

SprCSharp の新しい構成の作り方

SprCSharp の新しい構成の作成方法について、「Visual Studio 2015 を使用した “Physics を含む最小構成版の SprCSharp” の作成」を例として記述する。

最初に、他の構成を作成する場合または Visual Studio の別の版を使用する場合に、以下の文書をどのように読み替えたら良いかについて述べる。次に、新しいソリューションを作成するにあつての準備作業について述べ、その後、新しいソリューションの枠組みの作り方、新しいソリューションの設定の方法の順に述べる。

目次

1	読み替えルール	2
2	準備作業	4
2.1	現在すでにある構成	4
2.2	準備しておくこと	4
3	ソリューション枠組みの作成	6
3.1	ソリューション ‘SprCSharp<14.0><PH>’ の作成	6
3.2	プロジェクト ‘SprImport<14.0><PH>’ の追加	7
3.3	プロジェクト ‘SprExport<14.0><PH>’ の追加	8
3.4	プロジェクト ‘NameManager’ の追加	9
3.5	プロジェクト ‘TargetManager<PH>’ の追加	10
3.6	‘RunSwig.CSharp<14.0><PH>’ の追加	11
3.7	プロジェクト ‘Install’ の追加	12
3.8	Springhead Library プロジェクトの追加	12
3.9	プロジェクト ‘SprCsSample<14.0><PH>’ の追加	12
4	設定	13
4.1	ソリューション ‘SprCSharp’ のプロパティの設定	13
4.2	参照の設定	15
4.3	各プロジェクトの設定	15

1 読み替えルール

この文書では、“Visual Studio 2015”を示すために“<14.0>”という記法を、“Physicsを含む最小構成版”を示すために“<Physics>”, “<PH>”などという記法を用いる。他の構成を作成する場合または Visual Studio の別の版を使用する場合には、これら“<...>”の部分を適切に読み替える必要がある。次表に、その読み替え方法を示す。

この文書での記法	読み換え方法																				
<Physics>	最小構成作成の対象となる (Springhead の) プロジェクト名に読み替える。ただし、 <ul style="list-style-type: none">全体構成を作成する場合は“(空文字列)”とする。複数のプロジェクトを対象とする場合はプロジェクト名をカンマで区切って並べる。 (例) Physics を含む最小構成 → Physics Physics と Graphics → Physics,Graphics 全体構成 → (空文字列)																				
<PH>	構成に与える名称 (基本的にプロジェクトの略号) に読み替える。ただし、全体構成を作成する場合は‘ALL’と読み替える。 (例) Physics を含む最小構成 → PH Physics と Graphics → PHGR 全体構成 → ALL																				
<14.0>	使用する Visual Studio のバージョンに読み替える。Visual Studio のマクロ \$(VisualStudioVersion) の値を使用する。 (例) VS 2013 → 12.0 VS 2015 → 14.0																				
<EDP>	ソースレベルで陽に依存するプロジェクトに読み替える。次の表のとおり。 <table><tr><th>プロジェクト</th><th>依存するプロジェクト</th></tr><tr><td>Base</td><td>なし</td></tr><tr><td>Collision</td><td>Foundation</td></tr><tr><td>Creature</td><td>Foundation, Framework, Physics</td></tr><tr><td>FileIO</td><td>Foundation</td></tr><tr><td>Foundation</td><td>Base</td></tr><tr><td>Framework</td><td>FileIO, Foundation, Graphics, Physics</td></tr><tr><td>Graphics</td><td>Foundation</td></tr><tr><td>HumanInterface</td><td>Foundation</td></tr><tr><td>Physics</td><td>Collision, Foundation</td></tr></table>	プロジェクト	依存するプロジェクト	Base	なし	Collision	Foundation	Creature	Foundation, Framework, Physics	FileIO	Foundation	Foundation	Base	Framework	FileIO, Foundation, Graphics, Physics	Graphics	Foundation	HumanInterface	Foundation	Physics	Collision, Foundation
プロジェクト	依存するプロジェクト																				
Base	なし																				
Collision	Foundation																				
Creature	Foundation, Framework, Physics																				
FileIO	Foundation																				
Foundation	Base																				
Framework	FileIO, Foundation, Graphics, Physics																				
Graphics	Foundation																				
HumanInterface	Foundation																				
Physics	Collision, Foundation																				

