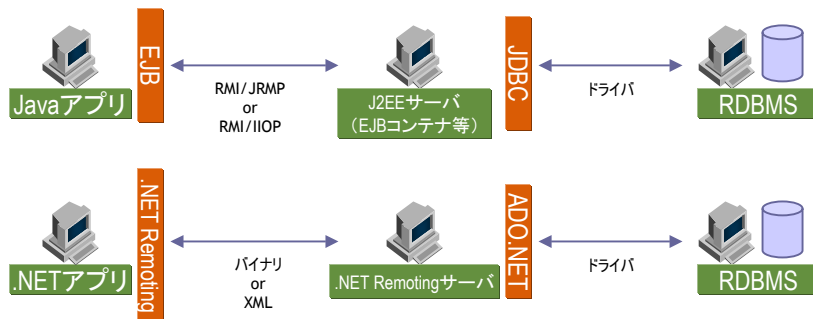


Borland®

Copyright © 2000, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

2層 から 3層へ (続き)

■ 3層



Borland®

Copyright © 2000, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

CORBAの概要

- CORBA(Common Object Request Broker Architecture)は、オブジェクト間コミュニケーションのために業界で開発された世界的に公認された標準技術
 - ISOの国際標準「ISO/IEC19500-2」として 2000年10月に採択
- オブジェクトマネジメントグループ (OMG)
 - 分散オブジェクト標準を開発する為に発足した業界コンソーシアム
 - ハードウェアベンダ、ソフトウェアベンダ、ユーザを含む800以上の企業がメンバーとして参加
 - 共通オブジェクトリクエストブローカーアーキテクチャ(CORBA)の仕様を開発
 - 最新のCORBAのドキュメントは、<http://www.omg.org/> から入手可能
 - 各種仕様の定義に関する提案を認可
 - 通信インフラストラクチャ
 - オブジェクト間の標準インタフェース
 - CORBAに基づく標準サービスの標準化

Borland®

Copyright © 2000, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

CORBAの良いところ

- あらかじめ決定しておく必要がない
 - プログラミング言語
 - ハードウェアプラットフォーム
 - オペレーティングシステム
 - サーバ/クライアントの分散の度合い
 - サーバプロセスをネットワークの何処に配置するか?
 - 各種サーバコンポーネントを何処のプロセス内にどれだけ配置するか?
 - サーバコンポーネントをリモートプロセス/インプロセスのどちらで実行するか?
- CORBAの鍵となる特性は相互運用性である
 - プラットフォーム、言語、ベンダをまたいだ相互運用性
 - CORBA製品やサービスのプラグ&プレイ による代用性
- Webサービス(SOAP)にはできないことが...

Borland®

Copyright © 2000, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

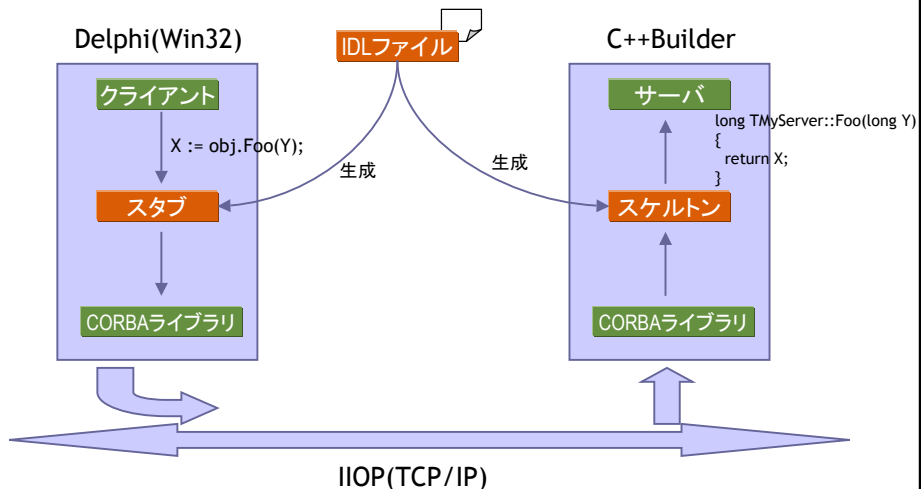
CORBAの一般的な構成要素

- インタフェース定義言語 (IDL)
 - Interface Definition Language (インタフェース定義言語)
 - IDLを使ってサーバのインタフェースを記述 (IDLファイル)
- プログラミング言語に対応するIDLマッピング
 - 例: IDLのlongは、Javaのint型、Delphi(Win32)のlongint型 ...
- オブジェクト間コミュニケーションのためのワイヤプロトコル (IIOP)
 - Internet Inter-ORB Protocol (インターネットORB間プロトコル)
 - TCP/IPを利用する
- ネットワーク上で引き渡し可能なオブジェクトハンドルのフォーマット (IOR)
 - Interoperable Object Reference (相互運用可能オブジェクト参照)
- プログラミング言語に対応したスタブ/スケルトン
 - CORBAミドルウェアに付属のIDLコンパイラを使用して、IDLファイルから自動生成されるコード
 - クライアントはスタブ、サーバはスケルトンを利用する

Borland®

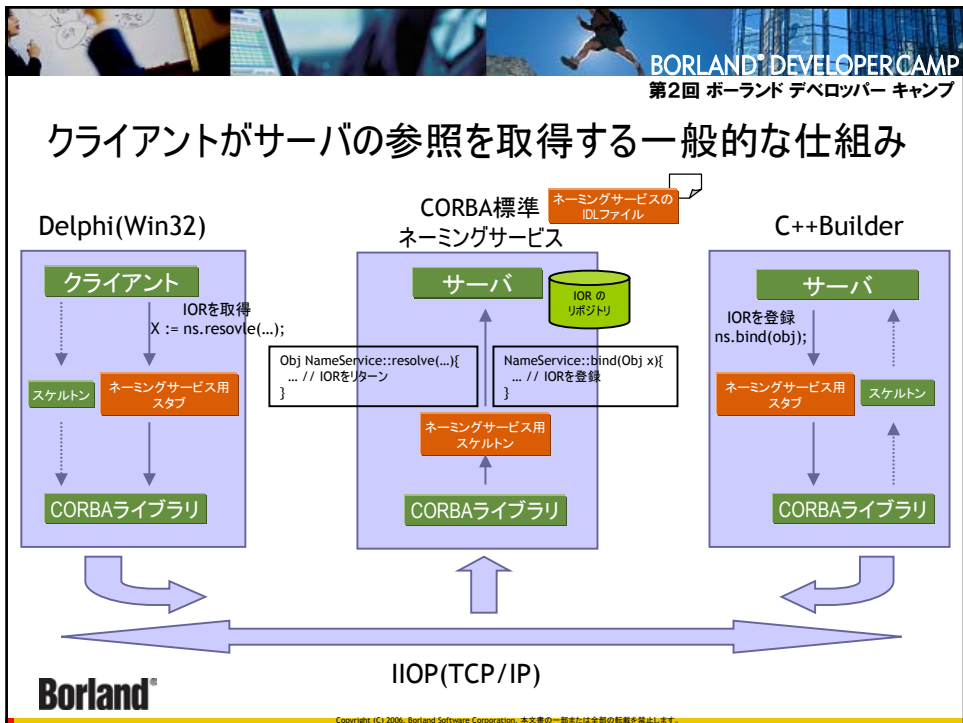
Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

クライアントがサーバを呼び出す一般的な仕組み



Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



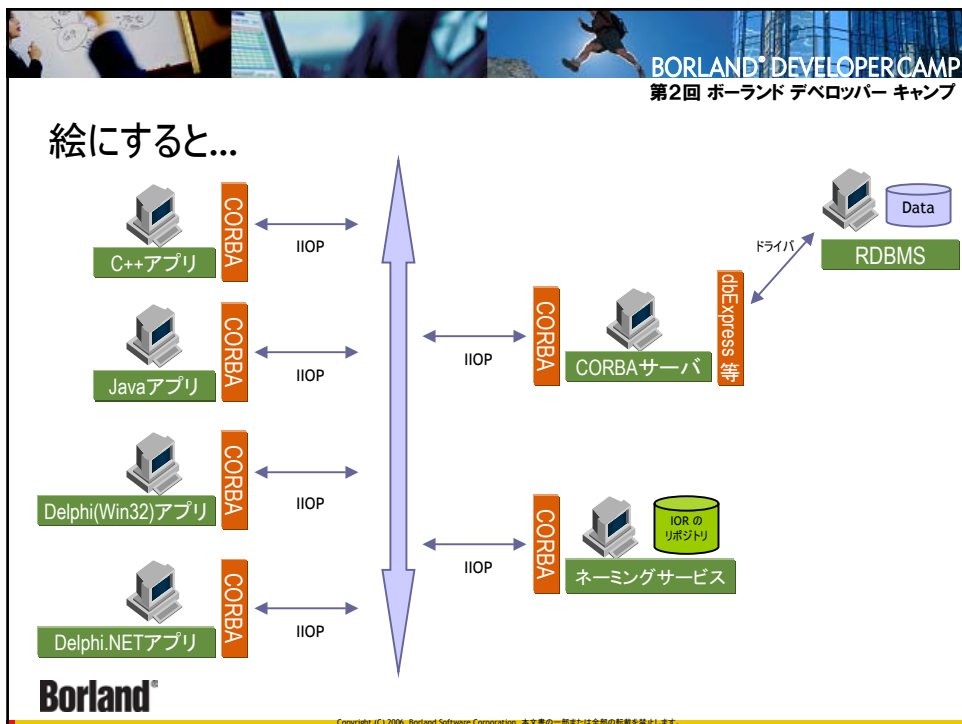
BORLAND® DEVELOPERCAMP
第2回 ボーランド デベロッパー キャンプ

