

32bit・64bit 環境兼用インストーラーの作成 (InstallScript プロジェクト)

注)このドキュメントは、InstallShield 2011 Premier Edition を基に作成しています。InstallShield 2011 以外のバージョンでは設定名などが異なる場合もあります。

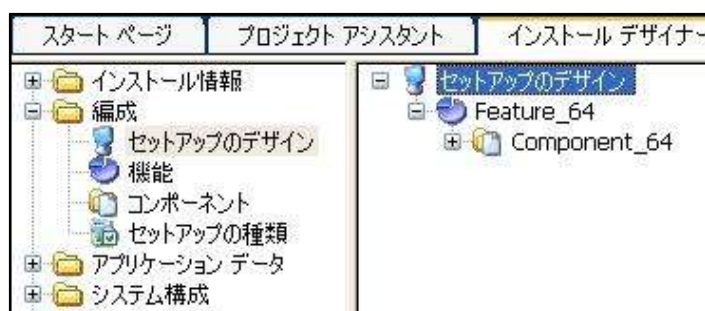
概要

InstallScript 形式プロジェクトにて、32Bit・64Bit両方の環境に対応したインストーラを作成する場合、32Bit環境用のモジュールを含む機能と64Bit環境用のモジュールを含む機能を作成して、特定の InstallScript コードを追加することで対応可能です。この記事ではInstallScript 形式プロジェクトを使用して、32bit環境・64bit環境の両方に対応したインストーラの作成する場合の手順について説明します。

A. 32bit用機能・64bit用機能の作成

32bit用のファイル・レジストリエントリを含む機能と64bit用のファイル・レジストリエントリを含む機能をそれぞれ作成します。

1. [編成]－[セットアップのデザイン]にて、既存の機能[DefaultFeature]とその配下のコンポーネント [DefaultComponent]をそれぞれ、[Feature_64][Component_64]とリネームします。



2. 同ビューにて、[セットアップのデザイン]を右クリックして[新しい機能]を選択します。新規追加された機能の名称を [Feature_32]に変更します。機能 Feature_32 を右クリックして[新しいコンポーネント]を選択します。新規追加されたコンポーネントの名称を[Component_32]に変更します。



3. コンポーネントを展開して[スタティック ファイルリンク]を選び、右のウィンドウにて Component_32 には 32bit 環境用のファイルを、Component_64 には 64 bit 環境用のファイルを含めます。



※32bit用のファイルと64bit用のファイルが完全に同名であり、かつ同階層に転送を行う構成の場合、非圧縮形式としてビルドを行うとインストーラに正しくファイルを含めることができません。この構成の場合は機能の[CD-ROM フォルダー]設定に任意の名称を設定してください。



4. 本件のサンプルでは、InstallScript によってターゲットOSを判定し、適切な機能を自動的に選択してインストールを行います。そのためこれら2つの機能 (Feature_32, Feature_64) に関しては、インストーラのダイアログよりユーザが間違えて選択を行ってしまわないように、機能の[表示]を[いいえ]に設定します。



B. レジストリセットの作成

32bit 用・64bit 用のレジストリセットを作成して、各コンポーネントに割り当てます。

1. [システム構成]－[レジストリ]ビューにて、[インストール先のコンピュータ]を右クリックして[レジストリ セットの新規作成]を選択します。新規作成されたレジストリセットを選択します。
2. 名称を[Registry_64]とリネームします。右のウィンドウ[レジストリ セットのインストール条件]にコンポーネント [Component_64]にのみ関連づけられるようにチェックをつけます。



3. HKEY_LOCAL_MACHINE を右クリックして、[新規作成]－[キー]を選び 64 bit 環境のレジストリキーを追加します。作成されたキーを選択して、右側のウィンドウ[インストール先コンピュータのレジストリ データ]を右クリックします。[新しい文字列]等を選び任意のキーを追加します。



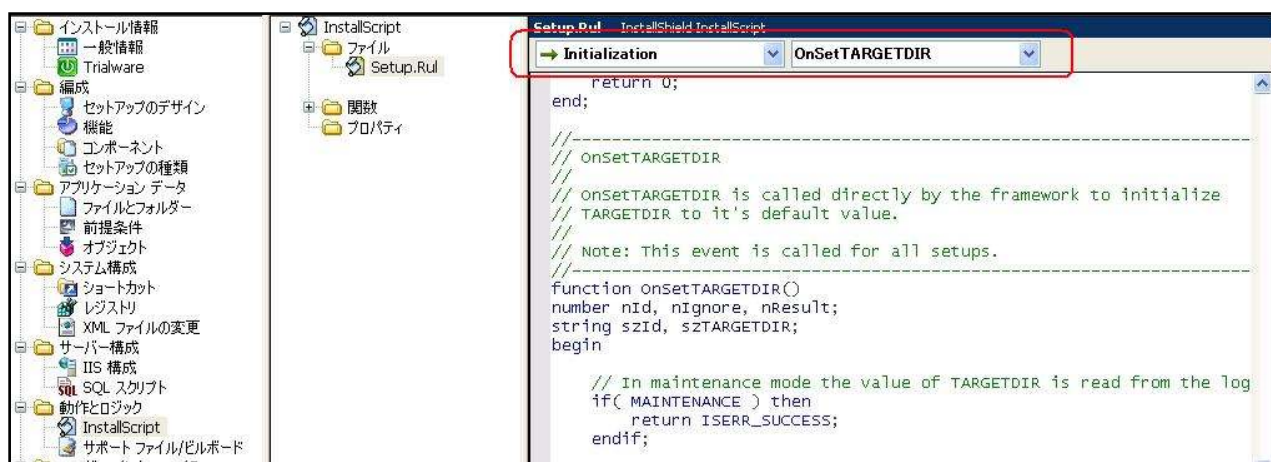
4. 上記の手順を繰り返して、32bit 用のレジストリセットを作成します。(セット名は Registry_32 とします。)