† 表は再帰的に見る。例えば、

Physics → Collision, Foundation

Collision → Foundation

Foundation → Base

なので、

⟨Physics⟩ → Base, Collision, Foundation

となる。

⟨IDP⟩

ライブラリレベルで陰に依存するプロジェクトに読み替える。

隠れ依存 — すなわち、⟨EDP⟩ だけではリンク時に未定義外部シンボルが出る場合。

(例) ⟨Physics⟩ → HumanInterface

2 準備作業

2.1 現在すでにある構成

現在既に作成済みの構成は、次のとおりである。

Solution 名	構成	Visual Studio
SprCSharp12.0	全体構成	2013
SprCSharp12.0PH	Physics 最小構成	2013
SprCSharp14.0	全体構成	2015
SprCSharp14.0PH	Physics 最小構成	2015

2.2 準備しておくこと

1. 構成の対象とするプロジェクト名を基にして、〈Physics〉, 〈PH〉, 〈EDP〉, 〈IDP〉 に相当するものを決める。ソリューションファイル名およびプロジェクトファイル名は 〈PH〉 を基に決めるので、これらが既存のものと重複しないようにすることが重要である。

(例) Physics 最小構成に Graphics を追加した構成の場合。

〈Physics〉	→	〈Graphics,Physics〉
〈PH〉	→	〈GRPH〉
〈EDP〉	→	Base, Collision, Foundation, Graphics
〈IDP〉	→	HumanInterface, ... (要調査)

2. ".i" ファイルの編集 (必要ならば)

"swig_sprcs.exe" が実行されるとき、次のようなマクロが引数で渡される。

〈Graphics〉	→	-DGraphics_ONLY
〈Graphics,Physics〉	→	-DGraphics_and_Physics
全体構成	→	何も定義されない

このマクロを用いて、必要ならば ".i" ファイルを編集する (この構成には不要なメソッドで、〈EDO〉 や 〈IDP〉 を拡張するようなものは %ignore しておく)。

(例) "Physics.i"

```
#ifdef Physics_ONLY
%ignore Spr::PHOpEngineIf::InitialHapticRenderer;
...
#endif
```

(参考) Project RunSwig_CSharp をビルドすると、

```
nmake -f RunSwig_CSharp/Makefile.RunSwig_CSharp
RunSwig_CSharp/RunSwig_CSharp.bat TARGET=<Physics>
nmake Makefile_CSharp.swig.<Physics>
RunSwig_CSharp/CSharpSwig.bat <Physics> <Macro>
swig_sprcs.exe -D<Macro>
```

と順次呼び出され、"swig_sprcs.exe" に上記のマクロが渡される。

3. プロジェクト 'TargetManager' を追加するためのメモ

新しい構成を作るときは、プロジェクト 'TargetManager' は既存のものを利用するのではなく、新規に作成しなければならない。ポイントは次のとおり。

- Makefile Project を作成する。
- Build Command Line には "TargetManager.bat <Physics>" を指定する。
- Rebuild All Command Line には "TargetManager.bat <Physics>" を指定する。

(例) 全体構成 → TargetManager.bat ALL
 <Physics,Graphics> → TargetManager.bat Physics,Graphics

