revenera

InstallShield 2021 Express Edition

ユーザー ガイド

法的情報

文書名:	InstallShield 2021 Express Edition ユーザー ガイド
部品番号:	ISE-2700-UG00
製品のリリース日:	2021 年 12 月

著作権情報

Copyright © 2020 Flexera.All Rights Reserved.

この出版物には、Flexera およびそのライセンサーによって所有されている機密情報、創造的な製作物が含まれています。本出版物の一部または全部を、Flexera からの事前の書面による明示的許可なしに、使用、複製、出版、配布、表示、改変または転載することはいかなる形態または手段を問わず厳重に禁止いたします。Flexera によって書面で明示されている場合を除き、この出版物の所有は、禁反言、黙示などによっても、Flexera が所有するいかなる知的財産権の下、ライセンスまたは権利を一切付与するものではありません。

本技術およびそれに関する情報のすべての複製は、Flexera より許可されている場合に限り、著作権および所有権に関する通知を完全な形で表示しなければなりません。

知的財産

Flexera が所有する商標および特許の一覧は、https://www.flexera.com/producer/company/about/intellectual-property/ を参照してください。Flexera 製品、製品ドキュメント、およびマーケティング資料で言及されているその他すべてのブランドおよび製品名は、各社の商標または登録商標です。

(米国内向け) 制限付権利に関する表示

本ソフトウェアは商業用コンピュータ ソフトウェアです。本ソフトウェアのユーザーまたはライセンス許可対象者が米国政府の代理、部署、その他の関連機関の場合、ソフトウェアまたは技術データおよびマニュアルを含むすべての関連文書の使用、複写、複製、開示、変更、公開、または譲渡に関して、ライセンス契約または本契約の条項ならびに民生機関については連邦調達規則第 12.212 条または軍事機関については国防連邦調達規則補遺第 227.7202 条による制限が適用されます。本ソフトウェアは完全に自費で開発されたものです。その他一切の使用は禁止されています。

目次

1 InstallShield 2021 – Express Edition ヘルプ ライブラリ 21

InstallShield 2021 エディション の新しい機能 22

InstallShield Express Edition 2021 R2 の新しい機能 23

InstallShield Express Edition 2021 R1 の新しい機能 23

InstallShield Express Edition の以前のバージョンの新機能 23

InstallShield 2020 Express Edition の新しい機能 24

InstallShield 2019 Express Edition の新しい機能 25

InstallShield 2018 Express Edition SP1 の新しい機能 28

InstallShield 2018 Express Edition の新しい機能 28

InstallShield 2016 SP2 Express Edition の新しい機能 32

InstallShield 2016 SP1 Express Edition の新しい機能 33

InstallShield 2016 Express Edition の新しい機能 33

InstallShield 2015 SP1 Express Edition の新しい機能 37

InstallShield 2015 Express Edition の新しい機能 37

InstallShield 2014 SP1 Express Edition の新しい機能 41

InstallShield 2014 Express Edition の新しい機能 41

InstallShield 2013 SP1 Express Edition の新しい機能 43

InstallShield 2013 Express Edition の新しい機能 46

InstallShield 2012 Spring SP1 Express Edition の新しい機能 48

InstallShield 2012 Spring Express Edition の新しい機能 49

InstallShield 2012 SP1 Express Edition の新しい機能 51

InstallShield 2012 Express Edition の新しい機能 51

InstallShield 2011 Express Edition の新しい機能 53

InstallShield 2010 Express Edition Expansion Pack for Visual Studio 2010 の新しい機能 60

InstallShield 2010 Express Edition SP1 の新しい機能 62

InstallShield 2010 Express Edition の新しい機能 62

InstallShield 2009 Express Edition の新しい機能 66

InstallShield 2008 Express Edition の新しい機能 72

InstallShield 12 Express Edition の新しい機能 79

ターゲット システムの要件 81

32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて 83
InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い 86
32 ビットと 64 ビット システムにおけるインストールの開発およびビルドの違い 87
ヘルプの使い方 89
お問い合わせ先 91

2 スタート ガイド 93
インストールの基本 94
インストールの概要 95
アプリケーション ライフサイクル 96
InstallShield の起動 97
InstallShield スタート ページ 97
プロジェクトについて 98
プロジェクトの種類 98
Express プロジェクト 99
QuickPatch プロジェクト 99
プロジェクトの使用 100
新規プロジェクトの作成 100
プロジェクトを開く 100
以前のバージョンの InstallShield で作成されたプロジェクトを開く 101
プロジェクトの保存 101
新しい名前と場所でプロジェクトを保存する 101
デフォルトのプロジェクトの場所を変更する 101
GUID 102
サンプル ファイル 103
プロジェクト アシスタント 103
プロジェクト アシスタントを使用する 103
プロジェクト アシスタント内を移動する 104
インストール デザイナーを開く 104
プロジェクト アシスタントを非表示にする 104
[アプリケーション情報] ページ 105
コントロール パネルの [プログラムの追加と削除] 105
インストールの会社名と製品名 105
[インストール要件] ページ 105
プロジェクト アシスタントでオペレーティング システム要件を指定する 105
インストールによる要件確認のタイミング 106
ソフトウェア要件のランタイム メッセージを変更する 106
カスタム インストール要件の作成 106
[インストール アーキテクチャ] ページ 107
プロジェクト アシスタントで機能を追加する 107
複数機能インストールを作成するかどうかを判断する 107
複数の機能を持つインストールを作成する 107
デフォルトの機能 108
機能の階層を定義する 108
[アプリケーション ファイル] ページ 108

プロジェクト アシスタントで機能にファイルを追加する 109
プロジェクト アシスタントで機能からファイルを削除する 109
固定のフォルダーの場所へファイルを追加する 109
追加の定義済みフォルダーを表示する 109
[アプリケーションのショートカット] ページ 110
ファイル拡張子 110
インストールに含まれていないファイルへのショートカットを作成する 110
プロジェクト アシスタントでデフォルトのショートカットを変更する 111
プロジェクト アシスタントでショートカットのターゲットにファイル拡張子を関連付ける 111
[アプリケーション レジストリ] ページ 111
レジストリの更新 112
プロジェクト アシスタントでレジストリ データを構成する 112
プロジェクト アシスタントでレジストリ データの値を変更する 112
機能にレジストリ データを関連付ける 112
レジストリ データで変数データ型を使用する 113
アプリケーション パス 113
[インストール インタビュー] ページ 114
プロジェクト アシスタントでインストールに使用するダイアログを指定する 114
エンドユーザーによるインストール先の変更を許可する 114
使用許諾契約書 115
インストールが選択できるインストールを作成する 115
[インストールのビルド] ページ 115
プロジェクト アシスタントからインストールをビルドする 116
プロジェクト アシスタントの完了後: 次のステップ 116
InstallShield インターフェイスを使って作業する 117
ビュー リストを表示する 117
Opening Views in the InstallShield ユーザー インターフェイスでビューを開く 117
様々なビューで、[グループ ボックス] 領域を使って作業する 117
ツールバーの表示または非表示 120
ツールバーにボタンおよびメニューを追加する 120
ツールバーからボタンおよびメニューを削除する 120
カスタム ツールバーの作成 121
[出力] ウィンドウを固定する/取り外す 121
InstallShield の詳細設定を構成する 122
デジタル署名のタイムスタンプ サーバーを変更する 122
Setup.exe にストリームされるファイルの圧縮レベルを構成する 123
.cab ファイルの最大サイズを構成する 125
デジタル署名のプラットフォーム アーキテクチャを構成する 126
InstallShield の以前のバージョンからの移行する 127
InstallShield 2014 Express Edition 以前からのプロジェクトをアップグレードする 127
InstallShield 2013 Express Edition 以前からのプロジェクトをアップグレードする 129
InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする 129
InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする 130
InstallShield 2011 Express Edition 以前からのプロジェクトをアップグレードする 131
InstallShield 2010 Express Edition 以前からのプロジェクトをアップグレードする 132
InstallShield 2009 Express Edition 以前からのプロジェクトをアップグレードする 134

InstallShield 2008 Express Edition 以前からのプロジェクトをアップグレードする 137
InstallShield 12 Express Edition 以前からのプロジェクトをアップグレードする 141
InstallShield Express 2.x からプロジェクトをアップグレードする 143

InstallShield Premier または InstallShield へのアップグレード 145

Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートする 150

InstallShield のアップデートを取得する 154

InstallShield 実行時の言語サポート 154
サポートされているランタイム言語 155

サポートされているアプリケーション プログラミング言語 157

3 チュートリアル 159

基本チュートリアル 161
プロジェクトの新規作成 161
セットアップの編成 161
アプリケーション データの指定 163
ターゲット システムの構成 163
セットアップ外観のカスタマイズ 164
セットアップ要件およびアクションの定義 165
リリースの準備 165
概要 167

4 インストールの作成 169

まず始めに 170
Windows Installer 入門 171
インストール中におけるユーザー アカウント制御のプロンプトの数を最小化する 171
Windows ロゴ プログラムの要件 175
現在のインストールによる同製品の将来のメジャー バージョンの上書きを防ぐ 175
非管理者パッチのインストールを準備する 176

インストール情報を指定する 177
一般的なプロジェクト設定を構成する 177
製品コードの設定 177
製品名の指定 177
製品バージョンを指定する 178
アップグレード コードを設定する 178
[プログラムの追加と削除] 情報を構成する 178
概要情報ストリーム データを入力する 178
デフォルトの製品インストール先フォルダー (INSTALLDIR) の設定 179
INSTALLDIR とレジストリ 180
INSTALLDIR のレジストリからの設定 180
ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護 180
プロジェクトで、ロックダウン環境でのアクセス許可タイプを選択する 182
Windows Installer インストールをログ記録するかどうかを指定する 183
製品のソフトウェア識別タグを含める 184

インストールのファイルを編成する 187
インストールをデザインする 187

アプリケーションを機能に分ける 187
機能の定義 188
機能の作成 188
サブ機能を作成する 189
機能の設定を構成する 189
機能の条件を設定する 189
機能をエンドユーザーへ表示する 190
機能のインストールを必須にする 191
機能のアドバタイズ 191
機能の "リモート インストール" 設定を設定する 192
[カスタム セットアップ] ダイアログで機能の順序を変える 192
セットアップの種類について 192
[セットアップの種類] で機能を指定する 193
セットアップの種類の名前を変更する 193
セットアップの種類を指定する 194
インストールにセットアップの種類を 1 つだけ含める 194
実行時にセットアップの種類にアクセスする 194
ファイルとフォルダーを含める 195
ファイルとフォルダーをプロジェクトに追加する 195
コンテキスト メニューを使ってファイルをドラッグ アンド ドロップする 197
64 ビット ソース マシンの 64 ビット System32 フォルダーからファイルを追加する 198
ハードコード化されたインストール先ディレクトリを指定する 199
ターゲット システムからファイルとフォルダーを削除する 200
プロジェクトでファイルとフォルダーを管理するときのヒント 200
個別のファイルに対してターゲット システム要件を指定する 201
ダイナミック ファイル リンク 202
ダイナミック ファイル リンクの制限事項 203
ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する 204
ファイルを動的に追加する 205
ターゲット マシン上でファイルを上書きする 206
[ファイル] ビューで定義済みフォルダーを表示する 207
プロジェクトでファイルとフォルダーを検索する 207
ファイルとフォルダーのアクセス許可を構成する 207
インストールに再配布可能ファイルを含める 208
再配布可能ファイルの出荷 209
再配布可能ファイルギャラリーを管理する 211
再配布可能ファイルをコンピューターにダウンロードする 213
再配布可能ファイルギャラリーに InstallShield 前提条件を追加する 214
再配布可能ファイルギャラリーから InstallShield 前提条件を削除する 214
マージ モジュールを参照する 214
マージ モジュールを参照した場合に起きること 215
再配布可能ファイル ギャラリーにマージ モジュールを追加する 215
再配布可能ファイルギャラリーからマージ モジュールを削除する 216
InstallShield 前提条件、マージ モジュール、およびオブジェクトをプロジェクトに組み込む 216
InstallShield 前提条件、マージ モジュール、およびオブジェクトをプロジェクトに追加する 216
プロジェクトから InstallShield 前提条件、マージ モジュール、またはオブジェクトを削除する 218

InstallShield 前提条件、マージ モジュールおよびオブジェクトのファイルを判別する *218*
プロジェクトに含まれている InstallShield 前提条件を使って作業する 218
セットアップ前提条件と機能前提条件の違い *219*
プロジェクトで、InstallShield 前提条件を機能に関連付ける *220*
プロジェクトで、機能から InstallShield 前提条件の関連付けを解除する *221*
InstallShield 前提条件のインストール順を指定する *221*
InstallShield 前提条件を含むリリースを構成する *222*
InstallShield 前提条件を含むディレクトリを指定する *222*
特定の InstallShield 前提条件の実行時の場所を指定する *223*
InstallShield 前提条件を含むリリースをビルドする *223*
InstallShield 前提条件を含むインストールの実行時の動作 *224*
インストールに InstallShield 前提条件が含まれていたアプリケーションをアンインストールする *227*
インストール プロジェクトに含まれているマージ モジュールとオブジェクトを使って作業する 227
マージ モジュールを含むディレクトリを指定する *228*
オブジェクトおよびマージ モジュールの構成を変更する *229*
マージ モジュールの除外と依存関係 *229*
マージ モジュールのインストール先をオーバーライドする *229*
マージ モジュールに関するトラブルシューティング *230*
Windows Installer 再配布可能ファイルをプロジェクトに追加する 230
Microsoft Windows Installer の前提条件を含める *231*
.NET Framework 再配布可能ファイルをプロジェクトへ追加する 231
Microsoft .NET Framework および Microsoft .NET Framework 言語パックの前提条件を含める *232*
DirectX 9.0 オブジェクトを含める 233
アプリケーションの依存関係を識別する 234
スタティック スキャン 234
ダイナミック スキャン 234
依存関係スキャナー結果の確認 235
依存関係スキャナーでファイルをフィルターする 235
COM サーバーの登録 237
COM 抽出のレジストリ変更をフィルターする 237
ターゲット システムの構成 241
ショートカットおよびプログラム フォルダーの作成 241
ショートカットの種類 241
ショートカットの作成 242
ショートカットのアイコンを指定する 242
ソース メディアにショートカットを置く 243
ショートカットにアクセスできるキーボード ショートカットを指定する 244
ショートカット名の変更 244
アンインストール ショートカットの作成 245
スタート画面上のデスクトップ アプリのタイルの外観を構成する 245
レジストリの編集 246
機能ごとにレジストリ エントリをフィルターする 247
レジストリキーの作成 248
レジストリ エントリをドラッグアンドドロップしてレジストリ キーを作成する 249
.reg ファイルからレジストリ データをインポートする 250
レジストリ キーの削除 251

レジストリ値の作成 251
レジストリ値データの変更 252
レジストリ値の削除 252
複数レジストリ文字列値を 1 行の文字列へ入力する 253
レジストリ エントリで環境変数を参照する 253
レジストリ キーのアクセス許可を構成する 254
Registry のプライマリ キーを指定する 254
レジストリ フラグ 255
レジストリ キーのインストール/アンインストール動作を設定する 256
ユーザーごとインストールでのレジストリ エントリの扱い方 256
[レジストリ] ビューのリフレッシュ 257
ファイル拡張子をアプリケーション アイコンに関連付ける 257
ファイル拡張子の関連付けを作成する 257
.ini ファイル データの変更 258
.ini ファイルの追加 258
既存の .ini ファイルをインポートする 259
.ini ファイルのセクションを指定する 259
.ini ファイルのキーワードとその値を指定する 260
ODBC リソースの構成 260
ODBC リソースを含める 260
追加の ODBC リソースを含める 260
ODBC リソースを機能に関連付ける 261
ODBC リソースの属性を設定する 262
新しい属性を ODBC リソースに追加する 262
ODBC リソースから属性を削除する 262
環境変数を使用する 263
環境変数の設定 263
Windows サービスのインストールおよび構成 263
ユーザーごとのインストールとマシンごとのインストールの違い 264
インストール動作のカスタマイズ 267
カスタム アクションを使用する 267
Windows Installer DLL カスタム アクション 269
MSI DLL カスタム アクションをプロジェクトに追加する 270
MSI DLL カスタム アクションの設定を構成する 270
DLL カスタム アクション 271
Classic DLL カスタム アクション関数プロトタイプ 271
新しい DLL カスタム アクション関数プロトタイプ 273
DLL カスタム アクションをプロジェクトに追加する 273
DLL カスタム アクションの設定を構成する 274
実行可能ファイル カスタム アクション 274
.exe カスタム アクションをプロジェクトに追加する 274
.exe カスタム アクションの設定を構成する 275
VBScript および JScript カスタム アクション 275
VBScript カスタム アクションの例 275
VBScript または JScript カスタム アクションをプロジェクトに追加する 276
VBScript または JScript カスタム アクションの設定を構成する 276

アクションの実行のオプション 276
カスタム アクションを起動するタイミングの変更 278
シリアル番号を検証するカスタム アクションを使用する 279
カスタム アクション ギャラリー 280
製品がインストールされているかどうかの確認をする 280
セットアップ ファイルを使用する 281
セットアップ ファイルの追加 281
ライセンス ファイルの追加 281
インストール中にセットアップ ファイルにアクセスする 282
サポート ファイルの並べ替え 282
Disk1 ファイル 282
Disk1 ファイルの追加 283
Disk1 フォルダーの追加 283
Disk1 ファイルとフォルダーを削除する 284
セットアップ ファイルを削除する 284
サーバーの構成 285
COM+ アプリケーションとコンポーネントの管理 285
COM+ アプリケーションの追加 285
COM+ アプリケーションの削除 286
インターネット インフォメーション サービス 286
InstallShield における IIS サポートのバージョン固有情報 286
IIS サポートの実行時要件 287
Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定する 288
Web サイトの作成とアプリケーションまたは仮想ディレクトリの追加 289
ネスト仮想ディレクトリの作成 291
TCP ポート番号とサイト番号の構成 291
Web サイトの IIS ホスト ヘッダー名を指定する 293
Web サイトの SSL 証明書を指定する 293
ファイルを IIS 仮想ディレクトリに追加する 294
[IIS 構成] ビューからアプリケーションと仮想ディレクトリを削除する 295
IIS サポートの機能の関連付け 295
Web サイト、アプリケーション、および仮想ディレクトリのアンインストール 295
Web サイトまたはアプリケーションの ASP.NET バージョンを設定する 295
Web サイト、アプリケーション、または仮想ディレクトリのアプリケーションのマッピングを定義する 296
Web サイトまたは仮想ディレクトリのタイムアウトのパラメーターを指定する 297
追加の IIS 仮想ディレクトリのプロパティを設定する 297
Web サイト、アプリケーション、または仮想ディレクトリのカスタム エラー メッセージを定義する 297
ターゲット マシン上で Web サービスを配布する 298
カスタム アクションを使って IIS 機能を抑制する 299
Web アプリケーションのフォーム認証を有効化する 299
IISROOTFOLDER サポートの追加 300
IIS_WEBSITE_NAME プロパティ 300
IIS_PORT_NUMBER プロパティ 300
エンドユーザー インターフェイスを定義する 301
ダイアログの使い方 301
インストールにダイアログを追加する 301

ダイアログのテーマ 301
ダイアログ テーマの選択または変更 *302*
ダイアログのテーマ *302*
ダイアログのビットマップ イメージ 302
[スプラッシュ ビットマップ] ダイアログのスプラッシュ ビットマップを変更する *305*
エンドユーザー ダイアログのビットマップ イメージを変更する *306*
エンドユーザー ダイアログのバナー ビットマップを変更する *306*
エンドユーザー ダイアログのグローバル ダイアログ イメージを変更する *307*
エンドユーザー ダイアログのグローバル ダイアログ バナーを変更する *307*
Windows Vista 以降のシステムの再起動を最小限にする 308
[カスタム セットアップ] ダイアログのオプション 308
実行時に使用許諾契約書を表示する 309
インストールからダイアログを削除する 309
実行時のテキストとメッセージを編集する 309
実行時の文字列を編集する 310
テキストとメッセージ文字列にコメントを追加する 310
テキストとメッセージ文字列のフォントを変更する 310
文字列のエクスポート 311
文字列テーブルのインポート 311
ビルボードを表示する 312
ビルボード ファイルの種類 312
ビルボードの種類 312
インストールで使用するビルボードの種類を指定する *315*
Adobe Flash アプリケーション ファイル ビルボードを追加する 315
イメージ ビルボードの追加 316
ビルボード設定の設定を構成する 316
リリースをビルドまたは起動せずにビルボードをプレビューする 316
ビルボードの順番を設定する 317
ビルボードを含むインストールの実行時の動作 317
ビルボードの削除 318
メンテナンスおよびアンインストールのためのインストールを作成する 319
製品が作成したレジストリデータの削除 319
インストールのビルド、テスト、および配布 321
リリースの構成とビルド 321
リリースのビルド 321
セットアップランチャーの作成 322
セットアップランチャーのファイルのプロパティをカスタマイズする *323*
64 ビット セットアップ ランチャーのビルド 326
ビルドのキャンセル 327
ビルド中に製品バージョンを変更する 327
リリースのビルド場所 328
ビルド ログとレポート 328
クイック ビルド 328
クイック ビルドを実行する *329*
コマンドラインを使ったビルド 329
コマンドラインからのビルド *329*

.ini ファイルでコマンドライン ビルド パラメーターを渡す *329*
Microsoft ビルド エンジン (MSBuild). 332
MSBuild を使用して、コマンドラインからリリースをビルドする. *337*
単一の自己展開型インストール ファイルを作成する. 338
Windows Vista 以降のプラットフォームでのセットアップランチャーの必要実行レベルを指定する 338
インターネットで配布するインストールを作成する 339
Web 配布リリースの作成 *341*
デジタル署名とセキュリティ. 341
ビルド時にリリースとそのファイルにデジタル署名を行う *343*
リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する 344
メディアの圧縮. 345
フロッピー ディスクの配布. 345
フロッピー ディスクで配布するリリースをビルドする *345*
ボリューム ラベルの設定 346
Windows Installer パッケージ (.msi) を出力として作成する. 346
Autorun 347
CD または DVD の Autorun を有効にする *347*
インストールのテストと実行. 347
インストールのテストと実行. 347
サイレント モードでインストールを実行する 348
条件インストール. 348
インストールの配布 348
フォルダーまたは FTP サイトにリリースを自動的に配布する. 348

5 アプリケーションのアップデート 351
アップグレードの概要 351
メジャーアップグレード. 351
マイナーアップグレード 352
スモール アップデート 353
パッチの適用 353
QuickPatch プロジェクト. 354
最適なアップグレード ソリューションの決め方. 355
アップグレードのパッケージ オプション. 356
アップグレードおよび QuickPatch プロジェクトを使用する 359
ファイルの上書き規則について. 359
パッケージ コード、製品バージョン、および製品コードのアップデート 359
完全インストール アップグレードの作成. 360
アップグレード中にグローバル アセンブリ キャッシュからアセンブリが削除されるのを防ぐ方法. 361
アップグレード項目の追加 362
アップグレード プロパティの構成 362
アップグレード項目の削除 363
パッチ時の考慮事項 363
パッチ シーケンス 364
パッチのアンインストール 366
非管理者パッチ. 366
QuickPatch プロジェクトの作成および QuickPatch パッケージの適用. 367

既存の QuickPatch をパッチする QuickPatch プロジェクトを作成する 367
QuickPatch パッケージを簡素化するかどうかを指定する. 368
パッチのターゲット リリースを指定する 369
QuickPatch を実行するカスタム アクションを指定する 369
QuickPatch へファイルを追加する 370
QuickPatch の識別情報を指定する. 370
QuickPatch のアンインストールを有効にする 371
QuickPatch パッケージをシーケンスする. 371
QuickPatch パッケージに署名する. 371
QuickPatch パッケージをパスワードで保護する 372
QuickPatch パッケージに Update.exe アップデート起動ツールをビルドをするかどうかを指定する 372
アップデート起動ツールのファイルのプロパティをカスタマイズする. 373
[パッチするファイル] エクスプローラーからファイルを削除する 374
QuickPatch を使用したインストール済みファイルの変更と削除. 374
QuickPatch を使用したレジストリデータの追加、変更および削除 375
グローバル アセンブリ キャッシュのアセンブリをパッチする. 375
QuickPatch の適用 376
FlexNet Connect を利用してエンドユーザーにアップグレードの通知をする 377

6 追加のインストール オプション 379

ターゲット システムの要件を指定する. 379
製品のオペレーティング システム要件を指定する 379
製品のプロセッサの要件を指定する 380
製品の RAM 要件を指定する 380
製品の画面の解像度要件を指定する 380
カラー深度の要件を指定する 381
製品のソフトウェアの要件を指定する. 381
Windows Installer プロパティを使用する 382
Windows Installer プロパティ リファレンス. 383
パブリック プロパティが制限付きパブリック プロパティである必要があることを指定する 400
実行時のシリアル番号検証を実装する. 401
インストール中とインストール後のシリアル番号へのアクセス 402

1 InstallShield プロジェクトを DevOps でビルドする. 403

Azure DevOps ビルド拡張. 403
InstallShield StandAlone Build と Docker. 404

2 InstallShield と外部アプリケーションの統合 405

Microsoft Visual Studio との統合. 405
Microsoft Visual Studio で InstallShield プロジェクトを作成する 406
Microsoft Visual Studio で InstallShield プロジェクトを開く 406
Visual Studio ソリューションで VSSolutionFolder パス 変数を使用する. 407
Visual Studio ソリューションにリファレンスを追加する. 407
InstallShield ツールバーまたはコマンドを Visual Studio ツールバーに追加する. 408
Microsoft Visual Studio でリリースをビルドする. 408

.NET アセンブリをプロジェクトに追加する 409
Web サービスまたは Web アプリケーションからプロジェクト出力を追加する 410
.NET Framework サポートをインストール プロジェクトへ追加する 410
Microsoft Visual Studio Team Foundation Server との統合 410
InstallShield プロジェクトを Team Explorer に追加する 411

3 リファレンス 413
メニュー、ツールバー、および ウィンドウのリファレンス 415
メニュー 415
[ファイル] メニュー 415
[編集] メニュー 416
[表示] メニュー 416
[移動] メニュー 417
[プロジェクト] メニュー 418
[ビルド] メニュー 419
[ツール] メニュー 419
[ヘルプ] メニュー 420
ツールバー 420
標準ツールバー 421
出力ウィンドウ 422
ダイアログ ボックス リファレンス 423
[.NET 1.1/2.0 コア言語] ダイアログ ボックス 424
[.NET 1.1/2.0 言語パック] ダイアログ ボックス 424
[MIME の種類を追加] ダイアログ ボックス 425
[アプリケーション拡張子マッピング] ダイアログ ボックス 425
[アプリケーションのマッピング] ダイアログ ボックス 426
[インストール先ファイルの参照] ダイアログ ボックス 427
[ディレクトリの参照/INSTALLDIR の設定] ダイアログ 427
[ファイルの参照] ダイアログ ボックス 428
ショートカットのターゲットを参照するダイアログ ボックス 430
[タイトル ターゲットの参照] ダイアログ ボックス 430
[証明書の選択] ダイアログ ボックス 430
[条件ビルダー] ダイアログ ボックス 433
[コンテンツ ソース パス] ダイアログ ボックス 434
[カスタム エラー] ダイアログ ボックス 434
[依存関係] ダイアログ ボックス 434
[ダイアログのイメージ] ダイアログ ボックス 435
[セットアップのデジタル署名] ダイアログ ボックス 435
[レジストリ データの編集] ダイアログ ボックス 437
[エラー マッピングのプロパティ] ダイアログ ボックス 437
[ファイルの詳細] ダイアログ ボックス 437
[ファイルのプロパティ] ダイアログ ボックス 438
[全般] タブ 439
[COM と .NET の設定] タブ 441
[詳細] タブ 442
ファイル 削除の [プロパティ] ダイアログ ボックス 444

[フォルダーのプロパティ] ダイアログ ボックス 445
[全般] タブ 446
[ファイルのリンク] タブ 446
[InstallShield 前提条件のプロパティ] ダイアログ ボックス 448
[Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックス 449
[マージ モジュールの構成可能な値] ダイアログ ボックス 450
[マージ モジュールのプロパティ] ダイアログ ボックス 451
[MIME の種類] ダイアログ ボックス 451
[MSI 値] ダイアログ ボックス 452
[複数行文字列値] ダイアログ ボックス 452
[新規プロジェクト] ダイアログ ボックス 453
[オプション] ダイアログ ボックス 453
[全般] タブ 454
[ファイルの場所] タブ 454
[プリファレンス] タブ 455
[マージ モジュールのオプション] タブ 455
[.NET] タブ 456
[ファイル ビュー] タブ 457
[ファイルの拡張子] タブ 458
[前提条件] タブ 459
[出力] ダイアログ ボックス 459
ファイルとディレクトリの [アクセス許可] ダイアログ ボックス 460
レジストリ キーの [アクセス許可] ダイアログ ボックス 463
[アイコンの選択] ダイアログ ボックス 465
[INSTALLDIR の設定] ダイアログ ボックス/[DATABASEDIR の設定] ダイアログ ボックス 465
[設定] ダイアログ ボックス 466
[MSI ログファイル] タブ 467
[システム ハードウェア要件] ダイアログ ボックス 467
[マージ モジュール検索パスのアップデート] ダイアログ ボックス 468
[Express プロジェクト名のアップグレード] ダイアログ ボックス 468
ウィザード リファレンス 469
新規 QuickPatch 作成ウィザード 469
[ようこそ] パネル 469
[プロジェクト名] パネル 470
[パッチのリリース] パネル 470
[完了] パネル 470
DirectX オブジェクト ウィザード 470
[ようこそ] パネル 471
[オブジェクトの設定] パネル 471
[概要] パネル 472
ダイナミック スキャン ウィザード 472
[ようこそ] パネル 473
[ファイルのフィルター] パネル 473
[実行可能ファイルの指定] パネル 473
[アプリケーション ファイルの指定] パネル 473
アプリケーションの起動 474

[アプリケーションが実行中です] パネル 474
[ファイル選択] パネル 474
[スキャン結果] パネル 475
[ダイナミック スキャン ウィザード完了] パネル 475
文字列テーブルのエクスポート ウィザード 475
[ようこそ] パネル 475
[ファイル名] パネル 475
[ウィザードの完了] パネル 476
REG ファイルのインポート ウィザード 476
[ようこそ] パネル 476
[レジストリ ファイルのインポート] パネル 477
[競合オプションのインポート] パネル 477
[インポート進行状況] パネル 477
文字列テーブルのインポート ウィザード 477
[ようこそ] パネル 478
[ファイル名] パネル 478
[ウィザードの完了] パネル 478
再配布可能ファイル ダウンローダー ウィザード 478
スタティック スキャン ウィザード 479
[ようこそ] パネル 479
[ファイルのフィルター] パネル 480
[スキャンの進行状況] パネル 480
[ファイル選択] パネル 480
[スキャン結果] パネル 481
[スタティック スキャン ウィザードの完了] パネル 481
システム検索ウィザード 481
[ようこそ] パネル 482
検索する対象を指定してくださいパネル 482
検索方法パネル (システム検索メソッドの定義) 482
[ファイルの詳細の指定] パネル (ファイル検索オプション) 484
検索方法パネル (フォルダー検索オプション) 484
検索方法パネル(特定のフォルダーオプション) 484
検索方法パネル (レジストリ検索オプション) 485
検索方法パネル (.ini ファイル検索 オプション) 486
この値の処理方法を指定してくださいパネル 486
Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード 486
[ようこそ] パネル 487
[ソリューション] パネル 487
Visual Studio デプロイメント プロジェクト インポート ウィザード 487
[ようこそ] パネル 488
[プロジェクトファイル] パネル 488
[オプション] パネル 488
[概要] パネル 493
Web の配布ウィザード 494
[ようこそ] パネル 494
[リンク タイプ] パネル 494

[Windows Installer エンジン オプション] パネル 494
[Windows Installer の場所] パネル 495
[詳細設定] パネル 495
[Internet Explorer 対象のデジタル署名とセキュリティ] パネル 496
[概要] パネル 497
ビュー リファレンス 499
[セットアップの編成] ビュー 499
[一般情報] ビュー 500
一般情報の設定 500
[機能] ビュー 514
機能の設定 515
[セットアップの種類] ビュー 517
[アップグレード パス] ビュー 517
アップグレード パスの設定 518
[アップデート通知] ビュー 521
[アプリケーション データの指定] ビュー 521
[ファイル] ビュー 522
インストール先フォルダー 524
[ファイルおよび機能] ビュー 527
[再配布可能ファイル] ビュー 527
[依存関係] ビュー 530
[ターゲット システムの構成] ビュー 531
[ショートカット/フォルダー] ビュー 532
ショートカットの設定 533
フォルダーの設定 539
[タイル構成] の設定 539
[レジストリ] ビュー 541
[ODBC リソース] ビュー 542
ODBC リソースの設定 542
[INI ファイルの変更] ビュー 544
.ini ファイルの設定 544
.ini ファイル キーワードの設定 545
[ファイルの拡張子] ビュー 546
ファイル拡張子設定 546
[環境変数] ビュー 547
環境変数の設定 548
≫ [IIS 構成] ビュー 550
Web サイトの設定 550
"アプリケーション" と "仮想ディレクトリ" の設定 559
[コンポーネント サービス] ビュー 565
[インストール] タブ 566
[サービス] ビュー 567
[サービス] ビューの設定 568
[セットアップ外観のカスタマイズ] ビュー 571
[ダイアログ] ビュー 571
[ビルボード] ビュー 572

ビルボード設定 *573*
Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定 *574*
[テキストとメッセージ] ビュー 577
[セットアップ要件およびアクションの定義] ビュー 577
[要件] ビュー 577
[カスタム アクション] ビュー 578
MSI DLL カスタム アクションの設定 *579*
DLL カスタム アクション設定 *581*
実行可能ファイル カスタム アクションの設定 *583*
VBScript カスタム アクションの設定 *586*
JScript カスタム アクションの設定 *588*
[セットアップのファイル] ビュー 589
[リリースの準備] ビュー 590
[リリース] ビュー 590
Express の設定 *591*
メディアの種類 *592*
[ビルド] タブ *593*
Setup.exe タブ *596*
[署名] タブ *603*
[.NET/J#] タブ *607*
[インターネット] タブ *612*
[イベント] タブ *612*
QuickPatch プロジェクト 614
[パッチ設定の定義] ビュー 614
[一般情報] ビュー *615*
製品のプロパティ 615
ビルドの設定 615
[共通] タブ 616
[識別] タブ 616
[デジタル署名] タブ 617
[詳細] タブ 618
履歴 624
カスタム アクション 625
[ファイル] ビュー *625*
新規ファイルの設定 625
変更/削除されたファイルの設定 626
[レジストリ] ビュー *628*
エラーと警告 631
ビルド エラーと警告 631
アップグレード エラーと警告 677
Windows Installer ランタイム エラー 680
Setup.exe ランタイム エラーと警告 682
Visual Studio プロジェクトのインポート エラーと警告 684
InstallShield カスタム アクション リファレンス 707
コマンドライン ツール 715
IsCmdBld.exe 715
MsiExec.exe 718

Setup.exe 718
エンドユーザー ダイアログ 723
すべてのエンドユーザー ダイアログのためのグローバル ダイアログ設定 723
[スプラッシュ ビットマップ] ダイアログ 725
[インストール - ようこそ] ダイアログ 727
[使用許諾契約書] ダイアログ 728
Readme ダイアログ 730
[ユーザー情報] ダイアログ 731
[インストール先のフォルダー] ダイアログ 734
[データベース フォルダー] ダイアログ 735
[セットアップの種類] ダイアログ 737
[カスタム セットアップ] ダイアログ 738
[インストールの準備完了] ダイアログ 740
[インストールの準備] ダイアログ 742
[セットアップの完了] ダイアログ 743
MsiRMFilesInUse ダイアログ 746

4 よく寄せられる質問 (FAQ) 749

用語集 751

索引 783

1

InstallShield 2021 – Express Edition ヘルプ ライブラリ

最新のテクノロジをサポートする速くて簡単なインストールにおいて、InstallShield は業界標準です。InstallShield を使用して、信頼できるインストールを記録的に短い時間でビルドすることができます。また、最新テクノロジがサポートされていますので、インストール関連のエラーを避けることができます。

InstallShield ヘルプ ライブラリには、InstallShield の機能性および機能についての情報が掲載されています。ヘルプ ライブラリは、次のセクションに分かれています。

テーブル 1-1 ・ ヘルプ ライブラリのセクション

セクション	説明
InstallShield 2021 エディション の新しい機能	InstallShield 2021 Express Edition の新しい機能および強化内容に関する情報です。
InstallShield Express Edition の以前のバージョンの新機能	InstallShield の以前のバージョンで加えられた変更に関する情報です。
ターゲット システムの要件	ターゲット システムの要件を一覧にします。
InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い	管理者権限を持たずに InstallShield を実行している場合に利用できない機能についてアラートします。また、管理者と非管理者コンテキストを切り替えた場合で、プロジェクトでマップされたドライブを使用した際に発生する可能性のある問題についても説明します。
32 ビットと 64 ビット システムにおけるインストールの開発およびビルドの違い	InstallShield を 32 ビット システムで使用している場合に現れる可能性がある、64 ビット システムとの相違点をハイライトします。
ヘルプの使い方	InstallShield ドキュメントに関する情報が提供されています。

テーブル 1-1・ヘルプ ライブラリのセクション (続き)

セクション	説明
スタート ガイド	InstallShield に慣れていただき、インストール プロジェクトの作成の開始、および InstallShield ユーザー インターフェイスのカスタマイズに役立つ情報が掲載されています。
インストールの作成	ユーザーフレンドリーで信頼できるインスールの作成方法、およびその作成手順がステップごと ([プログラムの追加と削除] のための情報指定からインストールのビルド、テストおよび配布まで) に説明されています。
アプリケーションのアップデート	製品アップデートのために用意されている様々なタイプのアップグレードおよびパッチの計画および実装の仕方をステップごとに見ていくことができます。また、FlexNet Connect を利用して、どのようにエンドユーザーに入手可能なアップグレードおよびパッチについて通知するかについても説明されています。
追加のインストール オプション	InstallShield で幅広く提供されているオプションの説明です (例、Windows Installer サービス をインストールに含める、アプリケーションのターゲット システム要件を指定する)。
InstallShield と外部アプリケーションの統合	InstallShield と Microsoft Visual Studio や Microsoft Visual Studio Team Foundation Server (TFS) などのサードパーティ ツールとの統合について詳細を説明します。
リファレンス	InstallShield ユーザー インターフェイスのリファレンス情報 (インストールの作成、ビルド、実行時に発生する可能性があるエラーと警告、およびサンプル エンドユーザー ダイアログ) が包括的に網羅されています。
よく寄せられる質問 (FAQ)	InstallShield およびプロジェクトの作成について頻繁に寄せられる質問に答えるヘルプ トピックを見つけることができます。

***メモ・** InstallShield ヘルプライブラリは、InstallShield とインタラクトするよう設計されているので、InstallShield 内からヘルプを開くことをお勧めします。ヘルプファイルを別のフォルダーやシステムにコピーすると、多くの機能が正常に機能しないことがあります。*

InstallShield に関してよく寄せられる質問や、ドキュメントに記載されていない新規の情報については ナレッジベースを参照してください。

InstallShield 2021 エディション の新しい機能

InstallShield Express Edition には、以下のような新しい機能が搭載されています:

- InstallShield Express Edition 2021 R2 の新しい機能
- InstallShield Express Edition 2021 R1 の新しい機能

InstallShield Express Edition 2021 R2 の新しい機能

InstallShield Express Edition には、次の強化機能が含まれています。

- ARM 固有のフォルダーにファイルをインストールできる機能

ARM 固有のフォルダーにファイルをインストールできる機能

InstallShield Express エディションを使って、ARM 固有のフォルダーを定義済みフォルダーから InstallShield プロジェクトに追加することができます。ただし、InstallShield Express エディションではテンプレート概要が Arm64 に更新されません。引き続き x86(Intel) テンプレート概要を使用して、ファイルを ARM 固有のフォルダーにコピーしてください。

InstallShield Express Edition 2021 R1 の新しい機能

InstallShield Express Edition には、以下のような新しい機能が搭載されています:

- Windows 11 のサポート

Windows 11 のサポート

InstallShield Express Edition 2021 R1 で作成された Setup は今回より、Windows 11 バージョンで実行することが可能です。

InstallShield Express Edition の以前のバージョンの新機能

このセクションでは、InstallShield Express Edition の以前のバージョンでリリースされた機能と強化内容が説明されています。:

- InstallShield 2020 Express Edition の新しい機能
- InstallShield 2019 Express Edition の新しい機能
- InstallShield 2018 Express Edition SP1 の新しい機能
- InstallShield 2018 Express Edition の新しい機能
- InstallShield 2016 SP2 Express Edition の新しい機能
- InstallShield 2016 SP1 Express Edition の新しい機能
- InstallShield 2016 Express Edition の新しい機能
- InstallShield 2015 SP1 Express Edition の新しい機能
- InstallShield 2015 Express Edition の新しい機能
- InstallShield 2014 SP1 Express Edition の新しい機能
- InstallShield 2014 Express Edition の新しい機能
- InstallShield 2013 SP1 Express Edition の新しい機能

- InstallShield 2013 Express Edition の新しい機能
- InstallShield 2012 Spring SP1 Express Edition の新しい機能
- InstallShield 2012 Spring Express Edition の新しい機能
- InstallShield 2012 SP1 Express Edition の新しい機能
- InstallShield 2012 Express Edition の新しい機能
- InstallShield 2011 Express Edition の新しい機能
- InstallShield 2010 Express Edition Expansion Pack for Visual Studio 2010 の新しい機能
- InstallShield 2010 Express Edition SP1 の新しい機能
- InstallShield 2010 Express Edition の新しい機能
- InstallShield 2009 Express Edition の新しい機能
- InstallShield 2008 Express Edition の新しい機能
- InstallShield 12 Express Edition の新しい機能

InstallShield 2020 Express Edition の新しい機能

InstallShield Express Edition には、以下のような新しい機能が搭載されています:

- InstallShield Express Edition 2020 R3 の新しい機能
- InstallShield 2020 Express Edition の新しい機能

InstallShield Express Edition 2020 R3 の新しい機能

- InstallShield Azure DevOps ビルド拡張

InstallShield Azure DevOps ビルド拡張

InstallShield 2020 R3 では、Azure DevOps Pipelines で InstallShield プロジェクトをビルドする拡張が追加されました。タスクを構成して InstallShield プロジェクトをビルドするには、InstallShield Azure DevOps ビルド拡張のサポートの技術情報の記事 (英語) を参照してください。

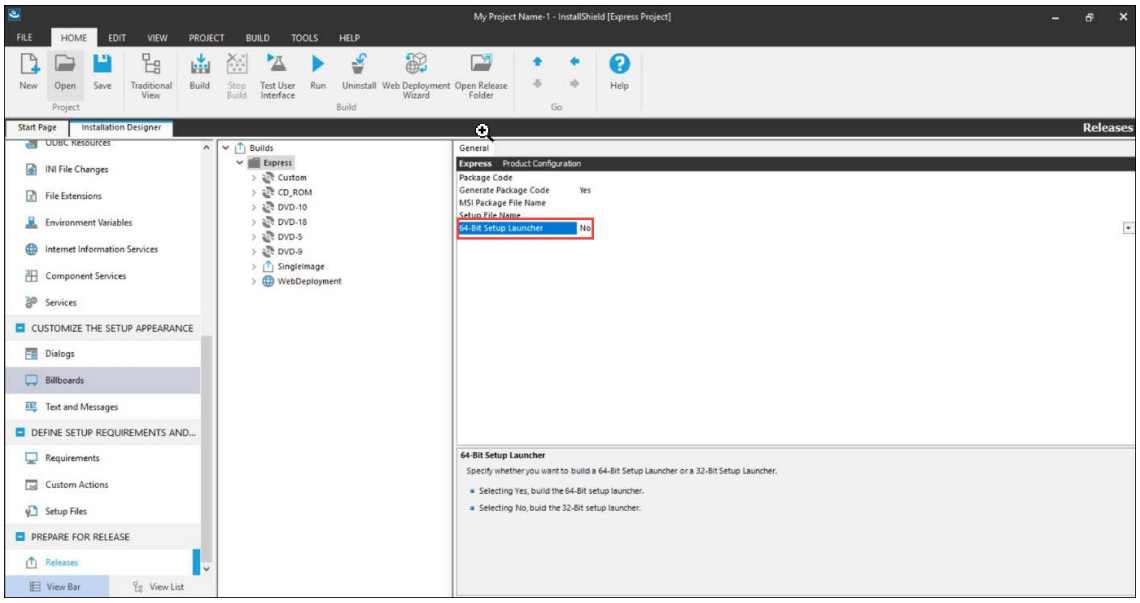
InstallShield 2020 Express Edition の新しい機能

- ピュア 64 ビット インストーラー
- AWS CloudHSM ベースのデジタル署名をサポート

ピュア 64 ビット インストーラー

InstallShield 2020 R1 Express Edition より、64 ビット ランチャーを使用するインストーラーを作成できるようになりました。

ピュア 64 ビット インストーラーを作成するには、[製品の構成] ビューに移動して "64 ビット セットアップ ランチャー" 設定で [はい] を選択します。



AWS CloudHSM ベースのデジタル署名をサポート

InstallShield を使って、今回より AWS CloudHSM ベースのデジタル署名を使ってインストーラーにデジタル署名を行うことができます。この機能を有効化するには、次のプロパティを <<InstallShield_Location>/Support/<0409¥0411>Settings.xml ファイルに追加してください。

```
<!-- Specify Platform = X86 | X64 for Digital Signing -->
<DigitalSignature Platform="X64"/>
```

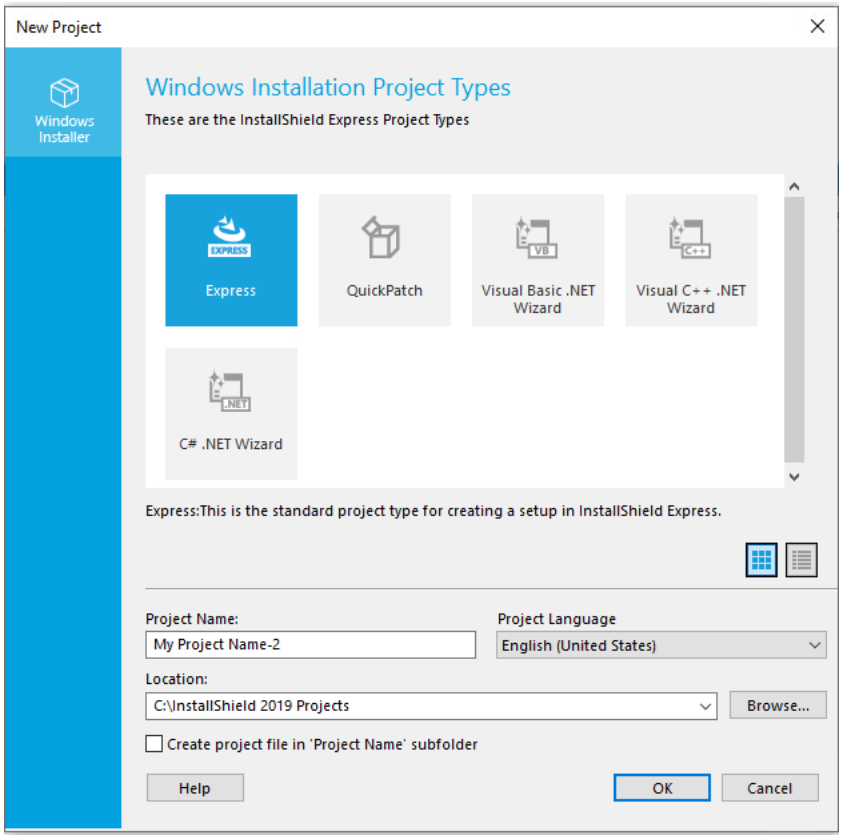
InstallShield 2019 Express Edition の新しい機能

InstallShield Express Edition には、以下のような新しい機能が搭載されています:

- 新しいプロジェクトウィザード
- 強化機能

新しいプロジェクトウィザード

新しいプロジェクト ウィザードを使って、異なる InstallShield Express プロジェクト タイプを容易に選択できます。



Express、QuickPatch、Visual Basic.NET ウィザード、Visual C++.NET ウィザード、および C#.NET ウィザードなどの異なるプロジェクト タイプを作成できます。

強化機能

InstallShield 2019 Express Edition で追加された強化機能については、次のセクションを参照してください：

- InstallShield 2019 R2 Express Edition の強化機能
- InstallShield 2019 Express Edition の強化機能

InstallShield 2019 R2 Express Edition の強化機能

InstallShield 2019 Express Edition で追加された強化機能については、次のセクションを参照してください：

連続する署名の遅延

InstallShield 2019 R2 Express Editionでは、連続するデジタル署名の間に遅延を設定することができます。この機能は、タイムスタンプ サーバーが連続する署名要求の処理に失敗した場合のみ必要になります。

Settings.xml の <DevStudio/Build> ノードの下にある <DelayBetweenSigning default="1500"/> ノードで、ミリ秒単位で指定する必要があります。

InstallShield と共にインストールされている Settings.xml ファイルを検出します。Settings.xml は、使用している InstallShield の言語バージョンに応じて、次のいずれかの場所にインストールされています:

- 英語—InstallShield Program Files Folder¥Support¥0409
- 日本語—InstallShield Program Files フォルダー¥Support¥0411

SQL 2012 Native Client 前提条件のアップデート

***プロジェクト・**この情報は、次のプロジェクトの種類に適用します :*

- *基本の MSI*
- *InstallScript*
- *InstallScript MSI*

InstallShield 2019 R2 Express Edition には、最新バージョンの 2012 Native Client 用の Microsoft SQL Server 2012 Native Client 前提条件 (x86 および x64) が含まれています。

InstallShield 2019 Express Edition の強化機能

InstallShield 2019 Express Edition で追加された新しい機能については、次のセクションを参照してください:

- 選択された証明書の詳細を表示する
- 定義済みインストール条件の追加
- デフォルト サーバーを SHA-2 サーバーに更新
- オペレーティング システム要件として Windows Server 2019 を追加

選択された証明書の詳細を表示する

InstallShield 2019 Express Edition では、証明書の全般的な情報や、セキュリティ、証明書パスなどの情報が [証明書の選択] ダイアログ ボックスの [詳細表示] オプションに一覧表示されます。

定義済みインストール条件の追加

InstallShield に新しい定義済みシステム検索が追加されました:

- Microsoft .NET Framework 4.6.2

インストールで上記が必要な場合、[システム検索] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、このシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

デフォルト サーバーを SHA-2 サーバーに更新

InstallShield 2019 Express Editionで:

- SHA-1 を使ってパッケージに署名を行うと、パッケージは次を使ってタイムスタンプを取得します:
  - <DigitalSignature Timestamp="http://timestamp.verisign.com/scripts/timstamp.dll"/>
- SHA-256 を使ってパッケージに署名を行うと、パッケージは次を使ってタイムスタンプを取得します:
  - <DigitalSignature TimestampRFC3161="http://sha256timestamp.ws.symantec.com/sha256/timestamp"/>

オペレーティング システム要件として Windows Server 2019 を追加

InstallShield 2019 では、オペレーティング システム要件セクションに Windows Server 2019 のオプションを追加できるようになりました。オプションを選択/選択解除して、プロジェクトに適切なインストール条件を設定することができます。

追加された前提条件

InstallShield 2019 には、以下の前提条件が追加されています:

Microsoft .Net Core 2.1 ランタイム 前提条件

InstallShield の [再配布可能ファイル] ビューには、今回より Microsoft .Net Core 2.1 ランタイム 前提条件が追加されています。

InstallShield 2018 Express Edition SP1 の新しい機能

InstallShield 2018 SP1 Express Edition で追加された新しい機能についての詳細は、次のセクションを参照してください:

SHA1 & SHA256 を使用するデュアル署名のサポート

以前のリリースでは、署名ダイジェスト ハッシュ アルゴリズムの選択肢は次に限られていました:

- 証明書ハッシュ
- SHA-1
- SHA-256

InstallShield 2018 SP1 では、これらの署名ダイジェスト ハッシュ アルゴリズム以外にもデュアル署名 - (SHA-1 および SHA-256) ダイジェストを選択できます。

メモ・ *両方のダイジェスト (SHA1 およびSHA256) の使用は、.msi のデジタル署名でサポートされていません。*

InstallShield 2018 Express Edition の新しい機能

InstallShield 2018 Express Edition で追加された新しい機能についての詳細は、次のセクションを参照してください:

- FlexNet Code Aware を使ったオープン ソース リスク評価の実行
- Web アプリケーションのフォーム認証を設定する

FlexNet Code Aware を使ったオープン ソース リスク評価の実行

InstallShield には、今回より FlexNet Code Aware との完全な統合が含まれています。FlexNet Code Aware は、製品のセキュリティおよび知的財産 (IP) コンプライアンス リスクを迅速にスキャンする、自動オープン ソース リスク評価およびパッケージ検出ソリューションです。

- サポート対象ファイル形式
- FlexNet Code Aware を実行する
- FlexNet Code Aware レポートの読み方

サポート対象ファイル形式

FlexNet Code Aware では次のファイル分析がサポートされています:

- Java パッケージ
- Node パッケージ
- Nuget パッケージ
- RPM パッケージ
- Ruby パッケージ
- EXE & DLL ファイル

セキュリティ脆弱性は、National Vulnerability Database (NVD) と照合されます。

FlexNet Code Aware を実行する

FlexNet Code Aware は、InstallShield の一部であり、アクティベーションの際にアクティベーション ID は不要です。

InstallShield 内部から FlexNet Code Aware を実行するには、InstallShield **[プロジェクト]** メニューから **[FlexNet Code Aware を使ってプロジェクトをスキャン]** をクリックするか、標準ツールバーにある [FlexNet Code Aware] アイコンをクリックしてください。

***メモ・**この FlexNet Code Aware メニュー オプションは、InstallShield プロジェクトが現在開いていない場合は無効です。*

FlexNet Code Aware がプロジェクトのスキャンを完了したとき [Results Summary (結果の概要)] ビューが開いて、キャン済みファイルの数、オープン ソース パッケージ、ならびに検出された脆弱性の数が表示されます。

[View Report (レポートを表示)] ボタンをクリックすると、完全なレポートが表示されます。

FlexNet Code Aware レポートの読み方

[Results Summary (結果の概要)] 画面で **[View Report (レポートを表示)]** をクリックすると、**[Initial Summary (初期の概要)]** と **[Package Inventory (パッケージ インベントリ)]** で構成される完全な FlexNet Code Aware レポートが開きます。

- Initial Summary (初期の概要)] ビュー—**[Initial Summary (初期の概要)]** ビューにはスキャン概要ならびに、運用時のリスク、セキュリティ脆弱性の影響、およびライセンス リスクについての評価が表示されます。FlexNet Code Aware [Initial Summary (初期の概要)] ビューには、次の情報が表示されます:
  - [Scan Summary (スキャン概要)] – このセクションには、ファイル タイプの内訳、分析済みファイルの割合、および検出数を含む、スキャン済みのコードベースについての詳細が表示されます。
  - [Operational Risk (運用リスク)] – このセクションには、パッケージと知的財産 (IP) との問題、およびパッケージとセキュリティ脆弱性の組み合わせに基づいた合成リスク評価が表示されます。
  - [Security Vulnerability Exposure (セキュリティ脆弱性の影響)] および[License Exposure (ライセンスのリスク)]—これらのセクションには、識別された問題の種類とカテゴリーの内訳が表示されます。
- Package Inventory (パッケージ インベントリ)] ビュー—**[Package Inventory (パッケージ インベントリ)]** ビューは **[Scan Summary (スキャン概要)]** セクションで **[View full package inventory (フル パッケージ インベントリの表示)]** をクリックすると表示され、検出されたオープンソースおよびサードパーティ パッケージお

よび関連ライセンス、セキュリティ脆弱性、依存関係、ならびに検出された著作権ステートメントの完全リストを表示します。

[Package Inventory (パッケージ インベントリ)] ビューには、ターゲットのクエリを実行して、一覧を様々なパッケージの種類に絞り込むためのフィルターが用意されています。

追加パッケージの詳細を表示するには、確認したいパッケージの [Vulnerabilities (脆弱性)] 列に一覧表示されている脆弱性カウントをクリックしてください。[Vulnerabilities Detail (脆弱性の詳細)] ページが開き ([Package Inventory (パッケージ インベントリ)] ビューの一部と重なる)、選択されたパッケージの詳細情報が表示されます。

Web アプリケーションのフォーム認証を設定する

InstallShield 2018 Express Edition に、Web アプリケーションでフォーム認証を設定するための新しいオプションが追加されました。この新しいオプション [フォーム認証] は、Web サイトの [IIS 構成] ビューで [認証済みアクセス] セクションの下に表示されます。

[フォーム認証] オプションを [はい] に設定してフォーム認証を有効化します。ASP.NET フォーム ベースの認証は、多くの要求を受け取るパブリック Web サーバー上にあるサイトまたはアプリケーションに適しています。この認証モードでは、オペレーティング システムが提供している認証方法に頼ることなく、アプリケーション レベルでのクライアント登録および認証を管理することができます。

重要 *・フォーム認証は、Web サーバーにユーザー名とパスワードをテキスト形式で送信します。ホームページ以外のアプリケーション内のすべてのページ、およびログオン ページには、Secure Sockets Layer (SSL) 暗号化を使用してください。*

強化機能

InstallShield 2018 Express Edition には、次のような新しい強化機能が含まれています。

- 製品構成のセットアップ ファイル名にプロパティの値を含める
- 概要情報ストリーム コメントおよびパッケージ ファイル名を設定するための新しい MSBuild パラメーター
- 追加された前提条件

製品構成のセットアップ ファイル名にプロパティの値を含める

InstallShield 2018 Express Edition では今回より、製品リリースのセットアップ名およびパッケージ ファイル名に Property テーブルからのプロパティの値を含めることができます。

たとえば、[リリース > Express] ビューの [全般] タブにある **"セットアップ ファイル名"** または **"MSI パッケージ ファイル名"** フィールドに次の任意のプロパティを入力することができます:

setup[ProductVersion]
setup[CustomVersion]
setup[ProductCode]
setup[ProductCode][ProductVersion]

たとえば、**"セットアップ ファイル名"** フィールドに setup[ProductVersion] と入力した場合、セットアップ名が setup14.10.1234.exe となります。

概要情報ストリーム コメントおよびパッケージ ファイル名を設定するための新しい MSBuild パラメーター

InstallShield 2018 Express Editionで今回追加された MSBuild パラメーターを使って、インストーラーにコメントの追加を設定、およびインストーラーのパッケージ ファイル名の設定を行うことができます。

- 概要情報ストリーム コメントを設定する新しいパラメーター
- パッケージ ファイル名を設定する新しいパラメーター

概要情報ストリーム コメントを設定する新しいパラメーター

[**一般情報**] ビューの **"概要情報ストリーム コメント"** フィールドでインストーラーにコメントを追加することができます。

InstallShield 2018 express Edition ではまた、ビルド時にコメントを入力できるオプションも追加されています。MSBuild.exe タスクに SummaryInfoComments という名前の新しいパラメーターが追加されました。これを使って、次の例のようにビルド番号を含めるなど、ビルド時に [**概要情報ストリーム コメント**] を設定することができます。

MSBuild.exe c:¥installers¥Setup.sln /Property:SummaryInfoComments="ここにコメントを入力"

SummaryInfoComments プロパティを使って追加されたコメントは、ビルド済みのインストーラーの [**プロパティ**] ダイアログ ボックスで参照することができます。

パッケージ ファイル名を設定する新しいパラメーター

[**リリース > Express**] ビューで [**全般**] タブにある **"MSI パッケージ ファイル名"** フィールドで、インストーラーのパッケージ ファイル名を指定できます。

InstallShield 2018 Express Edition ではまた、パッケージ ファイル名を設定できるオプションもあります。MSBuild.exe タスクに MSIPackageFileName という名前の新しいパラメーターが追加されました。これを使って、次の例のようにビルド時にインストーラーのパッケージ ファイル名を設定することができます。

MSBuild.exe c:¥installers¥Setup.isproj /Property:MSIPackageFileName="MySetup"

MSIPackageFileName パラメーターの値を入力するとき、InstallShield が .msi ファイルに使用する (ピリオドまたはファイル拡張子を含まない) ファイル名を入力する必要があります。

追加された前提条件

InstallShield 2018 Express Edition には、以下のような追加前提条件が含まれています:

- Visual C++ 2017 x86 および x64 前提条件
- Microsoft SQL Server 2014 SP1 および SP2 前提条件
- Microsoft .NET Framework 4.7 前提条件

Visual C++ 2017 x86 および x64 前提条件

Microsoft Visual Studio 2017 がリリースされたため、InstallShield は今回より、Visual C++ 2017 x86 および x64 の前提条件を含みます。

Microsoft SQL Server 2014 SP1 および SP2 前提条件

Microsoft SQL Server 2014 には 2 つのサービス パックがリリースされているため、InstallShield は今回より、Microsoft SQL Server 2014 SP1 および SP2 両方の前提条件を含みます。

Microsoft .NET Framework 4.7 前提条件

InstallShield 2018 は今回より、Microsoft .NET Framework 4.7 の前提条件を含みます。

InstallShield 2018 R2 では今回より、.net 4.7 の定義済みインストール条件を追加することができます:

- 4.7
- 4.7.1
- 4.7.2

InstallShield 2016 SP2 Express Edition の新しい機能

新しい機能

FlexNet Code Aware との統合

InstallShield には、今回より FlexNet Code Aware との統合が含まれています。FlexNet Code Aware は、製品のセキュリティおよび知的財産 (IP) コンプライアンス リスクを迅速にスキャンする、自動オープン ソース リスク評価およびパッケージ検出ソリューションです。

FlexNet Code Aware の現在のリリースでは、次のファイル分析がサポートされています:

- Java パッケージ
- Node パッケージ
- Nuget パッケージ
- RPM パッケージ
- Ruby パッケージ
- EXE & DLL ファイル

セキュリティ脆弱性は、National Vulnerability Database (NVD) と照合されます。

FlexNet Code Aware を実行する

FlexNet Code には InstallShield とは別のライセンスが必要です。トライアル/評価版もあります。詳細については、Revenera Web サイトの [FlexNet Code Aware 製品] ページ を参照してください。

InstallShield 内部から FlexNet Code Aware を実行するには、InstallShield の **[プロジェクト]** メニューから **[FlexNet Code Aware を使ってプロジェクトをスキャン]** をクリックします。このメニュー オプションは、InstallShield プロジェクトが現在開いていない場合は無効です。FlexNet Code Aware アイコンも、InstallShield 標準ツールバーに表示されます。

FlexNet Code Aware がプロジェクトのスキャンを完了したとき概要にはキャン済みファイルの数、オープン ソース パッケージ、ならびに検出された脆弱性の数が表示されます。FlexNet Code Aware のフル ライセンス版では、**[View report (レポートの表示)]** ボタンが使用できます。このレポートで提供される詳細については、「FlexNet Code Aware レポートの読み方」を参照してください。

FlexNet Code Aware レポートの読み方

***メモ・** FlexNet Code Aware レポートは、トライアル/評価版モードでは使用できません。FlexNet Code Aware のフル ライセンス版が必要です。*

FlexNet Code Aware レポートを参照するためには、FlexNet Code Aware がプロジェクトをスキャンした後に表示される概要ダイアログで **[View repoert (レポートの表示)]** をクリックします。

FlexNet Code Aware レポートは、いくつかのセクションから構成されています:

- 最初の [Summary (概要)] ビューには、**Scan Summary (スキャン概要)**、**Operational Risk (運用リスク)評価**、**Security Vulnerability Exposure (セキュリティ脆弱性の公開)**、および **License Exposure (ライセンスの公開)** が提示されます。
  - **[Scan Summary (スキャン概要)]** セクションは、ファイル タイプの内訳、分析済みファイルの割合、および検出数を含む、スキャン済みのコードベースについての詳細が表示されます。
  - **[Operational Risk (運用リスク)]** セクションには、パッケージと知的財産 (IP) との問題、およびパッケージとセキュリティ脆弱性の組み合わせに基づいた合成リスク評価が表示されます。
  - **[Security Vulnerability Exposure (セキュリティ脆弱性の公開)]** および **[License Exposure (ライセンスの公開)]** セクションには、識別された問題の種類とカテゴリーの内訳が表示されます。
- [Package Inventory (パッケージ インベントリ)] ビューは **[Scan Summary (スキャン概要)]** セクションで **[view full package inventory (フル パッケージ インベントリの表示)]** をクリックすると表示され、検出されたオープンソースおよびサードパーティ パッケージおよび関連ライセンス、セキュリティ脆弱性、依存関係、ならびに検出された著作権ステートメントの完全リストを表示します。

  [Package Inventory (パッケージ インベントリ)] ビューには、ターゲットのクエリを実行して、一覧を様々なパッケージの種類に絞り込むためのフィルターが用意されています。

パッケージの詳細を確認する

[Package Inventory (パッケージ インベントリ)] レポート ページの [Vulnerabilities (脆弱性)] 列にある脆弱性カウントをクリックして、各パッケージの詳細を確認します: パッケージ インベントリ レポートの一部が含まれた **[Vulnerabilities detail (脆弱性の詳細)]** が表示されます:

InstallShield 2016 SP1 Express Edition の新しい機能

新しい機能

Microsoft Visual Studio 2017 のサポート

InstallShield 2016 Express Edition には、Visual Studio 2017.のサポートが含まれています。このバージョンの Visual Studio 内部から InstallShield 2016 Express Edition プロジェクトを作成できます。

InstallShield 2016 Express Edition の新しい機能

新しい機能

InstallShield 2021 Express Editionには、以下のような新しい機能が搭載されています。

- Windows オペレーティング システムの最新版リリースをサポート
- タイルの構成
- 新しい Microsoft Visual C++ 2015、.NET Framework 4.6 その他用の InstallShield 前提条件
- Adobe Reader、Microsoft Office、および .NET Framework の前提条件システム検索

Windows オペレーティング システムの最新版リリースをサポート

InstallShield 2016 は、Windows オペレーティング システムの最新版リリースをサポートします。

- Windows 10 Anniversary Update
- Windows Server 2016

InstallShield をこれらのオペレーティング システムにインストールできるだけでなく、これらのオペレーティング システムをターゲットにするインストールを作成することができます。

タイルの構成

Windows 8 からアプリケーション タイルのグリッドをスタート画面に表示できるようになりました。これは、今までのショートカットの一覧に取って代わるもので、ショートカットの代わりにタイルを配置します。InstallShield は、スタート画面上のデスクトップ アプリのタイルの外観をカスタマイズすることができます。次のタイル構成設定が使用できます:

- アプリケーション名を中サイズ (150x150) のタイルに含めるとき、明色または暗色のテキストを切り替える
- タイル背景色を選択
- カスタム タイル イメージ (小: 70x70、中:150x150) の使用オプション
- アプリケーション名を中サイズ タイルに表示または非表示を選択

[タイルの構成] ノードが、メインの **[ショートカット/フォルダー]** ビューに表示されます。適用するすべてのタイル構成が一覧表示されます。

詳しくは、次のトピックを参照してください:

- スタート画面上のデスクトップ アプリのタイルの外観を構成する
- [タイル構成] の設定

新しい Microsoft Visual C++ 2015、.NET Framework 4.6 その他用の InstallShield 前提条件

InstallShieldには、Express プロジェクトに追加することができる以下の新しい InstallShield 前提条件が含まれています:

- Microsoft .NET Framework 4.6.2 (完全版)
- Microsoft .NET Framework 4.6.1 (完全版)
- Microsoft .NET Framework 4.6.1 (Web インストーラー)
- Microsoft ReportViewer 2015
- Microsoft SQL Server 2014 Express System CLR Types (x86)
- Microsoft SQL Server 2016 Express RTM (x64)
- Microsoft SQL Server 2016 Express RTM LocalDB (x64)

- Microsoft Visual C++ 2015 Update 3 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2015 Update 3 再配布可能パッケージ (x64)
- Windows Management Framework 4.0 for Windows 7 SP1 および Server 2008 R2 SP1 (x64)
- Windows Management Framework 4.0 for Windows Server 2012 (x64)
- Windows Management Framework 5.0 for Windows 7 SP1 (x86)
- Windows Management Framework 5.0 for Windows 7 SP1 および Server 2008 R2 SP1 (x64)
- Windows Management Framework 5.0 for Windows 8.1 (x86)
- Windows Management Framework 5.0 for Windows 8.1 および Server 2012 R2 (x64)
- Windows Management Framework 5.0 for Windows Server 2012 (x64)

これらの前提条件は、サポートされているターゲット システムに適切なテクノロジをインストールします。

メモ・** .NET Framework の* **Web** *前提条件には、インターネット接続が必要です。この前提条件は、必要に応じて、必須の再配布可能ファイルをダウンロードします。.NET Framework の完全***な前提条件は、インターネットへの接続が不要なスタンドアロン インストールです。*

Adobe Reader、Microsoft Office、および .NET Framework の前提条件システム検索

InstallShield に新しい定義済みシステム検索が追加されました:

- Adobe Reader 11
- Adobe Reader DC
- Microsoft Office 2013
- Microsoft Office 2016
- Microsoft .NET Framework 4.5.1
- Microsoft .NET Framework 4.5.2
- Microsoft .NET Framework 4.6
- Microsoft .NET Framework 4.6.1
- Microsoft .NET Framework 4.6.2

インストールでこれらの 1 つ以上が必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

強化機能

InstallShield 2016 には、次の新しい強化内容が追加されています:

- 追加のビューで、機能ごとに項目をフィルターできる機能
- デジタル署名の強化機能

追加のビューで、機能ごとに項目をフィルターできる機能

次のビューには今回より、プロジェクトに含まれる任意の機能ごとにビュー リストをフィルターすることができる [ビュー フィルター] が用意されています。

- **[環境変数] ビュー**—このビューの上部にある [ビュー フィルター] リストを使って、プロジェクトに含まれる特定機能に関連付けられた環境変数を表示/非表示にすることができます。[ビューリスト] から機能を選択して、その機能のみを後に続くイベント (たとえば、環境変数の作成、変更、または削除) に関連付けることができます。最後に、プロジェクトに含まれるすべての環境変数を表示するには、[ビュー フィルター] リストで [すべてのアプリケーション データ] オプションを選択します。詳細については、「[環境変数] ビュー」を参照してください。
- **[INI ファイルの変更] ビュー**—このビューの上部にある [ビュー フィルター] リストを使って、プロジェクトに含まれる特定機能に関連付けられた初期化 (.ini) ファイルを表示/非表示にすることができます。[ビューリスト] から機能を選択して、その機能のみを後に続くイベント (たとえば、.ini ファイルの作成、インポート、変更、または削除」) に関連付けることができます。結果となる変更は、機能がインストールされるときにターゲット システム上で実行時に行われます。最後に、プロジェクトに含まれるすべての .ini ファイルを表示するには、[ビュー フィルター] リストで [すべてのアプリケーション データ] オプションを選択します。詳細については、「[INI ファイルの変更] ビュー」を参照してください。

デジタル署名の更新

InstallShield 2015 より、インストールおよびファイルをビルド時に署名する際、SHA-256 ハッシュ アルゴリズムを使ったデジタル証明書を使用できるサポートが追加されました。

InstallShield 2016 では、Windows Installer および InstallScript プロジェクトの SHA-256 デジタル証明書サポートが次のように強化されています:

- **[証明書の選択]** ダイアログ ボックスの新しい [署名ダイジェスト] ドロップダウンを使って、ダイジェストの種類を指定できる機能
- 今回より、RFC3161 タイムスタンプがサポートされていて、settings.xml で次の記述を含んで指定できます:
  - .msi、.exe、および .dll ファイルでは、**DigitalSignature/@Timestamp** は **Authenticode** または **RFC3161 サーバー**
  - UWP アプリ パッケージファイルに使用される **DigitalSignature/@TimestampRFC3161** ファイルは **RFC3161 サーバー** でなくてはなりません
- 証明書ストアにある類似した名前の証明書も処理します

InstallShield 2019 Express Edition では、タイムスタンプ サーバーが SHA-2 サーバーに設定されています:

- <DigitalSignature Timestamp="http://sha256timestamp.ws.symantec.com/sha256/timestamp"/>

***重要・**2016 年 1 月以降に作成またはタイムスタンプが付けられたすべての新しい署名は、SHA-256 に基づく必要があります。SHA-1 証明書を使って署名されているすべてのファイルを継続してサポートするためには、2016 年 1 月以前の日時を使ったタイムスタンプを含める必要があります。これらのファイルは、すべての現在のバージョンの Windowsですべての SHA-1 サポートが停止される 2020 年 1 月 14 日まで、MOTW (Mark of the web) システムを使って引き続き使用することができます。*

InstallShield 2015 SP1 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

Windows 10 のサポート

InstallShield は、Windows 10 をサポートします。

Microsoft Visual Studio 2015 のサポート

InstallShield は、Visual Studio 2015 をサポートします。このバージョンの Visual Studio 内部から InstallShield プロジェクトを作成できます。

Microsoft Visual C++ 2015 および .NET Framework 4.6 用の新しい InstallShield 前提条件

InstallShield には、作成中のプロジェクトに追加することができる以下の新しい InstallShield 前提条件が含まれています:

- Microsoft Visual C++ 2015 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2015 再配布可能パッケージ (x64)
- Microsoft .NET Framework 4.6 Full
- Microsoft .NET Framework 4.6 Web

これらの前提条件は、サポートされているターゲット システムに適切なテクノロジをインストールします。

InstallShield 2015 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

Windows 10 ベース システムのサポート

InstallShield は、Windows 10 をサポートします。

Windows 10 をターゲットにする

Windows 10 が搭載されているシステム上では、Windows Installer プロパティ VersionNT および VersionNT64 が 603 を示します。これは、もともと Windows 8.1 のバージョン番号として使用されたものです。従って、Windows 10 をターゲットに特定して .msi パッケージの条件を作成することは不可能です。

Windows Installer 5.0 および Windows 7 より、.msi パッケージの DLL アクションには shim が適用されるため、オペレーティング システム バージョンの取得が阻止されます。API の **GetVersion**、**GetVersionEx**、および **RtlGetVersion** は、もともと Windows Vista のバージョン番号である 6.0.6000 を返します。従って、DLL カスタム アクションから実際の Windows のバージョン番号を取得することはできません。

前述の Windows Installer 動作のため、.msi パッケージが実行中のバージョンを検出することは容易ではありません。ターゲット システムの OS 要件を指定できる領域、たとえばプロジェクト アシスタントの [インストール要件] ページ、または [要件] ビューでは、新しい実行時の動作を反映するように、[Windows 8.1] オプションの名前が **「Windows 8.1 または Windows 10」** に変更されました。

Windows 10 にインストール可能な InstallShield 前提条件は、必要に応じて、これらのシステムにインストールされるように更新されています。以前これらのシステムでは、前提条件がデフォルトで実行されない場合がありました。

Microsoft Visual Studio 2015 のサポート

InstallShield は、Visual Studio 2015 をサポートします。このバージョンの Visual Studio 内部から InstallShield プロジェクトを作成できます。

デジタル署名の強化機能

InstallShield には、ビルド時にインストールおよびファイルにデジタル署名を行うための、いくつかの強化機能が含まれています。

SHA-256 デジタル証明書のサポート

InstallShield では、インストールおよびファイルをビルド時に署名する際、HA-256 ハッシュ アルゴリズムを使ったデジタル証明書を使用できます。

SHA-1 はセキュリティの脆弱性があるため、SHA-256 の使用が推奨されます。Microsoft は、Windows では 2016 年 1 月以降に SHA-1 証明書を使って署名およびタイムスタンプが追加されているアイテムを信頼しないことを発表しました。さらに、証明書を発行する組織である証明機関では、SHA-1 証明書が段階的に廃止されます。したがって、InstallShield プロジェクトに含まれる任意の SHA-1 証明書は、SHA-256 証明書と差し替えることが推奨されます。最新情報および特定の詳細については、証明機関にお問い合わせください。

InstallShield でリリースに署名するための SHA-1 証明書を SHA-256 証明書に置き換えるには、[リリース] ビューの [署名] タブを使って、現在の証明書への参照を SHA-256 証明書と置き換えます。

プロジェクトで、SHA-256 証明書を使った署名が構成されている場合、InstallShield はビルド時に署名を行うファイルの署名に SHA-256 ハッシュを使用します。プロジェクトで SHA-1 証明書を使った署名が構成されたままになっている場合、InstallShield は SHA-1 ハッシュを使用します。また、SHA-1 証明書を使用すると、今回より SHA-1 の使用についてアラートするビルド警告 -7346 が発生します。

InstallShield の以前のバージョンでは、SHA-1 または SHA-256 証明書のいずれかを使って署名が行なわれる際、ファイルの署名に SHA-1 ハッシュが使用されました。

詳細については、「デジタル署名とセキュリティ」を参照してください。

証明書を参照するための証明書ストアを使用できる機能

ファイルおよびインストールの署名に使用するデジタル署名情報を指定するとき、InstallShield では今回より、使用する証明書を含む証明書ストアを参照することができます。このサポートは、マシン上の .pfx 証明書ファイルを指定する方法の代替として提供されています。

証明書ストアまたは .pfx 証明書を使うかどうかを指定するには、[リリース] ビューの [署名] タブにある "デジタル証明書ファイル" 設定を使用します。この設定で省略記号ボタン (...) をクリックすると、新しい [証明書の選択] ダイアログ ボックスが開きます。このダイアログ ボックスを使って、ストア名 (Personal、Trusted Root Certification Authorities、Enterprise Trust、Intermediate Certification Authorities)、ストアの場所 (ユーザー、マシン)、および特定の証明書を識別するためのサブジェクトといった情報を指定することができます。別の方法として、このダイアログ ボックスで使用する .pfx ファイルの名前とパスを指定することもできます。

ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。

証明書ストアを、QuickPatch の署名に使用することもできます。QuickPatch パッケージで証明書ストアまたは .pfx 証明書情報を指定するには、QuickPatch プロジェクトで [一般情報] ビューの [ビルドの設定] 領域を使用します。この領域には、新しいサポートを含む [デジタル署名] タブがあります。

詳しくは、次を参照してください:

- デジタル署名とセキュリティ
- [証明書の選択] ダイアログ ボックス
- [署名] タブ ([リリース] ビュー内のリリース)
- [デジタル署名] タブ (QuickPatch プロジェクト)

InstallShield では、今回より .spc および .pvk ファイルを使った署名がサポートされていません。これらのファイルを .pfx ファイルに変換する方法については、「デジタル署名とセキュリティ」を参照してください。

UAC ダイアログ ボックスにプログラム名を指定できる機能

[リリース] ビューの [署名] タブには、"署名の説明" 設定がります。この設定を使って、ビルド時に InstallShield が署名を行なう Setup.exe ファイル、.msi ファイル、その他のインストール ファイルの UAC ダイアログ ボックスで "プログラム名:" ラベルの右側に表示するテキストを指定します。UAC ダイアログ ボックスは、エンド ユーザーが署名されたファイルを起動したとき、昇格された権限が必要な場合に開きます。

"署名の説明" 設定を空白のままに残すと、InstallShield は UAC ダイアログ ボックスのテキストとして、ファイル名を拡張子なしで使用します。

詳細については、「[署名] タブ」を参照してください。

64 ビット開発システム上で、ソースマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示できる機能

InstallShield を 64 ビット開発システム上で使用する場合、InstallShield が表示する [レジストリ] ビューは、使用中のマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示します:

- HKEY_LOCAL_MACHINE¥Software
- HKEY_LOCAL_MACHINE¥Software¥Wow6432Node

このサポートによって、このビューのインストール先ペインの適切な領域ににソース領域からのエントリをドラッグ アンド ドロップできるようになるため、64 ビット マシン上でのインストールの開発が容易になります。

以前、64 ビット開発システム上で InstallShield を使用した場合、InstallShield の [レジストリ] ビューのソース ペインにはレジストリの HKLM¥Software 部分にある 64 ビットデータが表示されませんでした。さらに、ソース ペインには、マシンの HKLM¥Software¥Wow6432Node 領域の 32 ビットデータが HKLM¥Software 領域に表示されました。

インストールで 32 ビット領域へのリダイレクトを行わなずに 64 ビット ターゲット システム上の 64 ビット領域にレジストリ データをインストールする場合、レジストリ データを [レジストリ] ビューのインストール先ペインにある HKEY_LOCAL_MACHINE¥SOFTWARE (64 ビット) ノードに配置しなくてはなりません。[レジストリ] ビューのソースペインから 64 ビット データを、ビュー内の 64 ビット以外のインストール先ペインにドラッグするだけでは、そのコンポーネントが 64 ビットであるとマークされません。

詳細については、次を参照してください。

- 32 ビットと 64 ビット システムにおけるインストールの開発およびビルドの違い
- レジストリ エントリをドラッグアンドドロップしてレジストリ キーを作成する
- [レジストリ] ビュー

新しい Microsoft Visual C++ 2015、.NET Framework 4.6 その他用の InstallShield 前提条件

InstallShield には、プロジェクトに追加することができる新しい InstallShield 前提条件が含まれています:

- Microsoft Visual C++ 2015 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2015 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2013 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2013 再配布可能パッケージ (x64)
- Microsoft .NET Framework 4.6 Full
- Microsoft .NET Framework 4.6 Web
- Microsoft .NET Framework 4.5.2 (完全版)
- Microsoft .NET Framework 4.5.2 (Web インストーラー)
- Microsoft SQL Server 2012 Express SP2 (x86)
- Microsoft SQL Server 2012 Express SP2 (x86 & x64Wow)
- Microsoft SQL Server 2012 Express SP2 (x64)
- Microsoft SQL Server 2012 Express SP2 LocalDB (x86)
- Microsoft SQL Server 2012 Express SP2 LocalDB (x64)
- Microsoft SQL Server 2012 Express SP2 Management Objects (x86)
- Microsoft SQL Server 2012 Express SP2 Management Objects (x64)
- Microsoft SQL Server 2012 Express SP2 System CLR Types (x86)
- Microsoft SQL Server 2012 Express SP2 System CLR Types (x64)
- Windows 7 (x86) 用 Internet Explorer 11.0
- Windows 7 と Windows Server 2008 R2 (x64) 用の Internet Explorer 11.0
- Microsoft ReportViewer 2012

これらの前提条件は、サポートされているターゲット システムに適切なテクノロジをインストールします。

Microsoft SQL Server 2012 Express SP2 前提条件は、Microsoft SQL Server 2012 Express SP1 前提条件にとって代わります。

Internet Explorer 10 および 11 用の新しい定義済みシステム検索

InstallShield には、ターゲット システム上で Internet Explorer 10 または Internet Explorer 11を確認する新しい定義済みのシステム検索が含まれています。インストールまたはプロジェクトでこれらのバージョンのどちらかが必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらの

システム検索の 1 つをプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

強化機能

[ファイルとフォルダー] ビューのパフォーマンス強化

サイズが大きいプロジェクトの [ファイル] ビューををより短時間でロードできるように、InstallShield が強化されています。

InstallShield 2014 SP1 Express Edition の新しい機能

InstallShield には、以下のような強化機能が搭載されています。

ポルトガル実行時文字列の更新

ポルトガル語 (ブラジル) および ポルトガル語 (ポルトガル) のデフォルトの実行時文字列は、1990 年のポルトガル語新正書法に基づいて更新されています。この新正書法は、2009 年以来 6 年間の移行期間を経て、2014 年の終りに完全実施されます。

InstallShield 2014 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

新しい .NET Framework 4.5.1 用 InstallShield 前提条件

InstallShield には、プロジェクトに追加することができる新しい InstallShield 前提条件が含まれています:

- Microsoft .NET Framework 4.5.1 (完全版)
- Microsoft .NET Framework 4.5.1 (Web インストーラー)

これらの前提条件は、サポートされているターゲット システムに、.NET Framework 4.5.1 をインストールします。

Microsoft SQL Server 2014 の前提条件

InstallShield には、プロジェクトに追加することができる新しい SQL Server 2014 関連の InstallShield 前提条件が含まれています。

- Microsoft SQL Server 2014 Express RTM (x64)
- Microsoft SQL Server 2014 Express RTM (x86 & x64Wow)
- Microsoft SQL Server 2014 Express RTM (x86)
- Microsoft SQL Server 2014 Express RTM LocalDB (x64)
- Microsoft SQL Server 2014 Express RTM LocalDB (x86)

これらの InstallShield 前提条件は、サポートされているターゲット システムにテクノロジーがインストールされます。

Microsoft Visual C++ 2012 Update 4 用の新しい InstallShield 前提条件

InstallShield には、プロジェクトに追加することができる新しい InstallShield 前提条件が含まれています:

- Microsoft Visual C++ 2012 Update 4 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2012 Update 4 再配布可能パッケージ (x64)

これらの前提条件は、サポートされているターゲット システムに様々なテクノロジをインストールします。

サブフォルダーにサポート ファイルを含める機能

InstallShield では、今回より、サポートファイルにユーザー設定のフォルダー構造を指定できます。[セットアップ ファイル] ビューのノードの 1 つの下にサブフォルダーを追加するには、そのノードを右クリックしてから [新しいフォルダー] をクリックします。サブフォルダーが追加され、必要に応じてその名前を変更することができます。ネストされたフォルダー構造を追加することもできます。ファイルをサブフォルダーに追加するには、右側のペインを選択してから、[ファイルの挿入] をクリックします。実行時、インストールがサポート フォルダーとファイルをターゲット システム上の一時ディレクトリである SUPPORTDIR にコピーして、製品のインストール処理中に使用できるようにします。サポート フォルダーとファイルは、インストールが完了すると削除されます。

