

デバイスドライバーのインストール

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

InstallShield のインストーラは DIFX(Microsoft Windows Driver Install Framework) に準拠したデバイスドライバーのインストールをサポートしています。この記事では、基本のMSI 形式インストーラにてデバイスドライバーのインストールを行う場合の手順についてご説明いたします。

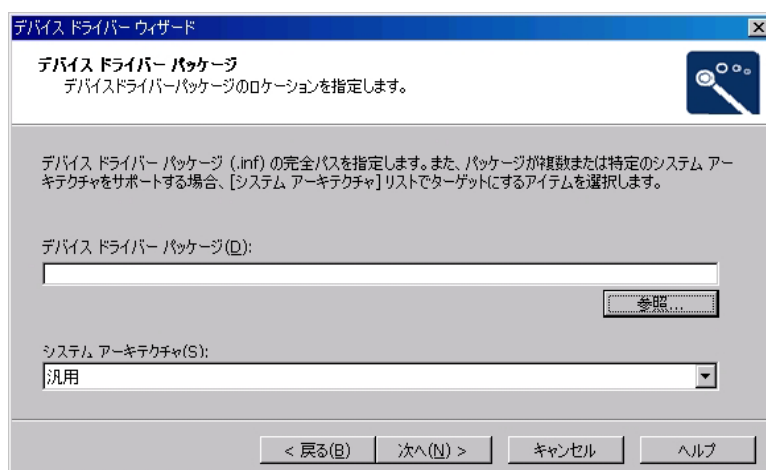
A. デバイスドライバーウィザードによるドライバーの追加

デバイスドライバーウィザードを使用してプロジェクトへデバイスドライバーファイルを追加します。

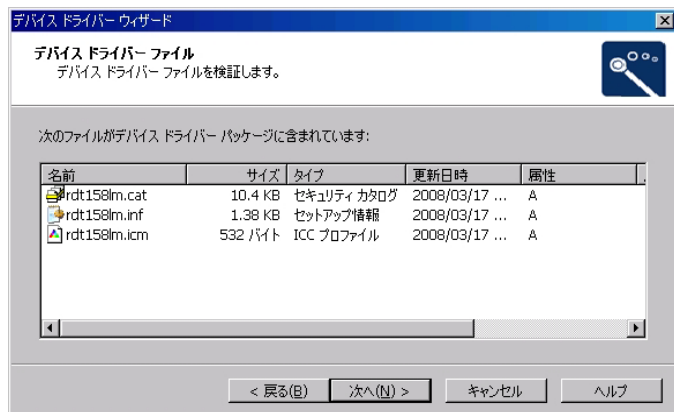
1. [編成]—[セットアップのデザイン]ビューにて、[セットアップのデザイン]を右クリックして、[新しい機能]を選択します。追加された機能を右クリックして、[デバイスドライバーウィザード]を選びます



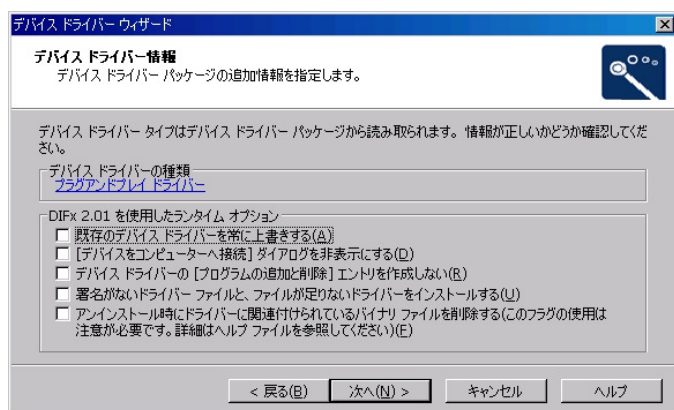
2. デバイスドライバーウィザードが起動します。[デバイスドライバーパッケージ]ダイアログで、インストーラに含める対象となるデバイスドライバーパッケージ(.inf ファイル)を指定します。