C. TARGETDIRの動的切り替えを行う InstallScript コードの追加

実行環境に応じて、TARGETDIR のパスを動的に切り替えるためのコードを追加します。

1. [動作とロジック]—[InstallScript]ビューにて、[Setup.Ru]を選択します。
2. スクリプトエディタ上部のコンボボックスを[Initialization]—[OnSetTARGETDIR]に切り替えて、OnSetTARGETDIR のコードを追加します。



3. OnSetTARGETDIR に以下のコードを追加します。

(灰色の部分が追加箇所です)

```

:
:

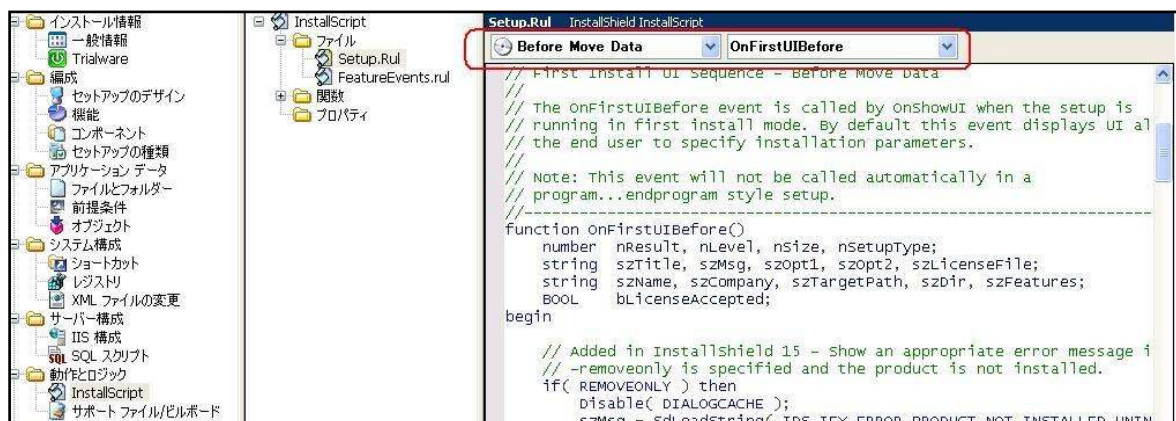
// Use the TARGETDIR from the media if anything was read.
if( nResult >= ISERR_SUCCESS && StrLengthChars( szTARGETDIR ) ) then
TARGETDIR = szTARGETDIR;
endif;

//***** 追加行
if ( SYSINFO.bIsWow64 ) then
//64bit 環境だった場合
TARGETDIR = "<FOLDER_APPLICATIONS64>%*3264combine_sample";
else
//32bit 環境だった場合
TARGETDIR = "<FOLDER_APPLICATIONS>%*3264combine_sample";
endif;
//***** ここまで

:
:
```

D. 32Bit 用・64Bit 用機能の自動切り替えを行う InstallScript コードの追加

1. スクリプトエディタ上部のコンボボックスを [Before Move Data]-[OnFirstUIBefore] に切り替えて、OnFirstUIBefore のコードを追加します。