詳細については、「セットアップ ファイルの追加」を参照してください。

ファイルとフォルダーの削除に関するサポート

InstallShield では、実行時にターゲット システムから削除するファイルとフォルダーを簡単に指定することができるビルトイン サポートが提供されています。このファイルとフォルダーの削除機能は、アプリケーションによって作成されるファイルの削除など、インストールが追跡を行わない処理に使用すると便利です。

ファイルまたはフォルダーの削除は、次のイベントの 1 つにスケジュールできます:

- ファイルまたはフォルダーの機能がインストールされるとき
- ファイルまたはフォルダーの機能がアンインストールされるとき
- ファイルまたはフォルダーの機能がインストールまたはアンインストールされるとき

削除されるアイテムがフォルダーの場合、そのフォルダーが空の場合のみ削除されます。

プロジェクト内のファイルまたはフォルダーの削除を構成するには、[ファイル] ビューを使います。このビューで、削除するファイルまたはフォルダーを含むフォルダーを選択します。次に、**[インストール先コンピューターのファイル]** ペインを右クリックしてから [ファイルの削除を追加] をクリックします。InstallShield が表示する [プロパティ] ダイアログ ボックスで、使用可能な削除の設定を構成します。

詳しくは、次を参照してください:

- ターゲット システムからファイルとフォルダーを削除する
- ファイル 削除の [プロパティ] ダイアログ ボックス

強化機能

InstallShield には、以下のような強化機能が搭載されています。

実行時に [使用許諾契約] ダイアログで [印刷] の選択肢を提供できる機能

[使用許諾契約] ダイアログの [印刷] ボタンの機能が強化されました。エンド ユーザーが [印刷] ボタンをクリックしたときにデフォルト プリンターに直接印刷する代わりに、今回より、印刷選択ダイアログ ボックスが開きます。

レジストリ関連のシステム検索における 64 ビット サポート

システム検索ウィザードを使って、Windows Installer が実行する検索を定義できます。Windows Installer は、ターゲット システム上にある特定のファイル、フォルダー、レジストリ キー、または .ini の値を検索できます。システム検索ウィザードのレジストリ関連のパネルには、64 ビット ターゲット システムのレジストリの 64 ビット領域をチェックするかどうかを指定できる新しいチェック ボックスが追加されました。

- システム検索ウィザード
- 検索方法パネル (レジストリ検索オプション)
- 32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて

新しい FlexNet Connect 13.06 再配布可能ファイル

InstallShield はプロジェクトで FlexNet Connect 13.06 をサポートします。InstallShield の [アップデート通知] ビューで、2 つの FlexNet Connect 13.06 マージ モジュール (Common Software Manager が含まれているマージ モジュールと、含まれていないマージ モジュール) のどちらかを含みます。

InstallShield 2013 SP1 Express Edition の新しい機能

InstallShield 2013 Service Pack 1 (SP1) には、Windows 8.1、Windows Server 2012 R2、および Visual Studio 2013 をサポートするための変更が含まれています。

Windows 8.1 および Windows Server 2012 R2 システムをターゲットにできる機能

InstallShield では、インストールに Windows 8.1 または Windows Server 2012 R2 が必要であることを指定できます。また、これらのオペレーティング システムに対する機能条件をビルドすることもできます。

Windows 8.1 および Windows Server 2012 R2 にインストール可能な InstallShield 前提条件は、必要に応じて、これらのシステムにインストールされるように更新されています。以前これらのシステムでは、前提条件がデフォルトでは実行されませんでした。これは、次の InstallShield 前提条件に適用します:

- FSharp Redistributable Package 2.0
- JRE_SE 1.7.0_02 (x64)
- JRE_SE 1.7.0_02 (x86)
- Microsoft .NET Framework 3.0 OS Component
- Microsoft .NET Framework 3.5 SP1 (Windows 機能)
- Microsoft .NET Framework 4.5 Full
- Microsoft .NET Framework 4.5 Web
- Microsoft App-V 5.0 SP1 Desktop Client (x64)
- Microsoft App-V 5.0 SP1 Desktop Client (x86)

- Microsoft ReportViewer 2010
- Microsoft SQL CE 3.5 SP2
- Microsoft SQL Server 2005 Express SP3 (x86 & x64Wow)
- Microsoft SQL Server 2005 Express SP3 (x86)
- Microsoft SQL Server 2008 Express SP1 (x64)
- Microsoft SQL Server 2008 Express SP1 (x86 & x64Wow)
- Microsoft SQL Server 2008 Express SP1 (x86)
- Microsoft SQL Server 2008 Management Objects 10.00.2531 (IA64)
- Microsoft SQL Server 2008 Management Objects 10.00.2531 (x64)
- Microsoft SQL Server 2008 Management Objects 10.00.2531 (x86)
- Microsoft SQL Server 2008 Native Client 10.00.2531 (IA64)
- Microsoft SQL Server 2008 Native Client 10.00.2531 (x64)
- Microsoft SQL Server 2008 Native Client 10.00.2531 (x86)
- Microsoft SQL Server 2008 R2 Express RTM (x64)
- Microsoft SQL Server 2008 R2 Express RTM (x86 & x64Wow)
- Microsoft SQL Server 2008 R2 Express RTM (x86)
- Microsoft SQL Server 2008 R2 Express SP2 (x64)
- Microsoft SQL Server 2008 R2 Express SP2 (x86 & x64Wow)
- Microsoft SQL Server 2008 R2 Express SP2 (x86)
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1 (IA64)
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1 (x64)
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1 (x86)
- Microsoft SQL Server 2012 Express LocalDB RTM (x64)
- Microsoft SQL Server 2012 Express LocalDB RTM (x86)
- Microsoft SQL Server 2012 Express RTM (x64)
- Microsoft SQL Server 2012 Express RTM (x86 & x64Wow)
- Microsoft SQL Server 2012 Express RTM (x86)
- Microsoft SQL Server 2012 Native Client (x64)
- Microsoft SQL Server 2012 Native Client (x86)
- Microsoft SQL Server Compact 4.0 (x64)
- Microsoft SQL Server Compact 4.0 (x86)
- Microsoft SQL Server Native Client 9.00.4035 (IA64)
- Microsoft SQL Server Native Client 9.00.4035 (x64)

- Microsoft SQL Server Native Client 9.00.4035 (x86)
- Microsoft SQL Server System CLR Types 10.00.2531 (IA64)
- Microsoft SQL Server System CLR Types 10.00.2531 (x64)
- Microsoft SQL Server System CLR Types 10.00.2531 (x86)
- Microsoft Visual C++ 2005 SP1 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2538242(x64)
- Microsoft Visual C++ 2005 SP1 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2538242(x86)
- Microsoft Visual C++ 2005 SP1 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2005 SP1 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2008 SP1 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2538243(x64)
- Microsoft Visual C++ 2008 SP1 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2538243(x86)
- Microsoft Visual C++ 2008 SP1 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2008 SP1 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2010 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2010 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2010 RTM 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2467173 (x64)
- Microsoft Visual C++ 2010 RTM 再配布可能パッケージ MFC のセキュリティ更新プログラム KB2467173 (x86)
- Microsoft Visual C++ 2010 SP1 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2010 SP1 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2012 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2012 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2012 Update 1 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2012 Update 1 再配布可能パッケージ (x86)
- Microsoft VSTO 2010 Runtime (x64)
- Microsoft VSTO 2010 Runtime

Microsoft Visual Studio 2013 のサポート

InstallShield は、Visual Studio 2013 をサポートします。このバージョンの Visual Studio 内部から InstallShield プロジェクトを作成できます。

新しい Microsoft SQL Server 2012 Express SP1 用の InstallShield 前提条件

InstallShield には、プロジェクトに追加可能な新しい SQL Server 関連の InstallShield 前提条件が含まれています:

- Microsoft SQL Server 2012 Express SP1 LocalDB (x64)
- Microsoft SQL Server 2012 Express SP1 LocalDB (x86)
- Microsoft SQL Server 2012 Express SP1 (x64)

- Microsoft SQL Server 2012 Express SP1 (x86 & x64Wow)
- Microsoft SQL Server 2012 Express SP1 (x86)
- Microsoft SQL Server 2012 Express SP1 Management Objects (x64)
- Microsoft SQL Server 2012 Express SP1 Management Objects (x86)
- Microsoft SQL Server 2012 Express SP1 System CLR Types (x64)
- Microsoft SQL Server 2012 Express SP1 System CLR Types (x86)

これらの InstallShield 前提条件件は、サポートされているターゲット システムにテクノロジーがインストールされます。

新しい、Visual Studio ソリューション フォルダーの定義済みパス変数

ハイレベルなベース ディレクトリを参照する、VSSolutionFolder と呼ばれる新しい定義済みパス変数をプロジェクトで使用できます。このサポートを使うと、InstallShield プロジェクトで Visual Studio ソリューション フォルダー内にある姉妹プロジェクトのファイルへのスタティック リンクを含めることができます。異なるマシン上のプロジェクトで作業を行う場合、VSSolutionFolder パス変数を使用するスタティック リンクは、姉妹プロジェクトのファイルへの正しいパスを参照することができます。

VSSolutionFolder パス変数は、InstallShield が Visual Studio ソリューション内で開かれたときに自動的に定義されます。また、MSBuild を使って InstallShieldプロジェクトを含むソリューションをビルドするときにも、自動的に定義されます。ただし、Visual Studio ソリューションなしで InstallShieldプロジェクトを開いた場合、VSSolutionFolder が自動的に定義されることはありません。たとえば、InstallShieldプロジェクトを、Visual Studio を開かずに InstallShieldインターフェイスで直接開いた場合、VSSolutionFolder は定義されません。同様に、コマンドライン ツール IsCmdBld.exe や、MSBuild で .isproj ファイルを使用する場合、VSSolutionFolder は定義されません。IsCmdBld.exe を使って InstallShieldプロジェクトのリリースをビルドするには、-L コマンドライン パラメーターを使って、VSSolutionFolder の値を設定します。このプロパティは、デフォルトのターゲット ファイルが使用されたとき、ItemGroup InstallShieldPathVariableOverrides として公開されます。

InstallShieldプロジェクトで VSSolutionFolder パス変数を含むパスを持つソース ファイルを含み、それを VSSolutionFolder パス変数がサポートされていない環境でビルドすると、次のようなビルド エラーが発生する可能性があります:

- -6103: ファイル <VSSolutionFolder>¥MyFile.exe が見つかりません
- -6271: ファイル <VSSolutionFolder>¥MyFile.exe が見つかりませんでした。このファイルの MsiFileHash テーブルをビルド中にエラーが発生しました。指定した場所にファイルが存在することを確認します。

詳しくは、次を参照してください:

- Visual Studio ソリューションで VSSolutionFolder パス 変数を使用する
- ファイルとフォルダーをプロジェクトに追加する
- MSBuild を使用して、コマンドラインからリリースをビルドする

InstallShield 2013 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

.NET Framework 3.5 SP1、Microsoft Visual C++ 2012、および SQL Server 2008 R2 Express SP2 用の新しい InstallShield 前提条件

InstallShieldには、Express プロジェクトに追加することができる以下の新しい InstallShield 前提条件が含まれています:

- Microsoft .NET Framework 3.5 SP1 (Windows 機能)
- Microsoft SQL Server 2008 R2 Express SP2 (x64)
- Microsoft SQL Server 2008 R2 Express SP2 (x86 & x64Wow)
- Microsoft SQL Server 2008 R2 Express SP2 (x86)
- Microsoft Visual C++ 2012 Update 1 再配布可能パッケージ (x64)
- Microsoft Visual C++ 2012 Update 1 再配布可能パッケージ (x86)

これらの前提条件は、サポートされているターゲット システムに様々なテクノロジをインストールします。

Windows サービスをインストールおよび開始できる機能

InstallShield は、サービスのインストール中にサービスをインストール、およびアンインストール中にサービスを削除するサポートを含みます。また、オプションでインストール後にサービスを開始、システム起動時に毎回自動的にサービスを開始、または (サービス コントロール マネージャを通してサービスが要求されたときに) オンデマンドでサービスを開始するサポートが含まれています。

プロジェクトに含まれるサービスの情報を構成するには、[サービス] ビューを使用します。

詳しくは、次を参照してください:

- Windows サービスのインストールおよび構成
- [サービス] ビュー

64 ビットの場所をターゲットするインストールを作成する機能

Microsoft では、既存の 32 ビットのアプリケーションがシームレスに動作する 64 バージョンの Windows がデザインされています。また、同一コードが再コンパイルされたバージョンが 64 ビット アプリケーションとしてシームレスに動作する 64 ビット バージョンの Windows もデザインされています。これらをサポートするため、64 ビット バージョンの Windows では、32 ビットの部分と 64 ビットの部分が、ファイルを個別の場所に格納する方法 (例、Program Files と Program Files (x86)、System32 と SysWow64) とレジストリ キーを分ける方法 (HKLM¥Software と HKLM¥Software¥Wow6432Node) という 2 つ方法で分離されています。したがって、エンドユーザーが 32 ビット版 Windows Installer ベースのインストールを 64 ビット システムにインストールしようとすると、Program Files、System32、および HKLM¥Software といった場所へのインストールが構成されているファイル、フォルダー、およびレジストリ エントリは、それぞれ Program Files (x86)、SysWow64、HKLM¥Software¥Wow6432Node にリダイレクトされます。

InstallShield Express Edition では、今回より、64 ビット システムをターゲットとして 64 ビットの場所 (たとえば、Program Files (x86) ではなく Program Files、SysWow64 ではなく System32、HKLM¥Software¥Wow6432Node ではなく HKLM¥Software) にインストールする 64 ビット パッケージを作成するためのサポートが含まれています。以前、このサポートは InstallShield Premier および InstallShield でのみ提供されていました。

この 64 ビット サポートを有効化するため、InstallShield Express Edition には以下の変更が追加されています:

- [ファイル] ビューに、64 ビットの場所用の定義済みフォルダーが含まれています。ファイルまたはフォルダ—を 64 ビット フォルダーにインストールすることを指定するには、適切な定義済み 64 ビット フォルダーにファイルまたはフォルダ—を追加してください。たとえば、フォルダーを 64 ビット システム上の 64

ビット Program Files フォルダーにインストールするには、このビューでそのフォルダーを [ProgramFiles64Folder] ノードに追加します。64 ビット フォルダーは、デフォルトでは表示されません。このビューで 64 ビット フォルダーを表示するには: **[インストール先コンピューターのフォルダー]** ペインでフォルダーを右クリックして、[定義済みフォルダーを表示] をポイントしてから [ProgramFiles64Folder]をクリックします。

- [レジストリ] ビューに、64 ビット レジストリの場所のサポートが追加されました。このビューの **[インストール先コンピュータのレジストリ ビュー]** ペインの SOFTWARE レジストリ エントリは 2 つのノード (SOFTWARE (32 ビット) および SOFTWARE (64 ビット))に分割されました。レジストリ エントリを 64 ビット の場所にインストールすることを指定するには、そのエントリを SOFTWARE (64 ビット) ノード、またはサブノードに追加します。

プロジェクト内の任意のファイル、フォルダー、レジストリ エントリを 64 ビットの場所にインストールするように構成した場合、ビルド時に InstallShield は、64 ビット システム上の 64 ビットの場所にインストールする 64 ビット (x64) .msi パッケージをビルドします。

64 ビット Windows Installer ベースのインストールは、64 ビット システム上でのみ 64 ビットの場所にインストールします。これを 32 ビット システム上で実行することはできません。32 ビット Windows Installer ベースのインストールを 64 ビット システム上で実行することができますが、64 ビットの場所にインストールすることはできません。製品が 32 ビット システムと 64 ビット システムの両方をターゲットにする場合、InstallShield Express Edition を使って 32 ビット システムをターゲットとするプロジェクトと、64 ビット システムをターゲットとするプロジェクトを個別に作成できます。

詳細については、次を参照してください。

- 32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて
- ファイルとフォルダーをプロジェクトに追加する
- [ファイル] ビューで定義済みフォルダーを表示する
- レジストリの編集
- インストール先フォルダー

Windows 8 スタート画面にショートカットをピン留めしないように防ぐサポート

InstallShield では、Windows 8 ターゲット システム上で、インストールに含まれる各ショートカットをデフォルトでスタート画面にピン留めするかどうかを指定できます。インストールの一部であるツールまたは従属的な製品のショートカットのピン留めを無効化したい場合があります。ショートカットのピン留めを無効化した場合でも、システム上のすべてのショートカットを含むアプリケーション一覧には、そのショートカットが表示されます。

ショートカットを [スタート] 画面にピン留めするのを防ぐには、[ショートカット/フォルダー] ビューでショートカットの新しい "Windows 8 スタート画面にピン留めする" 設定を使用します。

詳細については、「[ショートカット/フォルダー] ビュー」を参照してください。

InstallShield 2012 Spring SP1 Express Edition の新しい機能

InstallShield 2012 Spring Service Pack 1 (SP1) には、Windows 8、Windows Server 2012、および Visual Studio 2012 の最終版をサポートするための変更が含まれています。また、その他の変更も含まれています。

Visual Studio 2012、.NET Framework 4.5、および Visual C++ 2012 のサポート

InstallShield Visual Studio 2012 の最終版をサポートするための変更が含まれており、このバージョンの Visual Studio インターフェイス内部でインストールおよび製品の開発が可能です。

さらに、InstallShield には 2 つの .NET Framework 用 InstallShield 前提条件および新しい 2 つの Visual C++ 用 InstallShield 前提条件が提供されています:

- Microsoft .NET Framework 4.5 Full
- Microsoft .NET Framework 4.5 Web
- Microsoft Visual C++ 2012 再配布可能パッケージ (x86)
- Microsoft Visual C++ 2012 再配布可能パッケージ (x64)

.NET Framework の Web 前提条件には、インターネット接続が必要です。この前提条件は、必要に応じて、必須の再配布可能ファイルをダウンロードします。完全な前提条件は、インターネットへの接続が不要なスタンドアロン インストールです。

追加の変更

InstallShield 2012 Spring SP1 で解決されている問題の一覧は、リリース ノートをご覧ください。リリース ノートは、InstallShield の [ヘルプ] メニューからご覧になることができます。

InstallShield 2012 Spring Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

Windows 8 および Windows Server 2012 システムをターゲットできる機能

InstallShield では、インストールに Windows 8 または Windows Server 8 が必要であることを指定できます。また、これらのオペレーティング システムに対する機能条件をビルドすることもできます。

Windows 8 および Windows Server 2012 にインストール可能な InstallShield 前提条件は、必要に応じて、これらのシステムにインストールされるように更新されています。以前これらのシステムでは、前提条件がデフォルトでは実行されませんでした。これは、次の InstallShield 前提条件に適用します:

- FSharp Redistributable Package 2.0
- Microsoft ReportViewer 2010
- Microsoft SQL CE 3.5 SP2
- Microsoft SQL Server 2005 Express SP3
- Microsoft SQL Server 2008 Express SP1
- Microsoft SQL Server 2008 Management Objects 10.00.2531
- Microsoft SQL Server 2008 Native Client 10.00.2531
- Microsoft SQL Server 2008 R2 Express RTM
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1

- Microsoft SQL Server Native Client 9.00.4035
- Microsoft SQL Server System CLR 型 10.00.2531
- Microsoft Visual C++ 2005 SP1 Redistributable MFC セキュリティ更新プログラム KB2538242
- Microsoft Visual C++ 2005 SP1 Redistributable Package
- Microsoft Visual C++ 2008 SP1 Redistributable MFC セキュリティ更新プログラム KB2538243
- Microsoft Visual C++ 2008 SP1 Redistributable Package
- Microsoft Visual C++ 2010 再配布可能パッケージ
- Microsoft Visual C++ 2010 RTM Redistributable MFC セキュリティ更新プログラム KB2467173
- Microsoft Visual C++ 2010 SP1 Redistributable Package
- Microsoft VSTO 2010 Runtime

Microsoft Visual Studio 2012 のベータ サポート

InstallShield には Visual Studio 2012 ベータのサポートが含まれています。このバージョンの Visual Studio 内部から InstallShield プロジェクトを作成できます。

Microsoft .NET Framework 4.5 前提条件

InstallShield には、Express プロジェクトに追加することができる 2 つの新しい .NET 関連の InstallShield 前提条件が含まれています。

- Microsoft .NET Framework 4.5 Full
- Microsoft .NET Framework 4.5 Web

これらの InstallShield 前提条件は、サポートされているターゲット システムに、ベータ バージョンの .NET Framework 4.5 をインストールします。

Web 前提条件には、インターネットへの接続が必要です。この前提条件は、必要に応じて、必須の再配布可能ファイルをダウンロードします。完全な前提条件は、インターネットへの接続が不要なスタンドアロン インストールです。

Microsoft SQL Server 2012 の前提条件

InstallShield には、Express プロジェクトに追加することができる新しい SQL Server 2012 関連の InstallShield 前提条件が含まれています。

- Microsoft SQL Server 2012 Express
- Microsoft SQL Server 2012 Express LocalDB
- Microsoft SQL Server 2012 Native Client

InstallShield には、Microsoft SQL Server 2012 Express の依存関係である Microsoft .NET Framework 3.5 SP1 Update KB956250 をインストールする InstallShield 前提条件も含まれています。

これらの InstallShield 前提条件は、サポートされているターゲット システムにテクノロジーがインストールされます。

SQL Server Compact 4.0 および JRE SE 1.7 の新しい InstallShield 前提条件

InstallShield には、Express プロジェクトに追加することができる新しい InstallShield 前提条件が含まれています:

- Java Runtime Environment Second Edition (JRE SE) 1.7
- SQL Server Compact 4.0

これらの InstallShield 前提条件は、サポートされているターゲット システムにテクノロジーがインストールされます。

新しい FlexNet Connect 13.03 再配布可能ファイル

InstallShield は、Express プロジェクトで FlexNet Connect 13.03 をサポートします。InstallShield の [アップデート通知] ビューで、2 つの FlexNet Connect 13.03 マージ モジュール (Common Software Manager が含まれているマージ モジュールと、含まれていないマージ モジュール) のどちらかを含みます。

InstallShield 2012 SP1 Express Edition の新しい機能

強化機能

InstallShield には、以下のような強化機能が搭載されています。

ソフトウェア識別タグのデジタル署名サポート

プロジェクトにソフトウェア ID タグを含めて [リリース] ビューで .pfx ファイルを使ってリリースに署名を行うことを設定すると、InstallShield がビルド時にタグをデジタル署名します。タグ ファイルに署名するためには、.NET Framework 2.0 以降をビルドマシンにインストールする必要があります。

詳細については、「製品のソフトウェア識別タグを含める」を参照してください。

InstallShield 2012 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

Internet Explorer 9、SQL Server 2008 R2 Native Client、Windows Identity Foundation、その他の再配布可能ファイル用の新しい InstallShield 前提条件

InstallShield には、プロジェクトに追加可能な新しい InstallShield 前提条件が含まれています:

- Internet Explorer 9
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1
- Windows Identity Foundation
- Microsoft VSTO 2010 Runtime (x64)
- Microsoft Office 2010 PIA (この前提条件は Microsoft Office 2010 Primary Interop Assemblies をインストールします。この前提条件を使用するには、マイクロソフトの Web サイトから PrimaryInteropAssembly.exe ファイルをダウンロードおよび実行して、.msi ファイルを抽出します。)

COM 抽出機能の強化

InstallShield では、COM 抽出時に新しい監視方式をサポートします。Windows Vista 以降のシステム上で InstallShield を使用している場合、この新しい方式がデフォルトとなります。この方法は、カーネル ドライバーを使って、ビルド時のダイナミック COM 抽出中、およびデザイン時のスタティック COM 抽出中に変更されたレジストリ領域を監視します。この新しい方式は、DLL が既存のレジストリ エントリを読み込んでビルド マシンへの変更を妨げる以前の方法の利点を組み合わせたものです。

必要な場合、UseAPIRegistryHooks レジストリ値 (32 ビット マシンの場合は HKEY_LOCAL_MACHINE¥SOFTWARE¥InstallShield¥RegSpy レジストリ キーに含まれる、64 ビット マシンの場合は HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥InstallShield¥RegSpy レジストリ キーに含まれる) の値データを設定して 3 つの異なる COM 抽出方式を切り替えることができます。使用可能な REG_DWORD 値データ:

- 0—API フックを使って、既存 DLLのレジストリ エントリを読み取ります。
- 1—レジストリのリダイレクトを使って、ビルド マシン上の登録済み DLL への変更を防止します。値を設定しなかった場合、これが Windows XP および Windows Server 2003 システム上でのデフォルト動作となります。
- 2—新しいカーネル モードの監視を使って、2 つのメソッドの両方の利点を組み合わせます。値を設定しなかった場合、これが Windows Vista 以降のシステム上でのデフォルト動作となります。

Adobe Reader 10、Internet Explorer 9、および Microsoft Office 用の定義済みシステム検索

InstallShield に新しい定義済みシステム検索が追加されました:

- Adobe Reader 10
- Internet Explorer 9
- Microsoft Office 2010
- Microsoft Office 2007
- Microsoft Office 2003

インストールでこれらの 1 つ以上が必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

ソフトウェア識別タグ機能のサポート

ISO/IEC 19770-2 は、ソフトウェア識別タグを作成するための国際規格です。ソフトウェア識別タグは、製品名、製品エディション、パブリッシャーなど、ソフトウェアに関する説明的な情報を含む XML ベースのファイルです。ソフトウェア資産管理ツールは、企業でインストールされているソフトウェアについての正確なアプリケーション ID を提供する目的で、タグ内のデータを収集するツールです。

ソフトウェア識別タグ機能は、業界標準として現在進化しつつあるもので、この機能により、独立系ソフトウェア ベンダーは、顧客に対して、ソフトウェア資産管理およびライセンス最適化イニシアチブに有用な、より適切な情報を提供することができる、より洗練されたアプリケーションを作ることができるようになります。製品のインストール パッケージに識別タグを持たせることで、顧客は、インストールした製品の内部的使用状況を監視できるツールが使えるようになります。これにより、顧客がソフトウェア会社などから入手して精神のライセンスの数を管理および最適化することができるようになり、ライセンス契約内容に違反するリスクがなくなります。

InstallShield の [一般情報] ビューには、製品の識別タグを作成するために必要な情報を指定するための新しい設定がいくつかあります。また、ビルド時にタグを自動生成して、それをインストールに含めるかどうかを指定できる新しい "ソフトウェア識別タグの使用" 設定も追加されました。この設定のデフォルト値は [はい] です。

"ソフトウェア識別タグの使用" 設定で [はい] が選択されている時に、4 つの必須識別設定 ([一般情報] ビューの "一意な ID"、"一意な登録 ID"、"タグ作成者"、および "タグ作成者 ID" 設定) のうち最低 1 つの値が設定されていなかった場合、空白の設定ごとに、ビルド警告 -7235 が一回発生します。このビルド警告では、特定の必須タグが空白であるために、ソフトウェア識別タグが作成されず、インストールに含まれなかったことが通知されます。この警告を解決するには、各設定に適切な値を入力するか、または "ソフトウェア識別タグの使用" 設定で [いいえ] を選択します。

この機能は Express プロジェクトに適用します。

詳細については、次を参照してください。

- 製品のソフトウェア識別タグを含める
- 一般情報の設定

InstallShield 2011 Express Edition の新しい機能

新しい機能

InstallShield には、以下のような新しい機能が搭載されています。

Team Foundation Server (TFS) との統合

InstallShield では、Team Foundation Server (TFS) 2010 との統合サポートが強化されています。

Visual Studio 2010 内から InstallShield を使用する場合、Source Control Explorer にアクセスして、InstallShield プロジェクトを Team Foundation バージョン コントロールと統合し、InstallShield プロジェクトと Visual Studio ソリューションへの変更を管理することができます。

Team Foundation Build を使って、InstallShield プロジェクトと Visual Studio ソリューションを定期的、またはオンデマンドでコンパイル、テスト、およびデプロイすることもできます。インストールはソリューションがビルドされる度に、自動的に最新のソース ファイルで更新されます。

さらに、InstallShield と Visual Stuio が搭載された同じマシン上に Team Explorer をインストールした場合、Visual Studio で開かれている InstallShield プロジェクト内から Team Explorer を使用できます。これで、次のようなタスクを行うことができます:

- InstallShield プロジェクトでの作業中にソース管理エクスプローラーを使用。
- InstallShield プロジェクトと Visual Studio ソリューションのビルドを構成。
- 新しいビルドをキューに配置。
- InstallShield プロジェクトと Visual Studio ソリューションのバグやタスクなどの作業項目を追跡する。

詳細については、次を参照してください。

- Microsoft Visual Studio Team Foundation Server との統合
- InstallShield プロジェクトを Team Explorer に追加する

SQL Server 2008 R2 Express、SQL Server Native Client、Visual C++ 2010 用の新しい InstallShield 前提条件およびその他の再配布可能ファイル

InstallShield には、Express プロジェクトに追加することができる多くの新しい InstallShield 前提条件が追加されています:

- Microsoft SQL Server 2008 R2 Express
- Microsoft SQL Server 2008 Native Client 10.00.2531
- Microsoft SQL Server Native Client 9.00.4035
- Microsoft SQL Server System CLR 型 10.00.2531
- Microsoft SQL Server 2008 Management Objects 10.00.2531
- Microsoft Visual C++ 2010 再配布可能パッケージ
- Microsoft Visual C++ 2008 SP1 Redistributable Package
- Windows Installer 3.1 - 日本語
- MSXML 6.0 SP1 - 日本語
- Microsoft .NET Framework 4.0 Client Language Pack - 日本語
- Microsoft .NET Framework 4.0 Full Language Pack - 日本語

InstallShield 前提条件の検索パスを指定できる機能

InstallShield では、今回より、InstallShield 前提条件ファイル (.prq ファイル) を検索するフォルダを指定できます。この機能によって、複数の開発者の間で InstallShield 前提条件を共有し、それらをソース コード管理システムで保管することが容易になります。以前は、InstallShield が .prq ファイルを検索するのは、*InstallShield Program Files フォルダー*¥SetupPrerequisites のみでした。

InstallShield でフォルダーを指定するとき、いくつかの方法があります:

- InstallShield 内部から編集またはビルドを行う場合、[ツール] メニューで [オプション] をクリックすると表示される [オプション] ボックスにある新しい [前提条件] タブを使って、マシン共通および現在のユーザーのフォルダをコンマ区切りのリストで指定できます。このタブは、マージ モジュールの検索パスを指定できる、[オプション] ダイアログ ボックスの [マージ モジュール] タブと似ています。
- ISCmdBld.exe を使ってコマンドラインからビルドを行う場合、新しい -prqpath パラメーターを使って、コンマ区切りのフォルダーのリストを指定できます。

  .ini ファイルを使って ISCmdBld.exe パラメーターを指定する場合、.ini ファイルの [Mode] セクションで新しい PrerequisitePath パラメーターを使用して、フォルダのコンマ区切りのリストを指定できます。
- MSBuild または Team Foundation Server (TFS) を使ってビルドする場合、InstallShield タスクで新しい PrerequisitePath パラメーターを使います。このパラメーターは、デフォルトのターゲット ファイルが使用されたとき、ItemGroup InstallShieldPrerequisitePath として露出されます。複数のパスを指定するには、順序指定されたパスの配列を使用します。

詳しくは、次を参照してください:

- InstallShield 前提条件を含むディレクトリを指定する
- [前提条件] タブ
- IsCmdBld.exe

- .ini ファイルでコマンドライン ビルド パラメーターを渡す
- Microsoft ビルド エンジン (MSBuild)

Setup.exe および Update.exe のカスタム バージョン リソースのプロパティを指定できる機能

InstallShield では、今回より、ビルド時に作成する Setup.exe ファイルでカスタム バージョン リソースのプロパティを使用できます。バージョン リソースのプロパティは、Setup.exe の [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Setup.exe ファイルを右クリックして、[プロパティ] をクリックしたときに表示されます。このサポートは、Express プロジェクトで提供されています。カスタム バージョン リソースのプロパティを指定できる機能は、QuickPatch プロジェクトで作成した Update.exe ファイルにも使用できます。

[リリース] ビューにあるリリースの [Setup.exe] タブに 2 つの新しい設定が追加されました。これらの設定を使って、カスタム著作権情報およびファイルの説明を指定することができます。

InstallShield がビルド時に Setup.exe 起動ツールの以下のバージョン リソースを構成するとき、今回より、[一般情報] ビューおよび [リリース] ビューで入力されたカスタム情報を使用します。

- 会社名
- 製品名
- 製品バージョン
- 著作権情報
- ファイル バージョン
- ファイルの説明

以前、InstallShield は多くの場合にカスタム情報を使用しませんでした。たとえば、InstallShield が以前に作成した Setup.exe ファイルには、その Setup.exe ファイルをビルドした InstallShield のバージョン固有の詳細が含まれました。このため、Setup.exe の [プロパティ] ダイアログ ボックスに、製品の著作権情報ではなく、InstallShield の著作権情報が表示されました。

"会社名"、"製品名"、"製品バージョン"、"説明"、"著作権情報" といった新しい設定を使って、InstallShield で Update.exe ファイルをビルドするときに使用するカスタム情報を指定できます。QuickPatch プロジェクトの場合、[一般情報] ビューの [ビルドの設定] 領域にある [詳細] タブで設定できます。

以前、InstallShield は Update.exe ファイルにカスタム バージョン リソース情報を使用しませんでした。

詳しくは、次を参照してください:

- セットアップランチャーのファイルのプロパティをカスタマイズする
- Setup.exe タブ
- アップデート起動ツールのファイルのプロパティをカスタマイズする
- [詳細] タブ

カスタム アクションの強化: 新しいカスタム アクションの種類 (新しいメンテナンス モード シーケンスのオプション、ロールバックおよび コミット カスタム アクションのサポート)

InstallShield には、カスタム アクションをサポートするいくつかの新しい機能が搭載されています。

MSI DLL カスタム アクション

InstallShield では、今回より、プロジェクトに MSI DLL カスタム アクションを追加できます。MSI DLL カスタム アクションは C または C++ DLL からエクスポートされたエントリポイント関数を呼び出します。DLL は、インストールのランタイムでインストールするか、.msi パッケージの Binary テーブル内に埋め込むことができます。MSI DLLカスタム アクションは、以前のバージョンの InstallShield でサポートされていた唯一の DLL の種類である標準 DLL カスタム アクションよりも柔軟に DLL ファイルのソース場所を選択することができます。インストールの一部では標準 DLL アクションを Binary テーブルに格納することはできないため、標準 DLL アクションの場合は製品とともにインストールする必要があります。しかし、インストールのこの同じ部分で、MSI DLL アクションの場合は Binary テーブルに格納するか、または製品と共にインストールすることができます。

MSI DLL カスタム アクションの場合、戻り値ゼロは成功、ゼロ以外の戻り値は失敗またはキャンセルを示します。これは、標準 DLL カスタム アクション以外、その他すべて種類のカスタム アクションで共通です。標準 DLL カスタム アクションの場合、ゼロ以外の戻り値は成功、ゼロは失敗を意味します。

JScript カスタム アクション

InstallShield では、プロジェクトに JScript カスタム アクションを追加できます。JScript カスタム アクションを含む Visual Studio セットアップ プロジェクト (.vdproj) を Express プロジェクト (.ise) に変換すると、InstallShield は今回より、JScript カスタム アクションを含みます。

個別のメンテナンスおよびアンインストールシーケンス

[カスタム アクション] ビューに、メンテナンス中にのみ発生させるアクションをスケジュールすることができる [メンテナンス中のカスタム アクション] ノードが追加されました。以下のシーケンス ノードが、[アンインストール中のカスタム アクション] ノードから、この新しい [メンテナンス中のカスタム アクション] ノードに移動しました:

- 初期化の後 (最初のダイアログの前)
- [メンテナンス - ようこそ] ダイアログの後
- [メンテナンスの種類] ダイアログの後
- [削除の準備完了] ダイアログの後
- [セットアップ進行状況] ダイアログの後
- [セットアップの完了] ダイアログの後

2 つのシーケンス、[システム変更の前] および [システム変更の後] は、引き続きアンインストールに適用するため、[アンインストール中のカスタム アクション] ノードの下に残ります。これらのシーケンス中にスケジュールされているカスタムアクションはすべて、アンインストール中に実行されますが、修復などのメンテナンス処理中には実行されません。

ロールバック アクション、コミット アクション、および関連設定

[カスタム アクション] ビューに追加された新しい "スクリプト内実行" 設定を使って、実行シーケンスの「遅延」、「ロールバック」、「コミット」のどの反復でカスタム アクションをトリガーするのかを選択できます。遅延アクションは、システムにランタイム変更を行います。ロールバック アクションは、インストールが完了する前にエラーが発生した場合、またはエンド ユーザーがインストールのキャンセルを行った場合に、システムに行われた変更を元に戻すために実行されます。コミット アクションは、遅延アクションによって保存された一時的な情報のクリーンアップを行います。"スクリプト内実行" 設定を使って、アクションをユーザー コンテキスト (インストールを実行中のユーザーの権限) で実行するのか、システム コンテキスト (昇格された権限) で実行するのかを指定することもできます。

この新しい "スクリプト内実行" 設定は、即時実行モードでシーケンスされていないカスタム アクションを選択したときに、[カスタム アクション] ビューに表示されます。即時実行モードで実行されるアクションは、常にユーザー コンテキストで実行し、Windows Installer のプロパティの設定や、ターゲット システムのチェックを行います。

以前の InstallShield Express Edition では、インストールの遅延シーケンスにスケジュールされたカスタム アクションはすべて、ユーザーコンテキストで遅延実行モードで起動しました。InstallShield はロールバックまたはコミット カスタム アクションをサポートせず、システム コンテキストでのカスタム アクションの実行もサポートしませんでした。

詳細については、次を参照してください。

- カスタム アクションを使用する
- Windows Installer DLL カスタム アクション
- VBScript および JScript カスタム アクション
- アクションの実行のオプション

Visual Studio セットアップとマージ モジュール プロジェクトを既存の InstallShield プロジェクトにインポートする機能 (プロジェクト コンバーターの強化)

InstallShield を使って、Visual Studio セットアップ プロジェクトまたは Visual Studio マージ モジュール プロジェクト (.vdproj) を Express プロジェクト (.ise) にインポートできます。この機能を使って、Visual Studio プロジェクトに含まれている同じデータと設定を含む InstallShield インストール プロジェクトを開発できます。ウィザードを使って、プロジェクト出力、ファイル、レジストリ キー、ファイル拡張子、カスタム アクション、ターゲット システム検索、および起動条件を Visual Studio プロジェクトから既存の InstallShield プロジェクトにインポートします。

Visual Studio プロジェクトを既存の InstallShield プロジェクトにインポートするには、InstallShield の Visual Studio デプロイメント プロジェクト インポート ウィザードを使います。このウィザードでは、Visual Studio プロジェクト内の特定の設定をインポートするか、無視するかを選択することができます。

Visual Studio プロジェクトを新しい InstallShield プロジェクトに変換するための、これまでのサポートが拡張されました。Visual Studio プロジェクトに定義済み前提条件が含まれている場合、InstallShield は今回より、プロジェクトの変換中にそれらを対応する InstallShield 前提条件に変換します。これと同じ前提条件の変換機能が、Visual Studio プロジェクトを InstallShield プロジェクトにインポートするための新しいウィザードにも搭載されています。

Visual Studio プロジェクトに 1 つ以上のプロジェクト出力が含まれている場合は、変換プロセスの代わりにインポート ウィザードを使用します。InstallShield プロジェクトは、Visual Studio セットアップまたはマージ モジュール プロジェクトおよびそのプロジェクト依存関係が含まれているのと同じ Visual Studio ソリューションに含まれていなくてはなりません。インポート ウィザードを使ってプロジェクト出力を InstallShield プロジェクトにインポートするためには、InstallShield を Visual Studio 内部で開いてください。

詳しくは、次を参照してください:

- Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートする
- Visual Studio デプロイメント プロジェクト インポート ウィザード

SQL Server 2008 Express SP1 および Adobe Reader 9 のための定義済みシステム検索

InstallShield に新しい定義済みシステム検索が追加されました:

- SQL Server 2008 Express SP1

- Adobe Reader 9

インストールでこれらのいずれか、または両方が必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

この機能は Express プロジェクトに適用します。

IIS Web サイト、アプリケーション、および仮想ディレクトリの MIME の種類を構成する機能

[IIS 構成] ビューに新しく追加された "MIME の種類" 設定を使って、プロジェクトに含まれる Web サイト、アプリケーション、または仮想ディレクトリの MIME の種類を構成できます。この設定を使って、ターゲット システム上の Web サーバーからブラウザーまたはメール クライアントに送信できるコンテンツの種類を指定できます。

この機能は Express プロジェクトに適用します。

詳細については、次を参照してください。

- Web サイトにおける MIME の種類設定
- アプリケーションまたは仮想ディレクトリにおける MIME の種類設定

イメージ ビルボードのループ サポート

[ビルボード] ビューに新しく追加された "ビルボードのループ" 設定を使って、インストールがファイルの転送を完了するまでイメージ ビルボードを継続してループ表示してから適切な [セットアップの完了] ダイアログを表示するかどうかを指定できます。

この設定で [はい] を選択して、ビルボードに割り当てた時間よりもファイルの転送に時間がかかった場合、インストールは最初のビルボードから再び表示します。必要な場合、ループはファイルの転送が終了するまで継続します。この設定のデフォルト値は [いいえ] です。これは、InstallShield の以前のバージョンの動作と同じです。

以前は、ビルボードに割り当てられた時間よりもファイルの転送に時間がかかった場合、インストールはファイル転送が終了するまで最後のビルボードを表示し続け、ビルボードをループしませんでした。

この機能は Express プロジェクトに適用します。

詳細については、次を参照してください。

- ビルボードを含むインストールの実行時の動作
- ビルボード設定

新しい FlexNet Connect 12.01 再配布可能ファイル

InstallShield、InstallShield プロジェクトで FlexNet Connect 12.01 をサポートします。InstallShield の [アップデート通知] ビューを使って、2 つの FlexNet Connect 12.01 マージ モジュールうち、いずれかが含まれています (Common Software Manager が含まれているマージ モジュールと、含まれていないマージ モジュール)。

強化機能

InstallShield には、以下のような強化機能が搭載されています。

InstallShield の Unicode ビュー

InstallShield のいくつかのビューは、すべての言語からの文字を表示および入力できるように強化されています。たとえば、英語マシン上で [要件] ビューにおいてソフトウェアの要件を構成するときに、システム ソフトウェア要件メッセージに中国語の文字を使用できるようになりました。以前は、メッセージ内の文字が疑問符で表示されました。

InstallShield で Unicode サポートが強化された領域は、[要件] ビューおよび [リリース] ビュー内のリリースのタブです。InstallShield 2010 Express Edition より、その他の多くのビューで Unicode がサポートされています。

強化は、Express プロジェクトに適用します。

Visual Studio Web セットアップ プロジェクトからプロジェクト出力を追加するサポート

Web セットアップ プロジェクトを含む Visual Studio ソリューションと、InstallShield インストール プロジェクトを作成した場合に、Visual Studio 内部から InstallShield を使用しているとき、今回より、Web セットアップ プロジェクトのプロジェクト出力を InstallShield プロジェクトに追加することができます。

Update.exe マニフェストに必要実行レベルを指定する機能

"必要実行レベル" 設定は、QuickPatch プロジェクトで [一般情報] ビューの [ビルドの設定] 領域にある [詳細] タブで使用できます。この新しい設定を使って、Windows Vista 以降のプラットフォーム上でプロジェクトの Update.exe ファイルがアップグレードを実行するために必要とする最低実行レベルを指定します。InstallShield が、必要レベルを指定するマニフェストを追加します。デフォルトで、InstallShield は以前のセットアップランチャーのマニフェストで構成されたレベルを使用します。

以前、"必要実行レベル" 設定は Setup.exe セットアップランチャーでのみ使用できました。Update.exe パッチを作成するとき、InstallShield は以前のセットアップ ランチャーのマニフェストで構成された実行レベルを必要としました。

詳細については、「[詳細] タブ」を参照してください。

非圧縮 .cab ファイルの作成機能

[リリース] ビューで選択されたリリースの [ビルド] タブに、新しい "Cab の最適化タイプ" 設定が追加されました。"圧縮" 設定で [圧縮] またはカスタム オプションの 1 つを選択した場合、"Cab 最適化タイプ" 設定を使って、リリースの .cab ファイルをビルドするときに InstallShield が使用する圧縮の種類を指定します。使用可能なオプションには、[LZX 圧縮]、[MSZIP 圧縮]、または [非圧縮] があります。

"Cab の最適化タイプ" 設定は、これまでの "サイズの最適化" 設定に取って代わります。"サイズの最適化" 設定は、LZX 圧縮と MSZIP 圧縮のみをサポートし、非圧縮オプションは提供されていませんでした。

この強化は、Express プロジェクトで提供されています。

詳細については、「[ビルド] タブ」を参照してください。

[再配布可能ファイル] ビューと [前提条件] ビューの新しい [更新] ボタン

[再配布可能ファイル] ビューに、表示される再配布可能ファイルのリストを更新するための新しい [更新] ボタンが追加されました。以前は、InstallShield でこのビューが開かれている状態でコンピュータに再配布可能ファイルを追加したとき、更新済みのリストを表示するためには、プロジェクトを一度閉じてから再び開く必要がありました。

重要な情報

InstallShield の複数エディションをインストールする

InstallShield 2011 (InstallShield Premier、InstallShield、または Express) は、同時に同じシステム上に 1 つのエディションのみをインストールできます。

InstallShield の複数バージョンをインストールする

InstallShield 2021 Express Edition は、同じマシン上で別のバージョンの InstallShield と共存することができます。

InstallShield 2010 Express Edition Expansion Pack for Visual Studio 2010 の新しい機能

InstallShield 2010 Expansion Pack for Visual Studio 2010 は、Visual Studio 2010 および .NET Framework 4 の最終版をサポートするための変更が含まれています。また、その他の変更も含まれています。

Microsoft .NET Framework 4 前提条件

InstallShield には、Express プロジェクトに追加することができる 4 つの新しい .NET 関連の InstallShield 前提条件が含まれています。

- Microsoft .NET Framework 4.0 Full
- Microsoft .NET Framework 4.0 Full (Web ダウンロード)
- Microsoft .NET Framework 4.0 Client
- Microsoft .NET Framework 4.0 Client (Web ダウンロード)

これらの前提条件ついて以下の点にご注意ください:

- 完全版の前提条件は、.NET Framework 4 をターゲットにするアプリケーションを実行および開発するのに必要な .NET Framework ランタイムと関連ファイルをインストールします。
- クライアント前提条件は、ほとんどのクライアント アプリケーションを実行するのに必要な .NET Framework ランタイムと関連ファイルをインストールします。
- 2 つの Web ダウンロード前提条件には、インターネットへの接続が必要です。これらの前提条件のダウンロードには、適切な場合、再配布可能ファイルが必要です。その他 2 つの前提条件は、インターネットへの接続が不要なスタンドアロン インストールです。

.NET Framework 4.0 には、Windows Installer 3.1 以降、および Windows Imaging Component が必要です。したがって、.NET Framework 4.0 Full 前提条件は、以下の InstallShield 前提条件を依存関係とします:

- Windows Installer 3.1 (x86)
- Windows Imaging Component (x86)
- Windows Server 2003 SP1 (x86) 用 Windows Installer 3.1
- Windows Imaging Component (x64)
- Windows Server 2003 SP1 (IA64) 用 Windows Installer 3.1
- Windows Server 2003 SP1 (x64) 用 Windows Installer 3.1

- Windows XP (x64) 用 Microsoft Windows Installer 3.1

また、.NET Framework 4.0 Client 前提条件は、以下の InstallShield 前提条件を依存関係とします:

- Windows Installer 3.1 (x86)
- Windows Imaging Component (x86)
- Windows Server 2003 SP1 (x86) 用 Windows Installer 3.1
- Windows Imaging Component (x64)
- Windows Server 2003 SP1 (x64) 用 Windows Installer 3.1
- Windows XP (x64) 用 Microsoft Windows Installer 3.1

したがって、任意の .NET Framework 4.0 前提条件をプロジェクトに追加すると、InstallShield はデフォルトで Windows Installer と Windows Imaging Component の前提条件ををインストールに追加します。このため、インストールのサイズが大きくなることがあります。

Visual Studio 2010 サポートのための追加前提条件

InstallShield には、Express プロジェクトに追加することができる以下の新しい InstallShield 前提条件が含まれています:

- Microsoft SQL CE 3.5 SP2
- Microsoft ReportViewer 2010
- Microsoft VSTO 2010 Runtime
- FSharp Redistributable Package 2.0
- Microsoft Office 2007 PIA (この前提条件は Microsoft Office 2007 Primary Interop Assemblies をインストールします。この前提条件を使用するには、マイクロソフトの Web サイトから PrimaryInteropAssembly.exe ファイルをダウンロードおよび実行してo2007pia.msi ファイルを抽出します。)システム上の .msi パッケージの場所によって、.prq ファイル内の o2007pia.msi インストールのパスを変更しなくてはならない場合があります。)

.NET Framework 4 用の定義済みシステム検索

InstallShield に 2 つの新しい定義済みシステム検索が追加されました:

- Microsoft .NET Framework 4.0 Full パッケージ
- Microsoft .NET Framework 4.0 Client パッケージ

インストールでこれらのいずれかが必要な場合、プロジェクト アシスタントの [インストール要件] ページまたは [要件] ビューを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

この機能は Express プロジェクトに適用します。

InstallShield 2010 Express Edition SP1 の新しい機能

InstallShield 2010 Express Edition Service Pack 1 (SP1) には、Windows 7、Windows Server 2008 R2、および Windows Installer 4.5 の最終版をサポートするための変更が含まれています。また、その他の変更も含まれています。

重要 *・InstallShield 2010 SP1 で InstallShield 2010 を開くとき、プロジェクトを InstallShield 2010 SP1 にアップグレードすることを許可する必要があります。InstallShield 2010 SP1 では、InstallShield 2010 プロジェクトで使用できないテーブルをサポートするため、アップグレード中にこれらのテーブルを追加しなくてはなりません。InstallShield 2010 SP1 プロジェクトを以前のバージョンの InstallShield (SP1 を適用する前の InstallShield 2010 を含む) で開くことはできませんので、ご注意ください。このため、複数のユーザーが InstallShield プロジェクトを開いて編集する場合、すべてのユーザーが同時に SP1 パッチを適用するようにしてください。*

InstallShield 2010 プロジェクトを InstallShield 2010 SP1 で開くと、そのプロジェクトを新しいバージョンに変換するかどうかをたずねるメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前にプロジェクトのバックアップ コピーが作成されます。

Setup.exe マニフェストには、Windows 7 および Windows Server 2008 R2 システム上で Program Compatibility Assistant のトリガーを回避するための、互換性セクションが追加されました。

InstallShield プロジェクトで、インストール用のセットアップランチャーの作成を構成した場合、InstallShield がセットアップランチャーに対して作成するマニフェストには、今回より互換性セクションが含まれます。以前は、この互換性セクションが含まれておらず、Windows 7 および Windows Server 2008 R2 システム上でインストールの終了時に Program Compatibility Assistant (PCA) ダイアログ ボックスが表示されました。この PCA ダイアログ ボックスは、プログラムが正しくインストールされなかった可能性があることを通知しました。このダイアログ ボックスは、インストールがアプリケーションのアンインストール キーを作成しなかった場合に表示されました。この状況は、エンド ユーザーがインストールをキャンセルした場合、またはインストールが正しく完了しなかった場合に発生します。

追加の変更

InstallShield 2010 Express Edition SP1 で解決されている問題の一覧は、リリース ノートをご覧ください。リリース ノートは、InstallShield の [ヘルプ] メニューからご覧になることができます。

InstallShield 2010 Express Edition の新しい機能

新しい機能

InstallShield 2010 Express Edition には、以下のような新しい機能が搭載されています。

Windows 7 および Windows Server 2008 R2 システムをターゲットにできる機能

InstallShield では、インストールに Windows 7 または Windows Server 2008 R2 が必要であることを指定できます。また、これらのオペレーティング システム用の機能条件やカスタム アクション条件をビルドすることができます。

Windows 7 および Windows Server 2008 R2 でタスクバーにインストールの進行状況を表示するサポート

Windows 7 と Windows Server 2008 R2 で実行されるインストールは、今回より、ファイル転送中に Windows タスクバーに進行状況バーを表示します。これは、[ビルボード] ビューで構成されたビルボードを表示するインストールに適用します。進行状況バーは、Windows の以前のバージョンのタスクバーには表示されません。また、セットアップの初期化中、または InstallShield 前提条件のインストール中にも表示されません。

Beta Windows Installer 5 におけるユーザーごとインストールをサポート

[ダイアログ] ビューにある "[すべてのユーザー] オプションの表示" 設定では、Windows Installer 5 のベータ版で提供されている MSIINSTALLPERUSER プロパティが使用できます。この設定を使って、エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供するかどうかを指定できます。この設定で選択した値に基づいて、[インストールの準備完了] ダイアログに、エンド ユーザーが製品をインストールする方法を指定できるボタンが含まれます。このボタンは、Windows 7 または Windows Server 2008 R2 が搭載されたシステム上でインストールが実行されたときに表示されます。

"[すべてのユーザー] オプションの表示" 設定は、[ダイアログ] ビューでメイン [ダイアログ] ノードを選択すると使用できます。以前は、このビューの [ユーザー情報] ダイアログを選択すると使用できました。

この機能は、Express プロジェクトで提供されています。

詳細については、次を参照してください。

- ユーザーごとのインストールとマシンごとのインストールの違い
- すべてのエンドユーザー ダイアログのためのグローバル ダイアログ設定

Windows Installer 5 Beta における大きいパッケージのインストールにかかる所要時間の短縮機能をサポート

[一般情報] ビューにある "高速インストール" 設定では、大きい Windows Installer パッケージのインストールにかかる所要時間を短縮するのに役立つ 1 つまたは複数のオプションを選択できます。たとえば、インストールについてシステム復元ポイントを保存しないことを指定できます。インストールがファイル コスティングのみを実行して、その他のコスト チェックをスキップするように指定することも可能です。

この設定は、新しい Windows Installer プロパティ MSIFASTINSTALL を構成します。このプロパティは、コマンドラインで設定できます。Windows Installer 5 で、このプロパティがサポートされています。以前のバージョンの Windows Installer はこれを無視します。

この設定は、Express プロジェクトで提供されています。

詳細については、"高速インストール" 設定の説明を参照してください。

ファイル、フォルダー、およびレジストリ キーのアクセス許可を設定するための新しいサポート

InstallShield では、ロックダウン環境で製品を実行するエンド ユーザー向けに、ファイル、フォルダー、およびレジストリ キーを保護するための新しい方法が提供されています: 新しいカスタム InstallShield 処理を使うと、InstallShield は .msi データベースのカスタム ISLockPermissions テーブルに製品のアクセス許可情報を格納します InstallShield はまた、アクセス許可を設定するためのカスタム アクションをプロジェクトに追加します。このサポートは、Express プロジェクトで提供されています。

以前、InstallShield でアクセス許可を設定するための唯一のオプションは、従来型の Windows Installer 処理のみでした。このオプションを使うと、 .msi データベースの LockPermissions テーブルに製品のアクセス許可情報が格納されます。新しい [カスタム InstallShield 処理] オプションには、従来型の Windows Installer 処理よりも多くの利点があります:

- カスタム オプションを使うと、従来型の Windows Installer 処理オプションではサポートされていない、多くのよく知られているセキュリティ識別子 (SID) を使用できます。
- 従来型のオプションとは違い、カスタム オプションでは、サポート対象の SID に翻訳されたユーザー名を使用できます。従来型のオプションで、非英語システム上で翻訳された名前を使ってアクセス許可を設定すると、インストールが失敗する可能性があります。
- カスタム オプションを使うと、指定するアクセス許可を特定のユーザーまたはグループが所持することを拒否できます。従来型の処理で、これは不可能です。
- カスタム オプションを使って、ターゲット システムに既存するファイル、フォルダー、またはレジストリ キーに、アクセス許可を追加できます。このとき、そのオブジェクトに既存するアクセス許可は削除されません。従来型の処理では、既存するアクセス許可が削除されました。
- このカスタム オプションを使って、フォルダー (またはレジストリ キー) のアクセス許可を構成し、そのフォルダーのサブフォルダーおよびファイル (またはレジストリ キーのサブキー) すべてに同じアクセス許可を適用するかどうかを指定できます。従来型の処理では、フォルダー内のサブフォルダーまたはファイル (レジストリ キーの下にあるサブ キー) にアクセス許可を構成する場合、ターゲット システム上で作成された親は、自動的に子のアクセス許可を継承します。
- カスタム オプションを使って、インストール中に作成される新しいユーザーのアクセス許可を構成できます。従来型の処理では、実行時にユーザーがターゲット システム上に既存する必要があったため、これは不可能でした。

[一般情報] ビューに追加された "ロックダウンの設定方法" 設定を使って、プロジェクトに含まれるファイル、フォルダー、およびレジストリ キーに設定する新しいアクセス許可について、新しいカスタム InstallShield 処理または従来型の Windows Installer 処理のどちらを使用するかを指定できます。プロジェクトで既にいくつかのアクセス許可を構成済みの場合にこの設定の値を変更する場合、InstallShield では、既存アクセス許可に別の処理方法を使用するかどうかを指定できます。すべての新しいプロジェクトにおける、この設定のデフォルト値は [カスタム InstallShield 処理] オプションです。プロジェクトを InstallShield 2009 Express Edition 以前から InstallShield 2010 Express Edition にアップグレードした場合、この設定のデフォルト値は [従来型の Windows Installer 処理] オプションです。この新しい設定は、Express プロジェクトで使用できます。

詳細は、次を参照してください:

- ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護
- プロジェクトで、ロックダウン環境でのアクセス許可タイプを選択する

Windows Installer、.NET Framework、Crystal Reports、およびその他の再配布可能ファイルのための新しい InstallShield 前提条件

InstallShield には、Express プロジェクトに追加することができる多くの新しい InstallShield 前提条件が追加されています:

- Windows Installer 4.5 (Windows Installer 4.5 用の InstallShield 前提条件は、Microsoft KB958655 の修正を含みます。)
- Windows Installer 4.5 Update (Windows Installer 4.5 用の InstallShield 前提条件は、Microsoft KB958655 の修正を含みます。このアップデートには、Windows Installer 4.5 がターゲット システム上にインストール済みでなくてはなりません。)
- Windows Installer 3.1、Windows Installer 3.0、および Windows Installer 2.0 (これらのバージョンの Windows Installer 再配布可能ファイルは、以前、[リリース] ビューを使ってプロジェクトに Windows Installer を追加すると利用することができました。InstallShield 前提条件としては提供されていませんでした。)

- .NET Framework 3.0 SP1
- .NET Framework 2.0 SP2
- Internet Explorer 8
- Microsoft SQL Server 2008 Express SP1
- Microsoft SQL Server 2005 Express SP3
- Microsoft Visual C++ 2005 SP1 Redistributable Package
- Crystal Reports Basic for Visual Studio 2008 (Visual Studio 2008 と共にインストールされた Crystal Reports Basic と共に使用できる前提条件。システム上の .msi パッケージの場所によって、.prq ファイル内の Crystal Reports Basic のパスを変更しなくてはならない場合があります。)

仮想ディレクトリを使わずに IIS Web サイトを Web サイトに追加できる機能

InstallShield では、今回より、IIS Web アプリケーションを Web サイトに追加できます。そのためには、[IIS 構成] ビューで Web サイトを右クリックしてから、[新しいアプリケーション] をクリックします。新しいアプリケーションを追加した後、その設定を右側のペインで構成できます。

また、InstallShield ではアプリケーション無しで仮想ディレクトリを作成することもできます。以前は、仮想ディレクトリを作成するたびに、アプリケーションも自動的に作成されました。

この機能は Express プロジェクトに適用します。

詳しくは、次を参照してください:

- Web サイトの作成とアプリケーションまたは仮想ディレクトリの追加
- ≫ [IIS 構成] ビュー

新しい FlexNet Connect 11.6 再配布可能ファイル

InstallShield は、Express プロジェクトで FlexNet Connect 11.6 をサポートします。InstallShield の [アップデート通知] ビューを使って、2 つの FlexNet Connect 11.6 マージ モジュールうち、いずれかが含まれています (Common Software Manager が含まれているマージ モジュールと、含まれていないマージ モジュール)。これらのマージモジュールは、FlexNet Connect 11 マージ モジュールに取って代わります。

強化機能

InstallShield 2010 Express Edition には、次のような新しい強化機能が含まれています。

ユーザビリティの強化点

InstallShield の一部のビューは、その生産性とユーザビリティが強化されています。たとえば、一部のビューにはオプションを簡単に見つけられるツールバーが含まれています。いくつかのビューでは、グリッド内の行を整列する方法をカスタマイズできます。検索機能があるビューでは、より迅速に検索が行われます。主要な例は以下のとおりです:

- [再配布可能ファイル] ビュー – このビューに含まれる新しいツールバーとグループ ボックス領域は、強力な検索および組織化機能を提供します。列ヘッダーをグループ ボックス領域にドラッグ アンド ドロップして、再配布可能ファイルのリストを階層形式で表示することができます。また、ツールバーの検索ボックスに文字列を入力すると、InstallShield はその文字を含まないすべての再配布可能ファイルを隠します。
- [IIS 構成] ビュー – このビューは IIS 7 に似せて外観デザインが一新され、設定がタブではなくグリッドで表示されるようになりました。グリッドには、カテゴリ別、またはアルファベット順にグリッド設定を並べ替

えるためのボタンがあります。このビューのグリッドの中から設定の 1 つを選択すると、右下のペインにその設定のヘルプ情報が表示されます。

- [一般情報] ビュー – このビュー内の設定はいくつかのカテゴリにグループ分けされていて、特定の設定を見つけやすくなっています。カテゴリ別またはアルファベット順にグリッド設定を並べ替えることができるボタンもあります。

また、リリースをビルドまたはプロジェクトをアップグレードするときに表示される [出力] ウィンドウが強化されました。[出力] ウィンドウ、またその個別のタブは、InstallShield のワークスペースの任意の側面に固定するか、独立した場所にドラッグすることができます。[出力] ウィンドウまたはそのタブの 1 つを InstallShield インターフェイスの端にドラッグすると、固定ウィンドウとして表示されます。[出力] ウィンドウまたはそのタブの 1 つを InstallShield インターフェイスの端から離れた場所にドラッグすると、取り外されます。

詳細については、次を参照してください。

- 様々なビューで、[グループ ボックス] 領域を使って作業する
- [出力] ウィンドウを固定する/取り外す

.NET Framework および Internet Explorer 8 のための定義済みシステム検索

InstallShield に 2 つの新しい定義済みシステム検索が追加されました:

- Microsoft .NET Framework 3.5 SP1
- Internet Explorer 8

インストールでこれらのいずれか、または両方が必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンドユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

この強化は Express プロジェクトに適用します。

InstallShield 2009 Express Edition の新しい機能

新しい機能

InstallShield 2009 Express Edition には、以下のような新しい機能が搭載されています。

InstallShield 前提条件を機能と関連付けて、インストールを連鎖させる機能

InstallShield では、InstallShield 前提条件を 1 つまたは複数の機能と関連付けられるようになりました。この新しい種類の InstallShield 前提条件は、*機能前提条件*と呼ばれます。機能前提条件は、前提条件を含む機能がインストールされたときに、その前提条件がシステム上に既にインストールされていない場合にインストールされます。

プロジェクトに InstallShield 前提条件を含めると、複数のインストールを連鎖することができるため、1 度に 1 つの実行シーケンスのみしか実行できない Windows Installer 制限をバイパスできます。Setup.exe セットアップランチャーは、連鎖を管理するブートストラップ アプリケーションとしての役割を果たします。

[再配布可能ファイル] ビューを使って、InstallShield 前提条件をプロジェクトに追加して、その前提条件をメインインストールの前に実行するのか、メイン インストールに含まれる 1 つまたは複数の機能と関連付けるのかを指定できます。

以前は、すべての InstallShield 前提条件インストールが、メイン インストールの実行前に実行されたため、InstallShield 前提条件を機能に関連付けることはできませんでした。この種類の前提条件は今後も使用できますが、今回より、*セットアップ前提条件*と呼ばれます。

詳しくは、次を参照してください:

- セットアップ前提条件と機能前提条件の違い
- プロジェクトで、InstallShield 前提条件を機能に関連付ける
- InstallShield 前提条件を含むインストールの実行時の動作
- プロジェクトに含まれている InstallShield 前提条件を使って作業する

ビルボードの強化 - Adobe Flash アプリケーション ファイル (.swf) ビルボードと、より多くのイメージ ビルボード ファイル タイプ (.gif、jpg、jpeg)、新しいビルボード スタイル、およびビルボードのプレビュー機能をサポート

InstallShield には新しいビルボード関連の機能が含まれており、インストールのファイル転送段階における外観をより柔軟に制御することができます。

- プロジェクトに Adobe Flash アプリケーション ファイル (.swf) をビルボードとして追加できます。

  Flash アプリケーション ファイルは、ビデオ、動画、音声、インタラクティブ インターフェイス、ゲーム、テキスト、その他の .swf ファイルがサポートするあらゆる要素で構成されます。

- InstallShield では .gif、jpg、および jpeg ファイルをビルボードとして使用できます。以前は、.bmp ファイルしかサポートされていませんでした。
- InstallShield には新しい "ビルボードの種類" 設定が含まれていて、インストールで使用するビルボードのスタイルを指定できます。たとえば、インストールが全画面背景を使用し、ビルボードを前景に、また小さい進行状況ボックスを画面の右下に表示するスタイルがあります。別のスタイルでは、インストールがビルボードを表示する標準サイズのダイアログを表示します。このダイアログの下の部分に、進行状況バーが表示されます。
- InstallShield では、リリースをビルドおよび実行せずに、実行時にビルボードがどのように表示されるのかをプレビューできます。ビルボードをプレビューすると、そのビルボードに現在構成されている背景色、位置、および関連設定を使ったビルボードの外観を確認できます。

InstallShield の [ビルボード] ビューでは、ビルボード ファイルを追加したり、ビルボード関連の設定を構成したり、ビルボードをプレビューしたりできます。

この機能は Express プロジェクトに適用します。

詳しくは、次を参照してください:

- ビルボードを表示する
- ビルボード ファイルの種類
- ビルボードの種類
- Adobe Flash アプリケーション ファイル ビルボードを追加する
- イメージ ビルボードの追加
- リリースをビルドまたは起動せずにビルボードをプレビューする
- ビルボードを含むインストールの実行時の動作

- [ビルボード] ビュー

Windows Installer 4.5 再配布可能ファイル

InstallShield には、Windows Installer 4.5 用に以下の InstallShield 前提条件ファイル (.prq) が含まれています:

- Windows Vista と Server 2008 (x86) 用 Windows Installer 4.5
- Windows Server 2003 SP1 以降 (x86) 用 Windows Installer 4.5
- Windows XP SP2 以降 (x86) 用 Windows Installer 4.5

この機能は Express プロジェクトに適用します。

詳細については、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。

Setup.exe および Update.exe ブートストラッパの Unicode バージョンを作成する機能

今回より、プロジェクトで Setup.exe セットアップランチャーを作成するとき、Unicode バージョンで作成するか、または ANSI バージョンで作成するかを指定することができます。以前、プロジェクトにセットアップランチャーが含まれているとき、常に ANSI バージョンがビルドされていました。Unicode バージョンのビルドはサポートされていませんでした。

Unicode セットアップランチャーは、ターゲット システムで 2 バイト言語のための適切なコード ページが実行されているいないにかかわらず、セットアップランチャーのユーザー インターフェイスで 2 バイト文字を正しく表示することができます。ANSI セットアップランチャーは、ターゲット システムで適切なコード ページが実行されている場合のみ、セットアップランチャー ダイアログで 2 バイト文字を正しく表示します。適切なコード ページが実行されていない場合、これらのダイアログで 2 バイト文字が文字化けして表示されます。

[リリース] ビューでリリースについて表示される Setup.exe タブに新しく追加された "最短初期化時間" 設定を利用して、Unicode を使うか、または ANSI を使うかを指定することができます。すべての新しい Express プロジェクトでは、Unicode がデフォルトとして使用されます。

QuickPatch パッケージについて Update.exe アップデート起動ツールを作成するときも、Unicode バージョンで作成するか、または ANSI バージョンで作成するかを指定することができます。[一般情報] ビューの [ビルドの設定] 領域にある [詳細] タブに新しく追加された "アップデート ランチャの種類" 設定を利用して、Unicode を使うか、ANSI を使うかを指定することができます。すべての新しい QuickPatch プロジェクトでは、Unicode がデフォルトとして使用されます。

Setup.exe ブートストラッパのログファイルを作成する機能

Express プロジェクトで、新しい /debuglog コマンドライン パラメーターが Setup.exe セットアップランチャーに追加されました。このコマンドライン パラメーターを使用して、デバッグ用のログ ファイルを生成することができます。詳細については、「/debuglog」を参照してください。

新しい Microsoft .NET 再配布可能ファイル

InstallShield には、今回新しく Express プロジェクトに追加することができる .NET 関連の InstallShield 前提条件が多数含まれています。

- Microsoft .NET Framework 3.5 (Web ダウンロード版)
- Microsoft .NET Framework 3.5 (完全パッケージ)
- Microsoft .NET Framework 3.5 言語パック
- Microsoft .NET Framework 3.0 SP1 (Web ダウンロード版)

- Microsoft .NET Framework 3.0 言語パック
- Microsoft .NET Framework 2.0 SP1

[再配布可能ファイル] ビューを使って、これらの InstallShield 前提条件をどれでもプロジェクトに追加することができます。

詳細については、次を参照してください。

- .NET Framework 再配布可能ファイルをプロジェクトへ追加する
- Microsoft .NET Framework および Microsoft .NET Framework 言語パックの前提条件を含める

Setup.exe にストリームされるファイルを圧縮し、圧縮レベルを指定できる機能

今回より、Setup.exe セットアップランチャーを使用するリリースをビルドしたとき、InstallShield によって、Setup.exe ファイルにストリームされるファイルが圧縮されるようになりました。InstallShield が使用するデフォルトの圧縮レベルは、ファイルのサイズと実行時に圧縮ファイルを展開するために必要な時間のバランスをとっての目安です。圧縮レベルを変更する場合、または圧縮をしない場合、マシン全体に適用する設定を利用してデフォルトのレベルをオーバーライドすることができます。

デフォルトで、ビルド時に InstallShield がファイルを Setup.exe ファイルにストリームするとき、.cab ファイル拡張子を持つファイルは圧縮されません。これは、.cab ファイルが既に圧縮されているファイルであるためです。デフォルトの圧縮除外一覧を変更して、他の種類のファイルや特定のファイルを必要に応じて選択することができます。除外一覧は、マシン全体に適用される設定です。

詳細については、「Setup.exe にストリームされるファイルの圧縮レベルを構成する」を参照してください。

マルチパート .cab ファイルのサポート

.cab ファイルには、いくつかの制限事項があります。たとえば、単一 .cab ファイルの最大サイズは 2 GB に設定されています。また、サイズの大きい .cab ファイルを署名しようしたとき、およびサイズの大きい署名済み .cab ファイルのデジタル署名を検証しようとしたときにトラブルが生じた経験があるユーザーもいるかもしれません。今回より、これらの制限事項を回避するために、.cab ファイルのデフォルトの制限が 600 MB に設定されました。InstallShield で、リリースの .cab ファイルを作成しているとき、この制限に達すると、データが 2 つ以上の .cab ファイルに分割され、マルチパートの .cab ファイルが作成されます。

最大サイズは必要に応じて変更することができます。InstallShield でマルチパート .cab ファイルを作成しない場合、単一の .cab ファイルを作成するように構成できます。

この機能は Express プロジェクトに適用します。また、この機能は、すべてのファイルが単一ファイルの .msi パッケージまたは Setup.exe セットアップランチャーに埋め込まれている圧縮済み SingleImage リリースをビルドしている場合のみ適用します。

詳細については、「.cab ファイルの最大サイズを構成する」を参照してください。

Windows Server 2008 上に IIS 7 Web サイトをインストールするためのサポート

InstallShield では、今回より、Windows Server 2008 システム上で IIS 7 Web サイトと仮想ディレクトリの作成ならびに管理ができるようになりました。この機能は、Express プロジェクトで提供されています。

追加された Microsoft SQL Server 2005 Express SP2 前提条件

InstallShield に Microsoft SQL Server 2005 Express Edition SP2 の InstallShield 前提条件が追加されました。この InstallShield 前提条件は、Express プロジェクトに追加することができます。

Microsoft Visual Studio 2008 サポート

InstallShield が Visual Studio 2008 と統合され、インストールと製品の開発を同じ Visual Studio インターフェイス内で行えるようになりました。

Visual Studio セットアップ プロジェクトを InstallShield プロジェクトに変換できる機能

InstallShield を使って、今回より、Visual Studio 2008、Visual Studio 2005、Visual Studio .NET 2003、または Visual Studio .NET セットアップ プロジェクト (.vdproj) を Express プロジェクト (.ise) に変換できます。

詳細については、「Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートする」を参照してください。

新しい FlexNet Connect 11 再配布可能ファイル

InstallShield は、FlexNet Connect 11 をサポートします。InstallShield の [アップデート通知] ビューを使って、2 つの FlexNet Connect 11 マージ モジュールうち、いずれかが含まれています (Common Software Manager が含まれているマージ モジュールと、含まれていないマージ モジュール)。

強化機能

InstallShield 2009 Express Edition には、次のような新しい強化機能が含まれています。

ベスト プラクティス ダイナミック ファイル リンク

プロジェクトにダイナミック ファイル リンクを追加、または変更するとき、InstallShield がコンポーネントを作成する方法について、新しいベスト プラクティスを使用するか、これまでと同様にディレクトリごとに 1 つのコンポーネント作成するのかを指定できるようになりました。

コンポーネントは、個別にインストール可能な最小の製品構成単位です。InstallShield Express Edition では、コンポーネントが自動で作成されます。

コンポーネント作成のベスト プラクティスに従うと、ダイナミック リンクを持つフォルダーにある各ポータブル実行可能ファイル (PE) ファイルにコンポーネントが別々に作成されます。このコンポーネント作成メソッドを、改良された QuickPatch パッケージの機能と併用することで、Windows Installer のコンポーネント規則に準拠するパッチを作成することができます。

以前、ダイナミック ファイル リンクをプロジェクトに追加すると、ビルド時に、ダイナミック リンクを持つすべてのファイルについてコンポーネントが 1 つ自動的に作成されていました。ただし、ダイナミック ファイル リンクに PE ファイルが含まれているとき、コンポーネントの作成時に Windows Installer ベスト プラクティスが実行されませんでした。

デフォルトで、InstallShield は .exe、.dll、.ocx、.vxd、.chm、.hlp、.tlb、および .ax を PE ファイルとして認識します。[オプション] ダイアログ ボックスにある [ファイルの拡張子] タブで、この一覧を変更することができます。

[ベスト プラクティス] ダイナミック ファイル リンクは、Express プロジェクトに適用します。

詳しくは、次を参照してください:

- ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する
- [ファイルの拡張子] タブ

仮想ディレクトリを使わずに IIS Web サイトをインストールできる機能

InstallShield は、今回より、仮想ディレクトリを一切含まない IIS Web サイトのインストールをサポートできるようになりました。このサポートは、InstallShield で作成されたすべての新しい Web サイトに有効です。

以前、InstallShield では、仮想ディレクトリのない Web サイトのインストールはサポートされていませんでした。そのため、インストールに含まれる Web サイトが仮想ディレクトリを持たない場合、実行時 Web サイトが作成されませんでした。

InstallShield 2008 Express Edition プロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、プロジェクトに既に Web サイトが含まれている場合、仮想ディレクトリのない Web サイトはインストールできません。仮想ディレクトリのない Web サイトのインストールできるようにするには、それを手動で InstallShield 2021 Express Edition プロジェクトから削除し、新しい Web サイトとして再度プロジェクトに追加する必要があります。

この強化は Express プロジェクトに適用します。

詳細については、「Web サイトの作成とアプリケーションまたは仮想ディレクトリの追加」を参照してください。

QuickPatch パッケージの簡素化

今回より、一般的に以前の InstallShield でビルドされたパッケージに比べて新しいサブ機能とビルトイン InstallShield カスタム アクションの数が少ない、簡素化された QuickPatch パッケージのビルドが可能となりました。QuickPatch プロジェクトの [詳細] タブに追加された "QuickPatch の簡素化" 設定で、この新しいタイプの QuickPatch パッケージを作成するかどうかを指定できます。

詳細については、次を参照してください。

- QuickPatch パッケージを簡素化するかどうかを指定する
- [詳細] タブ

パッチと QuickPatch パッケージをパスワードで保護する機能

今回より、QuickPatch パッケージをパスワードで保護するためのパスワード設定が追加されました。これらの設定は、QuickPatch プロジェクトの [詳細] タブ上にあります。

QuickPatch パッケージをパスワードで保護すると、すべてのエンド ユーザーはパッケージをインストールする時に、アップデートを起動するためのパスワード (大文字と小文字を区別する) を入力しなくてはなりません。

詳しくは、次を参照してください:

- QuickPatch パッケージをパスワードで保護する
- [詳細] タブ

/v コマンドライン パラメーターを複数回使用して、Setup.exe から .msi ファイルに複数のパラメーターを渡すことができる機能

Setup.exe から Msiexec.exe へ複数の引数を渡す場合、コマンドラインで /v オプションを各引数ごとに 1 回ずつ、複数回使用することができます。以前、/v オプションの使用は 1 回に限られていたため、すべてのパラメーターはこのインスタンスを通して渡されていました。

この強化は Express プロジェクトに適用します。

詳細については、「Setup.exe」を参照してください。

.NET Framework 用の定義済みシステム検索

InstallShield に、次の新しい定義済みシステム検索が追加されました。

- Microsoft .NET Framework 3.5
- Microsoft .NET Framework 3.0 SP1
- Microsoft .NET Framework 3.0
- Microsoft .NET Framework 2.0 SP1
- Microsoft .NET Framework 2.0
- Microsoft .NET Framework 1.1
- Microsoft .NET Framework 1.0

インストールでこれらのいずれかが必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

この強化は Express プロジェクトに適用します。

InstallShield 2008 Express Edition の新しい機能

新しい機能

InstallShield 2008 Express Edition には、以下のような新しい機能が搭載されています。

Express プロジェクト用の新しいエンド ユーザー ダイアログ テーマ

ダイアログ テーマは、エンドユーザー ダイアログに統一感のとれた個性的な印象を与えることができる、あらかじめ定義されている 1 セットのイメージです。[ダイアログ] ビューの新しい "グローバル ダイアログ テーマ" 設定で選択したテーマ オプションを変更して、プロジェクトに提供されているテーマから任意のテーマを選択し、プロジェクトで使用されているすべての内部および外部ダイアログ (Setup.exe 初期化ダイアログを含む) に適用することができます。

この機能に関する詳細は、次を参照してください:

- ダイアログのテーマ
- ダイアログ テーマの選択または変更
- すべてのエンドユーザー ダイアログのためのグローバル ダイアログ設定

デジタル署名の強化機能

ビルド時に、製品の実行可能ファイルを含むインストール内のすべてのファイルにデジタル署名が可能になりました。また、今回より、デジタル署名に personal information exchange ファイル (.pfx) が使用できるようになりました。Express プロジェクト タイプでは、この機能がサポートされています。

[リリース] ビューで新しい [署名] タブを使って、InstallShield がファイルに署名するときに使用するデジタル署名に関する情報 (証明機関より付与されたデジタル証明書ファイルを含む) を指定します。[署名] タブでまた、デジタル署名をするインストール内のファイルを指定することもできます。

署名に .pfx ファイルを指定すると、InstallShield でファイルが署名されるとき SignTool.exe が使用されます。.spc ファイルと .pvk ファイルを指定すると、ファイルの署名に Signcode.exe が使用されます。.pfx ファイルは、より多くの異なる環境 (ロックされたビルド マシンなど) で動作するため、より頻繁に利用されています。InstallShield でデジタル署名パスワードを指定するとき、.pfx ファイルを使用している場合、パスワードのプロンプトは表示されなくなります。.spc ファイルと .pvk ファイルを使用している場合は、パスワードのプロンプトが表示されることがあります。

以前、InstallShield では、.msi ファイルと Setup.exe ファイルのみ署名が可能でした。また、デジタル署名として指定できるのは .spc ファイルと .pvk ファイルのみで、.pfx ファイルを指定することはできませんでした。

詳しくは、次を参照してください:

- デジタル署名とセキュリティ
- ビルド時にリリースとそのファイルにデジタル署名を行う
- [署名] タブ

インターネット インフォメーション サービス (IIS) 7.0 と SSL のサポート

InstallShield には今回より IIS 7 のサポートも含まれています。

また、インストールに、Web サイトの SSL 証明書を含めることもできます。SSL サーバー証明書を含めることにより、ユーザーは Web サーバーの認証および Web コンテンツの有効性の確認を行うことができると共に、セキュリティで保護された接続を確立することができます。

詳細については、次を参照してください。

- ≫ [IIS 構成] ビュー
- Web サイトの SSL 証明書を指定する
- InstallShield における IIS サポートのバージョン固有情報

新しい Microsoft .NET Framework 3.0 前提条件条件

InstallShield には今回、Express プロジェクトに追加することができる .NET Framework 3.0 前提条件が含まれています。

詳細については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。

Visual C++ 8.0 マージ モジュールの追加

InstallShield に、Visual C++ 8.0 SP1 のマージ モジュール (バージョン 8.0.50727.762) が追加されました。

ダイアログ ボタンの UAC シールド アイコンのサポート

インストールが Windows Vista システム上で実行されているとき、インストールが昇格された権限でまだ実行されていない場合、ReadyToInstall ダイアログの [インストール] ボタンと ReadyToRemove ダイアログの [削除] ボタンに [ユーザー アカウント制御 (UAC)] シールド アイコンが表示されます。

InstallShield は昇格された権限で実行されます。このため、Windows Vista システムでインストールを InstallShield 内から起動したとき、インストールは昇格された権限で実行され、[ユーザー アカウント制御 (UAC)] シールド アイコンは ReadyToInstall と ReadyToRemove ダイアログの [インストール] ボタンに表示されません。

UAC シールド アイコンがあるダイアログのサンプル スクリーンは、「[インストールの準備完了] ダイアログ」をご覧ください。

追加された SQL Server 2005 Express Edition SP1 セットアップ前提条件

InstallShield に Microsoft SQL Server 2005 Express Edition SP1 のセットアップ前提条件が追加されました。このセットアップ前提条件は、Express プロジェクトに追加することができます。

更新された DirectX 9.0c オブジェクト

DirectX 9.0c オブジェクトは今回より、すべての最新の DirectX 9.0c コアとオプションのコンポーネントをインストールします。

また、DirectX 9 オブジェクト ウィザードにも一部変更が加えられました。このウィザードでは今回より、再配布可能ファイルを Disk1 フォルダーに含めるか、または .msi ファイルにストリームするかを指定することができます。この変更により、圧縮インストールで DirectX 9 オブジェクトが使用できるようになりました。また、サイレント インストールでも今回より DirectX 9 オブジェクトが使用できるようになりました。

DirectX インストールを起動するカスタム アクションは今回より、Windows Vista システムで昇格された権限を使って実行できるように、[実行] シーケンスにスケジュールされ、遅延システム コンテキストで実行されます。

詳しくは、次を参照してください:

- DirectX 9.0 オブジェクトを含める
- DirectX オブジェクト ウィザード

Windows Server 2008 システムをターゲットする機能

InstallShield では、Windows Server 2008 がインストールに必須であると指定することができます。また、機能およびコンポーネントに Windows Server 2008 に関連する条件をビルドすることもできます。

新しい MSXML 6 SP1 セットアップ前提条件

InstallShield に、Express プロジェクトに追加することができる新しい MSXML 6.0 SP1 セットアップ前提条件が追加されました。

FlexNet Connect サポート

Express プロジェクトに FlexNet Connect 6.1 または 5.x の再配布可能ファイルを追加することができます。[アップデート通知] ビューで、プロジェクトに含める FlexNet Connect のバージョンを選択することができます。バージョン 6.1、または [オプション] ダイアログ ボックスの [マージ モジュール] タブにある [マージ モジュールの場所] 領域で指定されている場所にインストールされている任意のレガシー バージョンを含めることができます。

[アップデート通知] ビューに、FlexNet Connect 6.1 がサポートする新しい "ベンダー データベース" 設定が追加されました。

強化機能

InstallShield 2008 Express Edition には、次のような新しい強化機能が含まれています。

リリースにおけるユーザビリティの強化点

[リリース] ビュー (旧 [リリースのビルド] ビュー) のリリースの設定が、カテゴリ別に複数のタブで再構成されました。

[リリースの配布] ビューにあった設定は、[リリース] ビューの新しい [ポストビルド] タブに移されました。[ポストビルド] タブには、ビルド時にリリースをフォルダーまたは FTP サイトに自動的に配布できるように構成できる設定があります。

[リリース] ビューでリリースをクリックしたときに表示されるコマンドに、新しい [配布] コマンドが追加されました。このコマンドを選択すると、リリースに関連するすべてのファイルが [ポストビルド] タブで指定された場所にコピーされます。