3. [デバイスドライバーファイル]ダイアログでは、指定したデバイスドライバーパッケージファイルに基づいてインストーラに含まれるファイルが表示されます。



4. [デバイスドライバー情報]ダイアログにて、デバイスドライバーの種類の確認、および適宜指定を行うオプションが設定可能です。



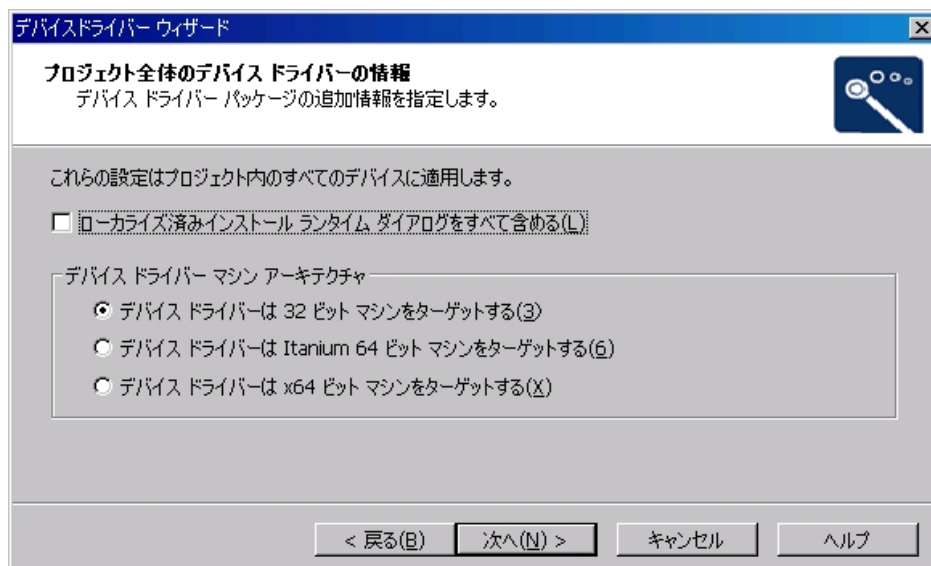
● デバイス ドライバーの種類

デバイス ドライバーの種類は、パッケージファイル(inf ファイル)の [Version] セクション DriverPackageType の記載により判別されます。

DriverPackageType 値	対応するドライバー タイプ
ClassFilter	クラスフィルター ドライバー
FileSystem	ファイルシステム ドライバー
FileSystemFilter	ファイルシステム フィルタードライバー
KernelModule	エクスポート ドライバー
KernelService	カーネルサービス ドライバー
Network	ネットワーク ドライバー
PlugAndPlay	プラグアンドプレイ ドライバー

※ DriverPackageType に記載が無い場合、種類は常に「Plug and Play ドライバー」と判別されます。

5. [プロジェクト全体のデバイス ドライバーの情報]ダイアログでは、デバイスドライバーのターゲットとなるマシンのアーキテクチャ(32BitCPU、64BitCPU 等)の指定を行います。



6. メニューバー[ビルド]ー[ビルド(B)] または、F7 ボタンよりプロジェクトのビルドを行ないます。生成された MSI インストーラの動作を確認します。

B. 32bit 用 MSI ファイルと 64Bit 用 MSI ファイルを同じプロジェクトを使用して生成する

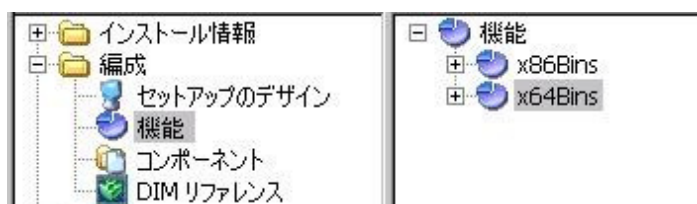
※手順 B は InstallShield 2011 以降の環境でお使いいただけます

