

ObjectARX のコンパイル方法

Microsoft Visual C++ 4.2 を使って AutoCAD R14 用の ARX を作る (スタティック DLL)

Microsoft Visual C++ 4.2 を使って AutoCAD R14 用の ObjectARX を作る方法を説明します。Visual C++ のバージョンは 5.0 でもコンパイルは可能なようですが、SDKに付属しているライブラリは 4.2 のものなので、特別な理由がなければ 4.2 を使いましょう。

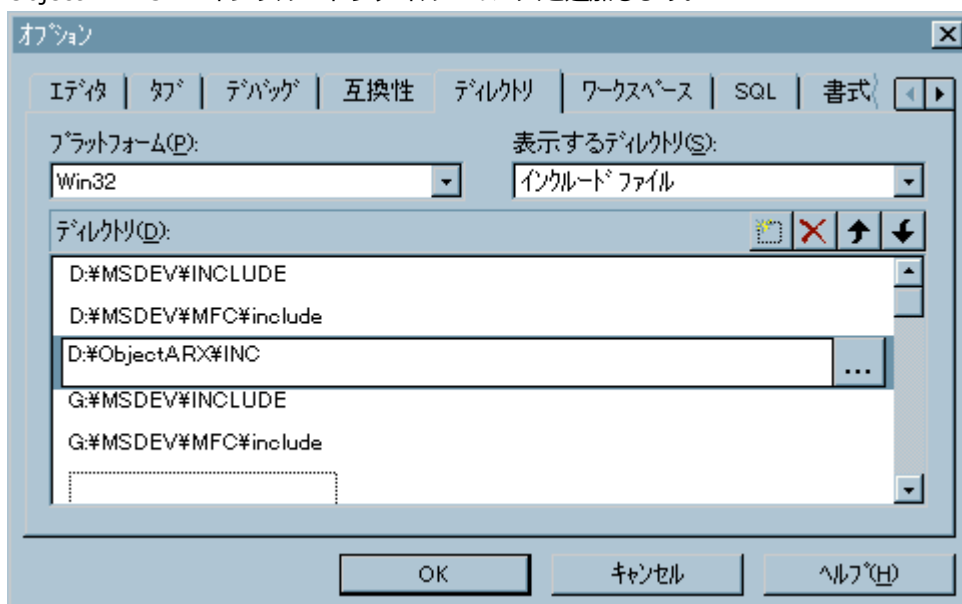
※スタティックライブラリを使った場合、ヘルプファイルに次のような警告文があります。

スタティックライブラリを使用した場合、一部の AutoCAD 14.0 MFC API が作動しないでしょう。詳しくは [ObjectARX Reference Manual online] を読んでください。