実際の作成方法は次章 (3 ソリューション枠組みの作成) で述べる。

3 ソリューション枠組みの作成

ソリューション枠組みは次の手順で作成する。〈ID〉については下記の注を参照のこと。

1. ソリューション 'SprCSharp<ID>' およびプロジェクト 'SprCSharp<ID>' (C#の Class Library) を新規に作成する。
2. プロジェクト 'SprImport<ID>' (C#の Class Library) を新規に作成して追加する。
3. プロジェクト 'SprExport<ID>' (C++ の Class Library) を新規に作成して追加する。
4. プロジェクト 'NameManager' (既存のプロジェクト) を追加する。
5. プロジェクト 'TargetManager<ID>' (既存のプロジェクト) を追加するか、またはプロジェクト 'TargetManager<ID>' (C++ の Makefile Project) を新規に作成して追加する。
6. プロジェクト 'RunSwig-CSharp<ID>' (C++ の Makefile Project) を新規に作成して追加する。
7. プロジェクト 'Install' (既存のプロジェクト) を追加する。
8. Springhead Library のプロジェクト (既存のプロジェクト) を追加する。
9. プロジェクト 'SprCsSample<ID>' (C#の Console Application) を新規に作成して追加する。

新規にプロジェクトを作成する場合には、既存のディレクトリ内に新規ファイルが作成されないように注意すること (既存のファイルが上書きされる恐れがある)。そのためには、新規プロジェクト作成時に指定する Name が、既存のディレクトリ名と一致しないようにすることが必要となる (そのために 〈ID〉 を付加している)。

〈ID〉=〈PH〉 5. TargetManager のとき
〈ID〉=〈14.0〉〈PH〉 それ以外のとき

これに伴い、ファイルのコピーおよびソリューションファイルの編集作業が必要となる。作業内容については、これ以降で詳しく述べる。

以降、ソリューションを作成するディレクトリを "DIR" とし、例示するファイル操作等は、このディレクトリで行なうものとして記述する。

3.1 ソリューション 'SprCSharp<14.0>〈PH〉' の作成

ソリューション 'SprCSharp<14.0>〈PH〉.sln' とプロジェクト 'SprCSharp<14.0>〈PH〉.csproj' とを同時に作る。

1. Visual Studio を起動
[New Project] - [Visual C#] - [Class Library]
Name: SprCSharp<14.0><PH>
Location: DIR

Visual Studio を終了 (変更を保存)

2. ディレクトリ "SprCSharp<14.0><PH>" (= <NEWDIR> とする) が作成されるのでソリューションファイルおよびプロジェクトファイルをコピー

```
DIR> cp <NEWDIR>\SprCSharp<14.0><PH>.sln .
DIR> cp <NEWDIR>\SprCSharp<14.0><PH>.csproj SprCSharp
† コピー先のディレクトリ名に注意
```

3. ソリューションファイル "DIR\SprCSharp<14.0><PH>.sln" を編集

```
Project(...) = "... "SprCSharp<14.0><PH>.csproj",...
↓
Project(...) = "... "SprCSharp\SprCSharp<14.0><PH>.csproj",...
† ディレクトリ名に注意
```

4. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer でプロジェクト 'SprCSharp<14.0><PH>' を展開

Class1.cs を [Delete]

[Add]–[Existing Item...] で次のものを追加

```
"cs_object.cs"
"module.wrapper.cs"
"module.wrapflushrightper.cs.prologue"
"wrapper.cs"
"CSUtility.cs"
```

次のものは、存在すれば [Add]–[Existing Item...] で追加、存在しなければ [Add]–[New Item...]–[Visual C# Items]–[Class] で追加 (この場合ソースファイルはダミー)

```
"CSBase.cs"
"CSCollision.cs"
"CSFoundation.cs"
"CSPhysics.cs"
```

† ここで追加するのは <EDP> に対応したもののみ

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

5. ディレクトリ "SprCSharp<14.0><PH>" を削除

3.2 プロジェクト 'SprImport<14.0><PH>' の追加

新規に作成して追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック

[Add]–[New Project...]–[Visual C#]–[Class Library]