詳しくは、次を参照してください:

- [ビルド] タブ
- Setup.exe タブ
- [署名] タブ
- [.NET/J#] タブ
- [インターネット] タブ
- [イベント] タブ
- インストールのテストと実行
- フォルダーまたは FTP サイトにリリースを自動的に配布する

[ファイル] ビュー、[レジストリ] ビュー、および [再配布可能ファイル] ビューにおけるユーザビリティの強化

[ファイル] ビューにおける強化内容は次のとおりです:

- **[インストール先コンピューターのファイル]** ペイン内でファイルを右クリックしてから、新しい [1 つ上のフォルダーを開く] コマンドをクリックできます。Windows エクスプローラー ウィンドウが開き、右クリックして選択したファイルを含むフォルダーが表示されます。
- **[インストール先のコンピューターのファイル]** ペインを右クリックすると、新しい [追加] ボタンを使用できます。このコマンドを使うと [開く] ダイアログ ボックスが表示され、プロジェクトに追加するファイルを参照することができます。
- このビューの右上に、新しいリンク ([ソース ペインの表示] または [ソース ペインの非表示]) が追加されました。この新しいリンクを使うと、このビューの上部に表示される **[ソース コンピューターのフォルダー]** ペインおよび **[ソース コンピューターのファイル]** ペインを表示または非表示に切り替えることができます。この 2 つのペインを非表示にして Windows エクスプローラー ウィンドウを開き、InstallShield 内に表示されている残りの 2 つのペインに Windows エクスプローラー ウィンドウから直接ファイルをドラッグ アンド ドロップすることができます。

[レジストリ] ビューの右上にも新しいリンク ([ソース ペインの表示] または [ソース ペインの非表示]) が追加されました。この新しいリンクを使うと、このビューの上部に表示される **[ソース コンピューターのフォルダー]** ペインおよび **[ソース コンピューターのファイル]** ペインを表示または非表示に切り替えることができます。

また、[再配布可能ファイル] ビューに 2 つの強化点が加えられました:

- このビューの右側のペインに、左上のペインで選択されたマージ モジュール、オブジェクト、またはセットアップ前提条件に関する詳細が表示されます。このビューの右上にある [詳細の表示] または [詳細の非表示] リンクをクリックすると、この詳細ペインを表示または非表示に切り替えることができます。
- セットアップ前提条件の [詳細] ペインに、選択されたセットアップ前提条件に関する完全な情報が表示されます。この情報には、前提条件に構成されている条件、コマンドライン パラメーター、およびその他の情報が含まれます。

[ショートカット/フォルダー] ビューにおける機能強化

[ショートカット/フォルダー] ビューの一部が強化されました。

- ショートカットに使用されるアイコンを変更するには、そのショートカットを右クリックして、新しい [**ショートカット アイコンの変更**] コマンドをクリックします。[アイコンの変更] ダイアログ ボックスが開き、ショートカットが実行時にターゲット システムで作成されるときに使用されるアイコン ファイルと関連付けられたアイコン インデックスを選択することができます。
- [ショートカット] エクスプローラーに一覧表示されるショートカットは、ターゲット システムで使用されるアイコン イメージと共に表示されます。以前、[ショートカット] エクスプローラーでは、アイコンがショートカットに指定されていても、すべての種類のショートカットに異なるイメージが使用されていました。

強化内容についての詳細は、次を参照してください:

- ショートカットのアイコンを指定する
- [ショートカット/フォルダー] ビュー

セットアップ前提条件の機能強化

選択したリリースについて、Express プロジェクトのセットアップ前提条件をどこに配置するかを指定できる ″セットアップ前提条件の場所″ 設定が [リリース] ビューの Setup.exe タブに追加されました。

デフォルト値は、[個々の選択に従う] です。このオプションでは、[再配布可能ファイル] ビューで個々の前提条件について指定された場所が使用されます。

この他に、[Web からダウンロードする]、[Setup.exe から抽出する]、[ソース メディアからコピーする] というオプションがあります。これらの 3 つのオプションは、[再配布可能ファイル] ビューで各セットアップ前提条件のプロパティについて指定された場所をオーバーライドします。

詳細については、次を参照してください。

- 特定の InstallShield 前提条件の実行時の場所を指定する
- リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する
- Setup.exe タブ

SecureCustomProperties プロパティの強化されたサポート

[実行] シーケンスで昇格された権限が必要なインストールの [ユーザー インターフェイス] シーケンスにパブリック プロパティが設定されている場合、そのプロパティの値を [実行] シーケンスに渡すためには、プロパティが SecureCustomProperties プロパティの値としてリストされているか、または制限付きパブリック プロパティである必要があります。

InstallShield は今回より、場合によって [ユーザー インターフェイス] シーケンスから [実行] シーケンスに渡す必要があるプロパティを SecureCustomProperties プロパティに自動的に追加します。詳細については、「パブリック プロパティが制限付きパブリック プロパティである必要があることを指定する」を参照してください。

Express プロジェクトにおけるダウングレードの自動防止

エンドユーザーが現在のバージョンの製品をインストールしたとき、同製品の将来のメジャー バージョンを上書きできないようにするために、現在のインストールが将来のメジャー バージョンを上書きするのを防ぐサポートが自動的に追加されます。詳細については、「現在のインストールによる同製品の将来のメジャー バージョンの上書きを防ぐ」を参照してください。

ALLUSERS と [ユーザー情報] ダイアログの変更

InstallShield 2008 Express Edition より、すべての新規 Express プロジェクトでは、デフォルトで ALLUSERS プロパティが 1 に設定されています。ほとんどのインストールは、マシンごとに管理者権限を使用して実行される必要があるため、これが推奨される実装です。

InstallShield 12 Express Edition 以前で作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、ALLUSERS プロパティの値は自動的に変更されません。また、このプロパティが以前のプロジェクトで定義されていない場合も、自動的に追加されません。[一般情報] ビューに、ALLUSERS の値を設定できる新しい ALLUSERS 設定が追加されました。

また、InstallShield 2008 Express Edition から、デフォルトで、すべての Express プロジェクトの [ユーザー情報] ダイアログは、エンドユーザーが製品をすべてのユーザーにインストールするか、または現在のユーザーのみにインストールするかを指定できるラジオ ボタンを表示しないようになっています。このダイアログについては、これが推奨される実装です。

InstallShield 12 Express Edition 以前で作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、[ユーザー情報] ダイアログは自動的に変更されません。[ダイアログ] ビューを利用して、このダイアログのラジオ グループ ボタンの表示と非表示を切り替えることができます。

詳しくは、次を参照してください:

- ユーザーごとのインストールとマシンごとのインストールの違い
- 一般情報の設定
- ALLUSERS
- [ユーザー情報] ダイアログ

コマンドラインまたは MSBuild タスク パラメーターで製品バージョンを変更する機能

-y コマンドライン パラメーターが、IsCmdBld.exe を使用したコマンドライン ビルド用に追加されました。このパラメーターを使用して、コマンドライン ビルドから製品のバージョンを指定することができます。

また、MSBuild の InstallShield タスクに、MSBuild で製品バージョンを指定することができる ProductVersion パラメーターが追加されました。このプロパティは、デフォルトのターゲット ファイルが使用されたとき、プロパティ InstallShieldProductVersion として露出されます。

-y コマンドライン パラメーターまたは InstallShield タスク ProductVersion パラメーターは、製品バージョンのビルド バージョン (3 番目のフィールド) を増加するとき、特に便利です。

詳しくは、次を参照してください:

- IsCmdBld.exe
- Microsoft ビルド エンジン (MSBuild)

IIS Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定するための新しい設定

IIS Web サーバーを構成して、#exec ディレクティブの CMD コマンドがシェル コマンドの実行に使用されるのを防いだり、CMD コマンドがこのタイプのコマンドの実行に使用されることを許可することができます。HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥W3SVC¥Parameters レジストリ キーの SSIEnableCmdDirective レジストリ値によって、CMD コマンドが許可されているかどうかが判別されます。

InstallShield の [IIS 構成] ビューに、新しい "SSIEnableCmdDirective レジストリ値" 設定が追加されました。この設定を使って、インストールがターゲット システム上で SSIEnableCmdDirective レジストリ値をどのように構成するのかを指定することができます。また、SSIEnableCmdDirective レジストリ値を実行時に変更しないように指定することもできます (デフォルト動作)。

詳細については、「Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定する」を参照してください。

IIS Web サイト用の新しい "ホスト ヘッダー名" 設定

[IIS 構成] ビュー内の Web サイトについて、[Web サイト] タブに追加された新しい "ホスト ヘッダー名" 設定を使って、インストール中に追加された IIS Web サイトを識別するホスト ヘッダー名を指定することができます。

詳細については、「Web サイトの IIS ホスト ヘッダー名を指定する」を参照してください。

.msi ファイルから参照されていないディレクトリを削除する機能

[リリース] ビューの [ビルド] タブに、新しい "未使用のディレクトリを保持する" 設定が追加されました。この設定を使って、選択されたリリースをビルドするときに、InstallShield が .msi ファイルの Directory テーブルから未使用のディレクトリを削除するかどうかを指定することができます。デフォルト値は [いいえ] です。

この設定は、Express プロジェクトで提供されています。

詳細については、「[ビルド] タブ」を参照してください。

InstallFinalize アクションの後 COM+ アプリケーションをインストールするかどうかを指定できる新しいチェック ボックス

[コンポーネント サービス] ビューの [インストール] タブ に新しい **[InstallFinalize アクションの後でインストールする]** チェック ボックスが追加されました。プロジェクト内の選択された COM+ アプリケーションに、グローバル アセンブリ キャッシュ (GAC) にインストールする必要がある .NET アセンブリが含まれている場合、このチェック ボックスを選択します。このチェック ボックスを選択すると、ISComponentServiceFinalize アクションは選択した COM+ アプリケーションを InstallFinalize アクションの後でインストールします。Windows Installer は、InstallFinalize が実行されるまでスクリプト内のセッションで加えられた変更をコミットしません。

Express プロジェクトに追加された定義済みシステム検索

InstallShield に、次の新しい定義済みシステム検索が追加されました。

- Adobe Reader 7
- Adobe Reader 6
- Internet Explorer 7.0

インストールでこれらの製品のいずれかが必要な場合、[要件] ビューまたはプロジェクト アシスタントの [インストール要件] ページを使って、これらのシステム検索をプロジェクトに追加することができます。エンド ユーザーがインストールを起動すると、Windows Installer はターゲット システムの要件が満たされているかどうかを確認します。要件が満たされていない場合、インストールでシステム検索用に定義されているエラー メッセージが表示されます。

パッチ表示情報における強化

[識別] タブ (旧名 [アンインストール] タブ) では、Windows Installer 3.0 以降を実行しているシステムの [プログラムの追加と削除] で QuickPatch パッケージについて表示される情報を指定することができます。QuickPatch プロジェクトの [一般情報] ビューにあるこのタブには、表示名、製造元名、サポート URL などのアイテムについ

ての設定があります。QuickPatch プロジェクトで最新のセットアップを変更するたびに、最新のセットアップからの [プログラムの追加と削除] 情報が [識別] タブにある設定の値として使用されます。必要に応じて、[識別] タブの値をオーバーライドすることもできます。また、**[パッチのアンインストールを許可する** (Windows Installer 3.0 が必要)] チェック ボックスも今回 [共通] タブに追加されました。この設定は以前、[アンインストール] タブで提供されていました。

詳細については、次を参照してください。

- [共通] タブ (QuickPatch プロジェクト)
- [識別] タブ (QuickPatch プロジェクト)

最短初期化時間の指定機能

[リリース] ビューにあるリリースについての Setup.exe タブに "最短初期化時間" 設定が新しく追加されました。この設定を利用して、エンドユーザーがこのリリースを実行した時に、インストールが初期化ダイアログ (およびスプラッシュ画面) を表示する最短時間 (秒) を指定できます。

詳細については、「Setup.exe タブ」を参照してください。

InstallShield 12 Express Edition の新しい機能

InstallShieldには、以下のような新しい機能が搭載されています。

Windows Vista システムをターゲットする機能

InstallShield では、Windows Vista がインストールに必須であると指定することができます。また、機能に Windows Vista に関連する条件をビルドすることもできます。

ユーザー アカウント制御のサポート

InstallShield では、Microsoft が Windows Vista に追加したユーザー アカウント制御機能がサポートされています。[一般情報] ビューに新しく追加された "管理者権限" 設定を使って、インストールが管理者権限を必要とするかどうかをプロジェクト全体に対して指定することができます。また、[リリースのビルド] ビューの "必要実行レベル" 設定では、Setup.exe ファイルが Windows Vista プラットフォーム上でインストール (セットアップランチャー、すべてのセットアップ前提条件、および .msi ファイル) を実行するときに必要な最小特権レベルを指定できます。

詳細については、次を参照してください:

- 一般情報の設定
- Windows Vista 以降のプラットフォームでのセットアップランチャーの必要実行レベルを指定する

デジタル署名の強化

インストールにデジタル署名情報を指定した場合、InstallShield は自動的に MsiDigitalCertificate および MsiPatchCertificate テーブルへ必要な情報を追加します。**MsiPatchCertificate** テーブルには、ユーザー アカウント制御 (UAC) のパッチを有効化するために必要な情報が含まれています。これにより、管理者以外のユーザーが適用することができる QuickPatch を作成することができます。

また、プロジェクト アシスタントの [インストールのビルド] ページで、インストールにデジタル署名情報を指定できるようになりました。また、[リリースのビルド] ビューですべてのメディア タイプに対してデジタル署名情報を指定できるようになりました。以前は、WebDeployment メディア タイプのみにデジタル署名情報を指定することができました。

詳細については、次を参照してください。

- 非管理者パッチのインストールを準備する
- デジタル署名とセキュリティ

再起動マネージャー インフラストラクチャによる再起動回数の最小化をサポート

インストール終了後のシステム再起動は、エンドユーザーにとって不都合なものです。Certified for Windows Vista ロゴ プログラムの要件の 1 つに、エンドユーザーがインストール完了後自動的にアプリケーションを閉じて再起動を行うことができるオプションを含まなくてはならないという項目があります。

この品質ガイドラインをサポートするため、すべて Express プロジェクトで MsiRMFilesInUse ダイアログが提供されています。インストール中に更新が必要なファイルが他のアプリケーションによって使用中の場合、インストールでこのダイアログが表示されます。詳しくは、次を参照してください:

- Windows Vista 以降のシステムの再起動を最小限にする
- MsiRMFilesInUse ダイアログ

プロジェクト全体における Windows Installer 4.0 ログ ファイルのサポート

InstallShield では、Windows Installer 4.0 が実装されるとき、インストールのログ記録を行うかどうかをプロジェクト全体に指定することができます。このため、コマンドラインを使用したり、レジストリを使ってログ パラメーターの設定したりする必要がありません。また、ログ記録されるメッセージの種類をカスタマイズすることもできます。

ログ記録を有効にするには、[一般情報] ビューの新しい "MSI ログの作成" 設定を使用します。この設定の省略記号ボタンボタンをクリックすると、[Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックスが表示されます。このダイアログ ボックスで、ログ記録を行うかどうか指定します。また、ログ記録されるメッセージの種類をカスタマイズするとき、このダイアログ ボックスで、デフォルトのログ パラメーターをオーバーライドします。

ログ記録を有効にすると、Windows Installer 4.0 によりログ ファイルが製品のインストール中に作成され、MsiLogFileLocation プロパティにログ ファイルのパスが書き込まれます。また、[セットアップの完了]、[セットアップの異常終了]、および [セットアップの中断] ダイアログに **[Windows Installer ログの表示]** チェック ボックスが追加されました。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。

詳細については、「Windows Installer インストールをログ記録するかどうかを指定する」を参照してください。

複数言語ユーザー インターフェイス (MUI) のサポート

Windows Installer 4.0 によって実行される複数言語アプリケーション用のインストールを作成するとき、InstallShield を使って Windows 複数言語ユーザー インターフェイス (MUI) のサポートを含むショートカットを作成することができます。[ショートカット/フォルダー] ビューでは、選択されたショートカットに対して、次の 4 つの新しい設定が使用できるようになりました:

- 表示リソース DLL
- 表示リソース ID

- 説明リソース DLL
- 説明リソース ID

これらの新しい設定は、Windows Installer 4.0 の Shortcut テーブルにある新しい 4 つの列に対応しています。詳細については、「ショートカットの設定」を参照してください。

COM 抽出と依存関係スキャナーにおけるレジストリおよびファイルのフィルター機能の強化

InstallShield と共にインストールされている新しい Filters.xml を編集して、COM サーバーから必要のない COM データが抽出されるのを防ぐことができます。この Filters.xml ファイルを編集することにより、COM 抽出から除外されるレジストリ キーの一覧をカスタマイズすることができます。

Filters.xml ファイルに、スタティック依存関係スキャン、ダイナミック依存関係スキャン、および Visual Basic 依存関係スキャンで含めるまたは除外するファイルがリストされます。以前は、2 つの異なるファイル (Userscan.ini と Iswiscan.ini) が除外および選択の対象になるファイルをリストするために使用されていました。

詳細については、次を参照してください。

- COM 抽出のレジストリ変更をフィルターする
- 依存関係スキャナーでファイルをフィルターする

強化されたスタート ページ

スタート ページで表示される最近開いたプロジェクトの一覧に、プロジェクトの種類を表示する列が挿入されました。また、一覧に表示されるプロジェクトの最大数が、4 つから 8 つに増えました。

アップグレードとアセンブリに関する Windows Installer の問題に対するワークアラウンド

デフォルトで、[アップグレード パス] ビューで作成されたアップグレードは、製品の新しいバージョンをインストールする前に古いバージョンを削除するように構成されます。言い換えると、RemoveExistingProducts アクションが InstallFinalize アクションの前にスケジュールされています。製品がグローバル アセンブリ キャッシュ (GAC) にインストールされているアセンブリを含んでいる場合、このデフォルト シーケンス動作により、アップグレードが適用された後にアセンブリが GAC から消えてしまうという問題が発生する場合があります。この問題は、Windows Installer の参照カウントがアセンブリに対して適切に実行されないという問題に起因しており、結果的に、アップグレードでアセンブリが削除された後再インストールされないという問題につながっています。

この Windows Installer の問題を回避するために、InstallShield では今回より、新しいバージョンの製品が古いバージョンが削除される前にインストールされるようにプロジェクトを構成することができます。

詳細については、「アップグレード中にグローバル アセンブリ キャッシュからアセンブリが削除されるのを防ぐ方法」を参照してください。

ターゲット システムの要件

InstallShield を使って、Windows ベースのシステムをターゲットにするインストールを素早くビルド、テストおよび配布することができます。

デスクトップ コンピューターの要件

オペレーティング システム

ターゲット システムは、次のオペレーティング システムの最小要件を満たさなくてはなりません:

- Windows XP SP3
- Windows Server 2003 SP2
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2
- Windows 8
- Windows Server 2012
- Windows 8.1
- Windows Server 2012 R2
- Windows 10
- windows 11
- Windows Server 2022

ターゲット システムで、SSE2 インストラクション セットがサポートされていることが必須です。

インストーラー エンジンの要件

各インストーラー エンジンの最小ターゲット システム要件は、以下のとおりです。

テーブル 1-2 ・ デスクトップ コンピューターのターゲット システム要件

インストーラー エンジン	オペレーティング システム、その他の要件
Windows Installer 5.0 **(これは再配布可能ファイルとしては利用できません。)**	Windows 7 以降、Windows Server 2008 R2 以降
Windows Installer 4.5	Windows XP SP2 以降、Windows Server 2003 SP1 以降
Windows Installer 4.0 **(これは再配布可能ファイルとしては利用できません。)**	Windows Vista、Windows Server 2008
Windows Installer 3.1	Windows 2000 SP3 以降、Windows Server 2003 以降
Windows Installer 3.0	Windows 2000 SP3 以降、Windows Server 2003 以降

テーブル 1-2・デスクトップ コンピューターのターゲット システム要件

インストーラー エンジン	オペレーティング システム、その他の要件
Windows Installer 2.0	Windows 95 以降 (InstallShield は Windows 95、Windows 98、Windows NT 4、または Windows Me のサポートを含みません。)

32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて

製品が 32 ビットと 64 ビットのターゲット システム両方をサポートする場合、製品のインストールを作成するとき、以下の基本的な Windows Installer の競合についてご注意ください。

- 32 ビット Windows Installer パッケージで、64 ビット ファイルまたはレジストリの場所にインストールすることはできません。
- 64 ビット Windows Installer パッケージを 32 ビット システムで実行することはできません。

Windows Installer は、32 ビット システム上で 32 ビットの場所、および 64 ビット システム上で 64 ビットの場所にインストールする、マルチアーキテクチャの Windows Installer パッケージの作成をサポートしません。

32 ビットと 64 ビット サポートとの重要な相違点

32 ビットと 64 ビット Windows Installer パッケージの違い

Template Summary プロパティが Intel であるとき、Windows Installer パッケージは、32 ビットとして取り扱われます。Template Summary プロパティが x64 である場合、Windows Installer パッケージは 64 ビットとして取り扱われます。32 ビット Windows Installer パッケージは、32 ビット システムと、ほとんどの 64 ビット システム上で実行します。ただし、64 ビット Windows Installer パッケージは、64 ビット システム上でのみ実行します。64 ビット Windows Installer パッケージを 32 ビット システム上で実行すると、ランタイム エラー 1633 が発生します:

「このインストール パッケージはこの種類のプロセッサでサポートされていません。プロダクト ベンダーに問い合わせてください。」

InstallShield Express Edition は、Template Summary プロパティを自動的に構成します。

32 ビットと 64 ビット コンポーネントの違い

コンポーネントは、個別にインストール可能な最小の製品構成単位です。コンポーネントには、ターゲット システムにインストールされるファイル、レジストリ エントリ、その他のデータが含まれます。(詳細については、「インストールの基本」を参照してください。InstallShield Express Edition は、ビルド時に Windows Installer パッケージの各コンポーネントを自動的に作成し、適切な場合は、それぞれのコンポーネントを 64 ビットとしてマークします。

32 ビット Windows Installer パッケージに 64 ビット コンポーネントを含めることはできませんが、64 ビット Windows Installer パッケージには 32 ビット コンポーネントと 64 ビット コンポーネントのどちらか、あるいはその両方を混在させることができます。

Windows Installer が 64 ビット システム上で 64 ビットの場所にインストールする場合、64 ビットの場所用のデータを含む 64 ビット Windows Installer パッケージのコンポーネントを 64 ビットとしてマークする必要があります。64 ビット Windows Installer パッケージのコンポーネントが 64 ビットとしてマークされていない場合、そのデータは 32 ビットの場所にインストールされます。

32 ビット Windows Installer パッケージの 32 ビット コンポーネントは、32 ビットおよび 64 ビット システム上で、32 ビットの場所にインストールされます。64 ビット Windows Installer パッケージの 32 ビット コンポーネントは、64 ビット システム上で 32 ビットの場所にインストールされます。

32 ビットと 64 ビット ファイルの場所の違い

Microsoft では、既存の 32 ビットのアプリケーションがシームレスに動作する 64 バージョンの Windows がデザインされています。また、同一コードが再コンパイルされたバージョンが 64 ビット アプリケーションとしてシームレスに動作する 64 ビット バージョンの Windows もデザインされています。これをサポートするため、Windows の 64 ビット バージョンは 32 ビット ファイルと 64 ビットファイルを個別の場所に格納して分離します。

64 ビット ターゲット システムは、通常 2 つの Program Files フォルダーを持ちます:

- Program Files – 64 ビット アプリケーション用
- Program Files (x86) – 32 ビット アプリケーション用

64 ビット ターゲット システムは、一般的に 2 つの Common Files フォルダー (各 Program Files フォルダーに 1 つ) を持ちます:

- Program Files¥Common Files – 64 ビット アプリケーション用
- Program Files (x86)¥Common Files – 32 ビット アプリケーション用

64 ビット ターゲット システムには、2 つのシステム フォルダーがあります:

- System32 – 64 ビット ライブラリと実行可能ファイル用
- SysWOW64 – 32 ビット ライブラリと実行可能ファイル用

32 ビット Windows Installer パッケージで、製品のファイルとフォルダーを Program Files フォルダーにインストールするように構成されている場合、Windows Installer は 32 ビット システム上の Program Files フォルダーに、これらのファイルとフォルダーをインストールします。ただし、エンド ユーザーが 32 ビット インストールを 64 ビット システムで実行した場合、Windows Installer はファイルとフォルダーを Program Files (x86) フォルダーにインストールします。

同様に、32 ビット Windows Installer パッケージで、製品のファイルとフォルダーを System32 フォルダーにインストールするように構成されている場合、Windows Installer はこれらのファイルとフォルダーを 32 ビット システム上では System32 に、64 ビット システム上では SysWOW64 にインストールします。

32 ビットと 64 ビット レジストリの場所の違い

64 ビット システム上で 32 ビット データと 64 ビット データの分離は、レジストリでも行われます。64 ビット ターゲット システムは、通常 2 つの HKEY_LOCAL_MACHINE¥Software キーを持ちます:

- HKLM¥Software – 64 ビット アプリケーション用
- HKLM¥Software¥Wow6432Node – 32 ビット アプリケーション用

32 ビット Windows Installer パッケージで、HKLM¥Software の下にある製品のレジストリ データをインストールするように構成されている場合、Windows Installer は 32 ビット システム上でそのデータを HKLM¥Software キーの下に、64 ビット システム上で HKLM¥Software¥Wow6432Node の下にインストールします。

***ヒント・** 64 ビット システム上で 32 ビット アプリケーションによるレジストリの表示を確認するには、32 ビット バージョンのレジストリ エディター (SysWOW64 フォルダーの* regedit.exe *ファイル) を起動します。*

32 ビットと 64 ビット Windows Installer パッケージを決定する

ほとんどの開発者は、すべてのエンド ユーザー向けに 32 ビット Windows Installer パッケージの作成を選択します。通常、32 ビット パッケージはエンド ユーザーが Windows の 32 ビット バージョンか 64 ビット バージョンのいずれを実行しているかに関わらず、エンド ユーアーのマシン上で正しく実行し、製品が設計通りに動作します。

製品が 32 ビット システム用の 32 ビットの製品ファイル、および 64 ビット システム用の 64 ビットの製品ファイルで構成されている場合、32 ビット システム上にインストールするエンド ユーザー向けに 32 ビット Windows Installer パッケージを、また 64 ビットシステム上にインストールするエンド ユーザー向けに 64 ビット Windows Installer パッケージを個別に作成することがあります。

製品が 64 ビット版 Windows を持つエンド ユーザーのみをターゲットとする場合、32 ビットではなく、64 ビット Windows Installer パッケージを作成したい場合があります。

InstallShield における 32 ビットと 64 ビットのサポート

InstallShield では、次の種類のインストールを作成するためのサポートが提供されています:

- 32 ビット ファイルとレジストリの場所にインストールする 32 ビット Windows Installer パッケージ。
- 64 ビット ファイルとレジストリの場所にインストールする 64 ビット Windows Installer パッケージ。この種類のパッケージは、32 ビット ファイルとレジストリの場所にもインストールが可能です。

InstallShield が 64 ビットまたは 32 ビット Windows Installer パッケージのどちらをビルドするかを決定する方法。

ビルド時に次の条件の 1 つまたは両方が True 評価されたとき、InstallShield は 64 ビット Windows Installer パッケージを作成します (Template Summary プロパティには x64 が指定されています):

- Express プロジェクトの [ファイル] ビューで、1 つ以上のファイルまたはフォルダ—が CommonFiles64Folder、ProgramFiles64Folder、または System64Folder にインストールされるように構成されている。
- Express プロジェクトの [レジストリ] ビューで、HKEY_LOCAL_MACHINE¥SOFTWARE (64 ビット) ノードの下に 1 つ以上のレジストリ エントリが含まれている。

また、InstallShield が 64 ビット Windows Installer パッケージをビルドするとき、64 ビットの場所のファイル、フォルダー、およびレジストリ エントリを含むコンポーネントを 64 ビットとしてマークします。

前述の条件がいずれも True でない場合、InstallShield は 32 ビット Windows Installer パッケージを作成します (Template Summary プロパティには Intel が指定されています): また、パッケージに含まれるコンポーネントは 64 ビットとしてマークされません。このような状況の場合、すべてのファイル、フォルダー、レジストリ エントリは、ターゲット システムが64 ビットであっても、すべて 32 ビットの場所にインストールされます。

***エディション・** 一部の 64 ビット ターゲット システム (たとえば、Windows Server Core システム) は、WOW64 (32-bit Windows-on-Windows) をサポートしません。これらの 64 ビット ターゲット システムは 32 ビット Windows Installer パッケージを実行できません。InstallShield Premier および InstallShield は、これらのシステムで動作するピュア 64 ビット専用 Windows Installer パッケージの作成をサポートします。*

また、InstallShield Premier および InstallShield には、単一のプロジェクト ファイルから 2 つの Windows Installer パッケージ (32 ビット版と 64 ビット版) を作成する機能、およびリリース フラグを使ってビルド時に各リリースでプロジェクトの様々な部分を含めるまたは除外することができる機能がサポートされています。InstallShield Express Edition を使って 32 ビット Windows Installer パッケージおよび 64 ビット Windows Installer パッケージを作成するには、2 つの個別の Express プロジェクトを作成しなくてはなりません。

InstallShield Premier Edition では、スイート/アドバンスト UI インストールがサポートされていて、64 ビット Windows Installer パッケージと 32 ビット Windows Installer パッケージを組み合わせて単一のインストールを作成することで、製品のインストール配布を簡素化することができます。スイート/アドバンスト UI インストールは、実行時に各ターゲット システムに適切なパッケージを実行します。

Express プロジェクトに 64 ビット マージ モジュールを含める

InstallShield では、Express プロジェクトに 64 ビット マージ モジュールを含めることができます。1 つ以上の 64 ビット コンポーネントが含まれた 64 ビット マージ モジュールを Express プロジェクトに追加する場合、64 ビットの場所にインストールされるように構成された 1 つ以上のファイル、フォルダー、またはレジストリ エントリを含まなくてはなりません。そうでない場合、ビルド時に次のエラーが発生します。

エラー -5008: この 32 ビット パッケージを 64 ビット データに含めることはできません。64 ビット データは、マージ モジュールに含まれている可能性があります。

Express プロジェクトに 64 ビットの場所にインストールするように構成された 1 つ以上のファイル、フォルダー、またはレジストリ エントリが含まれていない場合、InstallShield がビルド時に 32 ビット Windows Instaloler パッケージを作成しようとします。ただし、32 ビット Windows Installer パッケージに 1 つ以上の 64 ビット コンポーネントを含む 64 ビット マージ モジュールを含めることはできません。

64 ビット セットアップ ランチャーのビルド

InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い

管理者権限を持たずに InstallShield を起動した場合、以下の機能は使用できなせん:

- **COM 抽出**—COM サーバーからの COM 情報の抽出には、管理者権限が必要です。

  InstallShield を管理者権限を持たずに実行している場合に、プロジェクトで COM サーバーから COM 情報の抽出を行うように指定してからリリースをビルドしようとすると、ビルド エラー -6017 が発生します。

- **再配布可能ファイルのダウンロード**—[再配布可能ファイル] ビュー内から再配布可能ファイルをダウンロードするには、管理者権限が必要です。これは、InstallShield がファイルをマシンごとの場所にダウンロードする際に管理者権限が必要なためです。

  [再配布可能ファイル] ビュー内から再配布可能ファイルをダウンロードしようとしたとき、管理者権限が無い場合には次のメッセージが表示されます:

  ダウンロードが失敗しました。管理者として実行していること、および使用中のマシンがインターネットに接続されていることを確認してください。再試行しますか?

- **InstallShield 前提条件に [すべてのユーザー] の場所を指定できる機能**—[オプション] ダイアログ ボックスの [前提条件] タブを使って、[再配布可能ファイル] ビューに表示する InstallShield 前提条件を含むフォルダーを指定できます。このタブで [すべてのユーザー] の場所を変更するためには管理者権限が必要です。これは、InstallShield がレジストリ内でマシンごとの場所に情報を書き込むためです。したがって、InstallShield を管理者権限を持たずに実行している場合、このタブの [すべてのユーザー] の場所は無効となっています。

- **マージ モジュールに [すべてのユーザー] の場所を指定できる機能**—[オプション] ダイアログ ボックスの [マージ モジュール] タブを使って、[再配布可能ファイル] ビューに表示するマージ モジュールを含むフォルダーを指定できます。このタブで [すべてのユーザー] の場所を変更するためには管理者権限が必要です。これは、InstallShield がレジストリ内でマシンごとの場所に情報を書き込むためです。したがって、InstallShield を管理者権限を持たずに実行している場合、このタブの [すべてのユーザー] の場所は無効となっています。
- **Regasm.exe および InstallUtilLib.dll の場所を編集できる機能** – [オプション] ダイアログ ボックスの [.NET] タブで、.NET Framework に含まれているユーティリティである Regasm.exe および InstallUtilLib.dll ファイルの場所を指定できます。これらのユーティリティは COM interop と .NET カスタム アクションで利用されます。[.NET] タブでこれらの場所を変更するためには管理者権限が必要です。これは、InstallShield がレジストリ内でマシンごとの場所に情報を書き込むためです。したがって、InstallShield を管理者権限を持たずに実行している場合、このタブの場所設定は無効となっています。

管理者コンテキストと非管理者コンテキストの切り替えを行い、プロジェクトでマップされたドライブの場所を使用している場合、問題が発生することがあります。たとえば、管理者権限を持たずに Windows Explorer を使ってドライブ名を共有ネットワーク フォルダーにマップした場合、InstallShield の非管理者インスタンスではこのドライブにアクセスできますが、管理者インスタンスではアクセスできません。同様に、管理者権限を持っている場合に Windows Explorer を使ってドライブ名を共有ネットワーク フォルダーにマップすると、InstallShield の管理者インスタンスではこの場所にアクセスできますが、非管理者インスタンスではアクセスできません。このため、プロジェクトでネットワークの場所を参照する場合は、UNC パス (たとえば、¥¥server¥share) を使用するか、管理者と非管理者の両方でドライブ名をマッピングすることをお勧めします。

Visual Studio 内から InstallShield を使用している場合、管理者権限を持たない可能性があります。デフォルトでは、Windows Vista 以降のシステム上で Visual Studio のショートカットをダブルクリックして起動した場合、管理者権限はありません。

タスク ***Windows Vista 以降のシステム上で管理者権限を使って Visual Studio 内部から InstallShield を実行するには、以下の手順に従います:***

1. [スタート] メニューで Visual Studio のショートカットを右クリックしてから、**[管理者として実行]** をクリックします。
2. 新しい InstallShield プロジェクトを作成するか、または既存のプロジェクトを開きます。詳細は、以下のどちらかのトピックを参照してください:
   - Microsoft Visual Studio で InstallShield プロジェクトを作成する
   - Microsoft Visual Studio で InstallShield プロジェクトを開く

32 ビットと 64 ビット システムにおけるインストールの開発およびビルドの違い

InstallShield は、32 ビットと 64 ビット システムの両方で実行できる 32 ビット アプリケーションです。場合によって、InstallShield は、32 ビット システムで使用されているか、または 64 ビット システムで使用されているかによって動作が異なります。また、使用しているオペレーティング システムによっても、違いが見られることがあります。次のセクションは、これらの違いを説明します。

InstallShield と同様に、コマンドライン ビルド (ISCmdBld.exe) も 32 ビット アプリケーションです。したがって、同じプラットフォーム特有の違いは、このツールでも発生します。

プロジェクトへシステム ファイルを追加する

64 ビット システム上で InstallShield を使用している場合に、ソースの場所が開発マシン上の 64 ビット システム フォルダー (System32) であるシステム ファイルをプロジェクトに追加するとき、[ファイル] ビューの上部にあるソース コンピューターのペインからインストール先コンピューターのペインの適切な場所にドラッグすることはできません。

開発マシン上の 64 ビット System32 ファイルを InstallShield プロジェクトに追加するには、マシン上の Sysnative フォルダーを参照してから、プロジェクトに適切なファイルを選択します。詳細については、「64 ビット ソース マシンの 64 ビット System32 フォルダーからファイルを追加する」を参照してください。

64 ビット開発システム上で、ソースマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示する

InstallShield を 64 ビット開発システム上で使用する場合、InstallShield が表示する [レジストリ] ビューは、使用中のマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示します:

- HKEY_LOCAL_MACHINE¥Software
- HKEY_LOCAL_MACHINE¥Software¥Wow6432Node

このサポートによって、プロジェクトのレジストリ データの変更を構成する際、これらのソース領域からこのビューのインストール先ペインの適切な領域にエントリーをドラッグ アンド ドロップすることが可能となります。

インストールで 32 ビット領域へのリダイレクトを行わなずに 64 ビット ターゲット システム上の 64 ビット領域にレジストリ データをインストールする場合、レジストリ データを [レジストリ] ビューのインストール先ペインにある HKEY_LOCAL_MACHINE¥SOFTWARE (64 ビット) ノードに配置しなくてはなりません。[レジストリ] ビューのソースペインから 64 ビット データを、ビュー内の 64 ビット以外のインストール先ペインにドラッグするだけでは、そのコンポーネントが 64 ビットであるとマークされません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

64 ビット ファイルをスキャンして依存関係を確認する

InstallShield で依存関係スキャナーを使って、プロジェクトに追加する必要がある可能性がある依存関係を判別する場合、64 ビット ファイルの依存関係の確認は、InstallShield を 64 ビット オペレーティング システムでを使用している場合のみ可能です。

32 ビット システムで InstallShield を使用している場合、InstallShield の依存関係スキャナーを使って、プロジェクトの 64 ビット ファイルの依存関係を確認することはできません。32 ビット システムで 64 ビットの依存関係をプロジェクトに追加する場合、必須ファイルおよびマージ モジュールをプロジェクトに手動で追加する方法があります。

ファイルの依存関係のスキャンに関する詳細は、「アプリケーションの依存関係を識別する」を参照してください。

64 ビット .NET Installer クラスと COM Interop を使用する

Windows の 64 ビット バージョンで InstallShield を使用している場合、InstallShield の [ツール] メニューにある [オプション] をクリックすると表示される [オプション] ボックスでは、今回より、.NET Framework に含まれている Regasm.exe および InstallUtilLib.dll ファイルの場所として、32 ビット と 64 ビットの 2 つのパスを指定できます。InstallShield は、.NET Installer クラスと COM Interop を含むリリースで、ビルド時に指定されたパスを使用します。

64 ビット バージョンの Windows 上で ISCmdBld.exe を使ってコマンドラインからビルドしている場合に、既存の -t パラメーターを使って .NET Framework の 32 ビット バージョンのパスを指定すると、ISCmdBld.exe は、64 ビット .NET Installer クラスと COM Interop 用に Regasm.exe および InstallUtilLib.dll の 64 ビットの場所を使用します。

MSBuild または Team Foundation Server (TFS) を使ってビルドしている場合に、InstallShield タスクで既存の DotNetUtilPath パラメーターを使って .NET Framework の 32 ビット バージョンのパスを指定すると、ビルドは、64 ビット .NET Installer クラスと COM Interop 用に Regasm.exe および InstallUtilLib.dll の 64 ビットの場所を使用します。

32 ビット オペレーティング システムで InstallShield を使用している場合、64 ビットの場所の設定は無効になり、64 ビット .NET Installer クラスと COM Interop の 32 ビット サポートのみ有効になります。このサポートは、ISCmdBld.exe を使ってコマンドラインから実行するビルド、および、MSBuild または TFS を介して実行するビルドにも適用します。

詳しくは、次を参照してください:

- [.NET] タブ ([オプション] ダイアログ ボックス)
- IsCmdBld.exe
- Microsoft ビルド エンジン (MSBuild)

ヘルプの使い方

Revenera は、役に立つ情報やヘルプ リソースがいつでもすぐ取り出せることがいかに重要かを理解しています。InstallShield インターフェイスの様々なビュー内部に埋め込まれているインライン ヘルプとは別に、InstallShield には製品と共にインストールされるヘルプ ライブラリ (Web で使用できるオンライン ヘルプ システム、および PDF フォーマットで提供されるドキュメント) が含まれています。

InstallShield ヘルプ ライブラリ

製品に関してご不明な点がありましたら、まず InstallShield ヘルプ ライブラリを参照してください。これは総合的なユーザー ガイドです。

InstallShield の **[ヘルプ]** メニューから InstallShield ヘルプ ライブラリにアクセスするには、F1 を押すか、インターフェイスの [ヘルプ] ボタンをクリックします。

InstallShield ヘルプ ライブラリの利用に、インターネットの接続は必要ありません。オンライン ヘルプ ビューアーは、基本的に、個人のニーズに基づいてテクニカル情報を表示、検索、フィルターするためのツールです。

コンテキスト ヘルプを使用する

プロジェクトの作業中に、ソフトウェア オブジェクトをクリックすると、[ヘルプ] ウィンドウにヘルプ情報が表示されます。これは、「コンテキスト」ヘルプとも呼ばれます。

ヘルプ情報は、エクスプローラー ウィンドウで表示される選択されたソフトウェア オブジェクトの各プロパティでも参照することができ、プロパティの設定方法を見ることができます。

Web ベースのオンライン ヘルプ

Web ベースのオンライン ヘルプは、24 時間いつでも、http://docs.revenera.com から利用することができます。このヘルプ リソース センターより、最新の情報がほぼリアル タイムに入手可能です。

PDF 版のドキュメント

InstallShield ドキュメントの PDF 版は https://community.revenera.com/ からご利用いただけます。

お問い合わせ先

Revenera は本社をイリノイ州イタスカに置き、世界各地に拠点を持ちます。製品についてのお問い合わせなど、連絡先は当社の Web サイトをご覧ください：

http://www.revenera.com

2

スタート ガイド

InstallShield では、安定した Windows Installer インストールを簡単に作成することができる強力な機能と、作業時間を大幅に節約することができるツールが多数提供されています。InstallShield ヘルプ ライブラリは、InstallShield で提供されている機能をより効果的に活用するためのリソースです。まず最初にどのトピックから読み始めるかは、InstallShield インストール作成ソフトウェアの利用経験によって異なります。下のテーブルは、それぞれの経験レベルに応じて様々なトピックにリンクしています。

テーブル 2-1 • スタート ガイド ロード マップ

経験レベル	トップ ヘルプ トピック
インストールを以前に作成した経験なし。	インストールに関する一般的な情報については、次のトピックを参照してください。 • インストールの基本 • アプリケーション ライフサイクル
InstallShield を利用するのは初めて。	インストール作成経験はあるが、InstallShield を利用するのは初めての場合、次のトピックを参照してください。 • InstallShield インターフェイスを使って作業する • プロジェクトについて • チュートリアル • サポートされているアプリケーション プログラミング言語
他の Revenera 製品を利用した経験がある。	Revenera の他の製品を利用したことがある場合、次のトピックを参照してください。 • InstallShield の以前のバージョンからの移行する • InstallShield Premier または InstallShield へのアップグレード • プロジェクト アシスタント • インストールの概要

テーブル 2-1 • スタート ガイド ロード マップ (続き)

経験レベル	トップ ヘルプ トピック
InstallShield の中級、または上級レベルユーザー。	InstallShield を使い慣れている場合、次のトピックの参照をお勧めします: • エラーと警告

インストールの基本

インストールとは、一言で説明すると、ファイルおよびプログラムをユーザーのマシンにインストールするために使われる「パッケージ」を意味します。インストールは、インストーラー エンジンと対話するロジック、並びにアプリケーション ファイルで構成される集合体です。インストールの最も基本的な役割は、アプリケーション ファイルをソース媒体からエンド ユーザーのコンピューターへ転送することです。Windows オペレーティング システムは大変複雑なため、InstallShield のようなユーティリティのヘルプなしに、効果的で首尾一貫したインストールを簡単に作成することはできません。

インストールは、「製品」、「機能」、「コンポーネント」の 3 つの階層に分けられます。次の図はこの階層を説明します。

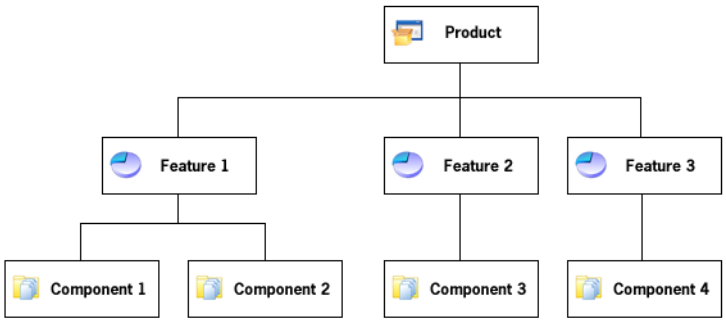


図 2-1: インストールの階層構造

*製品*は、インストール プロジェクトの中で最も上位にある構成区分です。製品は、通常 1 つのメイン アプリケーション (たとえば、ワードプロセッサ) と、そのアプリケーションが必要とするすべてのファイルとデータで構成されます。一連のアプリケーションが 1 つの製品となることも可能です。

*機能*は、エンドユーザーから見て、個別にインストール可能な最小の製品構成単位です。インストール プログラムの設計者は一般的に、どの機能をインストールし、どの機能をソース メディアに残すのかをユーザーが選択できるようにします。ワードプロセッサ製品では、メインの実行可能ファイルが 1 つの機能で、オプションの辞書が別の機能となっている場合があります。機能には必要な要素のすべてが含まれており、兄弟機能を必要としないのが原則です。たとえば、類語辞典機能はユーザーがインストールしないことを選択できる辞書機能を必要とはしません。ただし、機能にサブ機能を含めることもできます。サブ機能があると、エンドユーザーはインストールするファイルやデータをさらに細かく選択できるようになります。

プロジェクト内の各機能は 1 つ以上の*コンポーネント*で構成されます。コンポーネントは、個別にインストール可能な最小の製品構成単位です。InstallShield Express Edition では、コンポーネントが自動で作成されます。各コンポーネントに分かれた機能の内訳は、InstallShield ユーザー インターフェイスのどこにも表示されません。また、コンポーネントはエンドユーザーには表示されません。各コンポーネントは類似のプロパティを持つファイル (およびその他のリソース) を含んでいます。たとえば、コンポーネント内のファイルはすべてエンド ユーザーのマシン上の同じディレクトリにインストールされます。また、コンポーネント内のファイルはすべて同じ

オペレーティング システムまたは言語に適用される必要があります。辞書機能には複数の言語別辞書コンポーネントが含まれている場合が考えられます。コンポーネントはファイルだけでなく、レジストリデータ、ショートカット、ファイル拡張子情報、およびユーザーのマシンに書き込まれるその他のシステムデータを通常含んでいます。

インストールの概要

セットアップについて詳しくご存知ない方でも、過去に使用した経験はあるでしょう。コンピューターにソフトウェアをインストールしたことがある方は、実行中のインストールをエンド ユーザーの視点でご覧になっています。セットアップはファイルをソース メディアからローカル ドライブに転送します。レジストリ エントリの作成、ショートカットの作成や COM サーバーの登録もします。一般的に、セットアップはターゲット マシンおよびユーザーの情報を収集します。

インストールの典型的な要素

ファイル転送の実行

ファイル転送とは、CD またはフロッピー ディスクのようなソース メディアから、ファイルをエンド ユーザーのマシンにあるローカル ドライブへコピーすることを指します。エンドユーザーが選ぶ構成によって、すべてのファイルまたは一部のみがローカル ディスクに転送されます。ファイル転送の間、セットアップは新機能や便利なヒントなどの製品情報を提供するビルボードを表示します。ファイル転送処理の進行状況がわかるステータスバーも表示されます。

ユーザー インターフェイスの表示

セットアップのユーザー インターフェイスはエンドユーザーに情報やセットアップの設定オプションを提供します。エンドユーザーは、ユーザー インターフェイスを通して製品の一部をインストールするか、一部のファイルをソース メディアに残すか、使用許諾契約を表示するか、またはインストールを正しく構成するためのインストーラーに情報を提供するかなどを選択できます。ユーザー インターフェイスはニーズに合わせてカスタマイズが可能です。たとえば、ソフトウェアを不正な使用から保護するために、インストールの前にシリアル番号の入力をユーザーに求めるように指定できます。

ショートカットの作成

ショートカットはセットアップ中にエンド ユーザーのマシンで作成ができるファイルおよびアプリケーションへのリンクです。ショートカットは通常、ターゲット マシンのデスクトップまたは [スタート] メニューに配置され、プログラムまたはそのファイルへ素早く簡単にアクセスできるようにします。

ファイル関連の登録

識別されたファイルの種類を製品が使用している場合は、そのファイルの種類をエンド ユーザーのシステムに登録する必要があります。たとえば、メモ帳は .txt 拡張子でファイルを作成します。エンド ユーザーのシステムでファイルの種類の識別を可能にするには、システムのレジストリに登録されている必要があります。ファイルタイプの登録のプロセスはセットアップ時に処理されています。

COM、COM+、および DCOM ファイルの登録

ActiveX、COM や COM+ ファイルなどの COM サーバーは、アプリケーションがファイルのインターフェイスにアクセスできるように特殊な登録方法が必要です。これまでは、EXE、DLL、および OCX はインストールの間にファイルを登録させる、自動登録機能を備えていました。ただし、しかし、自己登録だけでは、問題が発生する場合もあります。エンドユーザーは、どの情報が登録されたか、またファイルがアンインストールされたときにレジストリ エントリが完全にクリーンアップされたか確かではありません。

Windows Installer サービスが提供しているソリューションは、必要なレジストリ エントリをセットアップの間書き込み、COM コンポーネントがアンインストールされるときに削除します。この方法は、COM サーバーが正しく登録されているか確認します。

製品をアンインストールに登録

製品をアンインストールする場合、オペレーティング システムは製品の存在を認知している必要があります。セットアップはオペレーティング システムに製品を登録するので簡単にアンインストールができます。Windows 2000 ロゴ プログラム要件では、この登録を必要とします。この処理で登録されるほとんどの情報は、コントロール パネルの [プログラムの追加と削除] でエンドユーザーに提供されます。たとえば、テクニカル サポートの連絡先情報、製品アップデートの情報、製品バージョン、および製品のパブリッシャーの情報はこのプロセスですべて登録されます。

アプリケーション ライフサイクル

アプリケーションのライフ サイクルは、顧客がアプリケーションをインストールした段階ですべて終了ではありません。ソフトウェア ベンダーとして、初回インストールが顧客のデスクトップで完了しただけでは、アプリケーションが成功したとは言えません。顧客は、製品のアップデートや強化内容、および重要な情報に簡単にアクセスできることを当然と考えています。この意味で顧客とのコミュニケーションやアプリケーションの状態を監視できることは、将来的な利益と発展には非常に重要です。

ソフトウェア ベンダーの中には、顧客主導のコミュニケーションを要求しがちですが、積極的に顧客との対話の機会を持たないベンダーは、多くのチャンスを逃すことになります。顧客が Web サイトまたはユーザー コミュニティを頻繁に訪れない限り、彼らはアップデート、アップグレード、パッチ、および一般的な技術記事などの情報を見逃すことになり、ベンダーは収益とサービスの機会を逃す結果となります。

図 2-2: FlexNet Connect のアプリケーション ライフサイクルの管理方法

上記のダイアグラムは、アプリケーション ライフサイクルの管理における FlexNet Connect の役割を図式したものです。

1. **インストールの作成**—を利用して、ソフトウェア開発者はすべてのプラットフォーム上で実行可能なインストールを容易に作成することができます。
2. **インストールの実行**—InstallShield 技術を使って作成されたインストールは、世界中 4 億以上のマシン上にインストールされています。
3. **アップデートの作成**—InstallShieldを使用すると、ソフトウェア開発者は素早くパッチおよびアップデートをビルドすることができます。
4. **ユーザーへの通知**—FlexNet Connect は、新しいアップデートのインストールが入手可能であることを各ユーザーに通知します。
5. **ダウンロードとインストール** — FlexNet Connect はアップデートのダウンロードおよびインストールをシームレスに統合し、一括処理します。
6. **レポートの表示** — FlexNet Connect はアップデートの利用率について即座にフィードバックを提供します。

InstallShield の起動

InstallShield のスタート ページは、InstallShield が起動されたとき最初に表示されるページです。スタート ページを使うと、製品情報、前回開いたプロジェクト、InstallShield リソースに簡単にアクセスすることができます。

InstallShield スタート ページ

InstallShield スタートページから、製品情報、最近開いたプロジェクト、InstallShield のリソースに簡単にアクセスすることができます。[スタート ページ] は次のセクションに分かれています。

テーブル 2-2 · スタート ページのセクション

セクション	説明
プロジェクト タスク	プロジェクト タスクをクリックして、新規プロジェクトを簡単に作成したり、既存プロジェクトを開くことができる他、InstallShieldのインストール付属のサンプル プロジェクトを参照することもできます。
ヘルプ トピック	アクセスの多いヘルプ トピックをこのセクションに示します。[スタート] ページから InstallShield ヘルプ ライブラリ全体にアクセスするには、F1 を押すか [リソース] セクションのヘルプ ライブラリ リンクをクリックします。
(最近開いたプロジェクト)	スタート ページの中央には、前回アクセスしたプロジェクト、プロジェクト タイプ、および前回変更が加えられた日時が表示されます。
スタート ガイド	InstallShield およびインストール作成ツールに関する知識レベルに基づいて、InstallShield ヘルプ ライブラリのどのトピックを読むべきかを案内するスタート ガイドをクリックしてください。

テーブル 2-2 • スタート ページのセクション (続き)

セクション	説明
リソース	[リソース] セクションには、役に立つ InstallShield 情報へのリンクが表示されています。 • ヘルプ ライブラリ – InstallShield のマニュアルを表示します。 • InstallShield コミュニティ–他の InstallShield ユーザーとの情報交換、質問の投稿、または回答の検索ができる Web ベースのフォーラムです。 • オンライン セミナー – InstallShield の評価、また Revenera 製品を最大限に利用するのに役立つ Web ベースの無料セミナーへのリンクです。 • ダウンロード–最新版の InstallShield 前提条件、InstallShield マージ モジュール、オブジェクト、ならびにサービス パック、パッチなど、ご使用中の InstallShield バージョンで利用できるダウンロードへのリンクです。 • リリース ノート – Revenera Web サイトのナレッジ ベースに掲載されているリリース ノートを表示します。 • 既知の問題–ご使用中の InstallShield バージョンに関する既知の問題を取り扱うナレッジベース記事を表示し、ワークアラウンドおよび解決策を提供します。 • アップグレード アラート –ナレッジベース記事が表示され、InstallShield の初期のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードしたときに発生する可能性がある問題についての情報が提供されます。また、新しい InstallShield 2021 Express Edition プロジェクトと以前のバージョンからアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。 • RSS フィード –InstallShield ナレッジ ベース記事をサブスクライブできる Web ページが表示されます。
お問い合わせ	Revenera Web サイトの [サポート] にアクセスするには、このセクションにあるリンクの 1 つをクリックしてください。

プロジェクトについて

インストールを作成すると、InstallShield プロジェクト ファイル (.ise) も作成されます。このファイルに、インストール ファイル (.msi) ファイルをビルドするために必要なすべてのロジックや情報が格納されます。InstallShield プロジェクトは、プロジェクトの出力を構成するファイル、フォルダーおよびオペレーションを指定します。プロジェクトの出力はインストールです。

プロジェクトは、必要度に応じてシンプルにも複雑にもなります。シンプルなプロジェクトは、ファイル、機能、レジストリ エントリのみということもあります。より複雑なプロジェクトはこれらの他さらに、再配布可能ファイル、初期化ファイルの変更を含み、外部の .dll ファイル関数を呼び出すこともあります。

プロジェクトの種類

InstallShield は様々なプロジェクトの種類を提供し、エンドユーザー向けの最適なプロジェクトの作成を支援します。

- **Express プロジェクト**—Express プロジェクトは、インストールのユーザー インターフェイスを提供するために Windows Installer を利用します。このプロジェクトの種類を選択した場合、機能を作成し、アプリケーション ファイルおよび他の配布可能なデータをすべて指定する必要があります。
- **QuickPatch プロジェクト** – このプロジェクト タイプは、サイズの小さいシングル アップグレードをエンドユーザーへ配布したいインストール作成者にお勧めします。QuickPatch の作成は、新規 QuickPatch 作成ウィザードを使って開始してください。
- **Visual Basic .NET ウィザード**—Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード を起動するとき、このオプションを選択します。
- **Visual C++ .NET ウィザード**—Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード を起動するとき、このオプションを選択します。
- **C# .NET ウィザード**—Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード を起動するとき、このオプションを選択します。

Express プロジェクト

Express は InstallShield でインストールを作成するための標準プロジェクト タイプです。Express プロジェクトは、インストール全体を実行に Windows Installer サービスを使用します。インストールのユーザー インターフェイスのフローは、.msi パッケージで直接作成することができ、Windows Installer サービスはユーザー インターフェイスのレンダリング機能を使用して、エンドユーザーに UI を表示します。

QuickPatch プロジェクト

QuickPatch プロジェクトは、規模の小さいシングル アップグレードをユーザーへ配布したいインストール作成者へお勧めするプロジェクトの種類です。カスタム アクションの追加、.ini データの変更などのより広範囲におよび変更には通常、標準パッチが必要です。

QuickPatch はカスタマイズ可能な範囲が限られてはいますが、[パッチのデザイン] ビューを使わないシンプルなパッチ構成方法として利用できます。基本的にどちらのパッチ作成方法も同じ配布タイプ (.msp と .exe ファイル) を作成します。

QuickPatch では、次のすべてを実行することができます。

- 元のインストールまたは以前の QuickPatch へ新しいファイルを追加する。
- 元のインストールのファイルを削除する。
- 以前の QuickPatch と共に追加されたファイルを削除する。
- 上記と同じ操作をレジストリ エントリで実行する。
- 元のインストールに含まれていたが、現在の QuickPatch プロジェクトには適用しないカスタム アクションを削除する。

QuickPatch プロジェクトの作成は、新規 QuickPatch 作成ウィザードで始めます。ウィザードを完了すると、QuickPatch プロジェクトに必要な基本条件をすべて満たすことができます。プロジェクト設定はそのあと InstallShield で開いた時に構成することができます。

プロジェクトの使用

InstallShield では、インストール プロジェクトから InstallShield 内で機能性を再使用することができる InstallShield オブジェクト プロジェクトまで、様々な種類のプロジェクトを作成、編集、アップグレードおよび保存することができます。

このセクション内のページでは、特定のプロジェクト タイプの作成方法、プロジェクトの保存方法、およびその他の関連情報を含むさまざまなトピックが網羅されています。

新規プロジェクトの作成

InstallShield プロジェクトを新規に作成するとき、その方法はいくつかあります。

タスク ***新しいプロジェクトを作成するには、以下のいずれかを行います。***

- ツールバーまたは **スタート ページ**で **[新規プロジェクト]** ボタンをクリックします。
- Ctrl+N を押します。
- **[ファイル]** メニューで、**[新規]** をクリックします。

これらの手順を行うと、[新規プロジェクト] ダイアログ ボックスが起動されるので、そこから作成するプロジェクトを選択できます。[新規プロジェクト] ダイアログ ボックスでインストール プロジェクトを選択すると、プロジェクト作成を支援するプロジェクト アシスタントが起動します。

Microsoft Visual Studio 内からプロジェクトを作成する

Microsoft Visual Studio .NET ワークスペース内から InstallShield プロジェクトを作成することができます。詳細については、「Microsoft Visual Studio で InstallShield プロジェクトを作成する」を参照してください。

プロジェクトを開く

タスク ***既存の InstallShield プロジェクトを開くには、次のいずれかの操作を実行します。***

- ツールバーにある **[プロジェクトを開く]** ボタンをクリックする。
- **[ファイル]** メニューで、**[開く]** をクリックします。
- CTRL+O を押す。
- **スタート ページ**で、**[既存のプロジェクトを開く]** リンクをクリックするか、または [最近開いたプロジェクト] リンクをクリックする。
- デスクトップ上または Windows エクスプローラー内でプロジェクト ファイルをダブルクリックする。

特定のファイルまたはファイル リンクをクリックした場合を除き、上のすべてのオプションで [開く] ダイアログ ボックスが開くので、ここでプロジェクト ファイルを参照する。

開きたいインストール プロジェクトがバージョン 3 以前の InstallShield Express で作成されている場合は、「InstallShield の以前のバージョンからの移行する」を参照してください。

以前のバージョンの InstallShield で作成されたプロジェクトを開く

タスク *以前のバージョンの InstallShield で作成されたプロジェクトを開くには、以下の手順に従います:*

1. **[ファイル]** メニューで、**[開く]** をクリックします。**[InstallShield Express プロジェクトを開く]** ダイアログ ボックスが開きます。
2. **[ファイルの種類]** リストから、開きたいプロジェクトの種類を選択します。
3. 必要なプロジェクトファイルを選択します。
4. **[開く]** をクリックして、このプロジェクトの移行を開始します。

プロジェクトの保存

新しいプロジェクトを作成すると、プロジェクトは自動的に [新規プロジェクト] ダイアログ ボックス で指定した名前と場所を使って保存されます。

InstallShield では、開いたプロジェクトのコピーを新しいプロジェクトとして別の名前をつけて新しい場所に保存することができます。

新しい名前と場所でプロジェクトを保存する

新しい名前をつけて新しい場所にプロジェクト保存したとき、名前が変更されたプロジェクトファイルおよびその関連ファイル並びにフォルダーのコピーが新しい場所に保存されます。次回プロジェクトを保存する際、変更はすべてこの新しいフォルダーに保存されます。

タスク *新しい名前を付けて、新しい場所にプロジェクトを保存するには、次の手順を実行します。*

1. **[ファイル]** メニューで、**[名前を付けて保存]** をクリックします。**[名前を付けて保存]** ダイアログ ボックスが開きます。
2. **[名前を付て保存]** ボックスで、適切な場所を選択します。

メモ: *プロジェクトを外部依存関係と共に保存すると、新しいプロジェクトはこれらのファイルの元のコピーをポイントします。コピーは複製されません。したがって、オリジナルのプロジェクトを削除する場合は、新しいプロジェクトで使用する可能性のあるファイルを削除しないように注意してください。*

デフォルトのプロジェクトの場所を変更する

すべての新しいプロジェクトは次のデフォルトの保存場所に保存されます:

C:¥InstallShield 2021 Projects

タスク *インストール プロジェクトに新しいデフォルトの場所を指定するには、以下の手順に従います:*

1. [**ツール**] メニューから [**オプション**] を選択します。Options ダイアログ ボックスが開きます。
2. [**ファイルの場所**] タブをクリックします。
3. [**プロジェクトの場所**] ボックスで、新しいパスを入力するか、または [**参照**] をクリックして適切な場所を見つけます。
4. [OK] をクリックします。

InstallShield で作成するすべての新しいプロジェクトのデフォルトの場所には、指定された場所が使用されます。このフォルダーはすべての新規プロジェクトに対して使用されますが、既存のプロジェクトは以前の場所にそのまま保存されます。

GUID

GUID は、グローバル固有 ID (Globally Unique Identifier) の略です。GUID は 128 ビット長で、各 GUID は GUID 生成に使用されるアルゴリズムによって一意になるように生成されます。GUID は固有でなければならないため、COM クラスや製品コード、およびその他のさまざまなコードを識別するのに使用することができます。

たとえば製品をインストールした後で、HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Uninstall の下にキーを作成し、インストールの製品コードに従って命名することができます。このキーは製品名に従って命名されていたことがありました。しかしこのために競合が起きる可能性がありました。2 つのインストールが同じマシンにインストールされて、両方が同じ製品名を共有する場合、同じレジストリキーが共有されることになったからです。GUID が使用されるようになったため、このような競合が起きることがなくなりました。

GUID は {5D607F6A-AF48-4003-AFA8-69E019A4496F} のような形式です。GUID の文字はすべて大文字でなければなりません。

プロジェクトの GUID

インストール プロジェクトを作成すると、プロジェクトに関連した多くの異なる GUID ができます。

テーブル 2-3 • GUID

GUID 名	説明
製品コードと製品 GUID	製品 GUID は、アプリケーションを一意に識別します。
パッケージコードとパッケージ GUID	パッケージ コードはインストール パッケージを一意に識別します。
QuickPatch GUID	パッチ GUID は QuickPatch パッケージを一意に識別します。
アップグレード コードとアップグレード GUID	アップグレード GUID はアップグレードの目的で製品のファミリーを識別します。これはアップグレードに重要です。

プロジェクトで GUID を変更する必要がある場合についての情報は、「パッケージ コード、製品バージョン、および製品コードのアップデート」を参照してください。

サンプル ファイル

いくつかのサンプル ファイルが InstallShield に含まれています。これらのファイルは、次の場所の Samples フォルダーに格納されています。

InstallShield Program Files フォルダー¥Samples

これらのファイルは、InstallShield ヘルプ ライブラリで提供されているチュートリアルと共に使用することができます。

プロジェクト アシスタント

InstallShield が提供するプロジェクト アシスタントを使って、基本インストール プロジェクトを短期間で簡単に作成することができます。プロジェクト アシスタントはインストール プロジェクト作業のフレームワークを提供し、プロジェクト作成プロセスの手順を案内し、その関連情報を提供します。

プロジェクト アシスタントのしくみ

インストール プロジェクトを新しく作成すると、[プロジェクト アシスタント] ビューが自動的に開きます。プロジェクト アシスタントは Express プロジェクト タイプで使用できます。

プロジェクト アシスタントで入力した情報は、基本となるプロジェクト ファイルに直接保存されます。そこで、インストール デザイナー (下記参照) に切替えて、InstallShield インターフェイスの機能を最大限に活かして情報を表示したり変更したりすることができます。この場合、プロジェクト アシスタントを使用すると、必要に応じてインストール デザイナーを使用する高度なインストールのために基礎を作成します。

インストールデザイナとの統合

Express プロジェクト タイプに提供されている [インストール デザイナー] タブには、InstallShield インターフェイスのビューがすべて表示されます。ここから、より複雑でパワフルな要素をインストール プロジェクトに追加することができます。プロジェクト アシスタントでインストール プロジェクトを作成してから、インストールデザイナー を使用してプロジェクト要素を微調整することができます。

インストール デザイナーとプロジェクト アシスタントは同時に実行されます。一方に変更を加えると、すぐにもう一方にその変更が反映されます。たとえば、[インストール デザイナー] タブである機能を削除すると、その機能はインストール プロジェクトからなくなり、プロジェクト アシスタントでも表示されなくなります。

プロジェクト アシスタントを使用する

インストール プロジェクトを新しく作成すると、プロジェクト アシスタントが自動的に開きます。ウェルカム ページには、インストール デザインのダイアグラムがあり、インストール作成の手順が一目で分かるようになっています。プロジェクト アシスタントを使ってプロジェクトを作成するか、[インストールデザイナ] タブをクリックして基本のインストール プロジェクトをさらに定義します。

リストコントロールの使用

適用する場合は各ページには最大で 3 つのリストコントロールが含まれ、これらはインストールの作成と情報の検索に役立ちます。

- **その他のオプション**—プロジェクト アシスタント ページにある特定部分に関する追加構成オプションを提供します。これらは、プロジェクト アシスタントの機能のための他より一般的ではないオプションです。
- **他の場所**—現在のプロジェクト アシスタントのページに対応するインストール デザイナー内のビュー。リンクをクリックすると、インストール デザイナが起動され、そのビューがアクティブになります。
- **ヘルプ リンク**—このリストは現在のプロジェクト アシスタント ページに関連するヘルプ トピックへのリンクを提供します。

プロジェクト アシスタント内を移動する

タスク ***別のプロジェクト アシスタントに移動するには、次のどれかを実行します。***

- 特定のページに直接移動するには、ページの下部にあるナビゲーション バーで表示されている該当するアイコンをクリックします。
- プロジェクト アシスタントの手順に従うには、以下の通りに実行します。
  - [次へ] または [戻る] 矢印ボタンをクリックして、前/後ろに移動します。
  - CTRL+TAB を押すと、次のページに移動し、CTRL+SHIFT+TAB を押すと、前のページに移動します。
- [ホーム] に戻ってインストールデザイン解説図を表示するには、ナビゲーションバーの [ホーム] ボタンをクリックします。

インストール デザイナーを開く

[インストール デザイナー] タブには、InstallShield インストール開発環境 (IDE) のビューが表示されます。ここから、より複雑でパワフルな要素をインストール プロジェクトに追加することができます。インストール デザイナーでビューを開くには、[インストール デザイナー] タブをクリックします。

メモ: *インストール デザイナーとプロジェクト アシスタントは同時に実行されます。一方に変更を加えると、すぐにもう一方にその変更が反映されます。*

プロジェクト アシスタントを非表示にする

タスク ***プロジェクト アシスタントを非表示にするには、以下の手順を実行します。***

[ビュー] メニューで、**[プロジェクト アシスタント]** をクリックします。

プロジェクト アシスタント コマンドの隣にあったチェック マークが消えて、プロジェクト アシスタントが非表示となります。

プロジェクト アシスタントは [ビュー] メニューで選択されるまで非表示のままです。新しいプロジェクトを作成して、プロジェクト アシスタント コマンドが選択解除されている場合は、インストール デザイナーがデフォルトのタブとなります。

[アプリケーション情報] ページ

[アプリケーション情報] ページで、プロジェクトがインストールするアプリケーションに関する情報を指定することができます。これには、アプリケーション名とバージョン、会社名、Web サイト アドレス、およびアプリケーション アイコンがあります。

コントロール パネルの [プログラムの追加と削除]

コントロール パネルの [プログラムの追加と削除] には、コンピューター システムにインストールされているアプリケーションの一覧が表示されます。特定のプログラムについての情報を表示したり、プログラムを追加、変更または削除することができます。

プロジェクト アシスタントの [アプリケーション情報] ページに提供する情報は、アプリケーションがインストールされた時にその [プログラムの追加と削除]の情報を完成する際に利用されます。

インストールの会社名と製品名

会社名と製品名は、インストール プロジェクトの数箇所で使用されます。

インストール プロジェクト内で会社名はどのように使われますか？

会社名は、アプリケーションのデフォルト インストール ディレクトリを設定するのに使用されます。また、エンド ユーザーのシステム上のアプリケーションのコントロール パネルにある [プログラムの追加と削除] でも使用されます。

インストール プロジェクト内で製品名はどのように使われますか？

製品名はアプリケーションの [プログラムの追加と削除] (サポート情報リンク) で使用されてます。これはデフォルト インストール ディレクトリの設定にも使用されます。

[インストール要件] ページ

[インストール要件] ページでは、ターゲット システムのインストールの要件を簡単に設定することができます。たとえば、アプリケーションに特定の OS が必要な場合、このページの最初の部分で示すことができます。

プロジェクト アシスタントでオペレーティング システム要件を指定する

プロジェクト アシスタントの [インストール要件] ページでオペレーティング システム要件を指定すると、InstallShield が起動条件を作成します。これらの条件は .msi ファイルの LaunchCondition テーブルに追加されます。

InstallShield がオペレーティング システム起動条件を作成する方法

[インストール要件] ページでオペレーティング システム要件を指定すると、アプリケーションをサポートしないオペレーティング システムを除外することになります。

たとえば、最新 Windows オペレーティング システムのチェック ボックスのみを選択した場合、InstallShield は [インストール要件] ページで選択しなかったオペレーティング システムを除外する起動条件を作成します。この起動条件タイプでは、将来的にリリースされる Windows オペレーティング システムのバージョンが自動的にサポートされます。これは起動条件で除外されていないためです。

インストールによる要件確認のタイミング

必要なソフトウェアまたはオペレーティング システムがターゲット システムに確実に存在するよう、インストールはファイル転送を開始する前にインストールについてこれらの必要条件を確認します。

ソフトウェア要件のランタイム メッセージを変更する

インストールにソフトウェア要件があり、ターゲット システムに選択されたソフトウェアがない場合、ランタイムのメッセージがインストール中に表示されます。表示されるメッセージは編集できます。

タスク ***ランタイム メッセージを編集するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、**[インストール要件]** ページを開きます。
2. ソフトウエア要件に関する質問には、**[はい]** を選びます。
3. アプリケーションに必要なソフトウェアを選択します。デフォルトのランタイム メッセージが右側に表示されます。
4. ランタイム メッセージを編集します。

カスタム インストール要件の作成

プロジェクト アシスタントまたはインストール デザイナー (IDE) の [必要条件] ビューを使って、インストールプロジェクトでターゲット システムの必要条件を指定することができます。システム ソフトウェア要件を指定するとき、システム検索ウィザード を使用します。[システム検索ウィザード] は、インストール前にターゲットのシステム上にある特定のファイル、フォルダー、レジストリキーまたは .ini 値を探す Windows Installer 機能を提供します。

タスク ***プロジェクト アシスタントのシステム検索ウィザードへアクセスするには、以下の手順を実行します。***

1. プロジェクト アシスタントで、**[インストール要件]** ページを開きます。
2. **[他のオプション]** 領域で、**[カスタム ソフトウェア条件の作成]** をクリックします。

[インストール アーキテクチャ] ページ

[インストール アーキテクチャ] ページで、インストール プログラムがエンド ユーザーに対して表示する機能を指定することができます。機能とは、ユーザーから見て個別にインストール可能な最小の製品構成単位のことです。インストール中に [カスタム] セットアップ タイプを選択すると、個々の機能がエンドユーザーに対して表示されます。

メモ: *機能には、サブ機能やサブサブ機能などを含めることも可能で、セットアップ プログラムの要求に応じて必要なレベルの多層構造化が可能です。*

プロジェクト アシスタントで機能を追加する

タスク ***機能を追加するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、[**インストール要件**] ページを開きます。
2. **インストール アーキテクチャをカスタマイズしますか?** の質問で、[**はい**] を選択します。
3. メイン機能を追加するには、[**インストール アーキテクチャ**] エクスプローラーをクリックします。サブ機能を追加するには、親機能となる機能をクリックしてから [**新規作成**] をクリックします。プロジェクト アシスタントは新しい機能を作成します。
4. 機能に名前を付けるか、後で [**名前の変更**] をクリックして名前を入力します。

複数機能インストールを作成するかどうかを判断する

機能は、エンド ユーザーから見たインストールの構成ブロックです。このため、機能はインストール内部での機能上はっきりと区別できる要素を象徴しなくてはなりません。

アプリケーションに機能的に異なるブロックが複数存在する場合、複数機能インストールを作成しなくてはなりません。たとえば、インストールにアプリケーション (.exe ファイル) とヘルプ ライブラリ (.hlp ファイル) を含む場合、インストール プロジェクトは各要素につき 1 つの機能とした最低 2 つの機能を含まなくてはなりません。

プロジェクト アシスタント内での複数機能インストールの作成に関する詳しい情報は、「複数の機能を持つインストールを作成する」をご覧下さい。

複数の機能を持つインストールを作成する

インストール デザイナー (IDE) のプロジェクト アシスタントを利用して、複数機能を持つインストールを作成することができます。

タスク ***プロジェクト アシスタントで複数機能インストールを作成するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、[**インストール要件**] ページを開きます。
2. インストール アーキテクチャをカスタマイズするには [**はい**] を選択します。

3. [**インストール アーキテクチャ**] エクスプローラーをクリックしてから、[**新規作成**] をクリックします。プロジェクト アシスタントは新規機能を作成します。
4. F2 を押すか、機能を右クリックして、[**名前の変更**] を選択して新しい機能の名前を付けます。
5. 同じレベルに別の機能を追加するには、[**インストール アーキテクチャ**] エクスプローラーをクリックしてから [**新規作成**] をクリックします。サブ機能を作成するには、親機能となる機能をクリックしてから [**新規作成**] をクリックします。プロジェクト アシスタントは新しい機能を作成します。
6. 必要に応じて機能およびサブ機能を追加します。

デフォルトの機能

*デフォルトの機能*の概念はプロジェクト アシスタントでのみ存在します。インストール プロジェクトに追加されたすべてのリソース (たとえば、ファイルまたはレジストリデータ) は機能に割り当てる必要があります。リソースが機能に割り当てられていない場合、それらは実行時にターゲット システムへインストールされません。

デフォルトの機能を利用すると、プロジェクト アシスタントでのオーサリング作業が簡素化されます。プロジェクトリソースが確実にインストールされるよう機能への割り当てに気を配る必要はありません。レジストリデータを追加するとき、新規ショートカットを作成するとき、またはすべてのアプリケーションデータが選択された時にファイルを追加するとき、これらすべてのリソースはデフォルト機能へ追加されます。これにより、プロジェクト アシスタントに追加するすべてのプロジェクト リソースは、ユーザーがインストールを実行したときにターゲット システムへインストールされます。

デフォルトの機能の設定

デフォルトの機能は、プロジェクト アシスタントの [インストール アーキテクチャ]ページで設定することができます。

機能が存在しない、またはデフォルトの機能が選択されていない場合はどうなりますか?