作成するシステムの概要

- CORBA (IIOP) を利用
- CORBA のネーミングサービスを利用
- 従業員データを検索したい
- RDBMS を使用する
- サーバもクライアントもとりあえずスタンドアローンアプリケーションで作る
- CORBAサーバ/CORBAクライアントを違う言語・プラットフォームで置き換え可能
- **とにかく無償またはオープンソースのソフトウェアだけを使用**
 - JBuilder2006 Foundation版
 - Turbo C++ Explorer版
 - Turbo Delphi Explorer版
 - Turbo Delphi.NET Explorer版, Turbo C# Explorer版
 - オープンソースのRDBMS とドライバ
 - オープンソースの各種CORBA (IIOP) 実装

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND® DEVELOPERCAMP
第2回 ボーランド デベロッパー キャンプ

IDLを書いてみる

- 名前は Demo.idl (拡張子は一般的に.idl)
- module(名前空間), interface, メソッド, 引数, 戻り値などを定義する

```
// Demo空間
module Demo {
    // 社員データ
    struct Employee {
        long id; // 社員番号
        wstring data; // 社員情報 ← マルチバイトなのでwstring(UNICODE)
    };
    // アプリケーション例外を定義
    exception NoSuchEmployee {
        wstring reason;
    };
    // インターフェース
    interface EmployeeManager {
        // 社員番号で検索する
        Employee findByPrimaryKey(in long id) raises(NoSuchEmployee);
    };
};
```

in はインプット用引数の意味(※) 例外をスローすることがある

戻り値(※)

Borland®

(※)CORBAでは一般的にNULL(nil)は扱えません

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND® DEVELOPER CAMP

CORBAサーバの実装

C++Builderを使い、IDLで定義したインターフェースを実装します
dbExpress を使って MySQL5.0.x にアクセスします

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。



BORLAND® DEVELOPER CAMP

第2回 ボーランド デベロッパー キャンプ

C++Builderで使えるオープンソースCORBA実装

- TAO (ADAPTIVE Communication Environment(ACE)が基盤)
 - <http://www.cs.wustl.edu/~schmidt/TAO.html>
 - ライセンスは、独自ライセンス。かなり緩やか。
 - 最新バージョンは、2006年7月9日リリースのVer5.5.2

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。

C++Builder2006(BCC5.8.2)でTAOをビルドする

- <http://deuce.doc.wustl.edu/ACE+TAO.zip> をダウンロード
- C:\YACE_wrappers に展開する(デフォルト)
- 以下、コマンドラインでmakeを行う
 - cd C:\YACE_wrappers
 - C:\YACE_wrappers>echo #include "ace/config-win32.h"> ace\config.h
 - C:\YACE_wrappers>set ACE_ROOT=C:\YACE_wrappers
 - C:\YACE_wrappers>set TAO_ROOT=%ACE_ROOT%\TAO
 - C:\YACE_wrappers>set BCBVER=8
 - C:\YACE_wrappers>set PATH=%ACE_ROOT%\bin;%ACE_ROOT%\lib;%PATH%
 - C:\YACE_wrappers>make -f Makefile.bor all
 - cd TAO
 - C:\YACE_wrappers\TAO>make -f Makefile.bor
- C:\YACE_wrappers\bin にIDLコンパイラ等が生成される
- C:\YACE_wrappers\lib に各種DLLとそのインポートライブラリ(.lib)が生成される

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++Builder2006でCORBAサーバを実装する

- コンソールアプリケーションのプロジェクトを新規作成して保存する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - 例: tao_idl.exe Demo.idl
- 生成された DemoC.cpp, DemoS.cpp をプロジェクトに追加する
- プロジェクトの「インクルードファイルの検索パス」に次のフォルダを追加する
 - C:\YACE_wrappers
 - C:\YACE_wrappers\TAO
 - C:\YACE_wrappers\TAO\orbsvcs
- プロジェクトに次のインポートライブラリを追加する
 - ACE_b.lib
 - TAO_b.lib
 - TAO_PortableServer_b.lib
 - TAO_CosNaming_b.lib
 - TAO_AnyTypeCode_b.lib

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++Builder2006でCORBAサーバを実装する(続き)

- CORBAサーバ実装クラスの雛型を作成する(※便宜上ヘッダ内に全部書きます)

EmployeeManagerImpl.h

```
#ifndef EmployeeManagerImplH
#define EmployeeManagerImplH
...
#include "DemoS.h"
...
class EmployeeManagerImpl : public POA_Demo::EmployeeManager
{
public:
    ::Demo::Employee* findByPrimaryKey(::CORBA::Long id)
    {
        ACE_THROW_SPEC ((::CORBA::SystemException, ::Demo::NoSuchEmployee)) {
            ::Demo::Employee* retval = NULL; // 実際にはNULLは返さない
            return retval;
        }
    }
public:
    EmployeeManagerImpl() {}
    virtual ~EmployeeManagerImpl() {}
};
#endif
```

サーバ実装用の親クラス

メソッドをオーバーライドして実装

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++Builder2006でCORBAサーバを実装する(続き)

- main関数を実装します
 - CORBAエンジンの初期化、EmployeeManagerImplインスタンスの生成、CORBAエンジンへの登録処理など

Unit1.cpp

```
static CORBA::ORB_var orb;
...
int main(int argc, char* argv[]) {
    try {
        orb = CORBA::ORB_init(argc, argv);

        CORBA::Object_var obj = orb->resolve_initial_references("RootPOA");
        PortableServer::POA_var rootPOA = PortableServer::POA::_narrow(obj);
        CORBA::PolicyList policies;
        policies.length(2);
        policies[(CORBA::ULong)0] = rootPOA->create_lifespan_policy(PortableServer::PERSISTENT);
        policies[(CORBA::ULong)1] = rootPOA->create_id_assignment_policy(PortableServer::USER_ID);
        PortableServer::POAManager_var poa_manager = rootPOA->the_POAManager();
        PortableServer::POA_var myPOA = rootPOA->create_POA("demo_poa", poa_manager, policies);

        EmployeeManagerImpl managerServant;
        PortableServer::ObjectId_var managerId = PortableServer::string_to_ObjectId("EmployeeManager");
        myPOA->activate_object_with_id(managerId, &managerServant);
        poa_manager->activate();
    }
}
```

CORBAエンジンの初期化など

EmployeeManagerImplインスタンスの生成、CORBAエンジンへの登録

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++Builder2006でCORBAサーバを実装する(続き)

- ネーミングサービスへの登録、CORBAサーバの起動とシャットダウン処理を追加します

```
static CORBA::ORB_var orb;
...
static BOOL __stdcall MyConsoleHandler(DWORD dwCtrlType) {
    if( dwCtrlType == CTRL_C_EVENT ) {
        orb->shutdown((CORBA::Boolean)1); ← CORBAサーバをシャットダウンする
        return TRUE;                               orb->run();から復帰する
    }
    return FALSE;
}
int main(int argc, char* argv[]) {
    ...
    obj = orb->resolve_initial_references("NameService");
    CosNaming::NamingContext_var root_ctx = CosNaming::NamingContext::_narrow(obj);
    CosNaming::Name name;
    name.length(1);
    name[0].id = (const char *)"EmployeeManager";
    name[0].kind = (const char *)"";
    CORBA::Object_var reference = myPOA->servant_to_reference(&managerServant);
    root_ctx->rebind(name, reference); ← ネーミングサービスへの登録

    CORBA::String_var string_ref = orb->object_to_string(reference);
    cout << string_ref << " is ready" << endl;

    SetConsoleCtrlHandler(MyConsoleHandler, TRUE);
    orb->run(); ← CORBAサーバがリクエスト処理を開始する
    ...                               shutdownまでorb->run();から戻らない
}
```

まずはネーミングサービスを起動する

- ネーミングサービスには、TAOの実装を利用します
- ネーミングサービスを起動する
 - set PATH=C:\YACE_wrappers\lib;%PATH%
 - cd C:\YACE_wrappers\TAO\orbsvcs\Naming_Service
 - C:\Y...>Naming_Service.exe -ORBEndpoint iiop://127.0.0.1:12345
- -ORBEndpointで使用するIPアドレスとポートを固定します(12345番)
- -f "ファイル名" オプションにより、リポジトリの永続化が可能
 - ネーミングサービスの再起動後もCORBAサーバの再起動は不要
 - 例: Naming_Service.exe -f c:\Yns.db