MSI 形式インストーラーでは Windows Installer の仕様により 32Bit 環境と 64Bit 環境の両方に対応したパッケージを作成することはできません。32Bit 用・64Bit 用 のデバイスドライバーを含んだインストーラーに関して、それぞれ別々のパッケージとして MSI インストーラを作成する必要があります。

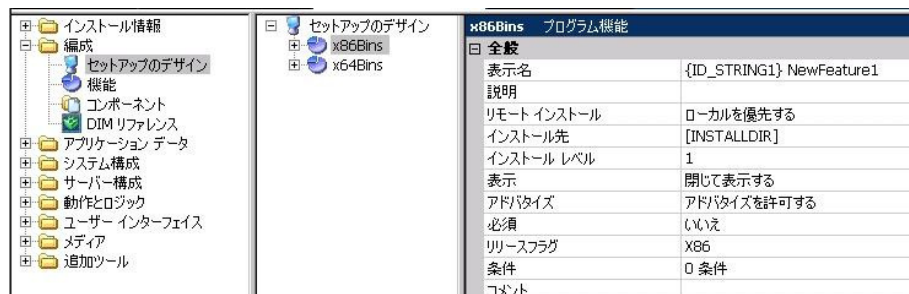
InstallShield では、[リリースフラグ]の機能を使用することで、同一のプロジェクトを使用して 32Bit 用・64Bit 用のインストーラをビルドすることが可能です。

(上記内容の詳細につきましては [別資料「32bit 用インストーラと 64Bit 用インストーラを同一のプロジェクトで作成する」](#)をご参照ください。)

1. [編成]ー[セットアップのデザイン]にて、二つの機能を作成します。それぞれの機能が 32Bit 用・64Bit 用機能となりますので、分かりやすい名称に変更します。



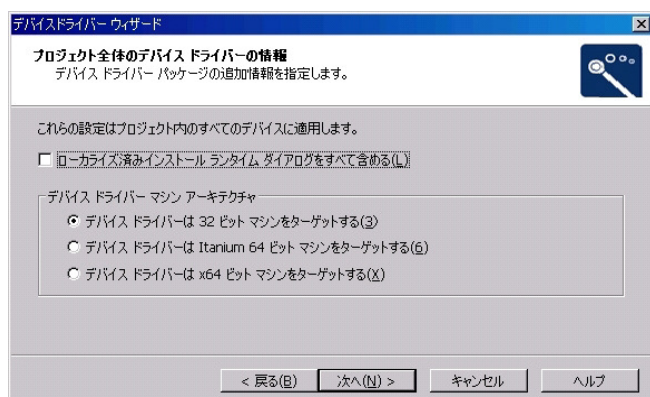
2. それぞれの機能に対してリリースフラグを設定します。機能を選択して右のビュー[全般]ー[リリースフラグ]に任意の文字列を設定します。以下の図では、32Bit 用機能に「X86」というリリースフラグを設定しています。64Bit 用機能には「X64」というリリースフラグを設定します



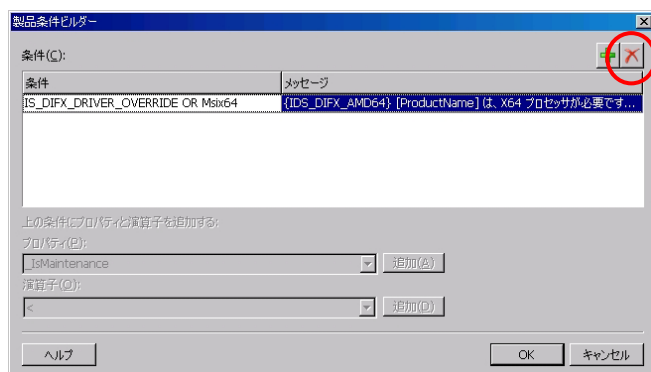
3. 作成した機能を右クリックして、[デバイスドライバーウィザード]を選択します。前述の手順Aを元に、それぞれの機能に 32Bit 用デバイスドライバー・64Bit 用デバイスドライバーを含めます。