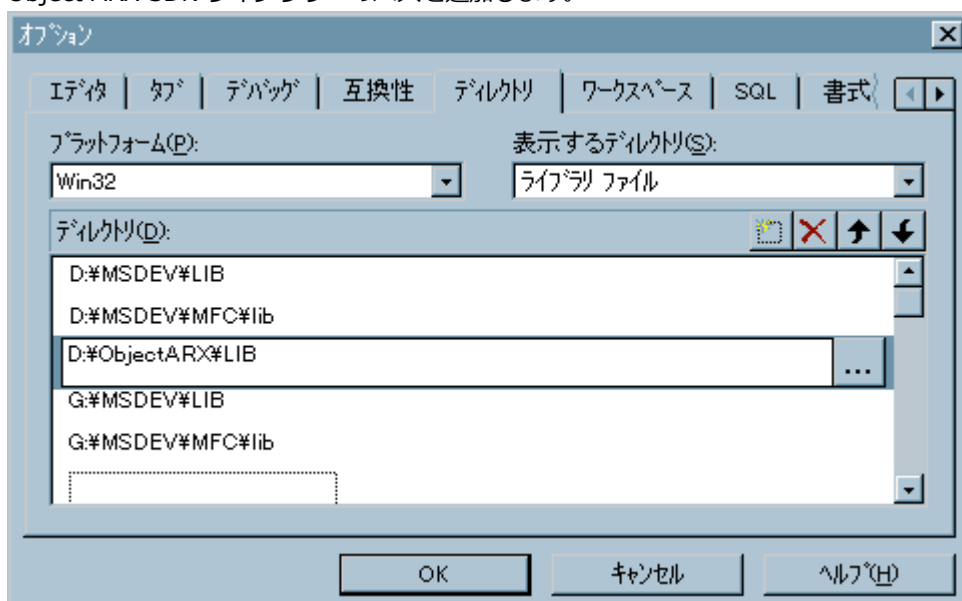
パスの設定

ライブラリとインクルードファイルへのパスを設定します。すでにこの作業が済んでいる場合は次のステップへ進んで下さい。

1. プルダウンメニューの [ツール] [オプション] から、オプションダイアログボックスを開きます。
2. Object ARX SDK インクルードファイルへのパスを追加します。



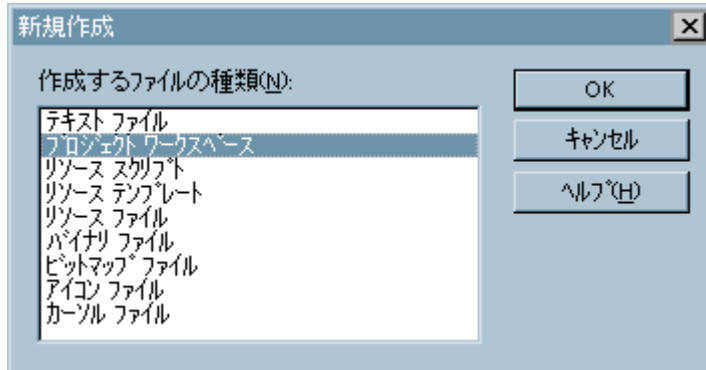
3. Object ARX SDK ライブラリへのパスを追加します。



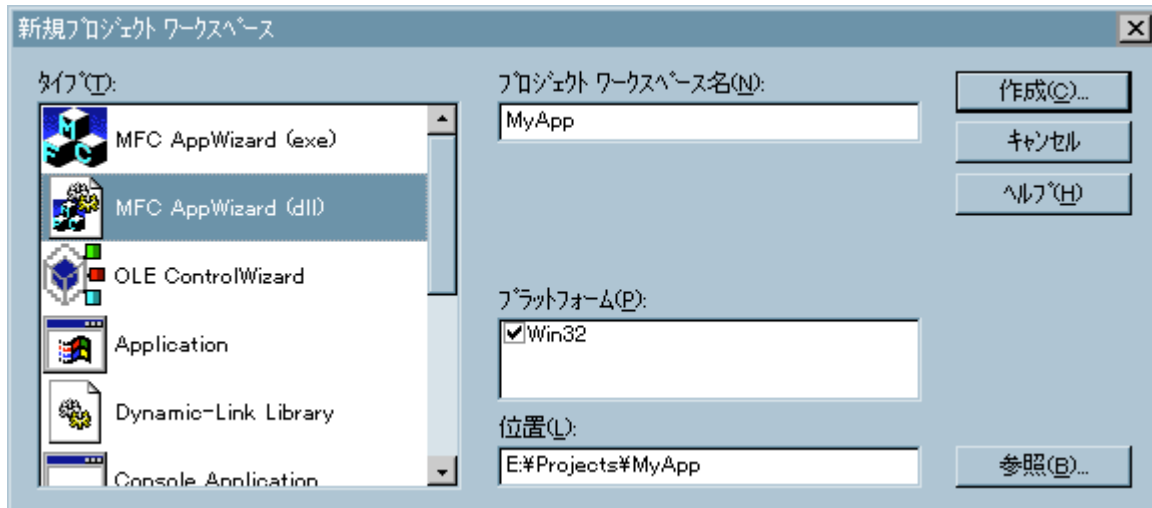
ワークスペースを作る

次はワークスペースを作ります。後でワークスペース名の変更をしようとする手間なので、きちんと名前を決めてから作りましょう。

1. [ファイル] メニューの [新規作成] をクリックします。
[新規作成] ダイアログから [プロジェクトワークスペース] を選択して、[OK] をボタンをクリックします。