Name: SprImport<14.0><PH>

Location: DIR

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

2. ディレクトリ "SprImport<14.0><PH>" (= <NEWDIR> とする) が作成されるのでプロジェクトファイルをコピー

```
DIR> cp <NEWDIR>\SprImport<14.0><PH>.csproj SprImport
```

† コピー先のディレクトリ名に注意

3. ソリューションファイル "DIR\SprCSharp<14.0><PH>.sln" を編集

```
Project(...) = "... " <NEWDIR>\SprImport<14.0><PH>.cspproj",...
```

↓

```
Project(...) = "... "SprImport\SprImport<14.0><PH>.cspproj",...
```

† ディレクトリ名に注意

4. ソリューション "SprCSharp<14.0><PH>.sln" を起動する

Solution Explorer でプロジェクト 'SprImport<14.0><PH>' を展開

Class1.cs を [Delete]

[Add]–[Existing Item...] で次のものを追加

```
"module.wrapper.cs"
```

```
"module.wrapper.cs.prologue"
```

```
"wrapper.cs"
```

```
"CSUtility.cs"
```

次のものは、存在すれば [追加]–[既存の項目] で追加、存在しなければ [Add]–[New Item...]–[Visual C# Items]–[Class] で追加 (この場合ソースファイルはダミー)

```
"CSBase.cs"
```

```
"CSCollision.cs"
```

```
"CSFoundation.cs"
```

```
"CSPhysics.cs"
```

† ここで追加するのは <EDP> に対応したもののみ

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

5. ディレクトリ "SprImport<14.0><PH>" を削除

3.3 プロジェクト 'SprExport<14.0><PH>' の追加

新規に作成して追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック

[Add]–[New Project...]–[Visual C++]–[CLR]–[Class Library]

Name: SprExport<14.0><PH>

Location: DIR

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

2. ディレクトリ "SprExport<14.0><PH>" (= <NEWDIR> とする) が作成されるのでプロジェクトファイルをコピー

```
DIR> cp <NEWDIR>\SprExport<14.0><PH>.vcxproj SprExport
```

```
DIR> cp <NEWDIR>\SprExport<14.0><PH>.vcxproj.filters SprExport
```

† コピー先のディレクトリ名に注意

3. ソリューションファイル "DIR\SprCSharp<14.0><PH>.sln" を編集


```
Project(...) = "... " <NEWDIR>\SprExport<14.0><PH>.vcxproj",...
```

↓

```
Project(...) = "... "SprExport\SprExport<14.0><PH>.vcxproj",...
```

† ディレクトリ名に注意

4. ソリューション "SprCSharp<14.0><PH>.sln" を起動する

Solution Explorer でプロジェクト 'SprExort<14.0><PH>' を展開

[Header Files]

フォルダ内のファイルをすべて [Remove]–[Remove]

[Add]–[Existing Item...] で次のものを追加

"CSUtility.h"

[Resource Files]

フォルダ内のファイルをすべて [Remove]–[Remove]

フォルダ内のファイルをすべて [Delete]

[Source Files]

フォルダ内のファイルをすべて [Remove]–[Remove]

[Add]–[Existing Item...] で次のものを追加

"module.wrapper.cpp"

"module.wrapper.cpp.prologue"

"wrapper.cpp"

"CSUtility.cpp"

次のものは、存在すれば [Add]–[Existing Item...] で追加、存在しなければ [Add]–[New Item...]–[Visual C++]–[C++ File(.cpp)] で追加 (この場合ソースファイルはダミー)

"CSBase.cpp"

"CSCollision.cpp"

"CSFoundation.cpp"

"CSPhysics.cpp"

† ここで追加するのは (EDP) に対応したもののみ

ReadMe.txt を [Remove]–[Remove]

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

5. ディレクトリ "SprExport<14.0><PH>" を削除

3.4 プロジェクト 'NameManager' の追加

既存のプロジェクトを追加する。

1. ソリューション 'SprCSharp<14.0><PH>.sln' を起動

Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック

[Add]–[Existing Project...]