[インストール アーキテクチャ] ページへ移動したとき、および [アプリケーション ファイル]、[アプリケーション ショートカット]、または [アプリケーション レジストリ] ページにデータを追加したとき、InstallShield は 最初のルート機能をデフォルトの機能として選択します。機能が存在しない場合、InstallShield がサイレントで作成します。

機能の階層を定義する

最上位の機能とは、機能階層の最も上にある機能です。最上位の機能にはインストールするアプリケーション、ヘルプ ライブラリ機能、そしてサンプル プロジェクト機能などを含みます。

最上位の機能の下には、サブ機能または子機能があります。これはインストールの都合上、別の機能に依存している機能です。親機能 (最上位の機能) がターゲット システムにインストールされないと、子機能もインストールされません。

[アプリケーション ファイル] ページ

[アプリケーションファイル] ページでは、機能に関連付けるファイルを指定することができます。

プロジェクト アシスタントで機能にファイルを追加する

タスク ***ファイルを機能に追加するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、[**アプリケーション ファイル**] ページを開きます。
2. ページ上部にある機能リストで、ファイルが含まれる機能を選択します。
3. [**インストール先コンピューター**] エクスプローラーで、ファイルを追加するフォルダーを選択します。
4. [**ファイルの追加**] をクリックします。[**開く**] ダイアログ ボックスが開きます。
5. 追加するファイルを参照します。
6. [**開く**] をクリックして、選択した機能にファイルを追加します。「追加したファイルには依存関係がある可能性があります」というメッセージが表示されます。
7. インストール プロジェクトに依存関係を自動的に追加するようにする場合は、[**はい**] をクリックします。

プロジェクト アシスタントで機能からファイルを削除する

タスク ***ファイルを機能から削除するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、[**アプリケーション ファイル**] ページを開きます。
2. 削除するファイルをクリックして、Delete キーを押します。

固定のフォルダーの場所へファイルを追加する

ターゲット システム上のどの場所にプロジェクトファイルをインストールするのかが明白な場合、固定フォルダーの場所へのパスをハードコードで入力することができます。

タスク ***プロジェクト アシスタントでファイルを固定フォルダーの場所へ追加するには、以下の手順を実行します。***

1. プロジェクト アシスタントで、[**アプリケーション ファイル**] ページを開きます。
2. [**インストール先コンピューター**] を右クリックして、[**新しいフォルダー**] を選択します。
3. 新しいフォルダーの名前には、たとえば C: の様にインストール先を含むドライブ名を入力します。
4. ドライブ名フォルダーの下にサブフォルダーを追加して、詳しいインストール先のパスを定義します。

追加の定義済みフォルダーを表示する

プロジェクト アシスタントの [アプリケーション ファイル] ページには、一般的によく利用される定義済みフォルダーが表示されます。このページで定義済みフォルダーを表示または非表示にすることができます。

タスク ***追加定義済みフォルダーを表示するには、以下の手順に従います:***

1. プロジェクト アシスタントで、[**アプリケーション ファイル**] ページを開きます。
2. [**インストール先コンピューター**] を右クリックして、[**定義済みフォルダーを表示**] を選択します。
3. 定義済みフォルダーのリストで表示するフォルダーを選択します。

ヒント: *定義済みフォルダーを非表示にするには、定義済みフォルダーのリストで該当するものについて選択を解除します。*

[アプリケーションのショートカット] ページ

[アプリケーション ショートカット] ページでは、ターゲット システムのデスクトップ、または [スタート] メニュー上にあるアプリケーション ファイルのショートカットを指定できます。デフォルトで、このページは [プロジェクト アシスタント] を使ってプロジェクトに追加された各実行可能ファイルのショートカットを表示します。これらは削除することができ、インストール プロジェクトに入れたショートカットを他のファイルに追加することもできます。

ファイル拡張子

ファイル名拡張子の関連付け、つまりファイルの関連付けは、特定の種類のファイルを開くのにどのアプリケーションを使用すべきかを Windows に指示するためのレジストリ設定です。たとえば、Windows は通常 Windows の Notepad を使用してテキストファイル (.txt の拡張子を持つファイル) を開き、Microsoft ペイントを使用してビットマップファイル (.bmp 拡張子を持つファイル) を開きます。

ファイル拡張子を使って、ファイルにアクセスせずにファイルの種類を識別することができます。ファイル名の末尾に接尾辞 (**.abc**) が追加されます。ファイル拡張子はまた、別のアプリケーションが拡張子に基づいてそのファイルと互換性があるかどうか (たとえば、そのファイルを開くのか変更するのか) を識別するのにも便利です。

InstallShield で、[ファイルの拡張子] ビューで独自のファイル拡張子を登録することができます。ファイル拡張子を登録すると、エンド ユーザーがファイルを開いたとき、ターゲット マシンのオペレーティング システムが特定のファイル拡張子と共にアプリケーションを利用してファイルを開くよう指示することができます。

タスク ***プロジェクト アシスタントの [ファイル拡張子] ビューへアクセスするには、以下の手順を実行します。***

[**アプリケーション ショートカット**] ページの [**その他のオプション**] エリアで、[**ファイルの拡張子**] をクリックします。[**ファイルの拡張子**] ビューが開きます。

インストールに含まれていないファイルへのショートカットを作成する

既にターゲット システム上に存在するファイルを対象としたショートカットを作成するようにインストールを設定できます。このファイルはインストール プロジェクトに含まれている必要はありません。

タスク ***インストールに含まれていないファイルへのショートカットを作成するには、以下の手順を実行します。***

1. インストール デザイナーを開きます。
2. **[ターゲット システムの構成]** の下にあるビュー リストにある **[ショートカット/フォルダー]** をクリックします。
3. **[ショートカット]** エクスプローラで、ショートカットのインストール先を右クリックしてから、**[既存ファイルへの新しいショートカット]** をクリックします。**[ショートカット ターゲットの参照]** ダイアログが開きます。
4. ターゲット ファイルの場所を参照して **"ファイル名"** 設定にファイルの名前を入力します。
5. [OK] をクリックします。
6. ショートカットの設定を構成します。

プロジェクト アシスタントでデフォルトのショートカットを変更する

タスク ***デフォルトのショートカットを変更するには、以下の手順に従います:***

1. **プロジェクト アシスタント**で、**[アプリケーション ショートカット]** ページを開きます。
2. 変更するショートカットを選択します。
3. 必要に応じて変更を行います。

プロジェクト アシスタントでショートカットのターゲットにファイル拡張子を関連付ける

ファイル拡張子とショートカットのターゲットを関連付けることができます。これを行うとき、Windowsはターゲット ファイルを使用し、指定された拡張子のファイルを開きます。たとえば、.txt と入力すると、エンドユーザーが .txtファイルを開く際、このショートカットのターゲットファイルが起動して開きます。

タスク ***ショートカットのターゲットをファイル拡張子に関連付けるには、以下の手順に従います:***

1. **プロジェクト アシスタント**で、**[アプリケーション ショートカット]** ページを開きます。
2. ショートカットをクリックして、ショートカット オプションを有効にします。
3. **[ショートカットとファイル拡張子を関連付けるオプション]** を選択します。
4. このショートカットのターゲットと関連付けるファイル拡張子を入力します (例、**txt**)。カンマで区切って複数の拡張子を追加することもできます。

[アプリケーション レジストリ] ページ

[アプリケーション レジストリ] ページでは、アプリケーションに必要なレジストリ データを指定できます。

レジストリの更新

レジストリはコンピューターの構成情報が入ったデータベースです。コンピューターのレジストリに情報にはユーザープロフィール、コンピューターにインストールされたハードウエアおよびソフトウェア、プロパティ設定が含まれます。

アプリケーションに必要なレジストリデータの調べ方

アプリケーション開発者にレジストリ情報をたずねてください。特に、インストールするアプリケーションにユーザー固有 (HKEY_CURRENT_USER) またはマシン固有 (HKEY_LOCAL_MACHINE) の設定が必要かどうかを知る必要があります。

開発者からインストールに追加する .reg ファイルを受け取ります。InstallShield では、.reg ファイルをインストール プロジェクトにインポートすることができます。

プロジェクト アシスタントでレジストリ データを構成する

タスク ***レジストリ データを構成するには、以下の手順に従います。***

1. **プロジェクト アシスタント**で、**[アプリケーションのレジストリ]** ページを開きます。
2. レジストリ データの構成についての質問に対して、**[はい]** を選択します。
3. データを追加するレジストリ アイテムを右クリックして、**[新規]** を選択してから **[キー]** をポイントします。
4. キーに名前を入力します。
5. キーを右クリックし、[**新規**]を選択してから適切なコマンドをポイントします。登録するデータのタイプによって、[デフォルト値]、[文字列値]、[バイナリ値]、[DWORD 値]、[拡張可能文字列値] または [複数行値] から選択します。

プロジェクト アシスタントでレジストリ データの値を変更する

タスク ***レジストリ データを変更するには、以下の手順に従います。***

1. **プロジェクト アシスタント**で、**[アプリケーションのレジストリ]** ページを開きます。
2. データをダブルクリックする。**[編集]** ダイアログ ボックスが開きます。
3. データを編集して、**[OK]** をクリックします。

機能にレジストリ データを関連付ける

Express プロジェクトでは、プロジェクト アシスタントの [アプリケーション レジストリ] ページで追加したすべてのレジストリ データがプロジェクトのデフォルトの機能へ追加されます。インストール デザイナーでレジストリ データを別の機能に関連付けることができます。

タスク *インストール デザイナーを使って、レジストリ データをデフォルトの機能以外の機能に関連付けるには、以下の手順を実行します:*

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**レジストリ**] をクリックします。
2. [**ビュー フィルター**] リストで、レジストリ データを関連付ける機能を選択します。
3. レジストリ データを適切なレジストリの場所で作成するか、またはドラッグアンドドロップします。

レジストリ データで変数データ型を使用する

プロジェクト: *この情報は Express プロジェクトに適用します。*

InstallShield ではインストール プロジェクト用レジストリデータを作成する際に、変数データ型またはプロパティを利用することができます。

タスク *レジストリで INSTALLDIR を変数として使用するには、以下の手順に従います:*

1. **プロジェクト アシスタント**で、[**アプリケーションのレジストリ**] ページを開きます。
2. [**はい**] を選択して、アプリケーションがインストールするレジストリ データを構成することを示します。
3. HKEY_CLASSES_ROOT を右クリックし、[**新規**] をポイントしてから、[**キー**] をクリックします。
4. キーに [**インストール先**] という名前を付けます。
5. [**インストール先**] キーを右クリックし、[**新規作成**] をポイントしてから、[**文字列値**] をクリックします。
6. 文字列値に My Installation Location と名前を付けます。
7. My Installation Location キーをダブルクリックします。[**データの編集**] ダイアログ ボックスが開きます。
8. ***"値データ"*** フィールドに [INSTALLDIR] と入力します。

実行時に、[INSTALLDIR] の値はインストール ディレクトリと置き換えられます。

アプリケーション パス

アプリケーション パスのレジストリ キーには、指定アプリケーションの .dll ファイル用のプライベート検索パスとして Windows が使用するデータが含まれます。アプリケーションの .dll ファイルを PATH 環境変数で指定されていないディレクトリ (および、アプリケーションのディレクトリ以外) にインストールする場合は 、インストール中に .dll ファイル ディレクトリを含む適切なアプリケーション パスを設定するようにしてください。アプリケーション パス情報は、レジストリの HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥AppName.exe の下に格納されます。

[インストール インタビュー] ページ

[インストール インタビュー] ページで、インストール プログラムの実行時に、エンド ユーザーに表示するダイアログ ボックスを指定できます。このページでの回答に従って、プロジェクト アシスタントは対応するダイアログ ボックスをインストール プロジェクトに追加します。

プロジェクト アシスタントでインストールに使用するダイアログを指定する

プロジェクト アシスタントの [インストール インタビュー] ページに表示される質問に答えると、作成したインストール プログラムが実行されるときにエンド ユーザーに対して表示されるダイアログを指定することができます。

タスク ***インストールのダイアログを指定するには、次の手順に従います。***

1. **使用許諾契約ダイアログを表示しますか?**–**[はい]** を選択して、使用許諾契約書ファイルを参照します。
2. **ユーザーに対して会社名とユーザー名の入力をプロンプトしますか?** – **[はい]** を選択してこの情報を要求するダイアログを表示します。
3. **ユーザーがアプリケーションのインストール場所を変更することができるようにしますか?** – **[はい]** を選択して、エンド ユーザーがインストールの場所を変更できるようにします。詳細については、「エンドユーザーによるインストール先の変更を許可する」を参照してください。
4. **ユーザーがアプリケーションの特定部分のみ選択してインストールできるようにしますか?** – **[はい]** を選択して、エンド ユーザーがアプリケーションの特定部分のみインストールできるようにします。詳細については、「インストールが選択できるインストールを作成する」を参照してください。
5. **インストールの完了時にアプリケーションを起動するオプションをユーザーに提供しますか?**–**[はい]** を選択して、アプリケーション ファイルを参照します。このオプションが **[はい]** に設定されると、最後のダイアログにはチェック ボックスがあり、エンドユーザーが **[完了]** ボタンを押すとアプリケーションが直ちに起動します。

ヒント: *カスタム グラフィックをインストール ダイアログに追加するには、[他のオプション] セクションでリンクをクリックし、[ダイアログイメージ] ダイアログ ボックスを起動します。*

エンドユーザーによるインストール先の変更を許可する

エンドユーザーがシステム上でソフトウェアをインストール場所を選べるように、インストール場所の変更を許可することができます。

Windows Installer プロパティ INSTALLDIR はデフォルトのインストール ディレクトリとして機能します。ユーザーがインストール先を変更できるようにした場合、インストール中に [インストール先フォルダー] ダイアログが表示されます。

使用許諾契約書

アプリケーションをインストールするためには、エンド ユーザーは特定の法的条件に同意しなくてはなりません。たとえば、ほとんどのソフトウェア ベンダーはユーザーによるソフトウェアのコピーや第三者への配布を禁止しています。

エンドユーザーがソフトウェアインストールに関連する法的条件を確実に認識するよう、インストールは実行時にエンドユーザー使用許諾契約書 (EULA) を [使用許諾契約書] ダイアログに表示することができます。EULA はソフトウェアの使用についてエンドユーザーとの間に交わされる法的契約です。

[使用許諾契約書] ダイアログは、使用許諾契約テキストおよび [はい]/[いいえ] オプションを表示します。エンドユーザーが EULA に同意しない場合、ソフトウェアはインストールされることなくインストール処理が終了します。

タスク ***プロジェクト アシスタントで [使用許諾契約書] ダイアログをプロジェクトへ追加するには、以下の手順を実行します。***

1. **プロジェクト アシスタント**で、**[インストール インタビュー]** ページを開きます。
2. **[はい]** を選択して **[使用許諾契約]** ダイアログの追加を指定します。
3. 使用許諾契約書ファイルへのパスを入力する、またはファイルを参照します。ファイルはリッチ テキスト形式 (.rtf) でなくてはなりません。

インストールが選択できるインストールを作成する

エンドユーザーがインストールする部分を選択してシステムにインストールするようなインストールを作成することができます。これは、インストール内で使用できる機能の一覧を表示するカスタムインストールです。エンドユーザーは実行時に表示されるダイアログでインストールする機能を選択することができます。

たとえば、インストールの中にアプリケーションの実行可能ファイル (.exe) ファイル、ドキュメント (.chm) ファイル、およびサンプルファイルが入っているとします。これらのファイルはすべて異なる機能の中に入っていて、オプション一覧からエンドユーザーに提供されます。エンドユーザーがアプリケーションのみを必要とする場合は、実行可能ファイルだけをインストールし、ドキュメント ファイルとサンプル ファイルはインストールしないように選択することができます。

[インストールのビルド] ページ

ヒント: *以下の手順は、(Visual Studio との統合を行わず) InstallShield 内部でビルドされたリリースに適用します。Visual Studio 内部から InstallShield リリースをビルドする方法については、「Microsoft Visual Studio でリリースをビルドする」を参照してください。*

[インストールのビルド] ページでは、ビルドするタイプを指定し、再配布可能ファイルをコピーする場所も指定することができます。パッケージにデジタル署名を行うこともできます。

プロジェクト アシスタントからインストールをビルドする

ヒント: *以下の手順は、(Visual Studio との統合を行わず) InstallShield 内部でビルドされたリリースに適用します。Visual Studio 内部から InstallShield リリースをビルドする方法については、「Microsoft Visual Studio でリリースをビルドする」を参照してください。*

タスク ***インストールをビルドするには次の手順を実行します。***

1. **プロジェクト アシスタント**で、[**インストールのビルド**] ページを開きます。
2. インストールのイメージの種類を選択します。
3. InstallShieldで、ビルド後に自動的にインストールを別の場所にコピーする場合、各ビルド オプションの**オプション配布の設定** リンクをクリックして、場所を指定します。

   ビルド後に InstallShield がインストールを配布するように設定する場合、各ビルドオプションの **オプション配布**の設定 リンクをクリックして、**ビルド後に配布** をチェック ボックスを選択します。
4. Setup.exe ファイルにデジタル署名を行い、アプリケーションに含まれるコードが変更または破損していないことをエンド ユーザーに対して保証するには、[**セットアップにデジタル署名する**] **ハイパーリンク** をクリックします。[**セットアップにデジタル署名する**] ダイアログ ボックスが開きます。必要に応じて設定を指定します。
5. [**インストールのビルド**] をクリックします。

出力ウィンドウが開き、ビルドプロセスについての情報が表示されます。[出力] タブがログファイル情報を表示すると、ビルドが完了します。

プロジェクト アシスタントの完了後: 次のステップ

プロジェクト アシスタントの一連のページでフィールドへの入力を完了すると、インストール プロジェクトの枠組みが完成します。これは機能するインストールとして利用することができ、また必要に応じてさらにカスタマイズすることも可能です。

プロジェクトをさらにカスタマイズする

インストールデザイナから、InstallShield で利用可能なすべてのインストール作成ビューへ簡単にアクセスすることができます。[プロジェクト アシスタント] ワークスペースの上部にある [インストールデザイナ] タブをクリックしてビューを表示します。

インストール デザイナー内には、作成中のプロジェクトの種類で利用可能なビューが一覧となった [ビュー リスト] があります。ワークスペースの左側に [ビュー リスト] を表示するには、F4 を押します。

InstallShield ビューについて詳しい情報

各 InstallShield ビューの詳細については、「ビュー リファレンス」を参照してください。

InstallShield インターフェイスを使って作業する

InstallShield インターフェイスは、メニュー バー、ツールバー、ダイアログ ボックスなど通常の Windows ベースの要素をもったグラフィックなユーザー インターフェイスです。このセクションでは、これらの要素を使用した基本的なタスクの実行方法、および、インターフェイスのカスタマイズの方法について説明します。

ビュー リストを表示する

タスク ***InstallShield インターフェイスでビュー リストを表示するには、以下の手順に従います:***

1. InstallShield インターフェイス上部の **[インストール デザイナー]** タブをクリックします。
2. 以下のいずれかの方法でビュー リストを表示します。
   - **[ビュー]** メニューで、**[ビュー リスト]** をクリックします。InstallShield インターフェイスの左側に、**ビュー リスト**が表示されます。
   - ツールバーの **[ビュー リスト]** ボタンをクリックします。
   - F4 を押すと、**[ビュー リスト]** が表示または非表示になります。

Opening Views in the InstallShield ユーザー インターフェイスでビューを開く

InstallShield での多くの手順では、まず最初にインストール開発環境 (IDE)で特定のビューを開きます。

タスク ***ビューを開くには、以下の手順を実行します。***

1. **[インストール デザイナー]** タブをクリックします。IDE の左側に**ビュー リスト**が表示されます。**ビュー リスト** が表示されない場合、「ビュー リストを表示する」を参照してください。
2. **ビュー リスト**で、開くビューを選択します。使用できるすべてのビューを表示するには、[ビュー リスト] フォルダーを展開します。

様々なビューで、[グループ ボックス] 領域を使って作業する

InstallShield の多くのビューには、ビュー内の行をグループ分けすることができるグループ ボックス領域があります。このグループ ボックスを含むビューでは、行ヘッダーをグループ ボックス ([フィールドごとにグループ分けするときは、ここにフィールド名をドラッグします。] と表示される領域) にドラッグするするだけで、複数階層のグループに分けることができます。ビュー内の行は、グループ ボックスの配列に従って階層構造で表示されます。[再配布可能ファイル] ビューは、グループ ボックスを含むビューの 1 つです。

グループ ボックスを使用する際、以下の点を参考にしてください:

- 列ヘッダーをグループ ボックス領域に移動するとき、列ヘッダーをドラッグしてグループ ボックスにドロップします。
- 列ヘッダーをグループ ボックス領域にコピーするとき、CTRL キーを押しながら列ヘッダーをドラッグしてグループ ボックスにドロップします。この場合、列ヘッダーはそのままの位置に残り、またグループ ボックスにも表示されます。
- グループ ボックスのヘッダーをグループ ボックス領域にドロップするとき、別の列ヘッダー上にドロップすることができます。これによって、行が階層構造で表示されます。
- 列ヘッダーをグループ ボックスから削除するには、グループ ボックス領域からそれをドラッグして列ヘッダーの行にドロップします。列ヘッダーの行をドラッグすると、それをドロップしたときに列ヘッダー行のどの位置に表示されるのかが、矢印で示されます。
- グリッド内のアイテムを特定の列ヘッダーごとに並べ替えるには、グループ ボックス内または列ヘッダー行内の列ヘッダーをクリックします。

次の例を使って、ビュー内のコンテンツをグループ ボックスを使ってグループ化する様々な方法をデモンストレーションします。

デフォルトの動作: グループ ボックス領域を空白にする

デフォルトで、グループ ボックスに列ヘッダーは表示されません。次のスクリーン ショットは、[再配布可能ファイル] ビューの一部です。項目が [名前] 列で整列しています。

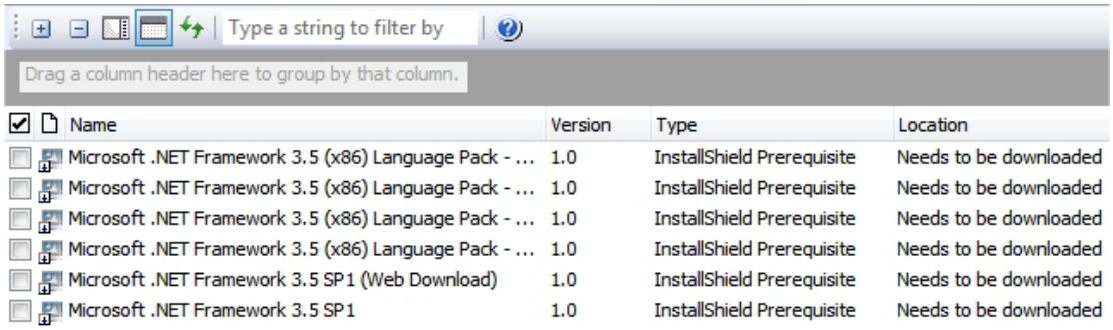


図 2-3: [グループ ボックス] 領域を空白にする

1 つの列ヘッダーでグループ化する

CTRL を押しながらグループ ボックスに列ヘッダーをドラッグ アンド ドロップするとき、グリッドの行が項目ごとにグループ化されます。次のスクリーン ショットは、[再配布可能ファイル] ビューの一部です。行は、プロジェクトに追加された再配布可能ファイル (チェック ボックスが選択状態になっている再配布可能ファイル) を識別しやすいように編成されています。

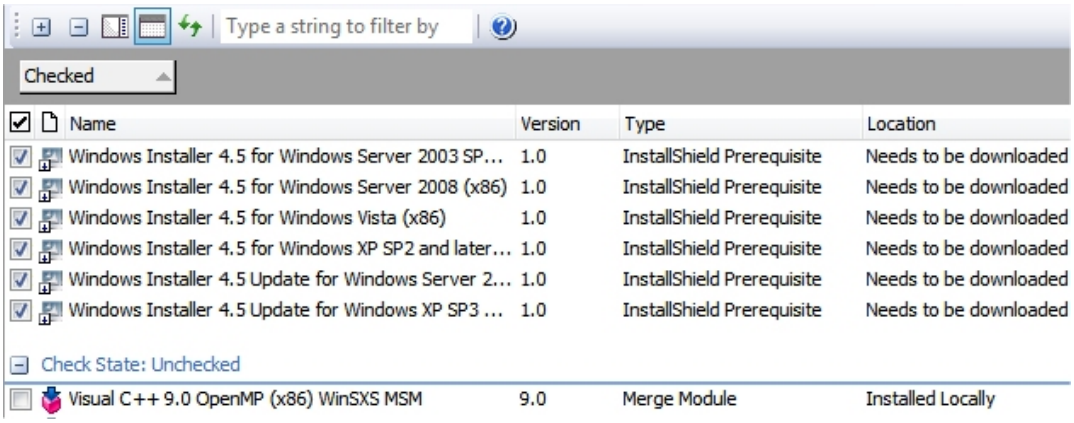


図 2-4: 1 つの列ヘッダーで行をグループ化する

2 つの列ヘッダーでグループ化する

CTRL を押しながらグループ ボックスに列ヘッダーをドラッグ アンド ドロップするとき、グリッドの行が複数の項目ごとにグループ化されます。次のスクリーン ショットは、[再配布可能ファイル] ビューの一部です。プロジェクトに追加された InstallShield 前提条件とマージ モジュールが、識別しやすくグループ化されます。

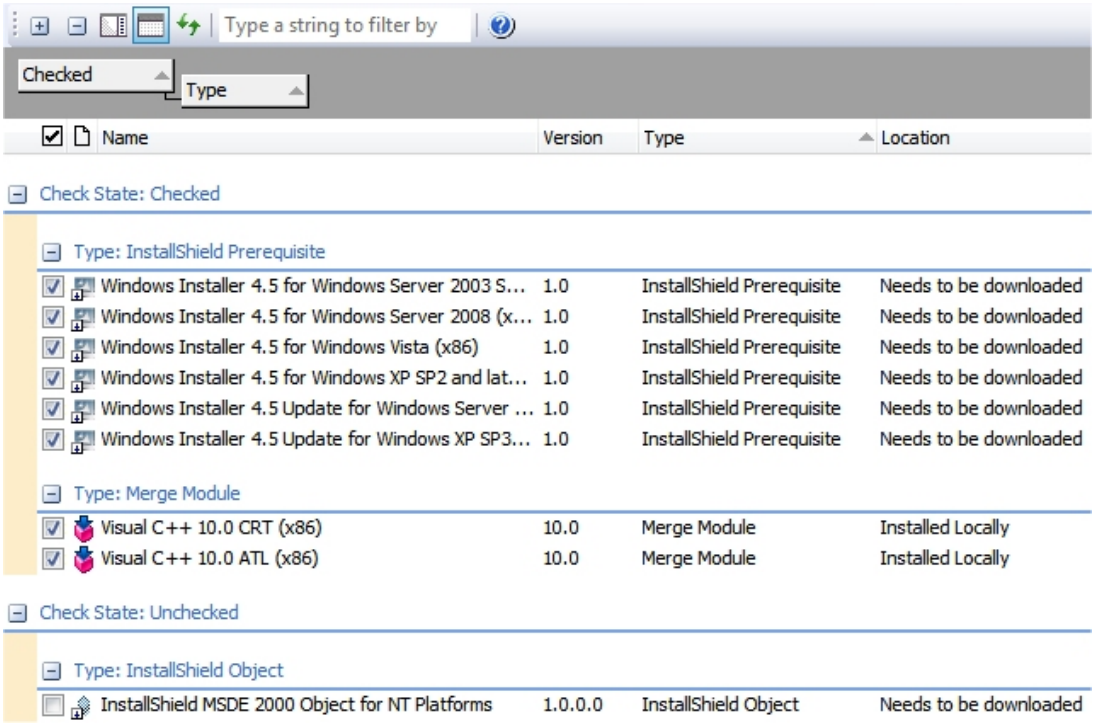


図 2-5: チェック ボックス列および種類列ごとに行を並べ替える

ツールバーの表示または非表示

タスク ***ツールバーを表示または非表示にするには、以下の手順を実行します。***

- ツールバーを右クリックして表示する、または非表示にするツールバーを選択します。
- **[ツール]** メニューで、**[カスタマイズ]** をクリックします。**[カスタマイズ]** ダイアログ ボックスが開きます。表示する各ツールバーのチェック ボックスを選択します。非表示にする各ツールバーのチェック ボックスを選択解除します。

ツールバーにボタンおよびメニューを追加する

タスク ***ツールバーにボタンまたはメニューを追加するには、以下の手順を実行します。***

1. 変更するツールバーが表示されていることを確認します。
2. **[ツール]** メニューで、**[カスタマイズ]** をクリックします。**[カスタマイズ]** ダイアログ ボックスが開きます。
3. **[コマンド]** タブをクリックします。
4. **[カテゴリ]** ボックスで、追加するボタンまたはメニューのカテゴリをクリックします。
5. **[コマンド]** ボックスから、ボタンまたはメニューを適切なツールバーへドラッグします。

ヒント: *独自のカスタムツールバーを作成するには、ボタンまたはメニューをツールバーの近くにある灰色の空白部分にドラッグします。*

ツールバーからボタンおよびメニューを削除する

タスク ***ツールバーからボタンまたはメニューを削除するには、以下の手順を実行します。***

1. 変更するツールバーが表示されていることを確認します。
2. **[ツール]** メニューで、**[カスタマイズ]** をクリックします。**[カスタマイズ]** ダイアログ ボックスが開きます。
3. 削除するボタンまたはメニューを右クリックして、**[削除]** をクリックします。

カスタム ツールバーの作成

タスク ***カスタム ツールバーを作成するには、以下の手順を実行します。***

1. **[ツール]** メニューで、**[カスタマイズ]** をクリックします。**[カスタマイズ]** ダイアログ ボックスが開きます。
2. **[ツールバー]** タブをクリックします。
3. **[新規]** ボタンをクリックします。**[新規ツールバー]** ダイアログ ボックスが開きます。
4. **[ツールバーの名前]** テキスト ボックスにツールバーの説明的な名前を入力し、**[OK]** をクリックします。
5. メニューまたはボタンを追加して、新規作成したツールバーをカスタマイズします。

[出力] ウィンドウを固定する/取り外す

[出力] ウィンドウ、またその個別のタブは、InstallShield のワークスペースの任意の側面に固定するか、独立した場所にドラッグすることができます。

[出力] ウィンドウまたはそのタブの 1 つを InstallShield インターフェイスの端にドラッグすると、固定ウィンドウとして表示されます。[出力] ウィンドウまたはそのタブの 1 つを InstallShield インターフェイスの端から離れた場所にドラッグすると、取り外されます。

タスク ***[出力] ウィンドウを取り外すには、以下の手順に従います:***

[出力] ウィンドウのタイトル バーを新しい場所にドラッグします。必要に応じて、[出力] ウィンドウのサイズを変更します。

タスク ***[出力] ウィンドウを固定するには、以下の手順に従います:***

[出力] ウィンドウのタイトル バーを InstallShield インターフェイスの右、左、上、下の端にドラッグします。

タスク ***[出力] ウィンドウのタブを取り外すには、以下の手順に従います:***

タブを新しい場所にドラッグします。必要に応じて、[出力] ウィンドウのサイズを変更します。

タスク ***[出力] ウィンドウのタブの 1 つを固定するには、以下の手順に従います:***

タブを InstallShield インターフェイスの右、左、上、下の端にドラッグします。

InstallShield の詳細設定を構成する

InstallShield プログラム ファイルの中に Settings.xml というファイルがあります。このファイルで、InstallShield のためのマシン全体に関する詳細設定が一部含まれています。InstallShield をインストールしたとき、使用している InstallShield の言語に応じて Settings.xml が次の場所の 1 つにインストールされます。

- **英語**—*InstallShield Program Files フォルダー*¥Support¥0409
- **日本語**—*InstallShield Program Files フォルダー*¥Support¥0411

通常、Settings.xml ファイルの変更は推奨されません。ただし、あるケースでは、このファイルでの変更が必要になる場合があります。このセクションでは、Settings.xml の変更が必要にある場合のシナリオがいくつか説明されています:

- デジタル署名のタイムスタンプ サーバーを変更する
- Setup.exe にストリームされるファイルの圧縮レベルを構成する
- .cab ファイルの最大サイズを構成する
- デジタル署名のプラットフォーム アーキテクチャを構成する

注意: *Settings.xml ファイルには重要なデータが含まれているため、ファイルが間違って編集された場合、InstallShield が正しく動作しなくなることがあります。このファイルを編集する場合、十分な注意が必要です。*

デジタル署名のタイムスタンプ サーバーを変更する

プロジェクト: *この情報は、次のプロジェクトの種類に適用します :*

- *Express*
- *QuickPatch*

リリースのデジタル署名情報を指定すると、InstallShield はビルド中に、VeriSign のサーバー (**http://timestamp.verisign.com/scripts/timstamp.dll**) をデフォルトのタイムスタンプ サーバーとして使用します。InstallShield には、そのデフォルト サーバーを異なるタイム スタンプ サーバーに変更できるマシン全体の設定があります。この設定を使って、タイムスタンプを無効にすることもできます。

InstallShield 2019 R2 以降の Express Editionでは、連続するデジタル署名の間に遅延を設定することができます。この機能は、タイムスタンプ サーバーが連続する署名要求の処理に失敗した場合のみ必要になります。

Settings.xml の <DevStudio/Build> ノードの下にある <DelayBetweenSigning default="1500"/> ノードで、ミリ秒単位で指定する必要があります。

注意: *次の手順では、InstallShield と共にインストールされている Settings.xml ファイルを変更する必要があります。このファイルには重要なデータが含まれているため、ファイルが間違って編集された場合、InstallShield が正しく動作しなくなることがあります。このファイルを編集する場合、十分な注意が必要です。*

タスク *デジタル署名のタイムスタンプ サーバーを構成するには、以下の手順に従います:*

1. InstallShield を閉じます。
2. InstallShield と共にインストールされている Settings.xml ファイルを見つけます。Settings.xml は、使用している InstallShield の言語に応じて、次のいずれかの場所にインストールされています:
   - **英語**—*InstallShield Program Files フォルダー*¥Support¥0409
   - **日本語**—*InstallShield Program Files フォルダー*¥Support¥0411
3. あとで元のバージョンに戻す必要があるときのために、Settings.xml ファイルのバックアップ コピーを作成します。
4. テキスト エディターまたは XML ファイル エディターを使って、Settings.xml ファイルを開きます。
5. <DigitalSignature> 要素を検索します。要素は次のように表示されています:

   <DigitalSignature Timestamp="http://timestamp.verisign.com/scripts/timstamp.dll"/>
6. 別のタイムスタンプ サーバーでオーバーライドするには、Timestamp 属性の値を適切な URL に設定します。

   タイムスタンプを無効にするには、Timestamp 属性の値を空白にします:

   <DigitalSignature Timestamp=""/>

   メモ: *タイムスタンプを無効にすると、デジタル署名の有効期間に影響します。*
7. Settings.xml ファイルを保存します。
8. XML コードが適切に構成されていることを確認してください。不適切なコードは、InstallShield で問題が発生する可能性があります。ほとんどの場合、インターネット エクスプローラーで Settings.xml ファイルを開いて、不適切に書かれた XML コードを確認することができます。ファイル内にある主な要素は、縮小、展開が可能です。これらが不可能な場合、コードにエラーが無いか確認してください。

デジタル署名情報を含むリリースをビルドするたびに、構成された設定に基づいて InstallShield によってタイムスタンプが設定されます。

Setup.exe にストリームされるファイルの圧縮レベルを構成する

InstallShield には、ビルド時に Setup.exe ファイルにストリームされるファイルに使用する圧縮レベルを指定することができる設定が含まれています (マシン全体に適用されます)。以下は、Setup.exe ファイルにストリームすることができるファイルの例です:

- すべての製品ファイル (すべてのファイルが Setup.exe セットアップランチャーに圧縮されるリリースの場合)
- "Setup.exe から抽出" の場所がある InstallShield 前提条件インストール
- "Setup.exe から抽出" の場所がある .NET Framework インストール
- "Setup.exe から抽出" の場所がある Windows Installer インストール

InstallShield では、ファイルが Setup.exe ファイルにストリームされるとき、圧縮から除外するファイルを指定することができます (ワイルドカード文字の使用可)。

注意: *次の手順では、InstallShield と共にインストールされている Settings.xml ファイルを変更する必要があります。このファイルには重要なデータが含まれているため、ファイルが間違って編集された場合、InstallShield が正しく動作しなくなることがあります。このファイルを編集する場合、十分な注意が必要です。*

タスク ***ストリームされたファイルの圧縮設定を構成するには、以下の手順に従います:***

1. InstallShield を閉じます。
2. InstallShield と共にインストールされている Settings.xml ファイルを見つけます。Settings.xml は、使用している InstallShield の言語に応じて、次のいずれかの場所にインストールされています:
   - **英語**—*InstallShield Program Files フォルダー*¥Support¥0409
   - **日本語**—*InstallShield Program Files フォルダー*¥Support¥0411
3. あとで元のバージョンに戻す必要があるときのために、Settings.xml のバックアップ コピーを作成します。
4. テキスト エディターまたは XML ファイル エディターを使って、Settings.xml ファイルを開きます。
5. <StreamCompression> 要素を検索します。要素は次のように表示されています:

   <StreamCompression exclude="*.CAB" compressionlevel="-1"/>
6. ファイルが Setup.exe ファイルにストリームされるとき、特定のファイルまたはファイルの種類を圧縮から除外する場合、除外属性の値をこれらのファイルの名前に設定します。次の事項に注意してください。
   - 複数のファイルを指定する場合、各ファイル名をカンマで区切ります。
   - ワイルドカード文字の指定には、アスタリスク (*) を使用します。

   たとえば、.cab ファイル、.exe ファイル、および **test.txt** という名前のファイルを圧縮から除外するように指定する場合、除外属性の値を次のように設定します:

   <StreamCompression exclude="*.CAB,*EXE,test.txt" compressionlevel="-1"/>

   .cab ファイルは圧縮ファイルであるため、デフォルトの値は ***.CAB** に設定されています。
7. 以下のいずれかを実行します。
   - 圧縮ファイルのサイズと実行時に圧縮ファイルを展開するために必要な時間のバランスを考慮に入れた圧縮レベルを使用する場合、compressionlevel 属性の値を **-1** に設定します。これがデフォルトの値です。
   - 特定の圧縮レベルを指定する場合、compressionlevel 属性の値に 0 から 9 の間の数値を指定します。0 は圧縮なしを意味し、9 が最大圧縮を意味します。

     一般的に、0 から 9 の値を指定した場合、指定した値が大きいほど、圧縮されたファイルのサイズは小さくなり、実行時にファイルを展開するときの時間が長くなります。
8. Settings.xml ファイルを保存します。
9. XML コードが適切に構成されていることを確認してください。不適切なコードは、InstallShield で問題が発生する可能性があります。ほとんどの場合、インターネット エクスプローラーで Settings.xml ファイルを開いて、

不適切に書かれた XML コードを確認することができます。ファイル内にある主な要素は、縮小、展開が可能です。これらが不可能な場合、コードにエラーが無いか確認してください。

圧縮されたリリースをビルドしたとき、構成した設定に従って、ファイルが Setup.exe ファイルにストリームされます。

.cab ファイルの最大サイズを構成する

メモ: *この情報は、すべてのファイルが単一ファイルの .msi パッケージまたは Setup.exe セットアップランチャーに埋め込まれている圧縮済みネットワーク イメージ リリースをビルドしている場合のみ適用します。*

.cab ファイルには、いくつかの制限事項があります。たとえば、単一 .cab ファイルの最大サイズは 2 GB に設定されています。また、サイズの大きい .cab ファイルを署名しようしたとき、およびサイズの大きい署名済み .cab ファイルのデジタル署名を検証しようとしたときにトラブルが生じた経験があるユーザーもいるかもしれません。

これらの制限事項のワークアラウンドとして、InstallShield では、圧縮 SingleImage タイプのリリースに対してビルドされる各 .cab ファイルの最大サイズを指定することができます (設定はマシン全体に適用されます)。InstallShield で、リリースの .cab ファイルを作成しているとき、構成した .cab ファイルのしきい値に達すると、データが 2 つ以上の.cab ファイルに分割され、マルチパートの .cab ファイルが作成されます。InstallShield でマルチパート .cab ファイルを作成しない場合、単一の .cab ファイルにデータを格納するように構成できます。

注意: *次の手順では、InstallShield と共にインストールされている Settings.xml ファイルを変更する必要があります。このファイルには重要なデータが含まれているため、ファイルが間違って編集された場合、InstallShield が正しく動作しなくなることがあります。このファイルを編集する場合、十分な注意が必要です。*

タスク ***InstallShield でマルチパート .cab ファイルを作成するかどうか、また .cab ファイルの最大サイズを指定する場合、以下の手順に従います:***

1. InstallShield を閉じます。
2. InstallShield と共にインストールされている Settings.xml ファイルを見つけます。Settings.xml は、使用している InstallShield の言語に応じて、次のいずれかの場所にインストールされています:
   - **英語**—*InstallShield Program Files フォルダー*¥Support¥0409
   - **日本語**—*InstallShield Program Files フォルダー*¥Support¥0411
3. あとで元のバージョンに戻す必要があるときのために、Settings.xml のバックアップ コピーを作成します。
4. テキスト エディターまたは XML ファイル エディターを使って、Settings.xml ファイルを開きます。
5. <CompressedNetworkCABSize> 要素を検索します。要素は次のように表示されています:

   <CompressedNetworkCABSize default="600"/>
6. 以下のいずれかを実行します。
   - .cab ファイルの最大サイズを設定する場合、デフォルト属性の値としてサイズを MB で入力します。上記の例では、最大サイズは **600** に設定されています。.cab ファイルの最大サイズは 2 GB (2048 MB) に設定されているため、値は 2048 以下でなければなりません。

デフォルト値は 600です。

- InstallShield でマルチパート .cab ファイルを作成しない場合、デフォルト属性の値を -1 に設定します。

7. Settings.xml ファイルを保存します。
8. XML コードが適切に構成されていることを確認してください。不適切なコードは、InstallShield で問題が発生する可能性があります。ほとんどの場合、インターネット エクスプローラーで Settings.xml ファイルを開いて、不適切に書かれた XML コードを確認することができます。ファイル内にある主な要素は、縮小、展開が可能です。これらが不可能な場合、コードにエラーが無いか確認してください。

アプリケーション プロジェクト タイプの 1 つに圧縮された SingleImage リリースをビルドしたとき、Settings.xml ファイルで構成した要件に従って .cab ファイルが作成されます。Settings.xml ファイルで指定した値に応じて、.cab ファイルが .msi パッケージのMedia テーブルに表示されます。

デジタル署名のプラットフォーム アーキテクチャを構成する

ファイルのデジタル署名に使用するプラットフォーム アーキテクチャを選択できます。

タスク ***デジタル署名のプラットフォーム アーキテクチャを構成するには、以下の手順に従います:***

1. **InstallShield** を閉じます。
2. InstallShield と共にインストールされている **Settings.xml** ファイルを検出します。**Settings.xml** は、使用している InstallShield の言語に応じて、次のいずれかの場所にインストールされています:
   - **英語**—*InstallShield Program Files フォルダー*¥Support¥0409
   - **日本語**—*InstallShield Program Files フォルダー*¥Support¥0411
3. 後で元のバージョンに戻す必要があるときのために、**Settings.xml** ファイルのバックアップ コピーを作成しておきます。
4. テキスト エディターまたは XML ファイル エディターを使って、**Settings.xml** ファイルを開きます。

   メモ: *ファイルが編集のための管理者モードで開きます。*

5. **<DigitalSignature>**. タグを検索します。実際のタグは **<DigitalSignature Platform="X86"/>** です。
6. 異なるプラットフォーム アーキテクチャを使用する場合、"プラットフォーム" を X86 または X64 のどちらかに設定します。
   たとえば、
   - 32ビット署名の場合: <DigitalSignature Platform="X86"/>
   - 64ビット署名の場合: <DigitalSignature Platform="X64"/>

メモ: 無効なエントリの場合、自動的に 32 ビット 署名が使用されます。

7. Settings.xml ファイルを保存します。

メモ: 32 ビットのオペレーティング システム上で 64 ビット署名を使用することはできません。

InstallShield の以前のバージョンからの移行する

以前のバージョンの InstallShield または InstallShield Express で作成したセットアップ プロジェクトがある場合、新しいバージョンの InstallShield でそれらのプロジェクトを開くことができます。以前のバージョンと InstallShield 2021 Express Edition では基本的な技術が異なるため、古いプロジェクトから新しいプロジェクトへのアップグレード プロセスにおいて、必ずしも 1 対 1 の相関関係で新規プロジェクトが作成されるわけではありません。たとえば、以前のバージョンの InstallShield Express に存在していた InstallShield オブジェクトは、マージ モジュールによって置換されます。場合によっては、以前のプロジェクトに 2 つのオブジェクトがあった場合でも、新しいプロジェクトでは 1 つのマージ モジュールに置換される場合があります。

InstallShield の以前のバージョンからの移行については、ドキュメントの本セクションを参照してください。

- InstallShield 2014 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2013 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2011 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2010 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2009 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 2008 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield 12 Express Edition 以前からのプロジェクトをアップグレードする
- InstallShield Express 2.x からプロジェクトをアップグレードする

InstallShield 2014 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2014 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2014 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .775 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .775 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2014 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

ターゲット システムとしてサポートされている Windows のバージョン リストに関する変更

今回より、Windows XP SP3 および Windows Server 2003 SP2 が、InstallShield で作成されたインストールを実行するターゲット システムで必要な Windows の最低バージョンです。

.spc および .pvk ファイルを使ったデジタル署名サポートの削除

InstallShield で、ビルド時に .spc および .pvk ファイルにデジタル署名を行うサポートは、今後使用できません。

InstallShield 2014 Express Edition 以前で、リリースまたは QuickPatch パッケージに .spc および .pvk ファイルを使って実行時にデジタル署名を行うように構成済みで、そのプロジェクトを InstallShield 2021 Express Edition で開こうとすると、アップグレード警告 -6048 (リリース) または -6050 (QuickPatch プロジェクト) が表示されます。この警告は、アップグレード中に InstallShield によって .pvk ファイルおよび関連パスワードがプロジェクトから削除されることを説明します。

InstallShield 2021 Express Edition でリリースまたパッチを正しくビルドするためには、リリースまたは QuickPatch プロジェクトから .spc の参照を削除する必要があります。これを、.pfx 証明書または証明書ストアにある証明書への参照と入れ換えることができます。詳しくは、次を参照してください:

- デジタル署名とセキュリティ
- ビルド時にリリースとそのファイルにデジタル署名を行う
- QuickPatch パッケージに署名する

.spc の参照を削除せずにリリースまたはパッチのビルドを試みると、.spc ファイルの削除が必要であることを通知するビルド エラー -7347 が表示されます。

.spc ファイルおよび .pvk ファイルを .pfx ファイルに変換する方法については、「デジタル署名とセキュリティ」を参照してください。

InstallShield インストールからの SignTool.exe および Signcode.exe の削除

InstallShield をインストールしたとき、今回より SignTool.exe および Signcode.exe は開発マシンにインストールされません。手動でファイルにデジタル署名を行う場合、SignTool.exe の使用を考慮してください。SignTool.exe は、Microsoft Windows Software Development Kit (SDK) に含まれていて、Visual Studio と共にインストールされます。

Trialware サポート

InstallShield は今後、Try and Buy/プロダクト アクティベーション タイプの Trialware を作成するためのサポートを含みません。今後、Trialware ビューは InstallShield に含まれていません。

InstallShield 2013 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2013 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2013 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .774 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .774 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2013 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2012 Spring Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2012 Spring Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .773 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .773 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2012 Spring Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

すべてのプロジェクトに影響する変更 (新規プロジェクトおよびアップグレードされたプロジェクト)

ターゲット システムの要件における変更

InstallShield は、今後 Windows 2000 システム用のインストール作成をサポートしません。

InstallShield は、今後 モバイル デバイス用のインストール作成をサポートしません。そのため、[モバイル デバイス] ビューおよびスマート デバイス プロジェクト タイプは、今回より InstallShield に含まれていません。スマート デバイス プロジェクトを InstallShield 2012 Spring Express Edition 以前から InstallShield 2021 Express Edition にアップグレードしようとすると、InstallShield 2021 Express Edition がエラー メッセージを表示して、プロジェクトを開くことができません。InstallShield 2012 Spring Express Edition 以前から InstallShield 2021 Express Edition にプロジェクトをアップグレードして、プロジェクトがデスクトップ プラットフォームをターゲットとし、モバイル デバイス サポートを含む場合、InstallShield はアップグレード中にモバイル デバイス サポートを削除して警告をログ記録します。

InstallShield を実行するシステムの要件に関する変更

InstallShield を実行するための最小オペレーティング システム要件は、Windows XP SP3 または Windows Server 2003 SP2 です。以前、オペレーティング システムの最小要件は、これらのオペレーティング システムのどちらかの RTM バージョンでした。

InstallShield 2012 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2012 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2012 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .772 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .772 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2012 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

InstallShield 2011 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2011 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2011 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .771 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .771 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2011 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

すべてのプロジェクトに影響する変更 (新規プロジェクトおよびアップグレードされたプロジェクト)

このセクションでは、新規プロジェクトおよび InstallShield の以前のバージョンからアップグレードされたプロジェクトに影響する変更について説明されています。

Express プロジェクトにおけるビルド警告 -7235

デフォルトで、ソフトウェア識別タグ機能は、すべての Express プロジェクトで有効になっています。これは、InstallShield 2021 Express Edition で作成された新しいプロジェクト、および InstallShield の以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクトに適用します。

Express プロジェクトで、必須識別タグ設定 ([一般情報] ビューの "一意な ID"、"タグ作成者"、"タグ作成者 ID" 設定) にデータを入力せずにリリースをビルドすると、そのプロジェクトでタグ機能を有効のままにしておいた場合、ビルド警告 -7235 が発生します。このビルド警告では、特定の必須タグが空白であるために、ソフトウェア識別タグが作成されず、インストールに含まれなかったことが通知されます。この警告を解決するには、[一般情報] ビューで、各設定に適切な値を入力するか、または "ソフトウェア識別タグの使用" 設定で [いいえ] を選択します。

COM 抽出の変更

InstallShield では、COM 抽出時に新しい監視方式をサポートします。Windows Vista 以降のシステムまたは Windows Server 2008 以降のシステム上で、InstallShield を使用している場合、この新しい方式がデフォルトとなります。この方法は、カーネル ドライバーを使って、ビルド時のダイナミック COM 抽出中、およびデザイン時のスタティック COM 抽出中に変更されたレジストリ領域を監視します。この新しい方式は、DLL が既存のレジストリ エントリを読み込んでビルド マシンへの変更を妨げる以前の方法の利点を組み合わせたものです。

必要な場合、UseAPIRegistryHooks レジストリ値 (32 ビット マシンの場合は HKEY_LOCAL_MACHINE¥SOFTWARE¥InstallShield¥RegSpy レジストリ キーに含まれる、64 ビット マシンの場合は HKEY_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥InstallShield¥RegSpy レジストリ キーに含まれる) の値データを設定して 3 つの異なる COM 抽出方式を切り替えることができます。使用可能な REG_DWORD 値データ:

- 0—API フックを使って、既存 DLLのレジストリ エントリを読み取ります。
- 1—レジストリのリダイレクトを使って、ビルド マシン上の登録済み DLL への変更を防止します。値を設定しなかった場合、これが Windows XP および Windows Server 2003 システム上でのデフォルト動作となります。
- 2—新しいカーネル モードの監視を使って、2 つのメソッドの両方の利点を組み合わせます。値が設定されていない場合、これが Windows Vista 以降および Windows Server 2008 以降のシステム上でのデフォルト動作となります。

この機能は Express プロジェクトに適用します。

InstallShield 2010 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2010 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2010 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .770 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .770 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2010 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

すべてのプロジェクトに影響する変更 (新規プロジェクトおよびアップグレードされたプロジェクト)

このセクションでは、新規プロジェクトおよび InstallShield の以前のバージョンからアップグレードされたプロジェクトに影響する変更について説明されています。

Visual Studio 2003 以前の統合サポートの終了

Visual Studio 内部から直接 InstallShield プロジェクトを作成、編集、およびビルドするには、Visual Studio 2005 以降が必要です。InstallShield を Visual Studio 2003 以前と統合することはできなくなりました。

InstallShield は Setup.exe および Update.exe の Unicode バージョンのみをビルド (今回より、ANSI バージョンは作成できません)

今回より、ではすべての Setup.exe と Update.exe ファイルが Unicode でビルドされます。これは、InstallShield 2021 Express Edition で作成されたすべての新しい Express プロジェクトと QuickPatch プロジェクトに適用します。また、InstallShield の以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたすべてのプロジェクトにも適用します。このため、セットアップランチャーを Unicode バージョンまたは ANSI バージョンのどちらでビルドするかを指定するためのこれまでの設定は削除されました:

- Express プロジェクトの [リリース] ビューにあるリリースの [Setup.exe] タブにあった "セットアップ ランチャの種類 " 設定が削除されました。
- QuickPatch プロジェクトの [一般情報] ビューの [ビルドの設定] 領域にある [詳細] タブから、"アップデート ランチャの種類" 設定が削除されました。

[リリース] ビューの [ポストビルド] タブの新しい名前

[リリース] ビューでリリースを選択すると表示されるタブの 1 つである [ポストビルド] タブの名前が変更されました。今回より、このタブは [イベント] タブと呼ばれます。この変更は、情報提供を目的として報告されています。

新規プロジェクトに影響し、アップグレードされたプロジェクトに影響しない変更

このセクションでは、潜在的に新規プロジェクトに影響し、以前のバージョンからアップグレードされたプロジェクトには影響しない InstallShield の変更について説明されています。アップグレードされたプロジェクトの場合、手動による変更が必要になる場合があります。

既存のプロジェクトにおける既存の InstallShield 前提条件のデザイン時およびビルド時の場所変更

今回より、InstallShield が InstallShield 前提条件ファイル (.prq ファイル)、その関連データ ファイル、および依存関係を検索するフォルダーを指定できるようになりました。以前は、InstallShield が .prq ファイルを検索するフォルダーは、*InstallShield Program Files フォルダー*¥SetupPrerequisites のみでした。

InstallShield 前提条件を *InstallShield Program Files フォルダー*¥SetupPrerequisites フォルダの場所から、[オプション] ダイアログ ボックスの [プロパティ] タブで定義した新しいカスタム場所 (またはここで検索パスを定義できる別の任意の場所) に移動させた場合、InstallShield 2010 Express Edition 以前のプロジェクトを InstallShield 2021 Express Edition にアップグレードするときに以下の手順が必要な場合があります:

1. [再配布可能ファイル] ビューで、プロジェクトに含まれているが、カスタム場所に存在する各 InstallShield 前提条件のチェック ボックスをクリアします。そのデータ ファイルまたは依存関係がデフォルトの場所からカスタム場所に移動されている各 InstallShield 前提条件のチェック ボックスもクリアします。
2. 新しい [更新] ボタンをクリックします。
3. 手順 1 で、プロジェクトから削除した各 InstallShield 前提条件のチェック ボックスを選択します。

InstallShield がプロジェクトの ISSetupPrerequisites テーブルから前提条件のパスを削除します。InstallShield 2010 Express Edition 以前のプロジェクトでは、このテーブルには完全パスが格納されました。前提条件のチェック ボックスをクリアしてから、[更新] ボタンをクリックしないで再び選択した場合、InstallShield は ISSetupPrerequisites テーブルにファイル名のみではなく、引き続き完全パスを使用します。

InstallShield 2010 Express Edition 以前のプロジェクトを InstallShield 2021 Express Edition にアップグレードした場合で、InstallShield 前提条件の場所を変更してから、プロジェクトにその前提条件を追加した場合、この [更新] 手順を行う必要はありません。また、InstallShield 2021 Express Edition を使って新しいプロジェクトを作成した場合も、[更新] 手順は不要です。両方の状況において、InstallShield はプロジェクトの ISSetupPrerequisites テーブルにパスを含みません。このため、デフォルト パスではなく、カスタム検索パスを利用することができます。

InstallShield 2009 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2009 Express Edition および以前のバージョンで作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードする際に発生する可能性のある問題についての情報です。また、新しい InstallShield 2021 Express Edition プロジェクトと InstallShield 2009 Express Edition および以前のバージョンから InstallShield 2021 Express Edition にアップグレードされたプロジェクト間の潜在的な動作の違いについてもアラートします。

InstallShield Express Edition の以前のバージョンで作成されたプロジェクトのアップグレードに関する一般情報

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .768 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .768 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield Express Edition で開くことはできませんので注意してください。

InstallShield Express Edition バージョン (InstallShield 2009 Express Edition 以前、InstallShield 12 Express Edition 以前、および InstallShield Express 5 以前) で作成した既存プロジェクトを InstallShield 2021 Express Edition にアップグレードすることができます。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition にアップグレードすることはできませんので、ご注意ください。

InstallShield の複数エディションをインストールする

InstallShield 2021(InstallShield Premier、InstallShield、または Express) は、同時に同じシステム上に 1 つのエディションのみをインストールできます。以前は、 バージョンの Premier Edition および InstallShield がインストールされた同じシステム上に Express Edition をインストールすることが可能でした。

InstallShield を実行できるサポート対象オペレーティング システムのリストにおける変更

今回より、InstallShield (オーサリング環境) を実行するシステムの最小オペレーティング システム要件は、Windows XP または Windows Server 2003 です。以前、最小オペレーティング システム要件は Windows 2000 XP3 でした。

すべてのプロジェクトに影響する変更 (新規プロジェクトおよびアップグレードされたプロジェクト)

このセクションでは、新規プロジェクトおよび InstallShield の以前のバージョンからアップグレードされたプロジェクトに影響する変更について説明されています。

Setup.exe が Windows 9x、Windows NT4、または Windows システムで実行不可能となりました

InstallShield で作成された Setup.exe インストールは、Windows 9x、Windows NT4、または Windows Me 上で実行不可能となります。エンド ユーザーが Setup.exe を Windows 9x または Windows Me システム上で実行しようとすると、「*FullSetup.exePathAndFileName* ファイルには、Windows の新しいバージョンが必要です。 Windows のバージョンをアップグレードしてください。」というエラー メッセージが表示されます。Windows のバージョンをアップグレードしてください。」というエラー メッセージが表示されます。Windows NT4 システム上では、Windows はメッセージボックスを表示して、「*FullSetup.exePathAndFileName* は有効な Windows NT アプリケーションではありません。」とエラーを通知します。

InstallShield では、機能またはカスタム アクションの条件を作成するとき、あるいは [要件] ビューでシステム ハードウェア要件を構成するときに、これらのレガシー オペレーティング システムを選択することができなくなりました。InstallShield 2009 Express Edition 以前で作成された InstallScript プロジェクトを InstallShield 2021 Express Edition にアップグレードしたときに、以前のプロジェクトのオペレーティング システムの設定にこれらのレガシー オペレーティング システムのみへの参照が含まれている場合、レガシー オペレーティング システムのオプションが、[任意の OS バージョン] オプションで置き換えられます。

Windows Installer 1.x 再配布可能ファイルが使用できなくなりました

Windows Installer 1.x 再配布可能ファイルは、サポートされていない Windows のレガシー バージョンのみをターゲットとするため、今後 InstallShield では使用できなくなりました。以前は、[リリース] ビューを使って、プロジェクトに Windows Installer 1.x 再配布可能ファイルを追加することが可能でした。

VBScript ランタイム ファイルの再配布可能ファイルが使用できなくなりました

InstallShield では、今回より、VBScript ランタイム ファイル用の InstallShield オブジェクトが提供されていません。この再配布可能ファイルは、今ではサポートされていない Windows の古いバージョンをターゲットとします。

QuickPatch の作成

InstallShield は、QuickPatch リリースの作成時に Windows Installer 4.5 パッチ テクノロジを使用します。この変更は、情報提供を目的として報告されています。

[再配布可能ファイル] ビューの変更

[再配布可能ファイル] ビューには、強力な検索機能と組織化機能を提供する新しいツールバーとグループ ボックス領域が追加されました。このビューの新しい [詳細の表示] ボタンを使って、選択した再配布可能ファイルについての詳細ペインの表示/非表示を切り替えることができます。詳細ペインで、再配布可能ファイルがインストールするファイルなどの情報を確認することができます。[詳細の表示] ボタンは、以前、このビューの右上にあった [詳細の表示] および [詳細の非表示] リンクの代わりとなります。

新しいグループ ボックス領域は、[再配布可能ファイル] ビューの新しいツールバーの下にあります。列ヘッダーを、このグループ ボックス領域にドラッグ アンド ドロップして、再配布可能ファイルのリストを階層形式で表示することができます。すべての再配布可能ファイルをチェック ボックスが選択されているアイテムと選択されていないアイテムごとに 2 つのグループに分けて表示するには、チェック ボックス列をグループ ボックス領域にドラッグします。これで、プロジェクトに含まれているすべての再配布可能ファイルを認識しやすくなります。これは、以前、任意の再配布可能ファイルを右クリックしてから [選択したアイテムのみを表示] をクリックしたときの動作と似ています。[再配布可能ファイル] ビューで [選択したアイテムのみを表示] コマンドは、今後使用できません。

詳細については、「様々なビューで、[グループ ボックス] 領域を使って作業する」を参照してください。

新規プロジェクトに影響し、アップグレードされたプロジェクトに影響しない変更

このセクションでは、潜在的に新規プロジェクトに影響し、以前のバージョンからアップグレードされたプロジェクトには影響しない InstallShield の変更について説明されています。アップグレードされたプロジェクトの場合、手動による変更が必要になる場合があります。

ファイル、フォルダー、およびレジストリ キーのアクセス許可を保護するためのサポート変更点

[一般情報] ビューに追加された "ロックダウンの設定方法" 設定を使って、プロジェクトに含まれるファイル、フォルダー、およびレジストリ キーに設定する新しいアクセス許可について、新しいカスタム InstallShield 処理または従来型の Windows Installer 処理のどちらを使用するかを指定できます。新しい [カスタム InstallShield 処理] オプションには、[従来型の Windows Installer 処理] オプションよりも多くの利点があります。

すべての新しいプロジェクトにおける、この設定のデフォルト値は [カスタム InstallShield 処理] オプションです。プロジェクトを InstallShield 2009 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードした場合、この設定のデフォルト値は [従来型の Windows Installer 処理] オプションです。

この新しい設定は、Express プロジェクトで使用できます。

詳細は、次を参照してください:

- ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護
- プロジェクトで、ロックダウン環境でのアクセス許可タイプを選択する

ユーザーごとインストールの Windows Installer 5 Beta サポートにおける [インストールの準備完了] ダイアログの変更

[ダイアログ] ビューにある "[すべてのユーザー] オプションの表示" 設定では、Windows Installer 5 のベータ版で提供されている MSIINSTALLPERUSER プロパティが使用できます。この設定を使って、エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供するかどうかを指定できます。この設定で選択した値に基づいて、[インストールの準備完了] ダイアログに、エンド ユーザーが製品をインストールする方法を指定できるボタンが含まれます。このボタンは、Windows 7 または Windows Server 2008 R2 が搭載されたシステム上でインストールが実行されたときに表示されます。

"[すべてのユーザー] オプションの表示" 設定は、[ダイアログ] ビューでメイン [ダイアログ] ノードを選択すると使用できます。以前は、このビューの [ユーザー情報] ダイアログを選択すると使用できました。

InstallShield 2021 Express Edition で新しい Express プロジェクトを作成したとき、"[すべてのユーザー] オプションの表示" 設定のデフォルト値は [いいえ] です。プロジェクトを InstallShield 2009 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードする場合、この値は次のように設定されます:

- 以前のプロジェクトで [いいえ] が選択されている場合、アップグレード後のプロジェクトでは [いいえ] が選択されます。
- 以前のプロジェクトで [はい] が選択されている場合、アップグレード後のプロジェクトでは [はい (すべてのシステム)] が選択されます。したがって、ターゲット システムに Windows 7 または Windows Server 2008 R2 が搭載されている場合、[インストールの準備完了] ダイアログに、エンド ユーザーが製品のインストール方法を指定できるボタンが含まれます。ターゲット システムが Windows Vista 以前、または Windows Server 2008 以前である場合、[ユーザー情報] ダイアログに、ユーザーが製品をインストールする方法を指定できるボタンが含まれます。

必要に応じて、値を変更できます。

Trialware サポート

Trialware ビューを含むのは InstallShield Premier Edition のみです。このエディションでは、Try and Die タイプの Trialware を作成できます。InstallShield は今後、Try and Buy/プロダクト アクティベーション タイプの Trialware を作成するためのサポートを含みません。

Compact プロジェクト

InstallShield は、今後 Compact プロジェクトをサポートしません。

Visual Studio の統合

Microsoft Visual Studio の統合は 1 回につき Express Edition の 1 バージョンとのみ可能です。システムで最後にインストールまたは修復された InstallShield のバージョンが Visual Studio の統合に使用されます。

InstallShield 2008 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition へアップグレードされたプロジェクトに影響が出る可能性がある変更についての説明です。

InstallShield Express Edition の以前のバージョンで作成されたプロジェクトをアップグレードする

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .765 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .766 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield Express Edition で開くことはできませんので注意してください。

次の InstallShield Express Edition バージョンで作成した既存プロジェクトを InstallShield 2021 Express Edition に移行することができます: InstallShield 12 Express Edition 以前と InstallShield Express 5 以前。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition に移行することはできませんの注意してください。

すべてのプロジェクトに影響する変更 (新規プロジェクトおよびアップグレードされたプロジェクト)

このセクションでは、新規プロジェクトおよび InstallShield の以前のバージョンからアップグレードされたプロジェクトに影響する変更について説明されています。

新しいリリースと新しい QuickPatch プロジェクト用の新しいデフォルト セットアップランチャーの値 : Windows Installer を含めない

新しい Express または QuickPatch プロジェクトを作成したとき、Windows Installer エンジンの再配布可能ファイルがデフォルトで含まれなくなりました:

- Express プロジェクトでは、Setup.exe タブにある "セットアップランチャー" 設定のデフォルト値は、現在 [**はい (MSI エンジンを含めない)**] が設定されています。この設定の以前のデフォルト値は [**はい (Windows NT および Windows 9x MSI エンジンを含める)**] でした。

- QuickPatch プロジェクトでは、[一般情報] ビューの [共通] タブにある **[Windows Installer 2.0 エンジンを含める]** チェック ボックスと **[Windows Installer 3.1 エンジンを含める]** チェック ボックスは、デフォルトでクリアになっています。以前、これらのチェック ボックスはデフォルトで選択されていました。

この変更は、InstallShield 2021 Express Edition で作成されたすべての新しい Express プロジェクトと QuickPatch プロジェクトに適用します。

InstallShield 2008 Express Edition 以前で作成された Express プロジェクトまたは QuickPatch プロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、上記の設定の値は自動的に変更されません。

ビルド時に Setup.exe と ISSetup.dll にストリームされるファイルの圧縮

今回より、Setup.exe セットアップランチャーを使用するリリースをビルドしたとき、InstallShield によって、ビルド時に Setup.exe ファイルにストリームされるファイルが圧縮されるようになりました。InstallShield が使用するデフォルトの圧縮レベルは、ファイルのサイズと実行時に圧縮ファイルを展開するために必要な時間のバランスをとっての目安です。これは、すべての新しいプロジェクトおよび InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードされた既存の Express プロジェクトに適用します。

圧縮レベルを変更する場合、または圧縮をしない場合、マシン全体に適用する設定を利用してデフォルトのレベルをオーバーライドすることができます。詳細については、「Setup.exe にストリームされるファイルの圧縮レベルを構成する」を参照してください。

以前、InstallShield には、ビルド時に Setup.exe ファイルにストリームされたファイルを圧縮するサポートが含まれていませんでした。このため、InstallShield 2008 Express Edition 以前でビルドされたリリースと InstallShield 2021 Express Edition のデフォルト圧縮レベルでビルドされた同じリリースを比較したとき、Setup.exe のファイル サイズに若干違いがある場合があります。また、ファイルの展開にかかる時間にも差が生じる場合があります。

マルチパート .cab ファイル

InstallShield には、今回より、ビルド時に SingleImage リリースについて作成した各 .cab ファイル に対して 600 MB がデフォルト制限として設定されています。 SingleImage リリースでは、すべてのファイルが単一ファイルの .msi パッケージまたは Setup.exe セットアップランチャーに埋め込まれています。InstallShield で、この種類のリリースについて .cab ファイルを作成しているとき、この制限に達すると、データが 2 つ以上の.cab ファイルに分割され、マルチパートの .cab ファイルが作成されます。これは、すべての新しいプロジェクトおよび InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードされた既存の Express プロジェクトに適用します。

.cab サイズの制限は必要に応じて変更することができます。InstallShield でマルチパート .cab ファイルを作成しない場合、単一の .cab ファイルを作成するように構成できます。詳細については、「.cab ファイルの最大サイズを構成する」を参照してください。

以前、InstallShield では、マルチパート .cab ファイルを作成できなかったため、.cab ファイルのサイズのビルトイン制限もありませんでした。

プロキシ サーバーのサポート

特定のファイルがターゲット システム上で必要な場合のみ、インストールがそのファイルをダウンロードするように構成できます。たとえば、Windows Installer エンジン、.NET Framework、および一部の InstallShield 前提条件が、一部またはほとんどのターゲット システム上に既存する可能性があります。これらのファイルをインストールに埋め込む代わりに、必要なファイルだけを実行時にダウンロードするようにプロジェクトを構成することができます。こうすることで、インストール全体のサイズを抑えることができます。

エンド ユーザーがプロキシ サーバーを使ってインターネットにアクセスする場合で、インストールがファイルをダウンロードするように構成されていると、今回より、インストールはダウンロード中に、Internet Explorer で手動で構成されたシステム プロキシ設定を使用するようになりました。これは、ターゲット システム上で別のブラウザーがデフォルトとして設定されている場合でも同じです。

InstallShield は、Internet Explorer の "設定を自動的に検出する" 設定をサポートしませんので、ご注意ください。(エンドユーザーが使用している Internet Explorer で、LAN 接続に対して [設定を自動的に検出する] チェックボックスが選択されているときに、インストールでファイルのダウンロードが必要な場合、ファイルのダウンロードができないため、インストールは失敗します。エンドユーザーが使用している Internet Explorer で、LAN 接続に対して [設定を自動的に検出する] チェック ボックスが選択されている可能性があるとき、ダウンロードされるように構成する代わりに、すべてのファイルをインストールに埋めこんだ方が良い場合ばあります。ファイルが埋めこまれている場合、失敗は避けられます。)ただし、InstallShield は Internet Explorer の LAN 接続用にセットアップされた自動構成スクリプト機能をサポートします。

この動作は、InstallShield 2009 におけるすべてのプロジェクト、および以前のバージョンで作成され、InstallShield 2009 にアップグレードされたプロジェクトでも同様です。

InstallShield 2008 以前では、デフォルト ブラウザーとして指定されているブラウザーで構成されているプロキシ サーバーの設定が使用されていましたが、場合によって失敗し、次のような問題が発生していました:

- Netscape 6 または 7 がデフォルト ブラウザーの場合、Netscape 4 の設定が使用される。Netscape 8 または 9 がデフォルト ブラウザーの場合、システム (Internet Explorer) の設定が使用される。
- Netscape 4 の設定が使用された場合、プロキシ サーバーの一覧のみが正しく読み込まれ、インポートされる。プロキシ バイパス一覧は読み込まれるが、正しくインポートされない。
- Internet Explorer 4 と互換性がない設定 (例、オートプロキシ スクリプト設定) はインポートされない。
- インストールでデフォルト ブラウザーの判別に使用する方法が Windows Vista と互換性がない。このため、Windows Vista で、デフォルト ブラウザーが正しく判別されない可能性があった。

新規プロジェクトに影響し、アップグレードされたプロジェクトに影響しない変更

このセクションでは、潜在的に新規プロジェクトに影響し、以前のバージョンからアップグレードされたプロジェクトには影響しない InstallShield の変更について説明されています。アップグレードされたプロジェクトの場合、手動による変更が必要になる場合があります。

Setup.exe および Update.exe ブートストラッパの Unicode バージョンを作成する機能

今回より、プロジェクトで Setup.exe セットアップランチャーを作成するとき、Unicode バージョンで作成するか、または ANSI バージョンで作成するかを指定することができます。以前、プロジェクトにセットアップランチャーが含まれているとき、常に ANSI バージョンがビルドされていました。Unicode バージョンのビルドはサポートされていませんでした。

Unicode セットアップランチャーは、ターゲット システムで 2 バイト言語のための適切なコード ページが実行されているいないにかかわらず、セットアップランチャーのユーザー インターフェイスで 2 バイト文字を正しく表示することができます。ANSI セットアップランチャーは、ターゲット システムで適切なコード ページが実行されている場合のみ、セットアップランチャー ダイアログで 2 バイト文字を正しく表示します。適切なコード ページが実行されていない場合、これらのダイアログで 2 バイト文字が文字化けして表示されます。

InstallShield 2021 Express Edition で新しい Express プロジェクトを作成すると、デフォルトのセットアップランチャーの種類は Unicode になります。また、InstallShield 2021 Express Edition で新しい QuickPatch プロジェクトを作成したときも、デフォルトのアップデート ランチャの種類は Unicode になります。

Express プロジェクトまたは QuickPatch プロジェクトを InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition へアップグレードした場合、既存のパッチのアップデート ランチャの種類はすべて ANSI です。種類は、必要に応じてオーバーライドが可能です。

ダイナミック ファイル リンク

プロジェクトにダイナミック ファイル リンクを追加、または変更するとき、InstallShield がコンポーネントを作成する方法について、新しいベスト プラクティスを使用するか、これまでと同様にディレクトリごとに 1 つのコンポーネント作成するのかを指定できます。これらのメソッドは、Express プロジェクトのダイナミック ファイル リンクに適用されます。

で新しいダイナミック ファイル リンクを作成すると、InstallShield はデフォルトでベスト プラクティス メソッドを使用します。

InstallShield 2008 Express Edition 以前で作成されたすべてのダイナミック ファイル リンクは、[1 つのディレクトリごとに 1 つのコンポーネント] メソッドを使用します。ダイナミック ファイル リンクがあるプロジェクトを InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードした場合、InstallShield は、既存するダイナミック ファイル リンクのコンポーネントを作成するとき、継続して [ディレクトリごとに 1 つのコンポーネント] メソッドを使用します。アップグレードされたプロジェクトで作成した新しいダイナミック ファイル リンクには、デフォルトでベスト プラクティス メソッドが使われます。2 つのコンポーネント作成メソッドおよび適切なメソッドの選択方法については、「ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する」を参照してください。

仮想ディレクトリのない IIS Web サイト

InstallShield は、今回より、仮想ディレクトリを一切含まない IIS Web サイトのインストールをサポートできるようになりました。このサポートは、新しい InstallShield 2021 Express Edition プロジェクトで作成されたすべての新しい Web サイトにも有効です。このサポートは、プロジェクトを InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition にアップグレードし、新しい Web サイトを追加した場合も有効です。

InstallShield 2008 Express Edition プロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、プロジェクトに既に Web サイトが含まれている場合、仮想ディレクトリのない Web サイトはインストールできません。仮想ディレクトリのない Web サイトのインストールできるようにするには、それを手動で InstallShield 2021 Express Edition プロジェクトから削除し、新しい Web サイトとして再度プロジェクトに追加する必要があります。

QuickPatch パッケージの簡素化

QuickPatch プロジェクトの [詳細] タブにある新しい "QuickPatch の簡素化" 設定は、InstallShield で QuickPatch パッケージのビルド方法を決定します。簡素化された QuickPatch には、簡素化されていない QuickPatch パッケージに比べ、より少ない数の新しいサブ機能とカスタム アクションが含まれています。

場合によって、QuickPatch パッケージの簡素化ができない場合があります。たとえば、インストール済みのファイルを削除するように QuickPatch パッケージを構成した場合、簡素化は行えません。

新しい QuickPatch プロジェクトを作成したとき、"QuickPatch の簡素化" 設定のデフォルト値は [はい] に設定されています。ただし、QuickPatch プロジェクトを InstallShield 2008 Express Edition 以前から InstallShield 2021 Express Edition へアップグレードした場合、この設定の値は [いいえ] になります。この値は、必要に応じて変更が可能です。詳細については、「QuickPatch パッケージを簡素化するかどうかを指定する」を参照してください。

SetARPINSTALLLOCATION カスタム アクションのデフォルト条件

デフォルトで、すべての新しい Express プロジェクトには、ビルトイン InstallShield カスタム アクション SetARPINSTALLLOCATION が含まれています。このカスタム アクション (ARPINSTALLLOCATION プロパティの値を製品のプライマリ フォルダーの完全修飾パスに設定します) は、[インストール]-[実行] シーケンスにシーケン

スされ、条件を持ちません。InstallShield 2008 Express Edition 以前では、このカスタム アクションのデフォルト条件は Not Installed でした。デフォルトの [Not Installed] 条件を使用すると、カスタム アクションはメンテナンス モード中に実行されず、その結果 ARPINSTALLLOCATION プロパティの値が空白になります。

InstallShield 12 Express Edition 以前からのプロジェクトをアップグレードする

以下は、InstallShield 12 Express Edition 以前から InstallShield 2021 Express Edition へアップグレードされたプロジェクトに影響が出る可能性がある変更についての説明です。

InstallShield Express Edition の以前のバージョンで作成されたプロジェクトをアップグレードする

InstallShield 2021 Express Edition を使って以前のバージョンで作成されたプロジェクトを開くと、プロジェクトを新しいバージョンに変換するかどうかを質問するメッセージ ボックスが表示されます。[変換する] を選択すると、変換が行われる前に、例えば .765 というファイル拡張子が付加されたプロジェクトのバックアップ コピーが作成されます。以前のバージョンの InstallShield でこのプロジェクトを再度開く場合、元のプロジェクトのファイル名から .765 を取り除いてください。InstallShield 2021 Express Edition プロジェクトを以前のバージョンの InstallShield Express Edition で開くことはできませんので注意してください。

次の InstallShield Express Edition バージョンで作成した既存プロジェクトを InstallShield 2021 Express Edition に移行することができます: InstallShield 12 Express Edition 以前と InstallShield Express 5 以前。InstallShield MultiPlatform または InstallShield Universal で作成されたプロジェクトは InstallShield 2021 Express Edition に移行することはできませんの注意してください。

ターゲット システムにおける Windows 9x、Windows NT 4、および Windows Me のサポートの終了

InstallShield では、Windows 9x、Windows NT4、および Windows Me システムのためのインストールを作成することができなくなりました。エンドユーザーのコンピューターで、これらのオペレーティング システムのいずれかが使用されているときに、InstallShield 2021 Express Edition でビルドされたインストールを実行しようとすると、予期しない結果が発生することがあります。

COM 抽出

デフォルトで、ビルド時に未使用のディレクトリを .msi ファイルから自動的に削除する

InstallShield 12 Express Edition 以前のバージョンを使って作成した Express プロジェクトを InstallShield 2021 Express Edition にアップグレードすると、[リリース] ビューの [ビルド] タブに新しく追加された ″未使用のディレクトリを保持する″ 設定は、デフォルトで [いいえ] に設定されています。したがって、Directory テーブルの Directory 列に一覧表示されているディレクトリが .msi ファイル内の既知の場所で参照されない場合、ビルド時に InstallShield が作成する .msi ファイルの Directory テーブルからそのディレクトリは削除されます。これはマージ モジュールがマージされてから削除されますが、.msi ファイルに存在するディレクトリのみが削除の対象となります。したがって、マージ モジュールの Directory テーブルに新しい未使用のディレクトリが含まれている場合、そのディレクトリはインストールに追加されます。

あるケースにおいては、"未使用のディレクトリを保持する" 設定の値を [はい] に変更したほうが良い場合があります。たとえば、プロジェクトに、プロジェクトの他の領域では参照されていないディレクトリを使用しているカスタム アクションが含まれている場合、"未使用のディレクトリを保持する" 設定の値を [はい] に設定することもできます。

ダウングレードの自動防止

新しい Express プロジェクトを作成したとき、[アップグレード パス] ビューに ISPreventDowngrade という名前のアップグレード アイテムが含まれています。ISPreventDowngrade は、現在のインストールが将来のメジャー バージョンの製品を上書きするのを防ぎます。プロジェクトを InstallShield 12 Express Edition 以前から InstallShield 2021 Express Edition に移行したとき、ISPreventDowngrade アイテムは自動的に追加されません。

ダウングレード防止のサポートを手動で追加する方法については、「現在のインストールによる同製品の将来のメジャー バージョンの上書きを防ぐ」を参照してください。

ダイアログ ボタンの UAC シールド アイコンのサポート

インストールが Windows Vista システム上で実行されているとき、インストールが昇格された権限でまだ実行されていない場合、[インストールの準備完了] ダイアログの [インストール] ボタンと [アンインストールの準備完了] ダイアログの [削除] ボタンに [ユーザー アカウント制御 (UAC)] シールド アイコンが表示されます。これは、すべての新しいプロジェクトおよび InstallShield 12 Express Edition 以前から移行されたプロジェクトに適用します。

InstallShield は昇格された権限で実行されます。このため、Windows Vista システムでインストールを InstallShield 内から起動したとき、インストールは昇格された権限で実行され、[ユーザー アカウント制御 (UAC)] シールド アイコンは [インストールの準備完了] と [アンインストールの準備完了] ダイアログで表示されません。

ALLUSERS と [ユーザー情報] ダイアログの変更

InstallShield 2008 Express Edition より、すべての新規 Express プロジェクトでは、デフォルトで ALLUSERS プロパティが 1 に設定されています。ほとんどのインストールは、マシンごとに管理者権限を使用して実行される必要があるため、これが推奨される実装です。

InstallShield 12 Express Edition 以前で作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、ALLUSERS プロパティの値は自動的に変更されません。また、このプロパティが以前のプロジェクトで定義されていない場合も、自動的に追加されません。

新規または移行したプロジェクトで ALLUSERS の値を変更するには、[一般情報] ビューを利用します。詳細については、「一般情報の設定」を参照してください。

また、InstallShield 2008 Express Edition から、デフォルトで、すべての Express プロジェクトの [ユーザー情報] ダイアログは、エンドユーザーが製品をすべてのユーザーにインストールするか、または現在のユーザーのみにインストールするかを指定できるラジオ ボタンを表示しないようになっています。このダイアログについては、これが推奨される実装です。

InstallShield 12 Express Edition 以前で作成されたプロジェクトを InstallShield 2021 Express Edition にアップグレードしたとき、[ユーザー情報] ダイアログは自動的に変更されません。

詳細については、「ユーザーごとのインストールとマシンごとのインストールの違い」を参照してください。

リリースの "キャッシュ パス"設定における新しいデフォルト値

今回より、[リリース] ビューの "キャッシュ パス" 設定における圧縮リリースのためのデフォルト値が *[LocalAppDataFolder]Downloaded Installations* に設定されています。以前のデフォルト値 *[WindowsFolder]Downloaded Installations* は、ロックされたシステムで使用できないことがあります。プロジェクトを InstallShield 12 Express Edition 以前から InstallShield 2021 Express Edition に移行したとき、"キャッシュ パス" 設定は自動的に変更されません。必要な場合、この値を手動で変更します。

QuickPatch プロジェクトにおけるパッチの作成

今回のリリースより、InstallShield は QuickPatch パッケージの作成に Patchwiz.dll のバージョン 3.1 を使用します。

DemoShield のサポート

DemoShield は現在販売されていません。サポートも終了しました。このため、今後 InstallShield に DemoShield は統合されません。

InstallShield Express 2.x からプロジェクトをアップグレードする

現在のバージョンの InstallShield と InstallShield Express 3.0 以前のバージョンとでは技術的に多くの違いがあります。InstallShield Express 2.x は、従来のセットアップ アーキテクチャに基づき、Setup.exe という実行可能ファイルが作成されます。この実行可能ファイルには、インストールを実行するために必要なすべての論理が含まれます。現在のバージョンの InstallShield では、出力ファイル は Windows Installer サービスとインタラクトするために必要なすべての情報を含んだリレーショナル データベース (.msi) です。これらの大きな違いにより、以前の InstallShield プロジェクトが現在のアーキテクチャに適合しない場合があります。また、移行の結果として作成された新しいプロジェクトにも相違が生じます。セットアップをビルドして配布する前に、これらの失われた部分を満たす必要があります。

プロジェクトの種類

InstallShieldの以前のバージョンでは、新しいセットアップ プロジェクトを作成するとき、プロジェクトの種類を選択するようにプロンプトが表示されていました。現在のバージョンの InstallShield には、プロジェクトの種類に相当するものはありません。作成するプロジェクトの種類は、セットアップに追加するファイルとどのようなシステム変更を加えるかによって異なります。あるプロジェクト タイプを選択して集められたすべての情報は、新しい InstallShield プロジェクトに移行されます。

ファイル、ファイルグループ、およびコンポーネント

セットアップに追加したすべてのファイルは新しい InstallShield プロジェクトに転送されます。旧バージョンの InstallShieldでは、ファイルはファイル グループに追加され、ファイル グループはコンポーネントに追加されました。現在のバージョンの InstallShield では、ファイルは機能に追加されます。上位レベルは存在しません。したがって、セットアッププロジェクトを移行する場合、ファイルはファイルグループに関連付けられません。その代わりに、すべてのファイルは機能に属します。旧プロジェクトの各コンポーネントに対して、1 つの機能が新しいプロジェクト内に作成されます。ファイル グループは、インストール先フォルダーにマップします。詳細については、「インストール先フォルダー」を参照してください。

セットアップの種類

[セットアップの種類] では、旧バージョンと同様の機能を現在バージョンの InstallShield でも提供しています。2 つの主な違いは、セットアップの種類が以前はコンポーネントに基づいていた点です。最新バージョンの InstallShield では、コンポーネントは存在しません。コンポーネントは機能に置換されました。以前の InstallShield Express プロジェクトで指定したセットアップの種類は、新しい InstallShield 内で再び作成されます。詳細については、「セットアップの種類について」を参照してください。

InstallShieldオブジェクト

InstallShield オブジェクトはマージ モジュールに置換されました。プロジェクトで使用したオブジェクトによって提供される機能を提供するマージ モジュールが存在する場合、そのモジュールはセットアップに追加されます。セットアップにマージ モジュールにも含まれているファイルが含まれている場合、モジュールは単一のファイルではなくセットアップに追加されます。

詳細については、「インストールに再配布可能ファイルを含める」を参照してください。

InstallShield エクステンション

InstallShield エクステンションはカスタム アクションに置換されました。カスタム アクションを使用すると、DLL の関数を呼び出したり、実行可能ファイルを起動することができます。これら 2 つのタスクの 1 つを実行するエクステンションがあった場合、そのエクステンションは新しいプロジェクト内でカスタム アクションに変換されます。詳細については、「カスタム アクションを使用する」を参照してください。

レジストリ エントリ

以前のプロジェクトのすべてのレジストリ データは、新しいプロジェクトに追加されます。

ショートカット

新しい InstallShield のショートカットは、以前のバージョンに比べ大幅に堅牢になりました。以前のセットアップにあったショートカットは、新しいセットアッププロジェクトに移行されます。ただし、最新バージョンの InstallShield には多くのショートカットの設定があるため、ホットキーの割り当て、アイコン インデックス、作業ディレクトリなど、プロジェクト内の各ショートカットの設定を、[ショートカット/フォルダー] ビューに移動して構成することも一案です。ただし、これらのプロパティは必須ではありません。

各設定の詳細については、「ショートカットの設定」を参照してください。

ダイアログ

最新バージョンの InstallShield の実行時ダイアログは旧バージョンのものとは外観が異なりますが、同様の機能を持ちます。たとえば、[セットアップの種類] ダイアログは、これまでと同じ目的で使用されます。このダイアログでは顧客が使用するセットアップの種類を選択できます。以前の InstallShield Express プロジェクトで選択した同じダイアログが、新しいプロジェクトに含まれます。

以前のバージョンの InstallShield Express と最新バージョンのダイアログの主な違いは、.rtf ファイルの使用にあります。使用許諾契約または readme ファイルにリンクする場合、以前のバージョンの InstallShield Express ではテキスト (.txt) ファイルを使用していました。最新バージョンの InstallShield では、リッチ テキスト (.rtf) ファイルを使用する必要があります。以前に作成された .txt ファイルは、移行プロセスの一部として .rtf ファイルに変換されます。

配布メディア

以前のバージョンの InstallShield Express で選択したメディアの種類は、新しい InstallShield セットアップには移行されません。[リリースのビルド] ビューの利用可能なメディアのタイプのリストからメディアを選択します。利用可能なそれぞれのタイプについての詳細は、「メディアの種類」を参照してください。

以前にセットアップをフロッピーディスクで配布してある場合は、「フロッピー ディスクで配布するリリースをビルドする」を参照してください。

16 ビット セットアップ

Windows Installer サービスは 32 ビットオペレーティング システム上でのみ実行可能なため、16 ビットセットアップは 32 ビットセットアップと同様には移行されません。16 ビットセットアップで作成したファイル、ショートカット、およびその他の設定はすべて移行されますが、セットアップに含めている自己登録 16-bit ファイルは自己登録しません。実際、セットアップが 16 ビットシステム上で実行できないため、16 ビットファイルをセットアップに含める理由がありません。

Languages

InstallShield には多数の言語のビルトイン サポートがあります。以前のバージョンの InstallShield Express のプロジェクト言語が サポート言語 の 1 つと一致する場合、新しい InstallShield のセットアップはその言語のランタイム文字列と一緒に作成されます。たとえば、フランス語にローカライズされたセットアップを最新バージョンの InstallShield に移行しようとした場合、新しい InstallShield プロジェクトのすべての設定はフランス語になります。以前のバージョンの InstallShield Express プロジェクトがサポートされていない言語で実行されていると、新しいプロジェクトのランタイム文字列は英語で表示されます。ただし、だからといってセットアップを希望の言語で実行できないわけではありません。すべてのランタイム文字列は [テキストとメッセージ] ビューで翻訳できます。

InstallShield Premier または InstallShield へのアップグレード

インストールに InstallShield Express Editionで提供されている以上の機能が必要になった場合、既存のプロジェクトを使って、InstallShield Premier または InstallShield でインストールを作成することができます。InstallShield Premier および InstallShield には、より複雑なインストールで必要なカスタマイズ機能が多数用意されています。

Premier Edition のみで提供されている機能

以下は、Premier Edition で提供されていて、InstallShield または Express Edition では提供されていない機能の一部です:

- **スイート/アドバンスト UI インストールの作成およびビルド機能**—複数の .msi パッケージ、.msp パッケージ、InstallScript パッケージ、.exe パッケージ、および Windows Installern トランザクションだけでなく、複数の InstallShield 前提条件にも対して、最新のカスタマイズ可能なユーザー インターフェイスを持つブートストラップ アプリケーションを作成することができます。スイート/アドバンスト UI インストールでは、複数の個別のインストールを、統合されたユーザー インターフェイスを使用する単一のインストールにパッケージ化し、条件付きで、必要に応じて、ターゲット システムでパッケージを起動するセットアップ ランチャー (Setup.exe) を使用します。
- **スイート/アドバンスト UI インストールで InstallScript アクション、およびその他の種類のアクションをサポート** – スイート/アドバンスト UI インストールでは、今回より、インストールに含めるパッケージの範囲

を超えた様々な実行時タスク行うための InstallScript アクションを起動できるビルトイン サポートが提供されています。アクションを使って、実行可能ファイルを実行、DLL 関数を呼び出し、PowerShell スクリプトを実行、スイート/アドバンスト UI プロパティを設定、InstallScript コードを実行、またはマネージ アセンブリでパブリック メソッドを呼び出すことができます。

- **仮想化サポート** – Microsoft App-V アシスタントは、仮想化パックに含まれています。このアシスタントを使って、Microsoft App-V 形式で、カスタム仮想アプリケーションを作成することができます。仮想化技術を使って、アプリケーションを独自の環境に隔離することで、既存アプリケーションとの競合を回避したり、基盤となるオペレーティング システムの変更を避けたりすることが可能です。

- **アプリケーション仮想化適合性スイート**–InstallShield では、製品を仮想化するための準備が整っているかどうかを判断するのに役立つ、新しい検証スイートが提供されています。これらのスイートに含まれている InstallShield 仮想化内部整合性検証ツール (ISVICE) を使って、Microsoft App-V 4.x、Microsoft App-V 5、Microsoft Server App-V、VMware ThinApp、および Citrix XenApp との適合性をチェックできます。顧客に仮想バージョンの提供を考慮する場合に、検証スイートを使用して、製品をどのようにビルドするか豊富な情報に基づいた意思決定を行うことができます。

- **DIM ファイルのサポート** – DIM プロジェクトの作成機能はInstallShield Premier Edition で使用できます。このサポートは、コラボレーション アドオン、InstallShield Developer Installation Manifest Editor でも提供されています。DIM ファイルを基本の MSI プロジェクトに追加する機能は、InstallShield Premier Edition で提供されています。

  DIM プロジェクトは、機能サイズのプロジェクトで、インストール パッケージの別個に分かれている部分を構成する製品ファイル、ショートカット、レジストリ エントリ、テキスト ファイルの変更、IIS Web サイト、および要素など、関連するアイテムを集めたものです。DIM を利用することにより、複数のチーム メンバーが、インストールの開発を同時に携わることができます。各ソフトウェア開発者またはチームメンバーは、異なる DIM について個別で作業することができ、リリース エンジニアは、1 つまたは複数の基本の MSI プロジェクトでそれらを参照することができます。

- **複数言語インストール** – 複数の言語でエンドユーザー テキストを表示する単一のインストール プロジェクトを作成し、言語固有のファイルの条件付きインストールを処理できます。翻訳済みの文字列を利用して、ダイアログおよびメッセージを 34 の追加言語の 1 つに変更することができます。

  プロジェクト: *アラビア語 (サウジアラビア) とヘブライ語の 2 つの言語サポートは、基本の MSI とマージ モジュール プロジェクトでのみ利用できます。*