試しにCORBAサーバを起動する

- プロジェクトをビルドして.exeを生成する(例: CPPServer.exe)
 - この段階ではまだCORBAサーバの実装はダミー(空っぽ)のまま
- CORBAサーバを起動する
 - set PATH=C:\ACE_wrappers\lib;%PATH%
 - C:\Y...>CPPServer.exe
 - ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
 - ORBSvcConf .\svc.conf
 - ORBEndpoint iiop://127.0.0.1:10000
 - -ORBInitRefでネーミングサービスのIPアドレスとポート番号を指定する
 - -ORBEndpointでCORBAサーバが使用するIPアドレスとポートを固定する(10000番)
 - -ORBSvcConfでCORBAエンジンのスレッド戦略を指定します
 - svc.confの中身(1クライアントごとに1スレッド)
 - static Server_Strategy_Factory "-ORBConcurrency thread-per-connection"
 - スレッドプールを利用するにはACEライブラリを使用するため、ソースコードを一部書き換えます

Borland®


Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

MySQL 5.0.x

- RDBMSとして MySQL 5.0.x を使用します
- MySQL
 - <http://www.mysql.com/>
 - ライセンスは、GPL または Commercial License。
 - 最新バージョンは、2006年7月27日リリースのVer5.0.24

Borland®


Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




BORLAND® DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ

MySQL 5.0.x 用オープンソースdbExpressドライバ

- C++Builder (CORBAサーバ) からMySQLへの接続には dbExpress を使用します
- BDS2006 Pro版, Turbo Pro版, Turbo Exp版のdbExpressドライバの制限
 - MySQL 5.0.x には接続できません
 - localhostのMySQLにしか接続できません(?)
- DbExpress drivers for MySQL V5.0
 - http://www.justsoftwaresolutions.co.uk/delphi/dbexpress_and_mysql_5.html
 - ライセンスは、LGPL。
 - 最新バージョンは、2006年8月3日リリースのもの
 - 基本的にDelphi7用


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

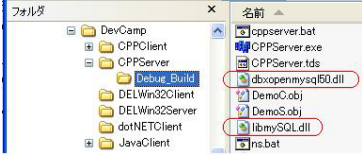



BORLAND® DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ


MySQLとdbExpressドライバのインストール

- MySQL 5.0.24 のインストール
 - <http://dev.mysql.com/downloads/mysql/5.0.html> から mysql-5.0.24-win32.zip をダウンロードして展開します
 - Setup.exe を実行し、C:\¥mysql5 にインストールし、サービスとして登録します
 - サーバのデフォルトの文字コードは sjis(シフトJIS) にしておきます
- dbExpressドライバのインストール
 - http://www.justsoftwaresolutions.co.uk/files/dbxopenmysql5_dll.zip をダウンロードして展開します
 - 次のファイルをC++Builder (CORBAサーバ) のプロジェクトフォルダにコピーします
 - MySQL 5.0.x に付属する libmySQL.dll
 - dbExpressドライバの dbxopenmysql50.dll

(※なお、IDEからMySQL5.0.x に接続するため、
通ったフォルダにコピーしても便利でしょう)




Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



 第2回 ボーランド デベロッパーズ キャンプ

MySQLの起動と初期データ

- MySQLをサービスから起動します
- MySQLに接続し、データベース(demo), テーブル(demotable), 初期データを作成します

```

C:\>C:\mysql\bin\mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 5.0.24-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.


mysql> CREATE DATABASE demo;
Query OK, 1 row affected (0.01 sec)
mysql> use demo;
Database changed
mysql> CREATE TABLE demotable (
  ->   ID integer NOT NULL,
  ->   DATA varchar(32) NOT NULL,
  ->   PRIMARY KEY(ID)
  -> );
Query OK, 0 rows affected (0.36 sec)

mysql> insert into demotable values(1, 'たかはし');
Query OK, 1 row affected (0.06 sec)


mysql> insert into demotable values(2, 'ともひろ');
Query OK, 1 row affected (0.03 sec)
    
```

demotableテーブル

ID	DATA
1	たかはし
2	ともひろ




Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




 第2回 ボーランド デベロッパーズ キャンプ

C++BuilderのCORBAサーバからMySQLに接続する

- プロジェクトに「データモジュール」を追加する
- TSQLConnectionコンポーネントを配置し、プロパティを設定する
- TSQLQueryコンポーネントを配置し、プロパティを設定する

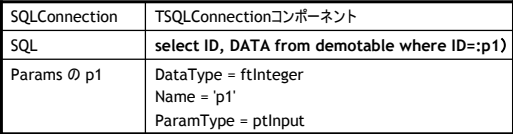



データモジュール




SQLConnection1_SqlQuery1

ConnectionName	MySQLConnection (任意)
DriverName	MySQL (任意)
GetDriverFunc	getSQLDriverMYSQL50
LibraryName	dbxopenmysql50.dll
LoginPrompt	False
Params	HostName=localhost Database=demo User_Name=root Password=pasword
VendorLib	libmysql.dll



パラメータ付きクエリー
 

SQLConnection	TSQLConnectionコンポーネント
SQL	select ID, DATA from demotable where ID=:p1
Params の p1	DataType = ftInteger Name = 'p1' ParamType = ptInput



Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++BuilderのCORBAサーバからMySQLに接続する(続き)

- 今回使用した dbExpressドライバ「dbxopenmysql50.dll」の注意点
 - Delphi7のdbExpressドライバインターフェースに準拠している
 - BDS2006, Turbo でもテストされており、利用可能
 - すべての機能が実装されているわけではない
- TSQLConnectionの接続プロパティ(Params)に ServerCharSet を実装してもらった
 - MySQLサーバのテーブルの文字コードを指定する(例: SJIS)
 - 例: ServerCharSet=SJIS
 - Delphi7では正しく認識されるが、BDS2006, Turbo では認識されない(VCLのバグ)
 - QCで報告済み<http://qc.borland.com/wc/qcmain.aspx?d=23693>
 - BDS2006, Turbo では、別途SQL文 “SET CHARACTER SET SJIS” を実行する
 - 例: データモジュールに初期化用のメソッドを追加


```
void __fastcall TDataModule2::Init()
{
    SQLConnection1->ExecuteDirect("SET CHARACTER SET SJIS");
}
```

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++BuilderのCORBAサーバからMySQLに接続する(続き)

- CORBAサーバの実装クラスでデータモジュールを使用する
 - 便宜上privateなsubメソッドを追加します

EmployeeManagerImpl.h

```
class EmployeeManagerImpl : public POA_Demo::EmployeeManager {
private:
    ::Demo::Employee* findByPrimaryKeySub(::CORBA::Long id) {
        auto_ptr<TDataModule2> data(new TDataModule2(NULL));
        data->Init();
        data->SQLQuery1->ParamByName("p1")->AsInteger = id; // パラメータの社員番号をセット
        data->SQLQuery1->Prepared = true;
        data->SQLQuery1->Open();
        try {
            if (!data->SQLQuery1->Eof)
            {
                ::Demo::Employee_var retval = new ::Demo::Employee(); // 戻り値
                int id = data->SQLQuery1->FieldByName("ID")->AsInteger;
                WideString str = data->SQLQuery1->FieldByName("DATA")->AsString; // UNICODEに変換
                retval->id = id;
                retval->data = ::CORBA::wstring_dup((CORBA::WChar*)str);
                return retval._retn(); // 社員データが見つかった
            }
        }
        _finally {
            data->SQLQuery1->Close();
        }
        return NULL; // 社員データが見つからなかった
    }
}
```

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++BuilderのCORBAサーバからMySQLに接続する(続き)

- subメソッド内で発生する恐れある例外を処理します
- subメソッドの戻り値に応じて、
 - 社員データ(::Demo::Employee)のポインタをCORBAのライブラリにリターンする
 - アプリケーション例外(::Demo::NoSuchEmployee)をスローする

EmployeeManagerImpl.h

```
...
public:
    ::Demo::Employee* findByPrimaryKey(::CORBA::Long id)
        ACE_THROW_SPEC ((::CORBA::SystemException, ::Demo::NoSuchEmployee)) {
        ::Demo::Employee* retval = NULL;
        try {
            retval = findByPrimaryKeySub(id);
        }
        catch(const Exception& ex) {
            throw ::CORBA::UNKNOWN();
        }
        catch(...) {
            throw ::CORBA::UNKNOWN();
        }
        if (retval == NULL)
            throw ::Demo::NoSuchEmployee(L"見つかりませんでした");
        return retval;
    }
...