2. OnFirstUIBefore に以下のコードを追加します。

```

:
:
Dlg_SdStartCopy2:
  szTitle = "";
  szMsg = "";
  {{{IS_SCRIPT_TAG(Dlg_SdStartCopy2)
  nResult = SdStartCopy2( szTitle, szMsg );
  }}}IS_SCRIPT_TAG(Dlg_SdStartCopy2)
  if (nResult = BACK) goto Dlg_ObjDialogs;

***** 追加行
  if ( SYSINFO.bIsWow64 ) then
    //64bit 環境だった場合
      //64bit 用機能を選択
      FeatureSelectItem(MEDIA,"Feature_64",TRUE);
      //32bit 用機能を除外
      FeatureSelectItem(MEDIA,"Feature_32",FALSE);
  else
    //32bit 環境だった場合
      //64bit 用機能を除外
      FeatureSelectItem(MEDIA,"Feature_64",FALSE);
      //32bit 用機能を選択
      FeatureSelectItem(MEDIA,"Feature_32",TRUE);
  endif;
***** ここまで

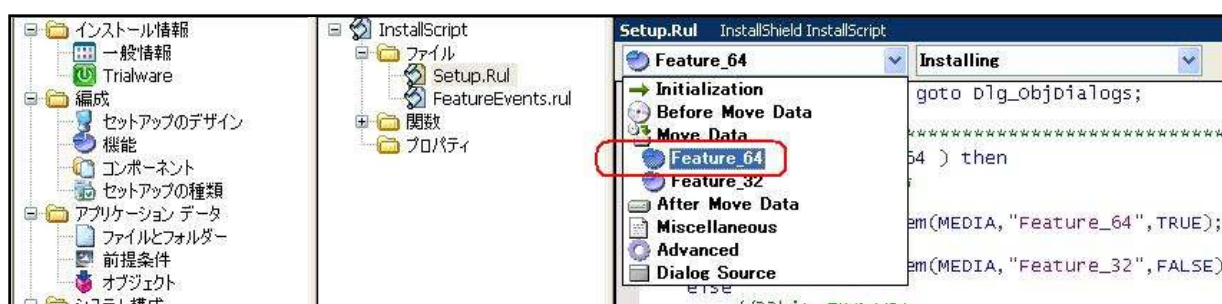
// Added in 11.0 - Set appropriate StatusEx static text.
  SetStatusExStaticText( SdLoadString( IDS_IFX_STATUSEX_STATICTEXT_FIRSTUI ) );
  :
  :

```

E.. 64bit 領域へファイル・レジストリの転送を行う InstallScript コードの追加

InstallScript インストーラでは、デフォルトの設定の場合、64bit 環境の 64bit 領域へファイル・レジストリエントリを作成することはできません。(C:\Program Files (x86) や レジストリでは HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node に転送が行われてしまいます。)64bit 領域へインストールを行う場合、リダイレクトを一時的に無効に設定する必要があります。

1. スクリプトエディタ上部のコンボボックスを [Feature_64]-[Installing] に切り替えて、Feature_64 機能のインストール中に実行されるコードを追加します。



2. Feature_64_Installing に、以下のコードを追加します。

```
export prototype Feature_64_Installing();
```

```
function Feature_64_Installing()
begin
```

```
    //File の COM 登録を正常に行うため、一時的にバッチメソッドを無効にします。
    Disable( SELFREGISTERBATCH );
```

```
    //64bit 領域(64Bit Windows フォルダ)に正しくファイルを転送するため、
    //一時的にリダイレクトを無効にします。
    Disable( WOW64FSREDIRECTION );
```

```
    //64bit 領域(wow64node ではない通常の箇所)に正しくレジストリエントリを作成するため、
    //一時的にリダイレクトを無効にします。
    REGDB_OPTIONS = REGDB_OPTIONS | REGDB_OPTION_WOW64_64KEY;
```

```
end;
```

3. スクリプトエディタ上部のコンボボックスを[Feature_64]-[Installed]に切り替えて、Feature_64 機能のインストール後に実行されるコードを追加します。

```

export prototype Feature_64_Installed();
function Feature_64_Installed()

begin

//バッチメソッドを有効にします。
Enable( SELFREGISTERBATCH );

//リダイレクトを有効にします。
Enable( WOW64FSREDIRECTION );

//REGDB_OPTION_WOW64_64KEY に ~ をつけて指定して、リダイレクトを有効に戻します。
REGDB_OPTIONS = REGDB_OPTIONS & ~REGDB_OPTION_WOW64_64KEY;

end;

```

4. 上記と同じ手順で、アンインストール中のイベント UnInstalling UnInstalled にも同じようにコードを追加します。

```

export prototype Feature_64_UnInstalling();
function Feature_64_UnInstalling()
begin

//File の COM 登録を正常に行うため、一時的にバッチメソッドを無効にします。
Disable( SELFREGISTERBATCH );

//64bit 領域(64Bit Windows フォルダ)に正しくファイルを転送するため、
//一時的にリダイレクトを無効にします。
Disable( WOW64FSREDIRECTION );

//64bit 領域(wow64node ではない通常の箇所)に正しくレジストリエントリを作成するため、
//一時的にリダイレクトを無効にします。
REGDB_OPTIONS = REGDB_OPTIONS | REGDB_OPTION_WOW64_64KEY;

end;

```

```

export prototype Feature_64_UnInstalled();
function Feature_64_UnInstalled()
begin

//バッチメソッドを有効にします。
Enable( SELFREGISTERBATCH );

//リダイレクトを有効にします。
Enable( WOW64FSREDIRECTION );

//REGDB_OPTION_WOW64_64KEY に ~ をつけて指定して、リダイレクトを有効に戻します。
REGDB_OPTIONS = REGDB_OPTIONS & ~REGDB_OPTION_WOW64_64KEY;

end;

```

F. インストーラのビルド / 32bit・64 bit 環境での実行

ビルドを行い、実行時の動作を確認します。

1. ツールバーの[ビルド]ボタンをクリックして ビルドを実行します。ビルドは[F7]キーからも実行可能です。



2. 生成された InstallScript インストーラを 32bit/64bit 環境でそれぞれ実行して、動作を確認します。
(画像は 64Bit 環境で実行した場合)