- **アラビア語 (サウジアラビア) とヘブライ語サポート** – InstallShield Premier Edition は、右から左方向へ読み書きするアラビア語 (サウジアラビア) とヘブライ語のサポートを含みます。デフォルトのエンド ユーザー ダイアログ文字列のすべてが、これらの言語で利用できます。

  これらの言語は右から左方向に読まれるため、Premier Edition では、アラビア語とヘブライ語ダイアログのミラーリングが サポートされています。このサポートにより、アラブ語とヘブライ語のダイアログでは、右から左方向へのレイアウトが使用されます。たとえば、英語やその他の左から右に読まれる言語のダイアログで右側にあるボタンは、右から左に読まれる言語のダイアログでは左側に移動されます。

- **ビルド前、ビルド中、およびビルド後に実行するコマンドを指定する機能**– InstallShield Premier Edition には、ビルド プロセスの様々な段階で実行するコマンドを指定できるリリース設定が含まれています。次のビルド イベントで実行するコマンドをスケジュールすることができます: (a) InstallShield がリリースのビルドを開始する前、(b) InstallShield が .msi パッケージおよび (製品のデータ ファイルが .cab ファイルに格納される場合) .cab ファイルをビルドした後、.msi パッケージに署名が行われて、Setup.exe ファイルにストリームされる前、(c) InstallShield がリリースをビルドならびに署名を行った後。

- **ビルド時またはオンデマンドで InstallShield が初期化する仮想マシンにインストールを配布できる機能** – インストールのビルドが成功するたびに InstallShield が仮想マシン (VM) を指定のスナップショットに戻し、VM の電源をオンにしてインストールを VM にコピーし、テストが可能な状態になるようプロジェクトを構成できます。また、これらのテスト準備を必要なときにオンデマンドで行うこともできます。このテスト準備機能により、手作業で行う処理を減らし、テストにかかる時間を短縮できます。VM は Microsoft Hyper-V Server、VMware ESX または ESXi Server、または VMware Workstation を指定できます。
- **InstallShield MSI ツール用の追加のライセンス** – InstallShield には、InstallShield MSI Diff、InstallShield MSI Query、InstallShield MSI Sleuth、および InstallShield MSI Grep 用のライセンスが追加で含まれています。これらのツールを利用して、Windows Installer パッケージに関連する問題を解決することができます。InstallShield Premier Edition には、別のマシンに InstallShield 以外のツールのみをインストールできる、個別のインストールおよび追加ライセンスが含まれています。詳しい使用条件については、InstallShield MSI ツールの使用許諾契約書を参照してください。
- **既存の IIS Web サイトからプロジェクトへの IIS データのインポート機能** – InstallShield では、IIS スキャナー (IISscan.exe) が提供されています。このコマンドライン ツールを使って、既存の IIS Web サイトをスキャンして、Web サイトに関する IIS データを記録できます。IIS スキャナーは、Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールすべての設定を含む XML ファイルを作成します。この XML ファイルを使って、InstallShield Premier Edition の [IIS 構成] ビューに IIS データをインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。
- **Web 配置パッケージの配置をサポート** – InstallShield Premier Edition で作成可能なスイート/アドバンスト UI インストールでは、Web 配置パッケージを IIS Web サーバーまたはクラウドに配置できるビルトイン サポートを利用できます。
- **InstallShield Collaboration** – InstallShield Premier Edition には、InstallShield Collaboration for Visual Studio のライセンスが含まれています。
- **ネットワーク リポジトリ** – ネットワーク リポジトリは、複数のインストール作成者が必要に応じアクセスでき、プロジェクトで再利用することができるインストール要素のコレクションです。ネットワーク リポジトリは、インストール作成者同士のコラボレーション作業を促進し、ネットワークに格納されます。
- **InstallShield ベスト プラクティス スイート**– InstallShield には、InstallShield ベスト プラクティス スイートという名前の 1 セットの検証ツールが含まれています。インストールがベスト プラクティス ガイドラインに違反している場合、このスイートの InstallShield ベスト プラクティス (ISBP) 検証ツールによって警告されます。
- **追加のダイアログ テーマ** – 一部のダイアログ テーマは InstallShield Premier Edition のみで提供されています。

InstallShield Premier および InstallShield でのみで提供されている機能

以下は、InstallShield Premier Edition および InstallShield で提供されていて、Express Edition では提供されていない機能の一部です:

- **アドバンスト UI インストールの作成およびビルド機能**–単一の .msi パッケージ、.msp パッケージ、および InstallScript パッケージだけでなく InstallShield 前提条件にも対して、最新のカスタマイズ可能なユーザー インターフェイスを持つブートスラップ アプリケーションを作成することができます。アドバンスト UI では、条件付きで、必要に応じて、ターゲット システムでパッケージを起動するセットアップ ランチャー (Setup.exe) を使用します。

(InstallShield Premier Edition の スイート/アドバンスト UI 機能には、この機能に対して、さらに拡張サポートが追加されています。スイート/アドバンスト UI 機能を利用すると、複数の .msi パッケージ、.msp パッケージ、InstallScript パッケージ、.exe パッケージ、サイドロード アプリケーション パッケージ (.appx)、および Windows Installern トランザクションだけでなく、複数の InstallShield 前提条件も、モダンでカスタマイズ可能なユーザー インターフェイスを持つ単一のアプリケーションにパッケージできます。)

- **Standalone Build**— このツールは InstallShield Premier および InstallShield で使用できます。Standalone Build を使って、ビルド マシン上に InstallShield のインストール ビルド機能のみ、および希望する任意の再配布可能ファイルをインストールします。Standalone Build 用の追加ライセンスを購入することもできます。
- **ダイアログ エディター** — ダイアログ エディターを利用して、既存のエンド ユーザー ダイアログのレイアウトを変更したり、新規カスタム ダイアログを作成することができます。プロジェクト全体でダイアログをインポートおよびエクスポートして、共有することができます。プロジェクトでサポートしている各言語用に、別のダイアログを作成します。
- **InstallShield MSI ツール** — InstallShield Premier および InstallShield には、InstallShield MSI Diff、InstallShield MSI Query、InstallShield MSI Sleuth、および InstallShield MSI Grep というツールが含まれています。これらのツールを利用して、Windows Installer パッケージに関連する問題を解決することができます。
- **オートメーション インターフェイス** — スクリプトを指定して、新しいファイルの追加、機能の追加または削除、製品名とアップグレード コードの変更、リリースの設定の変更、概要情報ストリーム項目の変更、リリース フラグの変更、プロパティの変更、ビルド処理の開始などを行うことができます。
- **リリースのカスタマイズ** — プロジェクトのどの部分を圧縮し、どの機能をどのディスクに配置して、どの言語を含めるか定義します。ローカライズ作業をサポートするために、言語に基づいてアプリケーションデータをフィルタリングしてください。
- **ソース コード管理の統合** — プロジェクトをソース コード管理システムにチェックインおよびチェックアウトするプロセスを単純化し、プロジェクトの差分を検出するときに容量を節約します。InstallShield Premier および InstallShield で提供されている SCC の統合は、様々なソース コード管理システムとの統合をサポートします。
- **柔軟なローカリゼーション サポート** — InstallShield Premier および InstallShield には [文字列エディター] ビューが搭載されていて、インストール中、実行時に表示される翻訳可能なテキスト文字列すべてを一箇所にまとめて管理できます。このビューを使って、ボタン テキストから機能の説明まで全ての文字列を編集できます。このビューを使って、文字列エントリを翻訳可能な形式にエクスポートしてから、翻訳済み文字列をプロジェクトにインポートすることもできます。
- **プロジェクト検証** — 標準 .cub ファイルを使って、インストールとマージ モジュールを検証します。アップグレードとパッチの検証を利用して、潜在的に存在するアップグレードに関する問題を検出し、リリースされる前にそれらを解決します。
- **パッチの作成** — QuickPatch プロジェクトの作成のほか、InstallShield Premier および InstallShield では、製品の以前のバージョンに対するアップデートを含む標準パッチを作成することができます。
- **複数製品バージョンの管理** — 1 つのプロジェクトから、評価版、デバッグ版、標準版、拡張版などのバージョンをビルドします。ユーザー定義フラグを使って、リリースに特定の機能、InstallShield 前提条件、およびその他の要素を選択 (または除外) することができます。
- **InstallScript のサポート** — InstallShield Premier および InstallShield では、シンプルかつパワフルなプログラム言語 InstallScript がサポートされています。InstallScript カスタム アクションを Windows Installer ベースのインストールに追加したり、Windows Installer エンジンの代わりに InstallScript エンジンを使用してインストール全体を制御する InstallScript プロジェクトを作成したりできます。

- **柔軟なカスタム アクション サポート** – InstallShield Premier および InstallShield は、Express Edition では提供されていない種類のいくつかのカスタム アクションをサポートします。これらの追加カスタム アクションを利用して、プロパティの設定、ディレクトリの設定、マネージ アセンブリ内でのパブリック メソッドの呼び出し、または特定の条件下でエラーメッセージを表示してインストールを中止することが可能です。
- **柔軟なショートカット サポート**– InstallShield Premier および InstallShield には、Express Edition では構成することができない、ショートカットの様々な詳細設定を構成するためのサポートが含まれています。たとえば、InstallShield Premier および InstallShield では、プロジェクト内のショートカットをタスクバーまたは [スタート] メニューにピン留めしないように防ぐことができます。また、エンド ユーザーが製品をインストールした後に、[スタート] メニューにあるショートカットを、新しくインストールされた製品として強調表示しないように防ぐことができます。これらのショートカット オプションは、しばしばインストールの一部となるツールまたは付属的な製品に使用されます。
- **マージ モジュールの作成と編集** – 他のアプリケーションのインストール パッケージで再利用できるように、プロジェクトの小さい単位をパッケージすることができます。独自に作成したもの、または、製品に含まれるすべてのモジュールを再利用できます。さらに詳細なカスタマイズを行うために、モジュールを開いて編集することができます。
- **プロジェクトのテンプレート** – インストール プロジェクトやマージ モジュールプロジェクトの作成するとき、その出発点として利用できるすべてのデフォルト設定とデザイン要素を含むプロジェクト テンプレートを作成できます。
- **複数の IIS Web サイト** – InstallShield Express および Limited Edition では、1 つのインストールによって 1 つの Web サイトのみインストールできます。InstallShield Premier および InstallShield では、1 つのインストールによって複数の Web サイトをインストールできます。
- **IIS アプリケーション プールと Web サービス拡張のサポート** – IIS アプリケーション プールと Web サービス拡張を管理します。
- **Windows サービスの詳細サポート**–InstallShield Premier および InstallShield では、サービスに関するいくつかの作業がサポートされていますが、InstallShield Premier および InstallShield では、サービスに関する追加の柔軟性が提供されています。たとえば、InstallShield Premier および InstallShield ではインストールおよびアンインストール中にサービスを開始、停止、または削除することができます。サービスはインストールの一部である場合と、ターゲット システムに既存する場合があります。これらのエディションではまた、Windows Installer 5 で利用可能な拡張サービス カスタマイズ オプションを構成することもできます。
- **SQL サポート** – SQL サーバーへの接続、データベース スキーマおよびデータのインポート、SQL スクリプトと機能の関連付けなど多数サポートされています。
- **テキスト ファイルまたは XML ファイルを変更できる機能** – [テキスト ファイルの変更] ビューまたは [XML ファイルの変更] ビューを使って、実行時にターゲット システム上で変更するファイルを構成できます。
- **64 ビット専用サポート** – Windows Server Core は、WOW64 (32-bit Windows-on-Windows) サポートの無効化をサポートします。この構成がより一般化するのに伴い、64 ビットのアプリケーションが 32 ビットの機能に全く依存しないでインストールできるようにすることが必要となります。これを実現するため、InstallShield Premier および InstallShield では 64 ビット専用 .msi パッケージのビルドが可能です。これらは、WOW 64 機能を搭載していない 64 ビット版の Windows ベースのシステム上で実行します。
- **InstallShield 前提条件エディター** – このツールを使用して、新しい InstallShield 前提条件を作成したり、既存の前提条件を変更したりできます。
- **Setup.exe および Update.exe のカスタム アイコン–ビルド時に作成する** – Setup.exe および Update.exe ファイルに使用するカスタム アイコン (.exe、.dll、または .ico ファイル) を指定します。アイコンは Setup.exe の [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザー

が Setup.exe ファイルを右クリックして、[プロパティ] をクリックしたときに表示されます。エンド ユーザーが Windows Explorer で Setup.exe ファイルを参照したときにも、このアイコンが表示されます。

- **Setup.exe の有効期限日** – Setup.exe の有効期限日と有効期限メッセージを設定します。エンド ユーザーが Setup.exe をプロジェクトで指定された日付以降に実行すると、有効期限切れメッセージが表示されて、インストールが終了します。
- **トランザクション処理を使った複数パッケージのインストールをサポート** – Windows Installer 4.5 以降では、トランザクション処理を使った複数パッケージのインストールをサポートします。InstallShield Premier および InstallShield では、インストール プロジェクトに連鎖 .msi パッケージを追加できます。パッケージ並びに追加 .msi パッケージは連鎖されて、単一トランザクションとして処理されます。トランザクションに含まれる 1 つまたは複数のパッケージが正しくインストールされなかった場合、またはエンド ユーザーがインストールをキャンセルした場合、Windows Installer は全てのパッケージについてロールバックを開始して、システムを以前の状態に復元します。
- **様々なプロジェクト要素をエクスポートおよび再利用できる機能** – 既存のプロジェクトの一部 (ダイアログ、カスタム アクション、または機能) をマージ モジュールまたは別のインストール プロジェクトに移動して、効率を高めます。
- **複数インスタンスのサポート** – エンド ユーザーが製品の複数のインスタンスを同じマシン上に同じユーザー コンテキストでインストールできるインストールを作成します。
- **デバイス ドライバー サポート** – InstallShield Premier および InstallShield に搭載されているデバイス ドライバー サポートは、Microsoft の DIFxApp (Driver Installation Frameworks for Applications) を使って、インストールからデバイス ドライバーをインストールする処理を簡素化します。
- **追加のダイアログ テーマ** – 一部のダイアログ テーマは InstallShield Premier および InstallShield のみで提供されています。Limited および Express Edition で提供されてるテーマは 2 種類だけです。
- **Visual Studio マージ モジュール プロジェクトの変換**–InstallShield Premier および InstallShield を使って、Visual Studio マージ モジュール プロジェクトを InstallShield マージ モジュール プロジェクトに変換できます。その他のプロジェクトで使用するマージ モジュールをビルドするとき、この変換処理が必要です。
- **COM+ アプリケーション プロキシ サポート** – インストール中に COM+ アプリケーション プロキシを管理します。COM+ アプリケーション プロキシはサーバー アプリケーション属性のサブセットで構成され、これはクライアント コンピューターからアプリケーションが存在するマシンへのリモート アクセスを可能にします。

各エディションで提供されている機能についての詳細情報は、InstallShield セールスに問い合わせるか、http://www.installshield.com にアクセスしてください。

Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートする

Visual Studio は、制限付きでセットアップおよびマージ モジュール プロジェクトの作成をサポートします。InstallShield を使って、高度な機能を使ったインストールを作成できるように、次の処理を行うことができます:

- Visual Studio セットアップまたはマージ モジュール プロジェクト (.vdproj) を Express プロジェクト (.ise) にインポート。インポート処理中に、Visual Studio プロジェクト内の特定の設定について、それらをインポートするか無視するかを選択できます。
- Visual Studio セットアップ プロジェクトを Express プロジェクトに変換。

インポートおよび変換機能を使って、Visual Studio プロジェクトに含まれている同じデータと設定を含む InstallShield インストール プロジェクトを作成できます。

メモ: *Visual Studio プロジェクトに 1 つ以上のプロジェクト出力が含まれている場合、InstallShield を使ってその Visual Studio プロジェクトを InstallShield プロジェクトにインポートできますが、InstallShield がその Visual Studio プロジェクトを InstallShield プロジェクトに変換することはできません。*

エディション: *InstallShield Premier および InstallShield では、その他にも Visual Studio マージ モジュール プロジェクトを InstallShield マージ モジュール プロジェクトに変換することができます。この処理は、その他のプロジェクトで使用するマージ モジュールをビルドするときに必要です。*

サポート対象の Visual Studio バージョン:

- Visual Studio 2010
- Visual Studio 2008
- Visual Studio 2005
- Visual Studio .NET 2003
- Visual Studio .NET

Visual Studio プロジェクトの代わりに InstallShield プロジェクトを使用することの利点

Visual Studio セットアップ プロジェクトを InstallShield Express プロジェクトに変換すると、InstallShield の機能を使ってプロジェクトをカスタマイズすることができます。

以下は、InstallShield プロジェクトで実行可能で Visual Studio プロジェクトでは実行不可能ないくつかのタスクの一覧です:

- 機能の管理。
- IIS Web サイト、アプリケーション、および仮想ディレクトリの管理。
- .msi データベースの Binary テーブルに格納されているか、製品と共にインストールされるソース ファイルを使用する DLL、実行可能ファイル、および VBScript 、および JScript カスタム アクションの作成。実行可能ファイル カスタム アクションは、ターゲット システム上に既に存在するソース ファイルもサポートします。(Visual Studio では、すべてのカスタム アクションは製品と共にインストールされる必要があります。)
- ファイル転送中に表示するビルボードの追加。
- ターゲット システムに既に存在するファイルのショートカットを作成する。
- COM+ アプリケーションを管理する。

インポート処理

InstallShield を使って、Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield Express プロジェクトにインポートできます。

メモ: *InstallShield プロジェクトにインポートする Visual Studio セットアップ またはマージ モジュール プロジェクトに 1 つ以上のプロジェクト出力が含まれている場合、その InstallShield は、Visual Studio セットアップまたはマージ モジュール プロジェクトおよびそのプロジェクトのすべての依存関係を含む、同じ Visual Studio ソリューションに含まれていなくてはなりません。*

プロジェクト出力を含む Visual Studio プロジェクトをインポートするには、Visual Studio 内部から InstallShield を使用しなくてはなりません。InstallShield プロジェクトが InstallShield で開かれていて、Visual Studio 内部から開かれていない場合に、プロジェクト出力を含む Visual Studio プロジェクトを InstallShield プロジェクトにインポートしようとすると、エラーが発生します。

タスク ***Visual Studio プロジェクト (.vdproj) を Express プロジェクト (.ise) にインポートするには、以下の手順に従います:***

1. Express プロジェクトを開くか、作成する。
2. **[プロジェクト]** メニューで、**[Visual Studio デプロイメント プロジェクト ウィザード]** ボタンをクリックします。
3. Visual Studio デプロイメント プロジェクト ウィザードのパネルを完成させる。

InstallShield は、ウィザードで構成された設定に基づいて Visual Studio プロジェクトを現在開かれている Express プロジェクトにインポートします。InstallShield がプロジェクトをインポートするとき、[出力] ウィンドウにプロジェクトのインポート ステータスが表示されます。出力ウィンドウでは、変換プロセスの各ステップが表示され、すべての変換エラーと警告が一覧で表示されます。InstallShield はすべてのファイル、レジストリ エントリ、およびその他のアプリケーション データを、Visual Studio プロジェクトから InstallShield プロジェクトの [常にインストール] 機能にインポートします。

変換プロセス

InstallShield を使って、Visual Studio セットアップ プロジェクトを変換すると、InstallShield では InstallShield Express プロジェクト (.ise) が作成されます。

タスク ***Visual Studio プロジェクト (.vdproj) を InstallShield プロジェクト (.ise) に変換するには、以下の手順に従います:***

1. InstallShieldを開きます。
2. **[ファイル]** メニューで、**[開く]** をクリックします。**[開く]** ダイアログ ボックスが開きます。
3. **[ファイルの種類]** ボックスで、**Visual Basic セットアップ プロジェクト(*.vdproj)** を選択します。
4. 開く Visual Studio プロジェクトの場所を参照し、プロジェクト ファイルを選択します。
5. **[開く]** ボタンをクリックします。

InstallShield は、Visual Studio プロジェクトの設定に基づいて InstallShield プロジェクトを作成します。InstallShield は、.ism ファイルを .vdproj ファイルと同じフォルダーに格納します。.ise ファイルが作成されると、[出力] ウィンドウにプロジェクト変換のステータスが表示されます。出力ウィンドウでは、変換プロセスの各ステップが表示され、すべての変換エラーと警告が一覧で表示されます。

変換処理が成功すると、その新しい InstallShield プロジェクトが InstallShield で表示されます。

インポート後および変換後のタスク

前提条件タスク

Visual Studio では、1 つ以上の定義済み前提条件を Visual Studio セットアップ プロジェクト内の 1 つ以上の構成に追加することができます。InstallShield のインポート処理は、すべての構成に含まれるすべての前提条件を、対応する InstallShield 前提条件に変換しようとします。InstallShield に対応する InstallShield 前提条件が含まれていない場合、警告 -9071 が発生して、その前提条件が変換されなかったことを通知します。

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれています。InstallShield 前提条件エディターを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。これらの InstallShield のエディションの 1 つをご利用の場合、製品のインストール中に起動される独自の InstallShield 前提条件を作成することができます。*

また、InstallShield Premier および InstallShield では、単一のプロジェクトからインストールの異なるバージョン (評価、デバッグ、標準、高度なバージョンなど) をビルドするためのリリース フラグを設定できます。これらのエディションを使って、ビルド時に選択済みのリリース フラグに基づいて InstallShield 前提条件を含んだり除外したりできます。たとえば、製品のデバッグ版を作成するときに、ビルドに InstallShield 前提条件を含めたくない場合、リリース フラグを適切な InstallShield 前提条件に割り当てて、リリースに含めるリリース フラグを指定します。

ユーザー インターフェイスのタスク

インポートおよび変換処理では、Visual Studio プロジェクトからのダイアログを InstallShield プロジェクトに組み込みません。プロジェクトをインポートまたは変換した後、InstallShield の [ダイアログ] ビューでプロジェクトに含まれるダイアログの設定を構成することができます。

言語のタスク

Visual Studio プロジェクトを InstallShield プロジェクトにインポートしたときに、以下の条件がある場合、InstallShield はプロジェクトに含まれる既存の文字列エントリ値を Visual Studio プロジェクトの言語のデフォルト文字列エントリ値で置換します:

- Visual Studio デプロイメント プロジェクト ウィザードで、Visual Studio プロジェクトの言語をインポートすることを指定した。
- Visual Studio プロジェクトの言語が InstallShield プロジェクトの言語と一致しない。(Visual Studio で、Localization プロパティがプロジェクトの言語を示します。)

たとえば、Visual Studio デプロイメント プロジェクト ウィザードで Visual Studio プロジェクトの言語をインポートする場合に、InstallShield プロジェクトの言語がスペイン語で、Visual Studio プロジェクトの言語がドイツ語のとき、InstallShield はスペイン語のランタイム文字列をデフォルトのドイツ語翻訳と置換します。このため、[一般情報] ビュー内の "発行者" 設定を更新している場合など、文字列エントリの値を編集した後にウィザードで Visual Studio プロジェクトの言語をインポートすると、InstallShield は "発行者" 設定の値およびその他の文字列の値を、デフォルトのドイツ語の文字列エントリの値に置換します。

このため、Visual Studio プロジェクトのインポート中にプロジェクト言語を変更する場合は、[一般情報] ビューと [テキストとメッセージ] ビューで設定を確認の上、適切な場合は文字列エントリを変更してください。

.NET インストーラー クラスのタスク

Visual Studio プロジェクトに .NET インストーラー クラス カスタム アクションが含まれる場合、InstallShield が変換処理中に .NET アセンブリの .NET インストーラー クラス情報を構成します。(InstallShield は、[ファイルのプロパティ] ダイアログ ボックスの [COM & .NET 設定] タブにある [Installer クラス] チェック ボックスを選択して、実行時に適切なタイミングでアセンブリの Install、Commit、Rollback、および Uninstall メソッドを呼び出すことを示します。)InstallShield は、Visutal Studio における .NET インストーラー クラス カスタム アクションの Condition プロパティをサポートしません。このため、Visual Studio プロジェクトに条件がある .NET インストーラー クラス カスタム アクションが含まれている場合、プロジェクトを変換したあと、InstallShield の [機能] ビューを使って、.NET アセンブリを含む機能の条件を作成したほうが良い場合があります。

追加のタスク

InstallShield の他のビューを利用すると、プロジェクトに追加の変更も行うことができます。

メモ: *Visual Studio では、アプリケーション フォルダーに複数のフォーマットされたプロパティが含まれたディレクトリ パス (例、[ProgramFilesFolder][Manufacturer]¥[ProductName]) を指定できます。Visual Studio プロジェクトは、実行時にディレクトリ カスタム アクションを使用して、パスを解決します。ただし、InstallShield では、この種類のディレクトリ パスはサポートされていません。したがって、InstallShield はパスをインポートおよび変換処理中に解決し、パスの* INSTALLDIR *プロパティを使用します。*

InstallShield のアップデートを取得する

インターネットに接続されている場合、ご使用中の InstallShield バージョンで利用可能な最新の InstallShield 前提条件、マージ モジュール、オブジェクト、およびサービス パック、パッチ、またはその他のアップデートを確認および取得することができます。

タスク ***アップデートを確認するには、以下の手順に従います:***

InstallShield でアップデート ダイアログが表示されます。アップデートが利用可能な場合、次の処理を行うことができます。

- システムに対するアップデートを確認する。
- アップデートに関する説明およびリリース ノートを確認する。
- アップデートをダウンロードおよびインストールする。

InstallShield 実行時の言語サポート

エディション: *言語サポートは、ご利用になる InstallShield のエディションによって異なります。*

特定の言語で実行するインストールを作成することができます。このインストールでは、インストール中に表示される実行時ダイアログのテキストが指定した言語で表示されます。

インストールを複数言語で配布する場合、まず最初に1つの言語でインストールを完成させる必要があります。プロジェクトを作成する場合、[新規プロジェクト] ダイアログ ボックスにある [プロジェクト言語] リストに表示されているサポート言語から 言語を 1 つ選択します。選択された言語が、インストールのすべての実行時ダイアログで使用される言語になります。

アプリケーションを配布する各言語でインストール プロジェクトを別々に作成することができます。たとえばインストールを英語とドイツ語の両方で配布する場合、プロジェクト言語に英語を選択したインストール プロジェクトと、プロジェクト言語にドイツ語を選択した 2つの異なるインストール プロジェクトを作成することができます。

インストールを別の言語で表示する場合、文字列エントリをエクスポートし、その文字列を翻訳し、再びプロジェクトにインポートします。

メモ: *文字列エントリのエクスポートとインポートを行って翻訳しないかぎり、プロジェクト作成後に言語を変更することはできません。*

プロジェクト作成後、[テキストとメッセージ] ビューで実行時ダイアログに表示されるテキストを編集することができます。

エディション: *InstallShield で提供されている Express プロジェクト タイプでは、各インストール プロジェクトにつき 1 つの言語のみサポートされます。複数の言語で実行するインストールを作成する場合は、複数言語で実行可能なプロジェクトを作成するためのビルトイン サポートが搭載されている InstallShield Premier Edition へのアップグレードを推奨します。また、ビルド時とランタイム時の両方において、インストールされるファイルを言語別に細かく管理する機能も提供されています。さらに、InstallShield Premier Edition では、あらかじめ組み込まれている言語以外にも、サポートされていない言語を追加することができます。*

サポートされているランタイム言語

エディション: *言語サポートは、ご利用になる InstallShield のエディションによって異なります。*

アラビア語 (サウジ アラビア) とヘブライ語のサポートは、Premier Edition でのみ提供されています。

次の表は、InstallShield がサポートするランタイム言語、ならびに各言語の言語識別子、または LCID を示します。ID は、特定の言語を識別する整数の値です。

テーブル 2-4 · 言語識別子

言語	Identifier
バスク語	1069
ブルガリア語	1026
カタロニア語	1027
中国語 (簡体字)	2052

テーブル 2-4 · 言語識別子 (続き)

言語	Identifier
中国語 (繁体)	1028
クロアチア語	1050
チェコ語	1029
デンマーク語	1030
オランダ語	1043
英語 (U.S.)	1033
フィンランド語	1035
フランス語 (カナダ)	3084
フランス語 (フランス)	1036
ドイツ語	1031
ギリシャ語	1032
ハンガリー語	1038
インドネシア語	1057
イタリア語	1040
日本語	1041
韓国語	1042
ノルウェー語	1044
ポーランド語	1045
ポルトガル語 (ブラジル)	1046
ポルトガル語 (ポルトガル)	2070
ルーマニア語	1048
ロシア語	1049
セルビア語 (キリル)	3098
スロバキア語	1051

テーブル 2-4 · 言語識別子 (続き)

言語	Identifier
スロヴェニア語	1060
スペイン語	1034
スウェーデン語	1053
タイ語	1054
トルコ語	1055

エディション· *InstallShield で提供されている Express プロジェクト タイプでは、各インストール プロジェクトにつき 1 つの言語のみサポートされます。複数の言語で実行するインストールを作成する場合は、複数言語で実行可能なプロジェクトを作成するためのビルトイン サポートが搭載されている InstallShield Premier Edition へのアップグレードを推奨します。また、ビルド時とランタイム時の両方において、インストールされるファイルを言語別に細かく管理する機能も提供されています。さらに、InstallShield Premier Edition では、あらかじめ組み込まれている言語以外にも、サポートされていない言語を追加することができます。*

サポートされているアプリケーション プログラミング言語

InstallShield は、Java, Pascal, C++, Visual Basic, Delphi, C# .NET, Visual Basic .NET, ASP .NET, および Cobol など、*任意のプログラム言語でアプリケーションのインストールを作成できます。*

上記の言語以外でアプリケーションを作成した場合でも、InstallShield 使ってそのアプリケーションのためのインストールを作成することができます。また、アプリケーションを配布はしないけれども、ファイルやデータベースをパッケージしたい場合、InstallShield を作業のために使用できます。

3

チュートリアル

このセクションでは、InstallShield を使ったインストールの作成手順をステップ バイ ステップで説明します。

テーブル 3-1 ・チュートリアルの種類

チュートリアル	説明
基本チュートリアル	InstallShield ビュー リストを使って、基本のインストール作成方法についてステップ バイ ステップで説明します。

基本チュートリアル

このチュートリアルは、InstallShield ビュー リストを使いながら、基本のインストールを作成する方法をステップバイ ステップで説明します。必要なファイルはすべて、*InstallShield Program Files フォルダー*¥Samples¥WindowsInstaller¥Basic Installation Project¥Data Files サブフォルダーにあります。

基本チュートリアルは次のタスクで構成されています。

- プロジェクトの新規作成
- セットアップの編成
- アプリケーション データの指定
- ターゲット システムの構成
- セットアップ外観のカスタマイズ
- セットアップ要件およびアクションの定義
- リリースの準備
- 概要

プロジェクトの新規作成

タスク ***チュートリアルを開始するには、以下の手順に従います:***

1. InstallShield を起動します。
2. [**ファイル**] メニューで、[**新規**] をクリックします。[**新規プロジェクト**] ダイアログ ボックスが開きます。
3. [**プロジェクト名**] ボックスに、**Othello.ise** と入力します。
4. [**プロジェクト言語**] ボックスで、**英語 (U.S)** を選択します。
5. [**OK**] をクリックします。

InstallShield のインストール開発環境 (IDE) が開きます。

セットアップの編成

プロジェクトのビルドに必要な最初の手順として、編成情報を設定します。これは、[ビュー リスト] にある [セットアップの編成] の手順に関連したビューで行います。

一般情報

[一般情報] ビューには、アプリケーションの情報、会社名、アプリケーションについて詳細を知るために顧客がアクセスできる Web サイトなどの情報を入力できます。

タスク ***一般情報を構成するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. **"製品名"** 設定に次を入力:

   Othello

3. **INSTALLDIR** 設定に次を入力:

   [ProgramFilesFolder]Othello

4. **"サブジェクト"** 設定に次を入力:

   Othello

5. **"作成者"** 設定には、自分の名前を入力します。
6. **"[概要情報ストリーム] のコメント"** 設定には次を入力:

   このインストールには、My Sample をインストールするために必要なロジックとデータが含まれています。

7. このパッケージのアイコンを提供します:
   a. **"表示アイコン"** 設定をクリックしてから、省略記号ボタン (...) をクリックします。**[アイコンの変更]** ダイアログ ボックスが開きます。
   b. **[参照]** ボタンをクリックします。
   c. 次のディレクトリに移動します:

      InstallShield Program Files フォルダー¥Samples¥WindowsInstaller¥Basic Installation Project¥Data Files

   d. Othello.exe を選択します。
   e. **[開く]** ボタンをクリックします。**[開く]** ダイアログ ボックスが閉じて、InstallShield が選択されたファイルのパスと名前を **"ファイル名"** 設定に追加します。
   f. **[OK]** をクリックします。

機能

インストール編成の次のステップは、機能を作成することです。これらの機能は、実行時にファイルとデータをインストールするために使用します。InstallShield は自動的に "常にインストール" と呼ばれる機能を作成します。これはプロジェクトに唯一必要な機能です。ただし、これを構成する必要があります。

タスク ***[常にインストール] 機能を構成するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストから **[機能]** をクリックします。
2. **"説明"** 設定に次を入力:

   Othello ゲームの実行可能ファイル。

3. **"必須"** 設定で **[はい]** を選択します。
4. **"可視"** 設定で **[非表示]** を選択します。
5. **"アドバタイズ"** 設定で **[アドバタイズ不可]** を選択します。

6. **"コメント"** 設定に次を入力:

   これが唯一の機能です。

セットアップの種類

[セットアップの種類] ビューでは、インストール中にエンド ユーザーが選択できるセットアップを設定できます。このインストールでは、必要なインストールの種類は 1 つだけです。

タスク ***セットアップの種類を構成するには、以下の手順を実行します。***

1. **[セットアップの編成]** の下にあるビュー リストにある **[セットアップの種類]** をクリックします。
2. [**標準**] チェック ボックスを選択します。
3. **[最小]** および **[カスタム]** チェック ボックス をクリアします。

メモ: *このチュートリアルでは残りのビューはこのプロジェクトと関係ないため無視します。ご自分でプロジェクトを作成される場合は、これらのビューが必要か確認してください。*

アプリケーション データの指定

この基本セットアップの設定の次の手順として、アプリケーションデータを指定します。このチュートリアルでは、既に定義されている機能へファイルの追加します。

ファイル

タスク ***既存の機能にファイルを追加するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. このビューの上部で、*InstallShield Program Files フォルダー*¥Samples¥WindowsInstaller¥Basic Installation Project¥Data Files フォルダーに移動してから、そのフォルダーをクリックして開きます。**[ソース コンピューターのファイル]** ペインにファイルが表示されます。
3. **[インストール先コンピューターのフォルダー]** ペインで、フォルダー名の一部として **[INSTALLDIR]** を含むフォルダーを見つけます。
4. **[ソース コンピューターのファイル]** ペイン から、Othello.exe と 3 つの .gif ファイルを **[インストール先コンピューターのフォルダー]** ペインにある [INSTALLDIR] フォルダーへドラッグします。

ターゲット システムの構成

ターゲット システムの設定には、ショートカット作成、レジストリ変更、ODBC リソースの設定、.ini ファイルの変更、ファイル拡張子の設定、および環境変数の設定などが含まれます。ただしこのチュートリアルでは、ショートカットの作成のみを取り上げます。

ショートカット/フォルダー

[ショートカット/フォルダー] ビューで、ターゲット システムにアプリケーションのショートカットを作成することができます。

タスク ***製品のショートカットを作成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[ショートカット/フォルダー]** をクリックします。
2. **[ショートカット]** エクスプローラーで、**[プログラム メニュー]** を右クリックしてから、**[新しいショートカット]** を選択します。**[ショートカット ターゲットを参照する]** ダイアログ ボックスが開きます。
3. [ファイル] ビューでプロジェクトに追加した Othello.exe ファイルのインストール先を参照します。
4. ショートカットを **Othello** と名付けます。
5. **"説明"** 設定に次を入力:

   Othello ゲームへのショートカット
6. **"作業ディレクトリ"** 設定で **[INSTALLDIR]** を選択します。
7. **"実行"** 設定で **[標準ウィンドウ]** を選択します。
8. **"機能"** 設定で **[常にインストール]** を選択します。

メモ: *このチュートリアルでは残りのビューはこのプロジェクトと関係ないため無視します。ご自分でプロジェクトを作成される場合は、これらのビューが必要か確認してください。*

セットアップ外観のカスタマイズ

セットアップの外観をカスタマイズする作業には、インストール中にエンドユーザーに表示されるダイアログの設定、ダイアログの表示方法の指定、インストール中にビルボードを表示するかどうかの指定、および表示するテキストの指定などが含まれます。このチュートリアルでは、ダイアログを設定するだけにします。

ダイアログ

[ダイアログ] ビューでは、インストールのユーザー インターフェイスを構成するエンド ユーザー ダイアログを選択および設定できます。チュートリアルでは、"グローバル ダイアログ設定" プロパティはデフォルト値のまま残します。

タスク ***エンドユーザーに表示するダイアログをカスタマイズするには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ダイアログ]** をクリックします。
2. **[ダイアログ]** エクスプローラーで、**[使用許諾契約書]** チェック ボックスをクリアして、**[インストール先フォルダー]** チェック ボックスを選択します。
3. **[インストール先フォルダー]** ダイアログを選択してから、**"インストール先変更の表示"** 設定で **[はい]** を選択します。

メモ: このチュートリアルでは残りのビューはこのプロジェクトと関係ないため無視します。ご自分でプロジェクトを作成される場合は、これらのビューが必要か確認してください。

セットアップ要件およびアクションの定義

[セットアップ要件およびアクションの定義] ビュー リストに含まれるビューでは、アプリケーションのインストールに必要な条件を指定し、Windows Installer で本来サポートしていない機能を追加し、またインストール中に必要なファイルを追加できます。このチュートリアルではインストールの最小要件だけを定義します。

要件

[要件] ビューでターゲット システムの必要条件を設定します。これらの条件が満たされない場合、製品はエンドユーザーのシステム上にインストールされません。Othello は比較的シンプルなアプリケーションですが、Pentium プロセッサが必要です。

タスク *インストールの必要条件を設定するには、以下の手順に従います:*

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[要件]** をクリックします。
2. **"プロセッサ"** 設定で、**[Pentium、またはそれ以上]** を選択します。

メモ: このチュートリアルでは残りのビューはこのプロジェクトと関係ないため無視します。ご自分でプロジェクトを作成される場合は、これらのビューが必要か確認してください。

リリースの準備

[リリース] ビューでは、配布メディアの構成/ビルド/テストのほか、それをエンドユーザーに配布することができます。

リリースのビルド

[リリース] ビューでは、インストール情報を .msi ファイルにコンパイルします。

タスク *このチュートリアルプロジェクトをリリースとしてビルドするには以下の手順に従います:*

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、**CD_ROM** を右クリックして **[ビルド]** を選択します。

ビルドが始まると、InstallShield インターフェイスの下部に [出力] ウィンドウが表示され、ビルドの進行状況とエラーメッセージがここに表示されます。

ビルドプロセスが完了すると、インストール ファイルは Othello プロジェクトがあるフォルダーのサブフォルダーに配置されます。

ヒント: *このチュートリアルでは CD_ROM リリースのビルドについて説明していますが、メディアの種類には他のオプションもいくつかあります。たとえば、WebDeployment タイプを使うと、Web 上での配布用に特別に設計されたインストールを作成することができます。*

リリースのテスト

[リリース] ビューではまた、インストール プログラムが期待通りに動作するかどうかをプレビューすることができます。

タスク ***リリースをテストするには次の操作を実行します。***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、**CD_ROM** を右クリックして **[セットアップの実行]** を選択します。インストールが起動して、Othello プログラムをインストールします。
3. インストール プログラムの指示に従います。これは、このインストール プログラムを配布したときにエンドユーザーが経験するランタイム経験と同じです。
4. インストールが完了したら、ファイルとデータが正しくインストールされていることを確認します。
5. C:¥Program Files¥Othello (またはこのプログラムのインストール先に選んだフォルダー) に移動して、Othello.exe があることを確認します。
6. デスクトップに **Othello** と名づけられたショートカットがあることも確認します。このショートカットをダブルクリックして、正しく設定されていることを確認します。

リリースの配布

このインストール作成における最終段階は、ターゲット メディアへの配布です。これはチュートリアルであって顧客に配布するアプリケーション用のインストールではないので、この手順はオプションとなります。

タスク ***リリースを配布するには、次の手順を実行します。***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. CD-ROM ドライブに空の CD-ROM が入っていることを確認します。
3. **[ビルド]** エクスプローラーで、**CD_ROM** をクリックします。
4. **[イベント]** タブをクリックします。
5. ***"フォルダーにコピー"*** 設定をクリックします。この設定の右側に省略記号ボタン (...)] ボタンが表示されます。
6. 省略記号ボタン (...) をクリックします。**[フォルダーの参照]** ダイアログ ボックスが開きます。
7. CD-ROM ドライブを参照して **[OK]** をクリックします。
8. **[ビルド]** エクスプローラーで、**CD_ROM** を右クリックして **[配布]** を選択します。

CD-ROM への直接コピーが可能な場合、必要なインストール ファイルはすべてこの場所にコピーされます。

概要

チュートリアルの完了です。ここで習得したことをインストール プロジェクトに適用してみましょう。ほとんどのインストール プロジェクトには、このチュートリアルで説明してある手順とコンセプトが必要です。

4

インストールの作成

コンピューターにアプリケーションをインストールしたことがある方は、実行中のインストールを既にエンドユーザーの視点でご覧になっています。インストールの主要な役割は、ソース メディアからローカル ドライブへファイルを転送することです。インストールはまた、ユーザー インターフェイスを表示してエンドユーザーの選択を取得し、ターゲット システムを構成し (たとえば、必要なレジストリ エントリやショートカットの作成)、またインストール済みアプリケーションの変更またはアンインストールを行います。インストールの作成は次のタスクの一部またはすべての処理を含みます。

インストール情報の指定

[一般情報] ビューで入力した基本情報はインストールのいろいろな箇所で使用されます。

ファイルの編成と転送

ファイル転送は CD または DVD のようなソース メディアからファイルをエンドユーザーのマシンにあるローカル ドライブへコピーします。エンドユーザーが選択した構成 (セットアップの種類や機能の選択) にしたがって、すべてのファイルまたは一部のみがローカル ディスクに転送されます。

インストールするファイルをセットアップの種類および機能に編成して、エンドユーザーが最も適切なファイルを選択できるようにします。

ターゲット システムの構成

ファイルをインストールするだけでなく、ショートカットやプログラムフォルダーの作成、レジストリの変更、初期化ファイル (.ini ファイル) データの変更、ODBC リソースの構成、環境変数や Windows サービスのインストールと開始など、多くのインストールではターゲット システムの構成が必要です。

インストール動作のカスタマイズ

InstallShield では、幅広いカスタマイズ オプションが提供されています。インストールはカスタム アクションを使って、.dll 関数の呼び出し、実行可能ファイルの起動、VBScript コードの実行、または JScript コードの実行を行います。

エンドユーザー インターフェイスの定義

インストールのエンドユーザー インターフェイスは、エンドユーザーに対して情報およびインストール構成オプションを提供します。エンドユーザーは、ユーザー インターフェイスを通して製品の一部をインストールするか、一部のファイルをソース メディアに残すか、使用許諾契約を表示するか、またはインストールを正しく構成するためのインストーラーに情報を提供するかなどを選択できます。

ユーザー インターフェイスはインストールのニーズに合わせたカスタマイズが可能です。たとえば、ソフトウェアを不正な使用から保護するために、インストールの前にシリアル番号の入力をユーザーに求めるように指定できます。ファイル転送中、インストールは新機能や便利なヒントなどの製品情報を提供するビルボードを表示します。ファイル転送処理の進行状況がわかるステータス バーも表示されます。

サーバーの構成

サーバー側のインストールでは、場合によって、新しいインターネット インフォメーション サービス (IIS) Web サイトの作成および管理、または COM+ アプリケーションとコンポーネントの管理が必要です。

インストールに FlexNet Connect のアップデート通知機能を追加する

FlexNet Connect を利用して、Web に接続しているエンドユーザーに対してアプリケーションのパッチ、アップデート、および製品情報が入手可能であることを自動的に通知することができます。FlexNet Connect 機能を活用するためには、元のインストールで FlexNet Connect の有効化が必要です。

インストールにメンテナンスとアンインストール機能を追加する

アプリケーションのアンインストール、変更、または修復を行うためには、アプリケーションの存在がオペレーティング システムで示されなくてはなりません。これに対応するには、アプリケーションを簡単にメンテナンスまたはアンインストールできるように、インストールでアプリケーションをオペレーティング システムに登録します。

この処理で登録されるほとんどの情報は、コントロール パネルの [プログラムの追加と削除] でエンドユーザーに提供されます。たとえば、テクニカル サポートの連絡先情報、製品更新の情報、製品のバージョン、および製品発行元の情報はこのプロセスで登録されます。

インストールのビルド、テスト、および配布

インストール プロジェクトを作成した後は、インストールのビルド、テストおよび配布が必要です。つまりユーザーへリリースするファイルを作成し、インストールのエラーをテストし、オプションとしてローカル、ネットワークまたは FTP サイトへファイルをコピーします。

まず始めに

InstallShield ヘルプ ライブラリの「まず始めに」は、インストール作成者が InstallShield で新規のインストール プロジェクトを作成するときに役に立つ情報が盛り込まれています。各トピックでは、Windows Installer、Designed for Windows ロゴ ガイドライン、およびインストール開発のその他の分野についてのバックグランド情報を読むことができます。

Windows Installer 入門

Windows Installer インストール プログラムは .msi パッケージとして配布され、パッケージは Windows Installer データベース (.msi データベース) と関連データ ファイル (.cab ファイル、非圧縮データ ファイルなど) から構成されます。.msi データベースは COM 構造のストレージとして実装され、ターゲット システム上で処理される変更について記述する何十ものテーブルで構成されます。次に、.msi テーブルのいくつかの例を挙げます:

- File – インストールされるファイルを記述する
- Registry – 書き込まれるレジストリ データを記述する
- Shortcut – ショートカットの設定を記述する

その他の .msi データベース テーブルは、インストールのユーザー インターフェイスの外観と動作を記述したり、Windows サービスおよび ODBC 情報を構成したり、ターゲット システムの特徴を確定したり、インストール中に使用されるアイコンやその他のバイナリ データを保管したりします。

開発者の視点から見て、Windows Installer インストール プログラムの最も大きな変更点は、スクリプトを明示的に書き込む必要がないという点です。その代わり、Windows Installer ベースのインストールは標準カスタムアクションを実行して、ダイアログを表示したり、ターゲット システムをクエリしたり、ターゲット システムに変更を加えたりします。これらのアクションはシーケンスに配置されていて、アクションの集合が順番に並べられています。

Windows Installer は製品のインストール管理専用のアプリケーション プログラム インターフェイス関数の集合 (または API) を含みます。Windows Installer で提供されている機能を活用するには、アプリケーションが Windows Installer API を呼び出す必要があります。

Windows オペレーティング システムに統合されている Windows Installer は、アプリケーションやシステム ツールを管理するためのインターフェイスだけでなく、コンポーネントの管理を行うための標準フォーマットも提供します。様々なバージョンの Windows Installer が Windows オペレーティング システムの再配布可能ファイルとして利用できます。

.msi データベース テーブルを直接編集して Windows Installer パッケージを作成することは可能ですが、多数のテーブルとその関連性を編集するには非常な困難を伴います。InstallShield は、Windows Installer 用のインストール開発過程を様々なビューに分け、.msi データベースにまつわる実装についての詳細の多くのを開発者の目から遮るグラフィカル エディターやウィザードを提供します。

インストール中におけるユーザー アカウント制御のプロンプトの数を最小化する

Windows Vista 以降およびユーザー アカウント制御 (UAC) の目的は、ユーザーが常に標準ユーザーとして実行できるようにすることです。昇格が必要になることは、ほとんどありませんが、必要となる場合、できるだけ短い時間に限られる必要があります。

InstallShield のいくつかの異なる領域では、昇格された権限を要求するために、インストールで UAC の同意または資格情報プロンプトを起動するかどうかが問題になります。Windows Vista 以降のシステムでエンド ユーザーがインストールを実行する際の UAC の動作を適切に定義するためには、これらの異なる設定をよく理解することが必要です。また、これらの設定を使って、場合により、インストール中に表示される UAC プロンプトの数を最小限に抑えることもできます。

構成方法によって、InstallShield 前提条件を含むインストールがインストール中のいくつかの時点で、Windows Vista 以降のシステム上で昇格された権限のプロンプトを表示することができます:

1. エンド ユーザーが Setup.exe ファイルを起動するとき
2. Setup.exe ファイルが、昇格された権限を必要とするセットアップ前提条件を起動するとき
3. インストールされる 1 つまたは複数の機能に関連付けられた機能前提条件が含まれるため、ISInstallPrerequisites カスタム アクションが Setup.exe ファイルを機能前提条件インストール モードで再起動するとき

   ISInstallPrerequisites カスタム アクションは、昇格された権限のプロンプトが表示される前に、機能前提条件が昇格された権限を必要とするかどうかを検証しませんので、ご注意ください。また、ISInstallPrerequisites カスタム アクションは、機能前提条件のインストールが必要かどうかを決定するために機能前提条件の条件を確認することもありません。昇格された権限のプロンプトは常に表示されます。
4. Windows Installer が .msi パッケージの [実行] シーケンスを開始するとき

InstallShield における UAC 関連の設定

以下は、Windows Vista 以降のシステム上で、インストール中に UAC プロンプトを表示するかどうかを判断するための InstallShield の設定です。

- **必要実行レベル**—[リリース] ビューにあるこの設定を使用して、インストールの Setup.exe ファイルが必要とする最小実行レベルを指定します。InstallShield は、Setup.exe ランチャに埋め込まれるアプリケーション マニフェストで選択した値 (管理者、最高権限、または起動者) を使用します。詳細については、「Windows Vista 以降のプラットフォームでのセットアップランチャーの必要実行レベルを指定する」を参照してください。
- **(InstallShield 前提条件で) 管理者権限を必要とする**—[再配布可能ファイル] ビューにある InstallShield 前提条件ための [詳細] ペインに、前提条件のインストールに管理者権限が必要かどうかを示す読み取り専用設定があります。
- **管理者権限が必要 ([一般情報] ビュー)**—[一般情報] ビューにあるこの設定を使用して、インストールの .msi パッケージの [実行] シーケンスで管理者権限が必要かどうかを指定します。[いいえ] を設定した場合、InstallShield は Word Count Summary プロパティ の 3 ビット目を設定して、製品のインストールに昇格された権限が必要であることを示します。詳細については、「[一般情報] ビュー」を参照してください。

また、InstallShield 前提条件の種類 (セットアップ前提条件または機能前提条件)—は、UAC プロンプトが Windows Vista 以降のシステム上でのインストール中に表示されるかどうかに影響を及ぼす可能性があります。これら 2 種類の InstallShield 前提条件についての詳細は、「セットアップ前提条件と機能前提条件の違い」を参照してください。

以下は、Windows Vista 以降における UAC 関連の動作についての注意点です:

- ″必要実行レベル″ が [起動者] に設定されていて、インストールが管理者権限を必要とする InstallShield 前提条件を含み、[一般情報] ビューの ″管理者権限″ が [いいえ] に設定されている場合、インストール中、プロンプトはエンドユーザーに表示されません。
- ″必要実行レベル″ が [起動者] に設定されていて、インストールが管理者権限を必要とするセットアップ前提条件を含み、[一般情報] ビューの ″管理者権限″ が [いいえ] に設定されている場合、インストール中、プロンプトはエンドユーザーに 1 回のみ表示されます。再起動がある場合、再起動ごとに、1 回のみ UAC プロンプトが表示されます。
- セットアップランチャーの完全ユーザー インターフェイスが表示される場合で、インストールする必要があるセットアップ前提条件がインストールに含まれているとき、セットアップランチャーは通常、メイン インストールが開始する前にセットアップ前提条件ダイアログを表示します。インストールが必要な 1 つまたは

複数のセットアップ前提条件が管理者権限を必要とするとき、メッセージ ボックスの [インストール] ボタンが、エンドユーザーに昇格された権限が必要であることを通知するシールド アイコンと共に表示されます。

- 再起動後、インストールが続行し、権限の昇格が必要な場合、続行メッセージ ボックスの [OK] ボタンがシールド アイコンと共に表示されます。権限の昇格が必要がない場合、シールド ボタンは表示されません。
- インストールに、ターゲット マシンにインストールしなければならないセットアップ前提条件が複数あり、このうち 1 つまたは複数のセットアップ前提条件が管理者権限を必要とする場合、UAC プロンプトが、一番最初のセットアップ前提条件がインストールされる前に表示されます。これにより、それぞれの前提条件のインストールごとに別々の UAC プロンプトを要求することなく、昇格された権限をすべての前提条件に対して使用できるようになる場合があります。ただし、セットアップ前提条件のインストールで再起動がある場合、管理者権限は失われることになり、これにより、残りの前提条件が管理者権限を必要とする場合、UAC プロンプトが表示されることがありますので注意してください。
- インストールに、ターゲット マシンにインストールしなければならないセットアップ前提条件が複数あり、このうち 1 つまたは複数のセットアップ前提条件が管理者権限を必要とする場合、UAC プロンプトが、一番最初のセットアップ前提条件がインストールされる前に表示されます。これにより、それぞれの前提条件のインストールごとに別々の UAC プロンプトを要求することなく、昇格された権限をすべての前提条件に対して使用できるようになる場合があります。ただし、セットアップ前提条件のインストールで再起動がある場合、管理者権限は失われることになり、これにより、残りの前提条件が管理者権限を必要とする場合、UAC プロンプトが表示されることがありますので注意してください。

  機能前提条件では動作が若干異なります。インストールが、前提条件に関連付けられている機能をインストールする場合、ISInstallPrerequisites カスタム アクションが機能前提条件インストール モードで Setup.exe を再起動するときに UAC プロンプトが表示されます。これは、機能前提条件のいずれかが昇格された権限を必要とするかどうかに関わらず発生します。また、これは機能前提条件のインストールが必要かどうかを決定するために機能前提条件のいずれかが評価される前に発生します。機能前提条件インストールによって再起動が行われる場合、管理者権限は失われますのでご注意ください。再起動のあと、再び ReadyToInstall ダイアログが表示され、エンド ユーザーは残りのインストールを続行するために [インストール] ボタンをクリックする必要があります。この場合、ISInstallPrerequisites カスタム アクションが機能前提条件インストール モードで Setup.exe を再起動するとき、UAC プロンプトが再び表示されます。
- [一般情報] ビューの "管理者権限が必要" が [いいえ] に設定されている状態で、適切な権限をもたずに .msi パッケージがタスクを実行しようとすると、Windows Installer によって実行時エラーが表示される場合があります。
- 権限がインストールの終わりで昇格され、[セットアップの完了] ダイアログが製品を起動すると、昇格された権限が製品に引き継がれます。通常、昇格された権限を使用したアプリケーションの実行は推奨されていません。

例: UAC プロンプトが Setup.exe と、再起動後に管理者権限が必要な前提条件に表示される

次のダイアグラムでは、Windows Vista 以降が標準ユーザーまたは限られた権限を持つ管理者ユーザーに昇格された権限を要求しています。この例は、Windows Vista 以降のシステム上の UAC のデフォルト設定に基づいています。この図表は、Setup.exe、2 つのセットアップ前提条件、機能前提条件、および .msi パッケージの [実行] シーケンスで昇格が必要なインストールを説明します。Setup.exe ファイルに、管理者を必要実行レベルとして指定するマニフェストがあるため、昇格された権限はインストールの各部分で使用されます。昇格された権限は再起動中に失われるため、2 つ目の UAC プロンプトが表示されます。

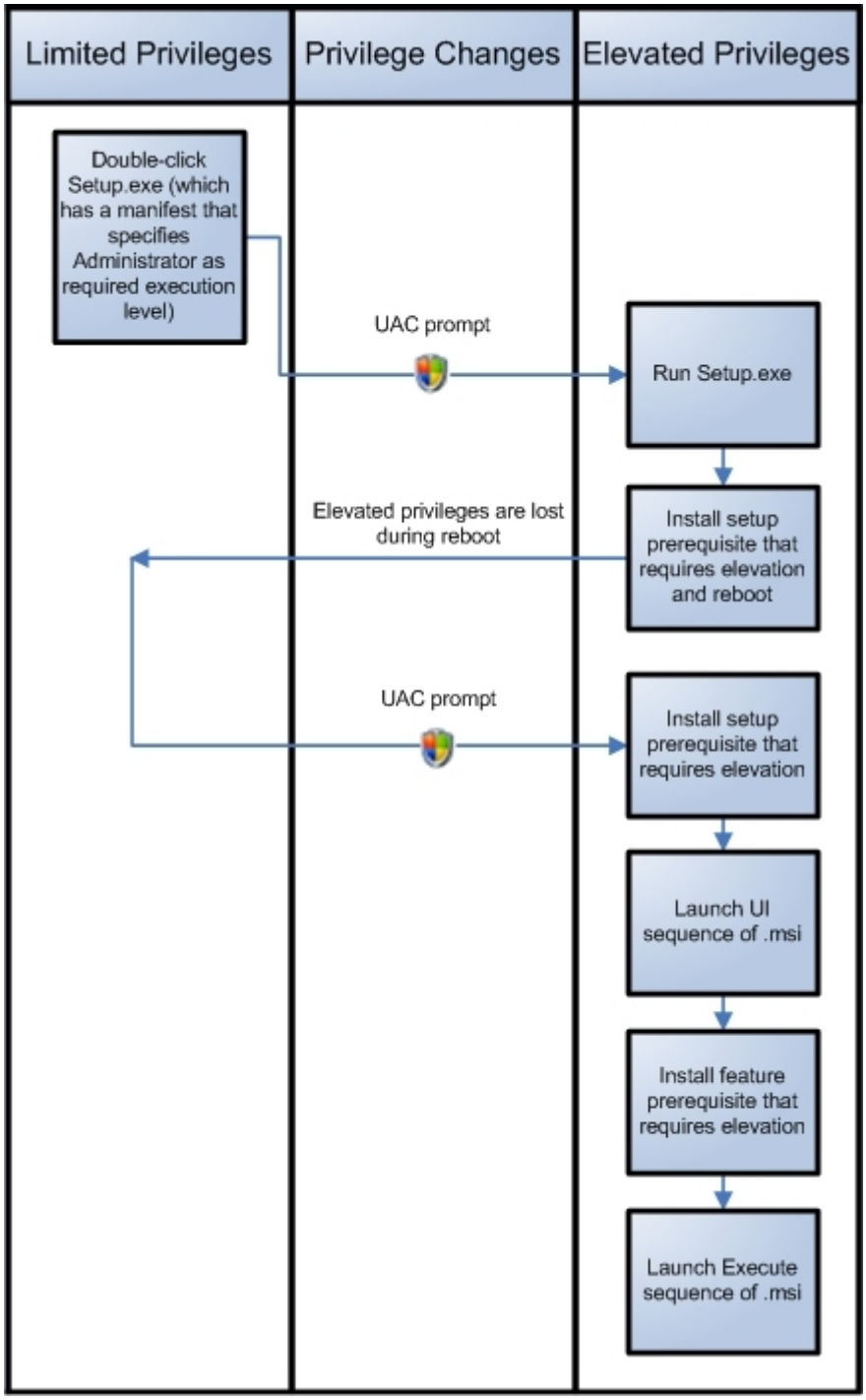


図 4-1: ″必要実行レベル″ が [管理者] に設定されているインストールのダイアグラム

Windows ロゴ プログラムの要件

WWindows 8 デスクトップ アプリの認定を受けるために製品とそのインストールが満たさなくてはならない要件の一覧が、マイクロソフトによって確立されています。要件では、製品が Windows システムで実行されたとき、より高い互換性と安定性、および安全性を持つことができる基準が規定されています。Windows 8 デスクトップ アプリの認定要件を満たす製品は、Compatible with Windows 8 ロゴを表示できます。

Windows ロゴの取得に関する詳細は、「MSDN」を参照してください。この Web サイトには、Windows ロゴ プログラムに関する情報が掲載されています。

エディション: *InstallShield Premier と InstallShield には、製品がマイクロソフトの Windows ロゴ プログラムのインストール要件を満たしているかどうかを識別するときに役立つ Windows 検証スイートが含まれています。パッケージまたはマージ モジュールが 1 つ以上の検証規則に違反した場合、InstallShield は違反の対象となった特定の規則をレポートし、問題を解決するための追加情報を提供します。*

Windows ロゴ: *InstallShield ヘルプ ライブラリ全体を通して、情報が Windows ロゴ プログラム ガイドラインのコンプライアンスに関連する箇所では Windows Logo ガイドライン アラートが表示されています。*

現在のインストールによる同製品の将来のメジャー バージョンの上書きを防ぐ

プロジェクト: *この情報は Express プロジェクトに適用します。*

InstallShield には、現在のインストールが同製品の将来のメジャー バージョンを上書きするのを防ぐ機能が含まれています。すべての新しい Express プロジェクトには、デフォルトで、この機能が含まれています。

Windows ロゴ: *Windows ロゴ プログラムの要件に従うと、インストール パッケージは、エンド ユーザーによって製品の以前のバージョンがインストールされたとき、将来のバージョンが上書きされるのを防ぐ必要があります。*

タスク

エンドユーザーが製品の現在のバージョンをインストールしたときに、将来のメジャー バージョンが上書きされるのを防ぐためのサポートを追加するには、以下の手順に従います:

1. **[セットアップの編成]** の下にあるビュー リストにある **[アップグレード パス]** をクリックします。
2. **[アップグレード パス]** エクスプローラーを右クリックし、**[ダウングレードの防止]** をクリックします。

ISPreventDowngrade アイテムが [アップグレード パス] エクスプローラーに追加されます。エンドユーザーがこのバージョンの製品をより新しいバージョンの製品に上書きインストールしようとしたとき、エラー メッセージが表示され、新しいバージョンが既にインストールされていることを通知します。エンドユーザーがエラー メッセージ ボックスを閉じると、インストールは終了します。

ヒント: プロジェクトからダウングレード防止機能を削除するには、[アップグレード パス] ビューにある ISPreventDowngrade アイテムを右クリックしてから、[削除] をクリックします。

非管理者パッチのインストールを準備する

Windows Installer 3.0 以上では、管理者以外によるインストールが可能なパッチを作成することができます。非管理者によるパッチは厳しい条件が満たされたときのみ利用することができます。たとえば、パッチが更新するベース インストールは、パッチ パッケージの署名に使用される証明書を含まなくてはなりません。満たされなければならない他の基準については、「非管理者パッチ」を参照してください。

タスク

後で非管理者 QuickPatch パッケージによってアップデートが可能なベース インストールを作成するには、次の手順を実行します。

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、作成または変更するメディア タイプを選択します。
3. **[署名]** タブをクリックします。
4. デジタル署名情報を指定します。

InstallShield が、必要な情報を MsiDigitalCertificate テーブルおよび MsiPatchCertificate テーブルに追加します。これによって、非管理者によるインストールが可能な QuickPatch パッケージの作成が可能となります。

インストール情報を指定する

インストール プロジェクトを作成を開始するとき、まず最初にインストールの重要情報を指定する必要があります。インストール情報は、インストールを編成するときに設定および決定が必要な上位レベルのプロパティおよび考慮によって定義することができます。これには、製品およびプロジェクトのプロパティの指定と ″プログラムの追加と削除″ のプロパティの構成が含まれます。

一般的なプロジェクト設定を構成する

InstallShield は、プロジェクト設定を単一インストール プロジェクトファイル (.ise ファイル) に保管します。このファイルには、プロジェクトについてのすべての情報が格納されています。[一般情報] ビューを使うと、作成者名、プロジェクトがサポートする言語、および記入するコメントなど、インストール プロジェクトについての基本情報を編集することができます。

[一般情報] ビューではまた、製品名、製品コード (GUID)、およびバージョン番号といった一般的な製品情報も構成できます。製品は、インストール プロジェクトの構成中で最上位のものです。インストールは複数の機能に分割でき、これらが製品のサブセットになります。

各プロジェクトとプロジェクト設定についての詳細は、「[一般情報] ビュー」を参照してください。

製品コードの設定

製品コードは製品を識別する固有の文字列です。インストールは実行時に製品コードを使用して、その製品が既にインストール済みであるかどうかを判断します。

製品を一意に識別する GUID を入力するか、[一般情報] ビューの ″製品コード″設定にある **[新しい GUID の生成]** ボタン ([...]) をクリックして、自動的に新しい GUID を生成します。インストールが、この GUID を実行時に登録します。

注意: *この製品コードは製品を一意に識別するため、製品のリリースを配布してからはこのコードを変更しないことをお勧めします。*

製品名の指定

[一般情報] ビューの ″製品名″ 設定に製品の名前を入力します。ここに入力する名前は、インストールを作成する製品の名前です。この値は、プロジェクト全体を通して、次のような名前に使用されます。

- プロジェクトの場所の下にあるソースファイルフォルダーの名前
- InstallShield がビルドする Windows Installer パッケージ (.msi ファイル) の名前
- 実行時ダイアログ
- Windows ロゴ要件に従って登録される情報レジストリキー情報値は以下の場所にあり、エンド ユーザーが製品を変更または削除するのに使用できる [プログラムの追加と削除] で使用されます。

  HKEY_LOCAL_MACHINE¥Software¥Microsoft¥Windows¥CurrentVersion¥Uninstall¥<製品コード>

入力値はソース ファイルのパスに組み込まれるため、製品名には次のいずれの文字も含めることはできません: ¥ / : * ? ″ < > | -

ヒント: 製品名にアンパサンド (&) を含める場合、エンド ユーザーダイアログで名前を正しく表示するには、2 つのアンパサンド (&&) を使用する必要があります。たとえば New & Improved Product *と表示する場合、製品名を* New && Improved Product *と入力します。*

製品バージョンを指定する

製品のバージョン番号を指定するとき、必ず有効な製品バージョンを入力してください。バージョンには、数値のみを使用できます。一般的なフォーマットは *aaa.bbb.ccccc* または *aaa.bbb.ccccc.ddddd* で、*aaa* はメジャー バージョン番号、*bbb* はマイナー バージョン番号、*ccccc* はビルド番号、および *ddddd* はバージョン番号を示します。*aaa* と *bbb* の最大値は 255 です。*ccccc* と *ddddd* の最大値は、65,535 です。

実行時に、インストールが、インストールされる製品のバージョン番号を登録します。完全なバージョン文字列が [プログラムの追加と削除] に表示されます。製品バージョン番号は、その一部をインストール エンジンがアップグレードの適用を判断するために使用するため、重要です。

製品バージョンは、[一般情報] ビューで構成できます。

製品のバージョンを指定するときに 4 番目のフィールド (*ddddd*) を含めることもできますが、インストールは異なる製品バージョンを区別するときに製品バージョンのこの部分を無視します。

リリースに Setup.exe が含まれる場合、指定した製品バージョンが Setup.exe の [プロパティ] ダイアログ ボックスに表示されます。詳細については、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。

アップグレード コードを設定する

アップグレード コードは関連のある一連の製品を識別する GUID です。Windows Installer は、インストール済みの製品のメジャー アップグレードを行う際に、製品のアップグレード コードを使用します。UpgradeCode プロパティに格納されるアップグレード コードは、製品のすべてのバージョンにおいて同一でなくてはなりません。

新しい製品の GUID を構成するには、[一般情報] ビューの "アップグレード コード" 設定を使用します。製品のより新しいバージョンのアップグレードを作成する場合、[アップグレード パス] ビューまたは QuickPatch プロジェクトで同じアップグレード GUID を入力します。

[プログラムの追加と削除] 情報を構成する

[一般情報] ビューの [プログラムの追加と削除] 領域にある設定は、[プログラムの追加と削除] ツールに表示される情報です。

特定の "プログラムの追加と削除" 設定に値を入力しなかった場合、サンプルの値が灰色テキストで表示されます。この値は例として表示されているため、最終的に Windows Installer データベースには含まれません。

概要情報ストリーム データを入力する

概要情報ストリームには、会社や、インストール中のソフトウェアに関する情報が含まれます。この情報は、.msi ファイルを右クリックし、[プロパティ] を選択してアクセスできます。その後、[概要] タブをクリックして、概要情報を確認します。

ビルド時の概要情報ストリームのコメントを設定する

ビルド時にコメントを入力できるオプションもあります。MSBuild.exe タスクの SummaryInfoComments パラメーターを使って、次の例のようにビルド番号を含めるなど、ビルド時に **[概要情報ストリーム コメント]** を設定することができます。

MSBuild.exe c:¥installers¥Setup.sln /Property:SummaryInfoComments="ここにコメントを入力"

デフォルトの製品インストール先フォルダー (INSTALLDIR) の設定

プロジェクトの INSTALLDIR プロパティは、製品に含まれるすべてのファイル用のデフォルト フォルダーとして機能します。その値の割り当て先は、Windows Installer フォルダーのプロパティである INSTALLDIR となり、これはデフォルトの機能のインストール先フォルダーの役割を果たします。

Windows ロゴ: *Windows ロゴ要件によると、ターゲット システムの言語にかかわらず、製品のファイルのデフォルトのインストール先を Program Files のサブフォルダー、またはエンド ユーザー のアプリケーション データ フォルダーにする必要があります。ProgramFilesFolder を製品のインストール先フォルダー設定の親フォルダーとして使用すると、ファイルは正しい場所にインストールされます。*

INSTALLDIR のデフォルト値は次の通りです:

[ProgramFilesFolder]*会社名¥製品名*

タスク ***製品の INSTALLDIR プロパティを設定するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. ビルトイン Windows Installer ディレクトリをパスの一部として使用するには、**INSTALLDIR** 設定で、省略記号ボタン (...) をクリックします。**[INSTALLDIR の設定]** ダイアログ ボックスが開きます。**[インストール先ディレクトリ]** ボックスで、インストール先のフォルダーを選択します。

   その代わりに、**INSTALLDIR** 設定に手動でパスを入力することもできます。

メモ: *[INSTALLDIR の設定] ダイアログ ボックスで新しいフォルダーを選択すると、INSTALLDIR 設定の値がオーバーライドされます。たとえば* **[ProgramFilesFolder]My Company¥Program** *のように、円記号でサブフォルダーを分割することによって、フォルダー プロパティのサブフォルダーを指定できます。*

インストール先フォルダーを指定するときは、閉じ角かっこ (]) または円記号 (¥) の前後にスペースを入れることはできません。たとえば、次のようなパスは有効ではありません。

[ProgramFilesFolder] ¥会社名¥default

[ProgramFilesFolder]YourCompany¥ default

INSTALLDIR などのインストール フォルダー プロパティを使用している場合、デフォルト値を指定していることになります。エンドユーザーは、Msiexec.exe 起動時に、コマンドラインでプロパティを設定したり、[カスタム セットアップ] ダイアログで機能に対して新しいインストール先フォルダーを選択することにより、この値を変更できます。

INSTALLDIR とレジストリ

インストールのデフォルト インストール先ディレクトリを、ターゲット システムのレジストリで見つかったキーの値に設定できます。このディレクトリは、[一般情報] ビューの INSTALLDIR プロパティに格納されています。

レジストリ エントリが INSTALLDIR の値になるように設定する場合、インストールを開始する前に、ターゲット システムにレジストリ エントリがなければならないという制約があります。レジストリ エントリが見つからない場合、INSTALLDIR に前回入力した有効値が設定されます。

たとえば、インストール プロジェクトを初回で作成したとき、INSTALLDIR の値は [ProgramFilesFolder]会社名¥Default に設定されます。これをレジストリから読み取るように変更しようとして、レジストリ エントリが見つからない場合、INSTALLDIR は以前の値である [ProgramFilesFolder]会社名¥Default に戻ります。

INSTALLDIR のレジストリからの設定

タスク ***レジストリから INSTALLDIR 値を設定するには、次の手順を実行します。***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. **INSTALLDIR** 設定で、使用する値を含むレジストリ キーへの完全パスを入力します。例:

   HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥App.exe¥

この値を **HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥App.exe** に設定すると、 は "App Paths" キー にある **App.exe** という文字列からこの値を取得します。この値に **HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥App Paths¥App.exe¥** (末尾に円記号を付加) を設定すると、インストールは **App.exe** と名づけられたキーのデフォルト値からこの値を取得します。

メモ: *行末の円記号 (¥) は、指定子の最後の部分がキーの名前であって値の名前ではないことを意味します。*

ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護

InstallShield では、ロックダウン環境において製品を実行するエンド ユーザー向けに、ファイル、フォルダー、およびレジストリ キーを保護するための異なる方法が提供されています:

- **従来型の Windows Installer 処理**—InstallShield は、.msi データベースの LockPermissions テーブルを使って、製品のアクセス許可情報を格納します。
- **カスタム InstallShield 処理**— InstallShield は、.msi データベースの ISLockPermissions テーブルを使って、製品のアクセス許可情報を格納します。InstallShield はまた、プロジェクトにカスタム アクションを追加します。

これらの方法を使って、特定グループおよびユーザーにファイル、フォルダー、またはレジストリ キーのアクセス許可を割り当てることができます。たとえば、管理者グループに特定のファイルについての [読み取り]、[書き込み]、および [削除] アクセス許可を割り当てることができますが、別のグループのすべてにユーザーについては [読み取り] 許可のみ割り当てることができます。

使用するオプションを決定する

次のテーブルでは、アクセス許可を設定するため様々な方法を比較します。

テーブル 4-1・ロックダウン環境下でオブジェクト (ファイル、フォルダー、およびレジストリ キー) をセキュリティ保護するための異なる方法

比較カテゴリ	利用できるサポートについて
広く知られているセキュリティ識別子 (SID)	・ **従来型の Windows Installer 処理** – 限られた数の SID (Administrators、Everyone) をサポートします。 ・ **カスタム InstallShield 処理**–多くの SID (Administrators、Authenticated Users、Creator Owner、Everyone、Guests、Interactive、Local Service、Local System、Network Service、Power Users、Remote Desktop Users、および Users) をサポートします。
翻訳された SID の名前	・ **従来型の Windows Installer 処理** – 翻訳された SID の名前をサポートしません。翻訳された名前を使用すると、インストールは失敗します。 ・ **カスタム InstallShield 処理**–サポートされている、よく知られる SID (Administrators、Authenticated Users、Creator Owner、Everyone、Guests、Interactive、Local Service、Local System、Network Service、Power Users、Remote Desktop Users、および Users) の翻訳されたすべての SID の名前をサポートします。
特定のアクセス許可を拒否できる機能	・ **従来型の Windows Installer 処理** – サポートされていません。この処理を使って、特定のアクセス許可を設定できます。アクセス許可を拒否することはできません。したがって、ユーザーにファイルの読み取り専用アクセスを付与することができます。ただし、ユーザーが読み取り専用アクセスを所有することを防ぐことはできません。 ・ **カスタム InstallShield 処理**–サポートされています。このオプションを使って、指定するアクセス許可を特定のユーザーまたはグループが所持することを拒否するかどうかを指定できます。
既存のアクセス許可に対する影響	・ **従来型の Windows Installer 処理** – 既存のアクセス許可を削除することができます。たとえば、ターゲット システム上のフォルダーに Everyone ユーザーのアクセス許可が既に設定されている場合で、インストールが Administrators ユーザーのアクセス許可を設定するとき、このオプションを使って Administrators ユーザー用のアクセス許可を設定できます。ただし、既存する Everyone のアクセス許可は削除されます。 ・ **カスタム InstallShield 処理**–カスタム オプションを使って、ターゲット システムに既存するファイル、フォルダー、またはレジストリ キーに、アクセス許可を追加できます。このとき、そのオブジェクトに既存するアクセス許可は削除されません。たとえば、ターゲット システム上のフォルダーに Everyone ユーザーのアクセス許可が既に設定されている場合で、インストールが Administrators ユーザーのアクセス許可を設定するとき、これらのオプションを使って既存する Everyone ユーザーのアクセス許可を削除することなく、Administrators ユーザー用のアクセス許可を設定できます。

テーブル 4-1・ロックダウン環境下でオブジェクト (ファイル、フォルダー、およびレジストリ キー) をセキュリティ保護するための異なる方法 (続き)

比較カテゴリ	利用できるサポートについて
子オブジェクト (サブフォルダー、ファイル、およびサブキー) にアクセス許可を反映させる機能	・ **従来型の Windows Installer 処理** – サポートされていません。フォルダー内のサブフォルダーまたはファイル (またはレジストリ キーの下にあるサブ キー) にアクセス許可を構成する場合、ターゲット システム上で作成された親は、自動的に子のアクセス許可を継承します。 ・ **カスタム InstallShield 処理**–サポートされています。このオプションを使って、フォルダー (またはレジストリ キー) のアクセス許可を構成し、そのフォルダーのサブフォルダーおよびファイル (またはレジストリ キーのサブキー) すべてに同じアクセス許可を適用するかどうかを指定できます。
インストール中に作成された新しいユーザーに対するアクセス許可を設定できる機能	・ **従来型の Windows Installer 処理** – サポートされていません。 ・ **カスタム InstallShield 処理**–サポートされています。インストール中に新しいユーザーが作成される場合、そのユーザーに対するアクセス許可を構成できます。

どちらのオプションでも、インストールの一部としてインストールされないオブジェクトに対してアクセス許可を設定することはできません。

[カスタム InstallShield 処理] オプションまたは [従来型の Windows Installer 処理] オプションについての詳細

Express プロジェクトでは、カスタム InstallShield 処理または Windows Installer 処理のどちらを使用するかを指定しなくてはなりません。詳しくは、「プロジェクトで、ロックダウン環境でのアクセス許可タイプを選択する」をご覧ください。

いずれかのオプションを使ってファイルまたはフォルダーのアクセス許可を設定する方法については、「ファイルとフォルダーのアクセス許可を構成する」を参照してください。これらのオプションの 1 つを使ってレジストリ キーのアクセス許可を設定する方法については、「レジストリ キーのアクセス許可を構成する」を参照してください。

プロジェクトで、ロックダウン環境でのアクセス許可タイプを選択する

InstallShield には、ロックダウン環境にあるエンド ユーザー向けにファイル、フォルダー、およびレジストリ キーのアクセス許可をインストールがどのように構成するかを指定できる、プロジェクト全体に反映される設定があります。

タスク ***プロジェクトに対して、ロックダウン環境でのアクセス許可タイプを選択するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. ″**ロックダウンの設定方法**″ 設定で、適切なオプションを選択します:
   - **カスタム InstallShield 処理**–InstallShield は、プロジェクトにカスタム テーブルとカスタム アクションを追加して、ターゲット システム上のアクセス許可を設定します。これがデフォルトの値です。

- **従来型の Windows Installer 処理** — InstallShield は、.msi データベースの LockPermissions テーブルを使って、製品のアクセス許可情報を格納します。

これらの 2 つのオプションについての詳細な比較情報は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

プロジェクトで次にファイル、フォルダー、またはレジストリ キーのアクセス許可を構成するとき、InstallShield は選択済みのロックダウン環境でのアクセス許可を使用します。

- [従来型の Windows Installer 処理] オプションを選択した場合、InstallShield はプロジェクトで LockPermissions テーブルを使用します。
- [カスタム InstallShield 処理] オプションを選択した場合、InstallShield はプロジェクトで ISLockPermissions テーブルを使用し、さらに ISLockPermissionsCost と ISLockPermissionsInstall カスタム アクションをプロジェクトに追加します。

"ロックダウンの設定方法" 設定の値を変更する時に、プロジェクトにファイル、フォルダー、またはレジストリ キーのアクセス許可が既に含まれている場合、InstallShield は適切なテーブルにアクセス許可データを移行するかどうかを問い合わせるメッセージ ボックスを表示します。データの移行を選択すると、選択済みのオプションに対応するテーブルにデータが移動されます。[カスタム InstallShield 処理] オプションから [従来型の Windows Installer 処理] オプションに切り替えると、プロジェクトから ISLockPermissionsCost と ISLockPermissionsInstall カスタム アクションも削除されます。

Windows Installer インストールをログ記録するかどうかを指定する

InstallShield では、Windows Installer 4.0 以降がインストールのログ記録を行うかどうかを、プロジェクト全体にわたって指定することができます。また、ログ記録されるメッセージの種類をカスタマイズすることもできます。

タスク ***Windows Installer 4.0 以降で、プロジェクト全体のログ情報を指定するには、次の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. **[MSI ログの作成]** 設定をクリックして、省略記号ボタン (...) ボタンをクリックします。**[Windows Installer 4.0 以降のログ記録オプション]** ダイアログ ボックスが開きます。
3. 適切なオプションを選択します。カスタム オプションを選択する場合、MsiLogging 値を入力します。
4. **[OK]** をクリックします。

結果は、[Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックスで行った選択によって異なります。

- [いいえ] を選択すると、ログ記録は行なわれません。これがデフォルトの値です。
- [はい] を選択すると、InstallShield は MsiLogging プロパティにデフォルト値である *voicewarmupx* を挿入します。Windows Installer 4.0 が搭載されたターゲット システム上でインストールが実行されると、次の処理が行なわれます。
  - *voicewarmupx* のデフォルト ログ モードに従って、インストーラーがログ ファイルを作成します。
  - インストーラーが、MsiLogFileLocation プロパティに、ログ ファイルのパスを挿入します。
  - SetupCompleteSuccess、SetupCompleteError および SetupInterrupted ダイアログに **[Windows Installer ログを表示]** チェック ボックスが追加されます。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。

- [カスタム] を選択すると、InstallShield は、このボックスに指定された値を MsiLogging プロパティに挿入します。Windows Installer 4.0 が搭載されたターゲット システム上でインストールが実行されると、次の処理が行なわれます。
  - このボックスに指定されたカスタム値に基づいて、インストーラーがログ ファイルを作成します。
  - インストーラーが、MsiLogFileLocation プロパティに、ログ ファイルのパスを挿入します。
  - SetupCompleteSuccess、SetupCompleteError および SetupInterrupted ダイアログに **[Windows Installer ログを表示]** チェック ボックスが追加されます。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。

以前のバージョンの Windows Installer は MsiLogging 設定を無視します。以前のバージョンの Windows Installer を実行中のシステム上で表示される実行時ダイアログでは、**[Windows Installer ログを表示]** チェック ボックスは表示されません。

重要: MsiLogFileLocation *プロパティは、読み取り専用のため、ログ ファイルの場所の設定および変更には使用できません。*

製品のソフトウェア識別タグを含める

プロジェクト: *この情報は Express プロジェクトに適用します。*

ISO/IEC 19770-2 は、ソフトウェア識別タグを作成するための国際規格です。ソフトウェア識別タグは、製品名、製品エディション、パブリッシャーなど、ソフトウェアに関する説明的な情報を含むサイズが小さい XML ベースのファイルです。ソフトウェア資産管理ツールは、企業でインストールされているソフトウェアについての正確なアプリケーション ID を提供する目的で、タグ内のデータを収集するツールです。

ソフトウェア識別タグ機能は、業界標準として現在進化しつつあるもので、この機能により、独立系ソフトウェア ベンダーは、顧客に対して、ソフトウェア資産管理およびライセンス最適化イニシアチブに有用な、より適切な情報を提供することができる、より洗練されたアプリケーションを作ることができるようになります。製品のインストール パッケージに識別タグを持たせることで、顧客は、インストールした製品の内部的使用状況を監視できるツールが使えるようになります。これにより、顧客が入手済み製品のライセンスの数を把握、管理および最適化することができるようになります。

適切にタグを作成するためには、[一般情報] ビューにある "製品名" や "製品バージョン" 設定といった基本設定を構成する必要があります。また、[一般情報] ビューにある識別タグ関連の設定も構成しなくてはなりません。

タスク ***インストールにソフトウェア識別タグを含めるには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. このビュー内の **[ソフトウェア識別タグ]** 領域で、必要に応じて設定の値を変更します。

   "ソフトウェア識別タグの使用" 設定を使って、インストールにタグを含めるかどうかを指定できます。デフォルト値の **[はい]** を選択してから、必要に応じて **[ソフトウェア識別タグ]** 領域にあるその他の設定を構成します。

ビルド時に以下の条件が True 評価された場合、InstallShield がビルドするインストールにソフトウェア識別タグが追加されます:

- [一般情報] ビューの "ソフトウェア識別タグの使用" 設定に、デフォルト値の [はい] が選択されている。
- [一般情報] ビューの "一意な ID""タグ作成者"、および "タグ作成者 ID" 設定に値が設定されている。

タグ機能が有効である場合に、前述の 3 つのタグ識別設定のうち 1 つでも値が不足している場合、InstallShield はビルド警告を生成して、リリースにタグが含まれなかったことを通知します。この警告を解決するためには、必要に応じて [一般情報] ビューの [ソフトウェア識別タグ] 領域にある設定を構成してください。

プロジェクトでタグ機能を使用するとき、InstallShield はタグをプロジェクトの [常にインストール] 機能に追加します。実行時、ファイルは 2 つの異なる場所にインストールされます:

- INSTALLDIR
- CommonAppDataFolder

プロジェクトにソフトウェア ID タグを含めて [リリース] ビューで .pfx ファイルを使ってリリースに署名を行うことを設定すると、InstallShield がビルド時にタグをデジタル署名します。タグ ファイルに署名するためには、.NET Framework 3.5 以降をビルドマシンにインストールする必要があります。

インストールのファイルを編成する

インストールの一番重要なタスクは、ファイルを配布メディアからエンドユーザーのハード ディスクへ転送することです。InstallShield インストールでは、ファイルは階層構造で編成されます。ファイルは機能 (およびオプションでサブ機能) に含まれ、セットアップの種類に関連付けられています。