```

CORBAのシステム例外
 ユーザ定義の例外
 戻り値のインスタンスはCORBAのライブラリが解放します

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

今回のCORBAサーバの実装で足りないもの

- 便宜上、以下の機能を実装していません
 - トランザクション処理(コミット,ロールバック)
 とか
 - RDBMSへの接続をプールする機能(コネクションプール)
 とか
 - データモジュールのインスタンス自体をプールする機能(インスタンスプール)
 とか
 - クライアントの認証(直接EmployeeManagerを取得させない)
 とか...

```
module Demo {
    ...
    interface EmployeeManager {
        Employee findByPrimaryKey(in long id) raises(NoSuchEmployee);
    };

    exception LoginError {
        wstring reason;
    };

    interface EmployeeManagerHome {
        EmployeeManager create(in string id, in string passwd) raises(LoginError);
    };
}

```

ネーミングサービスにはEmployeeManagerHomeのみを登録する
 認証が成功した場合に EmployeeManager の interface を返す



BORLAND® DEVELOPERCAMP

CORBAクライアントの実装

IDLで定義したインターフェースをスタブを利用するだけ

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。



BORLAND® DEVELOPERCAMP

C++Builder版クライアント

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。

C++Builderで作るCORBAクライアント

- CORBAサーバの実装に利用した TAO を利用
- コンソールアプリケーションのプロジェクトを新規作成して保存する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - 例: tao_idl.exe Demo.idl
- 生成された DemoC.cpp をプロジェクトに追加する
- プロジェクトの「インクルードファイルの検索パス」に次のフォルダを追加する
 - C:\YACE_wrappers
 - C:\YACE_wrappers\TAO
 - C:\YACE_wrappers\TAO\orbsvcs
- プロジェクトに次のインポートライブラリを追加する
 - ACE_b.lib
 - TAO_b.lib
 - TAO_CosNaming_b.lib
 - TAO_AO_TypeCode_b.lib

Borland

AO_TypeCode_b.lib

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

C++Builderで作るCORBAクライアント(続き)

- main関数で、エンジン初期化、ネーミングサービスから参照を取得、メソッド呼び出しを行う

```

...
#include "DemoC.h"
...
int main(int argc, char* argv[]) {
    ...
    CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);
    CORBA::Object_var obj = orb->resolve_initial_references("NameService");
    CosNaming::NamingContext_var root_ctx = CosNaming::NamingContext::_narrow(obj);
    CosNaming::Name name;
    name.length(1);
    name[0].id = (const char *) "EmployeeManager";
    name[0].kind = (const char *) "";
    CORBA::Object_var reference = root_ctx->resolve(name);
    Demo::EmployeeManager_var manager = Demo::EmployeeManager::_narrow(reference);

    Demo::Employee_var emp1 = manager->findByPrimaryKey(1);
    AnsiString data = (const wchar_t*) emp1->data;
    cout << data.c_str() << endl;

    try {
        Demo::Employee_var emp3 = manager->findByPrimaryKey(3);
    }
    catch(const Demo::NoSuchEmployee& ex) {
        AnsiString reason = (const wchar_t*) ex.reason;
        cout << reason.c_str() << endl;
    }
}
    
```


エンジンの初期化

ネーミングサービスから参照を取得

ID=1 で検索

ID=3 で検索するが、例外が発生


Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




BORLAND® DEVELOPER CAMP
第2回 ボーランド デベロッパー キャンプ

C++Builder版CORBAクライアントを起動する

- プロジェクトをビルドして.exeを生成する(例: CPPClient.exe)
- MySQL、ネーミングサービスおよび、CORBAサーバを起動しておく
- CORBAクライアントを起動する
 - set PATH=C:\ACE_wrappers\lib;%PATH%
 - C:\>CPPClient.exe
 - ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
- -ORBInitRefでネーミングサービスのIPアドレスとポート番号を指定する



Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁じます。



BORLAND® DEVELOPER CAMP

Java(JBuilder)版クライアント

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁じます。

Javaで使える無償またはオープンソースCORBA実装

- Sun製JDKに元々含まれているCORBA実装
- OpenORB
 - <http://sourceforge.net/projects/openorb>
 - ライセンスは、OpenORB Community Software License他
 - 最新バージョンは、2005年4月3日リリースのVer1.4.0(？)
 - J2EEサーバ Apache Geronimo に含まれている
- JacORB
 - <http://www.jacorb.org/>
 - ライセンスは、LGPL他
 - 最新バージョンは、2006年5月13日リリースのVer2.2.4(※)
 - J2EEサーバ JBoss に含まれている
- 今回は、JDK付属(Sun製)に代えて、JacORBを利用します

Borland®

※2.2.4は、不具合でJDK1.4.2, 1.3.1では動作しません


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

Java(JBuilder)で作るCORBAクライアント

- JacORBのインストール
 - <http://www.jacorb.org/releases/2.2.3/JacORB-2.2.3-binary.zip>をダウンロードして展開するだけ(デフォルトは、C:\¥JacORB-2.2.3)

Borland®

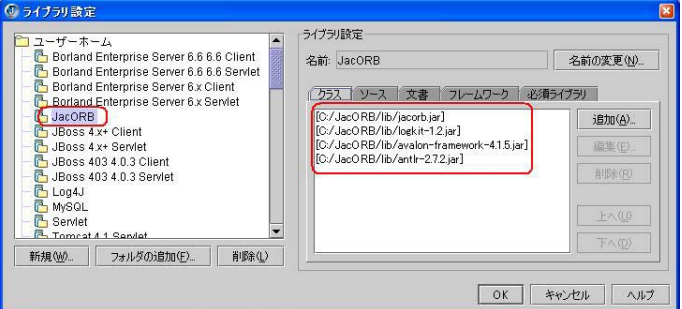
Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




 第2回 ボーランド デベロッパーズ キャンプ

Java (JBuilder) で作るCORBAクライアント(続き)

- JBuilderの[ライブラリ設定]で[JacORB]を作成し、ライブラリ(.jar)を設定する
- 新規にプロジェクトを作成し、空のアプリケーションを作成する
- プロジェクトの[必須ライブラリ]に[JacORB]を追加する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - C:\¥...>set CLASSPATH=C:\¥JacORB¥lib¥idl.jar;C:\¥JacORB¥lib¥logkit-1.2.jar
 - C:\¥...>java org.jacorb.idl.parser -d .¥src Demo.idl



Copyright © 2006, Borland Software Corporation. 本文書の一覧または本製品の複製を禁じます。



 第2回 ボーランド デベロッパーズ キャンプ

Java (JBuilder) で作るCORBAクライアント(続き)

- mainメソッドを記述するが、JacORBに依存したコードはありません。

```

package javaclient;
public class JavaClientMain {
    public static void main(String[] args) {
        try {
            org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);

            org.omg.CORBA.Object root_obj = orb.resolve_initial_references("NameService");
            org.omg.CosNaming.NamingContextExt root_ctx = org.omg.CosNaming.NamingContextExtHelper.narrow(root_obj);
            org.omg.CORBA.Object mgr_obj = root_ctx.resolve(root_ctx.to_name("EmployeeManager"));
            Demo.EmployeeManager manager = Demo.EmployeeManagerHelper.narrow(mgr_obj);

            Demo.Employee emp1 = manager.findByPrimaryKey(1);
            System.out.println(emp1.data);

            try {
                Demo.Employee emp3 = manager.findByPrimaryKey(3);
            } catch (Demo.NoSuchEmployee ex) {
                System.out.println(ex.reason);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
  
```


エンジンの初期化

ネーミングサービスから参照を取得

ID=1 で検索

ID=3 で検索するが、例外が発生

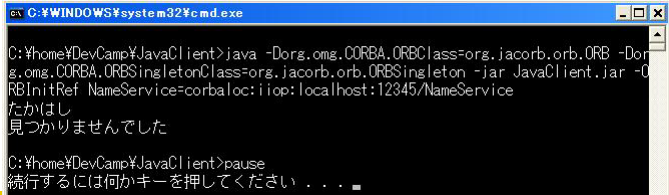
Copyright © 2006, Borland Software Corporation. 本文書の一覧または本製品の複製を禁じます。



BORLAND® DEVELOPER CAMP
第2回 ボーランド デベロッパーズ キャンプ

Java (JBuilder) 版CORBAクライアントを起動する

- JBuilderのアーカイブビルダー機能を使い、クライアントを一つの.jarファイルにまとめる
- 標準のSunのCORBA実装ではなく、JacORBのCORBA実装に切り替えるには？
 - 一般的にはJDKで規定されている次のシステムプロパティを設定する
 - Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB
 - Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton
- CORBAクライアントを起動する
 - C:¥...>java -Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB -Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton -jar JavaClient.jar -ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
- -ORBInitRefでネーミングサービスのIPアドレスとポート番号を指定する



Borland®



BORLAND® DEVELOPER CAMP

Delphi Win32版クライアント

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。

Delphi Win32で使えるオープンソースCORBA実装

- MTDORB — Multi-Threaded Delphi ORB
 - http://www.millennium-group.ru/index/lang/eng/parent_id/30/level/1
 - <http://sourceforge.net/projects/mtdorb>
 - ライセンスは、BSD License。
 - 最新バージョンは、2004年7月21日リリースのVer1.0 beta6
 - Kylixでも利用可能

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。


Delphi2006(Win32)でMTDORBをビルドする(※)