※注意※

ウィザードの最終ダイアログ [プロジェクト全体のデバイス ドライバーの情報]ダイアログの設定は、64Bit用デバイスドライバーを追加する場合も、[デバイス ドライバーは32ビットマシンをターゲットする]を選択してください。



このダイアログにて、[デバイス ドライバーはx64 ビット マシンをターゲットにする]を選んだ場合、自動的に64Bit 環境のみでの実行を許可するインストール条件が追加されてしまうため、32Bit環境で実行が行えなくなります。インストール条件が追加されてしまった場合は、[インストール情報]ー[一般情報]ー[全般]ー[インストール条件]にて、条件の削除を行ってください。(下図のダイアログで赤い丸印のボタンを押します)



- 32Bit 用デバイスドライバーのインストールを 64Bit 環境で許可しない場合、エラーのカスタムアクションを追加します。[動作とロジック]—[カスタム アクションとシーケンス]ビューにて、[カスタム アクションを]右クリックして、[新しいエラー]を選びます。



- 新規追加されたエラーカスタムアクションを選択して、右のウィンドウにて以下の設定を行います。

設定項目名	設定内容	説明
エラーメッセージ	[ProductName] は、X86 プロセッサが必要です。「OK」をクリックするとウィザードが閉じます。	表示するメッセージを指定します
インストール UI シーケンス	<最初のアクション>	
インストール UI 条件	(Not Installed) And (ISReleaseFlags><"X86") And VersionNT64	ISReleaseFlags プロパティにリリースフラグ X86 が含まれている場合かつ 64Bit 環境で実行されている
インストール 実行シーケンス	<最初のアクション>	
インストール 実行条件	(Not Installed) And (ISReleaseFlags><"X86") And VersionNT64	ISReleaseFlags プロパティにリリースフラグ X86 が含まれている場合かつ 64Bit 環境で実行されている

※ 64Bit 用 MSI インストーラは、32Bit 環境で動作しませんので、上記のカスタムアクションを別途追加する必要はございません。

- カスタムアクションが使用する DLL ファイルをリリースごとに切り替えるため、カスタムのパス変数を定義します。[メディア]—[パス変数]ビューにて、[新規作成]選択します。以下のパス変数を追加します。

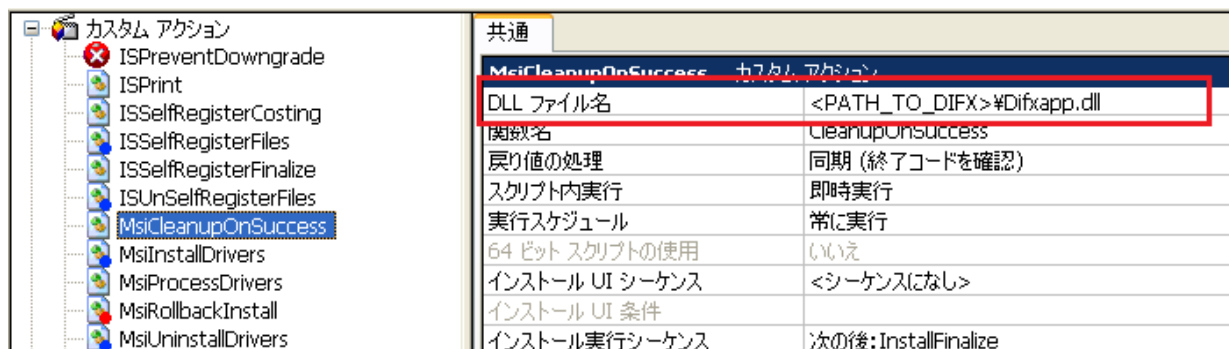
名前: PATH_TO_DIFX 定義された値: <ISProductFolder>%redist%0409%386



7. カスタムアクションが使用する DLL ファイルの参照先を手順 6 で作成したパス変数を使用する記述に変更します。[動作とロジック]—[カスタム アクションとシーケンス]ビューにて、以下のカスタムアクションの[DLL ファイル名]を修正します。