実行時に、エンドユーザーは、単純にセットアップの種類、または、可能な場合、インストールする機能およびサブ機能を選択するだけです。

あるファイルが、他のファイルの関数に依存してタスクを実行することがよくあります。ただし、インストール プロジェクトにアプリケーションファイルを含める際に、依存関係と呼ばれる他のファイルに気が付かない場合があります。依存関係ファイルが識別しやすくなるように、InstallShield には自動的に依存関係をプロジェクトに追加する 3 種類の依存関係スキャナーがあります。

プロジェクトへの個別ファイルの追加に加え、再配布可能ファイル (InstallShield 前提条件、マージ モジュールおよびオブジェクト) も含めることができます。再配布可能ファイルには、特定の機能性をインストールするのに必要なロジックとファイルが含まれています。たとえば、Java Runtime Environment (JRE) ファイルをインストールに含める場合、JRE の InstallShield 前提条件をインストール プロジェクトに追加します。

インストールをデザインする

機能とは、ユーザーから見て個別にインストール可能な最小の製品構成単位のことです。インストール中に [カスタム] セットアップ タイプを選択すると、個々の機能がエンドユーザーに対して表示されます。

機能

機能は、エンドユーザーから見たアプリケーションの構成要素です。各機能は、たとえばヘルプファイルのような製品の特定の機能を表します。エンドユーザーは、製品の別個の機能をインストールまたはアンインストールできます。

たとえば、ハードディスクの容量に制限のあるエンドユーザーは、製品チュートリアルをインストールしないよう選択できます。そのユーザーは、後で別のコンピューターを購入するか、または既存のコンピューター上のリソースに空きを作り、前にアンインストールした製品チュートリアルをインストールすることもできます。

アプリケーションの機能に対応し、異なるセットアップの種類で使用または削除できるように、アプリケーションを機能に分割する必要があります。

セットアップの種類

[セットアップの種類] を利用すると、エンドユーザーは製品全体または一部分を選択してインストールすることができます。デフォルトのセットアップの種類は「標準」、「最小」および「カスタム」です。

セットアップの種類は機能に基づいています。各セットアップの種類に関連する機能を選択します。次に、エンドユーザーが特定のセットアップの種類を選択すると、そのセットアップの種類に関連付けられた機能だけがインストールされます。

アプリケーションを機能に分ける

アプリケーションを、ヘルプ ファイル、クリップ アート ファイル、プログラム ファイルなどの非依存な部分に分けます。これにより、エンドユーザーはアプリケーションをインストールする際オプションを組み合わせることができます。たとえば、アプリケーションにサイズが大きく画像の多いクリップ アート ファイルが含まれてい

る場合、クリップ アート ファイルを機能にすることができます。これにより、エンドユーザーはファイルをインストールするかしないかを選択することができます。これは、使用できるリソースが限られている場合に極めて重要な機能です。

アプリケーションを機能に分ける際、かならずエンドユーザーが特定のニーズを実現するため、分けた部分をいくつかの方法で再度組み合わせることができるようにします。これを行う際、システム管理者から顧客サービス担当者、開発者などすべてのユーザーのニーズについて考慮します。ユーザーのすべてのグループに対応することによって、アプリケーションの配布と使用の増加を促すことになります。

各機能には、ヘルプ ファイルのように 1 つの機能性があり、この機能性にしたがって明確に定義を行うことで認識および理解されやすくします。機能には、ユーザーが選択で自ら機能をインストールして使用できる独立した機能性が必要です。

ある機能が別の機能を必要とする場合、依存する機能をもう一方の子にします。

混乱を避けるため、システムやアプリケーションの管理に関するあらゆる情報をユーザーに透過にします。

機能の定義

機能は、エンド ユーザーから見たプリケーションの構成要素です。これは、ヘルプ ファイルや製品スイートの一部などの製品の特定機能を表し、エンド ユーザーがこれらをインストールまたはアンインストールするかどうかを決定できます。アプリケーション全体は特定の目的を実行する機能ごとに分ける必要があります。機能は、[機能] ビューで作成できます。

機能には必要な要素のすべてが含まれており、兄弟機能を必要としないのが原則です。たとえば、類語辞典機能はエンドユーザーがインストールをしない選択ができる辞書機能を必要としません。ただし、"親" 機能のサブ機能をインストールに含めることもできます。サブ機能によって、インストールするファイルやデータをエンドユーザーがさらに細かく選択できるようになります。

サブ機能は機能をさらに分割したものです。すべての機能とサブ機能が表示される場合、エンドユーザーは機能のどの部分をインストールするかを [カスタム セットアップ] ダイアログで選択できます。

ヒント: *多くのサブ機能を作成できますが、編成のためにデザインを可能な限り単純にしておくようお勧めします。*

機能の作成

[機能] ビューを使って、プロジェクトの機能およびサブ機能を作成できます。

タスク ***機能を作成するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストから **[機能]** をクリックします。
2. **[機能]** エクスプローラーを右クリックし、**[新しい機能]** をクリックします。InstallShield はデフォルト名 **NewFeature***n* (*n* は連続番号です) を持つ新しい機能を作成します。
3. 機能の名前を入力します。機能の名前を変更するには、機能を右クリックして **[名前の変更]** をクリックします。
4. 右側のペインで機能の設定を構成します。

サブ機能を作成する

タスク *サブ機能を作成するには、以下の手順に従います:*

1. [**セットアップの編成**] の下にあるビュー リストから [**機能**] をクリックします。
2. [**機能**] エクスプローラーでサブ機能を含める親機能を右クリックして、[**新しい機能**] をクリックします。InstallShield はデフォルト名 **NewFeature***n* (*n* は連続番号です) を持つ新しい機能を作成します。
3. サブ機能の名前を入力します。サブ機能の名前を変更するには、それを右クリックして [**名前の変更**] をクリックします。
4. 右側のペインで機能の設定を構成します。

ヒント: *新しい機能を追加し、機能の名前に K***Feature 1¥Feature 2¥Feature 3** *と入力することで、ネストされた複数の機能を一度に作成することができます。InstallShield は、***Feature 3** *が* **Feature 2** *のサブ機能であり、Feature 2 が* **Feature 1** *のサブ機能である、ネストされた機能構造を作成します。*

機能の設定を構成する

タスク *機能の設定を構成するには、以下の手順に従います:*

1. [**セットアップの編成**] の下にあるビュー リストから [**機能**] をクリックします。
2. 構成する機能を選択します。
3. 右側ペインに表示されているグリッド内の設定を構成します。

機能の条件を設定する

インストールの機能に特定のオペレーティング システムまたはアプリケーションが必要であると指定する必要がある場合、一案として機能の 1 つを条件付きでインストールすることができます。ターゲット システムが要件を満たさない場合、機能はインストールされません。

メモ: *[常にインストール] 機能の条件は作成できません。*

タスク *プロジェクトの機能に条件を設定するには、以下の手順に従います:*

1. [**セットアップの編成**] の下にあるビュー リストから [**機能**] をクリックします。
2. 構成する機能を選択します。
3. **"条件"** 設定をクリックして、省略記号ボタン (...) をクリックします。[**条件ビルダー**] ダイアログ ボックスが開きます。
4. オペレーティング システム条件を作成するには、以下の手順に従います:

a. **[オペレーティング システム]** タブをクリックします

b. **[すべてのオペレーティング システム]** チェック ボックスをクリアして、条件でターゲットするオペレーティング システムを選択します。

ソフトウェア条件を作成するには、以下の手順に従います:

a. **[ソフトウェア]** タブをクリックします。

b. 以下のいずれかを実行します (複数可)。

- 選択された機能に対して、アプリケーションがターゲット システムに存在している必要がある場合、そのアプリケーションをチェック ボックスを、チェック済み状態 (緑色) に設定します。チェック ボックスをこの状態にするには、このチェック ボックスに緑色のチェックマークが表れるまで繰り返しクリックします。
- 選択された機能に対して、アプリケーションがターゲット システムにインストールされていてはいけない場合、そのアプリケーションをチェック ボックスを、赤い X 印に設定します。チェック ボックスをこの状態にするには、このチェック ボックスに赤の X が表われるまで繰り返しクリックします。
- ソフトウェア条件を削除にするには、チェック ボックスが空になるまで繰り返しクリックします。空のチェック ボックスは、条件がソフトウェアにまったく影響されないことを示します。
- アプリケーションがリストに無い場合、**[新規要件の作成]** をクリックします。**システム検索 ウィザード**が開き、ここで指定されたファイルがターゲット システムに存在するかどうかに基づいて新しい条件を作成します。新しい条件が作成され、**[システム ソフトウェア要件]** エクスプローラーに追加されます。

5. [OK] をクリックします。

条件がグリッドに追加されます。機能は、そのオペレーティング システム条件が 1つでも True で、且つ他の条件もすべて True のときインストールされます。

機能をエンドユーザーへ表示する

InstallShield では、その機能を [カスタム セットアップ] ダイアログでエンドユーザーにどう表示するかを指定します。

[機能] ビューにある機能の ″表示″ 設定を使って、機能を表示するかどうか、およびどのように表示するかを指定できます。選択可能なオプションは以下のとおりです:

テーブル 4-1 ・ ″表示″ 設定で使用できるオプション

オプション	説明
閉じて表示する	デフォルトでサブ機能が閉じた状態で、[カスタム セットアップ] ダイアログに機能を表示します。
展開して表示する	デフォルトでサブ機能が展開した状態で、[カスタム セットアップ] ダイアログに機能を表示します。
非表示	機能がエンドユーザーに対して [カスタム セットアップ] ダイアログに表示されます。

メモ: この設定で [表示しない] を選択しても、機能がインストールされるかどうかには影響しません。非表示にした機能は、自動的にすべてインストールされるというわけではなく、インストールが必要な機能の場合は選択解除できず、インストールするべきではない機能の場合は選択できないようになります。

機能のインストールを必須にする

機能の "必須" 設定を [はい] にすると、エンドユーザーが [カスタム セットアップ] ダイアログで選択を解除できなくなります。その機能はターゲット システムにインストールされます。

"必須" 設定が [いいえ] に設定されていると、機能はデフォルトでインストールされますが、エンドユーザーは選択を解除できます。

機能のアドバタイズ

InstallShield では、機能のアドバタイズを選択的に有効または無効にすることができます。アドバタイズされた機能は、インストール過程ですぐにはインストールされません。要求されたときにインストールされます。機能を割り当てると、その機能はすでにインストールされているように見えますが、エンドユーザーが要求するまではインストールされません。(機能を割り当てるとショートカットがインストールされ、コントロール パネルの [プログラムの追加と削除] アプレットからインストールできます。ただし割り当てられた機能は、ユーザーが要求するまではアドバタイズされるだけです。)パブリッシュされた機能は、インストーラーから要求されるまでターゲット システムに表示されません。(パブリッシュされた機能には、エンド ユーザー インターフェイス要素がありません。これらのインストールは、プログラム上で行うか、MIMEの種類を関連付けて行います。)

[機能] ビューの "アドバタイズ" 設定を使って、アドバタイズを許可するかどうかを指定します。この設定で選択できるオプションは、次のとおりです:

テーブル 4-2 · "アドバタイズ" 設定で使用できるオプション

オプション	説明
アドバタイズを許可する	エンド ユーザーは、CustomSetp ダイアログで、この機能のアドバタイズ オプションを選択できます。アドバタイズは許可されますが、インストール実行時のデフォルトのオプションではありません。
アドバタイズを優先する	機能は、デフォルトでアドバタイズされます。エンド ユーザーは、CustomSetup ダイアログの機能のアドバタイズ オプションを変更できます。
アドバタイズを許可しない	この機能には、アドバタイズは許可されません。エンドユーザーは、CustomSetup ダイアログで機能のアドバタイズを選択できません。
サポートされていない場合にアドバタイズを無効にする	アドバタイズメントは、Internet Explorer 4.01 以上のシステムでのみ機能します。この条件を満たさないターゲット システムでは、アドバタイズは許可されません。ターゲット システムがアドバタイズをサポートする場合、アドバタイズは許可されます。

機能のアドバタイズを有効にすると、アドバタイズを阻止するその他の要素がない限り、インストールが実行されているモードにかかわらず機能はアドバタイズされます。Custom Setup ダイアログで、エンドユーザーはすぐにインストールする機能とあとで使用する機能を制御できます。

アドバタイズする場合には、通常、アプリケーションでこの機能をサポートしている必要があります。たとえば、ある製品でスペルチェック機能をアドバタイズするとします。アプリケーションインターフェイスは、メニューコマンドまたはツールバーボタンから、スペルチェックを提供します。さらに、機能のインストール状態を確認し、カスタマーが [スペルチェック] コマンドまたはボタンをクリックするとそれがインストールされるようにアプリケーションに書き込む必要があります。

機能の ″リモート インストール″ 設定を設定する

機能の ″リモート インストール″ 設定によって、この機能のファイルをターゲット システムにインストールするか、または CD-ROM やネットワークサーバーのようなソース メディアから実行するかが決定されます。新しい機能のデフォルト値は ″ローカルを優先″ です。これは、選択された機能のファイルがターゲット システムにインストールされることを意味します。

タスク ***″リモート インストール″ 設定を変更して、機能のファイルがソース メディアからのみ実行されるようにするには、以下の手順に従います:***

1. **[編成]** の下のビュー リストにある **[機能]** をクリックします。
2. 構成する機能を選択します。
3. **″リモート インストール″** 設定で、**[ソースを優先]**を選択します。

ヒント: *[**親を優先**] を選択すると、サブ機能にその親機能と同じ値が指定されます。*

[カスタム セットアップ] ダイアログで機能の順序を変える

[カスタム セットアップ] ダイアログで表示される機能は、[機能] ビューの機能の一覧と同じ順番で表示されます。

タスク ***[カスタム セットアップ] ダイアログで機能の順序を変えるには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストから **[機能]** をクリックします。
2. 移動する機能を右クリックして、**[上に移動]** または **[下に移動]** をクリックします。機能を左右に移動して、別の機能のサブ機能にすることもできます。

ヒント: *単純なドラッグアンドドロップ操作で、機能の順序を変更をすることもできます。機能またはサブ機能は、すべてこの方法で移動できます。*

セットアップの種類について

[セットアップの種類] を利用すると、エンドユーザーは製品全体または一部分を選択してインストールすることができます。デフォルトのセットアップの種類は「標準」、「最小」および「カスタム」です。

[標準] セットアップ タイプは通常、アプリケーションの大部分の機能を含んでいます。[カスタム] セットアップ タイプは、インストールする機能をエンドユーザーが選択することができます。[最小] セットアップ タイプは通常、アプリケーションを実行するために必要な機能だけを含みます。この種類のセットアップはディスクの容量に制限があるノートブック コンピューターなどのエンドユーザーのためにデザインされています。

セットアップの種類は機能に基づいています。各セットアップの種類に関連する機能を選択します。次に、エンドユーザーが特定のセットアップの種類を選択すると、そのセットアップの種類に関連付けられた機能だけがインストールされます。

[セットアップの種類] で機能を指定する

タスク ***各セットアップの種類に含める機能を指定するには、次の手順を実行します:***

1. **[セットアップの種類]** ビューを開きます。
2. **[セットアップの種類]** エクスプローラーで、編集するセットアップの種類をクリックします。インストールに含まれるすべての機能は下のペインに表示されます。
3. 選択したセットアップの種類に含めない機能のチェック ボックスをクリアします。

セットアップの種類の名前を変更する

[セットアップの種類] ビューで名前を変更するのは、[テキストとメッセージ] ビューで [セットアップの種類] ダイアログのリソースを更新するのと同じです。

タスク ***セットアップの種類の名前を変更するには、以下の手順に従います:***

1. **[セットアップの種類]** ビューを開きます。
2. **[セットアップの種類]** エクスプローラーで、編集するセットアップの種類をクリックし、**[名前の変更]** をクリックします。

   また、セットアップの種類を右クリックして、F2 を押すこともできます。
3. セットアップの種類に新しい名前を付けます。この名前は、インストール中に [セットアップの種類] ダイアログで表示されます。

メモ: *セットアップの種類にアクセラレータ キーを提供する場合、名前の文字の前にアンパサンド (&) を入力してください。たとえば、***カスタム(&S)** *という名前は、S に下線が付いたカスタム(S) というラベルになります。エンドユーザーがセットアップ時に S キーを押すことでそのオプション ボタンを選択することができるようになります。アンパサンド (&) 記号をアクセラレータ キーとして設定するには、***&&** *のように入力します。*

セットアップの種類を指定する

タスク ***セットアップに含めるセットアップの種類を指定するには、以下の手順に従います:***

1. **[セットアップの種類]** ビューを開きます。
2. **[セットアップの種類]** エクスプローラーで、インストールに含めないセットアップの種類のチェック ボックスをクリアします。
3. オプションで、**"説明"** プロパティ フィールドに説明を入力することもできます。複数行にわたる説明を入力するには、エスケープ シーケンス ¥n を使います。たとえば、**InstallShield¥nExpress Edition** と入力した場合、**[セットアップの種類]** ダイアログに表示される説明は次のようになります:

   InstallShield
   Express Edition

インストールにセットアップの種類を 1 つだけ含める

すべてのエンドユーザーに同じファイルをインストールしてもらう必要がある場合、[標準] セットアップ タイプのみをプロジェクトに含めます。インストールが実行されるとき、[セットアップの種類] ダイアログ は表示されず、エンドユーザーはインストールする機能を選択することはできません。すべての機能がインストールされます。

タスク ***インストールにセットアップの種類を 1 つだけ含めるには、以下の手順に従います:***

1. **[セットアップの種類]** ビューを開きます。
2. **[セットアップの種類]** エクスプローラーで、**[最小]** と **[カスタム]** チェック ボックスをクリアします。

[セットアップの種類] ダイアログはインストールに含まれません。このため、エンドユーザーは製品を部分的に選択してインストールすることができません。

実行時にセットアップの種類にアクセスする

_IsSetupTypeMin プロパティは、エンドユーザーが [セットアップの種類] ダイアログで選択したセットアップの種類を格納します。

タスク ***インストールの実行時に選択したセットアップの種類にアクセスするには、以下のいずれかを実行します。***

- 次の VBScript コードを使用します:

```
' 選択されたセットアップの種類の値を取得
Dim sSetupType
sSetupType= Session.Property("_IsSetupTypeMin")

' 表示する。
MsgBox sSetupType
```

_IsSetupTypeMin プロパティには、デフォルトのセットアップの種類名 [標準]、[最小] および [カスタム] のみ含まれています。[セットアップの種類] ビューで [最小] を MySetupType に名前変更して、エンドユーザーが名前変更されたセットアップ タイプを選択した場合も、_IsSetupTypeMinには MySetupType ではなく [最小] が入ります。

ファイルとフォルダーを含める

ファイルは、製品の核であるとともにインストールの核でもあります。新しいファイルをプロジェクトに追加するとき、関連付ける機能を選択する必要があります。機能は、エンドユーザーがインストールを実行し、[カスタム] セットアップ タイプを選択したときに表示されるものです。機能がインストールに選択された場合、機能のファイルがターゲット システムにインストールされます。

[ファイル] ビューでプロジェクトにファイルを追加することができます。

ファイルとフォルダーをプロジェクトに追加する

[ファイル] ビューで [ファイル] エクスプローラーを使って、プロジェクトにフォルダーとファイルを追加します。

タスク ***[ファイル] エクスプローラーを使ってファイルとフォルダーを追加するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[機能]** リストで、追加するファイルまたはフォルダ—を含める機能を選択します。
3. **[インストール先コンピューターのフォルダー]** ペインで、フォルダーまたはファイルを追加するフォルダーを選択します。

   INSTALLDIR は、デフォルトでアプリケーションに含まれるファイルのルート ディレクトリであるため、最も一般的に使用されるターゲットの場所です。

   フォルダーまたはファイルを追加する定義済みフォルダーがこのペインに表示されていない場合、それを追加することができます。詳しくは、「[ファイル] ビューで定義済みフォルダーを表示する」をご覧ください。

ヒント: *64 ビット ターゲット システム上の 64 ビット ファイルの場所 (たとえば、Program Files Folder (x86) ではなく Program Files フォルダー) にインストールされる 64 ビット インストールを作成する場合 、64 ビット フォルダー (ProgramFiles64Folder、CommonFiles64Folder、または System64Folder)、または [ファイル] ビューのサブフォルダ—の 1 つ を選択します。*

64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

4. オプションで、インストールがターゲット システム上に作成するカスタム フォルダーを追加するには、新しいフォルダーを含めるフォルダーを右クリックしてから、**[新しいフォルダー]** をクリックします。
5. **[ソース コンピューターのフォルダー]** ペインで、追加するファイルが含まれるフォルダーに移動します。
6. **[ソース コンピューターのファイル] ペイン**から**[インストール先コンピューターのファイル] ペイン**に、そのファイルをドラッグします。

ヒント: フォルダー全体をインストール先フォルダーに追加する場合、[ソース コンピューターのフォルダー] ペインから [インストール先コンピューターのフォルダー] に、そのフォルダーをドラッグします。

ソースパス変数

定義済みフォルダーからファイルを追加するとき、通常 [インストール先コンピューターのファイル] ペインの [リンク先] 列には、ほとんどのファイルに対して絶対パスではなくパス変数が表示されます。これによってインストール プロジェクトの移植性が高まります。プロジェクトを別の開発マシンに移動する場合、パス変数によってファイルパスのアップデートが不要になります。

メモ: 以前のバージョンの InstallShield を使って作成したプロジェクトへ追加済みのファイルは、[リンク先] 列にある絶対パスを表示します。パス変数を表示するには、ファイルを削除して再びプロジェクトへ追加する必要があります。

サポートされているデフォルトのパス変数は次の通りです。

テーブル 4-3 · サポートされているデフォルト パス変数

変数	デフォルト パス
<CommonFilesFolder>	C:¥Program Files¥Common Files
<ISProductFolder>	C:¥Program Files¥InstallShield¥2021
<ISProjectDataFolder>	C:¥InstallShield 2021 Projects¥*プロジェクト名* *プロジェクト名* は、プロジェクト名からファイル拡張子を取ったものです。
<ISProjectFolder>	C:¥InstallShield 2021 Projects
<ISRedistPlatformDependentExpressFolder>	C:¥Program Files¥InstallShield¥2021¥Redist¥Language Independent¥i386 Express
<ISRedistPlatformDependentFolder>	C:¥Program Files¥InstallShield¥2021¥Redist¥Language Independent¥i386
<ProgramFilesFolder>	C:¥Program Files
<SystemFolder>	C:¥WINDOWS¥system32
<VSSolutionFolder>	状況によって異なる。この変数は、ハイレベルの基本ディレクトリを参照します。このサポートによって、InstallShield プロジェクトで Visual Studio ソリューション フォルダー内にある姉妹プロジェクトのファイルへのスタティック リンクを含むことができます。詳細については、「Visual Studio ソリューションで VSSolutionFolder パス 変数を使用する」を参照してください。
<WindowsFolder>	C:¥WINDOWS

コンテキスト メニューを使ってファイルをドラッグ アンド ドロップする

[ファイル] ビューを使うと、ソース コンピューターからターゲット システムのインストール先までフォルダーをドラッグ アンド ドロップすることができます。**[ソースコンピューターのフォルダー]** ペインから **[インストール先コンピューターのフォルダー]** ペインにファイルをドラッグする場合、多くのオプションがあります。

ヒント: *64 ビット システム上で InstallShield を使用している場合に、ソースの場所が開発マシン上の 64 ビット システム フォルダー (System32) であるシステム ファイルをプロジェクトに追加するとき、[ファイル] ビューの上部にあるソース コンピューターのペインからインストール先コンピューターのペインの適切な場所にドラッグすることはできません。詳細については、「64 ビット ソース マシンの 64 ビット System32 フォルダーからファイルを追加する」を参照してください。*

タスク ***コンテキスト メニュー コマンドを表示するには、以下の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[ソース コンピューターのフォルダー]** ペイン、または、**[ソース コンピューターのファイル]** ペインで、フォルダーまたはファイルを右クリックして、**[インストール先コンピューターのフォルダー]** ペイン、または、**[インストール先コンピューターのファイル]** ペインへドラッグします。そしてマウス ボタンを離します。

   フォルダーまたはファイルを追加する定義済みフォルダーが **[インストール先コンピューターのフォルダー]** ペインに表示されていない場合、それを追加することができます。詳しくは、「[ファイル] ビューで定義済みフォルダーを表示する」をご覧ください。

InstallShield は、いくつかのコマンドを含むコンテキスト メニューを表示します。

テーブル 4-4 • コンテキスト メニューから使用できるコマンド

オプション	説明
追加	選択されたフォルダー、サブフォルダーおよび/またはファイルを追加します。これはデフォルトのドラッグ アンド ドロップの動作と同じです。
ソース構造を保持したまま追加	ソース コンピューターのファイル/フォルダー構造を保持しながら、選択されたフォルダー、サブフォルダーおよび/またはファイルを追加します。 このコマンドは、ソース フォルダーが定義済みのインストール先フォルダーと一致する場合にのみ使用できます。さらに、ファイル/フォルダーをドロップするインストール先は、**[インストール先コンピューターのフォルダー]** ペインのインストール先コンピューターである必要があります。
フォルダーのみを追加	選択されたフォルダーと、選択されたフォルダーに含まれるすべてのサブフォルダーのみを追加します。このオプションは、選択されたフォルダーまたはサブフォルダーに含まれているファイルは追加されません。

テーブル 4-4 ・ コンテキスト メニューから使用できるコマンド (続き)

オプション	説明
ソース構造を保持したままフォルダーのみを追加	選択されたフォルダーと、選択されたフォルダーに含まれるすべてのサブフォルダーのみを追加します。このオプションは、選択されたフォルダーまたはサブフォルダーに含まれているファイルは追加されません。このオプションは、また、ソース コンピューターで見つかったフォルダー構造を保持します。 このコマンドは、ソース フォルダーが定義済みのインストール先フォルダーと一致する場合にのみ使用できます。さらに、ファイル/フォルダーをドロップするインストール先は、[**インストール先コンピューターのフォルダー**] ペインのインストール先コンピューターである必要があります。
Cancel	変更を加えずにドラッグアンドドロップ操作を終了します。

ヒント・ *64 ビット ターゲット システム上の 64 ビットの場所 (たとえば、Program Files Folder (x86) ではなく Program Files フォルダー) にインストールされる 64 ビット インストールを作成する場合 、64 ビット フォルダー (ProgramFiles64Folder、CommonFiles64Folder、または System64Folder)、または [**インストール先コンピューターのフォルダー**] ペインのサブフォルダーの 1 つ を使用します。*

64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

64 ビット ソース マシンの 64 ビット System32 フォルダーからファイルを追加する

64 ビット システムでは、System32 フォルダーは、64 ビット アプリケーションに予約されています。InstallShield の [ファイル] ビューで、開発マシンの 64 ビット システム フォルダーを表示しようとした時、32 ビット バージョンのフォルダーである SysWOW64 フォルダーが代わりに表示されます。このため、64 ビット システム上で InstallShield を使用している場合に、ソースの場所が開発マシン上の 64 ビット システム フォルダー (System32) であるシステム ファイルをプロジェクトに追加するとき、[ファイル] ビューの上部にあるソース コンピューターのペインからインストール先コンピューターのペインの適切な場所にドラッグすることはできません。

リダイレクトを回避して、開発マシン上の 64 ビット System32 ファイルを InstallShield プロジェクトに追加するには、マシン上の Sysnative フォルダーを参照してから、プロジェクトに適切なファイルを選択します。以下は、その手順の説明です。

注意・ *システム フォルダーは Windows によって保護されているため、通常、インストールにシステム ファイルを含めることは推奨されません。これらの場合、システム ファイルの配置/更新には、使用可能な場合、マイクロソフトの再配布可能ファイルを使用すること、または、Windows アップデートを使ってエンドユーザーに更新プログラムを取得してもらう方法を優先してください。*

タスク ***64 ビット Windows が搭載された開発システム上の 64 ビット System32 フォルダーのファイルをプロジェクトに追加するには、以下の手順に従います:***

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**インストール先コンピューターのフォルダー**] ペインで、ファイルを配置するフォルダーをクリックします。
3. [**インストール先コンピューターのフォルダー**] ペインを右クリックしてから [**ファイルの追加**] をクリックします。[**開く**] ダイアログ ボックスが開きます。
4. 次のパスを指定します (ドライブ文字は適切なドライブ文字で適宜置き換えてください):

   C:¥Windows¥Sysnative

5. プロジェクトに追加したい適切なファイルを選択してから、[**開く**] ボタンをクリックします。

ファイルがプロジェクトへ追加されます。Sysnative フォルダーが、追加したソース ファイルのパスの一部として使用されるようになります。WOW64 は Sysnative フォルダーを特殊なエイリアスとして認識するため、ファイル システムはこのフォルダーからアクセスのリダイレクトを行いません。

Windows ベースのシステムにおける Sysnative フォルダー サポートについて

Sysnative フォルダーの使用は、32 ビット マシンでサポートされていません。64 ビット システム上の InstallShield プロジェクトで Sysnative フォルダーを使用していて、32 ビット システムで、その InstallShield プロジェクトのリリースをビルドしようとした時、ソース ファイルが見つからなかったことを通知する 1 つ以上のビルド エラーまたは警告が生成されます。

ハードコード化されたインストール先ディレクトリを指定する

タスク ***特定のドライブをハードコード化されたインストール先として指定するには、次の手順を実行します:***

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**インストール先コンピューターのフォルダー**] ペインで、フォルダーを右クリックし、[**インストール先コンピューター**] を右クリックし、[**追加**] をクリックします。[**インストール先コンピューター**] エクスプローラーに新しいフォルダーが追加され、フォルダー名が編集できるように選択されます。
3. ドライブ文字を入力し、続けてコロンを入力します (例、C:)。
4. ENTER を押します。

タスク ***ドライブ文字フォルダーの下のフォルダーおよびサブフォルダーを指定してハードコード化されたインストール先パスを作成するには、以下の手順に従います:***

1. フォルダーを追加するドライブ フォルダー (たとえば C:) またはサブフォルダーを追加するフォルダーを右クリックし、[**追加**] をクリックします。新しいフォルダーが追加され、フォルダー名が編集できるように選択されます。
2. フォルダー名を入力します。

3. ENTER を押します。

ターゲット システムからファイルとフォルダーを削除する

InstallShield では、実行時にターゲット システムから削除するファイルとフォルダーを簡単に指定することができるビルトイン サポートが提供されています。このファイルとフォルダーの削除機能は、アプリケーションによって作成されるファイルの削除など、インストールが追跡を行わない処理に使用すると便利です。

ファイルまたはフォルダーの削除は、次のイベントの 1 つにスケジュールできます:

- ファイルまたはフォルダーの機能がインストールされるとき
- ファイルまたはフォルダーの機能がアンインストールされるとき
- ファイルまたはフォルダーの機能がインストールまたはアンインストールされるとき

削除されるアイテムがフォルダーの場合、そのフォルダーが空の場合のみ削除されます。

タスク ***ファイルとフォルダーの削除を構成するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[インストール先コンピューターのフォルダー]** ペインで、削除するファイルまたはフォルダーを含むフォルダーを選択します。
3. **[インストール先コンピューターのファイル]** ペインを右クリックしてから **[ファイルの削除を追加]** をクリックします。**[プロパティ]** ダイアログ ボックスが開きます。
4. 必要に応じて設定を指定します。詳細については、「ファイル 削除の [プロパティ] ダイアログ ボックス」を参照してください。

InstallShield によって、**[インストール先コンピューターのファイル]** ペインにファイルまたはフォルダー アイコンが追加されます。アイコンに赤い X 印がついて、削除される項目を参照していることを示します。

プロジェクトでファイルとフォルダーを管理するときのヒント

ドラッグ アンド ドロップ処理、CTRL+C や CTRL+P といった一般的なキーボード ショートカット、およびアイテムを右クリックすると表示されるコンテキスト メニューを使って、ファイルとフォルダーを異なるインストール先フォルダーや機能へ簡単に移動させることができます。

プロジェクト内のファイルとフォルダーのインストール先を管理する

タスク ***プロジェクト内のファイルとフォルダーのインストール先を管理するには、次の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. 以下のいずれかを実行します:
   - ファイルまたはフォルダーの設定を変更するには、アイテムを右クリックしてから、**[プロパティ]** をクリックします。**[プロパティ]** ダイアログ ボックス が開き、ここで必要に応じて設定を編集できます。

- あるインストール先フォルダーから別のインストール先フォルダーにファイルを移動させるには、[**インストール先コンピューターのファイル**] ペインの 1 つの場所から [**インストール先コンピューターのフォルダー**] ペイン内の適切なフォルダーにドラッグします。
- あるインストール先フォルダーから別のインストール先フォルダーにフォルダーを移動させるには、[**インストール先コンピューターのフォルダー**] ペインの 1 つの場所から同じペイン内の適切なフォルダーにドラッグします。
- 1 つの場所から別の場所にファイルまたはフォルダーをコピーするには、CTRL を押しながら、アイテムを 1 つの場所から別の場所へドラッグします。

  別の方法として、アイテムをクリップボードにコピーしてから、それを適切な場所に貼り付けることもできます。アイテムを右クリックしてから [**コピー**] をクリックするか、それをクリックして CTRL+C を押します。次にファイルを含める別のフォルダーを選択します。そのフォルダーを右クリックして [**貼り付け**] をクリックするか、フォルダーをクリックしてから CTRL+P を押します。
- プロジェクトからファイルまたはフォルダーを削除するには、それを右クリックしてから [**削除**] をクリックします。

プロジェクトに含まれるファイルと機能の関連付けを管理する

タスク ***プロジェクトに含まれるファイルと機能の関連付けを管理するには、次の手順に従います:***

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイルと機能**] をクリックします。
2. 機能の関連付けを管理するファイルを含む機能を選択します。右側の [**ファイル**] ペインにファイルが表示されます。
3. 以下のいずれかを実行します:
   - ファイルを 1 つの機能から別の機能にコピーするには、[**ファイル**] ペインからその機能をドラッグして適切な機能にドロップします。
   - ファイルを 1 つの場所から別の場所にコピーするには、CTRL を押しながら、ファイルを 1 つの機能から別の機能にドラッグします。

     別の方法として、ファイルをクリップボードにコピーしてから、それを適切な機能に貼り付けることもできます。ファイルを右クリックしてから [**コピー**] をクリックするか、それをクリックして CTRL+C を押します。次にファイルを含める別の機能を選択します。その [ファイル] ペインを右クリックして [**貼り付け**] をクリックするか、フォルダーをクリックしてから CTRL+P を押します。
   - ファイルを機能から削除するには、そのファイルを右クリックしてから [**削除**] をクリックします。

個別のファイルに対してターゲット システム要件を指定する

[ファイル] ビューでは、各ファイルごとにターゲット オペレーティング システム要件を指定することができます。

タスク ***ターゲットオペレーティング システム要件を指定するには、以下の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[インストール先コンピューターのファイル]** ペインで、ファイルを右クリックして **[プロパティ]** を選択します。**[プロパティ]** ダイアログ ボックスが開きます。
3. **[詳細]** タブをクリックします。
4. **[ターゲット オペレーティング システム]** 領域で、ファイルの対象となるオペレーティング システムを指定します。特定のオペレーティング システムを選択するには、**[すべてのオペレーティング システム]**チェック ボックスをクリアする必要があります。

ダイナミック ファイル リンク

ディレクトリのコンテンツをすべてプロジェクトに追加する場合、ダイナミック ファイル リンクを使用できます。ダイナミック リンクにソース フォルダーを選択すると、ビルド時にそのフォルダー内のファイルがリリースに追加されます。ソース フォルダーは常にビルドの前にスキャンされ、すべての新規または変更ファイルが自動的にリリースへ組み込まれます。ダイナミック ファイル リンクは、フォルダー内のファイル一覧 (および該当する場合、サブフォルダー内のファイル一覧) がビルドとビルドの間で変わる可能性があるとき便利です。

重要: *ダイナミック ファイル リンクの使用は注意が必要です。ダイナミック リンクが参照するソース ファイルからダイナミック リンクがあるファイルを誤って削除してしまった場合、そのファイルは、次回リリースをビルドしたとき、リリースに含まれません。このとき、ビルドの警告やエラーも表示されません。製品は問題なくインストールされる場合がありますが、誤って削除されたダイナミック リンクがあるファイルがインストールされないため、適切に動作しない可能性があります。このため、重要な実行可能ファイル (例、.exe,、dll、.ocx ファイル) にダイナミック ファイル リンクを使用しないことが推奨されます (特に、製品でこれらのファイルが正常に実行される必要がある場合)。*

*ダイナミック リンクを作成する*とき、*可能なかぎり、ディレクトリごとメソッドではなく、ベスト プラクティス メソッドを使用することをお勧めします。ただし、どちらのメソッドでも、ターゲット イメージにあるファイルがパッチのダイナミック リンクから削除されると、パッチが正しくインストールされない場合があります。*

ダイナミック リンクがあるファイルをフィルターする

ダイナミック ファイル リンクを構成するとき、ダイナミック リンクがあるフォルダーのサブフォルダーを含めるかどうかを指定することができます。ダイナミック リンクがあるファイルをさらに細かくフィルターする場合、ダイナミック リンクに含める、または除外するファイルの特定の名前を指定することができます。また、ワイルド カードを使用して、追加または除外する特定のファイルやファイルの種類のみを指定することができます。

たとえば、すべての画像ファイルがサウンド ファイルと共に 1 つのフォルダーの中にあるとき、画像ファイルのみダイナミック リンクを付加する場合、ダイナミック リンクがあるフォルダーに .bmp ファイルと .ico ファイルのみを含めるように指定することができます。これを行うには、以下の例のように、選択パターンにアスタリスク (*) を使用します:

.bmp、.ico

特定のファイルを選択または除外する場合、選択また除外のパターン ボックスに完全なファイル名を入力します。詳細については、「ファイルを動的に追加する」を参照してください。

InstallShield インターフェイスで、ダイナミック リンクがあるファイル/フォルダーを静的ファイル/フォルダーから識別する

InstallShield インターフェイスでダイナミック ファイルが表示されるとき、ファイルのアイコンに左隅に、ファイルにダイナミック リンクがあることを示すイメージが表示されます:

同じダイナミック ファイルのイメージが、ダイナミック ファイル リンクも含まれているサブフォルダーのアイコンにも含まれています。

InstallShield インターフェイスが静的ファイルとフォルダーに表示するアイコンは、このダイナミック リンクのイメージを含みません。

ダイナミック ファイル リンクの制限事項

重要: *ダイナミック ファイル リンクの使用は注意が必要です。ダイナミック リンクが参照するソース ファイルからダイナミック リンクがあるファイルを誤って削除してしまった場合、そのファイルは、次回リリースをビルドしたとき、リリースに含まれません。このとき、ビルドの警告やエラーも表示されません。製品は問題なくインストールされる場合がありますが、誤って削除されたダイナミック リンクがあるファイルがインストールされないため、適切に動作しない可能性があります。このため、重要な実行可能ファイル (例、.exe,、dll、.ocx ファイル) にダイナミック ファイル リンクを使用しないことが推奨されます (特に、製品でこれらのファイルが正常に実行される必要がある場合)。*

*ダイナミック リンクを作成する*とき、*可能なかぎり、ディレクトリごとメソッドではなく、ベスト プラクティス メソッドを使用することをお勧めします。ただし、どちらのメソッドでも、ターゲット イメージにあるファイルがパッチのダイナミック リンクから削除されると、パッチが正しくインストールされない場合があります。*

プロジェクトでダイナミック ファイル リンクの使用を検討するとき、次の制限事項に注意してください。

- ダイナミック リンク ファイル (DLF) へカスタム アクションを作成することはできません。
- ファイル拡張子を動的にリンクされたファイルに作成することはできません。
- 動的にリンクされたファイルから COM 情報を抽出することはできません。
- ″共有″、″パーマネント″、″上書き禁止″ といったプロパティを動的にリンクされたファイルに設定することはできません。
- .NET Installer クラス機能を、動的にリンクされたファイルに設定することはできません。
- COM Interop を動的にリンクされたファイルで有効にするように指定することはできません。
- デフォルトのファイル設定 ([読み取り専用]、[非表示] など) を変更することはできません。
- 動的にリンクされたファイルにファイルのアクセス許可を設定することはできません。
- 動的にリンクされたファイルへのショートカットを作成することはできません。
- 動的にリンクされたファイルについて、スタティック スキャンまたはダイナミック スキャンを実行することはできません。

- Setup Complete Success エンドユーザー ダイアログから動的にリンクされたファイルを起動することはできません。

プロジェクトに (ダイナミック リンクを使わずに) 直接追加したファイルは、すべて内部名 (FileKey) を持ちます。カスタム アクション、ファイル拡張子、ショートカット、または他の種類のアイテムを作成したとき、この内部名を実際にポイントします。

ダイナミック リンクを使ってファイルをプロジェクトを追加すると、そのファイルは物理的にプロジェクトに追加されていません。つまり、これらのファイルが、カスタム アクション、ファイル拡張子などに関連付けることができる FileKeys を含んでいないことを意味します。

ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する

InstallShield では、ダイナミックにリンクされたファイルのコンポーネントを作成するとき、ベストプラクティスを使用する方法と 1 つのディレクトリに対して 1 つのコンポーネントを配置する 2 つの作成方法が提供されています。(コンポーネントは、個別にインストール可能な最小の製品構成単位です。InstallShield Express Edition では、コンポーネントが自動で作成されます。詳細については、「インストールの基本」を参照してください。)

ベスト プラクティス方式を使用する

ダイナミック リンク作成のベスト プラクティスに従うと、ダイナミック リンクの選択と除外のフィルター基準を満たすすべてのファイルに対して、次のタスクがビルド時に実行されます:

- ダイナミック リンクがあるフォルダーにある各ポータブル実行可能 (PE) ファイルについてコンポーネントが別々に作成されます。各 PE ファイルは、そのコンポーネントのキー ファイルです。
- ダイナミック リンクのルート レベルにあるすべての非 PE ファイルがリンクを含むコンポーネントに追加されます。
- ダイナミック リンクにサブフォルダーが含まれている場合、サブフォルダー内にあるすべての非 PE ファイルに新しいコンポーネントが作成されます。ダイナミック リンクに複数のサブフォルダーが含まれている場合、各サブフォルダー内のすべての非 PE ファイルにコンポーネントが 1 つずつ別々に作成されます。

これは、すべての新しいダイナミック リンクについてのデフォルト機能です。

ヒント: *[オプション] ダイアログ ボックスの [ファイルの拡張子] タブでは、PE ファイルとして指定するファイルの種類を選択できます。*

ディレクトリごと方式を使用する

ダイナミック リンクの作成にディレクトリごとメソッドが使用されると、ダイナミック リンクの選択と除外のフィルター基準を満たすすべてのファイルに対して、次のタスクがビルド時に実行されます:

- ファイルの種類に関わらず、ダイナミック リンクがあるソース フォルダーのルート レベルにあるすべてのファイルにコンポーネントが 1 つ作成されます。
- ダイナミック リンクに 1 つまたは複数のサブフォルダーが含まれている場合、ファイルの種類に関わらず、各サブフォルダーのすべてのファイルにコンポーネントが 1 つずつ作成されます。サブフォルダーのコンポーネント内にある最初のダイナミック リンクが付いたファイルが、そのコンポーネントのキー ファイルです。

ダイナミック リンク作成のためのこのメソッドは、ベスト プラクティス メソッドが導入される前に InstallShield で提供されていた従来型のメソッドです。

利用するダイナミック リンク作成メソッドを判別する

ほとんどのダイナミック リンクの場合、ダイナミック リンクの作成に、ベスト プラクティス メソッドが好まれます。このメソッドを、改良された QuickPatch パッケージの機能と併用することで、Windows Installer のコンポーネント規則に準拠するパッチを作成することができます。

ベスト プラクティス メソッドは、プロジェクトを InstallShield Premier または InstallShield にアップグレードする可能性があるときも推奨されます。これらのエディションを利用して、マイナー アップグレードとスモール アップデートを作成することができます。マイナー アップグレードとスモール アップデートの場合、コンポーネント、キー ファイル、および機能-コンポーネントの編成は、以前およびその後の .msi データベース間で保持されている必要があります。パッチの場合、File テーブル キーも保持される必要があります。各コンポーネント名とコンポーネント コード（および、場合により、キー ファイル）はビルド時にダイナミック ファイル リンクのディレクトリごとメソッドによって変更されるため、問題が発生する場合があります。ベスト プラクティス メソッドの利点は、ディレクトリごとメソッドに比べ、予測可能性がより高くなることです。

初期のバージョンの製品にパッチを作成するとき、初期のインストールにディレクトリごとメソッドを使ったダイナミック リンクを含む場合、同ダイナミック リンクに対してディレクトリごとメソッドを続けて使用する必要があります。ただし、アップグレード プロジェクトに新しいダイナミック リンクを追加した場合、これらの新しいダイナミック リンクに対してベスト プラクティス メソッドを使用することができます。つまり、同じプロジェクトで両方の種類のダイナミック リンクを混合して使用し、アップグレードを配布するパッチを作成できるということです。

重要: *ダイナミック リンクがあるファイルのコンポーネントを作成するとき、可能なかぎり、ディレクトリごとメソッドではなく、ベスト プラクティス メソッドを使用することをお勧めします。ただし、どちらのメソッドでも、ターゲット イメージにあるファイルがアップグレードまたはパッチのダイナミック リンクから削除されると、マイナー アップグレード、スモール アップデート、またはパッチが正しくインストールされない場合があります。*

メモ: *パッケージに含まれているファイルがターゲット システムに既に存在するファイルを上書きするとき Windows Installer が使用する規則については、「ターゲット マシン上でファイルを上書きする」をご覧ください。*

利用するダイナミック リンクの作成メソッドを指定する

[フォルダーのプロパティ] ダイアログ ボックスの [ファイルのリンク] タブでは、利用するコンポーネントの作成メソッドを指定することができます。

ファイルを動的に追加する

タスク ***インストールでファイルを動的にフォルダーに追加するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[機能]** 一覧で、ファイルに関連付ける機能を選択します。
3. **[インストール先コンピューターのフォルダー]** ペインで、動的にリンクされたファイルを含むフォルダーを右クリックして、**[ダイナミック ファイル リンク]** をクリックします。**[プロパティ]** ダイアログ ボックスが開きます。

ヒント: [インストール先コンピューターのフォルダー] ペインで、動的にリンクされたファイルを含むフォルダーが既に存在しない場合、それをまず最初に作成します。詳しくは、「ファイルとフォルダーをプロジェクトに追加する」をご覧ください。

4. [ファイルのリンク] タブをクリックします。
5. ダイナミック リンクを定義して、[OK] をクリックしてください。

ターゲット マシン上でファイルを上書きする

[ファイル] ビューでは、インストールに含まれるファイルの"上書き" プロパティを指定することができます。Windows Installer Service はインストールに含まれるファイルがターゲット システムに既に存在する場合に、各ファイルを置換するかどうか判断するために上書きプロパティを利用します。

テーブル 4-5 · ファイルの上書き規則

"ファイル上書き" プロパティ値	ターゲットマシン上での動作
Windows Installer バージョン規則 (推奨) (W)	このオプションを選択すると、ターゲット システムに既に存在するファイルを置き換えするかどうかを判別に Windows Installer バージョン規則が使われます。Windows Installer は次の規則を強制します。 · **バージョン付きファイル**—ターゲット マシン上のファイルの方がインストールされるバージョンより新しい場合でも、常に最新バージョンのファイルが保持されます。さらに、バージョン指定されていないファイルではなく、バージョン指定されたファイルが保持されます。 · **ファイル言語**—他の要素が同じ場合、インストールと同じ言語を持つファイルが他言語のバージョンより優先的に保持されます。この規則に対する例外はファイルが複数言語だった場合にのみ適用されます。単一言語のバージョンのファイルに対して複数言語のファイルが保持されます。 · **日付**—ターゲットマシン上に既に存在するファイルの変更日がそのファイルの作成日以降であった場合、ファイルは上書きされません。この規則によってアップグレードまたは再インストール中にユーザー環境設定ファイルが消去されるのを防ぐことができます。
上書きしない	このオプションを選択すると、(ターゲット システムに存在する場合) ファイルは、ファイルのバージョンに関係なく、決して上書きされません。
常に上書き	このオプションを選択すると、(ターゲット システムに存在する場合) ファイルは、ファイルのバージョンに関係なく、常に上書きされます。

[ファイル] ビューで定義済みフォルダーを表示する

タスク *定義済みのインストール先フォルダーを表示するには、次の手順に従います:*

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**インストール先コンピューターのフォルダー**] ペインで、[**インストール先コンピューター**] を右クリックして、[**定義済みフォルダーの表示**] をポイントし、使用する定義済みフォルダーをクリックします。

プロジェクトでファイルとフォルダーを検索する

多数のフォルダーとファイルをプロジェクトに追加したとき、特定のフォルダーまたはファイルの検索に手間が掛かることがあります。[ファイル] ビューでフォルダーおよびファイルの検索を行うことができます。InstallShield は、マッチしたものをすべて見つけ、最初に検索されたものをハイライトで表示します。検索条件にマッチしたものがすべて見つかるまで検索を続けることができます。

タスク *プロジェクトでファイルとフォルダーを検索するには、以下の手順を従います。*

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**インストール先コンピューターのフォルダー**] ペインで、[**インストール先コンピューター**] を選択します。
3. [**編集**] メニューで、[**検索**] をクリックします。[**検索**] ダイアログ ボックスが開きます。

   また、CTRL+F を押しても同様の結果が得られます。
4. [**検索する文字列**] ボックスで、検索するテキストを入力します。***.exe** などのワイルドカードを利用することができます。
5. [**検索対象**] でファイル、フォルダー、またはその両方を検索するかどうかを指定します。
6. その他に必要な基準を指定します。
7. [**次を検索**] をクリックします。検索基準に一致したものがあったとき、最初のアイテムは、[**インストール先コンピューターのフォルダー**] ペインまたは [**インストール先コンピューターのファイル**] ペインのどちらかで選択されます。
8. 条件に一致する次のアイテムがある場合にそれを検索するには、F3 キーを押します。このステップを必要に応じて繰り返します。

ファイルとフォルダーのアクセス許可を構成する

InstallShield で、ロックダウンされた環境で製品を実行するエンドユーザーのために、ファイルとフォルダーを保護するための設定を構成することができます。ファイルまたはフォルダーのアクセス許可を特定のグループとユーザーに割り当てることができます。たとえば、管理者グループに特定のファイルについての [読み取り]、[書き込み]、および [削除] アクセス許可を割り当てることができますが、別のグループのすべてにユーザーについては [読み取り] 許可のみ割り当てることができます。

タスク *ファイルまたはフォルダーのアクセス許可を構成するには、以下の手順に従います:*

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. ファイルの場合: [**インストール先コンピューターのファイル**] ペインで、ファイルを右クリックして[**プロパティ**] を選択します。[**プロパティ**] ダイアログ ボックスが開きます。

   フォルダーの場合: [**インストール先コンピューターのフォルダー**] ペインで、フォルダーを右クリックして[**プロパティ**] を選択します。[**プロパティ**] ダイアログ ボックスが開きます。
3. [**アクセス許可**] ボタンをクリックします。[**アクセス許可**] ダイアログ ボックスが開きます。
4. 必要に応じて、アクセス許可を追加/変更/削除します。詳細については、「ファイルとディレクトリの [アクセス許可] ダイアログ ボックス」を参照してください。

プロジェクトの [一般情報] ビューにある "ロックダウンの設定方法" 設定の選択に従って、InstallShield は ISLockPermissions テーブルまたは LockPermissions テーブルのどちらかにアクセス許可データを追加します。詳細については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

インストールに再配布可能ファイルを含める

InstallShield は、一般によく利用されるサードパーティ再配布可能ファイルを含み、.NET Framework のような技術に対するサポートをインストールに簡単に組み込むことができます。プロジェクトへ再配布可能ファイルを追加すると、再配布可能ファイルおよびすべての関連ファイルがインストールへ追加されます。これによって再配布可能ファイルのパッケージ処理が簡素化され、内部または外部での利用で一貫性が保たれます。

[再配布可能ファイル] ビューには、InstallShield に含まれているすべての InstallShield 前提条件、マージ モジュール、およびオブジェクトが含まれています。

InstallShield 前提条件

InstallShield 前提条件 は、製品が必要とする製品、またはテクノロジ フレームワークのためのインストールです。既存の InstallShield 前提条件は、どれでもインストール プロジェクトに追加することができます。

プロジェクトに InstallShield 前提条件を含めると、複数のインストールを連鎖することができるため、1 度に 1 つの実行シーケンスのみしか実行できない Windows Installer 制限をバイパスできます。Setup.exe セットアップランチャーは、連鎖を管理するブートストラップ アプリケーションとしての役割を果たします。

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれています。これを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。*

InstallShield では、次の 2 つのタイプの InstallShield 前提条件がサポートされています:

- **セットアップ前提条件** – この種類の前提条件のインストールは、インストールの実行の前に実行されます。
- **機能前提条件** – この種類の前提条件は、1 つまたは複数の機能に関連付けられています。機能前提条件は、前提条件を含む機能がインストールされたときに、その前提条件がシステム上に既にインストールされていない場合にインストールされます。したがって、機能の条件がターゲット システム上で満たされていない場合、またはエンド ユーザーが機能のインストールを行わないことを選択した場合、その機能はインストール

されません。その結果、インストールされる別の機能にも機能前提条件が関連付けられていない限り、関連付けられた機能前提条件はいずれもインストールされません。

マージ モジュール

マージ モジュール (.msm ファイル) には、個別機能をインストールするために必要なロジックとファイルのすべてが含まれています。たとえば、一部のアプリケーションには、Visual Basic ランタイム ライブラリが必要です。機能にファイルを含めてインストール要件を調べる必要はなく、プロジェクトに含まれる機能の 1 つに Microsoft C++ ランタイム ライブラリを添付するだけでこれを実行できます。

メモ: *[再配布可能ファイル] ビューに含まれているマージ モジュールの多くは Microsoft またはその他のサードパーティによるものです。InstallShieldでは、これらのモジュールを無料配布することによって、インストール プロジェクトの作成を支援します。ただし、サードパーティが作成したモジュールに存在する問題を InstallShield が修正したり直すことはできません。サード パーティが作成したモジュールに関する問題は、ベンダーへお問い合わせください。*

オブジェクト

オブジェクトには、マージ モジュール同様、個別の機能をインストールするために必要なロジックとファイルがすべて含まれています。InstallShield に含まれる DirectX オブジェクトなどのオブジェクトは、ウィザードを使ってカスタマイズする必要があります。オブジェクトをインストールに追加すると、すぐに該当するカスタマイズウィザードが開きます。オブジェクトを追加時点でカスタマイズすることも、ウィザードをいったんキャンセルし、後でオブジェクトを右クリックして [オブジェクトの設定変更] を選択することによってカスタマイズすることもできます。

ライブ再配布可能ファイルギャラリー

多くの再配布可能ファイルはサイズが大きいため、プロジェクトで利用可能なものでも InstallShield のインストールと同時にコンピューターへ追加されない場合があります。その場合も、これらの再配布可能ファイルはインターネットからコンピューターへダウンロードすることができます。

構成可能マージ モジュール

再配布可能ファイルの出荷

InstallShield は、インストール プロジェクトに組み込むことが可能なサードパーティ再配布可能ファイルを提供します。プロジェクトに、たとえば Crystal Reports といった再配布可能技術を含む場合、その再配布可能ファイルについてベンダーからライセンスが付与されていなくてはなりません。適切なライセンス無しにこれらの技術を再配布することは法的に許可されていません。詳細は、ベンダーのマニュアルを参照してください。

InstallShield プロジェクトでリリースをビルドする時、InstallShield によってビルド出力に様々な InstallShield 再配布可能ファイルが含まれます。InstallShield の使用許諾契約に基づいて、ビルド出力に含まれるこれらの InstallShield 再配布可能ファイルを使用することができます。これらのファイルのほとんどは *InstallShield Program Files フォルダー*¥Redist フォルダーにインストールされ、必要に応じてビルドに含まれます。以下はInstallShield 再配布可能ファイルのリストです。

- _isres_LanguageID.dll
- ClrSuitePSHelper.dll

- ClrWrap.dll
- CommonHelper.dll
- corecomp.ini
- default.pal
- DLLWrap.dll
- dotnetfx.exe
- DotNetInstaller.exe
- EulaScrollWatcher.dll
- FileBrowse.dll
- IISHelper.dll
- IISRT.dll
- InstallShield.ClrHelper.dll
- InstallShield.Interop.Msi.dll
- ISBEW64.exe
- ISChain.exe
- ISChainPackages.dll
- ISComSrv.dll
- ISExpHlp.dll
- isexternalui.dll
- IsLockPermissions.dll
- ISNetAPI.dll
- ISNetApiRT.dll
- ISNetworkShares.dll
- ISRegSvr.dll
- Isrt.dll
- ISScheduledTasks.dll
- IsSchRpl.dll
- ISSetup.dll
- ISSQLSrv.dll
- IsWebDeploy.dll
- ISWindowsFeaturesAction.dll
- ISWindowsFeaturesAction64.dll
- ISXmlCfg.dll

- Layout.bin
- PowerShellWrap.dll
- PrqLaunch.dll
- QuickPatchHelper.dll
- SerialNumCAHelper.dll
- SetAllUsers.dll
- Setup.exe
- Setup.ini
- setup.inx
- setup.isn
- setup.ocx
- setup.skin
- Setup_UI.dll
- setupPreReq.exe
- SetupSuite.exe
- SetupSuite64.exe
- SFHelper.dll
- SQLRT.dll
- SuiteAppxHelper.exe
- XMLRT.dll
- 次のフォルダーのサブフォルダーにインストールされるイメージ ファイル

  InstallShield Program Files Folder¥Support¥Themes
- 次のフォルダー、およびこのフォルダーの scale-150 および scale-200 サブフォルダーにインストールされるイメージとアイコン ファイル

  InstallShield Program Files フォルダー¥Redist¥Language Independent¥OS Independent

再配布可能ファイルギャラリーを管理する

多くの再配布可能ファイルはサイズが大きいため、プロジェクトで利用可能なものでも InstallShield のインストールと同時にコンピューターへ追加されない場合があります。その場合も、これらの再配布可能ファイルはインターネットからコンピューターへダウンロードすることができます。

アイコンの種類から再配布可能ファイルのステータスを判断することができます。利用可能なアイコンのリストとそれぞれの説明は以下の通りです。

テーブル 4-6・再配布可能ファイルのアイコン

アイコン	説明
	この InstallShield 前提条件はコンピューターにインストールされます。
	この InstallShield 前提条件ははコンピューターにインストールされず、ダウンロードして利用することができます。
	この InstallShield 前提条件はプロジェクトに含まれていますが、その場所は [オプション] ダイアログ ボックスの [前提条件] タブ で指定されたディレクトリの 1 つとしてリストされません。
	このマージ モジュールはコンピューターにインストールされます。
	このマージ モジュールはコンピューターにインストールされず、ダウンロードして利用することができます。
	ご利用のコンピューターにはこのマージ モジュールの古いバージョンがインストールされています。最新版のダウンロードが可能です。
	このオブジェクトはコンピューターにインストールされます。
	このオブジェクトはコンピューターにインストールされず、ダウンロードして利用することができます。
	ご利用のコンピューターにはこのオブジェクトの古いバージョンがインストールされています。最新版のダウンロードが可能です。

***ヒント・** ご利用のコンピューターにインストールされていないオブジェクトまたはマージ モジュールを追加する場合、プロジェクトをビルドしたときにビルドエラーが生成されます。ビルドエラーを避けるには、リリースから再配布可能ファイルを削除するか、プロジェクトを再ビルドする前にダウンロードします。再配布可能ファイルがコンピューターにインストールされていない場合、その再配布可能ファイルの [場所] 列で **[ダウンロードが必要]** が指定されます。*

ライブ再配布可能ファイル ギャラリーについて

[再配布可能ファイル] ビューで表示される再配布可能ファイル ギャラリーには、インストールに含むことができる InstallShield 前提条件、マージ モジュール、オブジェクトが含まれています。

InstallShield 前提条件

InstallShield では、多数の InstallShield 前提条件が提供されています。すべての InstallShield 前提条件ファイル (.prq) は、次の場所に格納されています:

InstallShield Program Files フォルダー¥SetupPrerequisites

マージ モジュール

マージ モジュールは、さまざまなソースから利用することができます。InstallShield には多くの再配布可能モジュールが含まれていますが、新しいバージョンがある場合や、他のソフトウェア開発会社から必要なモジュールがリリースされている場合があります。

[再配布可能ファイル] ビューに一覧表示されているマージ モジュール ファイルのソースは、[オプション] ダイアログ ボックスの [マージ モジュール] タブで指定されたフォルダーです。[オプション] ダイアログ ボックスにアクセスするには、[ツール] メニューから [オプション] を選択します。

次のディレクトリは InstallShield に付属されているモジュールのデフォルトの場所です。

InstallShield Program Files フォルダー¥Modules¥i386

オブジェクト

InstallShield は、多くの再配布可能オブジェクトを提供します。さらに、他の開発者が作成したオブジェクトをプロジェクトに追加することも考えられます。

InstallShield に付属されているオブジェクトのデフォルトの場所は以下の通りです。

InstallShield Program Files フォルダー¥Objects

上記の場所に含まれるオブジェクトは 再配布可能ファイルビューに一覧表示されています。

再配布可能ファイルをコンピューターにダウンロードする

[再配布可能ファイル] ビューを使用して、最新の InstallShield 前提条件、マージ モジュールとオブジェクトを Revenera Web サイトからコンピューターへダウンロードすることができます。再配布可能ファイルがコンピューターにインストールされていない場合、その再配布可能ファイルの [場所] 列で **[ダウンロードが必要]** が指定されます。

タスク ***特定の InstallShield 前提条件、マージ モジュール、オブジェクトをダウンロードするには、以下の手順を実行します:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. すべてのタイプ、InstallShield 前提条件、マージ モジュール、またはオブジェクトのうち、どの種類の再配布可能ファイルを表示するのかを指定するには **[表示するオブジェクトタイプ]** リストで適切なオプションを選択します。
3. ダウンロードする InstallShield 前提条件、マージ モジュール、または、オブジェクトを右クリックして、**[選択したアイテムをダウンロード]** をクリックします。

タスク ***インストール プロジェクトに必要なすべての InstallShield 前提条件、マージ モジュール、およびオブジェクトをダウンロードするには、以下の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. InstallShield 前提条件、マージ モジュール、、またはオブジェクトのどれかを右クリックして **[必要なアイテムをすべてダウンロード]** をクリックします。

再配布可能ファイルギャラリーに InstallShield 前提条件を追加する

タスク ***再配布可能ファイル ギャラリーへ InstallShield 前提条件を追加するには、以下の手順を実行します。***

1. 新規または更新された InstallShield 前提条件 (.prq) ファイルを取得します。
2. Windows Explorer を利用して、新しい前提条件を次の場所にコピーします。

   InstallShield Program Files フォルダー¥SetupPrerequisites
3. 現在 InstallShield が開いている場合は閉じます。
4. InstallShield を起動します。

変更が [再配布可能ファイル] ビューに反映されます。

再配布可能ファイルギャラリーから InstallShield 前提条件を削除する

タスク ***再配布可能ファイル ギャラリーから InstallShield 前提条件を削除するには、以下の手順を実行します。***

1. InstallShield を閉じます。
2. Windows Explorer を利用してギャラリーから削除する InstallShield 前提条件を探して削除します。InstallShield セットアップ前提条件ファイルは、次のディレクトリに格納されています:

   InstallShield Program Files フォルダー¥SetupPrerequisites
3. InstallShield を起動します。

変更が [再配布可能ファイル] ビューに反映されます。

マージ モジュールを参照する

プロジェクトに追加するマージ モジュールが [再配布可能ファイル] ビューに表示されていない場合、それを検索してプロジェクトおよび [再配布可能ファイル] ビューに追加することができます。

タスク ***マージ モジュールを参照するには、次の操作を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. アイテムを右クリックして、**マージ モジュールの参照**をクリックします。**[開く]** ダイアログ ボックスが開きます。
3. マージ モジュールファイルを参照します。
4. [OK] をクリックします。

マージ モジュールを参照した場合に起きること

InstallShield は、マージ モジュールへの参照を明示パスとして維持しません。そのかわり、マージ モジュール GUID とマージ モジュールロケールに基づいてマージ モジュールのキーを生成します。InstallShield がマージ モジュールにアクセスする時、そのキーと一致するファイルを [マージ モジュールの場所] ボックスで指定したフォルダー内で探します。[マージ モジュールの場所] ボックスは、[オプション] ダイアログ ボックスの [マージ モジュール] タブにあります。

マージ モジュールを参照する場合、マージ モジュールを含むフォルダーへのパスがマージ モジュールの場所ボックス内のパスのリストへ追加されます。さらに GUID:ロケールキーが選択されたファイルに基づいてインストール プロジェクトに追加されます。

インストールへの影響

マージ モジュールの場所ボックスにある 2 つのマージ モジュールが GUID:ロケール キーを持つ場合、ファイル名が異なる場合でも 1 つだけがインストールに含まれます。InstallShield による [マージ モジュールの場所] ボックスの検索方法のため、どのマージ モジュールが含まれるかを予期するのは不可能です。

マージ モジュールの場所ボックス内のディレクトリ数の限定

共有マージ モジュールギャラリを使用する場合、ターゲットマシンに存在するバージョンより古いか新しいバージョンのマージ モジュールが存在する可能性があります。このため、マージ モジュールの場所ボックスのディレクトリ数を制限した方が賢明な場合があります。

タスク ***ディレクトリ数を制限するには、次のいずれかを実行します。***

- Windows エクスプローラーを使用して、**[マージ モジュールの場所]** ボックスに既にリストされているフォルダーの 1 つにマージ モジュールをコピーします。
- 検索パスからデフォルトのフォルダーを削除して、共有の場所だけが参照されるようにします。

再配布可能ファイル ギャラリーにマージ モジュールを追加する

タスク ***再配布可能ファイルギャラリーへマージ モジュールを追加するには、以下の手順に従います:***

1. 新規、またはアップデートされたマージ モジュールを取得します。
2. Windows Explorer を使用して、**[オプション]** ダイアログ ボックスの [マージ モジュールのオプション] タブで指定されたフォルダーの 1 つに新しいモジュールをコピーします。

   InstallShield に付属されているモジュールのデフォルトの場所は以下の通りです:

   Program Files フォルダー¥*InstallShield フォルダー*¥Modules¥i386
3. 現在 InstallShield が開いている場合は閉じます。
4. InstallShield を起動します。

変更が [再配布可能ファイル] ビューに反映されます。

再配布可能ファイルギャラリーからマージ モジュールを削除する

タスク ***再配布可能ファイル ギャラリーからモジュールを削除するには、以下の手順に従います:***

1. 現在 InstallShield が開いている場合は閉じます。
2. Windows Explorer を利用してギャラリーから削除するマージ モジュールを探して削除します。[オプション] ダイアログ ボックスの[マージ モジュールのオプション] タブで指定されたディレクトリをすべて検索して下さい。
3. InstallShield を起動します。

変更が [再配布可能ファイル] ビューに反映されます。

メモ: *インストールに現在関連付けられているマージ モジュールを削除すると* ***[マージ モジュールが見つかりません]*** *というメッセージが表示され、モジュールをインストールに追加できないことを通知します。*

InstallShield 前提条件、マージ モジュール、およびオブジェクトをプロジェクトに組み込む

InstallShield では、InstallShield 前提条件、マージ モジュール、および InstallScript オブジェクトとしてパッケージ化されている多数のサード パーティ再配布可能ファイルが提供されています。これらのビルトイン再配布可能ファイルをインストール プロジェクトに追加することができます。詳しくは、ドキュメントの該当セクションをご覧ください。

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれています。これを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。また、これらのエディションには、自作のマージ モジュールを作成するときに利用できるマージ モジュール プロジェクト タイプが含まれています。*

InstallShield 前提条件、マージ モジュール、およびオブジェクトをプロジェクトに追加する

2 つの種類の再配布可能ファイル (マージ モジュールおよびオブジェクト) は、インストールされるために機能への関連付けが必要です。マージ モジュールまたはオブジェクトの関連付けを行うとき、関連付けることができる機能やサブ機能の数に制限はありません

InstallShield 前提条件をプロジェクトに追加した場合、その前提条件はメインのインストールが開始される前に実行されるため、デフォルトでどの機能にも関連付けられていません。これらの前提条件は、*セットアップ前提条件*と呼ばれます。必要に応じて、InstallShield 前提条件を現在プロジェクトに存在する 1 つ以上の機能に関連付けることができます。

タスク ***プロジェクトに InstallShield 前提条件、マージ モジュールやオブジェクトを追加するには、以下の手順に従います:***

1. [**アプリケーション データの指定**] の下にあるビュー リストで、[**再配布可能ファイル**] をクリックします。
2. 追加する再配布可能ファイルの前にあるチェック ボックスを選択します。オブジェクトを選択した場合、関連するウィザードが開いてカスタマイズ処理の手順を案内します。
3. マージ モジュールまたはオブジェクトの場合: [**条件付きインストール**] ペインで、この再配布可能ファイルを含める必要がある各機能のチェック ボックスを選択します。

   前提条件を機能を関連付けられている場合: [**条件付きインストール**] ペインで、この前提条件を含める必要がある各機能のチェック ボックスを選択します。機能に前提条件を関連付けない場合、[**機能の選択の前にインストールする**] チェック ボックスを選択状態のままにしておきます。このチェック ボックスは、InstallShield 前提条件をプロジェクトに追加したとき、デフォルトで選択されています。

ヒント: *[再配布可能ファイル] ビューの右側のペインに、提供されている再配布可能ファイルの一覧から選択されたマージ モジュール、オブジェクト、または InstallShield 前提条件の詳細が表示されます。この詳細ペインで、再配布可能ファイルがインストールするファイルなどの情報を確認することができます。このビューで [詳細を表示] ボタンをクリックして、詳細の表示/非表示を切り替えることができます。*

メモ: *プロジェクトに追加した InstallShield 前提条件の [場所] 列で* ***[ダウンロードが必要]****を指定した場合、その前提条件はコンピューターにインストールされません。前提条件をプロジェクトに含める必要がある場合、コンピューターにインターネットからその前提条件をダウンロードすることができます。1 つまたは複数の前提条件をダウンロードしないでリリースをビルドした場合で、さらに前提条件を Setup.exe から抽出する、または (Web からエンド ユーザーのコンピューターへダウンロードするのではなく) ソース メディアからコピーするを指定した場合、1 つまたは複数のビルドエラーが生成される可能性があります。ビルド エラーを除去するには、プロジェクトから前提条件を削除してコンピューターにダウンロードするか、またはリリースのInstallShield 前提条件の場所をダウンロード オプションに変更してから、リリースを再ビルドします。*

InstallShield 前提条件とオブジェクトの取得

一部の InstallShield 前提条件とオブジェクトは、InstallShield と共にインストールされませんので注意してください。これらについては、場合により、ダウンロードする必要があります。詳細については、「InstallShield のアップデートを取得する」を参照してください。

プロジェクトから InstallShield 前提条件、マージ モジュール、またはオブジェクトを削除する

タスク *プロジェクトから InstallShield 前提条件、マージ モジュール、またはオブジェクトを削除するには、以下の手順に従います:*

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. インストールから削除する InstallShield 前提条件、マージ モジュール、またはオブジェクトの前にあるチェック ボックスをクリアします。

再配布可能ファイルと、それに関連付けられている依存関係はすべて自動的に削除されます。

InstallShield 前提条件、マージ モジュールおよびオブジェクトのファイルを判別する

InstallShield 前提条件、マージ モジュール、オブジェクト内にあるファイルの一覧を表示する必要がある場合は、[再配布可能ファイル] ビュー内でそれらを表示できます。このビューの右側のペインに、提供されている再配布可能ファイルの一覧から選択された InstallShield 前提条件、マージ モジュール、またはオブジェクトの詳細が表示されます。この詳細ペインで、再配布可能ファイルがインストールするファイルなどの情報を確認することができます。このビューで [詳細を表示] ボタンをクリックして、詳細の表示/非表示を切り替えることができます。

ヒント: *マージ モジュールまたはオブジェクトに含まれるファイルを参照する他の方法は、ナレッジベース記事 Q106474 を参照してください。この記事はダウンロードが可能な [マージ モジュール依存関係] ビューアーへのリンクを含んでいます。*

プロジェクトに含まれている InstallShield 前提条件を使って作業する

InstallShield 前提条件 は、製品が必要とする製品、またはテクノロジ フレームワークのためのインストールです。InstallShield で提供されている InstallShield 前提条件の一例として、Java Runtime Environment (JRE) および SQL Server Express Edition があります。既存の InstallShield 前提条件は、どれでもインストール プロジェクトに追加することができます。

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれています。これを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。*

プロジェクトに InstallShield 前提条件を含めると、複数のインストールを連鎖することができるため、1 度に 1 つの実行シーケンスのみしか実行できない Windows Installer 制限をバイパスできます。Setup.exe セットアップランチャーは、連鎖を管理するブートストラップ アプリケーションとしての役割を果たします。

[再配布可能ファイル] ビューでは、InstallShield 前提条件を Express プロジェクトに追加します。

セットアップ前提条件と機能前提条件の違い

メインのインストールの [ユーザー インターフェイス] シーケンスが開始する前に実行される InstallShield 前提条件は、*セットアップ前提条件*と呼ばれます。セットアップ前提条件は、インストールされている製品のすべての構成に対してインストールする必要がある、または、それ自身のインストールで使用する機能を提供するベースアプリケーションおよびテクノロジ フレームワークに適しています。プロジェクトに InstallShield 前提条件を追加すると、デフォルトでセットアップ前提条件タイプの InstallShield 前提条件になります。

Express プロジェクト タイプでは、InstallShield 前提条件をメインのインストールに含まれる機能に関連付けることができます。InstallShield 前提条件が 1 つまたは複数の機能と関連付けられている場合、それは *機能前提条件*と呼ばれます。機能の前提条件は、エンドユーザーがインストールする機能を選択したときインストールされます。つまり、マージ モジュール同様、機能前提条件は、それを含む 1 つまたは複数の機能がインストールされたときのみインストールされます。このため、機能前提条件は、インストールされている製品の一部の構成に対してのみインストールする必要がある、および、それ自身のインストールでは使用されないアプリケーションまたはコンポーネントに適しています。

ニーズに最も適した InstallShield 前提条件の種類を判別するときに役に立つ情報は、次のセクションで読むことができます:

セットアップ前提条件の特別考慮

次は、1 つまたは複数のセットアップ前提条件をプロジェクトに含めるとき、次のヒントを参考にしてください:

.NET Framework の要件

製品の実行に、.NET Framework がターゲット システムにインストールされている必要がある場合、.NET Framework 再配布可能ファイルをプロジェクトに含めることができます。.NET Framework がターゲット システムにないとき、インストール中にインストールされます。詳細については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。

エディション: *InstallShield Premier および InstallShield では、Windows Installer エンジンおよび .NET Framework のインストールの前後でインストールされるように InstallShield 前提条件を構成することができます。*

セットアップランチャーの代わりに .msi パッケージを起動する

セットアップ前提条件が含まれているインストールで、エンドユーザーが、製品の Setup.exe セットアップランチャーを起動する代わりに、.msi パッケージを直接起動した場合、セットアップ前提条件のインストールは実行されません。前提条件がターゲット システムに存在しないとき、製品が適切に動作しないことがあります。これは、.msi パッケージが Setup.exe ファイルにストリームされない非圧縮のリリースをビルドしたとき発生することがあります。

機能前提条件の特別考慮

次は、1 つまたは複数の機能前提条件をプロジェクトに含めるとき、次のヒントを参考にしてください:

Windows Installer の要件

プロジェクトに Windows Installer をインストールする前提条件が含まれている場合、前提条件は機能前提条件ではなくセットアップ前提条件になります。したがって、この前提条件を機能に関連付けることはできません。

.NET Framework の要件

Windows Installer をインストールする前提条件が含まれているプロジェクトで、インストールで .NET Framework が存在している必要がある場合 (ファイルを GAC にインストール場合など)、.NET Framework の前提条件は機能前提条件ではなくセットアップ前提条件である必要があります。したがって、この前提条件を機能に関連付けることはできません。

機能前提条件の再起動に関する潜在的な問題

プロジェクトに InstallShield 前提条件が含まれていて、潜在的に再起動の必要がある場合、この前提条件を機能に関連付けないことをお勧めします。機能前提条件によって再起動がトリガーされた場合、再起動のあと再び ReadyToInstall ダイアログが表示され、エンド ユーザーは [インストール] ボタンをクリックして残りのインストールを続行する必要があります。

必要ディスク容量の計算

Windows Installer がファイルのコスト計算関連のアクションを実行するとき、機能の前提条件で必要なディスク容量は自動的には含まれません。したがって、CustomSetup ダイアログが実行時に表示された場合、異なる機能に表示された空きディスク領域の容量は、機能前提条件に必要なディスク容量が考慮されないため正確ではない場合があります。また、可能性として、メイン インストールに機能前提条件を付加したとき、ターゲット システムのディスク空き容量が足りなくなるということも発生します。このような場合、インストールの途中でターゲット システムの空きディスク容量が足りなくなることがあります。

プロジェクトで、InstallShield 前提条件を機能に関連付ける

プロジェクトで、InstallShield 前提条件が機能と関連付けられている場合、それは *機能前提条件*と見なされます。機能に関連付けられていない場合、それは *セットアップ前提条件*と見なされます。

インストール プロジェクトに InstallShield 前提条件を追加したとき、デフォルトで、その前提条件がセットアップ前提条件として追加されます。セットアップ前提条件は、それをプロジェクトに既に存在する 1 つまたは複数の機能に関連付けることで機能前提条件にすることができます。

タスク ***InstallShield 前提条件を機能に関連付けるには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. 再配布可能ファイルの一覧で、機能に関連付ける InstallShield 前提条件を選択します。

   メモ: *InstallShield 前提条件のチェック ボックスは、既に選択されている必要があります。選択されたチェック ボックスは、前提条件がプロジェクトに含められることを示します。詳細については、「InstallShield 前提条件、マージ モジュール、およびオブジェクトをプロジェクトに追加する」を参照してください。*

3. **[条件付インストール]** ペイン内で、この InstallShield 前提条件を追加するすべての機能のチェック ボックスを選択します。

前提条件を新しい機能に関連付ける場合、まずその機能を作成する必要があります。新しい機能の作成方法については、「機能の作成」をご覧ください。

前提条件をプロジェクト内のすべての機能に関連付けたあとで新しい機能を追加した場合、その機能前提条件は新しい機能に自動的に関連付けられません。

メモ: 機能前提条件には、セットアップ前提条件には適用されない制限事項があります。詳細については、「セットアップ前提条件と機能前提条件の違い」を参照してください。

プロジェクトで、機能から InstallShield 前提条件の関連付けを解除する

プロジェクトで、InstallShield 前提条件が機能と関連付けられている場合、それは *機能前提条件*と見なされます。機能に関連付けられていない場合、それは *セットアップ前提条件*と見なされます。

タスク ***機能から InstallShield 前提条件を削除するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. 再配布可能ファイルの一覧で、機能との関連付けを解除する InstallShield 前提条件を選択します。
3. **[条件付きインストール]** ペインで、**[機能の選択の前にインストールする]** チェック ボックスを選択します。このチェック ボックスは、InstallShield 前提条件をプロジェクトに追加したとき、デフォルトで選択されています。

メモ: セットアップ前提条件には、機能前提条件にはない利点がいくつかあります。詳細については、「セットアップ前提条件と機能前提条件の違い」を参照してください。

InstallShield 前提条件のインストール順を指定する

[再配布可能ファイル] ビューでは、プロジェクトに複数の InstallShield 前提条件を含める場合、それらをインストールする順番を指定することができます。

タスク ***InstallShield 前提条件がターゲット マシン上にインストールされる順番を指定するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. 必要な InstallShield 前提条件をプロジェクトに追加していない場合、それを行います。
3. 任意の再配布可能ファイルを右クリックして、**InstallShield 前提条件の順番を設定**をクリックします。**InstallShield[前提条件のインストール順]** ダイアログ ボックスが開きます。
4. リストから前提条件を選択してから上下矢印をクリックしてインストールの順に並べます。

メモ: 順番を指定するとき、InstallShield では、セットアップ前提条件と機能の前提条件の違いは識別されません。したがって、プロジェクトにセットアップ前提条件と機能前提条件が混在する場合、[InstallShield 前提条件のインストール順] ダイアログ ボックスで、それらがすべて同じ一覧内に表示されます。実行時に、メインのインストールが起動される前、Setup.exe セットアップランチャーはセットアップ前提条件のみ評価し (必要時のみ)、[InstallShield 前提条件のインストール順] ダイアログ ボックスで指定した順番でそれらをインストールします。このあと、インストールの後半で、Windows Installer エンジンによって機能前提条件のみが評価され (必要時のみ)、それらが指定された順番でインストールされます。

セットアップ前提条件と機能前提条件 (2 つの異なる種類の InstallShield 前提条件) の違いについては、「セットアップ前提条件と機能前提条件の違い」を参照してください。

InstallShield 前提条件を含むリリースを構成する

InstallShield 前提条件を含むインストールをパッケージする場合、以下のいずれか 1 つの方法を使ってエンドユーザーに対して InstallShield 前提条件ファイルを提供することができます。

- InstallShield 前提条件ファイルを、ソース メディアに格納する。
- InstallShield 前提条件ファイルを Setup.exe に圧縮し、実行時に必要に応じて抽出されるように設定する。
- 必要な場合、インストールはプロジェクトに含まれた InstallShield 前提条件ファイルを、必要に応じて各前提条件の InstallShield 前提条件ファイル (.prq) で指定した URL からダウンロードすることができます。

インストールに含まれる各 InstallShield 前提条件の提供方法を指定することができます。詳細については、「特定の InstallShield 前提条件の実行時の場所を指定する」を参照してください。

リリースに含まれるすべての InstallShield 前提条件を同じ方法で利用可能にする場合、リリース レベルで個別の方法をオーバーライドすることも可能です。詳細については、「リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する」を参照してください。

エディション: *InstallShield Premier および InstallShield では、Windows Installer エンジンおよび .NET Framework のインストールの前後でインストールされるように InstallShield 前提条件を構成することができます。*

InstallShield 前提条件を含むディレクトリを指定する

InstallShield前提条件ファイル (.prq) のデフォルトの場所は、以下のとおりです:

InstallShield Program Files フォルダー¥SetupPrerequisites

InstallShield では、ローカル マシンまたはネットワーク上の追加の場所または代替の場所を指定することができます。この柔軟な機能を使って、InstallShield 前提条件をソース コード管理システムに格納し、InstallShield 前提条件の共通のセットを他のチーム メンバーと共有することができます。

InstallShield では、InstallShield 前提条件ファイル (.prq) の検索パスを指定するいくつかの方法があります:

- InstallShield 内部から編集またはビルドを行う場合、[ツール] メニューで [オプション] をクリックすると表示される [オプション] ボックスにある [前提条件] タブを使って、マシン共通および現在のユーザーのフォルダをコンマ区切りのリストで指定できます。
- ISCmdBld.exe を使って、コマンドラインからビルドする場合は、-prqpath パラメーターを使ってフォルダのコンマ区切りのリストを指定します。

  .ini ファイルを使って ISCmdBld.exe パラメーターを指定する場合、.ini ファイルの [Mode] セクションで新しい PrerequisitePath パラメーターを使用して、フォルダのコンマ区切りのリストを指定できます。
- MSBuild または Team Foundation Server (TFS) を使ってビルドする場合、InstallShield タスクで PrerequisitePath パラメーターを使います。このパラメーターは、デフォルトのターゲット ファイルが使用されたとき、ItemGroup InstallShieldPrerequisitePath として露出されます。複数のパスを指定するには、順序指定されたパスの配列を使用します。

ハードコード化されたパスの代わりに、以下の例のようにパスにパス変数を使用することができます:

<ISProductFolder>¥SetupPrerequisites,<ISProjectFolder>¥MyCustomPrerequisites

[再配布可能ファイル] ビューには、[オプション] ダイアログ ボックスの [前提条件] タブで指定されている様々な検索パスに含まれている .prq ファイルに対応する InstallShield 前提条件の名前がリストされます。同じ .prq ファイルが複数の検索パスに含まれている場合、InstallShield は最初に検出されたインスタンスのみを表示します。InstallShield は、最初に [前提条件] タブでユーザーごとの設定にリストされた各パスをチェックする。次に、マシン共通の設定にリストされている各パスをチェックします。

ビルド時、プロジェクトに 1 つ以上の InstallShield 前提条件が含まれている場合、InstallShield が指定された場所を検索して、必要に応じて適切な InstallShield 前提条件をリリースに含みます。同じ .prq ファイルが複数の検索パスに含まれている場合、InstallShield は最初に検出されたインスタンスのみをビルドに含みます。.prq ファイルの検索は以下の順で行われます:

1. InstallShield が –prqpath コマンドライン パラメーター、PrerequisitePath .ini ファイル パラメーター、または InstallShield タスクの PrerequisitePath パラメーターを通して指定されたパスをチェックする。
2. InstallShield が [前提条件] タブでユーザーごとの設定にリストされた各パスをチェックする。
3. InstallShield が [前提条件] タブのマシン共通の設定にリストされた各パスをチェックする。
4. 前述の場所のいずれにもパスが指定されていない場合、InstallShield はデフォルトの場所 (*InstallShield Program Files フォルダー*¥SetupPrerequisites) をチェックします。