- Sourceforgeのミラーサイトから
http://jaist.dl.sourceforge.net/sourceforge/mtdorb/mtdorb_1.0beta6.tar.gz
をダウンロードして C:¥mtdorb に展開する
- C:¥mtdorb にビルド設定ファイル「build.properties」を作成する
 - 付属の「C:¥mtdorb¥build.properties.pattern」をコピーして以下のように編集する

```
dir.delphi=C:¥¥Borland¥¥BDS¥¥4.0
delphi.version7=off
# Location
dir.bpl=C:¥¥Documents and Settings¥¥ken¥¥My Documents¥¥Borland Studio Projects¥¥Bpl
# Build
build.debug=off
```
- Apache Ant(ant.bat)が利用できるようにする(※JBuilder2006に付属しています)
 - C:¥...>set PATH=C:¥JBuilder2006¥thirdparty¥ant¥bin;%PATH%
- Antでビルド(コンパイル)する
 - cd C:¥mtdorb
 - C:¥mtdorb>ant.bat
- C:¥mtdorb¥dcu に各種.dcuが、「dir.bpl」配下に.bplと.dcpが生成される
※ビルドにはコマンドラインコンパイラ(dcc32.exe)が必要ですが、Turbo Explorerには付属しません
そのためIDEを利用して、必要に応じて.pasファイルをコンパイルし、.dcuを生成する必要があります

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。





BORLAND® DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ

Delphi Win32で作るCORBAクライアント

- コンソールアプリケーションのプロジェクトを新規作成して保存する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - 例:


```
set PATH=C:\Ymtdorb\build;%PATH%
C:\Y...>idltopas.exe Demo.idl
```
- 生成された Demo.pas, Demo_int.pas をプロジェクトに追加する
- プロジェクトの「ディレクトリ/条件」の「検索パス」に次のフォルダを追加する
 - C:\Ymtdorb\Ydcu
- 実行時に必要となるMTDORB付属のユーティリティライブラリ (MTDORB_UCUtils.dll)をプロジェクトフォルダにコピーする
 - UNICODE処理のために必要
 - C:\Y...>copy C:\Ymtdorb\build\YMTDORB_UCUtils.dll <プロジェクトフォルダ>


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND® DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ

Delphi Win32で作るCORBAクライアント(続き)

DELWin32Client.dpr

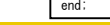
```

...
uses
  SysUtils, Classes, orb_int, orb, orbtypes, cosnaming, oosnaming_int,
  Demo in 'Demo.pas',
  Demo_int in 'Demo_int.pas';
} 必要なユニットの参照を追加する

var
  orb: IORB;
  manager: IDemo_EmployeeManager;

procedure init;
var
  i: Integer;
  props: TStrings;
  root_ref: IORBObject;
  root_ctx: INamingContext;
  obj_name: TName;
  obj_ref: IORBObject;
begin
  props := TStringList.Create();
  for i := 1 to ParamCount do
    props.Add(ParamStr(i));
  orb := ORB_Init(props);
  FreeAndNil(props);
  root_ref := orb.resolve_initial_reference('NameService');
  root_ctx := TNamingContext._narrow(root_ref);
  SetLength(obj_name, 1);
  obj_name[0].id := 'EmployeeManager';
  obj_name[0].kind := '';
  obj_ref := root_ctx.resolve(obj_name);
  manager := TDemo_EmployeeManager._narrow(obj_ref);
end;

```


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

Delphi Win32で作るCORBAクライアント(続き)

DELWin32Client.dpr

```
...
var
  ...
  manager: IDemo_EmployeeManager;
  ...
procedure test;
var
  emp: TDemo_Employee;
begin
  emp := manager.FindByPrimaryKey(1); ← ID=1 で検索
  Writeln(emp.data);

  try
    emp := manager.FindByPrimaryKey(3); ← ID=3 で検索するが、例外が発生
  except on E: TDemo_NoSuchEmployee do
    Writeln(E.reason);
  end;
end;

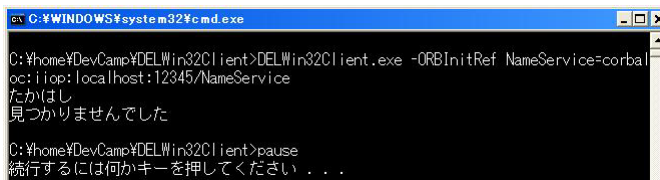
begin
  init;
  test;
end.
```

Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

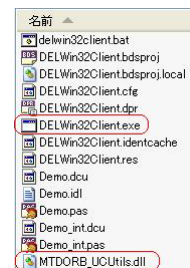
Delphi Win32版CORBAクライアントを起動する

- プロジェクトをビルドして.exeを生成する(例: DELWin32Client.exe)
- CORBAクライアントを起動する
 - ビルドした.exeと付属のMTDORB_UCUtils.dllがあれば良い
 - C:\Y...\>DELWin32Client.exe
 - ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
 - -ORBInitRefでネーミングサービスのIPアドレスとポート番号を指定する



```
C:\WINDOWS\system32\cmd.exe
C:\Yhome\DevCamp\DELWin32Client>DELWin32Client.exe -ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
たか(は)し
見つかりませんでした

C:\Yhome\DevCamp\DELWin32Client>pause
続行するには何かキーを押してください . . .
```



Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



Delphi .NET版クライアント

C#Builderでも作り方は同じです

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。




Delphi .NET(.NET1.1)で使えるオープンソースCORBA実装

- IIOP.NET
 - <http://iiop-net.sourceforge.net/>
 - <http://sourceforge.net/projects/iiop-net>
 - ライセンスは、LGPL。
 - 最新バージョンは、2006年8月22日リリースのVer1.9.0 rc0
 - .NET2.0でも利用可能

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。




 第2回 ボーランド デベロッパー キャンプ


.NET1.1でIIOP.NETをビルドする

- Sourceforgeのミラーサイトから
<http://jaist.dl.sourceforge.net/sourceforge/iiop-net/IIOPNet.src.1.9.0.rc0.zip>
 をダウンロードして C:\IIOPNet.src.1.9.0.rc0 に展開する
- JDK(Java Development Kit)1.4.x or 1.5.0 から orb.idl, ir.idl を借りてくる
 - 例:


```
C:\>copy C:\j2sdk1.4.2\lib\orb.idl C:\IIOPNet.src.1.9.0.rc0\IDL
C:\>copy C:\j2sdk1.4.2\lib\ir.idl C:\IIOPNet.src.1.9.0.rc0\IDL
```
- .NET1.1 SDK に付属する nmake コマンドを使用してビルドする(C#コンパイラを使用)
 - cd C:\IIOPNet.src.1.9.0.rc0
 - C:\IIOPNet.src.1.9.0.rc0>nmake build
- C:\IIOPNet.src.1.9.0.rc0\IDLToCLSCompiler\IDLCompiler\bin にIDLコンパイラが生成される
- C:\IIOPNet.src.1.9.0.rc0\IIOPChannel\bin に、CORBA用のアセンブリ「IIOPChannel.dll」が生成される



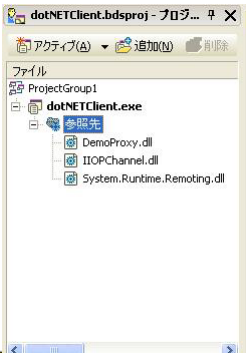
Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




 第2回 ボーランド デベロッパー キャンプ


Delphi .NETで作るCORBAクライアント

- コンソールアプリケーションのプロジェクトを新規作成して保存する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - set PATH=C:\IIOPNet.src.1.9.0.rc0\IDLToCLSCompiler\IDLCompiler\bin;%PATH%
 - C:\>IDLToCLSCompiler.exe DemoProxy Demo.idl
- IDLコンパイラ「IDLToCLSCompiler」は、IDLファイルから直接.NETアセンブリ(例: DemoProxy.dll)を生成します。スタブ用のソースコードを生成するわけではありません。
- プロジェクトの「参照」に次のアセンブリを追加する
 - 生成された DemoProxy.dll
 - IIOP.NETの IIOPChannel.dll
 - System.Runtime.Remoting.dll





Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ

Delphi .NETで作るCORBAクライアント(続き)

dotNETClient.dpr

```

...
uses
  SysUtils,
  System.Runtime.Remoting,
  System.Runtime.Remoting.Channels,
  Ch.Elca.Iiop,
  Ch.Elca.Iiop.Services,
  omg.org.CosNaming,
  Demo;

var
  manager: EmployeeManager;

procedure init;
var
  channel: IiopClientChannel;
  nc: NamingContext;
  name: array of NameComponent;
begin
  channel := IiopClientChannel.Create();
  ChannelServices.RegisterChannel(channel);


  nc := RemotingServices.Connect(typeof(NamingContext), ParamStr(1)) as NamingContext;
  SetLength(name, 1);
  name[0] := NameComponent.Create('EmployeeManager', '');
  manager := nc.resolve(name) as EmployeeManager;
end;
...
      
```

