<インストール実行時に、Windows の機能を有効化する方法>

注)このドキュメントは、InstallShield 2014 Premier Edition を基に作成しています。InstallShield 2014 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

インストーラーの実行中に、IIS(インターネット インフォメーションサービス)などの「Windows の機能」を有効化したい場合があります。

InstallShield の Premier Edition で提供されるスイート/アドバンストUI プロジェクトでは、Windows の機能を有効化 するための機能があり、製品インストーラーの実行時に Windows の機能を有効化することができます。

一方、Professional Edition では、Windows の機能を有効化するための機能は利用できないため、「展開イメージの サービスと管理 (DISM.exe) ツール」を使用して、コマンドをカスタムアクションから呼び出すことで、インストール時 に有効化させることが可能です。

DISM.ツールに関しての詳細は、以下をご参照ください。

DISM を使って Windows の機能を有効または無効にする https://msdn.microsoft.com/ja-jp/library/hh824822.aspx

この記事では、それぞれの Edition で「Windows の機能」を有効化する方法を紹介します。

A. Premier Edition のスイート/アドバンストUIプロジェクトを使用する場合

スイート/アドバンストUI プロジェクトを使用すると、製品自体のインストーラーを変更することなく、Windowsの機能の 有効化を行うことができます。今回は、IIS(インターネット インフォメーションサービス)を有効化する手順を紹介しま す。

- 1. スイート/アドバンスト UI プロジェクトを作成します
- 2. 機能を作成し、機能の内部名と[表示名]を設定します(内部名:ProductX、表示名:製品Xなど)

📮 🛅 インストール情報	□- 🕘 機能	ProductX 機能	
─────────────────────────────────────		□ 全般	
🚊 🧰 編成		表示名	{ID_STRING} 製品X
		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	
		表示	(まい)
🚊 🛅 動作とロジック		171 JZF	0
- 🕎 InstallScript		条件	(空白)
🧟 サポート ファイル		リリース フラグ	
		UI 選択の変更を許可する	はい

また、含める.msi パッケージを追加します

□ 🔁 インストール情報	□-Q パッケージ	ファイル			
一般情報	ProductX	名前	サイズ	リンク先	更新日間
	⊞- <mark></mark> <u>N97</u> -> 7741µ	n ProductX.msi	676 KB	<path_to_製< td=""><td>2013/07</td></path_to_製<>	2013/07

●スイートインストーラーの基本的な作成方法 Tips and Tricks(http://www.networld.co.jp/is/tips.htm)から以下の記事をご参照ください。 <u>スイートインストーラーの基本的な作成方法</u> http://www.networld.co.jp/is/pdf/SuiteInstaller_Basic.pdf

- 3. 追加したパッケージ名を選択して、[共通]タブを表示します
- 4. [Windows の機能] の緑の[+]ボタンから、[インターネット インフォメーションサービス]を選択します

□-€ パッケージ	共通 機能			
⊟∽¶∰ ProductX के.ि !१७/ठ~२१ उन्नर ॥	ProductX パッケージ	😭		
	⊡ 全般	~		
	パッケージ GUID	{2A81AB6D-ECE1-4F30-9462-CB7481899015}		
	表示名	{ID_STRING1} ProductX		
	田 ロケーション	ソース メディアからコピーする		
	パッケージの種類	プライマリ		
	キャッシュパス	[LocalAppDataFolder]Downloaded Installations		
	昇格された権限が必要	はい		
	マイナー アップグレード処理	なし		
	レポートされるステータス メッセージ	アクション テキスト		
	■ ログの有効化	いいえ		
	検出条件	(空白)		
	Ⅲ 対象条件	(条件)		
	Windows の機能	0 定義済み 🛛 🗛		
	リリース フラグ	インターネット インフォメーション サービス		
	□ 操作	127 491 1228A 232 5 EX		
	■ インストール	(# PowerShell		
	Ⅲ 削除	(# Microsoft .NET Framework 3.x		
	Ⅲ 徑復	() +76/		
	⊡ 変更	(J ///94		