DIR\NameManager\NameManager.vcxproj

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

3.5 プロジェクト 'TargetManager<PH>' の追加

プロジェクトファイル "DIR\TargetManager\TargetManager<PH>.vcxproj" が存在するか否かで処理が異なる。

- 存在する場合 — 既存のプロジェクトを追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック

[Add]–[Existing Project...]

DIR\TargetManager\TargetManager<PH>.vcxproj

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

- 存在しない場合 — 新規に作成して追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック

[Add]–[New Project...]–[Visual C++]–[General]–[Makefile Project]

Name: TargetManager<PH>

Location: DIR

[Finish]

[TargetManager<PH>]–[Properties]

[General]–[Project Defaults]–[Configuration Type] を Makefile にする

[Apply]–[OK]

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

2. ディレクトリ "TargetManager<PH>" (=(<NEWDIR> とする) が作成されるのでプロジェクトファイルをコピー

DIR> cp <NEWDIR>\TargetManager<PH>.vcxproj TargetManager

DIR> cp <NEWDIR>\TargetManager<PH>.vcxproj.filters TargetManager

† コピー先のディレクトリ名に注意

3. ソリューションファイル "DIR\SprCSharp14.0.sln" を編集

Project(...) = "... " <NEWDIR>\TargetManager<PH>.vcxproj", ...

↓

Project(...) = "... "TargetManager\TargetManager<PH>.vcxproj", ...

† ディレクトリ名に注意

4. ソリューション "SprCSharp<14.0><PH>.sln" を起動する

Solution Explorer で、Project TargetManager<PH> を展開

[Header Files] を [Delete]

[Resource Files] を [Delete]

[Source Files] を [Delete]

Readme.txt を [Remove]–[Remove]

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

3.6 'RunSwig_CSharp<14.0><PH>' の追加

新規に作成して追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動

Solution Explorer で、ソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック
[Add]–[New Project...]–[Visual C++]–[General]–[Makefile Project]

Name: RunSwig_CSharp<14.0><PH>

Location: DIR

[Finish]

[RunSwig_CSharp14.0]–[Properties]

[General]–[Project Defaults]–[Configuration Type] を Makefile にする

[Apply]–[OK]

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

2. ディレクトリ "RunSwig_CSharp<14.0><PH>" (=NEWDIR とする) が作成されるのでプロジェクトファイルをコピー

DIR> cp <NEWDIR>\RunSwig_CSharp<14.0><PH>.vcxproj RunSwig_CSharp

DIR> cp <NEWDIR>\RunSwig_CSharp<14.0><PH>.vcxproj.filters RunSwig_CSharp

† コピー先のディレクトリ名に注意

3. ソリューションファイル "DIR\SprCSharp14.0.sln" を編集

Project(...) = "... " <NEWDIR>\RunSwig_CSharp<14.0><PH>.vcxproj",...

↓

Project(...) = "... "RunSwig_CSharp\RunSwig_CSharp<14.0><PH>.vcxproj",...

† ディレクトリ名に注意

4. ソリューション "SprCSharp<14.0><PH>.sln" を起動する

Solution Explorer でプロジェクト 'RunSwig_CSharp<14.0><PH>' を展開

[Header Files] を [Delete]

[Resource Files] を [Delete]

[Source Files] を [Delete]

Readme.txt を [Remove]–[Remove]

[Add]–[Existing Item...] で次のものを追加

"CSharpSwig.bat"

"Makefile.RunSwig_CSharp"

"ReplaceInclude.py"

"RunSwig_Clean.bat"

"RunSwig_CSharp.bat"

"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

5. ディレクトリ "RunSwig_CSharp<14.0><PH>" を削除

3.7 プロジェクト 'Install' の追加

既存のプロジェクトを追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動
Solution Explorer でソリューション 'SprCSharp14.0 (xx projects)' を右クリック
[Add]–[Existing Project...]
DIR\Install\Install.vcxproj
"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