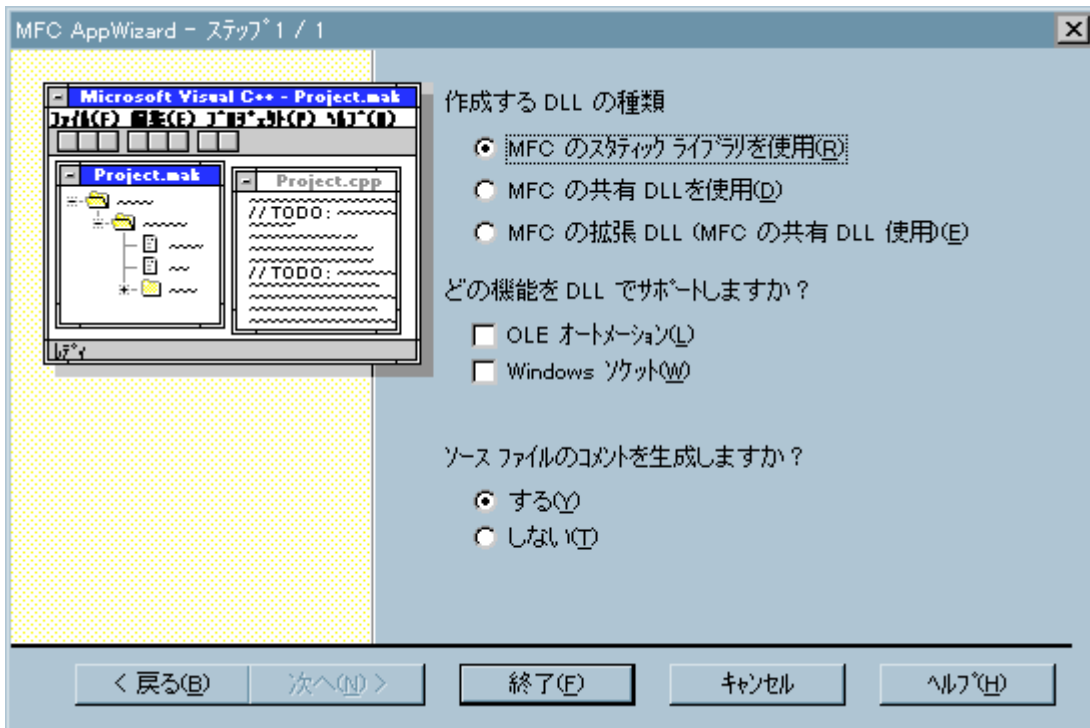


2. [新規プロジェクトワークスペース] ダイアログボックス
[タイプ] から [MFC AppWizard (dll)] を選択し、[位置] エディットボックスは必要に応じてフォルダを修正します。[プラットフォーム] の [Win32] チェックボックスを、チェックON の状態にして、[プロジェクトワークスペース名] にプロジェクト名を入力。最後に [作成] ボタンをクリックします。



3. [MFC AppWizard - ステップ 1/1] ダイアログボックス

[作成する DLL の種類] は [MFC のスタティックライブラリを使用] を選択します。[ソースファイルのコメントを生成しますか?] は、[する] がお勧めですね。最後に、[終了] ボタンをクリックします。



4. ワークスペースを閉じます

[ファイル] メニューの [ワークスペースを閉じる] を選択して、一旦ワークスペースを閉じます。

5. 不要なファイルの削除

エクスプローラ等を使って、今作ったワークスペースのフォルダから「プロジェクト名. CPP」というファイルと、「プロジェクト名. H」というファイルを削除します。プロジェクトに「MyApp」という名前をつけたのなら、「MyApp. CPP」と「MyApp. H」を削除します。