有効化したい機能名が追加されたことを確認します

Windows の機能	1 定義済み
Windows の機能	IIS-WebServerRole

5. ビルドして実行し、IIS が有効化されたことを確認します

●ビルトインで提供されていない Windows	vs 機能の有効化
スイートアドバンストUIプロジェクトでは、以 す。	以下の3つの Windows の機能がビルトインでサポートされていま
> インターネットインフォメーションサーと	ービス
PowerShell	
Microsoft .NET Framework 3.x	
これ以外の Windows 機能を有効化する場	場合には、[カスタム]を選択したあと、Windows の機能名を直接入力
します	
1117 1 AN-49824K	
Windows ())機能 リリース フラガ	
□ 操作	インターネット インフォメーション サービス
田 インストール	(t() PowerShell
田削除	(は) Microsoft .NET Framework 3.x
田変更	(at) カスタム
例)Telnet クライアントを有効化する場合	ĩ
I Windows の機能	1 定義済み
	TelnetClient
1版化白は、以下のようよ/Get-Features 2	イノノコノ Cher 記り るしこれ じさまり。
Dism /online /Get-Features	

B. Professional Edition でカスタムアクションを使用してDISMコマンドを実行する場合

InstallShield の Professional Edition ではスイート/アドバンスト UI プロジェクトは提供されていないため、方法 A を利用することができません。そのため、Professional Edition では、製品のインストーラー自体にカスタムアクション などを組み込む必要があります。Premier Edition でも、スイート/アドバンスト UI プロジェクトを使用したくない場合、 方法B を利用できます。

Windows の機能は、以下のDISM コマンドを実行することで有効化できます。 Dism /online /enable-feature /FeatureName: <機能名>

ここでは、IIS(インターネット インフォメーションサービス)を有効化するコマンドを .bat ファイルに記述して、そのバッチファイルを実行させることで、Windows の機能を有効化させる方法をご案内します。

<基本の MSI/ InstallScript MSI の場合>

1. バッチファイルを作成します

例:WindowsFeatureEnable.bat の内容

```
@echo off
echo IIS を有効化しています。このウィンドウは閉じないでください。
%windir%¥system32¥dism.exe /online /quiet /enable-feature /featurename:IIS-WebServerRole
```

- 2. 製品のプロジェクトを開き、「動作とロジック]ー「サポート ファイル」 ビューにて、 [サポートファイル] のツリーを展開して、 [言語非依存] を選択します
- 3. [ファイル]ペインを右クリックして、[ファイルの挿入]を選択します。ファイル選択のダイアログで、 "WindowsFeatureEnable.bat"を選択します
- 4. "WindowsFeatureEnable.bat"がサポートファイルとしてインストーラーに追加されたことを確認します

□ 🥃 サポートファイル	ファイル				
	名前	サイズ	タイプ	更新日時	Æ
	🚳 WindowsFeatureEnabe	152 /7구 ト	Windows /∖ຶyチ	2015/04/22 16:	ρ

5. [動作とロジック]-[カスタム アクションとシーケンス]ビューを表示し、[カスタムアクション]アイコンを右クリックして、[新しい EXE]-[ディレクトリを参照するパス]を選択します

👘 🚮	ላ ፖሳንョン カスタム アクシ	зン	
<u>⊕</u> ■ 7	カスタム アクション ウィザード(C)	Ins	
	新しい InstallScript(N)		が機能をインストールに実軟に追加できるよう設計さ によって実現できます。InstallShield は、数種の力;
	新しい EXE(E)	+	Binary テーブルに保存(S)
	新しい標準 DLL(S)	+	ディレクトリを参照するパス(P)
÷	新しい MSI DLL(M)	+	製品と共にインストール(I)
±	新しいマネージ コード(A)	+	プロパティ値のパス(V)
	新しいプロパティの設定(P)		起動する。

6. カスタムアクション名(WindowsFeatureEnable 等)を指定し、以下の設定を行います

設定プロパティ名	設定内容
作業ディレクトリ	SytemFolder
ファイル名とコマンドライン	cmd.exe /C "[SUPPORTDIR]¥WindowsFeatureEnable.bat"
スクリプト内実行	即時実行
インストール実行シーケンス	InstallFinalize の後
インストール実行条件	not Installed

🖃 🌠 カスタム アクション	共通	
WindowsFeatureEnable	WindowsFeatureEnable 7	コスタム アクション
● ● クリンコン テキスト	🗆 アクション	
■1日↓ ンニクノス	作業ディレクトリ	SystemFolder
■…● インストール	ファイル名とコマンドライン	cmd.exe /C "[SUPPORTDIR]¥WindowsFeatureEnable.bat"
	戻り値の処理	同期(終了コードを確認)
	スクリプト内実行	即時実行
	実行スケジュール	常に実行
	MSI タイプ番号	34
	コメント	
	ヘルプ ファイル パス	
	パッチのアンインストール中にま	ະ ເທເນີ
	回 シーケンス	
	団 インストール UI シーケンス	〈シーケンスになし〉
	□ インストール実行シーケンス	InstallFinalize の後
	インストール実行条件	not Installed