特定の InstallShield 前提条件の実行時の場所を指定する

InstallShield では、プロジェクトの各 InstallShield 前提条件に異なる実行時の場所を指定することができます。

タスク ***インストールに含まれる各 InstallShield 前提条件に異なる場所を指定するには、次の手順に従います。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. インストールに含める InstallShield 前提条件の 1 つのチェック ボックスを選択します。
3. InstallShield 前提条件を右クリックして、**[プロパティ]** をクリックします。InstallShield**[前提条件のプロパティ]** ダイアログ ボックスが開きます。
4. **[ビルドの場所]** リストで、適切なオプションをクリックします。

指定した場所は、リリース レベルでオーバーライドすることができます。個々の InstallShield 前提条件に選択した値のオーバーライドを禁止するには、リリース レベルの設定にある "InstallShield 前提条件の場所" 設定を [個々の選択に従う] に設定します。詳細については、「リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する」を参照してください。

InstallShield 前提条件を含むリリースをビルドする

InstallShield は、前提条件を含まないプロジェクトの Setup.exe ファイルをビルドするとき、以下の場所に格納されたベース Setup.exe ファイルからビルドを始めます。

InstallShield Program Files フォルダー¥redist¥Language Independent¥i386

但し、InstallShield が前提条件を含むプロジェクトの Setup.exe をビルドするとき、Setup.exe ファイルをベースとして利用することはできません。これは、Setup.exe ファイルに前提条件を含むことができないためです。代わりに SetupPrereq.exe と呼ばれる、多少サイズが大きいファイルを利用します。このベース SetupPrereq.exe ファイルは

ベース Setup.exe ファイルと同じディレクトリに配置されています。2 つの異なるベースファイル (Setup.exe および SetupPrereq.exe) が利用されるため、エンドユーザーに配布される最終ビルド Setup.exe ファイルの追加サイズ オーバーヘッドは、実際プロジェクトに前提条件を含む作業を行うインストール作成者にのみ負担が掛かることになります。

InstallShield 前提条件を含むインストールの実行時の動作

ヒント: *セットアップ前提条件と機能前提条件 (2 つの異なる種類の InstallShield 前提条件) の違いについては、「セットアップ前提条件と機能前提条件の違い」を参照してください。*

InstallShield 前提条件を含むインストールの概要

次のプロシージャは、エンドユーザーがセットアップ前提条件と機能前提条件を含むインストールを起動したときに、一般的に実行時に発生する事柄の説明です。

1. セットアップランチャー (通常、Setup.exe と呼ばれます) は、必要に応じて言語の選択ダイアログを表示します。
2. セットアップランチャーは、必要に応じてセットアップの前提条件ダイアログを表示し、セットアップ前提条件インストールを起動します。
3. インストールは、エンド ユーザーが機能を選択したり、項目を構成したりできるインストール UI を表示します。インストール UI は、進行状況ダイアログを表示します。
4. セットアップランチャーは、必要に応じて、次のように、機能の前提条件のインストールを起動します。
   a. ビルトイン InstallShield カスタム アクション ISInstallPrerequisites (SetupProgress ダイアログと ExecuteAction アクションの間にスケジュールされる) は、インストールすることが選択された機能を Windows Installer プロパティ IsPrerequisiteFeatures のリストと比較します。全く一致しなかった場合、機能前提条件はインストールされません。
   b. ISInstallPrerequisites アクションは、セットアップランチャーを検索および起動し、またインストール中の機能のリストを提供します。セットアップランチャーのパスは、Windows Installer プロパティ SETUPEXEDIR と SETUPEXENAME によって識別されます:

      [SETUPEXEDIR]¥[SETUPEXENAME]

      ISInstallPrerequisites は、その場所でセットアップランチャーが見つからなかった場合、別の場所を検索します。初回のインストールでは、SourceDir が確認されます。メンテナンス モードでは、インストール ソース パスに関連するパスが確認されます。

      ISInstallPrerequisites がセットアップランチャーが見つけられなかった場合、または、複数の .exe ファイルが見つかった場合、エンドユーザーにプロンプトが表示され、セットアップランチャー ファイルを参照するように要求します。エンドユーザーがファイルを識別すると、インストールが続行します。そうでない場合、インストールは終了します。
   c. セットアップランチャーは、機能のリストを評価してインストールする機能前提条件を選択し、また適切なインストールを起動します。
5. インストールはエンド ユーザーの選択にしたがって、ターゲット システム上で変更を完了します。
6. 進行状況ダイアログから [セットアップの完了] ダイアログに切り替わります。

InstallShield 前提条件を含むインストールのユーザー インターフェイス

ターゲット システムにインストールする必要があるセットアップ前提条件が 1 つ以上ある場合、通常、メイン インストールの実行が開始される前に、セットアップ前提条件ダイアログが表示されます。このセットアップ前提条件によって、ターゲット システムに不足している非表示のセットアップ前提条件がすべて表示されます。エンド ユーザーがこのダイアログで [インストール] ボタンをクリックすると、必要なセットアップ前提条件のインストールが起動されます管理者権限が必要とマークされているセットアップ前提条件が 1 つ以上あり、かつ、インストールがユーザー アカウント制御 (UAC) が有効にされたシステムで実行される場合、このダイアログの [インストール] ボタンが、昇格された権限が必要であることをエンドユーザーに通知するシールド アイコンと共に表示されます。

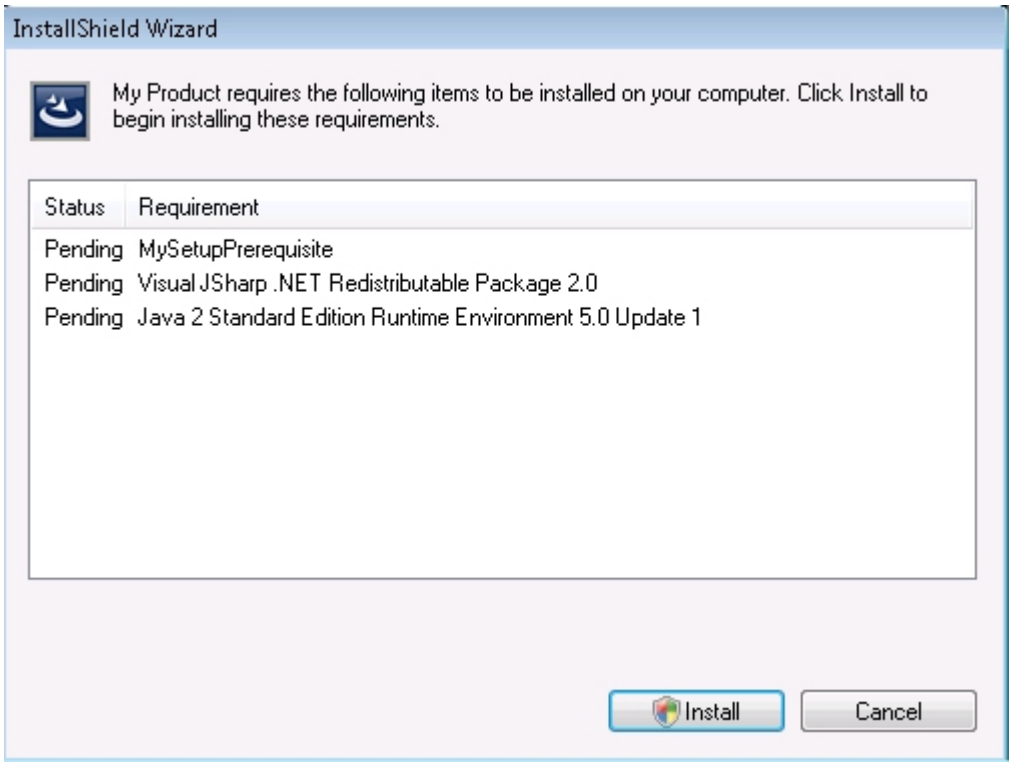


図 4-1: [インストールするセットアップ前提条件] リストを表示するサンプル [セットアップ前提条件] ダイアログ

セットアップ前提条件が非表示と構成された場合、この前提条件はセットアップ前提条件ダイアログに表示されませんが、インストールはされます。インストールに含まれるすべてのセットアップ前提条件が隠されている場合、セットアップ前提条件ダイアログの代わりに、セットアップランチャーの標準初期ダイアログがインストールで表示されます。

セットアップ前提条件のインストールが起動するファイルが .msi パッケージで、前提条件が進行状況を表示するようにマークされている場合、前提条件のインストール中、ユーザー インターフェイスで、Windows Installer からのインストールの進行状況メッセージおよびステータス バージョンが表示されます。

セットアップ前提条件がエンドユーザーによってオプションでインストールされると構成されている場合、エンド ユーザーが前提条件をインストールするかどうかを選択することができるメッセージ ボックスが表示されます。

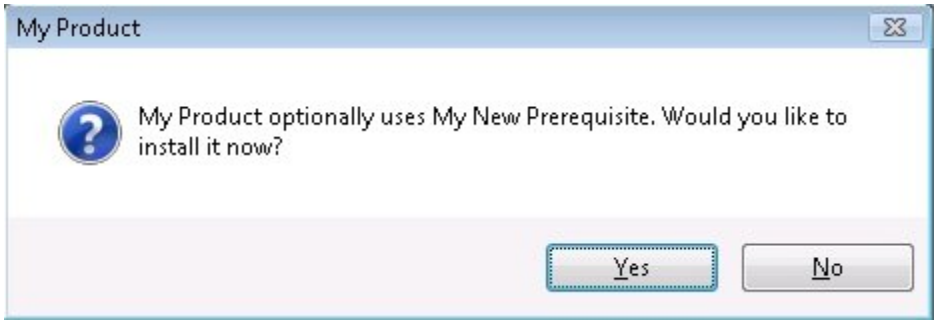


図 4-2: オプション前提条件のメッセージ ボックス

インストールに機能前提条件が含まれている場合、セットアップランチャーが表示するすべての前提条件のダイアログでそれらが表示されません。ただし、ユーザー インターフェイスでは、必要に応じて進行状況メッセージが表示されます。また、機能前提条件がオプションとマークされている場合、オプションの前提条件メッセージボックスが表示されます。

サイレント シナリオ—抑制されたユーザー インターフェイス

セットアップ前提条件と機能前提条件は、インストールがサイレントで実行されるときもインストールされます。InstallShield 前提条件は、次のすべてのシナリオでサポートされています:

- **サイレント セットアップランチャーと可視の .msi パッケージ** — セットアップランチャーのユーザー インターフェイスは抑制されますが、.msi パッケージのユーザー インターフェイスは表示されます。たとえば、エンドユーザーは次のコマンドライン ステートメントを使用できます:

  Setup.exe /s

  このシナリオでは、言語の選択ダイアログとセットアップ前提条件ダイアログは表示されません。

- **可視のセットアップランチャーとサイレント .msi パッケージ** — セットアップランチャーのユーザー インターフェイスは表示されますが、.msi パッケージのユーザー インターフェイスは抑制されます。たとえば、エンドユーザーは次のコマンドライン ステートメントを使用できます:

  Setup.exe /v"/qn"

  このシナリオでは、メイン インストールの言語の選択ダイアログとセットアップ前提条件ダイアログは表示されません。ただし、エンドユーザーは、インストールする機能を指定するとき、コマンドラインから ADDLOCAL、ADDSOURCE、ADDDEFAULT、ADVERTISE などの Windows Installer プロパティを設定することができます。

- **サイレント セットアップランチャーとサイレント .msi パッケージ** — セットアップランチャーと .msi パッケージのユーザー インターフェイスは抑制されます。たとえば、エンドユーザーは次のコマンドライン ステートメントを使用できます:

  Setup.exe /s /v"/qn"

  このシナリオでは、すべてのセットアップランチャーと .msi パッケージのダイアログが抑制されます。

メイン インストール内にある .msi パッケージの UI シーケンスがスキップされると、セットアップランチャーは ADDLOCAL、ADDSOURCE、ADDDEFAULT、ADVERTISE などの Windows Installer プロパティを評価して、インストールが必要な機能前提条件があるかどうかを判別し、その結果にしたがって、機能前提条件をインストールします。

UAC のプロンプト

構成方法によって、InstallShield 前提条件を含むインストールがインストール中のいくつかの時点で、Windows Vista 以降のシステム上で昇格された権限のプロンプトを表示することができます:

1. エンド ユーザーが Setup.exe ファイルを起動するとき
2. Setup.exe ファイルが、昇格された権限を必要とするセットアップ前提条件を起動するとき
3. Setup.exe ファイルが、昇格された権限を必要とする機能前提条件を起動するとき
4. Windows Installer が .msi パッケージの [実行] シーケンスを開始するとき

詳細については、「インストール中におけるユーザー アカウント制御のプロンプトの数を最小化する」を参照してください。

InstallShield 前提条件の動作を変更する

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれており、これを使って InstallShield 前提条件の動作を変更することができます。*

InstallShield の InstallShield 前提条件エディターを利用して、特定の前提条件の動作および構成の設定を構成することができます。たとえば、InstallShield 前提条件エディターを使って、次のタスクを実行することができます:

- 実行時に、前提条件を セットアップの前提条件ダイアログで表示するかどうかを指定します。
- 前提条件がオプションかどうかを指定します。
- Windows Installer エンジンおよび .NET Framework のインストールの前または後でインストールされるように InstallShield 前提条件を構成します。
- サイレントで実行、前提条件が起動されたとき、それに渡されるコマンドライン パラメーターを指定します。
- 前提条件に管理者権限が必要かどうかを指定します。

インストールに InstallShield 前提条件が含まれていたアプリケーションをアンインストールする

インストールがアプリケーションおよび 1 つまたは複数の InstallShield 前提条件から構成される場合があります。エンドユーザーがコントロール パネルのプログラムの追加と削除を使ってアプリケーションをアンインストールした場合、InstallShield 前提条件はマシン上にインストールされたままの状態です。InstallShield 前提条件インストールによって、[プログラムの追加と削除] にエントリが追加された場合、エンドユーザーは [プログラムの追加と削除] を通して InstallShield 前提条件を削除することができます。

インストール プロジェクトに含まれているマージ モジュールとオブジェクトを使って作業する

ドキュメントのこのセクションでは、インストール プロジェクト内からマージ モジュールおよびオブジェクトを使用する方法について説明します。

エディション: *InstallShield Premier および InstallShield では、ユーザーが自分でマージ モジュールを作成することができます。*

マージ モジュールを含むディレクトリを指定する

InstallShield では、ローカル マシンまたはネットワーク上でマージ モジュール (.msm ファイル) を格納する場所を指定できます。この柔軟な機能を使って、マージ モジュールをソース コード管理システムに格納し、マージ モジュールの共通のセットを他のチーム メンバーと共有することができます。

InstallShield では、マージ モジュールの検索パスを指定するいくつかの方法があります:

- InstallShield 内部から編集またはビルドを行う場合、[ツール] メニューで [オプション] をクリックすると表示される [オプション] ボックスにある [マージ モジュール オプション] タブを使って、マシン共通および現在のユーザーのフォルダをコンマ区切りのリストで指定できます。
- ISCmdBld.exe を使って、コマンドラインからビルドする場合は、-o パラメーターを使ってフォルダのコンマ区切りのリストを指定します。

  .ini ファイルを使って ISCmdBld.exe パラメーターを指定する場合、.ini ファイルの [Mode] セクションで新しい MergeModulePath パラメーターを使用して、フォルダのコンマ区切りのリストを指定できます。
- MSBuild または Team Foundation Server (TFS) を使ってビルドする場合、InstallShield タスクで MergeModulePath パラメーター を使います。このプロパティは、デフォルトのターゲット ファイルが使用されたとき、ItemGroup InstallShieldMergeModulePath として露出されます。複数のパスを指定するには、順序指定されたパスの配列を使用します。

ハードコード化されたパスの代わりに、以下の例のようにパスにパス変数を使用することができます:

<ISProductFolder>¥MergeModules,<ISProjectFolder>¥MyCustomMergeModules

[再配布可能ファイル] ビューには、[オプション] ダイアログ ボックスの [マージ モジュール] タブで指定されている様々な検索パスに含まれているマージ モジュールファイルに対応するマージ モジュールの名前がリストされます。同じマージ モジュールが複数の検索パスに含まれている場合、InstallShield は最初に検出されたインスタンスのみを表示します。InstallShield は、最初に [マージ モジュール] タブでユーザーごとの設定にリストされた各パスをチェックします。次に、マシン共通の設定にリストされている各パスをチェックします。

ビルド時、プロジェクトに 1 つ以上のマージ モジュールが含まれている場合、InstallShield が指定された場所を検索して、必要に応じて適切なマージ モジュールをリリースに含みます。同じマージ モジュールが複数の検索パスに含まれている場合、InstallShield は最初に検出されたインスタンスのみをビルドに含みます。マージ モジュールの検索は以下の順で行われます:

1. InstallShield が [マージ モジュール] タブでユーザーごとの設定にリストされた各パスをチェックする。
2. InstallShield が [マージ モジュール] タブのマシン共通の設定にリストされた各パスをチェックする。
3. InstallShield が -o コマンドライン パラメーター、MergeModulePath .ini ファイル パラメーター、または InstallShield タスクの MergeModulePath パラメーターを通して指定されたパスをチェックする。
4. 前述の場所のどこにもパスが指定されていない場合、InstallShield は次の順序でデフォルト ディレクトリをチェックします:
   a. *InstallShield Program Files フォルダー*¥System
   b. *InstallShield Program Files フォルダー*¥Modules¥i386

c. *InstallShield Program Files フォルダー*¥Objects

d. *InstallShield Program Files フォルダー*¥Modules¥i386¥Japanese

e. *InstallShield Program Files フォルダー*¥Modules¥i386¥German

f. *Program Files フォルダー*¥Common Files¥Merge Modules

オブジェクトおよびマージ モジュールの構成を変更する

オブジェクトまたはマージ モジュールをプロジェクトに追加した後、場合によって構成を変更する必要があります。

タスク ***オブジェクトまたはマージ モジュールをプロジェクトに含めた後、その構成を変更するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. 変更するオブジェクトまたはマージ モジュールを右クリックして、**[オブジェクトの設定を変更する]**、または **[マージ モジュールの構成]** を選択します。該当のウィザードが開いて、構成を変更することができます。

マージ モジュールの除外と依存関係

マージ モジュールをインストールに追加する場合、そのモジュールに適した条件が必要です。それらの条件とは、マージ モジュールの除外と依存関係です。

モジュールの依存関係

モジュールが作成される時、その依存関係も設定されます。つまり依存関係を含まない限り、モジュールが動作しないということです。InstallShield は、依存関係がローカルの再配布可能ファイルギャラリーに格納されている場合、モジュールの依存関係を自動的にインストール プロジェクトに関連付けます。これらのモジュールが見つからなかった場合は、各モジュールのコピーを取得して再配布可能ファイルビューでこれらをインストール プロジェクトに関連付ける必要があります。

モジュールの除外

一部のモジュールは他のモジュールがあると、正常に作動しません。この場合モジュール作成者は他のモジュールを除外して、再配布可能ファイルビューの [説明] ウィンドウに必要なモジュールが表示されるようにしなくてはなりません。新しく関連付けられたモジュールに除外があり、かつ除外されたモジュールが既にセットアップに関連付けられている場合、InstallShield はインストール内の同モジュールへのすべての参照を削除します。

注意: *除外するモジュールの追加後にインストールに追加されるすべての除外されたモジュールは削除されず、また互換性がないことを示す警告が表示されることもありません。*

マージ モジュールのインストール先をオーバーライドする

サードパーティ マージ モジュールは変更すべきではありませんが、いくつかのサードパーティ マージ モジュールおよび InstallShield 作成のマージ モジュールのインストール先を上書きすることは可能です。

メモ: この手順は、マージ モジュールの TARGETDIR ディレクトリ、または TARGETDIR から直接派生したディレクトリのみを転送します。マージ モジュールが定義済みフォルダー (たとえば、SystemFolder) へファイルを送るように構成されている場合 、モジュールのインストール先をオーバーライドすることはできません。

タスク ***マージ モジュールのインストール先をオーバーライドするには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[再配布可能ファイル]** をクリックします。
2. マージ モジュールの横にあるチェック ボックスを選択して、インストールに追加します。
3. モジュールを右クリックして、**プロパティ**をクリックします。[マージ モジュールプロパティ] ダイアログ ボックス が表示されます。
4. **[インストール先]** ボックスで、インストール先を入力するか、定義済みのインストール先から 1 つ選択します。
5. [OK] をクリックします。
6. **[条件付きインストール]** ペインで、マージ モジュールを含める機能を選択します (複数可)。

マージ モジュールに関するトラブルシューティング

インストールに現在関連付けられているマージ モジュールを削除すると、[マージ モジュールが見つかりません] というメッセージが表示されます。このことにより、そのモジュールをインストールに追加できないことがわかります。

Windows Installer 再配布可能ファイルをプロジェクトに追加する

Windows Installer は Windows のほとんどのバージョンに組み込まれていますが、Windows Inslaller ベースのインストールは Windows Installer の最新バージョンでのみ動作する特定の機能に依存している可能性があります。

InstallShield では、Windows Installer の再配布可能ファイルをプロジェクトに含めることができます。Windows Installer 実行可能ファイルのプロジェクトへの追加方法は、インストールに必要な Windows Installer のバージョンによって異なります。

Windows Installer 5 および Windows Installer 4 は、再配布可能ファイルとして利用することができない点にご留意ください。

Windows Installer の配布

デフォルトで、InstallShield は Setup.exe をセットアップ パッケージと同時に作成します。セットアップランチャーは、インストールで Windows Installer エンジンをインストールする場合、必ず必要です。

Windows Installer 再配布可能ファイルをプロジェクトに含めるには、以下のいずれかを実行します:

- **Windows Installer 4.5 の場合**—プロジェクトに 1 つまたは複数の Microsoft Windows Installer 前提条件を追加します。InstallShield は、異なるバージョンの Windows をターゲットとするいくつかのバージョンを含みます。詳細については、「Microsoft Windows Installer の前提条件を含める」を参照してください。
- **Windows Installer 3.1、3.0、または 2.0 の場合**—[リリース] ビューの Setup.exe タブでは、Setup.exe ランチャーを使用するかどうか、これらの Windows Installer 再配布可能ファイルの 1 つのバージョンを含めるかどうか、

および含める Windows Installer のバージョンなどの情報を指定することができます。詳細については、「Setup.exe タブ」を参照してください。

代りに、プロジェクトに 1 つまたは複数の Microsoft Windows Installer 前提条件を追加することともできます。詳細については、「Microsoft Windows Installer の前提条件を含める」を参照してください。

インストール プロセスの概要

実行時、Setup.exe は Windows Installer が既にターゲット システムにインストールされているかどうかを判別します。ターゲット システムに Windows Installer が見つかり、かつ、最小バージョンの要件を満たした場合、インストール パッケージが起動されます。Windows Installer がインストールされていない場合や、または新しいバージョンをインストールする必要がある場合、Setup.exe は Windows Installer をインストールしてからインストール パッケージを起動します。Windows Installer を更新するためにシステムを再起動する必要があることに注意してください。

Microsoft Windows Installer の前提条件を含める

InstallShield には、複数の Windows Installer バージョン用の InstallShield 前提条件が含まれています。[再配布可能ファイル] ビューを利用してプロジェクトに InstallShield 前提条件を追加できます。

.NET Framework 再配布可能ファイルをプロジェクトへ追加する

製品の実行に、.NET Framework がターゲット システムにインストールされている必要がある場合、.NET Framework 再配布可能ファイルをプロジェクトに追加することができます。.NET Framework がターゲット システムにないとき、インストール中にインストールされます。

プロジェクトに、.NET Framework の言語パックの再配布可能ファイルを含めることもできます。言語パックには、英語以外の言語のための翻訳済みテキスト (エラー メッセージなど) が含まれています。

.NET Framework および .NET Framework 言語パックをプロジェクトに追加する方法は、アプリケーションに必要な .NET Framework のバージョンによって異なります。

メモ: *一部の .NET Framework バージョンは、以前の .NET Framework バージョンを含みます:*

- *.NET Framework 3.5 は、.NET Framework 3.0 SP1 と .NET Framework 2.0 SP1 を含みます*
- *.NET 3.0 Framework SP1 は .NET Framework 2.0 SP1 を含みます。*
- *.NET 3.0 Framework RTM は .NET Framework 2.0 RTM を含みます。*

.NET サポートをプロジェクトに含めるには、以下のいずれかを実行します:

- **.NET Framework 4.5 Full、4.5 Web、4.0 Full、4.0 Client、3.5 SP1、3.5、3.0 SP1、3.0, 2.0 SP2、または 2.0 SP1 再配布可能ファイルの場合**—適切な Microsoft .NET Framework 前提条件を追加します。

  詳細については、「Microsoft .NET Framework および Microsoft .NET Framework 言語パックの前提条件を含める」を参照してください。

- **.NET Framework 2.0、1.1、または 1.0 再配布可能ファイルの場合**—[リリース] ビューの .NET/J# タブで、リリースの .NET に関する設定を構成します。

InstallShield 前提条件の取得

一部の InstallShield 前提条件は、InstallShield と共にインストールされていませんので注意してください。これらについては、場合により、ダウンロードする必要があります。詳細については、「InstallShield のアップデートを取得する」を参照してください。

Microsoft .NET Framework および Microsoft .NET Framework 言語パックの前提条件を含める

InstallShield には、一部のバージョンの .NET Framework と .NET Framework 言語パックの InstallShield 前提条件が含まれています。これらのバージョンの .NET Framework と言語パックを再配布する場合、これらの InstallShield 前提条件をプロジェクトに含めることができます。

以下は、InstallShield 前提条件として提供されている .NET Framework 再配布可能ファイルの一覧です。関連する言語パック 前提条件も、配布されている場合、含まれています。

- Microsoft .NET Framework 4.5(完全パッケージに 1 つ、Web パッケージに 1 つの InstallShield 前提条件が含まれています。サイズは Web パッケージのほうが小さいですが、実行時にターゲット システムでインターネット接続が必要になります。)
- Microsoft .NET Framework 4 Full。これは、.NET Framework 4 をターゲットにするアプリケーションを実行および開発するのに必要な .NET Framework ランタイムと関連ファイルをインストールします。(完全パッケージに 1 つ、Web ダウンロード パッケージに 1 つの InstallShield 前提条件が含まれています。サイズは Web ダウンロード パッケージのほうが小さいですが、実行時にターゲット システムでインターネット接続が必要になります。)
- Microsoft .NET Framework 4 Client。ほとんどのクライアントアプリケーションを実行するために必要な .NET Framework ランタイムおよび関連ファイルをインストールします。(完全パッケージに 1 つ、Web ダウンロード パッケージに 1 つの InstallShield 前提条件が含まれています。サイズは Web ダウンロード パッケージのほうが小さいですが、実行時にターゲット システムでインターネット接続が必要になります。)
- Microsoft .NET Framework 3.5 (完全パッケージに 1 つ、Web ダウンロード パッケージに 1 つの InstallShield 前提条件が含まれています。サイズは Web ダウンロード パッケージのほうが小さいですが、実行時にターゲット システムでインターネット接続が必要になります。)
- Microsoft .NET Framework 3.0 SP1 (これは、実行時にターゲット システムでインターネット接続が必要になる Web ダウンロード パッケージです。)
- Microsoft .NET Framework 3.0
- Microsoft .NET Framework 2.0 SP2
- Microsoft .NET Framework 2.0 SP1

ヒント: *.NET Framework 再配布可能ファイルの他のバージョンに関する詳細は、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。*

これらの InstallShield 前提条件のインストールは、サイレント モードで実行されます。したがって、.NET Framework インストールが実行されるときの言語は、問題になりません。

InstallShield 前提条件のインストールは、特定の HKEY_LOCAL_MACHINE キーの Install または InstallSuccess の値データを確認して、.NET Framework の対応するバージョンが既にターゲット マシンにインストールされているかどうかを確認します。詳細については、[再配布可能ファイル] ビューで InstallShield 前提条件をクリックして、[詳細] ペインで定義された条件を参照してください。

DirectX 9.0 オブジェクトを含める

DirectX は、最新のグラフィック カードを含む、マルチメディア アプリケーションおよびハードウェアの API ライブラリがサポートとしています。製品をターゲット システムにインストールするために DirectX が必要な場合、DirectX オブジェクトをプロジェクトに追加することができます。ターゲット システムに DirectX がない場合、DirectX がインストール時にインストールされます。

インストール後、DirectX ランタイムはアンインストールできません。DirectX はシステム コンポーネントであるため、エンドユーザーが DirectX をアンインストールするには、オペレーティング システムの再インストールが必要になります。

ヒント: *DirectX オブジェクトは InstallShield と共にインストールされないため、ダウンロードする必要があります。詳細については、「再配布可能ファイルをコンピューターにダウンロードする」を参照してください。*

再配布可能ファイル

DirectX は DirectX 9.0c コアおよびオプションのコンポーネントをすべてインストールします。

DirectX オブジェクトをプロジェクトに含める

DirectX オブジェクトを Express プロジェクトに含めると、DirectX オブジェクト ウィザードが起動されます。

DirectX オブジェクトは、圧縮、非圧縮の両方のインストールで使用することができます。DirectX オブジェクト ウィザードを利用して、DirectX ファイルを Disk1 フォルダーに含めるか、または .msi ファイルにストリームするかを指定することができます。

- ファイルを Disk1 フォルダーにあるフォルダーに含めるように指定すると、ビルド時に作成中のインストールに対して DirectX フォルダーが作成され、リリースの Disk1 フォルダーに配置されます。DirectX フォルダーは [セットアップ ファイル] ビューの Disk1 領域に表示されます。
- ファイルを Disk1 フォルダーに含めないように指定すると、ファイルはインストールの .msi ファイルに埋め込まれます。

メモ: *DirectX インストールを起動するカスタム アクションは Windows Vista 以降のシステムで昇格された権限を使って実行できるように、[実行] シーケンスにスケジュールされ、遅延システム コンテキストで実行されます。*

Express プロジェクトの DirectX オブジェクト ファイルを更新する

ある DirectX ファイルのアップデートを入手して、DirectX オブジェクトに含める場合、それらを適切な InstallShield Program Files サブフォルダーに他の DirectX ファイルと共に配置します。保存先:

InstallShield Program Files フォルダー¥Objects¥DirectX9c¥Redist

一部のファイルが製品で必要なく、インストールに含める必要がない場合、それらをフォルダーから削除することもできます。InstallShield の現在のバージョンがリリースされてからマイクロソフトがリリースしている可能性があるアップデートに関する詳細など、DirectX 再配布可能ファイルの関する詳しい情報は、最新の DirectX SDK または マイクロソフトの MSDNWeb サイト をご覧ください。

ビルド時に、InstallShield がリリースのビルドをするとき、DirectX フォルダー内の再配布可能ファイルはすべて使用されます。

アプリケーションの依存関係を識別する

あるファイルが、他のファイルの関数に依存してタスクを実行することがよくあります。ただし、インストールプロジェクトにアプリケーションファイルを含める際に、*「依存関係」*と呼ばれる他のファイルに気が付かない場合があります。InstallShieldでは、これらのファイルを見つけて作業を行う次のスキャン ウィザードが用意されています。スキャナーには、[依存関係] ビューからアクセスできます。

テーブル 4-7 • InstallShield で提供されているスキャン ウィザード

スキャナー	関数
スタティック スキャン ウィザード	プロジェクトのポータブル実行可能ファイル (.exe、.ocx、.com、.tlb、.hlp、および .chm) を探し、必要な依存関係を検出します。
ダイナミック スキャン ウィザード	実行可能ファイルが実行中にシステムを監視して、実行可能ファイルで必要となる可能性がある .dll または .ocx ファイルを確認します。

スタティックおよびダイナミック スキャン ウィザードを使用する場合、InstallShield を使ってスタティックまたはダイナミック スキャンを実行するとき必ず自動的に選択または除外されるファイルを指定することができます。詳細については、「依存関係スキャナーでファイルをフィルターする」を参照してください。

スタティック スキャン

スタティック スキャン ウィザードを利用して、プロジェクトに既に追加されているファイルをスキャンして必要な依存関係があるかどうかをチェックすることができます。このウィザードはプロジェクトに含まれるすべてのポータブル実行可能ファイル (.exe、.dll、.ocx、.sys、.com、.drv、.scr、および .cpl ファイル) をスキャンして、必要な依存関係を確認します。ウィザードで検出された依存関係のリストが表示され、それぞれをプロジェクトに含めるかどうかを指定することができます。

プロジェクトに追加された新規ファイルは、ファイルが依存している同じ機能に追加されるので、インストールが必要なときに確実に実行されます。

このウィザードに関する詳しい情報は、「スタティック スキャン ウィザード」をご覧ください。

ダイナミック スキャン

ダイナミック スキャン ウィザードは、実行可能ファイルの実行中にシステムを監視する使い易いツールです。ウィザードでは実行可能ファイルが必要とする可能性のある .dll および .ocx ファイルのリストが表示され、それぞれをプロジェクトに含めるかどうかを指定できます。

スキャンする実行可能ファイルは、既にプロジェクトに含められているものでも、後からウィザードで追加するものでも構いません。

このウィザードに関する詳しい情報は、「ダイナミック スキャン ウィザード」をご覧ください。

依存関係スキャナー結果の確認

スタティック スキャン ウィザード およびダイナミック スキャン ウィザードを使ってプロジェクトをスキャンして、製品が必要とする可能性のあるその他のファイルを識別することができます。両方のウィザード パネルには、プロジェクトに追加する必要がある可能性の高いファイルおよびマージ モジュールのリストが表示されます。スキャン結果をよく確認して、指定された各ファイルやマージ モジュールをプロジェクトに追加する必要があるかどうかを判断してください。

両方のスキャナーは、異なる方法で依存関係を識別します。1 つのキャナーでは識別されなかった依存関係が、もう 1 つのスキャナーによって識別される場合もあります。そのため、スタティック スキャナーおよびダイナミック スキャナーの両方を使って、依存関係の可能性があるファイルのより完全なリストを作成することをお勧めします。一部の状況において、依存関係スキャナーが製品で必要のないファイルまたはマージ モジュールを依存関係として識別する場合もあります。その場合、ウィザードでは識別された各依存関係を含めたり除外したりすることが可能なので、不要なファイルまたはマージ モジュールを除外することができます。さらに、InstallShield を使ってスタティックまたはダイナミック スキャンを行うときは常に自動的に含めたり除外したりするファイルをマシン全体で指定することも可能です。詳細については、「依存関係スキャナーでファイルをフィルターする」を参照してください。

依存関係を識別するときに最も良い結果を得るために、クリーン マシン上で製品とそのインストールを十分にテストすることが推奨されます。製品が予定通りに動作しなかった場合、マシン上で足りない依存関係がないか、またそれをインストールに含むべきかどうかを判断してください。

依存関係スキャナーでファイルをフィルターする

スタティックおよびダイナミック スキャン ウィザードを実行すると、インストールに追加したくない依存関係ファイルが一覧表示されることがあります。スキャナーを実行するたびにこれらのファイルが追加されないようにするために、Filters.xml を編集できます。このファイルを使うと、スキャナーが無視する、または含めるファイルを指定できます。

filters.xml は、以下の場所にあります:

InstallShield Program Files フォルダー¥Support

ファイルは、スキャナーが正しく動作するために、編集後もこの場所に残る必要があります。

ヒント: Filters.xml *ファイルを使って、COM 抽出中にどのレジストリ項目を除外するのかを制御することもできます。詳細については、「COM 抽出のレジストリ変更をフィルターする」を参照してください。*

ファイルを除外する

Filters.xml ファイルの <Exclude> 要素には、スキャナーが除外する各ファイルのサブ要素を追加します。ここにリストされたファイルは、スキャナーによってインストール プロジェクトに追加されることはありません。

デフォルトで <Exclude> 要素には、すべての Windows ベースのマシン上に存在する一般的なシステム ファイルのサブ要素が含まれています。

ファイルを含める

Filters.xml ファイルの <Include> 要素を使うと、Exclude 要素のサブ要素である個別のファイルをオーバーライドすることができます。スキャナーは、<Include> 要素のサブ要素にリストされているファイルをすべてインストール プロジェクトに追加します。これは <Exclude> 要素のサブ要素にリストされているファイルにも適用されます。

メモ: 次の重要なオペレーティング システム ファイルは、<Include> 要素のサブ要素に追加してスキャナーでは認識されません。

- *kernel32.dll*
- *ntdll.dll*
- *user32.dll*
- *gdi32.dll*
- *advapi32.dll*
- *shell32.dll*
- *ole32.dll*

<Exclude> および <Include> 要素にファイルを指定する

<Exclude> または <Include> 要素にファイルをリストする場合、そのファイルをサブ要素として追加しなくてはなりません。適切にフォーマットされたサブ要素のサンプルは次の通りです。

```
<File name="myfile.dll" path="[SystemFolder]" We="needthis"/>
```

テーブル 4-8 : <File> サブ要素で利用可能な属性

属性	説明
名前	この属性は小文字のみ使用可能です。この属性の値 (例えば、前述の **myfile.dll**) は、含めるまたは除外するファイルの名前を示します。
パス	この属性はオプションです。この属性の値 (例えば、前述の **[SystemFolder]**) は、ファイルのパスを示します。

その他の属性はすべてオプションで、スキャナーが認識することはありません。例えば前述の **We** 属性と、それに対応する **"needthis"** 値のように、特定の項目が含まれるまたは除外される理由を説明する属性を追加したい場合があります。

重要: *その場合、XML コードが適切に書かれていることを再確認してください。不適切な場合には、すべてのフィルターが失敗します。多くの場合、インターネット エクスプローラーで* Filters.xml *ファイルを開いて、不適切に書かれた XML コードを確認することができます。<Filters>、<Include>、および <Exclude> 要素は縮小および展開が可能です。これらが不可能な場合、コードにエラーが無いか確認してください。*

<Exclude> または <Include> 要素にサブ要素を追加した場合、それらが誤ってコメントアウトされたセクションに配置されていないことを確認してください。InstallShield は Filters.xml *ファイルのコメントアウト部分を無視します。*

次のサンプル XML コードで、Filters.xml ファイルの形式を説明します。

```
<Filters>
 <Include>
  <!--この要素にファイルを追加する方法
  -->
  <File name="mfc42.dll" We="needthis"/>
 <Include>
 <Exclude>
```

```
  <!--この要素にファイルを追加する方法
  -->
  <Registry key="HKEY_CLASSES_ROOT¥Interface¥{00020404-0000-0000-C000-000000000046}"/>
  <File name="12520437.cpx" path="[SystemFolder]" wrp="4.0-10.0" />
  <File name="12520850.cpx" path="[SystemFolder]" wrp="4.0-10.0" />
 </Exclude>
</Filters>
```

COM サーバーの登録

ほとんどのアプリケーションでは、正常に機能するために特定の COM サーバーが必要です。COM サーバーがオペレーティング システムで認識されるようにするには、登録が必要です。

InstallShield がサポートする、ターゲットマシンへの COM サーバー登録方法には 2 種類あります。

- COM 情報をファイルから抽出して、インストール中に COM サーバーを登録するときに使用することができます。
- 自己登録をサポートするファイルであれば、自己登録が可能です。自動登録は、Windows Installer で COM 情報を取り出すことによりファイルを登録または登録解除する場合に比べて信頼性に欠けます。

最初にリストされている、COM 情報を抽出する方法が推奨されます。

重要: *WinRunner など一部のアプリケーションは、COM 抽出エンジンにフック .dll ファイルを挿入します。その結果、COM 抽出が失敗してメッセージ「ISRegSpy は、次のモジュール %1 がこのプロセスにフックされ、そのために ISRegSpy の誤動作が発生していることを検出しました。アプリケーションをシャットダウンして COM 抽出を再起動してください。」このメッセージが表示された場合、ダイアログ ボックスの指示通りアプリケーションをシャットダウンして COM 抽出を再開します。*

自己登録ではない .exe ファイルには、"自己登録" プロパティを選択しないでください。.exe ファイルを自己登録するには、/regserver コマンドを使って .exe ファイルを起動する必要があります。ただし、.exe ファイルがコマンドライン スイッチをサポートしない場合、ビルド時の抽出中に .exe が起動されます。

タスク ***InstallShield に COM サーバーを登録するには、次の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. 登録するファイルを右クリックして **[プロパティ]** をクリックします。**[プロパティ]** ダイアログ ボックスが開きます。
3. **[COM と .NET の設定]** タブをクリックします。
4. **[登録タイプ]** リストから、ファイル上で実行する登録のタイプを選択します。
5. **[OK]** をクリックします。

COM 抽出のレジストリ変更をフィルターする

InstallShield による COM サーバーからの不必要な COM データの抽出 (ビルド時またはデザイン時) を阻止するために、Filters.xml ファイルを編集して特定のレジストリ キーを除外することができます。Filters.xml は、以下の場所にあります:

InstallShield Program Files フォルダー¥Support

ファイルは、COM 抽出が正しく動作するために、編集後もこの場所に残る必要があります。

ヒント: Filters.xml *ファイルを使って、依存関係のスキャン中にどのファイルを除外するまたは含めるのかを制御することもできます。詳細については、「依存関係スキャナーでファイルをフィルターする」を参照してください。*

COM 抽出からレジストリ キーを除外する

Filters.xml ファイルの <Exclude> 要素には、COM 抽出プロセスが除外する各レジストリ キーのサブ要素を追加します。ここにリストされているすべてのキーは、製品がアンインストールされるのと同時にアンインストールされることはありません。

デフォルトで、<Exclude> 要素は必要とされる一般的なシステム レジストリ キーのサブ要素を含みます。

<Exclude> 要素にレジストリ キーを指定する

<Exclude> 要素にキーをリストする場合、そのキーをレジストリ サブ要素として追加しなくてはなりません。

InprocServer32 レジストリ キー、そのすべての値、およびそのサブキーのすべてに対する変更を阻止する、適切にフォーマットされたレジストリ サブ要素のサンプルは次の通りです。

```
<Registry key="HKEY_LOCAL_MACHINE¥SOFTWARE¥Classes¥CLSID¥{00000231-0000-0010-8000-00AA006D2EA4}¥InprocServer32"/>
```

InprocServer32 レジストリ キーのデフォルト値のみに対する変更を阻止する、適切にフォーマットされたレジストリ サブ要素は次の通りです。

```
<Registry key="HKEY_LOCAL_MACHINE¥SOFTWARE¥Classes¥CLSID¥{00000231-0000-0010-8000-00AA006D2EA4}¥InprocServer32" value=""/>
```

InprocServer32 レジストリ キーのThreadingModel 値名のみに対する変更を阻止する、適切にフォーマットされたレジストリ サブ要素は次の通りです。

```
<Registry key="HKEY_LOCAL_MACHINE¥SOFTWARE¥Classes¥CLSID¥{00000231-0000-0010-8000-00AA006D2EA4}¥InprocServer32"
value="ThreadingModel"/>
```

テーブル 4-9・<Registry>サブ要素で利用可能な属性

属性	説明
キー	この属性は小文字のみ使用可能です。この属性の値 (例えば、前述の HKEY_LOCAL_MACHINE¥SOFTWARE¥Classes¥CLSID¥{00000231-0000-0010-8000-00AA006D2EA4}¥InprocServer32) は、フィルターを行うレジストリ キーの名前を示します。
値	この属性はオプションです。 ・ レジストリ キー全体に対する変更を阻止するには、値属性を含まないでください。 ・ 特定のキーのデフォルト値に対する変更を阻止するには、この属性の値をヌルに設定します。 ・ 特定のキーの値名に対する変更を阻止するには、この属性の値をそのレジストリ値の名前に設定します。

<Registry> サブ要素のその他の属性はオプションで、COM 抽出プロセスでは認識されません。特定の項目が除外される理由を説明するために、属性を追加することもできます。

重要・ *その場合、XML コードが適切に書かれていることを再確認してください。不適切な場合には、すべてのフィルターが失敗します。多くの場合、インターネット エクスプローラーで* Filters.xml *ファイルを開いて、不適切に書かれた XML コードを確認することができます。<Filters>、<Include>、および <Exclude> 要素は縮小および展開が可能です。これらが不可能な場合、コードにエラーが無いか確認してください。*

<Exclude> または <Include> 要素にサブ要素を追加した場合、それらが誤ってコメントアウトされたセクションに配置されていないことを確認してください。InstallShield は Filters.xml *ファイルのコメントアウト部分を無視します。*

次のサンプル XML コードで、Filters.xml ファイルの形式を説明します。

```
<Filters>
 <Include>
  <!--この要素にファイルを追加する方法
  -->
 <Include>
 <Exclude>
  <!--この要素にファイルを追加する方法
  -->
  <Registry key="HKEY_CLASSES_ROOT¥Interface¥{00020404-0000-0000-C000-000000000046}"/>
  <File name="12520437.cpx" path="[SystemFolder]" wrp="4.0-10.0" />
  <File name="12520850.cpx" path="[SystemFolder]" wrp="4.0-10.0" />
 </Exclude>
</Filters>
```

ターゲット システムの構成

すべてのインストールは、何らかの方法でターゲット システムを変更します。簡単なインストールでは、ファイルをコピーするだけのものもあります。より複雑なインストールでは、レジストリの変更、ファイルの関連付けを作成、.ini ファイルの編集、ショートカットの作成、ODBC リソースの構成、環境変数の使用、および Windows サービスのインストールと開始が行われます。ターゲット システムの構成方法に関する詳しい情報は、ドキュメントのこのセクションを参照してください。

ショートカットおよびプログラム フォルダーの作成

ショートカットとプログラム フォルダーを使用すると、インストールしたアプリケーションにすばやくアクセスできます。インストールでショートカットとプログラム フォルダーがデスクトップ、[スタート] メニュー、およびその他さまざまな場所に作成されるようにインストールを構成することができます。ショートカットとフォルダーは、それらが所属する機能がインストールに選択された場合にのみ、ターゲット システムに作成されます。

ショートカットの種類

InstallShield では、いくつかのタイプのショートカットが提供されています。

テーブル 4-1 ・ ショートカットの種類

ショートカットの種類	説明
新しいショートカット	プロジェクトに存在するファイルへの標準のショートカットを作成します。
新しいアドバタイズ ショートカット	アドバタイズ ショートカットの作成機能のファイルはエンドユーザーがショートカットを起動するまでターゲット システムにインストールされません。 ***メモ:*** *アドバタイズ ショートカットは、ポータブル実行可能ファイルとして指定されたファイルにのみ作成することができます。[オプション] ダイアログ ボックスの [ファイル拡張子] タブで、ポータブル実行可能ファイルとして扱うファイルを指定することができます。*
既存ファイルへの新しいショートカット	ターゲット システムに既に存在するファイルへのショートカットを作成します。たとえば、Internet Explorer または UNC パスへのショートカットを作成するときに利用できます。
新しいフォルダー	プログラム フォルダーを作成します。たとえば、会社名の下にショートカットを表示したい場合は、プログラム フォルダーを作成することができます。
新しいアンインストール ショートカット	製品のアンインストール プロセスを自動的に開始するショートカットを作成します。

ショートカットの作成

ショートカットを作成する前に、まずショートカットがポイントするファイルを含むコンポーネントを作成する必要があります。

タスク ***新しいショートカットを作成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストで **[ショートカット/フォルダー]** をクリックします。
2. **[ショートカット]** エクスプローラーで、インストール先ディレクトリの 1 つを右クリックして、適切なコマンドをクリックします。使用可能なコマンドの一覧は、「ショートカットの種類」をご覧ください。

   新しいショートカットが、**新しいショートカットN** というデフォルト名で追加されます (ここで *N* は連続番号です)。
3. 新しい名前を入力するか、または名前を後で右クリックしてから **[名前の変更]** を選択して新しい名前を付けます。
4. ショートカットの設定を構成します。

ショートカットとフォルダーについて構成可能な各設定についての詳細は、以下を参照してください:

- ショートカットの設定
- フォルダーの設定

メモ: *たとえば、会社名の下にショートカットを表示したい場合は、プログラム フォルダーを作成することができます。ショートカットのフォルダーを作成した後、そのフォルダーを右クリックし、[新しいショートカット] を選択して、ショートカットを作成できます。*

動的にリンクされたファイルへのショートカットを作成することはできません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。

ショートカットのアイコンを指定する

InstallShield では、実行時にターゲット システムで作成されるショートカットに使用されるアイコンを指定することができます。

タスク ***ショートカットのアイコンを指定するには、以下の手順に従います。***

1. **[ターゲット システムの構成]** の下にあるビュー リストで **[ショートカット/フォルダー]** をクリックします。
2. **[ショートカット]** エクスプローラーで、アイコンを指定するショートカットをクリックします。ショートカットの設定が、右のペインに表示されます。
3. ***"アイコン ファイル"*** 設定で、作成するショートカットのアイコンが含まれているファイルを指定します。アイコン リソースを含む .ico ファイル、または実行可能ファイル (.dll または .exe) を指定する必要があります。アイコンを含むファイルの完全修飾パスを入力するか、または省略記号 (...) ボタンをクリックして参照します。

4. 指定したアイコン ファイルに 1 つ以上のアイコン リソースがある場合、**"アイコン インデックス"** 設定にインデックスを入力します。

   負の数以外の整数を指定すると、実行可能ファイルのアイコン リソースの順番が参照されます。たとえば、0 はファイル内の最初のアイコン、1 は 2 番目のアイコン、2 は 3 番目のアイコンを参照します。

ショートカットがターゲット システム上に既に存在するファイルのためのものでない場合、[ショートカット] エクスプローラーでそのショートカットに表示されるアイコンが指定したアイコンに変更されます。

既存ファイルへのショートカットの場合、アイコン ファイルは実行時まで判明しません。したがって、[ショートカット] エクスプローラーでは、ターゲット システム上で実行時に使用されるアイコンの代わりに、各ショートカットに次のアイコンが表示されます。

"アイコン ファイル" 設定で選択されたファイルがアイコンを含まない場合、このアイコンが [ショートカット] エクスプローラーのショートカットに使用されます。

プロジェクト: *Windows Installer では、機能がアドバタイズされるときに別のアイコンが必要なため、InstallShield は指定した実行ファイルからアイコンを取り出します。*

ヒント: *[ショートカット] ビューのアイコンを右クリックしてから、[ショートカット アイコンの変更] をクリックしてショートカット アイコンを変更することもできます。"アイコン ファイル" 設定と "アイコン インデックス" 設定の値が、この方法で指定された値で更新されます。*

ソース メディアにショートカットを置く

タスク ***ソース メディアに機能のショートカットを配置するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストから **[機能]** をクリックします。
2. **[機能]** エクスプローラーで、機能をクリックします。
3. **"リモート インストール"** プロパティを **[ソースを優先]** に設定します。
4. 非圧縮 CD-ROM としてリリースをビルド します。
5. 機能のアドバタイズ ショートカットを作成します。

これらの設定で、機能に含まれるすべてのファイルは CD に残り、ローカルマシンには*インストールされません*。エンドユーザーがアドバタイズ ショートカットを起動すると、Windows Installer が起動し (エンドユーザーへは表示されません)、ResolveSource アクションが実行されます。このアクションは SourceDir の値を判断します。ビルドは非圧縮のため、SourceDir は CD ドライブへのパスを含み、CD 上でファイルへ解決します。

ショートカットにアクセスできるキーボード ショートカットを指定する

キーボード ショートカット (ホット キーとも呼ばれます) を使って、マウスを使う代わりに CTRL+ALT+A などのキーの組み合わせを押さえることで、素早く処理を行うことができます。キーボード ショートカットを製品のショートカットに割り当てると、エンド ユーザーは適切なホット キーを押してショートカットを起動できます。

注意: *ターゲット システム上の既存のキーボード ショートカットと競合する可能性があるため、ショートカットのキーボード ショートカットを構成することは避けることをお推めします。*

タスク ***キーボード ショートカットをプロジェクト内のショートカットに割り当てるには、以下の手順に従います。***

1. **[ターゲット システムの構成]** の下にあるビュー リストで **[ショートカット/フォルダー]** をクリックします。
2. **[ショートカット]** エクスプローラーで、ホットキーを指定するショートカットを選択します。
3. **"ホット キー"** 設定で、省略記号ボタン (...) をクリックします。**[ホットキー]** ダイアログ ボックスが開きます。
4. このショートカットに使用するキーボード ショートカットを押します。
5. **[OK]** をクリックします。

"ホット キー" 設定に、押されたキーの組み合わせを示す適切な 10 進数の値が表示されます。

たとえば、キーの組み合わせが CTRL+ALT+A の場合、この設定には 1601 と表示されます。この数値は、CTRL の 16 進数の値 (200) と ALT の 16 真数の値 (400)、および論理 Or 演算子を組み合わせて取得されます。次に、この数値 (600) に A キーの 16 進数値 (41) を追加し、最後に 10 進数値に変換します。この例では、10 進数に変換される数値は 641 で、変換後は 1601 となります。

ショートカット名の変更

新しいショートカットを作成する、デフォルトの内部名で表示されます。この名前はエンドユーザーに表示されませんが、プロジェクトに関連した名前に変更することができます。

タスク ***ショートカットの名前を変更するには、次の操作を実行します。***

1. **[ターゲット システムの構成]** の下にあるビュー リストで **[ショートカット/フォルダー]** をクリックします。
2. **[ショートカット]** エクスプローラーで、名前を変更するショートカットを右クリックして、**[名前の変更]** をクリックします。
3. 新しい名前を入力します。

アンインストール ショートカットの作成

アンインストール用のショートカットを作成することは可能ですが、[プログラムの追加と削除] コントロール パネルを使用して製品をアンインストールする方法をお勧めします。アンインストール ショートカットの作成は必要しない場合、以下で手順を実行します。

タスク ***製品のアンインストールを自動的に開始するショートカットを作成するには、以下の手順に従います:***

1. **[システム構成]** の下のビュー リストにある **[ショートカット]** をクリックします。
2. **[ショートカット]** エクスプローラーで、アンインストール ショートカットを含めるフォルダーを右クリックして、**[新しいアンインストール ショートカット]** をクリックします。***"アンインストール"*** というデフォルト名で新しいショートカットが作成されます。
3. ショートカットの名前を入力します。機能の名前を変更するには、機能を右クリックして **[名前の変更]** をクリックします。
4. 必要に応じて設定を指定します。

スタート画面上のデスクトップ アプリのタイルの外観を構成する

Windows 8 からアプリケーション タイルのグリッドをスタート画面に表示できるようになりました。これは、今までのショートカットの一覧に取って代わるもので、ショートカットの代わりにタイルを配置します。InstallShield は、スタート画面上のデスクトップ アプリのタイルの外観をカスタマイズすることができます。次のタイル構成設定が使用できます:

- アプリケーション名を中サイズ (150x150) のタイルに含めるとき、明色または暗色のテキストを切り替える
- タイル背景色を選択
- カスタム タイル イメージ (小: 70x70、中:150x150) の使用オプション
- アプリケーション名を中サイズ タイルに表示または非表示を選択

[タイルの構成] ノードが **[ショートカット/フォルダー]** ビューに表示されます。すべての該当するタイル構成が一覧表示されます。

メモ: *ショートカットおよびタイルは関連していますが、これらは異なります。ショートカットを個別に構成して、異なるアイコンを使用する同じ .exe ファイルに対して複数のショートカットを作成することもできますが、1 つの .exe ファイルに対して 1 つのタイル構成のみ作成することが可能です。*

タスク ***スタート画面上のデスクトップ アプリのタイルの外観を構成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[ショートカット/フォルダー]** をクリックします。
2. **[タイルの構成]** を右クリックしてから、**[タイル構成の追加]** コンテキスト メニュー オプションを選択します。.

*メモ: この手順は、.exe ファイルがプロジェクトに追加されていることを前提とします。.Exe ファイルがプロジェクトに含まれていない場合、**[構成の追加]** コンテキスト メニュー オプションは無効になります。*

*ヒント: 別の方法として、スタート画面タイルを構成する .exe ファイルをターゲットとするショートカットを右クリックしてから、**[タイルの構成]** コンテキスト メニュー オプションを選択します。新しいタイル 構成が追加および選択されます。ショートカットに構成が既に含まれている場合、**[タイルの構成]** オプションは無効化されます。*

3. **[タイルの構成]** で使用可能な任意の設定を使って、スタート画面上のデスクトップ アプリのタイルをカスタマイズします。使用可能な設定についての詳細は、「[タイル構成] の設定」を参照してください。

レジストリの編集

Windows レジストリは、アプリケーションとオペレーション システムで使用される構成情報を含むシステム全体のデータベースです。このレジストリは、次を含むすべての情報を格納します。

- 会社名、製品名、バージョン番号等のアプリケーション情報
- アプリケーションの実行を可能にするパス情報
- エンドユーザーが、システム上にある他のアプリケーションを妨げることなく簡単にアプリケーションをアンインストールができるアンインストール情報
- アプリケーションによって作成される文書のためのシステム全体のファイル関連付け
- ライセンス情報
- ウィンドウの位置等のアプリケーション オプションのデフォルトの設定

キー、値名、および値

レジストリは [コンピューター] エクスプローラー (またはターゲットのオペレーティング システムによっては [マイ コンピューター] エクスプローラー) の下に階層化されている一連のキーで構成されます。[コンピューター] のすぐ下には、いくつかのルートキーがあります。インストールは、レジストリのどのキーにもキーと値を追加することができます。以下は、通常インストールによって影響を受けるルートキーです。

- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT

キーは、レジストリ内の名前が付けられた場所です。キーは、サブキー、値名、値のペア、およびデフォルト (名前が付いていない)値を含むことができます。値名と値のペアは、キーの下の2 つの部分からなるデータ構造です。値名はキーの下のストレージの値を見分けます。また、その値は値名に関連付けられた実際のデータです。値名が値に特定されていない場合、その値はそのキーのデフォルト値になります。各キーは、デフォルト (名前の付いていない)値を 1 つのみ持つことができます。

terms キーとサブキーはお互い関連していることに注意してください。レジストリでは、別のキーの下のキーを、レジストリ階層の別のキーに関連してそれをどう参照するかによって、サブキーまたはキーとして参照することもできます。

InstallShield プロジェクトとレジストリ

InstallShield の [レジストリ] ビューを利用して、エンド ユーザーのレジストリを変更する作業を簡単に行うことができます。Windows のレジストリ エディターとほぼ同じようにこのビューを使用して、キーおよび値を作成します。

すべてのレジストリ データは機能に関連付けられている必要があります。機能がインストールに選択されると、その機能に関連付けられたレジストリ データがターゲット システムにセットアップされます。

注意: *レジストリは Windows オペレーティング システムの重要な部分なので、レジストリをむやみに変更または削除することは避けてください。不可欠なレジストリ キーが変更されると、システムが機能しなくなることがあります。*

HKEY_LOCAL_MACHINE¥Software と HKEY_LOCAL_MACHINE¥Software¥Wow6432Node の違い

[レジストリ] ビューで HKEY_LOCAL_MACHINE¥SOFTWARE (32 ビット) ノードにレジストリ データを追加すると、Windows Installer はそのデータを 32 ビット版のターゲット システムの HKEY_LOCAL_MACHINE¥Software キーの下にインストールしますが、64 ビット版のターゲット システムの場合は HKEY_LOCAL_MACHINE¥Software¥Wow6432Node キーの下にインストールします。

レジストリ データを [レジストリ] ビューの HKEY_LOCAL_MACHINE¥SOFTWARE (64 ビット) ノードに追加すると、InstallShield はビルド時にプロジェクトに 64 ビット Windows Installer パッケージを作成します。この 64 ビット Windows Installer パッケージを 32 ビット版のターゲット システム上で実行することはできません。この状況下では、Windows Installer は 64 ビット版のターゲット システム上で HKEY_LOCAL_MACHINE¥Software キーの下にレジストリ データをインストールします。

したがって、64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。32 ビット版のインストールを作成している場合、SOFTWARE (64 ビット) ノードの下にデータを入力することは避けます。

詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

ヒント: *64 ビット システム上で 32 ビット アプリケーションによるレジストリの表示を確認するには、32 ビット バージョンのレジストリ エディター (SysWOW64 フォルダーの* regedit.exe *ファイル) を起動します。*

機能ごとにレジストリ エントリをフィルターする

[レジストリ] ビューには、機能一覧が含まれています。[機能] 一覧にはプロジェクト内での機能とサブ機能の階層構造が含まれています。[ビューフィルター] では、レジストリ データをビューに表示する機能を選択することができます。

機能の階層にサブ機能が含まれる場合、親機能を選択するとその機能のレジストリ エントリのみが表示されます。ここにサブ機能のレジストリ エントリは表示されません。

プロジェクトでのすべてのレジストリ エントリの表示

インストールのすべてのレジストリ エントリを表示するには、[機能] 一覧で [すべてのアプリケーション データ] オプションを選択します。

[すべてのアプリケーションデータ] でビューをフィルターして、レジストリキーと値を変更、名前の変更、または削除することができます。

[レジストリ] ビューの [インストール先コンピューターのレジストリ] ビュー ペインでレジストリ キーをクリックすると、InstallShield は、そのキーのすべてのレジストリ データを [レジストリ] ビューの右下のペインに表示します。

キーに値を設定していない場合、すべてのアプリケーション データ を選択してもレジストリ データは表示されません。

レジストリキーの作成

タスク ***機能のインストール時にレジストリ キーがターゲット システムで作成されるように指定するには、次の手順を実行します。***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、新しいキーに関連付ける機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインで、その下に新しいキーを作成するレジストリ キーをクリックします。
4. **[インストール先コンピューターのレジストリ]** ビュー ペインで、レジストリ キーを右クリックして、**[新規作成]** をポイントして、**[キー]** をクリックします。

InstallShield が、**新しいキー -#***n* (ここで *n* は連続番号です) という名前で新しいキーを追加します。ここでキー名をわかりやすい名前に変更するか、または後でキーを右クリックして [名前の変更] を選択し、新しい名前を付けます。新しいキーが、空のデフォルト文字列値と共に作成されます。

値名とデータを変更するには、「レジストリ値の作成」を参照してください。

ヒント: *64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。*

レジストリ エントリをドラッグアンドドロップしてレジストリ キーを作成する

インストール プロジェクトにレジストリ エントリを追加する最も手軽な方法は、[レジストリ] ビューの [ソース] ペインからレジストリ エントリのひとつをドラッグし、[インストール先] ペインにドロップする方法です。キー全体を **[インストール先コンピューターのレジストリ] ビュー** ペインにドロップすると、そのキーのすべてのサブキーと値が選択した機能に追加されます。

タスク ***ソース コンピューターからインストール先コンピューターにレジストリ エントリをドラッグアンドドロップするには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、新しいキーに関連付ける機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインで、その下に新しいキーを作成するレジストリ キーをクリックします。
4. **[ソース コンピューターのレジストリ] ビュー** ペインで、インストールに含めるキーを選択し、**[インストール先コンピューターのレジストリ] ビュー** ペインのインストール先フォルダーにドラッグします。キー全体を [インストール先] ペインにドロップすると、そのキーのすべてのサブキーと値がインストールに追加されます。

ヒント: *64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。*

コンテキスト メニューを使ってキーをドラッグ & ドロップする

コンテキスト メニューを使って、複数のキーや値を一度に移動することができます。レジストリ エントリを右クリックし、それをインストール先にドラッグし、コンテキスト メニューから [オプション] をクリックします。

テーブル 4-2 ・レジストリ エントリのコンテキスト メニューから使用できるコマンド

オプション	説明
すべてのキーと値	すべての選択されたキー、サブキーおよび値を追加します。
キーとその値のみ	選択されたキーとそのキーの値のみを追加します。サブキーは追加されません。
このキーのみ	選択されたキーのみ追加し、そのサブキーまたは値は追加されません。
キャンセル	変更を加えずにドラッグアンドドロップ操作を終了します。

64 ビット開発システム上で、ソースマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示する

InstallShield を 64 ビット開発システム上で使用する場合、InstallShield が表示する [レジストリ] ビューは、使用中のマシンのレジストリの 32 ビットおよび 64 ビット領域の両方を表示します:

- HKEY_LOCAL_MACHINE¥Software
- HKEY_LOCAL_MACHINE¥Software¥Wow6432Node

このサポートによって、プロジェクトのレジストリ データの変更を構成する際、これらのソース領域からこのビューのインストール先ペインの適切な領域にエントリーをドラッグ アンド ドロップすることが可能となります。

インストールで 32 ビット領域へのリダイレクトを行わなずに 64 ビット ターゲット システム上の 64 ビット領域にレジストリ データをインストールする場合、レジストリ データを [レジストリ] ビューのインストール先ペインにある HKEY_LOCAL_MACHINE¥SOFTWARE (64 ビット) ノードに配置しなくてはなりません。[レジストリ] ビューのソースペインから 64 ビット データを、ビュー内の 64 ビット以外のインストール先ペインにドラッグするだけでは、そのコンポーネントが 64 ビットであるとマークされません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

データを他のマシンからインポートする

ただし、ドラッグ アンド ドロップ操作は、レジストリ エントリがインストール開発システムに存在する場合にのみ有効です。別のマシンからレジストリ データを取得する場合、REG ファイルのインポート ウィザードを使って、データをインポートできます。

.reg ファイルからレジストリ データをインポートする

InstallShield では、別のインストール プロジェクトのレジストリ (.reg) ファイル、または InstallShield の外で作成した既存のレジストリ ファイルをインポートすることができます。

InstallShield は、Regedit のエクスポートによって作成された .reg ファイル、またはその形式を正確に従うファイルをインポートすることができます。InstallShield では、複数行のレジストリ値はサポートされていません。

タスク ***.reg ファイルからレジストリ データをインポートするには、以下の手順を実行します。***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、.reg ファイルをインポートする機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインで、レジストリ データを追加するレジストリ キーを右クリックしてから、**[REG ファイルのインポート]** をクリックします。レジストリのインポート ウィザードが開きます。
4. レジストリ インポート ウィザードの指示に従ってレジストリ データを追加します。

レジストリ データを機能に追加すると、関連付けられた機能がインストールされるとき、ターゲット システムにインストールされます。

レジストリ キーの削除

タスク *レジストリ キーを削除するには、以下の手順を実行します:*

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、レジストリ キーを含む機能を選択するか、**[すべてのアプリケーション データ]** を選択して、製品のすべてのレジストリ キーを表示します。
3. **[インストール先コンピューターのレジストリ]** ビュー ペインで、削除するレジストリ キーを右クリックして、**[削除]** をクリックします。

レジストリ値の作成

タスク *新しいレジストリ値を作成するには、以下の手順に従います:*

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、値を追加するレジストリ キーを含む機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインで、値を追加するキーを右クリックして、**[新規作成]** をポイントし、登録するデータの種類をクリックします。以下は、使用可能なレジストリ値の種類です。

テーブル 4-3 · レジストリ値の種類

オプション	説明
デフォルト値	キーのデフォルト値。
文字列値	規定の長さのテキスト文字列。
バイナリ値	この値は 16 進数値として解析され、格納されます。
DWORD 値	4 バイト (32 bit) の長さの数値で表示されるデータ。
複数文字列値	ヌル文字で終わる文字列配列、および 2 つの ヌル文字で終わる文字列配列としてフォーマットされた複数テキスト文字列。このコマンドを選択すると、[複数行文字列値] ダイアログ ボックスが起動します。
展開可能な文字列値	この値は展開可能な文字列として解析され、格納されます。MSDN (Microsoft Developer's Network) によると、展開可能な文字列レジストリ値は、環境変数への非展開参照を含むヌル文字で終わる文字列です (例、*%PATH%*)。

InstallShield が、**New Value-#***n* (ここで *n* は連続番号です) という名前で新しい値を追加します。ここで値名をわかりやすい名前に変更するか、または後で値を右クリックして [名前の変更] を選択し、新しい名前を付けます。

ヒント: 64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

レジストリ値データの変更

タスク ***レジストリ値のデータを変更するには、以下の手順に従います:***

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**レジストリ**] をクリックします。
2. [**機能**] 一覧で、変更するレジストリ データを含む機能を選択します。
3. [**インストール先コンピューターのレジストリ データ**] ペインで、変更する値をダブルクリックします。[**データの編集**] ダイアログ ボックスまたは [**複数行文字列値**] ダイアログ ボックスが開きます。
4. ダイアログ ボックスで情報をすべて入力し、[OK] をクリックします。

Windows Installer のプロパティをレジストリ値に使って、後で自分の製品で使用する情報を保存することができます。たとえば、ソフトウェアのインストール先を保存する場合は、レジストリ値に [INSTALLDIR] と入力します。詳細については、「Windows Installer プロパティ リファレンス」を参照してください。

メモ: 角カッコ ([]) を含む値を追加するには、各カッコの前に円記号 (¥) を入れて、それをカッコで括ります。これを省略すると、Windows Installer はこの値をプロパティと認識します。たとえば、レジストリに [stuff] と書き込む場合、値名に [¥[]stuff[¥]] を使用します。

レジストリ値の削除

タスク ***レジストリ値をプロジェクトから削除するには、以下の手順に従います:***

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**レジストリ**] をクリックします。
2. [**機能**] 一覧で、削除するレジストリ値を含む機能を選択します。
3. [**インストール先コンピューターのレジストリ**] **ビュー** ペインで、削除する値を含むレジストリ キーをクリックします。すべてのレジストリ値が、[**インストール先コンピューターのレジストリ データ**] ペインに一覧表示されます。
4. [**インストール先コンピューターのレジストリ データ**] ペインで、削除するレジストリ値を右クリックしてから [**削除**] をクリックします。

複数レジストリ文字列値を 1 行の文字列へ入力する

タスク *複数の文字列値を一行の文字列に入力するには、以下の手順に従います:*

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**レジストリ**] をクリックします。
2. [**機能**] 一覧で、追加するレジストリ値を含む機能を選択します。
3. [**インストール先コンピューターのレジストリ**] **ビュー** ペインで、追加する値を含めるレジストリ キーを右クリックして、[**新規作成**] をポイントして、[**複数行文字列値**] をクリックします。[**複数行文字列値**] ダイアログ ボックスが開きます。
4. 値情報を入力して、[OK] をクリックします。

このエントリの ″種類″ フィールドには REG_MULTI_SZ、″データ″ フィールドには ファイル名パス と表示されます。

レジストリ エントリで環境変数を参照する

REG_EXPAND_SZ 文字列値を使用すると、レジストリに格納されているパスに環境変数を使用できます。これらのエントリは、オペレーティング システムで環境変数として認識されるために、特別な形式を必要とします。レジストリに表示される REG_EXPAND_SZ 値の形式は、%TEMP% です。TEMP は、TEMP ディレクトリの標準環境変数です。

タスク *レジストリ エントリの環境変数を参照するには、以下の手順に従います:*

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**レジストリ**] をクリックします。
2. [**機能**] 一覧で、追加するレジストリ値を含む機能を選択します。
3. [**インストール先コンピューターのレジストリ**] **ビュー** ペインで、追加する値を含めるレジストリ キーを右クリックして、[**新規作成**] をポイントして、[**文字列値**] をクリックします。
4. [**インストール先コンピューターのレジストリ データ**] ペインで、値を右クリックして [**変更**] を選択します。
5. 値データを入力するには、まず # 記号で始め、それから環境変数を入力します。環境変数名は % マークで囲みます。例、**#%TEMP%**。

このエントリの ″種類″ フィールドには REG_EXPAND_SZ、″データ″ フィールドには %TEMP% と表示されます。

ヒント: *64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。*

レジストリ キーのアクセス許可を構成する

InstallShield で、ロックダウンされた環境で製品を実行するエンドユーザーのために、レジストリ キーを保護するための設定を構成することができます。レジストリ キーのアクセス許可を特定のグループとユーザーに割り当てることができます。たとえば、管理者グループに特定のレジストリ キーについての [読み取り]、[書き込み]、および [削除] アクセス許可を割り当てることができますが、別のグループのすべてにユーザーについては [読み取り] 許可のみ割り当てることができます。

タスク ***レジストリ キーのアクセス許可を構成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[インストール先コンピューターのレジストリ] ビュー** ペインでレジストリ キーを右クリックし、**[アクセス許可]** ボタンをクリックします。**[アクセス許可]** ダイアログ ボックスが開きます。
3. 必要に応じて、アクセス許可を追加/変更/削除します。詳細については、「レジストリ キーの [アクセス許可] ダイアログ ボックス」を参照してください。

プロジェクトの [一般情報] ビューにある "ロックダウンの設定方法" 設定の選択に従って、InstallShield は ISLockPermissions テーブルまたは LockPermissions テーブルのどちらかにアクセス許可データを追加します。詳細については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

Registry のプライマリ キーを指定する

Windows Installer では、Registry テーブルに追加する各レジストリ キーと値について、一意のプラマリ キーが必要です。レジストリ エントリの作成状況が目で確認できるように、InstallShield ではビルド時に、データベースの Registry テーブルの各エントリに一意の名前が割り当てられます。

カスタム アクションを作成するときに、エントリのプライマリ キーが必要になるときがあります。InstallShield の [レジストリ] ビューでは、レジストリ キーまたは値のプライマリ キーを指定できます。

タスク ***レジストリ キーまたは値のプライマリ キーを指定するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、追加するレジストリ値を含む機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインで、指定するプライマリ キーに関連付けられているレジストリ キーを右クリックし、**[MSI 値]** をクリックします。**[MSI 値]** ダイアログ ボックスが開きます。
4. キーの名前を入力します。プライマリ キーは Windows Installer 識別子でなければならないので、名前には文字、数字、下線 (_)、およびピリオド (.) だけを使用し、名前の最初には文字または下線を使用します。

値を指定しない場合、InstallShield によって、Registry テーブルのこのエントリに一意のプライマリ キーが生成されます。

レジストリ フラグ

レジストリ フラグを使用すると、レジストリ エントリのインストールとアンインストールを制御できます。レジストリ エントリは、所属先の機能がインストールされるとデフォルトでインストールされます。ターゲット システムで所属先の機能が削除されたときは、これらのレジストリ エントリも削除されます。製品をアンインストールした後もレジストリ エントリをターゲット システムに残しておく場合や、既に存在しない場合にのみレジストリ エントリを作成する場合には、そのキーのインストールフラグを設定する必要があります。

InstallShield では、インストール動作はサブキー レベルで設定されます。あるキー以下のすべての値には同じインストール動作およびアンインストール動作が設定されなければなりません。

キーのレジストリ フラグを変更するには、[レジストリ] ビューでプロジェクトのキーの 1 つを右クリックしてから、以下のテーブルにリストされている任意のコマンドをクリックします。

テーブル 4-4 ・ レジストリ フラグ

アイコン	コマンド	説明
	自動	これはすべてのレジストリキーのデフォルトのオプションです。キーが既に存在しない場合、インストールがそれを作成します。アンインストール中、キーが空白の場合は、それが削除されます。
	インストールのみ (+)	***プロジェクト・*** *このレジストリ フラグの種類は、Express プロジェクトで使用できます。* キーが存在しない場合は、作成されます。このキーが所属する機能がアンインストールされるとき、このキーはターゲット システムに残ります。 このオプションは、サブキーまたは値を含まないキーにのみ使用できます。
	すべてのキーをアンインストール (-)	***プロジェクト・*** *このレジストリ フラグの種類は、Express プロジェクトで使用できます。* このフラグが空のキーに割り当てられた場合、キーはインストール時に作成されません。このフラグが値を含むキーに割り当てられていて、そのキーがターゲット システムに存在しない場合、インストール時にそのキーと値が作成されます。どちらの場合においても、キー、すべてのサブキー、および値はアンインストール時に削除されます。これはサブキーや値がインストール後に追加された場合も同様です。

テーブル 4-4 ・ レジストリ フラグ (続き)

アイコン	コマンド	説明
	存在しない場合はインストール、存在する場合はアンインストール (*)	*プロジェクト・ このレジストリ フラグの種類は、Express プロジェクトで使用できます。* このオプションは、1 つの例外 (自動レジストリ フラグ) を除いて、デフォルトの動作に似ています。自動レジストリ フラグの場合、アンインストール中にそのキーが空白でないとき、それは削除されません。**[存在しない場合はインストール、存在する場合はアンインストール (*)]** フラグの場合、アンインストール中にレジストリ キーが存在するときは、そのサブキーまたは値が残されているかどうかに関わらず、キーが削除されます。

レジストリ キーのインストール/アンインストール動作を設定する

InstallShield では、レジストリ エントリのインストールとアンインストール動作はサブキー レベルで設定されます。

タスク ***レジストリ キーのインストール/アンインストール動作を設定するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下にあるビュー リストにある **[レジストリ]** をクリックします。
2. **[機能]** 一覧で、動作を設定するレジストリ キーに対応する機能を選択します。
3. **[インストール先コンピューターのレジストリ] ビュー** ペインでレジストリ キーを右クリックし、適切な動作をクリックします。

インストール/アンインストール動作のオプション

InstallShield では、インストールとアンインストール動作はサブキー レベルで設定されます。あるキー以下のすべての値には同じインストール動作およびアンインストール動作が設定されなければなりません。使用可能なオプションの一覧は、「レジストリ フラグ」をご覧ください。

ユーザーごとインストールでのレジストリ エントリの扱い方

現在のユーザーが HKEY_LOCAL_MACHINE の下にあるキーを変更するための十分な権限を持っているとはかぎりません。そのような場合、HKEY_CURRENT_USER の下にエントリを書きこむ必要があります。

[レジストリ] ビューで HKEY_USER_SELECTABLE を選択すると、インストールのタイプとユーザーのアクセス権限にしたがって、適切なレジストリ ハイブの下にエントリが作成されます。

- ユーザーごとのインストール(インストールがユーザーレベルのアクセス権限を持つユーザーによって実行されている場合)では、これらのエントリは HKEY_CURRENT_USER の下に作成されます。
- マシンごとのインストール (ALLUSERS がヌル以外で、ユーザーが管理者である場合) では、エントリは HKEY_LOCAL_MACHINE の下に書き込まれます。

[レジストリ] ビューのリフレッシュ

タスク *[レジストリ] ビューをリフレッシュするには、以下の手順を実行します。*

F12 キーを押します。

ファイル拡張子をアプリケーション アイコンに関連付ける

ファイルの関連付けは、特定の種類のファイルを開くのにどの製品を使用すべきかを Windows に指示するためのレジストリ設定です。たとえば、Windows では、テキスト ファイル (.txt) を開くと、通常 Notepad.exe が起動します。

システムで登録済みファイルを表示および変更するには、Windows エクスプローラーを開き、[ツール] メニューで、[フォルダー オプション] をクリックします。ファイルの関連付けの構成方法を見るには、[フォルダー オプション] ダイアログ ボックス の [ファイルの種類] タブを使用します。

似たような方法で、Windows エクスプローラーでファイルを右クリックしてから [プロパティ] をクリックして該当のファイルに関連付けられているアプリケーションを識別することもできます。

ファイルの関連付けは HKEY_LOCAL_MACHINE¥SOFTWARE¥Classes と HKEY_CURRENT_USER¥SOFTWARE¥Classes の両方に格納され、マージされたデータは HKEY_CLASSES_ROOT の下に表示されます。

インストール プロジェクトのファイルの関連付けを作成する

ベストプラクティス ガイドラインは、プロジェクトで作成または使用されるすべての表示タイプのファイルに対してファイルの関連付けを作成することを推奨しています。InstallShield の [ファイルの拡張子] ビューでは、インストール プロジェクトでファイルの関連付けを素早く簡単に作成することができます。エンドユーザーがファイルの関連付けを含む機能をインストールと、ファイルの関連付けはターゲット マシンに登録されます。エントリが、レジストリの適切な場所に作成され、ProgID を通してファイルの種類が アプリケーションにリンク付けされます。ProgID は、時折、ファイルの種類のアプリケーション ID またはタグ名と呼ばれ、アプリケーションを一意に識別し、オペレーティング システムが関連付けを確認できるようにします。

ファイル拡張子の関連付けを作成する

ファイルの関連付けを作成して、ファイル拡張子をインストール プロジェクトの実行可能 (.exe) ファイルにリンク付けることができます。ファイルの関連付けを作成する前に、.exe ファイルがインストール プロジェクトに含まれている必要がありますので注意してください。

ファイルの関連付けを作成するとき、動的にリンクされたファイル を使うことはできません。動的にリンクされたファイルは、動的にリンクされたフォルダーのファイル リストで一番最初に来るファイルがデフォルトで選択されるため、特定のファイルを選択することはできません。

タスク *ファイル拡張子の関連付けを作成するには、以下の手順に従います:*

1. [**ターゲット システムの構成**] の下にあるビュー リストにある [**ファイル拡張子**] をクリックします。
2. [**ファイルの拡張子**] エクスプローラーを右クリックし、[**新しい拡張子**] をクリックします。InstallShield が、**ext***n* (ここで *n* は連続番号です) というデフォルト名で新しいファイル拡張子を追加します。デフォルトの名前を置換するには、使用する拡張子をドットを付けずに入力します (たとえば、**.txt** ではなく **txt** と入力します)。
3. 拡張子の設定を構成します。

.ini ファイル データの変更

初期化 (.ini) ファイルは、スタートアップまたは実行時にアプリケーションと Windows システムがアプリケーションのオプションを指定するために使用する情報を含む特殊なテキスト ファイルです。Boot.ini や Wininit.ini などの .ini ファイルは、オペレーティング システムで使用されます。.ini ファイルには、プロファイル、構成、言語、フォント、およびデバイス ドライバー情報を格納することができます。

一般的規則として、アプリケーションの設定は .ini ファイルではなくレジストリに格納することが推奨されています。

.ini ファイルのフォーマット

.ini ファイルの情報は、各カッコで囲まれたセクション名の下の格納され、各アイテムは、以下のように keyname=value というフォーマットで表記されます。

[セクション名]
keyname1=value1
keyname2=value2
keyname3=value3

キー データには、文字列値または整数値を使用できます。

InstallShield プロジェクトと .ini ファイル

InstallShield の [INI ファイル変更] ビューでは、製品のインストール中、ターゲット システム上で行う .ini ファイルの変更を指定することができます。ターゲット システムにあるすべての .ini ファイルを編集することができますが、システム .ini ファイルの変更は推奨しません。

.ini ファイルの編集には 3 つの手順があります。

1. .ini ファイル リファレンスの作成するか、または .ini ファイルをインポートします。
2. .ini ファイルにセクションを追加します。
3. .ini ファイルにキーワードを追加します。

.ini ファイルの追加

プロジェクトに .ini ファイルを追加するとき、インストールにある .ini ファイルへのリファレンスが作成されます。.ini ファイルに関連付けられた機能がターゲット システムにインストールされると、インストールは [INI ファイルの変更] ビューで構成した設定に従って .ini ファイルを変更します。

タスク *.ini ファイルを追加するには、以下の手順に従います:*

1. [**INI ファイルの変更**] ビューを開きます。
2. [**INI ファイル**] エクスプローラーを右クリックして、[**INI ファイルの追加**] をクリックします。
3. ターゲット マシンで編集するファイル名と拡張子を使用して 新しい .ini ファイルの名前を変更します (例、**Boot.ini**)。
4. .ini ファイルのプロパティを編集します。

.ini ファイルへのリファレンスを作成したら、次の手順に進み、セクションを .ini ファイルに追加します。

既存の .ini ファイルをインポートする

タスク *コンピューターに存在する .ini ファイルをインポートするには、以下の手順に従います:*

1. [**INI ファイルの変更**] ビューを開きます。
2. [**INI ファイル**] エクスプローラーを右クリックして、[**INI ファイルのインポート**] をクリックします。[**開く**] ダイアログ ボックスが開きます。
3. 追加する .ini ファイルを参照し、選択したあと [**開く**] をクリックします。

.ini ファイルが [INI ファイル] エクスプローラーに追加されます。.ini ファイルには各セクションおよびキーワード値の組み合わせは、[INI ファイル] エクスプローラー内の個々のアイテムによって表されます。

.ini ファイルのセクションを指定する

編集する .ini ファイルを指定した後、ファイル内の変更するセクションを指定します。各 .ini ファイルは、1 つ以上のセクションに分割され、それぞれのセクションにはキーワードがあります。セクションは、[SectionName] のように角かっこ [ ] で囲まれて識別されます。

タスク *.ini ファイルのセクションを指定するには、次の操作を実行します。*

1. [**INI ファイルの変更**] ビューを開きます。
2. まだ .ini ファイルを追加、またはインポートしていない場合、それを行います。
3. [**INI ファイル**] エクスプローラーで、セクションを作成する .ini ファイルを右クリックして、[**セクションの追加**] を選択します。セクションが、フォルダー アイコンと共に [**INI ファイル**] エクスプローラーに追加されます。
4. 対象の .ini ファイルで編集するセクションの名前を提供し、この新しいセクションの名前を変更します。角かっこは必要ありません。

.ini ファイルにセクションを追加すると、キーワードを追加できます。

.ini ファイルのキーワードとその値を指定する

.ini ファイルのキーワードは、.ini ファイルの構造の最下位レベルです。キーワードには、アプリケーションの終了から次回の起動までの間に保持しておく必要のあるデータを格納します。

プロジェクトに .ini ファイルに追加してから 1 つ以上のセクションを設定した後、セクションにキーワードを追加して、キーワードのプロパティを構成できます。キーワードのプロパティには、キーワードの値のほかに実行するアクション (データ値の置換、既存のデータ値への追加など) が含まれます。

タスク ***.ini ファイルにキーワードを追加するには次の手順を実行します。***

1. **[INI ファイルの変更]** ビューを開きます。
2. **[INI ファイル]** エクスプローラーで、キーワードを含めるセクションを右クリックして、**[キーワードの追加]** をクリックします。キーワード アイテムが **[INI ファイル]** エクスプローラーに追加されます。
3. 変更するキーワードの名前を使用して、この新しいキーワードの名前を変更します。新しいエントリの場合、.ini ファイルに表示されるとおりに正確に名前を入力します。
4. キーワードのプロパティを編集します。