6. 必要なファイルのコピー

エクスプローラ等を使って、ObjectARX SDK をインストールしたドライブの、/ObjectARX/utils/mfcextras/から、「ARXMFCSTATIC.CPP」というファイルを、今作ったワークスペースのフォルダにコピーします。

7. 再度ワークスペースを開きます

[ファイル] メニューの [開く] を選択して、今作ったワークスペースを開きます。[プロジェクトワークスペース] ウィンドウが表示されていない場合は、[表示] メニューの [プロジェクトワークスペース] をクリックして、[プロジェクトワークスペース] ウィンドウを表示させます。

8. プロジェクトの構成ファイルを変更します

[プロジェクトワークスペース] ウィンドウの下部に表示されている [FileView] タグをクリックしてファイルの構成を変更します。「5. 不要なファイルの削除」で削除した「プロジェクト名. CPP」がプロジェクト内に残っているので、このファイルをプロジェクトから除外します。

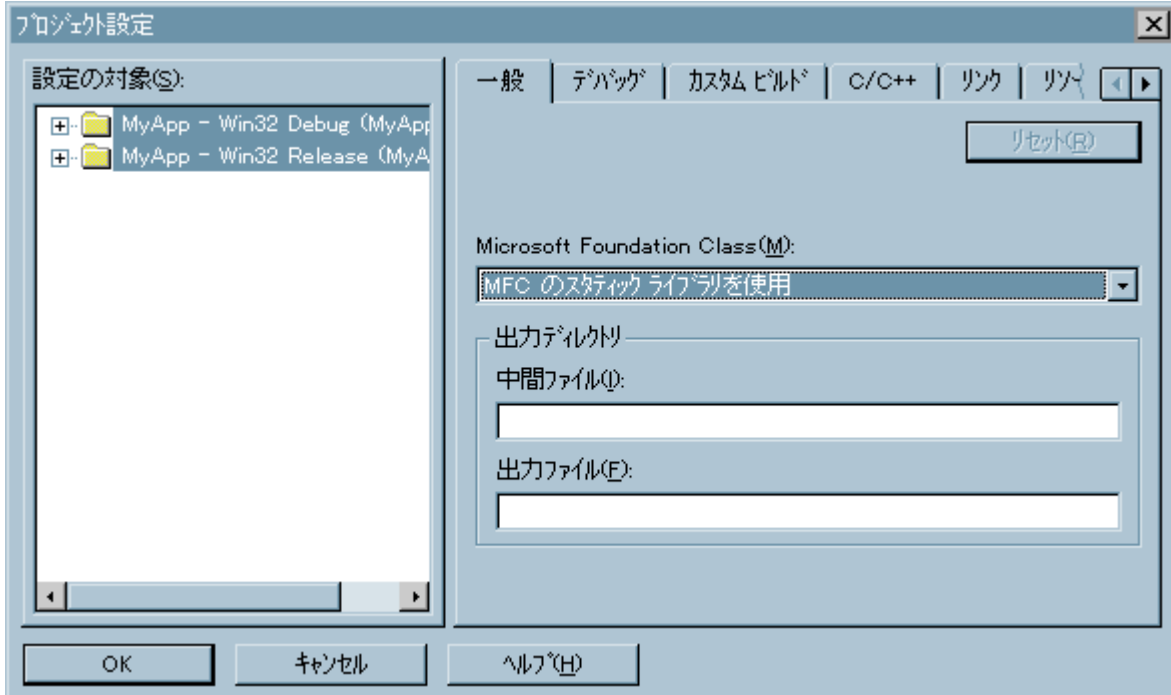
9. プロジェクトの構成ファイルを追加します

[挿入] メニューの [プロジェクトにファイルを追加] をクリックして、「6. 必要なファイルのコピー」で追加した「ARXMFCSTATIC.CPP」をプロジェクトに追加します。