3.8 Springhead Library プロジェクトの追加

既存のプロジェクトを追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動
<EDP> および <IDP> に含まれるすべてのプロジェクトについて
Solution Explorer でソリューション 'SprCSharp<14.0><PH> (xx projects)' を右クリック
[Add]–[Existing Project...]
<...>\Springhead2\src\<Project>\<Project><14.0>.vcxproj
2. "SprCSharp<14.0><PH>.sln" を終了 (変更を保存)

3.9 プロジェクト 'SprCsSample<14.0><PH>' の追加

新規に作成して追加する。

1. ソリューション "SprCSharp<14.0><PH>.sln" を起動
Solution Explorer でソリューション 'SprCSharp14.0 (xx projects)' を右クリック
[Add]–[New Project...]–[Visual C#]–[Console Application]
Name: SprCsSample<14.0><PH>
Location: DIR
"SprCSharp<14.0><PH>.sln" を終了 (変更を保存)
2. ディレクトリ "SprCsSample<14.0><PH>" (= <NEWDIR> とする) が作成されるのでプロジェクトファイルをコピー
DIR> cp <NEWDIR>\SprCsSample<14.0><PH>.csproj SprCsSample
† コピー先のディレクトリ名に注意
3. ソリューションファイル "DIR\SprCSharp14.0.sln" を編集
Project(...) = "... " <NEWDIR>\SprCsSample<14.0><PH>.csproj", ...
↓
Project(...) = "... "SprCsSample\SprCsSample<14.0><PH>.csproj", ...
† ディレクトリ名に注意
4. ディレクトリ "SprCsSample<14.0><PH>" を削除

4 設定

この章では、混乱が起きないと思われる箇所については〈14.0〉〈PH〉の記述を省略する。適宜補って解釈すること。

4.1 ソリューション ‘SprCSharp’ のプロパティの設定

Solution Explorer でソリューション ‘SprCSharp (xx projects)’ を選択
Menu: [Project]–[Properties] でプロパティページを開く

[Common Properties]

[Startup Project]–[Single startup project]

SprCsSample

[Project Dependencies]

次の表のうち、Springhead Library に属するプロジェクトについては、〈EDP〉または〈IDP〉に存在するものだけを対象とする。

プロジェクト	依存関係にあるプロジェクト
Base	RunSwig
Collision	Foundation, RunSwig
Creature	Foundation, Framework, Physics, RunSwig
FileIO	Foundation, RunSwig
Foundation	Base, RunSwig
Framework	FileIO, Foundation, Graphics, Physics, RunSwig
Graphics	Foundation, RunSwig
HumanInterface	Foundation, RunSwig
Install	SprCSharp, SprExport, SprImport
NameManager	なし
Physics	Collision, Fouondation, RunSwig
RunSwig	なし
RunSwig_CSharp	Base, Collision, Creature, FileIO, Foundation, Framework, Graphics, HumanInterface, NameManager, Physics, TargetManager
SprCSharp	RunSwig_CSharp, SprImport
SprCsSample	SprCSharp, SprExport, SprImport
SprExport	Base, Collision, Creature, FioeIO, Foundation, Framework, Graphics, HumanINterface, Physics, RunSwig_CSharp
SprImport	RunSwig_CSharp
TargetManager	なし

[Configuration Properties]

[Configuration]–[Configuration Manager...]