} 必要なユニットの参照を追加する

} IIOPチャネルを生成し登録する

} ネーミングサービスから
CORBAオブジェクトの参照を取得する

Copyright © 2006, Borland Software Corporation. 本スライドの複製は禁じます。



BORLAND DEVELOPER CAMP
 第2回 ボーランド デベロッパーズ キャンプ

Delphi .NETで作るCORBAクライアント(続き)

dotNETClient.dpr

```

...
var
  manager: EmployeeManager;
...
procedure test;
var
  emp: Employee;
begin
  emp := manager.findByPrimaryKey(1);
  Writeln(emp.data);


  try
    emp := manager.findByPrimaryKey(3);
  except on E: Demo.NoSuchEmployee do
    Writeln(E.reason);
  end;
end;

begin
  init;
  test;
end.
      
```

← ID=1 で検索

← ID=3 で検索するが、例外が発生

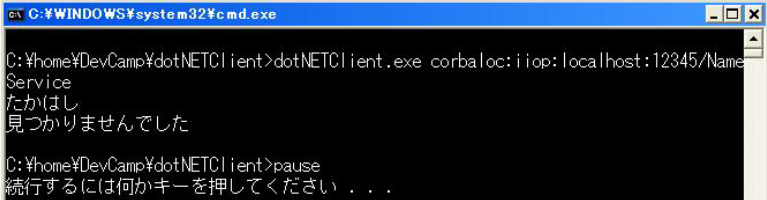
Copyright © 2006, Borland Software Corporation. 本スライドの複製は禁じます。




BORLAND® DEVELOPER CAMP
第2回 ボーランド デベロッパー キャンプ

Delphi .NET版CORBAクライアントを起動する


- プロジェクトをビルドして.exeを生成する(例: dotNETClient.exe)
- CORBAクライアントを起動する
 - ビルドした.exe と DemoProxy.dll, IIOPChannel.dll が必要
 - C:¥...>dotNETClient.exe corbaloc:iiop:localhost:12345/NameService
- コマンドラインオプションの文字列でネーミングサービスのIPアドレスとポート番号を指定する
 - プログラム内でそのオプション文字列を利用してネーミングサービスを見つけます
nc := RemotingServices.Connect(typeof(NamingContext), ParamStr(1)) as NamingContext;



```
C:\WINDOWS\system32\cmd.exe
C:\home\DevCamp\dotNETClient>dotNETClient.exe corbaloc:iiop:localhost:12345/NameService
たかひし
見つかりませんでした
C:\home\DevCamp\dotNETClient>pause
続行するには何かキーを押してください . . .
```




Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND® DEVELOPER CAMP
第2回 ボーランド デベロッパー キャンプ

今回のCORBAクライアントの実装で足りないもの

- 便宜上、以下の機能を実装していません
 - GUI
 - CORBAのシステム例外など、あらゆる例外の補足
 - 常に例外の発生を意識する必要があります



Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND® DEVELOPERCAMP

Delphi Win32版CORBAサーバ

Delphi Win32を使い、IDLで定義したインターフェースを実装します
dbExpressを使って、MySQL 5.0.x にアクセスします
オープンソースのO/Rマッパー「InstantObjects 2.0」を利用してみます

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。



BORLAND® DEVELOPERCAMP
第2回 ボーランド デベロッパー キャンプ

Delphi Win32でCORBAサーバを実装する

- 先ほどCORBAクライアントに利用した「MTDORB」でCORBAサーバを実装します
 - C++Builderで作成したCORBAサーバを、Delphi Win32版に取り替えます
- コンソールアプリケーションのプロジェクトを新規作成して保存する
- Demo.idl をプロジェクトフォルダにコピーしてコンパイルする
 - 例:

```
set PATH=C:\Ymtdorb\build;%PATH%
C:\Y...>idltopas.exe Demo.idl
```
- 生成された Demo.pas, Demo_int.pas をプロジェクトに追加する
- プロジェクトの「ディレクトリ/条件」の「検索パス」に次のフォルダを追加する
 - C:\Ymtdorb\dcu
- 実行時に必要となるMTDORB付属のユーティリティライブラリ (MTDORB_UCUtils.dll)をプロジェクトフォルダにコピーする
 - UNICODE処理のために必要
 - C:\Y...>copy C:\Ymtdorb\build\YMTDORB_UCUtils.dll <プロジェクトフォルダ>

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。

とりあえずCORBAサーバの雛型を作成する

- EmployeeManagerUnit.pas を新規作成して、空の実装を行う

```
...
uses
  SysUtils, orbtypes, exceptions, Demo, Demo_int;

type
  TEmployeeManagerImpl = class(TDemo_EmployeeManager_serv)
  protected
    function findByPrimaryKey(const id: long): TDemo_Employee; override;
  end;

implementation

function TEmployeeManagerImpl.findByPrimaryKey(const id: long): TDemo_Employee;
begin
  ;
end;
```

サーバ実装用の親クラス

メソッドをオーバーライドして実装

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

とりあえずCORBAサーバの雛型を作成する(続き)

DELWin32Server.dpr

```
...
uses
  SysUtils, Classes, Windows,
  orb_int, orb, orbtypes, cosnaming, cosnaming_int, poa, poa_impl, poa_int, policy_int,
  Demo_int in 'Demo_int.pas',
  Demo in 'Demo.pas',
  EmployeeManagerUnit in 'EmployeeManagerUnit.pas';

var
  orb: IORB;
  i: Integer;
  props: TStrings;
  poa_ref: IORBObject;
  root_poa: IPOA;
  poa_mgr: IPOAManager;
  policies: PolicyList;
  my_poa: IPOA;
  impl: TEmployeeManagerImpl;
  obj_id: ObjectID;
  root_ref: IORBObject;
  root_ctx: INamingContext;
  obj_name: TName;
  obj_ref: IORBObject;
  ior_str: string;

function MyConsoleHandler(CtrlType: DWord): Bool; stdcall;
begin
  Result := False;
  if CtrlType = CTRL_C_EVENT then
  begin
    orb.shutdown(True);
  end
end;
...
```

必要なユニットの参照を追加する

CORBAサーバをシャットダウンする

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

とりあえずCORBAサーバの雛型を作成する(続き)

DELWin32Server.dpr

```

begin
  props := TStringList.Create();
  for i := 1 to ParamCount do
    props.Add(ParamStr(i));
  end;
  orb := ORB_Init(props);
  FreeAndNil(props);
} CORBAエンジンを初期化する

poa_ref := orb.resolve_initial_reference('RootPOA');
root_poa := POA_narrow(poa_ref);
poa_mgr := root_poa.the_POAManager();
setlength(policies, 2);
policies[0] := root_poa.create_lifespan_policy(PERSISTENT);
policies[1] := root_poa.create_id_assignment_policy(USER_ID);
my_poa := root_poa.create_POA('demo_poa', poa_mgr, policies);

impl := TEmployeeManagerImpl.Create;
obj_id := String_ToObjectId('EmployeeManager');
my_poa.activate_object_with_id(obj_id, impl);
poa_mgr.activate;
} TEmployeeManagerImplインスタンスの生成、CORBAエンジンへの登録

root_ref := orb.resolve_initial_reference('NameService');
root_ctx := TNamingContext_narrow(root_ref);
setlength(obj_name, 1);
obj_name[0].id := 'EmployeeManager';
obj_name[0].kind := '';
obj_ref := my_poa.servant_to_reference(impl);
root_ctx.rebind(obj_name, obj_ref);
} ネーミングサービスへの登録

ior_str := orb.object_to_string(obj_ref);
writeln(ior_str + ' is ready');


SetConsoleCtrlHandler(@MyConsoleHandler, True);
orb.run;
end.
CORBAサーバがリクエスト処理を開始する
shutdownまでorb.run;から戻らない

```

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

試しにCORBAサーバを起動する

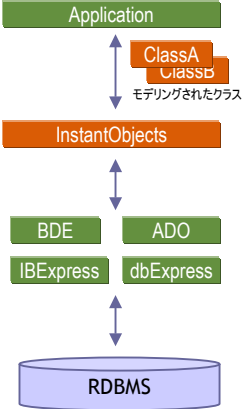
- プロジェクトをビルドして.exeを生成する(例: DELWin32Server.exe)
 - この段階ではまだCORBAサーバの実装はダミー(空っぽ)のまま
- CORBAサーバを起動する
 - C:\>DELWin32Server.exe
 - ORBIIOPAddr inet:127.0.0.1:10000
 - ORBConnectionsLimit 100
 - ORBConcModel thread_per_connection
 - ORBThreadPoolTimeout 10
 - ORBInitRef NameService=corbaloc:iiop:localhost:12345/NameService
 - -ORBIIOPAddrでCORBAサーバが使用するIPアドレスとポートを固定する(10000番)
 - -ORBInitRefでネーミングサービスのIPアドレスとポート番号を指定する
 - -ORBConcModel, -ORBConnectionsLimit, -ORBThreadPoolTimeout でCORBAエンジンのスレッド戦略を指定します
 - 1クライアントごとに1スレッド、100クライアント(100スレッド)まで接続可能、クライアント終了後のスレッドの破棄を10秒間隔