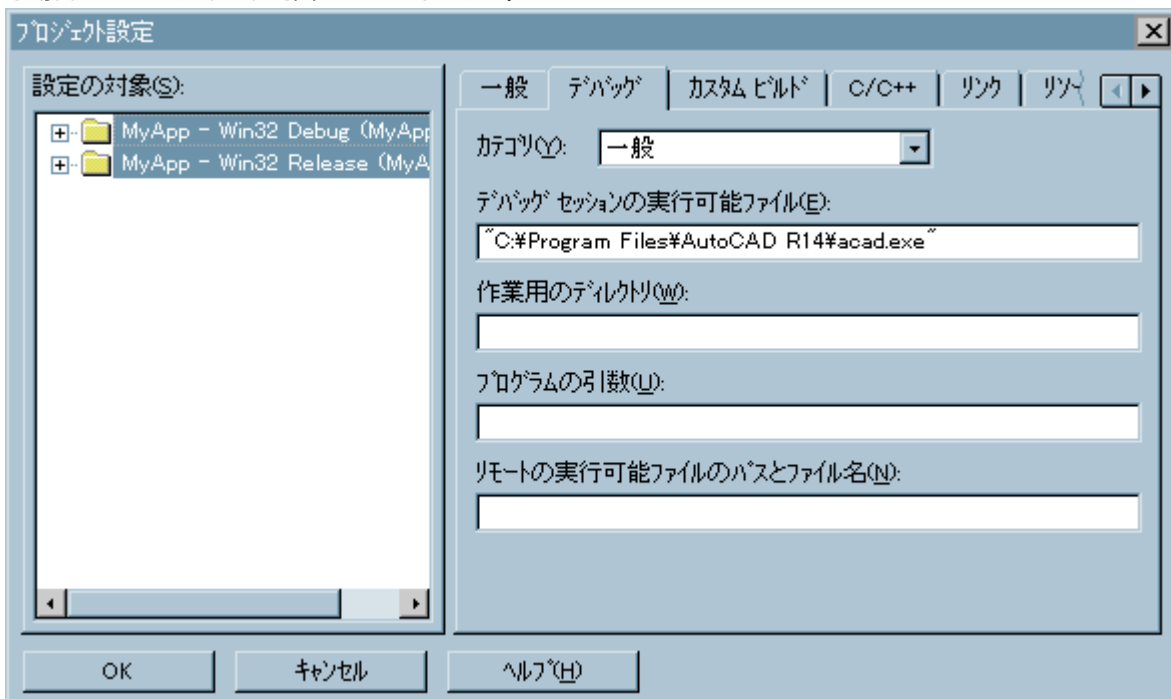
ワークスペースの設定

コンパイルのオプションなどを設定します。ここで説明していない設定はデフォルトでもコンパイル可能だと思いますが、不具合があった場合は修正して下さい。

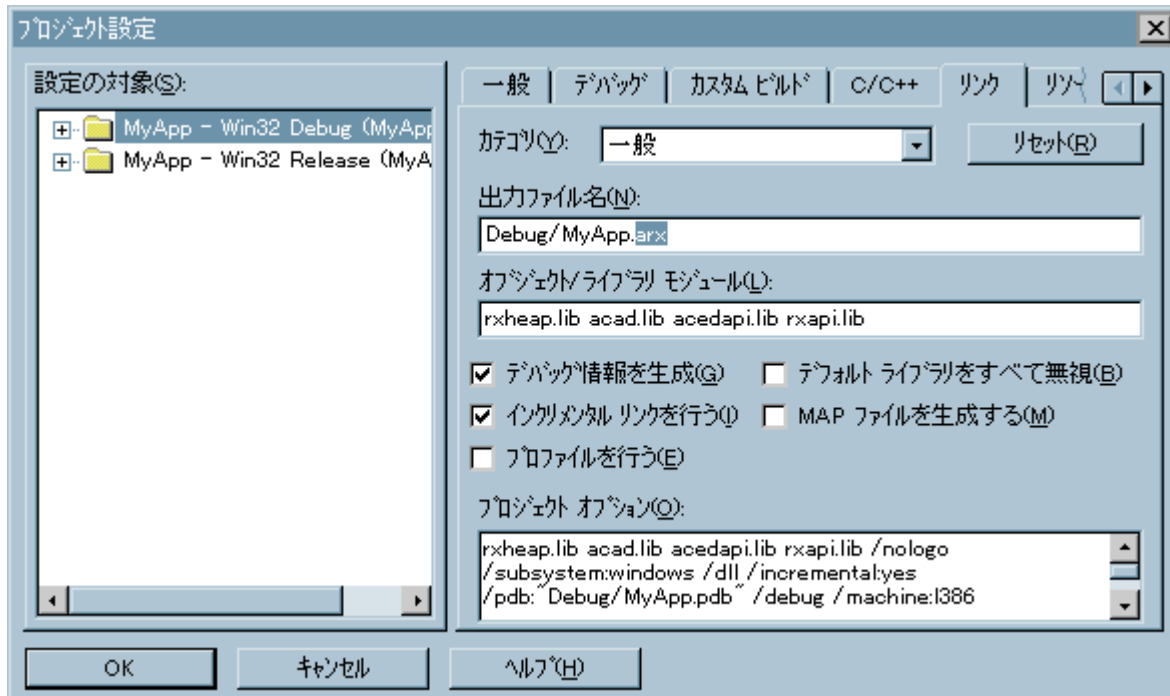
1. [ビルド] メニューの [設定] をクリックします
[プロジェクト設定] ダイアログが表示されるので、「WIN32 Debug」と「Win32 Release」の両方を選択（反転表示状態）にしておきます。