7. ビルドして、実行します

※コマンドの実行には管理者権限が必要なため、リリースの設定で「必要実行レベル」を「管理者」へ変更してください。

□ 😳 川−ス □ 🔓 製品構成 1	ビルド Setup.exe 署名 NET/J# インターネット イ	ベント
	リリース1 リリース	
●	□ セットアップ起動ツール	はい
	Setuplexe アイコン ファイル	
	MSI コマンドライン引数	
	パッケージ定義ファイルを作成する	いいえ
	必要実行レベル	管理者
	前提条件が昇格必要時のアドバタイズ	いいえ
	Ⅲ MSI エンジンを含める	いいえ
		1.0.5

インストーラーの実行中に以下の画面が表示され、IIS の有効化処理が実行されます

C:¥Windov	/s¥system32¥cmd.exe	- 🗆 🗙	
IIS を有効化しています。このウィンド	ウは閉じないでください。	^	
			- 🗆 🗙
			く
			ールしていま
i			
		~	
		< 戻る(B) 次へ(N)> キャンセル

<InstallScript プロジェクトの場合>

1. バッチファイルを作成します

例:WindowsFeatureEnable.bat の内容

@echo off echo IIS を有効化しています。このウィンドウは閉じないでください。 %windir%¥system32¥dism.exe /online /quiet /enable-feature /featurename:IIS-WebServerRole

- 2. 製品のプロジェクトを開き、「動作とロジック]ー「サポート ファイル/ビルボード」ビューにて、「サポートファイル]
 のツリーを展開して、「言語非依存」を選択します
- 3. [ファイル]ペインを右クリックして、[ファイルの挿入]を選択します。ファイル選択のダイアログで、 "WindowsFeatureEnable.bat"を選択します
- 4. "WindowsFeatureEnable.bat"がサポートファイルとしてインストーラーに追加されたことを確認します

□]] サポートファイル	ファイル						
	名前	サイズ	タイプ	更新日時	属性		
□ 本記 □ ■ ■ ビルボード	🖆 licensertf	58.4 KB	リッチ テキスト ド	2010/07/02 11:	A		
- 82 言語非依存	🚳 WindowsFeatureEnable.bat	152 バイト	Windows バッチ	2015/04/22 16:	A		

- 5. [動作とロジック]-[InstallScript] ビューで、[Setup.rul] を表示します
- 6. [イベントカテゴリ] から「After Move Data」を選択し、[イベント] から「OnFirstUIAfter」をクリックします
- 7. Setup.rul に OnFirstUIAfter イベントが追加されるので、以下のスクリプトを追加します

function OnFirstUIAfter()
:
string szProgram, szCmdline;
begin
szProgram = SUPPORTDIR + "WindowsFeatureEnable.bat";
szCmdline = ‴″;
LaunchApplication(szProgram, szCmdline ,″″, SW_NORMAL,INFINITE,LAAW_OPTION_WAIT);
· · ·

🖃 🕎 InstallScript	Setup.Rul InstallShield InstallScript
ーー ファイル └──☆ Setup Rul ーー 開数 └──☆ PB数 └──☆ OnFirstUIAfter ─── プロパティ	→ Initialization V OnSetTARGETDIR V
	<pre>// function OnFirstUIAfter() STRING szTitle, szMsg1, szMsg2, szOpt1, szOpt2; NUMBER bvOpt1, bvOpt2; string szProgram, szCmdline; 追加 begin</pre>
	<pre>szProgram = SUPPORTDIR + "WindowsFeatureEnable.bat"; szCmdline = ""; LaunchApplication(szProgram, szCmdline ,"", SW_NORMAL,INFINITE,LAAW_OPTION_WAIT);</pre>
	追加 ShowObjWizardPages(NEXT);

8. ビルドして、実行します

※コマンドの実行には管理者権限が必要なため、リリースの設定で「必要実行レベル」を「管理者」へ変更してください。

⊡-⊙ 	ビルド Setup.exe 署名 インターネット イベント		
	リリース 1 リリース		
	Single .exe ファイル名		
	Setuplexe アイコン ファイル		
	セットアップ コマンドライン		
	必要実行レベル	管理者	
	メディアのパスワード		

インストーラーの実行中に以下の画面が表示され、IIS の有効化処理が実行されます



以上