 第2回 ボーランド デベロッパーズ キャンプ

InstantObjects 2.0

- Delphi5～Delphi2006(Win32)に対応した、オープンソースの Object Persistence Framework
 - Object/Relational マッピングツール
 - <http://www.instantobjects.org/>
 - <http://sourceforge.net/projects/instantobjects/>
 - IDEのプラグインとして統合されて動作する
 - Turbo Explorer は ×
 - ライセンスは、MPL1.1。
 - 最新バージョンは、2006年8月7日リリースのVer2.0



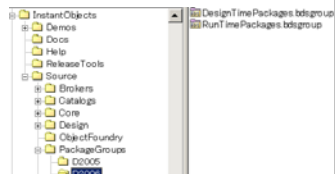

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。




 第2回 ボーランド デベロッパーズ キャンプ

Delphi2006(Win32)にInstantObjectsをインストール

- Sourceforgeのミラーサイトから
 - <http://heanet.dl.sourceforge.net/sourceforge/instantobjects/InstantObjects-2.0-src.zip> をダウンロードして、C:\¥InstantObjects に展開する
- IDEメニューから、次のプロジェクトグループファイルを開いて、すべてのプロジェクトをビルドする
 - C:\¥InstantObjects¥Source¥PackageGroups¥D2006¥RunTimePackages.bdsgroup
- IDEメニューから、次のプロジェクトグループファイルを開いて、すべてのプロジェクトをビルドする
 - C:\¥InstantObjects¥Source¥PackageGroups¥D2006¥DesignTimePackages.bdsgroup
- 次のフォルダに、各種の設計時パッケージ(.bpl)および実行時パッケージ(.bpl)が生成されることを確認する
 - C:\¥Documents and Settings¥ログイン名¥My Documents¥Borland Studio Projects¥Bpl
- 次のフォルダ配下に、各種の.dcuファイルが生成されることを確認する
 - C:\¥InstantObjects¥Source



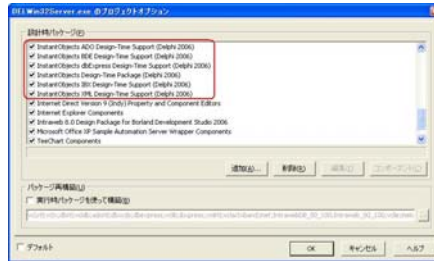
続く...


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

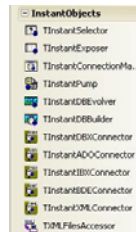
Delphi2006(Win32)にInstantObjectsをインストール(続き)

- メニューから[コンポーネント]-[パッケージのインストール...]を選択し、次の設計時パッケージをインストールする

- DclIOCore_D10.bpl
- DclIOBDE_D10.bpl
- DclIOADO_D10.bpl
- DclIOIBX_D10.bpl
- DclIODBX_D10.bpl
- DclIOXML_D10.bpl



- ツールパレットに「InstantObjects」カテゴリが追加される



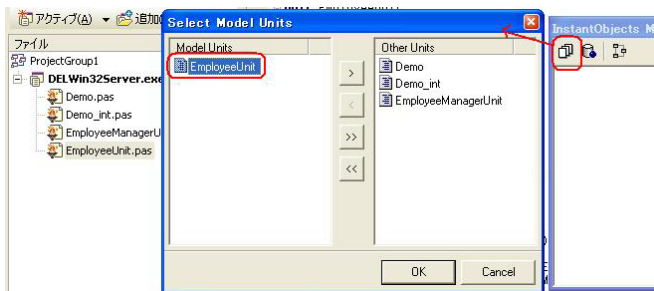
Borland®

Copyright © 2006, Borland Software Corporation. 本文書の一覧表は全製品の機能を表しません。

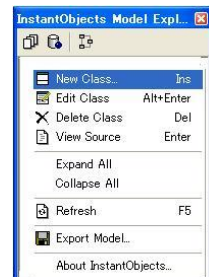
Model Explorerで社員データ用クラスをモデリングする

- 新規ユニットを作成して、EmployeeUnit.pasとしてプロジェクトに追加する
- メニューから[表示]-[InstantObjects Model Explorer]を選択する
- モデリングしたクラスからソースコードを自動生成する際に使用するユニットを選択する
 - 今回は EmployeeUnit.pas のみを選択しました(複数選択可)
- [New Class...]を選択し、クラスのモデリングを開始する

ユニットの選択

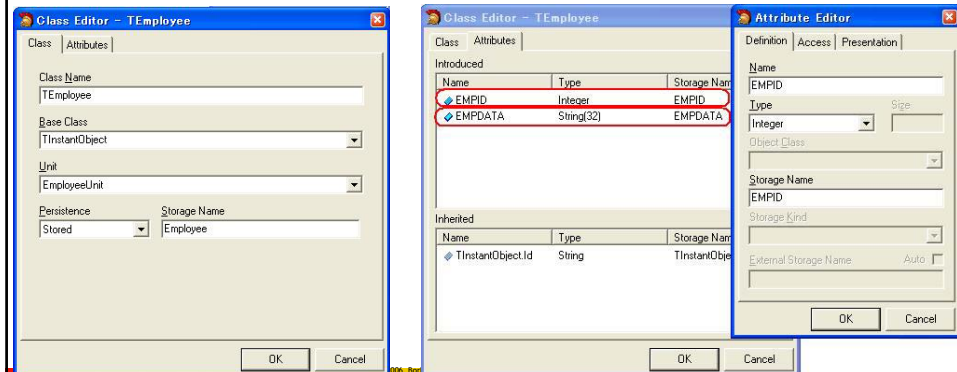


クラスの新規作成



Model Explorerで社員データ用クラスをモデリングする(続き)

- クラス名を「TEmployee」、クラスのソースコード出力するユニットをEmployeeUnit、永続化するテーブル名を「Employee」とする
- TEmployeeクラスのフィールドを追加する
 - 社員番号: フィールド名「EMPID」、データ型「Integer」、テーブル内のカラム名「EMPID」
 - 社員情報: フィールド名「EMPDATA」、データ型「String」、テーブル内のカラム名「EMPDATA」



Model Explorerで社員データ用クラスをモデリングする(続き)

- 設計したTEmployeeクラスがEmployeeUnit.pasに自動生成されます

```
unit EmployeeUnit;

interface

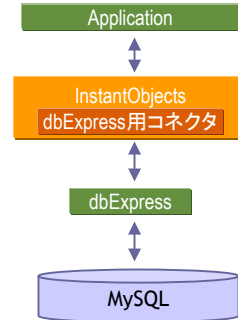
uses
  InstantPersistence;

type
  TEmployee = class(TInstantObject)
  {IOMETADATA stored:
    EMPID: Integer;
    EMPDATA: String(32); }
  _EMPDATA: TInstantString;
  _EMPID: TInstantInteger;
  private
    function GetEMPDATA: string;
    function GetEMPID: Integer;
    procedure SetEMPDATA(const Value: string);
    procedure SetEMPID(Value: Integer);
  published
    property EMPDATA: string read GetEMPDATA write SetEMPDATA;
    property EMPID: Integer read GetEMPID write SetEMPID;
  end;

...
```

データモジュールを作成する

- データモジュールを新規作成し、プロジェクトに追加します
- [dbExpress]カテゴリから[TSQLConnection]コンポーネントを貼り付け、対象のMySQLに接続するためのプロパティを設定します
 - LoginPromptをfalseにするのを忘れずに
- [InstantObjects]カテゴリから[TInstantDBXConnector]コンポーネントを貼り付け、TSQLConnectionに接続します
 - こちらも、LoginPromptをfalseにするのを忘れずに

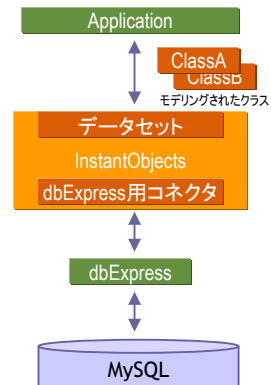
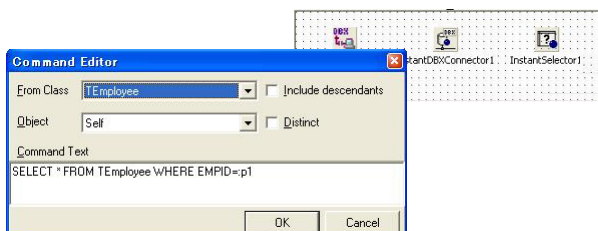


Borland®


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

データセットを作成する

- TEmployeeクラス型を使って「検索」「追加」「削除」「更新」を行うためのデータセットを提供するTInstantSelectorコンポーネントを使用します
- [InstantObjects]カテゴリから[TInstantSelector]コンポーネントを貼り付け、TInstantDBXConnectorに接続します
- TEmployeeクラス型用のデータセットを生成する
「SQLライクな文」を「半」自動生成させます
SELECT * FROM TEmployee WHERE EMPID=:p1
 - 社員番号での検索のためパラメータ :p1 を使います