2. [プロジェクト設定] ダイアログボックスの [一般] タグ
[Microsoft Foundation Class] コンボボックスから [MFC のスタティックライブラリ を使用] を選択します。



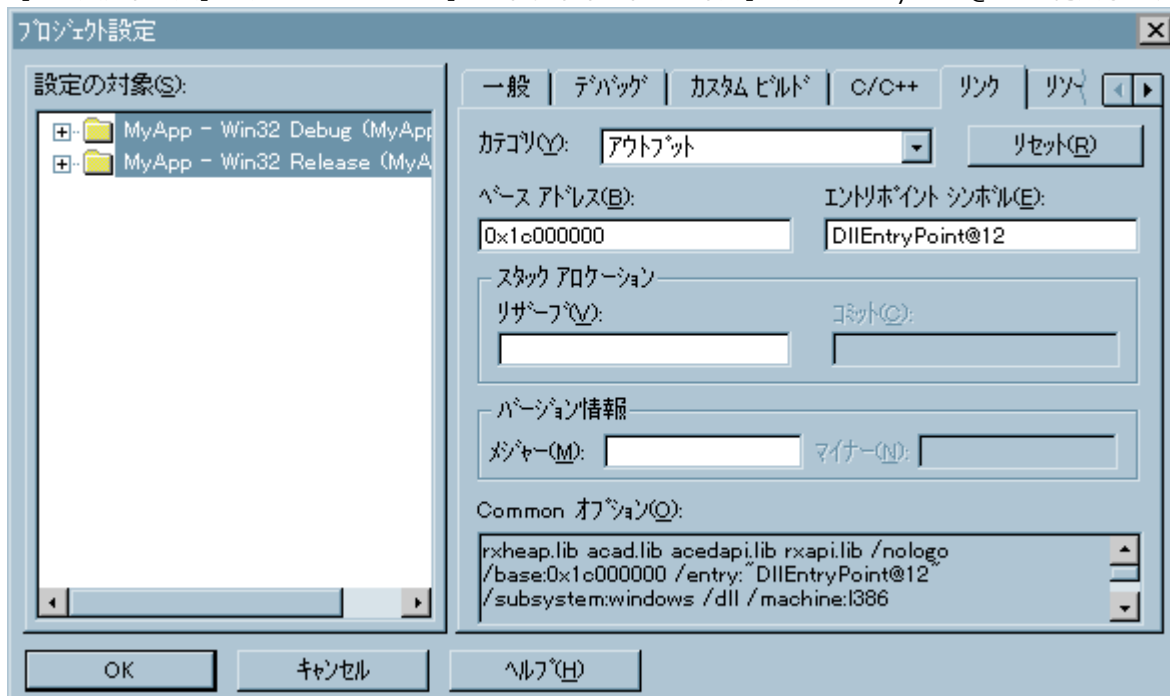
3. [プロジェクト設定] ダイアログボックスの [デバッグ] タグ、[カテゴリ] [一般]
[デバッグセッションの実行可能ファイル] に、ACAD.EXE をフルパスで記入します。（パスに空白文字が含まれている場合はダブルクォートで囲むようにします。）