カスタムアクション名	変更前	変更後
MsiCleanupOnSuccess	<ISProductFolder>%redist%0409%386%Difxapp.dll	<PATH_TO_DIFX>%Difxapp.dll
MsiInstallDrivers	<ISProductFolder>%redist%0409%386%Difxappa.dll	<PATH_TO_DIFX>%Difxappa.dll
MsiProcessDrivers	<ISProductFolder>%redist%0409%386%Difxapp.dll	<PATH_TO_DIFX>%Difxapp.dll
MsiRollbackInstall	<ISProductFolder>%redist%0409%386%Difxappa.dll	<PATH_TO_DIFX>%Difxappa.dll
MsiUninstallDrivers	<ISProductFolder>%redist%0409%386%Difxappa.dll	<PATH_TO_DIFX>%Difxappa.dll

※ 指定する DLL ファイル名は2種類(Difxapp.dll と Difxappa.dll)ある点にお気をつけください



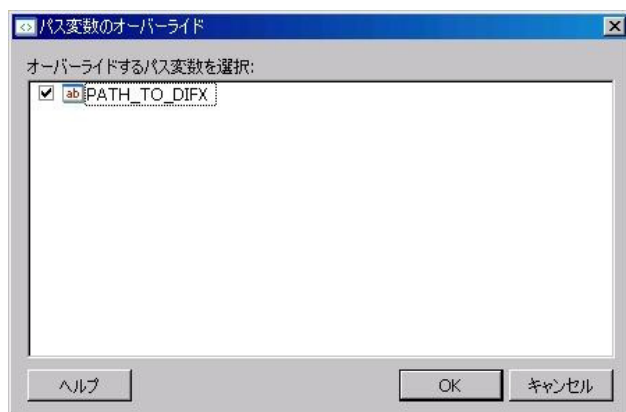
カスタムアクションアイコンの下に該当のカスタムアクションが見えない場合には、[すべてのカスタムアクションを表示]をクリックしてください

8. [メディア]—[リリース]ビューにて、製品構成を二つ作成します。それぞれの製品構成に各環境に対応した[製品構成フラグ][テンプレート概要]を設定します。(以下の画像では、64Bit 環境用の製品構成に製品構成フラグ[X64]、テンプレートの概要に[x64;1041]を指定しています)



※ 製品構成フラグは手順2で指定したリリースフラグと同等の文字列を使用します。例えばリリースフラグとして「X86」が指定された機能に関しては、製品構成フラグを[X64]としてビルドを行った場合、インストーラに含められません。

9. 設定を行った製品構成の配下にリリースを追加します。作成したリリースを選択して、右のビュー[ビルド]タブ [パス変数のオーバーライド]をダブルクリックします。表示されるダイアログにて、手順 6 で追加したパス変数 PATH_TO_DIFX にチェックをつけます。



10. [パス変数のオーバーライド]項目の下に追加される[PATH_TO_DIFX]パス変数のエントリに対して、32bit 用リリース・64Bit 用リリースに対してそれぞれ以下の階層を指定します。

32Bit 用リリース: <ISProductFolder>%redist%0409%i386

リリースフラグ	
パス変数のテスト値を使用する	いいえ
<input checked="" type="checkbox"/> パス変数のオーバーライド	1 パス変数
PATH_TO_DIFX	<ISProductFolder>%redist%0409%i386
未使用のディレクトリを保持する	いいえ
Trialware ビルドの無効化	いいえ
DIM の検証	いいえ

64Bit 用リリース: <ISProductFolder>%redist%0409%x64

リリースフラグ	
パス変数のテスト値を使用する	いいえ
<input checked="" type="checkbox"/> パス変数のオーバーライド	1 パス変数
PATH_TO_DIFX	<ISProductFolder>%redist%0409%x64
未使用のディレクトリを保持する	いいえ
Trialware ビルドの無効化	いいえ
DIM の検証	いいえ

11. リリースを右クリックして[ビルド]を選択します。作成したリリースをそれぞれビルドします。生成されたインストーラを、32Bit、64Bit 環境で実行して動作を確認します。