SprCSharp, SprCsSample, SprImport のすべてについて

1. Platform に ‘x86’ と ‘x64’ を追加する

C#プロジェクトには ‘Win32’ がないので ‘x86’ で代用する

‘Any CPU’ のドロップダウンリストで 〈New...〉 から選ぶ

Create new project platforms のチェックははずしておく

2. Platform から ‘Win32’, ‘x64’ 以外のものを削除する

3. Project contexts を次のように設定する

Project†	Config	Platform	Build
[Debug]–[Win32]			
LibProj	Debug	Win32	check
C#Proj	Debug	x86	check
Others	Debug	Win32	check
[Release]–[Win32]			
LibProj	Release	Win32	check
C#Proj	Release	x86	check
Others	Release	Win32	check
[Trace]–[Win32]			
LibProj	Trace	Win32	check
C#Proj	Release	x86	check
Others	Release	Win32	check
[Debug]–[x64]			
LibProj	Debug	x64	check
C#Proj	Debug	x64	check
Others	Debug	x64	check
[Release]–[x64]			
LibProj	Release	x64	check
C#Proj	Release	x64	check
Others	Release	x64	check
[Trace]–[x64]			
LibProj	Trace	x64	check
C#Proj	Release	x64	check
Others	Release	x64	check

† ただし

LibProj	=	Base, Collision, Creature, FileIO, Foundation, Framework, Graphics, HumanInterface, Physics, RunSwig
C#Projs	=	CpsCSharp, SprCsSample, SprImport
Others	=	Install, NameManager, RunSwig-CSharp, SprExport, TargetManager

とする。すなわち、

Configuration	Springhead Library 以外では ‘Trace’ がないので ‘Release’ で代用する。
Platforms	C#プロジェクトでは ‘Win32’ がないので ‘x86’ で代用する。

4.2 参照の設定

以下の表の各 *Project* について、

Solution Explorer で *Project* を選択

Menu: [Project]–[Add Reference...]

Name 欄に示されたプロジェクトの中で次のものをチェックして [OK]

<i>Project</i>	Name
Collision	Foundation
Creature	Foundation, Framework, Physics
FileIO	Foundation
Foundation	Base
Framework	FileIO, Foundation, Graphics, Physics
Graphics	Foundation
HumanInterface	Foundation
Physics	Collision, Foundation
SprCSharp	SprImport
SprCsSample	SprCSharp, SprImport
SprExport	Base, Collision, Creature, FileIO, Foundation, Framework, Graphics, HumanInterface, Physics

4.3 各プロジェクトの設定

4.3.1 NameManager

Solution Explorer で [NameManager]–[Properties] を選択

‘All Configurations’ – ‘All Platforms’ について

[Configuration Properties]–[General]

```

[Output Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
[Intermediate Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
[Configuration Type]
    Makefile
[Build Log File]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\ (次の行に続く)
    $(MSBuildProjectName).log
[Configuration Properties]--[NMake]
[Build Command Line]
    nmake -f Makefile.NameManager
[Rebuild All Command Line]
    nmake -f Makefile.NameManager
[Clean Command Line]
    nmake -f Makefile.NameManager clean
[Output]
    (空白)

```

4.3.2 RunSwig_CSharp

Solution Explorer で [RunSwig_CSharp]--[Properties] を選択

‘All Configurations’ – ‘All Platforms’ について

```

[Configuration Properties]--[General]
[Output Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
[Intermediate Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
[Configuration Type]
    Makefile
[Build Log File]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\ (次の行に続く)
    $(MSBuildProjectName).log
[Configuration Properties]--[NMake]
[Build Command Line]
    nmake -f Makefile.RunSwig_CSharp (次の行に続く)
    TARGET=<Physics> (次の行に続く)
    PLAT=$(PlatformShortName) PROJECTDIR=$(ProjectDir)
[Rebuild All Command Line]
    RunSwig_Clean && nmake -f Makefile.RunSwig_CSharp (次の行に続く)
    TARGET=<Physics> (次の行に続く)
    PLAT=$(PlatformShortName) PROJECTDIR=$(ProjectDir)
[Clean Command Line]

```


RunSwig_Clean
[Output]
(空白)