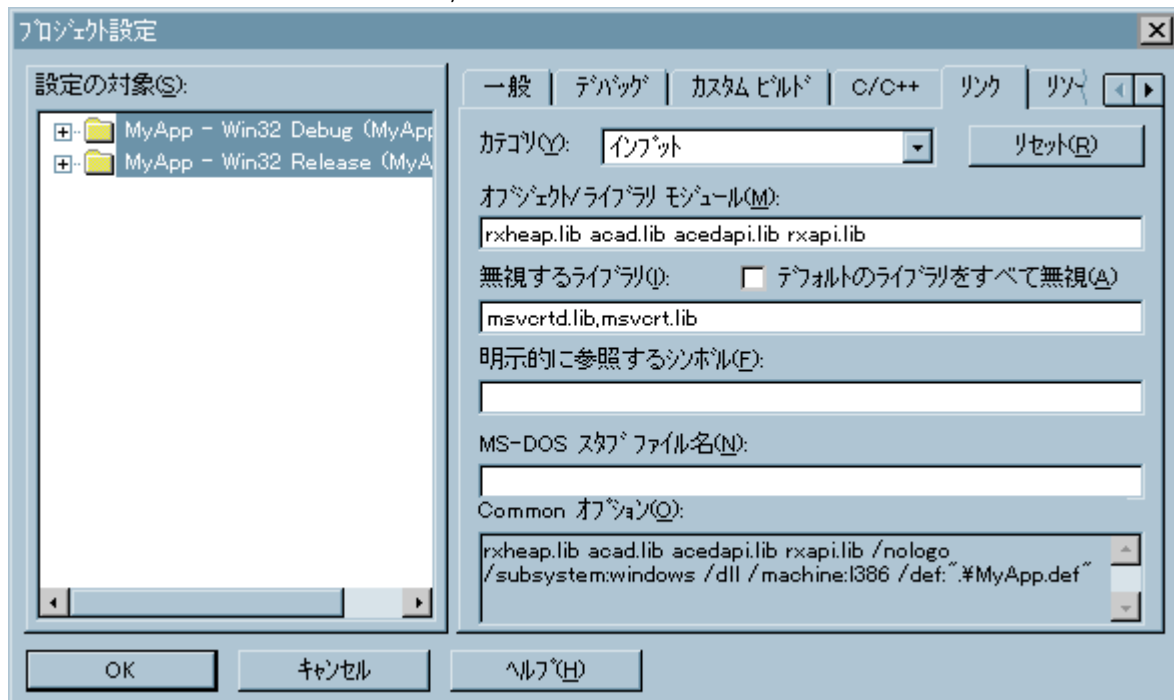
4. [プロジェクト設定] ダイアログボックスの [リンク] タブ、[カテゴリ] [一般]
 [オブジェクト/ライブラリモジュール] に rxheap.lib acad.lib acedapi.lib rxapi.lib など必要なライブラリを追加します。rxheap.lib は必須です。忘れずに追加して下さい。([出力ファイル名] の拡張子を「DLL」から「ARX」に変更しておく、後で変更する手間が省けます。)



5. [プロジェクト設定] ダイアログボックスの [リンク] タブ、[カテゴリ] [アウトプット]
 [ベースアドレス] に、0x1c000000 [エン트리ポイントシンボル] に、DllEntryPoint@12 と記入します。