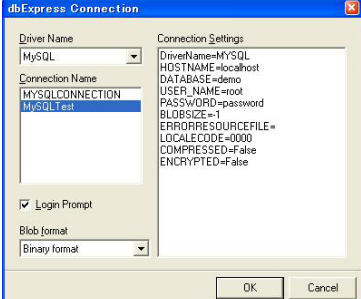

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND* DEVELOPER CAMP
第2回 ボーランド テレロツパー キャンプ


TEmployeeクラス用のテーブルを自動生成させる

- 対象となるデータベースへの接続を行い、テーブルを作成します
 - Database Builderを表示し、メニューから[New]-[dbExpress Connection]を選択します
 - 生成されたConnection名を適用なものに変更します
 - メニューから[Edit]を選択し、接続するMySQLを選択します
 - 最後に「Build」ボタンを押し、テーブルを生成します

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



BORLAND* DEVELOPER CAMP
第2回 ボーランド テレロツパー キャンプ

Database Builderで dbExpress を使用する際の注意点

- Database Builderで接続を行う際、dbExpressのドライバ名も含め、すべての情報を
 - C:\Borland\BDS\4.0\dbExpress\dbxdrivers.ini
 - C:\Borland\BDS\4.0\dbExpress\dbxconnections.ini
 から取得します。
- MySQL用の「DriverName」は、必ず[MySQL]である必要があります。
- そのため、MySQL 5.0.x に接続するためには予め、dbxdrivers.ini の[MySQL]セクションを次のように書き換えておく必要があります。

- なお、IDEから MySQL 5.0.x に接続するため
 - MySQL 5.0.x に付属する libmysql.dll
 - dbExpressドライバの dbxopenmysql50.dll
 をPATHの通ったフォルダにコピーしておく必要があります

dbxdrivers.ini

 [MySQL]
 GetDriverFunc=getSQLDriverMySQL50
 LibraryName=dbxopenmysql50.dll
 VendorLib=libmysql.dll
 HostName=ServerName
 Database=DBNAME
 User_Name=user
 Password=password
 BlobSize=-1
 ErrorResourceFile=
 LocaleCode=0000
 Compressed=False
 Encrypted=False

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

自動生成されたテーブルを確認する

- TEmployeeクラスがEmployeeテーブルにマッピングされました
- Class, Id, UpdateCountは、InstantObjectsにより内部的に使用されるカラム
- UpdateCountカラムは、複数クライアントからの同時更新時に衝突を検出するためのバージョンカラムとして利用されます(楽観的ロック)

```
mysql> use demo
mysql> show tables;
```

demotable
employee

```
mysql> show create table employee;
```

```
CREATE TABLE `employee` (
  `Class` varchar(32) NOT NULL,
  `Id` varchar(32) NOT NULL,
  `UpdateCount` int(11) default NULL,
  `EMPID` int(11) default NULL,
  `EMPDATA` varchar(32) default NULL,
  PRIMARY KEY (`Class`,`Id`)
) ENGINE=InnoDB DEFAULT CHARSET=sjis;
```

Borland®

Copyright © 2006, Borland Software Corporation. 本書の一部または全部の複製を禁じます。

初期データをテーブルに挿入する

- 説明の便宜上、予め、TEmployeeクラスのインスタンスをテーブルに永続化しておきます。

```
procedure TForm1.Button1Click(Sender: TObject);
var
  emp1, emp2: TEmployee;
begin
  emp1 := TEmployee.Create(InstantDBXConnector1);
  emp1.EMPID := 1;
  emp1.EMPDATA := 'たかはし';
  InstantSelector1.AddObject(emp1); // TEmployee型で追加する
  FreeAndNil(emp1);

  emp2 := TEmployee.Create(InstantDBXConnector1);
  emp2.EMPID := 2;
  emp2.EMPDATA := 'ともひろ';
  InstantSelector1.AddObject(emp2); // TEmployee型で追加する
  FreeAndNil(emp2);
end;
```

```
mysql> select * from employee;
```

Class	Id	UpdateCount	EMPID	EMPDATA
TEmployee	B15F548048F0084C88C4B9E561BA3170	1	1	たかはし
TEmployee	D1B7DEEE729E8D479EBAF74AFDC7FEC3	1	2	ともひろ

Borland®

Copyright © 2006, Borland Software Corporation. 本書の一部または全部の複製を禁じます。

Delphi Win32のCORBAサーバからMySQLに接続する

- CORBAサーバの実装クラスでデータモジュールを使用する。(便宜上privateなsubメソッドを追加します)
EmployeeManagerUnit.pas

```
...
function TEmployeeManagerImpl.findByPrimaryKeySub(const id: long): TDemo_Employee;
var
  data: TDataModule1;
  emp: TEmployee;
begin
  data := TDataModule1.Create(nil);
  try
    data.Init;
    data.InstanceSelector1.Params.ParamByName('p1').AsInteger := id;
    data.InstanceSelector1.Open;
    try
      if data.InstanceSelector1.ObjectCount = 1 then // 取得したインスタンス数(1)を確認
      begin
        emp := data.InstanceSelector1.Objects[0] as TEmployee; // データセットからTEmployee型で取り出す
        Result.id := emp.EMPID;
        Result.data := emp.EMPDATA;
        Exit;
      end;
      raise TDemo_NoSuchEmployee.Create('見つかりませんでした'); // 社員データが検索されなかった
    finally
      data.InstanceSelector1.Close;
    end;
  finally
    FreeAndNil(data);
  end;
end;
```


Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。

Delphi Win32のCORBAサーバからMySQLに接続する(続き)

- subメソッド内で発生する恐れある例外を処理します
- subメソッドの戻り値に応じて、
 - 社員データ(TDemo_Employee)をCORBAのライブラリにリターンする
 - アプリケーション例外(TDemo_NoSuchEmployee)をスローする

```
...
type
  TEmployeeManagerImpl = class(TDemo_EmployeeManager_serv)
  ...
  protected
    function findByPrimaryKey(const id: long): TDemo_Employee; override;
  end;
...
function TEmployeeManagerImpl.findByPrimaryKey(const id: long): TDemo_Employee;
begin
  try
    Result := findByPrimaryKeySub(id);
  except
    on noemp_ex: TDemo_NoSuchEmployee do ← ユーザ定義の例外
      raise;
    on general_ex: Exception do
      raise UNKNOWN.Create(general_ex.Message);
    else
      raise UNKNOWN.Create(); ← CORBAのシステム例外
    end;
  end;
end;
```

Copyright © 2006, Borland Software Corporation. 本文書の一部または全部の複製を禁じます。



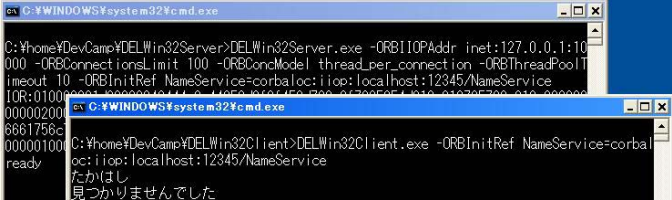
 第2回 ボーランド デベロッパー キャンプ


Delphi Win32のCORBAサーバとクライアントを起動する


- 次のファイルをCORBAサーバのプロジェクト配下にコピーします
 - MySQL 5.0.x に付属する libmySQL.dll
 - dbExpressドライバの dbxopenmysql50.dll
- MySQL、ネーミングサービスおよび、CORBAサーバを起動しておく
- 各種CORBAクライアントを起動する
 - C++Builder(TAO)版 ... OK
 - Java(JacORB)版 ... OK
 - Delphi Win32(MTDORB)版 ... OK
 - Delphi .NET(IIOP.NET)版 ... NG(IIOP.NETの実装の問題で呼び出しが失敗!?)
 - http://sourceforge.net/forum.php?thread_id=1560389&forum_id=274081

名前 ▲

- DataModuleUnit.dcu
- DataModuleUnit.dfm
- DataModuleUnit.pas
- dbxopenmysql50.dll
- DELWin32Server.dpr
- DELWin32Server.exe
- EmployeeManagerUnit.dcu
- EmployeeManagerUnit.dfm
- EmployeeManagerUnit.pas
- EmployeeUnit.dcu
- EmployeeUnit.dfm
- EmployeeUnit.pas
- libmySQL.dll
- MTDORB_UCUtils.dll








 第2回 ボーランド デベロッパー キャンプ

まとめ

- 今回使用したもの
 - Turbo C++, Delphi, Delphi.NET の Explorer版の機能
 - JBuilder
 - MySQL 5.0.x + dbExpressドライバ
 - TAO
 - JacORB
 - MTDORB
 - IIOP.NET
 - InstantObjects (BDS2006 Pro版、Turbo Pro版が必要となります)
- CORBAクライアントをCGIやSOAPサーバとして4層以上にする構成もあります
- CORBAサーバを別実装に変えても、クライアントへの影響が無い
 - サーバ例1: Linux + JacORB + JDBC + PostgreSQL
 - サーバ例2: MacOS X(PPC) + TAO + MySQL++(API) + MySQL



Copyright © 2006, Borland Software Corporation. 本スライドの一切または一部の複製を禁じます。



BORLAND® DEVELOPER CAMP

Q&A

Borland®

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。