4.3.3 SprCSharp

Solution Explorer で [SprCSharp]–[Properties] を選択

[Application]–[Assembly name]
SprCSharp
[Application]–[Default namespace]
SprCSharp
[Build]–[Output path]
(x86, x64) と (Debug, Release) との各組合せについて、
.. \<14.0>\bin\x86\Debug\
.. \<14.0>\bin\x86\Release\
.. \<14.0>\bin\x64\Debug\
.. \<14.0>\bin\x64\Release\
の中から適切なものを設定

4.3.4 SprCsSample の設定

4.3.4 SprCSharp と同様 (SprCSharp を SprCsSample と読み替える)

4.3.5 SprExport

Solution Explorer で [SprCSharp]–[Properties]–[Configuration Properties] を選択

‘All Configurations’ – ‘All Platforms’ について
(ただし、‡ を付した項目については、指定された Configuration/Platform について)

[General]
[Output Directory]
\$(SolutionDir)\$(VisualStudioVersion)\bin\ (次の行に続く)
\$(PlatformShortName)\\$(Configuration)\
[Intermediate Directory]
\$(VisualStudioVersion)\\$(Platform)\\$(Configuration)\
[Target Name]
SprExport
[Character Set]
Use Multi-Byte Character Set
[Common Language Runtime Support]
No Common Language Runtime Support
[C/C++]
[General]–[Additional Include Directories]

```

    ../../../../include;../../../../src;%(AdditionalIncludeDirectories)
[General]-[Common Language Runtime Support]
    No Common Language Runtime Support
[General]-[SDL checks]
    Yes (/sdl)
[General]-[Multi-processor Compilation]
    Yes (/MP)
[Optimization]-[Optimization]
    ‡ Debug では      Disables (/Od)
    ‡ Release では    Maximize Speed (/O2)
[Optimization]-[Enable Intrinsic Functions]
    ‡ Debug では      No
    ‡ Release では    Yes (/Oi)
[Optimization]-[Favor Size Or Speed]
    ‡ Debug では      Neither
    ‡ Release では    Favor fast code (/Ot)
[Optimization]-[Omit Frame Pointers]
    ‡ Debug では      No (/Oy-)
    ‡ Release では    Yes (/Oy)
[Optimization]-[Whole Program Optimization]
    No
[Code Generation]-[Enable C++Exceptions]
    Yes with SEH Exceptions (/EHa)
[Code Generation]-[Enable Floating Point Exceptions]
    Yes (/fp:except)
[Precompiled Headers]-[Precompiled Header]
    Not Using Precompiled Headers
[Advanced]-[Disable Specific Warnings]
    4244;4267
[Linker]
[General]-[Enable Incremental Linking]
    No (/INCREMENTAL:NO)
[General]-[Additional Library Directories]
    ‡ Win32 では      ../../../../lib/win32
    ‡ x64 では        ../../../../lib/win64
[Input]-[Additional Dependencies]
    XINPUT9_1_0.LIB;%(AdditionalDependencies)
[Input]-[Ignore Specific Default Libraries]
    XINPUT.LIB

```

4.3.6 SprlImport の設定

4.3.6 SprCSharp と同様 (SprCSharp を SprImport と読み替える)

4.3.7 TargetManager

全体構成 ('ALL') でも Physics 構成 ('PH') でもない場合のみ次の処理が必要 (プロジェクト 'TargetManager' を新規に作成した場合)

Solution Explorer で [TargetManager]–[Properties] を選択

‘All Configurations’ – ‘All Platforms’ について

```
[Configuration Properties]–[General]
  [Output Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
  [Intermediate Directory]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\
  [Configuration Type]
    Makefile
  [Build Log File]
    $(VisualStudioVersion)\$(Platform)\$(Configuration)\ (次の行に続く)
    $(MSBuildProjectName).log
[Configuration Properties]–[NMake]
  [Build Command Line]
    TargetManager <Physics>
  [Rebuild All Command Line]
    TargetManager <Physics>
  [Clean Command Line]
    (空白)
  [Output]
    (空白)
```

- 以上 -