6. [プロジェクト設定] ダイアログボックスの [リンク] タブ、[カテゴリ] [インプット] [無視するライブラリ] に、msvcrt.lib,msvcrt.lib を追加します。



7. ダイアログを閉じます
[OK] ボタンをクリックして、ダイアログボックスを閉じます。

ファイルの修正

ファイルを修正してスケルトンの作成とテストコマンドの追加を行います。

1. StdAfx.h を修正します

StdAfx.h ファイルの先頭と末尾に、次の様に追加します。

```
#if defined( _DEBUG) && !defined(DEBUG_AUTOCAD)
#pragma message("Building debug version ...")
#define _DEBUG_WAS_DEFINED
#undef _DEBUG
#endif
(ここに StdAfx.h の本文が入ります)
#ifdef _DEBUG_WAS_DEFINED
#define _DEBUG
#undef _DEBUG_WAS_DEFINED
#endif
```

2. ARX をアンロードできるように arxmfcDll.cpp を修正します

AcRx::AppRetCode acrxEntryPoint(AcRx::AppMsgCode msg, void*)関数を、次のように編集します。

```
AcRx::AppRetCode acrxEntryPoint( AcRx::AppMsgCode msg, void* pkt)
{
    switch( msg )
    {
        case AcRx::kInitAppMsg:
            initApp();
            break;
        case AcRx::kUnloadAppMsg:
            unloadApp();
            break;
        case AcRx::kInitAppMsg:
            acrxDynamicLinker->unlockApplication(pkt);
            initApp();
            break;
        default:
            break;
    }
    return AcRx::kRetOK;
}
```

3. プロジェクト名. def ファイルを修正します

EXPORTS の部分を修正します。

```
EXPORTS acrxEntryPoint
    _SetacrxPtp
```

4. [ビルド] メニューの [ビルド] をクリックします

これで、ARX のスケルトンが完成します。このままでは何もしない ARX なので、確認用のコマンドを ARX に追加してみましょう。(もしここでエラーが出るようなら、設定が間違っていないか、もう一度確認して下さい。)

5. arxmfc.dll に、コマンドを追加します。
追加するコマンドは、“こんにちは”を表示するだけの簡単なものです。

```
void Hello()
{
    ads_printf("こんにちは");
}
void initApp()
{
    InitMFC();
    acedRegCmds->addCommand("MY_APP", "HELLO", "HELLO", ACRX_CMD_MODAL, Hello);
}
void unloadApp()
{
    Acad::ErrorStatus es;
    if ((es = acedRegCmds->removeGroup("MY_APP")) != Acad::eOk)
        ads_printf("¥nFailed: Remove error. Status %d¥n", es);
    EndMFC();
}
```

6. コンパイル
修正が完了したら、もう一度コンパイルします。これで HELLO というコマンドが入った ARX が出来上がりました。Microsoft Visual C++ 4.2 を終了して、AutoCAD を立ち上げましょう。
7. ARX をロードします
AutoCAD のコマンドラインから、ARX と入力します。ファイル名入力用ダイアログボックスが表示されるので、今作った ARX を指示して下さい。
8. コマンドを実行します
AutoCAD のコマンドラインから、HELLO と入力すると、コマンドラインに、“こんにちは”と表示されます。

デバッグする場合

1. Microsoft Visual C++ 4.2 でソースファイルを開き、プログラム途中で中断したい部分にカーソルを置き、[F9]キーを押して、ブレークポイントをセットします。
2. [ビルド]メニューの[実行]で ACAD.EXE を実行します。もし、AutoCAD が立ち上がらない場合は、Step3 で行った [プロジェクト設定] ダイアログボックスの [デバッグ] タブ、[カテゴリ] [一般] を確認して下さい。
3. AutoCAD コマンドラインから ARX をロードして、作成したコマンドを発行すれば、ブレークポイント位置でプログラムが停止します。