ODBC リソースの構成

システム構成のより複雑な領域の 1 つに、ODBC ドライバー、データソース名 (DSN)、トランスレーターの設定があります。ODBC リソースは、すべての必須属性とともにシステムに正しく登録されている必要があります。また、ドライバーおよびトランスレーターの場合、インストール .dll ファイルなどの必要なファイルをインストールする必要があります。このプロセスは、ODBC リソース ビューを使って単純化できます。このビューでは、開発システムにインストールされているドライバー、データ ソース、およびトランスレーターを選択できます。開発システムにないドライバーやデータ ソースをプロジェクトに追加することもできます。

ODBC リソースを含める

タスク ***インストールに ODBC リソースを含めるには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、**[ドライバーと DSN]** アイテム、および/または **[トランスレーター]** アイテムを展開します。
3. 含める ODBC リソースのチェック ボックスを選択します。

追加の ODBC リソースを含める

[ODBC リソース] ビューのリストに表示されていない ODBC ドライバーまたは DSN はインストールする必要がある場合があります。以下は、そのインストール方法の説明です。

開発システムにインストールされているトランスレーターのみが、[ODBC リソース] ビューのリストに表示されます。このビューのリストに表示されていないトランスレーターをプロジェクトに追加するには、まずトランスレーターを開発システムにインストールする必要があります。トランスレーターがインストールされると、[ODBC リソース] ビューでトランスレーターをプロジェクトに追加することができます。

タスク ***[ODBC リソース] ビューのリストに表示されていないドライバーを含めるには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、**[ドライバーと DSN]** を右クリックして、**[ドライバーの挿入]** をクリックします。InstallShield が、**NewDriver***n* (ここで *n* は連続番号です) という名前で新しいドライバーを追加します。この新しいドライバーのチェック ボックスはデフォルトで選択されています。
3. ドライバーの新しい名前を入力します。
4. **[ODBC の属性とプロパティ]** ペインで、ドライバーのプロパティを構成します。

タスク ***[ODBC リソース] ビューのリストに表示されていない DSN を含めるには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** エクスプローラーで、新しい DSN を含めるライバを右クリックして、**[新しいドライバー]** をクリックします。InstallShield が、**NewDataResource***n* (ここで *n* は連続番号です) という名前で新しい DSN を追加します。この新しい DSN のチェック ボックスはデフォルトで選択されています。
3. DSN の新しい名前を入力します。
4. **[ODBC の属性とプロパティ]** ペインで、DSN のプロパティを構成します。

ODBC リソースを機能に関連付ける

プロジェクトのほとんどのデータ同様に、ODBC リソースも機能に関連付ける必要があります。機能がターゲット システムにインストールされるとき、ODBC リソースは機能の一部としてインストールされます。リソースがインストールされる複数の機能に関連付けられている場合、リソースが一回のみインストールされます。

タスク ***ODBC リソースを機能に関連付けるには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、チェック ボックスが選択されているドライバー、DSN またはトランスレーターをクリックします。
3. **[関連付けられている機能]** ペインで、ODBC リソースを含める各機能のチェック ボックスを選択します(複数可)。

メモ: *デフォルトでは、ODBC リソースは、[常にインストール] 機能に関連付けられています。[ODBC リソース] エクスプローラーでは、少なくとも 1 つの機能を選択する必要があるので、まず別の機能を選択しない限り [常にインストール] チェック ボックスの選択を解除することはできません。ODBC リソースに関連付けられている*

機能が 1 つだけの場合、別の機能を少なくとも 1 つ選択しない限り、その機能の選択をクリアすることはできません。

ODBC リソースの属性を設定する

インストール プロジェクトに追加した ODBC ドライバーまたは DSN は、すべて属性を構成することができます。プロパティおよび許可されている属性と値に関する詳細については、ベンダーのマニュアルを参照してください。

タスク ***ODBC リソースの属性を設定するには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、属性を構成するドライバー、DSN またはトランスレーターをクリックします。リソースのチェック ボックスは既に選択されている必要があります。
3. **[ODBC の属性とプロパティ]** ペインで、リソースの設定を構成します。

新しい属性を ODBC リソースに追加する

InstallShield では、インストール プロジェクトに追加した ODBC ドライバーまたは DSN に属性を追加することができます。プロパティおよび許可されている属性と値に関する詳細については、ベンダーのマニュアルを参照してください。

トランスレーターに属性を追加することはできません。

タスク ***新しい属性をドライバーまたは DSN に追加するには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、新しい属性を含めるドライバー、または DSN をクリックします。リソースのチェック ボックスは既に選択されている必要があります。
3. **[ODBC の属性とプロパティ]** ペインで、グリッドの最後の行をクリックし、新しいプロパティと対応する値を追加します。

ODBC リソースから属性を削除する

タスク ***ODBC リソースから属性を削除するには、以下の手順に従います:***

1. **[ODBC リソース]** ビューを開きます。
2. **[ODBC リソース]** ペインで、削除する属性を含む ODBC リソースをクリックします。リソースのチェック ボックスは既に選択されている必要があります。
3. **[ODBC の属性とプロパティ]** ペインで、削除するプロパティを右クリックして、**[削除]** をクリックします。

対応する行が [ODBC の属性とプロパティ] ペインから削除されます。

環境変数を使用する

環境変数は、インストールと共にターゲット システム上に設定できる名前と値の組み合わせで、アプリケーションおよび実行されているその他のプログラムによってアクセスできます。環境変数はレジストリに格納されます。

[環境変数] ビューでは、環境変数の新規作成、既存の変数値の変更、変数の削除などを行うことができます。環境変数は、環境変数をプロジェクトに追加したときに設定したプロパティに従って、アプリケーションのインストール時に作成、削除または変更されます。

環境変数の設定

タスク ***新しい環境変数を作成するか、または既存の環境変数を変更するには、以下の手順に従います:***

1. **[環境変数]** ビューを開きます。
2. **[環境変数]** を右クリックして、**[環境変数の追加]** をクリックします。InstallShield が、**NewEnvironment***n* (ここで *n* は連続番号です) という名前で変数を追加します。
3. 変更、削除または作成する変数名を入力します。
4. 環境変数のプロパティを編集します。

Windows サービスのインストールおよび構成

エディション: *InstallShield Express Edition は、サービスのインストール中にサービスをインストール、およびアンインストール中にサービスを削除するサポートを含みます。また、オプションでインストール後にサービスを開始、システム起動時に毎回自動的にサービスを開始、または (サービス コントロール マネージャを通してサービスが要求されたときに) オンデマンドでサービスを開始するサポートが含まれています。*

InstallShield Premier および InstallShield は、サービスにさらなる柔軟性を提供します。これらのエディションをでは、インストールまたはアンインストール中にサービスを開始、停止、または削除することができます。これらのエディションではまた、Windows Installer 5 で利用可能な拡張サービス カスタマイズ オプションを構成することもできます。さらに、InstallShield Premier および InstallShield では、ターゲット システムに既存するサービスを構成することもできます。

Windows サービスは、ログイン済みのユーザーが居ない場合でも、Windows ベースのシステム上で様々なシステムタスクを管理するためにバックグラウンドで実行される実行可能ファイルです。サービスは実行可能ファイルですが、これはサービスとして設計されたものであり、任意の実行可能ファイルをサービスとして使用することはできません。Windows サービスは、システム起動時に毎回実行することも、必要に応じてオンデマンドで実行することも可能です。InstallShield を使って、新しい Windows サービスをインストール、または既存のサービスを構成することができます。Windows で提供されているサービス管理ツールを使って、システムにインストール済みのサービスを参照および構成できます。

[サービス] ビューを使って、インストール中にインストール、およびアンインストール中に削除するサービスの情報を指定します。

タスク *サービスをインストールするには、以下の手順に従います:*

1. サービス実行可能ファイルをプロジェクトに追加します。ファイルの追加に関する情報は、「ファイルとフォルダーをプロジェクトに追加する」を参照してください。

   Windows Installer はドライバー サービスをサポートしていないので、サービスは単一実行可能ファイル (.exe) である必要があります。

   また、サービスの機能の "リモート インストール" 設定は [ローカルを優先] でなくてはなりません。詳細については、「機能の "リモート インストール" 設定を設定する」を参照してください。

2. **[ターゲット システムの構成]** の下にあるビュー リストにある **[サービス]** をクリックします。

3. **[サービス]** ノードを右クリックして、**[サービスの追加]** をクリックします。**[インストール先ファイルの参照]** ダイアログ ボックスが開きます。

4. 手順 1 でプロジェクトに追加した実行可能ファイルを参照します。新しいサービスが追加されます。

5. サービスの新しい名前を入力するか、後で F2 キーを押して名前を変更します。

   ここで入力する名前は、サービスの [プロパティ] ダイアログ ボックスに表示される名前と一致しなくてはなりません。(インストール済みサービスのプロパティにアクセスするには: [サービス] 管理ツールで、サービスを右クリックしてから [プロパティ] を選択します。)

6. 追加したサービスを選択してから、右側のペインに表示される設定を必要に応じて構成します。各設定についての詳細は、「[サービス] ビュー」を参照してください。

メモ: *サービスの設定を構成するにあたっては、サービスの技術的な詳細についてよく知っておく必要があります。*

ユーザーごとのインストールとマシンごとのインストールの違い

2 つの Windows Installer プロパティおよび現在のユーザーの権限によって、製品のショートカットやレジストリ エントリなどの構成情報がターゲット マシン上で All Users プロファイルまたは現在のユーザーのプロファイルのどちらに格納されるべきかが決定されます。

- ALLUSERS によって、構成が格納される場所が判別されます。
- MSIINSTALLPERUSER は、Windows Installer によって、パッケージが現在のユーザーに対してのみインストールされることを示します。

  MSIINSTALLPERUSER プロパティは、Windows Installer 5 および Windows 7、または Windows Server 2008 R2 で使用できます。以前のバージョンの Windows Installer と Windows は、このプロパティを無視します。

ALLUSERS、MSIINSTALLPERUSER、および Windows 7 または Windows Server 2008 R2

ALLUSERS プロパティが 1 に設定されていて、**MSIINSTALLPERUSER** に 1 が設定されている場合、Windows Installer は、ユーザーごとにインストールを実行します。

マシンごとのインストール時、Windows Installer は昇格された権限を必要とし、ファイルとレジストリ エントリがマシンごとの場所に配置します。ユーザー アカウント制御 (UAC) がターゲット システムで提供されている場合、マシンごとのインストールでは、通常、ユーザーの権限レベルに応じて、同意または認証情報を求めるプロンプトが表示されます。ユーザーごとのインストールでは、Windows Installer によって認証情報を求めるプロンプトは表示されず、ファイルとレジストリ エントリはユーザーごとの場所に配置します。

詳しい情報は、MSDN Web サイトの「Single Package Authoring」を参照してください。

Windows Vista 以降における ALLUSERS の効果

スクリプト内実行設定がシステム コンテキストで遅延実行になっているカスタム アクションは、Windows で LocalSystem アカウントに与えられた権限を使ってアクションを実行するときに使用されます。これは、Windows Installer がシステム コンテキストで実行されるためです。システム コンテキストで遅延としてマークされていないアクションは、ユーザーを偽装して実行され、インストールを起動するユーザーが所有する権限を持ちます。

ユーザーごとのインストール (ALLUSERS が設定されていないインストール) が実行されると、"システム コンテキストで遅延" のアクションは、通常の遅延または即時カスタム アクションと同じコンテキストで (ユーザーを偽装して) 実行されます。これにより、次のような状況で、カスタム アクションの実行時に問題が起きる可能性があります。

- Windows Installer インストールを起動するユーザーが管理者ではないとき、またはユーザーがインストールを Windows Vista 以降で実行していて、ユーザーが管理者グループに属し、ユーザーがデフォルトで管理者権限を持たないとき。
- カスタム アクションは、マシンのマシンごとの場所でリソース (Program Files フォルダーにあるファイル、HKEY_LOCAL_MACHINE のレジストリ キーまたは値) を変更しようと試みるとき。

これは、Windows XP または Windows の初期のバージョンでは問題にはならない可能性がありますが、デフォルトで Windows Vista 以降では、ユーザーは完全管理者権限は与えられません。したがって、ALLUSERS が設定されていない場合、"システム コンテキストで遅延" のアクションがユーザー偽装で実行されると、カスタム アクションは失敗する可能性があります。

この動作を防ぐには、マシンごとのインストールを常に、プロパティ マネージャーで ALLUSERS を設定して実行することをお勧めします。マシンごとのインストールは通常、ユーザーごとのインストールよりも管理が簡単です。

ALLUSERS のデフォルト値

ALLUSERS プロパティは、デフォルトで 1 に設定されています。インストールを構成して、管理者権限を使わないでユーザーごとにインストールできるようにする場合、ALLUSERS プロパティの値を変更したほうが良い場合があります。

タスク ***ALLUSERS の値を構成するには、以下の手順に従います:***

1. **[セットアップの編成]** の下にあるビュー リストにある **[一般情報]** をクリックします。
2. **ALLUSERS** 設定で、適切な値を選択します。

使用可能なオプションについては、「一般情報の設定」を参照してください。

[インストールの準備] ダイアログ と [ユーザー情報] ダイアログのデフォルト コントロール

″All Users オプションの表示″ 設定を使って、エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供するかどうかを指定します。この設定は、[ダイアログ] ビューで [ダイアログ] エクスプローラーをクリックしたときに表示されます。以下は、″All Users オプションの表示″ 設定に選択できるオプションです:

- いいえ—エンド ユーザーが製品をインストールする方法を指定できるオプションを表示しません。
- はい (Windows 7 以降のみ)—ターゲット システムに Windows 7 または Windows Server 2008 R2 がインストールされている場合、[インストールの準備] ダイアログにボタンが追加されます。これらのボタンを使って、エンド ユーザーは製品をインストールする方法を指定できます。昇格された権限が必要な場合、[すべてのユーザー] ボタンにシールド アイコンが含まれます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 2 に、MSIINSTALLPERUSER プロパティが 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 1 に設定され、MSIINSTALLPERUSER プロパティは設定されません。
- はい (すべてのシステム)—ターゲット システムに Windows 7 または Windows Server 2008 R2 がインストールされている場合、[インストールの準備] ダイアログにボタンが追加されます。これらのボタンを使って、エンド ユーザーは製品をインストールする方法を指定できます。昇格された権限が必要な場合、[すべてのユーザー] ボタンにシールド アイコンが含まれます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 2 に、MSIINSTALLPERUSER プロパティが 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 1 に設定され、MSIINSTALLPERUSER プロパティは設定されません。

  ターゲット システムに Windows Vista 以前、または Windows Server 2008 以前がインストールされている場合、InstallShield は [ユーザー情報] ダイアログにラジオボタンを追加します。これらのラジオ ボタンを使って、エンド ユーザーはプロジェクトをインストールする方法を指定できます。エンド ユーザーが [すべてのユーザー] ラジオ ボタンを選択した場合で、エンド ユーザーが昇格された権限を持つ場合、ALLUSERS プロパティは 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合で、エンド ユーザーが昇格された権限を持つ場合、ALLUSERS プロパティは空白文字列 (″″) に設定されます。

  デフォルト値は [いいえ] です。

インストール動作のカスタマイズ

インストール作成の重要な要素は、それをエンドユーザーのニーズに合わせてカスタマイズすることです。「インストール動作のカスタマイズ」のヘルプ トピックで、インストールの機能を拡張するのに役立つ InstallShield の様々な機能について説明されています。たとえば、Windows Installer が直接サポートしていない機能を追加するカスタム アクションを作成するのに役立ちます。プロジェクトのインストール動作をカスタマイズする方法についての詳細は、ドキュメントのこのセクションを参照して下さい。

カスタム アクションを使用する

Windows Installer には、インストールを実行する多くの標準アクションがあります。ただし、インストールで拡張機能が必要になる場合があります。そのような場合、カスタム アクションを使用すると標準アクションの機能を拡張できます。これはダイナミック リンク ライブラリ、実行可能ファイル、またはスクリプトをインストールに含めることによって行うことができます。次のテーブルは、プロジェクトに追加できる異なるカスタム アクションの種類について説明します。

テーブル 4-1 ・カスタム アクションの種類

アクションの種類	Description
Windows Installer DLL (MSI DLL)	この種類カスタムアクションは、固定プロトタイプを使って関数をエクスポートする C 関数の呼び出しが可能なライブラリです。 この種類のアクションの利点は、以下のとおりです: ・ Windows Installer DLL は、通常、最小限の依存関係を含む。 ・ ほとんどの Windows 関数がシステム ライブラリで露出される。 ・ 実行可能ファイル アクションとは異なり、この種類のアクションは Windows Installer API を使って実行中の Windows Installer セッションにアクセスをして、プロパティ、ディレクトリ、その他を使った作業を行う。 MSI DLL を作成するには、C/C++ および Windows API コードの知識を必要とします。

テーブル 4-1・カスタム アクションの種類 (続き)

アクションの種類	Description
標準 DLL	この種類のアクションは InstallShield の拡張で、C 関数の呼び出しが可能な DLL から関数を呼び出すことができます。アクションに選択する関数プロトタイプの種類 (クラシックまたは新規) によって、関数に使用するシグネチャが決まります。 ・ **クラシック**—この種類の DLL のプロトタイプは、InstallShield Express の以前のバージョンで使用されたフォーマットと同じフォーマットを使用しなくてはなりません。 ・ **新規**—この種類の DLL のプロトタイプは、MSI DLL カスタム アクションが必要とするシグネチャと同じシグネチャを使用します。プロジェクトに追加する新しいカスタム アクションにこの種類のシグネチャを使用する場合、標準 DLL よりも柔軟なスケジュール オプションが提供されている MSI DLL カスタム アクションを使用することをお勧めします。 ***重要・*** *標準 DLL カスタム アクションの場合、ゼロ以外の戻り値は成功、ゼロは失敗を意味します。MSI DLL カスタム アクションの場合、戻り値ゼロは成功、ゼロ以外の戻り値は失敗またはキャンセルを示します。*
実行可能ファイル	この種類のアクションはインストールに含まれている (一時サポートファイル、または製品とともにインストールされる) またはターゲットシステムに既存する実行可能ファイルを起動します。 この種類のアクションの利点は、実行可能ファイルで多くのオペレーティング システムの機能が露出されている点です。たとえば、この種類のアクションを使ってテキスト ファイルやその他の一般的な種類のファイルを開いたり、既存のディレクトリのアクセス許可を設定したり、バッチ ファイルを起動したりできます。 実行可能ファイル アクションは、実行中のインストール セッションにアクセスすることができません。したがって、(コマンドライン引数を除いて) Windows Installer プロパティを実行可能ファイルに渡したり、実行可能ファイルから戻したりはできません (レジストリやファイルなどの外部ストレージを除く)。 ドキュメントを開く、またはバッチ ファイルを起動するとき、その種類のドキュメントを処理する実行可能ファイルを明示的に参照しなくてはなりません。

テーブル 4-1・カスタム アクションの種類 (続き)

アクションの種類	Description
VBScript または JScript	この種類のアクションは VBScript または JScript コードを実行します。 この種類のアクションの利点は、以下のとおりです: ・ スクリプト アクションは、多くの場合簡単に実装できます。 ・ 実行可能ファイル アクションとは異なり、この種類のアクションは Windows Installer API を使って実行中の Windows Installer セッションにアクセスをして、プロパティ、ディレクトリ、その他を使った作業を行う。 ・ この種類のアクションは、文字列の操作といったシンプルなタスクに便利です。 VBScript および JScript アクションがターゲット システムにアクセスするとき、または変更を加えるときにウィルス対策/スパイウェア対策プログラムがトリガされることがよくありますので、ご注意ください。

エディション: *InstallShield Premier および InstallShield では、Express Edition で提供されていない、いくつかのカスタム アクションのサポートが提供されています。これらの追加カスタム アクションを利用して、InstallScript コードの実行、プロパティの設定、ディレクトリの設定、マネージ アセンブリ内でのパブリック メソッドの呼び出し、または特定の条件下でエラーメッセージを表示してインストールを中止することが可能です。*

Windows Installer DLL カスタム アクション

インストールで InstallShield または Windows Installer サービスではサポートされていないアクションを実行する必要がある場合、Windows Installer .dll ファイルからエントリ ポイント関数を呼び出すカスタム アクションを作成することができます。頻繁に *MSI DLL アクション*と呼ばれるこの種類のカスタム アクションを使う場合、関数を次のシグネチャで定義しなくてはなりません。

UINT _stdcall *FunctionName* (MSIHANDLE hInstall) {...}

関数名には異なる名前を使用できますが、戻り値の型、呼び出し規則、および単一のパラメーターには、前述のシグネチャで指定された型を使用する必要があります。関数シグネチャで使用される MSIHANDLE データ型は、実行中のインストールへのハンドルです。

以下は、**MyFunctionName** と呼ばれる関数のサンプル コードです。

```
UINT _stdcall MyFunctionName (MSIHANDLE hInstall)
{
    MessageBox(
        GetForegroundWindow( ),
        TEXT("これは MyFunctionName"),
        TEXT("カスタム アクション"),
        MB_OK | MB_ICONINFORMATION);
    return ERROR_SUCCESS;
}
```

エントリ ポイント関数は、アクションが正しく終了したことを示す ERROR_SUCCESS を戻します。カスタム アクションが 0 以外の値を戻したとき、カスタム アクションの "終了コードを無視する" 設定で [いいえ] が選択されている場合は、インストールが終了します。カスタム アクションの "終了コードを無視する" 設定で [はい] が選択されている場合、カスタム アクションの戻り値にかかわらずインストールは続行します。

MSI DLL の準備ができたら、カスタム アクションを設計して、エントリポイントの関数を呼び出す必要があります。カスタム アクションを作成する方法については、「MSI DLL カスタム アクションをプロジェクトに追加する」を参照してください。

要件

- Header: Msiquery.h で宣言
- Library: Msi.lib を使用

Msiquery.h と Msi.lib はいずれも、Windows Installer SDK に含まれています。Windows Installer SDK は マイクロソフトの Web サイトからダウンロードできます。上記の 2 つのファイルのほかに、Windows Installer SDK には Windows Installer API についてのマイクロソフトによる文書の決定版も含まれています。

MSI DLL カスタム アクションをプロジェクトに追加する

[カスタム アクション] ビューの中のアイテムは、インストールまたはアンインストール中に起動されるタイミングに従って時系列で編成されています。カスタム アクションをプロジェクトに追加する際、アクションを適切なインストールまたはアンインストール アイテムに追加してカスタム アクションの起動されるタイミングを指定します。

タスク ***MSI DLL カスタム アクションをプロジェクトに追加するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**カスタム アクション**] をクリックします。
2. [**インストール中のカスタム アクション**] エクスプローラー、[**メンテナンス中のカスタム アクション**] エクスプローラー、または [**アンインストール中のカスタム アクション**] エクスプローラーで、作成するカスタム アクションを含めるアイテムをクリックします。
3. アイテムを右クリックして、[**新しい DLL**] を選択します。InstallShield が、**NewCustomAction***n* (ここで *n* は連続番号です) という名前で新しいカスタム アクションを追加します。
4. カスタム アクションの名前を入力します。
5. カスタム アクションの設定を構成します。

MSI DLL カスタム アクションの設定を構成する

MSI DLL ファイル カスタム アクションをプロジェクトに追加すると、その設定を構成する必要があります。

タスク *プロジェクトで MSI DLL カスタム アクションの設定を構成するには、以下の手順に従います:*

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**カスタム アクション**] をクリックします。
2. [**インストール中のカスタム アクション**] エクスプローラー、[**メンテナンス中のカスタム アクション**] エクスプローラー、または [**アンインストール中のカスタム アクション**] エクスプローラーで、構成するカスタム アクションを含めるアイテムをクリックします。
3. 右側に表示されているグリッド内の設定を構成します。

DLL カスタム アクション

インストールで InstallShield または Windows Installer サービスではサポートされていないアクションを実行する必要がある場合、.dll ファイルから関数を呼び出すカスタム アクションを作成することができます。この .dll ファイルにより、シリアル番号を確認するなどの必要に応じた機能を実行できます。

DLL 関数を呼び出すカスタム アクションを作成する際の最初の手順は、DLL を書き込むことです。カスタム アクションを書き込む方法として、2 つの関数プロトタイプを使用できます。

- クラシック DLL カスタム アクション関数プロトタイプ—この種類の DLL のプロトタイプは、InstallShield Express の以前のバージョンで使用されたフォーマットと同じフォーマットを使用しなくてはなりません。
- 新規 DLL カスタム アクション関数プロトタイプ—この種類の DLL のプロトタイプは、MSI DLL カスタム アクションが必要とするシグネチャと同じシグネチャを使用します。プロジェクトに追加する新しいカスタム アクションにこの種類のシグネチャを使用する場合、標準 DLL よりも柔軟なスケジュール オプションが提供されている MSI DLL カスタム アクションを使用することをお勧めします。

DLL の準備ができたら、カスタム アクションを設計して、エントリポイントの関数を呼び出す必要があります。カスタム アクションを作成する方法については、「DLL カスタム アクションをプロジェクトに追加する」を参照してください。

Classic DLL カスタム アクション関数プロトタイプ

InstallShield では、カスタム アクションの結果として呼び出される DLL で、エントリ ポイント関数の正確なプロトタイプが必要です。

Revenera では、Windows プログラミングや DLL デバッグに関する技術的なサポートは提供していません。DLL 関数は、製作者が正しく作成する責任があります。カスタム DLL 関数のプロトタイプは以下のように作成します。返される種類、またはパラメーターの種類や数が異なると、カスタム アクションが失敗する原因となります。

```
LONG WINAPI Foo(HWND, LPTSTR , LPTSTR, LPTSTR, LPTSTR);
```

InstallShield は関数プロトタイプを使用して、以下の情報を DLL に渡します。

1. パラメーター 1 は、インストールのウィンドウ ハンドルを渡します。この関数は常に NULL を返します。
2. パラメーター 2 は、ソース ディレクトリ [SRCDIR] を渡します。
3. パラメーター 3 は、サポート ディレクトリ [SUPPORTDIR] を渡します。
4. パラメーター 4 は、メイン ターゲット ディレクトリ [INSTALLDIR] を渡します。
5. パラメーター 5 は、データベース ディレクトリ [DATABASEDIR] を渡します。

カスタム アクションのプロトタイプを作成して、[カスタマー情報] 実行時ダイアログに入力されるシリアル番号を処理する場合、パラメーター 4 はシリアル番号になります。

DLL 関数の本文は、必要に応じて自在に作成できます。当然ながら、インストールにより関数に渡される値は有益に活用することができます。

DLL 関数は、ルーチンの終了を通知する状態フラグとして、LONG 型の値を返す必要があります。関数が 0 を戻したとき、カスタム アクションの "終了コードを無視する" 設定で [いいえ] が選択されている場合は、インストールが終了します。関数がその他の値を戻したとき、カスタム アクションの "終了コードを無視する" 設定で [はい] が選択されている場合は、インストールが続行します。

サンプル DLL 関数

このセクションでは、サンプルのソースコードを示します。このサンプルには、関数 Foo() があります。これは、関数に渡された値を示すメッセージボックスを表示します。

```
#include <windows.h>
#ifdef _cplusplus
extern "C" {
#endif

//Foo() 関数の定義
LONG WINAPI Foo(HWND hwnd, LPSTR szSrcDir, LPSTR szSupport, LPSTR szInst, LPSTR szDbase)
  { CHAR szTmp[1024];
  int ret;

  //文字列を構成し、Foo() に渡された値を表示します。
  wsprintf(szTmp, "クステンションが呼び出されました。 hwnd=%x szSrcDir=%s szSupport=%s szInst=%s.次のいずれかを選択してください" ¥
            "終了しますか?", hwnd, szSrcDir, szSupport, szInst);

  //メッセージ ボックスに文字列を表示します。
  ret=MessageBox(hwnd, szTmp, "テスト エクステンション", MB_YESNO);
  if (ret==IDYES)
    //0 が返されると、インストールは終了します。
    return(0);
  else
    //0 以外が返されると、インストールは続行します。
    return(1);
}

#ifdef _cplusplus
}
#endif
```

関数によってゼロ以外の値が戻されると、インストールは続行します。ゼロが戻されると、インストールが終了します。

上記の DLL 関数を呼び出せるようにするには、DLL を作成し関数を正しくエクスポートするときに、定義 (.def) ファイルを含める必要があります。次の定義ファイルをプロジェクトに含めます。LIBRARY の後の名前は、DLL に指定した名前です。

```
; mydll.def : DLL のモジュール パラメーターを宣言します。.

LIBRARY  MYDLL
DESCRIPTION 'サンプル Windows ダイナミック リンク ライブラリ'
```

```
EXPORTS
   Foo @1
```

新しい DLL カスタム アクション関数プロトタイプ

新しい関数シグネチャを使用すると、現在実行中の .msi データベースのハンドルを取得することができます。データベースのハンドルを取得すると、任意の Windows Installer API を呼び出すことができます。以下の例は ProductName プロパティ の値の読み出し方法を説明します。

```
UINT _stdcall Action(MSIHANDLE hInstall)
 {
   TCHAR buffer[32] = {0};
   DWORD dWord = 32;
   MsiGetProperty(hInstall, TEXT("製品名"), buffer, &dWord);
   MessageBox(0,buffer,TEXT("製品名を表示"), MB_OK);
   return 1;
 }
```

関数はゼロ以外の値を戻して、アクションが正しく終了したこと、並びにインストールが続行することを示します。関数が数値 0 を戻したとき、カスタム アクションの "終了コードを無視する" 設定で [いいえ] が選択されている場合は、インストールが終了します。カスタム アクションの "終了コードを無視する" 設定で [はい] が選択されている場合、カスタム アクションの戻り値にかかわらずインストールは続行します。

要件

- Header: Msiquery.h で宣言
- Library: Msi.lib を使用

Msiquery.h と Msi.lib はいずれも、Windows Installer SDK に含まれています。Windows Installer SDK は マイクロソフトの Web サイトからダウンロードできます。上記の 2 つのファイルのほかに、Windows Installer SDK には Windows Installer API についてのマイクロソフトによる文書の決定版も含まれています。

DLL カスタム アクションをプロジェクトに追加する

[カスタム アクション] ビューの中のアイテムは、インストールまたはアンインストール中に起動されるタイミングに従って時系列で編成されています。カスタム アクションをプロジェクトに追加する際、アクションを適切なインストールまたはアンインストール アイテムに追加してカスタム アクションの起動されるタイミングを指定します。

タスク ***DLL カスタム アクションをプロジェクトに追加するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[カスタム アクション]** をクリックします。
2. **[インストール中のカスタム アクション]** エクスプローラー、**[メンテナンス中のカスタム アクション]** エクスプローラー、または **[アンインストール中のカスタム アクション]** エクスプローラーで、作成するカスタム アクションを含めるアイテムをクリックします。
3. アイテムを右クリックして、**[新しい DLL]** を選択します。InstallShield が、**NewCustomAction***n* (ここで *n* は連続番号です) という名前で新しいカスタム アクションを追加します。

4. カスタム アクションの名前を入力します。
5. カスタム アクションの設定を構成します。

DLL カスタム アクションの設定を構成する

DLL ファイル カスタム アクションをプロジェクトに追加すると、その設定を構成する必要があります。

タスク *プロジェクトの DLL カスタム アクションの設定を構成するには、以下の手順に従います:*

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**カスタム アクション**] をクリックします。
2. [**インストール中のカスタム アクション**] エクスプローラー、[**メンテナンス中のカスタム アクション**] エクスプローラー、または [**アンインストール中のカスタム アクション**] エクスプローラーで、構成するカスタム アクションを含めるアイテムをクリックします。
3. 右側に表示されているグリッド内の設定を構成します。

実行可能ファイル カスタム アクション

別のベンダーのインストールを再パッケージできないために、サード パーティー インストールの起動が必要な場合があります。または、インストール実行中に .avi の実行が必要な場合があります。そのような場合は、実行可能ファイルを起動するカスタム アクションを作成できます。

サポートされているカスタム アクション タイプのいずれかにより、インストールで実行可能ファイルが起動されます。この種類のカスタム アクションを作成する方法については、「.exe カスタム アクションをプロジェクトに追加する」を参照してください。

.exe カスタム アクションをプロジェクトに追加する

[カスタム アクション] ビューの中のアイテムは、インストールまたはアンインストール中に起動されるタイミングに従って時系列で編成されています。カスタム アクションをプロジェクトに追加する際、アクションを適切なインストールまたはアンインストール アイテムに追加してカスタム アクションの起動されるタイミングを指定します。

タスク *.exe カスタム アクションをプロジェクトに追加するには、以下の手順に従います:*

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**カスタム アクション**] をクリックします。
2. [**インストール中のカスタム アクション**] エクスプローラー、[**メンテナンス中のカスタム アクション**] エクスプローラー、または [**アンインストール中のカスタム アクション**] エクスプローラーで、作成するカスタム アクションを含めるアイテムをクリックします。
3. アイテムを右クリックして、[**新しい EXE**] を選択します。InstallShield が、**NewCustomAction***n* (ここで *n* は連続番号です) という名前で新しいカスタム アクションを追加します。
4. カスタム アクションの名前を入力します。

5. カスタム アクションの設定を構成します。

.exe カスタム アクションの設定を構成する

.exe ファイル カスタム アクションをプロジェクトに追加すると、その設定を構成する必要があります。

タスク ***プロジェクトの .exe カスタム アクションの設定を構成するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**カスタム アクション**] をクリックします。
2. [**インストール中のカスタム アクション**] エクスプローラー、[**メンテナンス中のカスタム アクション**] エクスプローラー、または [**アンインストール中のカスタム アクション**] エクスプローラーで、構成するカスタム アクションをクリックします。
3. 右側に表示されているグリッド内の設定を構成します。

VBScript および JScript カスタム アクション

インストール要件によっては、既存の VBScript ファイル (.vbs) または JScript (.js) を呼び出すカスタム アクションを作成する必要がある場合があります。この種類のカスタム アクションを作成する方法については、「VBScript または JScript カスタム アクションをプロジェクトに追加する」を参照してください。

VBScript カスタム アクションの例

INSTALLDIR の値を決定し、ファイルが存在するかどうかによってその値を変更する、VBScript カスタム アクションの例を以下に示します。

```
' INSTALLDIR の値を取得
Dim sInstalldir
sInstalldir = Session.Property("INSTALLDIR")

' 表示する。
MsgBox sInstalldir

' ファイルの存在を確認
Dim pFs
Set pFs = CreateObject("Scripting.FileSystemObject")
Dim sSomeFile
sSomeFile = sInstalldir & "31337.txt"
If pFs.FileExists(sSomeFile) Then
  ' 存在する場合、INSTALLDIR を変更
  Session.Property("INSTALLDIR") = sInstalldir & "New"
End If
```

VBScript または JScript カスタム アクションをプロジェクトに追加する

[カスタム アクション] ビューの中のアイテムは、インストール、メンテナンス、またはアンインストール中に起動されるタイミングに従って時系列で編成されています。カスタム アクションをプロジェクトに追加する際、アクションを適切なインストール、メンテナンス、またはアンインストール アイテムに追加してカスタム アクションの起動されるタイミングを指定します。

タスク ***VBScript または JScript カスタム アクションをプロジェクトに追加するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[カスタム アクション]** をクリックします。
2. **[インストール中のカスタム アクション]** エクスプローラー、**[メンテナンス中のカスタム アクション]** エクスプローラー、または **[アンインストール中のカスタム アクション]** エクスプローラーで、作成するカスタム アクションを含めるアイテムをクリックします。
3. アイテムを右クリックして、**[新しい VBScript]** または **[新しい Jscript]** を選択します。InstallShield が、**NewCustomAction***n* (ここで *n* は連続番号です) という名前で新しいカスタム アクションを追加します。
4. カスタム アクションの名前を入力します。
5. カスタム アクションの設定を構成します。

VBScript または JScript カスタム アクションの設定を構成する

VBScript または JScript カスタム アクションをプロジェクトに追加すると、その設定を構成する必要があります。

タスク ***プロジェクトの VBScript または JScript カスタム アクションの設定を構成するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[カスタム アクション]** をクリックします。
2. **[インストール中のカスタム アクション]** エクスプローラー、または **[アンインストール中のカスタム アクション]** エクスプローラーで、構成するカスタム アクションをクリックします。
3. 右側に表示されているグリッド内の設定を構成します。

アクションの実行のオプション

カスタム アクションは、シーケンスに表示されている順に実行されます。一部のアクションは、インストールの開始時に実行する必要があり、その他のアクションは遅延する必要があります。[カスタム アクション] ビューにある "スクリプト内実行" 設定を使って、シーケンス内のどの反復処理 (遅延、ロールバック、またはコミット) によってアクションがトリガーされるのかを選択します。この設定を使って、遅延、ロールバック、およびコミット アクションのコンテキストを指定することもできます: アクションは、インストールを実行中のユーザーの権限を使ったユーザー コンテキスト、または昇格された権限を使ったシステム コンテキストで実行できます。[カスタム アクション] ビュー内の特定のカスタム アクションに対してこの設定が表示されない場合、そのアクションは即時実行としてスケジュールされます。即時実行カスタム アクションは、常にユーザー コンテキストで実行します。

即時実行

名前のとおり、即時実行カスタム アクションは、内部 Windows Installer インストール スクリプトがコンパイルされるときに実行します。.msi ファイルが起動すると、Windows Installer サービスはインストールデータベースのすべてのテーブルを内部スクリプトに変換します。このスクリプトは、インストールのすべてのアクションを表示順に循環することによってビルドされます。このスクリプトのビルドはすぐに実行されます。即時実行として設定されているアクションに遭遇すると、そのアクションが実行されます。したがってこのアクションはファイル転送が起きる前に起動します。インストールのエンドユーザー インターフェイスが完全にロードする前に起動する可能性もあります。

一般に、「即時実行」にスケジューリングされているカスタム アクションはターゲット システムの変更は行わず、プロパティのみを設定してターゲット システムを照会 (たとえば、ターゲット システムが製品のシステム要件に合っているか確認するなど) します。Windows Installer のプロパティを設定するカスタム アクションと、[ユーザー インターフェイス] シーケンスで発生するカスタム アクションはすぐに実行されるようにスケジューリングする必要があります。

このタイプのアクションはシステム変更が行われる前に起動するため、インストールにインストールされているファイルを使用できません。

遅延実行

[遅延実行] は、Windows Installer が、[即時実行] の実行中に作成された内部スクリプトが実行されるときに行われます。このスクリプトが完全に生成されると、Windows Installer は新しくコンパイルされたスクリプトを実行します。スクリプトはシーケンス内のすべてのアクションを順番に起動します。ただし、アクションをすぐに実行するようスケジューリングした場合は、そのアクションは遅延実行時に再度起動しません。

遅延実行中に起動したアクションは、インストールの一部としてインストールされているファイルにアクセスできます。したがって、インストールのこの段階で製品とともにインストールされた DLL ファイルから関数を呼び出すカスタム アクションを呼び出すことができます。ただし、遅延実行カスタム アクションを正しく実行するには、InstallInitialize と InstallFinalize の間で行う必要があります。

ターゲット システムにインストールされているファイルを使用するカスタム アクションや、既に実行されたその他のシステム変更を使用するカスタム アクションは、遅延実行を行うようスケジューリングする必要があります。

ロールバック実行

ロールバックは、インストールにエラーが発生した時や、エンドユーザーがインストール完了前にキャンセルした時に実行されます。ロールバック実行オプションを使用すると、ロールバック中にのみアクションが実行されるよう設定できます。したがって、ロールバック実行できるアクションは、遅延実行アクションのようにインストールスクリプトに書き込まれます。遅延実行アクションと異なり、ロールバック実行アクションはロールバック中にしか実行されません。(ロールバック カスタム アクションは、インストールが遅延実行中に失敗した場合にのみ実行されます。)

インストール中にターゲット システムに変更を加えるカスタム アクションは、ロールバック中にロールバック実行カスタム アクションでやり直されます。たとえば、ファイルを作成するカスタム アクションがある場合、ファイルを削除する 2 つ目のカスタム アクションを作成して、2 番目のアクションをロールバック実行にスケジューリングします。(ロールバックカスタム アクションは、元に戻すカスタム アクションの前にスケジューリングしてください。)

コミット実行

コミット実行は、InstallFinalize アクションが正常に完了するまで実行されません。つまり、インストールがファイル転送、COM サーバー登録、およびショートカットとレジストリ エントリの作成を完了するまで行われません。それから、コミット実行に設定されているアクションはシーケンスで表示されている順番に実行されます。

たとえば、一時ファイルを作成するカスタム アクションがある場合、ファイルを削除する 2 つ目のカスタム アクションを作成して、そのアクションをコミット実行にスケジューリングします。

システム コンテキストでの遅延、ロールバック、またはコミット実行

遅延実行アクション同様、システム コンテキストでの遅延実行アクションも、Windows Installer によって生成されたスクリプトが実行されるまで起動しません。ただし、このタイプのアクションはユーザー偽装を実行しません。

同様に、システム コンテキストでのロールバック実行およびシステム コンテキストでのコミット実行アクションは、ユーザー偽装なしで実行します。

Windows Vista 以降のシステムで、昇格された権限が必要なシステム変更を行う場合、遅延カスタム アクションはシステム コンテキストで実行しなくてはなりません。Windows Vista 以降のシステムで管理者がインストールを実行する場合でも、ユーザー コンテキストでの遅延カスタム アクションは、昇格した権限を持たない可能性があるため、昇格された権限を必要とするカスタム アクションをユーザー コンテキストでの遅延アクションとするのは避けるようにしてください。

メモ: *システム コンテキストで実行するカスタム アクションは、一般的にネットワーク リソースにアクセスできません。*

カスタム アクションを起動するタイミングの変更

[カスタム アクション] ビューの中のアイテムは、インストール、メンテナンス、またはアンインストール中に起動されるタイミングに従って時系列で編成されています。カスタム アクションをプロジェクトに追加する際、アクションを適切なインストール、メンテナンス、またはアンインストール アイテムに追加してカスタム アクションの起動されるタイミングを指定します。

タスク ***カスタム アクション起動のタイミングを変更するには、以下の手順に従います:***

- カスタム アクションを現在の位置から新しい位置にドラッグします。
- カスタム アクションを右クリックして、**[上へ移動]** または **[下へ移動]** を選択します。
- カスタム アクションをクリックしたあと、CTRL+シフト+上矢印、または CTRL+シフト+下矢印を押します。

[カスタム アクション] ビューは以下の主要なランタイムのカテゴリーに分かれています:

- **インストール中のカスタム アクション**—このランタイム カテゴリーにスケジュールされているアクションは、製品の初回インストール中に起動します。
- **メンテナンス中のカスタム アクション**—このランタイム カテゴリーにスケジュールされているアクションは、変更および修復モード中に起動します。また、これらのカスタム アクションは、製品がサイレントでアンインストールされるときにも起動します。

- **アンインストール中のカスタム アクション**—このランタイム カテゴリーにスケジュールされているアクションは、製品のアンインストール中に起動します。

シリアル番号を検証するカスタム アクションを使用する

シリアル番号の検証には、カスタム DLL ファイルを使用する必要があります。この種類の DLL ファイルのコード例は以下の場所にあります:

InstallShield Program Files フォルダー¥Samples¥WindowsInstaller¥ValidateSerialNumber

この例では、シリアル番号のフォーマットは次のようになります。

Field1-1505-XXXXXXXXXX

最初のセクションは上記のように表示される必要があります。ただし、大文字と小文字の区別はありません。2 番目のセクション (上記例では 1505) は、5 で割り切れる値でなければなりません (たとえば、1111 は無効です)。最後のセクションは 10 桁の任意の英数字です。ただし、このフォーマットは、あくまでも例にすぎません。シリアル番号関数には、任意のフォーマットが使用できます。

入力文字のフィルター

不正な文字の入力をフィルターするには、[ダイアログ] ビューの [ユーザー情報] ダイアログにある "シリアル番号テンプレート" フィールドに **??????-####-??????????** と入力します。疑問符 (?) は英数字を示し、ナンバー記号 (#) は数字を示します。この方法でテンプレートを設定すると、シリアル番号のフィールドは 3 つの部分に分かれ、各セクションで異なる種類の入力をフィルターできます。

その他の考慮事項

関数の検証プロパティ、正常時戻り値プロパティ、および再試行回数制限プロパティでは、シリアル番号の検証 DLL ファイルの機能をさらにカスタマイズすることができます。"検証関数" プロパティでは、エンドユーザーから提供されたシリアル番号を検証する .dll 内の関数の名前を入力します。

.dll は、実行されるたびに特定の値を戻します。InstallShield に含まれているサンプル .dll は、成功時に 1 を戻し、失敗時に -1 を戻します。したがって、"正常時戻り値" フィールドには **1** を入力します。自分で .dll ファイルを作成する場合、正常時戻り値には、ゼロ以外のすべての値を選択することができます。

"再試行回数制限" プロパティでは、無効なシリアル番号を入力できる回数を設定します。たとえば、シリアル番号を **3** 回まで入力できるようにする場合は、このフィールドに 3 と入力します。無効なシリアル番号が 3 回入力されると、インストールは終了します。

サンプル .dll ファイルには、入力されたシリアル番号を分解してユーザーに表示するユーザー インターフェイス要素が含まれています。さらにこの要素は、シリアル番号に不正な部分がある場合にはその部分も表示します。この表示は、シリアル番号検証の .dll ファイルをテストおよびトラブルシューティングするときに役立ちますが、最終版のインストールには残さないようにします。

メモ: *InstallShield は、シリアル番号 DLL ファイル カスタム アクションをユーザー定義のカスタム アクションの後にシーケンスします。つまり、[ユーザー情報] ダイアログの後に独自のカスタム アクションを追加すると、インストールはこのカスタム アクションを、シリアル番号 DLL ファイル カスタム アクションの前に起動します。*

カスタム アクション ギャラリー

[カスタム アクション] ギャラリーには、既に構成済みの一般的な機能性を持つカスタム アクションが含まれています。このギャラリーのカスタム アクションには、何の入力も必要ありません。これらは、インストール プロジェクトに追加され、インストール中に実行されるようにデザインされています。ギャラリーから作成中のインストールへアクションを挿入するには、[カスタム アクション] ビューにあるアイテムの 1 つを右クリックして、[ギャラリーから追加] をポイントし、プロジェクトに挿入するカスタム アクションを選択します。

現在、このギャラリーにリストされているアクションは、[最後に再起動をスケジュールする] アクションのみです。このアクションをプロジェクトに追加すると、インストールの Setup Complete Success ダイアログの後へ自動的に追加されます。このアクションにより、インストールの完了後ターゲットマシンは再起動し、.dll ファイルやその他のファイルを起動時にシステムに登録できます。このカスタム アクションの設定を変更すること、またはインストールの別の部分へスケジュールすることはお勧めしません。たとえばスケジュールを変更すると、インストールがファイル転送の完了前にターゲットマシンを再起動する原因となります。

製品がインストールされているかどうかの確認をする

次のコードは、ProductCode {8FC71000-88A0-4B41-82B8-8905D4AA904C} がマシンにインストールされているかどうかを確認します。このコードは、New Style DLL カスタム アクションとしても使用できます。

```
UINT _stdcall CheckProduct(MSIHANDLE hInstall)
{
    int RetVal = 0;
    RetVal = MsiQueryProductState("{8FC71000-88A0-4B41-82B8-8905D4AA904C}");
        if (RetVal==5))
        {
            MessageBox(GetForegroundWindow(),TEXT("インストールされています"),TEXT("My Product"), MB_OK);
            return 1;
        }
        else
        {
            MessageBox(GetForegroundWindow(),TEXT("インストールされていません"),TEXT("My Product"), MB_OK);
            return 0;
        }
}
```

以下は MsiQueryProductState の戻り値の一覧です。

テーブル 4-2 • MsiQueryProductState の戻り値

戻り値	説明
-1	製品はアドバタイズもインストールもされていません
1	製品はアドバタイズされていますが、インストールされていません
2	製品は別のユーザーにインストールされています。
5	製品は現在のユーザーにインストールされています。

セットアップ ファイルを使用する

セットアップ ファイル (サポート ファイルとも呼ばれます) は、アプリケーションのインストール処理中の時だけターゲット システムで使用できるファイルです。サポート ファイルはインストールが始まるとターゲット システムの一時ディレクトリにコピーされ、インストールが完了すると削除されます。サポートディレクトリ (SUPPORTDIR) はダイナミックファイルの場所を表し、ターゲット システムごとに異なります。また、同じシステムでもインストールするたびに異なります。

[セットアップ ファイル] ビューを使うと、インストール時にのみターゲット システム上で使用できるようにしたいファイルを追加および削除することができます。

インストール中に特定のセットアップ ファイルにアクセスするには、サポート ディレクトリ (SUPPORTDIR) の値をクエリしてから、SUPPORTDIR 値の後にファイル名を追加して、ファイルの完全パスを取得します。

セットアップ ファイルの追加

タスク ***セットアップ ファイルをインストール プロジェクトに追加するには、以下の手順に従います:***

1. **[セットアップ要件およびアクションの定義]** の下にあるビューリストで、**[セットアップ ファイル]** をクリックします。
2. オプションで、ファイルに 1 つ以上のサブフォルダーを作成することもできます。その場合、**[サポート ファイル]** エクスプローラーで、[英語 (U.S)] 領域 (プロジェクトが英語以外の場合は、該当する言語領域) を右クリックしてから、**[新しいフォルダー]** をクリックします。新しいフォルダーが追加されます。フォルダーに使用する名前を入力します。
3. **[サポート ファイル]** エクスプローラーで、追加するサポート ファイルが必要なアイテムをクリックします。たとえば、プロジェクトが英語以外の言語の場合、英語 (US) アイテム、または他の言語のアイテムにセットアップ ファイルを追加することができます。
4. **[ファイル]** ペインで右クリックして **[ファイルの挿入]** をクリックします。**[開く]** ダイアログ ボックスが開きます。
5. 含めるファイルを参照します。複数のファイルを選択するには、ファイルをクリックしながら CTRL キーを押します。
6. **[OK]** をクリックします。

InstallShield は、ファイルを [ファイル] ペインに追加します。

ライセンス ファイルの追加

[サポート ファイル] ビューで、使用許諾契約書を含むテキスト ファイル (例、License.txt) を追加できます。

タスク *インストール プロジェクトにライセンスを追加するには、以下の手順に従います:*

1. **[セットアップ要件およびアクションの定義]** の下にあるビューリストで、**[セットアップ ファイル]** をクリックします。
2. オプションで、ライセンスに 1 つ以上のサブフォルダーを作成することもできます。その場合、**[サポート ファイル]** エクスプローラーで、[英語 (U.S)] 領域 (プロジェクトが英語以外の場合は、該当する言語領域) を右クリックしてから、**[新しいフォルダー]** をクリックします。新しいフォルダーが追加されます。フォルダーに使用する名前を入力します。
3. **[サポート ファイル]** エクスプローラーで、ライセンス ファイルが必要なアイテムをクリックします。たとえば、プロジェクトが英語以外の言語の場合、英語 (US) アイテム、または他の言語のアイテムにライセンス ファイルを追加することができます。
4. **[ファイル]** ペインで右クリックして **[ファイルの挿入]** をクリックします。**[開く]** ダイアログ ボックスが開きます。
5. 含めるライセンス ファイルを参照します。複数のファイルを選択するには、ファイルをクリックしながら CTRL キーを押します。
6. **[OK]** をクリックします。

ライセンス ファイルが [ファイル] ペインに追加されます。

インストール中にセットアップ ファイルにアクセスする

インストール中に特定のセットアップ ファイルにアクセスするには、サポート ディレクトリ (SUPPORTDIR) の値をクエリしてから、SUPPORTDIR 値の後にファイル名を追加して、ファイルの完全パスを取得します。カスタム アクションとして実行したとき、次の VBScript は SUPPORTDIR のロケーションを提供します。

```
MsgBox (Session.Property("SUPPORTDIR"))
```

サポート ファイルの並べ替え

[セットアップ ファイル] ビューの [ファイル] ペインで、ファイルとフォルダーを並べ替えることができます。

タスク *[セットアップ ファイル] ビューでファイルとフォルダーを並べ替えるには、以下の手順に従います:*

1. **[サポート ファイル]** エクスプローラーで、並べ替えるサポート ファイルとフォルダーを含むアイテムをクリックします。
2. **[ファイル]** ペインで、その順番でソートする列の見出しをクリックします。

Disk1 ファイル

[セットアップ ファイル] ビューにある Disk1 機能を使用して、インストール メディアの Disk1 に入れるファイルとフォルダーを追加することができます。これらのファイルとフォルダーは、インストールの実行時、ターゲット システムに自動的にインストールはされません。代わりに、アプリケーションまたはインストール プログラムからインストール メディアにリンクさせることができます。

たとえば、エンドユーザーがアクセスできる大きい再配布可能ファイルをアプリケーションに含め、アプリケーションのインストールには含めないような場合です。このようなファイルは Disk1 フォルダーに入れます。

Disk1 ファイルは、InstallShield がメディアのサイズを計算するときに考慮されます。このため、Disk1 ファイルを追加しても、リリースのディスク サイズの仕様を超えることはありません。この機能は、インストール時にソース メディアからインストール プログラムを起動できるよう、ディスク イメージに再配布可能ファイルを格納する場合に便利です。

Disk1 ファイルの追加

タスク ***Disk1 ファイルをインストール プロジェクトに追加するには、以下の手順に従います:***

1. **[セットアップ要件およびアクションの定義]** の下にあるビューリストで、**[セットアップ ファイル]** をクリックします。
2. **[サポート ファイル]** エクスプローラーで、**Disk1** をクリックします。
3. **[ファイル]** ペインで右クリックして **[ファイルの挿入]** をクリックします。**[開く]** ダイアログが開きます。
4. Disk1 に追加するファイルを参照します。複数のファイルを選択するには、ファイルをクリックしながら CTRL キーを押します。
5. **[OK]** をクリックします。

InstallShield は、ファイルを [ファイル] ペインに追加します。

Disk1 フォルダーの追加

Disk1 フォルダーにはフォルダー全体およびそのコンテンツを追加することができます。プロジェクトをビルドしたとき、フォルダーとファイルがインストール メディアのルートに追加されます。

タスク ***Disk1 フォルダーをインストール プロジェクトに追加するには、以下の手順に従います:***

1. **[セットアップ要件およびアクションの定義]** の下にあるビューリストで、**[セットアップ ファイル]** をクリックします。
2. **[サポート ファイル]** エクスプローラーで、**Disk1** をクリックします。
3. **[ファイル]** ペインの任意の場所を右クリックし、**[フォルダーの挿入]** をクリックします。**[フォルダーの参照]** ダイアログ ボックスが開きます。
4. Disk1 に追加するフォルダーを参照します。
5. **[OK]** をクリックします。

フォルダーが [ファイル] ペインに追加されます。

Disk1 ファイルとフォルダーを削除する

タスク *Disk1からファイルを削除するには、以下の手順に従います:*

1. [**セットアップ要件およびアクションの定義**] の下にあるビューリストで、[**セットアップ ファイル**] をクリックします。
2. [**サポート ファイル**] エクスプローラーで、Disk1 をクリックします。
3. [**ファイル**] ペインで、削除するファイルまたはフォルダーを右クリックして、[**削除**] をクリックします。

セットアップ ファイルを削除する

タスク *プロジェクトからセットアップ ファイルを削除するには、以下の手順に従います:*

1. [**セットアップ要件およびアクションの定義**] の下にあるビューリストで、[**セットアップ ファイル**] をクリックします。
2. [**サポート ファイル**] エクスプローラーで、削除するサポート ファイルを含む言語固有のアイテムをクリックします。
3. [**ファイル**] ペインで、ファイルを右クリックし [**削除**] をクリックします。

サーバーの構成

インストールを作成しているとき、ターゲット システムにインストールされるテクノロジに対してサーバー側のサポートを提供する必要があることに気がつくことがあります。InstallShield では、サーバー側のインストールを簡単に構成することができます。新しいインターネット インフォメーション サービス (IIS) Web サイトの管理、および InstallShield で作成したインストールで COM+ アプリケーションとコンポーネントの管理を行うことができます。

COM+ アプリケーションとコンポーネントの管理

InstallShield の [コンポーネント サービス] ビューでは、インストール パッケージ用の COM+ アプリケーションとコンポーネントを管理することができます。

InstallShield のコンポーネント サービスに関する次の情報をお読みください。

- COM+ システム アプリケーション以外のみプロジェクトに追加することができます。したがって、InstallShield は [コンポーネント サービス] ビュー の [COM+ アプリケーション] エクスプローラーの下に、COM+ システム アプリケーション以外のみを表示します。
- ローカル マシンにインストールされている COM+ アプリケーションのみが、[コンポーネント サービス] ビューに表示され、プロジェクトに追加することが可能です。

[コンポーネント サービス] ビューの外観は、[コントロール パネル] の [コンポーネント サービス] 管理ツールに似ています。

エディション: *InstallShield Premier および InstallShield では、COM+ サーバー アプリケーションとアプリケーション プロキシ両方を管理することができる追加機能が [コンポーネント サービス] ビューで提供されています。COM+ アプリケーション プロキシはサーバー アプリケーション属性のサブセットで構成され、これはクライアント コンピューターからアプリケーションが存在するマシンへのリモート アクセスを可能にします。*

COM+ アプリケーションの追加

タスク ***COM+ アプリケーションをインストールに追加するには、以下の手順に従います:***

1. **[コンポーネント サービス]** ビューを開きます。
2. **[COM+ アプリケーション]** の下で、プロジェクトの追加するアプリケーションのチェック ボックスを選択します。タブ付きのプロパティ シートが開きます。
3. **[インストール]** プロパティシートで、サーバー インストールのパラメーターを設定します。またこのプロパティシートで、COM+ アプリケーションが所属する機能を選択します。
4. タブが付いている別のプロパティシートで標準 COM+ アプリケーションプロパティを設定します。これらのプロパティ シートは、[コントロール パネル] にある [コンポーネント サービス] 管理ツールから処理されるとき、各 COM+ アプリケーションに関連付けられたプロパティ ダイアログ ボックスのタブに対応しています。

COM+ アプリケーションの削除

タスク *COM+ アプリケーションをインストールから削除するには、以下の手順に従います:*

1. **[コンポーネント サービス]** ビューを開きます。
2. **[COM+ アプリケーション]** の下で、プロジェクトから削除するアプリケーションのチェック ボックスを選択します。

インターネット インフォメーション サービス

インターネットインフォメーションサービス (IIS) は Microsoft が開発した Web サーバーです。Web ベースのアプリケーションをビルドおよび配布、Web サイトの管理、およびインターネットまたはイントラネットへ情報をパブリッシュするための安定したプラットフォームを提供します。

InstallShield の [IIS 構成] ビューでは、新しい IIS Web サイト、アプリケーション、および仮想ディレクトリを作成および管理することができます。

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

InstallShield における IIS サポートのバージョン固有情報

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

以下は、特定のバージョンの IIS に関する情報です。

- IIS は Windows 2000 Server 以降および Windows XP 以降のシステムに含まれています。IIS 6 は Windows Server 2003 システムでのみ利用できます。IIS 7 は Windows Vista と Windows Server 2008 システムで提供されています。IIS 7.5 は Windows 7 と Windows Server 2008 R2 システムで提供されています。IIS は自動的にインストールされません (デフォルト)。
- [IIS 構成] ビューにある一部の Web サイトと仮想ディレクトリの設定は、特定のバージョンの IIS に適用します。これらの設定についてのバージョン固有の情報は、InstallShield のインライン ヘルプ ペインに表示されます。バージョン固有のプロパティが構成されていて、ターゲット システムに対応するバージョンの IIS がないとき、IIS はバージョン固有のプロパティを無視します。

  たとえば、IIS 7 と IIS 6 は、アプリケーションまたは仮想ディレクトリの "アプリケーション保護" プロパティをサポートしません。[IIS 構成] ビューで、このプロパティは、アプリケーションまたは仮想ディレクトリの [アプリケーションの設定] 領域にある設定を使って構成されます。[IIS 構成] ビューでこの設定を選択すると、右下に表示されるヘルプ ペインに、この設定が IIS 6 以降には適用しないことが示されます。この設定が選択されているときに、エンドユーザーが製品を IIS 6 以降があるターゲット システムにインストールすると、"アプリケーション保護" 設定は無視されます。
- Windows Vista 以降と Windows Server 2008 以降のシステムでは、インストール プロジェクトで [IIS 構成] ビューの "コンテンツのソース パス (ローカルまたは UNC)" 設定で指定された物理パスにある構成ファイルに個別の Web サイト、アプリケーション、および仮想ディレクトリの設定が保管されます。したがって、各 Web サイト、アプリケーション、または仮想ディレクトリは、一意の物理パスを持ちます。一意の物理パスを持たなかった場合、同じ物理パスを持つ 2 つの異なる仮想ディレクトリがあるとき、たとえばテスト環境などで予期しない動作に遭遇する可能性があります。

  たとえば、同じ物理パスがある 2 つの仮想ディレクトリがあり、ディレクトリの参照が片方のみで有効にされているとき、作成された 2 つ目の仮想ディレクトリの "ディレクトリの参照" 設定は、1 つ目の仮想ディレクトリの設定をオーバーライドします。
- Windows Server 2003 を持つシステムでは、IIS 6 がインストールされていない場合、他の IIS ディレクトリおよびサイトは上記にかかわらず作成されます。IIS 6 固有の設定はスキップされます。
- IIS 5.1 for Windows XP Professional は、一回につき 1 つの Web サイトのみサービス可能です。これは IIS 5.1 の制限によるものです。
- InstallShield はバージョン 5 およびそれ以降の IIS をサポートします。

IIS サポートの実行時要件

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含ま

れています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

InstallShield インストールの IIS サポートは、ターゲット マシンに IIS がインストールされていて、エンド ユーザーが管理者権限を持つ場合のみ動作します。

IIS の設定を含むパッケージのインストール実行中、InstallShield インストールはターゲットマシンに IIS があるかどうかを確認します。IIS がインストールされていない場合、インストールはエンドユーザーにダイアログを表示して IIS がインストールされていないことを通知します。ダイアログで、エンドユーザーは [中止]、[再試行]、[無視] のいずれかを選択することができます。

- エンドユーザーが [中止] を選択すると、インストールが終了します。
- エンドユーザーが IIS をインストールしてから、[再試行] を選択したとき、インストールは IIS の存在を再確認し、インストールを続行します。エンドユーザーが IIS をインストールしていないにもかかわらず [再試行] を選択したとき、インストールは IIS の存在を再確認し、ダイアログを再度表示します。
- エンドユーザーが [無視] を選択すると、インストールは続行しますが、IIS Web サイトおよび仮想ディレクトリは構成されません。

メモ: *InstallShield では、インストールを実行している ターゲット マシン以外のターゲット マシンでの Web サイトの作成はサポートされていません 。*

Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定する

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

サーバー側インクルード (SSI) ディレクティブは、コンテンツを Web ページに挿入するようにWeb サーバーに指示します。#exec タイプのディレクティブによって、Web サーバーは Web ページにシェル コマンドの出力を含めることができます。

IIS Web サーバーを構成して、#exec ディレクティブの CMD コマンドがシェル コマンドの実行に使用されるのを防いだり、CMD コマンドがこのタイプのコマンドの実行に使用されることを許可することができます。HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥W3SVC¥Parameters レジストリ キーの SSIEnableCmdDirective レジストリ値によって、CMD コマンドが許可されているかどうかが判別されます。

InstallShield では、インストール時に、ターゲット システム上で SSIEnableCmdDirective レジストリ値をどう構成するかを指定することができます。インストールで SSIEnableCmdDirective レジストリ値を変更しない場合、そのように指定することもできます。

セキュリティに関する懸念により、デフォルトの SSIEnableCmdDirective 値は FALSE (0) になっています。FALSE (0) 値により、エンドユーザーによって承認されていないサーバー側での実行可能ファイルの実行を防ぐことができます。

タスク ***Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. 中央のペインで、**[Web サイト]** エクスプローラーをクリックします。右側のペインにショートカットの設定が表示されます。
3. **"SSIEnableCmdDirective レジストリ値"** 設定で、適切なオプションを選択します。
   - **無視する** – ターゲット システム上の SSIEnableCmdDirective レジストリ値を変更しません。デフォルトでは、これが設定されています。
   - **FALSE (0)**–ターゲット システム上の SSIEnableCmdDirective レジストリ値を 0 に設定します。これにより、サーバー側インクルードの #exec CMD ディレクティブがシェル コマンドの実行に使用されるを防ぐことができます。この値を選択すると、IIS Web サーバーに #exec CMD ディレクティブに依存するアプリケーションが存在した場合、インストール プロジェクトの Web サイトおよび仮想ディレクトリがインストールされたあと、これらのアプリケーションが誤作動を起こす可能性があります。
   - **TRUE (1)**–ターゲット システム上の SSIEnableCmdDirective レジストリ値を 1 に設定します。これにより、サーバー側インクルードの #exec CMD ディレクティブがシェル コマンドの実行で使用できるようになります。

FALSE または TRUE オプションを選択すると、値 (FALSE の場合 0、TRUE の場合 1) がINSTALLSHIELD_SSI_PROP プロパティに格納されます。

インストール内にある Web サイトまたは仮想ディレクトリがターゲット システムにインストールされて、**"SSIEnableCmdDirective レジストリ値"** 設定で FALSE または TRUE オプションが選択されている場合、SSIEnableCmdDirective レジストリ値がターゲット システムで更新されます。

メモ: *製品のインストール中に SSIEnableCmdDirective レジストリ値が変更された場合でも、ターゲット システムから製品がアンインストールされるときに、SSIEnableCmdDirective レジストリ値が変更されることはありません。*

Web サイトの作成とアプリケーションまたは仮想ディレクトリの追加

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*

- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

InstallShield の [IIS 構成] ビューで、プロジェクトに IIS Web サイトを追加します。このビューではまた、Web サイトにアプリケーションおよび仮想ディレクトリを追加することもできます。

タスク ***実行時にターゲット システム上で Web サイトを作成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーを右クリックして、**[Web サイトの追加]** をクリックします。InstallShield が新しい Web サイトを追加します。
3. 設定を構成する Web サイトを選択します。

ヒント: *InstallShieldでは、プロジェクトで Web サイトの TCP ポートとサイト番号を指定できます。これらの設定は、実行時に新しい Web サイトが作成されるか、既存の Web サイトが更新されるのかを判断するのに役立ちます。詳細については、「TCP ポート番号とサイト番号の構成」を参照してください。*

タスク ***実行時にターゲット システム上でアプリケーションを作成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、アプリケーションを含める Web サイトを右クリックして、**[新しいアプリケーション]** をクリックします。新しいリリースが追加されます。
3. 設定を構成するアプリケーションを選択します。

タスク ***実行時にターゲット システム上で仮想ディレクトリを作成するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、ア仮想ディレクトリを含める Web サイトを右クリックして、**[新しい仮想ディレクトリ]** をクリックします。新しい仮想ディレクトリが追加されます。
3. 設定を構成する仮想ディレクトリを選択します。

ヒント: *Web サイト、アプリケーション、および仮想ディレクトリを機能に関連付ける方法については、「IIS サポートの機能の関連付け」を参照してください。*

ネスト仮想ディレクトリの作成

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

既存の仮想ディレクトリの下に、仮想サブディレクトリを作成することができます。

また、インストールの一部としてインストールされる仮想ディレクトリの下にも仮想サブディレクトリを作成することができます。親仮想ディレクトリは、仮想サブディレクトリの前にインストールされる必要があります。

タスク ***既存の仮想ディレクトリの下に仮想ディレクトリを作成するには、以下の手順を実行します。***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、ネスト仮想ディレクトリを含める Web サイトを選択します。
3. 新しい Web サイトを右クリックして **[新しい仮想ディレクトリ]** を選択します。新しい仮想ディレクトリが追加されます。
4. **[全般]** タブをクリックします。
5. **"名前"** 設定で、既存のディレクトリ名と、作成するネスト仮想サブディレクトリの名前を指定します。2 つの名前はスラッシュで区切ります。

   たとえば、**VirtualDirectory** という名前の既存の仮想ディレクトリの下に **MySubDirectory** という名前の仮想ディレクトリを作成する場合、次のように入力します:

   VirtualDirectory/MySubDirectory

メモ: *親ディレクトリがターゲット システムに既に存在しない場合、エンドユーザーが IIS マネージャーでそのディレクトリを開いたとき、ターゲット システムでエラーが表示されます。*

TCP ポート番号とサイト番号の構成

InstallShieldでは、プロジェクトで Web サイトの TCP ポートとサイト番号を指定できます。これらの設定は、実行時に新しい Web サイトが作成されるか、既存の Web サイトが更新されるのかを判断するのに役立ちます。これらはまた、[インターネット インフォーメーション サービス] ビューで構成された Web サイトの設定が、ターゲット システムの Web サイトに適用されるかどうかにも影響します。

タスク *Web サイトに TCP ポート番号とサイト番号を指定するには、以下の手順に従います:*

1. [**ターゲット システムの構成**] の下のビュー リストにある [**IIS 構成**] をクリックします。
2. [**Web サイト**] エクスプローラーで、構成する Web サイトを選択します。
3. [**Web サイト**] タブをクリックします。
4. [**TCP ポート**] ボックスと [**サイト番号**] ボックスで、適切な番号を入力します。

実行時の動作

実行時に、Web サイトがターゲット システムに存在しない場合、インストールはそれを次の規則に従って作成します。

テーブル 4-1 • 各種サンプル "TCP ポート番号" および "サイト番号" 設定値の実行時の結果

InstallShield の "TCP ポート番号" 設定	InstallShield の "サイト番号" 設定	実行時の結果
0	ゼロ以外の値	"TCP ポート番号" 設定が 0 で、"サイト番号" 設定が 0 以外のとき、インストールは Web サイトのアプリケーションおよび仮想ディレクトリをシステムの最初のサイト番号にインストールします。指定されたサイト番号は無視されます。 たとえば、ターゲット システム上の最初のサイト番号が 1 のとき、"サイト番号" 設定に異なるゼロ以外の番号 (たとえば、3) が指定された場合でも、インストールは Web サイトのアプリケーションと仮想ディレクトリをサイト番号 1 にインストールします。 インストールは、[IIS 構成] ビューで構成された Web サイト設定を、ターゲット システム上の Web サイトに一切適用しません。
80 (ゼロ以外の値)	0 (デフォルト値)	指定された TCP ポートがターゲット システムに存在するとき、インストールは TCP ポートで実行中の Web サイト (この例では、ポート 80) にアプリケーションと仮想ディレクトリをインストールします。インストールは、[IIS 構成] ビューで構成された Web サイト設定を、ターゲット システム上の Web サイトに一切適用しません。 TCP ポートがターゲット システムに存在しない場合、プロジェクトで構成された Web サイトの設定を使って新しい Web サイトが作成されます。またインストールは、Web サイトのアプリケーションと仮想ディレクトリをインストールします。

テーブル 4-1・各種サンプル "TCP ポート番号" および "サイト番号" 設定値の実行時の結果 (続き)

InstallShield の "TCP ポート番号" 設定	InstallShield の "サイト番号" 設定	実行時の結果
81 (ゼロ以外の値)	3 (ゼロ以外の値)	指定された TCP ポートおよびサイト番号がターゲット システムに存在するとき、インストールは TCP ポートで実行中の Web サイト (この例では、ポート 3) にアプリケーションと仮想ディレクトリをインストールします。インストールは、[IIS 構成] ビューで構成された Web サイト設定を、ターゲット システム上の Web サイトに一切適用しません。 TCP ポートがターゲット システムに存在するが、サイト番号が存在しない場合、インストールは新しい Web サイトおよびそのアプリケーションと仮想ディレクトリを、既存のポートに新しいサイト番号を使ってインストールします。またインストールは、[IIS 構成] ビューで設定された Web サイトのプロパティを構成します。 TCP ポートがターゲット システムに存在しないが、サイト番号が存在する場合、インストールは新しい Web サイトおよびそのアプリケーションと仮想ディレクトリをこの TCP ポートにインストールします。またインストールは、[IIS 構成] ビューで設定された Web サイトのプロパティを構成します。

Web サイトの IIS ホスト ヘッダー名を指定する

InstallShield では、ホスト ヘッダー名を指定して、インストール中にインストールされる IIS Web サイトを識別することができます。ホスト ヘッダー (ドメイン名とも呼ばれます) を利用して、複数の Web サイトを Web サーバー上の IP アドレスに割り当てることができます。

タスク *Web サイトのホスト ヘッダー名を指定するには、以下の手順に従います:*

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、ホスト ヘッダー名を指定する Web サイトを選択します。
3. **[Web サイト]** タブをクリックします。
4. **"ホスト ヘッダー名"** 設定で、使用するホスト ヘッダー名を入力します。例:

   www.mycompany.com

Web サイトの SSL 証明書を指定する

サーバー証明書を利用して、ユーザーは Web サーバーの認証および Web コンテンツの有効性の確認を行うことができると共に、セキュリティで保護された接続を確立することができます。では、実行時にインストールできるように、インストールに Web サイトのサーバー証明書を含めることができます。

タスク *Web サイトにインストールする SSL 証明書を指定するには、以下の手順に従います:*

1. [**ターゲット システムの構成**] の下のビュー リストにある [**IIS 構成**] をクリックします。
2. [**Web サイト**] エクスプローラーで、SSL 証明書を指定する Web サイトを選択します。
3. **"SSL 証明書"** 設定で、省略記号ボタン (...) をクリックします。[**開く**] ダイアログ ボックスが開きます。
4. インストールするセキュリティ証明書ファイル (.cer または .pfx) を選択して、[**開く**] をクリックします。
5. 証明書にパスワードが必要な場合、**"SSL 証明書パスワード"** 設定でパスワードを指定します。

.cer ファイルが Binary テーブルに格納されます。実行時、インストールで Web サイトと仮想ディレクトリがインストールされるとき、SSL 証明書もインストールされます。

ファイルを IIS 仮想ディレクトリに追加する

タスク *ファイルを IIS 仮想ディレクトリに追加する:*

1. IIS Web サイトをプロジェクトに追加していない場合は、それを行います。InstallShield が自動的に定義済みのパス [IISROOTFOLDER] を [**ファイル**] ビューへ追加します。
2. [**アプリケーション データの指定**] の下にあるビュー リストで、[**ファイル**] をクリックします。
3. [**機能**] リストで、ファイルに関連付ける機能を選択します。
4. [**インストール先コンピューターのフォルダー**] ペインで、ファイルを [IISROOTFOLDER] フォルダー、または [IISROOTFOLDER] フォルダーのサブフォルダーに追加します。
5. [**ターゲット システムの構成**] の下のビュー リストにある [**IIS 構成**] をクリックします。
6. 新しい仮想ディレクトリを作成します。
7. [**Web サイト**] エクスプローラーで、作成した仮想ディレクトリをクリックします。
8. **"コンテンツ ソース パス (ローカルまたは UNC)"** 設定で省略記号ボタン (...) をクリックします。[**ディレクトリの参照**] ダイアログ ボックスが開きます。デフォルトでは、これらのファイルは IISROOTFOLDER に格納されています。
9. [**ファイルとフォルダー**] ビューで追加した新しいファイルを含むディレクトリと同じターゲット ディレクトリを入力します。
10. [OK] をクリックします。

インストール時、ファイルはターゲット ディレクトリ フォルダーへコピーされます。さらに IIS が存在する場合、ターゲット システム上のフォルダーに仮想ディレクトリが構成されます。

[IIS 構成] ビューからアプリケーションと仮想ディレクトリを削除する

タスク *アプリケーションまたは仮想ディレクトリをインストールから削除するには、以下の手順に従います:*

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、仮想ディレクトリを右クリックしてから **[削除]** をクリックします。

IIS サポートの機能の関連付け

InstallShield では、プロジェクトで、機能に Web サイトを関連付けることができます。その機能がインストールされるとき、Web サイトとそのアプリケーションおよび仮想ディレクトリのすべてがインストールされます。選択された機能がインストールされない場合、Web サイトとそのアプリケーションおよび仮想ディレクトリはインストールされません。

タスク *プロジェクトで、機能に Web サイトを関連付けるには、以下の手順に従います:*

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. エクスプローラーで、機能に関連付ける Web サイトをクリックします。
3. **"機能"** 設定で、選択した IIS データを含める既存の機能の名前を選択します。

ヒント: *プロジェクトから機能を削除した場合、機能に関連付けられている Web サイト、アプリケーション、および仮想ディレクトリもプロジェクトから削除されます。*

Web サイト、アプリケーション、および仮想ディレクトリのアンインストール

インストールによって作成された Web サイトは、次の条件の両方が True とならない限り、削除されません。

- Web サイトにアプリケーションまたは仮想ディレクトリが含まれていない。
- [IIS 構成] ビューにある Web サイトの **"アンインストール時に削除する"** 設定の値が [はい] である。

機能がアンインストールされると、その Web サイト、IIS アプリケーション、および仮想ディレクトリもアンインストールされます。

Web サイトまたはアプリケーションの ASP.NET バージョンを設定する

InstallShield では、インストールにある Web サイトまたはアプリケーションの ASP.NET バージョンを設定することができます。ASP.NET バージョンを指定すると、Web サイトまたはアプリケーションが作成された後、ASP.NET IIS の Registration Tool (Aspnet_regiis.exe) がインストールで実行され、Web サイトまたはアプリケーションを指定したバージョンにマッピングします。

Web サイトに ASP.NET バージョンを指定すると、IIS はその値を実行時に作成された Web サイトおよびすべてのそのアプリケーションに使用します。

重要: *マイクロソフト社は、Aspnet_regiis.exe ツールの機能に制限があるため、Windows Vista または Windows Server 2008 以降のシステムでの使用を推奨していません。結果として、場合により、アプリケーションのマッピングを [IIS 構成] ビューで手動で定義する必要があります。詳細については、「Web サイト、アプリケーション、または仮想ディレクトリのアプリケーションのマッピングを定義する」を参照してください。*

ASP.NET 3.0 には Aspnet_regiis.exe ツールは含まれていません。したがって、ASP.NET バージョンを バージョン 3 の ASP.NET に設定できません。

タスク ***プロジェクトで Web サイトまたはアプリケーションの ASP.NET バージョンを指定するには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、ASP.NET バージョンを指定する Web サイトまたはアプリケーションを選択します。Web サイトまたはアプリケーションの設定が右側に表示されます。
3. **"ASP .NET バージョン"** 設定で、アプリケーションに必要な .NET Framework のバージョン番号を完全な形で入力するか、一覧から選択します。

   たとえば、バージョン 2 の ASP.NET を指定する場合、**2.0.50727** と入力します。バージョン 1.1 の ASP.NET を指定するには、**1.1.4322** と入力します。

Web サイト、アプリケーション、または仮想ディレクトリのアプリケーションのマッピングを定義する

InstallShield では、ファイル名拡張子とこれらのファイルを処理するアプリケーション間のマッピングを定義することができます。

タスク ***アプリケーションのマッピングの追加/編集/削除を行うには、以下の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、構成する Web サイト、アプリケーション、または仮想ディレクトリを選択します。
3. **"アプリケーションのマッピング"** 設定で、省略記号ボタン (...) をクリックします。**[アプリケーションのマッピング]** ダイアログ ボックスが開きます。
4. 以下のいずれかを実行します。
   a. 新しいマッピングを追加するには、**[追加]** ボタンをクリックします。**[アプリケーション拡張子マッピング]** ダイアログ ボックスが開きます。詳細については、「[アプリケーション拡張子マッピング] ダイアログ ボックス」を参照してください。
   b. 既存のマッピングを変更するには、編集するマッピングを選択して、**[編集]** ボタンをクリックします。
   c. 既存のマッピングを削除するには、それを選択して、**[削除]** ボタンをクリックします。

Web サイトまたは仮想ディレクトリのタイムアウトのパラメーターを指定する

InstallShieldWeb サイト、アプリケーション、または仮想ディレクトリのタイムアウトのパラメーターを指定する

タスク *タイムアウトのパラメーターを指定するには、以下の手順に従います:*

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、構成する Web サイト、アプリケーション、または仮想ディレクトリを選択します。
3. **"アプリケーションの設定"** 領域で、**[構成]** ボタンをクリックします。**[アプリケーションのマッピング]** ダイアログ ボックスが開きます。
4. **"セッション タイムアウト(分)"** と **"ASP スクリプト タイムアウト(秒)"** 設定で、適切なタイムアウト値を指定します。

追加の IIS 仮想ディレクトリのプロパティを設定する

[IIS 構成] ビュー内の [Web サイト] エクスプローラーにある仮想ディレクトリを選択したときに表示されるタブには、IIS 仮想ディレクトリの最も一般的な設定項目が含まれています。これらのタブに含まれていない、その他の IIS 仮想ディレクトリ設定を行うこともできます。

タスク *[IIS 構成] ビューに表示されていない設定を構成するには、以下の手順に従います:*

1. VBScript によって遅延されるカスタム アクションを作成します。
2. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[カスタム アクション]** をクリックします。
3. カスタム アクションを **[インストール中のカスタム アクション]** エクスプローラーにある **[セットアップが正常に完了した後]** ダイアログに追加します。
4. ADSI オブジェクト モデルを利用して設定を行います。

ADSI オブジェクト モデルを使った IIS 仮想ディレクトリ の設定方法については MSDN の文書を参照してください。

Web サイト、アプリケーション、または仮想ディレクトリのカスタム エラー メッセージを定義する

エンドユーザーが Web サイトに接続しようとして、HTTP (Hypertext Transfer Protocol) エラーが発生した場合、エンド ユーザーのブラウザーはエラーを説明するデフォルトのメッセージを表示します。HTTP エラー コードをファイルまたは URL (Uniform Resource Locator) へマッピングすることで、インストールが IIS を構成して、デフォルトのエラー メッセージの代わりにカスタム エラー メッセージを表示するよう設定することができます。

タスク ***Web サイト、アプリケーション、または仮想ディレクトリのカスタムエラーメッセージを構成するには、以下の手順を実行します。***

1. カスタムエラーメッセージを含むファイルを作成し、インストールに追加します。
2. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
3. HTTP エラーメッセージをカスタマイズする Web サイト、アプリケーション、または仮想ディレクトリを選択します。Web サイト、アプリケーション、または仮想ディレクトリの設定が、右側に表示されます。
4. **"カスタム エラー"** 設定で、省略記号ボタン (...) をクリックします。**[カスタム エラー]** ダイアログ ボックスが開きます。
5. 変更する HTTP エラーコードを選択し、**[プロパティの編集]** ボタンをクリックします。**[新しいプロパティの追加]** ダイアログ ボックスが開きます。
6. ファイルへエラーコードをマップするには、以下の手順を実行します。
   a. **[メッセージの種類]** リストから **[ファイル]** を選択します。
   b. **[ファイル]** ボックスで、インストール内のカスタム エラー メッセージをポイントするパスおよびファイル名を入力するか、[参照] ボタンを押してファイルを指定します。

   URL へエラー コードをマップするには、以下の手順を実行します。

   a. **[メッセージの種類]** 一覧から URL を選択します。
   b. **URL** ボックスで、カスタム エラー メッセージへの URL を入力します。

ターゲット マシン上で Web サービスを配布する

ターゲット システムへ Web サービスを配布するには、Web サービス専用のファイルを特定の場所にコピーし、そのフォルダーに仮想ディレクトリ名を割り当てる必要があります。これにより、HTTP を通して Web サービスにアクセスできるようになります。

ヒント: *仮想ディレクトリをプロジェクトに追加する方法については、「Web サイトの作成とアプリケーションまたは仮想ディレクトリの追加」をご覧ください。*

タスク ***ターゲットマシンへ Web サービスを配布するには、以下の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[機能]** リストで、Web サービスに関連付ける機能を選択します。
3. **[インストール先コンピューターのフォルダー]** ペインで、ターゲット システムにファイルをインストールするためのフォルダー (ターゲット ディレクトリ) を選択します。そのフォルダーにファイルを追加します。このフォルダーを [IISROOTFOLDER] というフォルダーで作成した新しいフォルダーにすることもできます。
4. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
5. **[Web サイト]** エクスプローラーで、Web サイトに関連付けられている仮想ディレクトリを選択します。

6. **"コンテンツ ソース パス (ローカルまたは UNC)"** 設定で省略記号ボタン (...) をクリックします。**[ディレクトリの参照]** ダイアログ ボックスが開きます。**[ファイルとフォルダー]** ビューで追加した新しいファイルを含むディレクトリと同じターゲット ディレクトリを入力します。

インストール時、ファイルはターゲット ディレクトリ フォルダーへコピーされます。さらに IIS が存在する場合、ターゲット システム上のフォルダーに仮想ディレクトリが構成されます。

カスタム アクションを使って IIS 機能を抑制する

カスタム アクションを使って IIS を停止すると、インストールで IIS がロックしたファイルを上書きできます。最新版の IIS を使用している場合、次の VBScript を実行して Web サイトをアンロードし、.dll ファイルをインストールすることができます。これによって、IIS を再開させる必要が無くなり、実行中の Web サイトすべてをドロップすることもありません。

例:

Dim DirObjSet DirObj = GetObject("IIS://LocalHost/W3SVC/1/Root/")DirObj.AppUnloadset dirObj = nothing"

Web アプリケーションのフォーム認証を有効化する

***プロジェクト**・この情報は、次のプロジェクトの種類に適用します :*

- *基本の MSI*
- *InstallScript MSI*

Web サイトの **[IIS 構成]** ビューで **認証済みアクセス]** セクションの下にある **"フォーム認証"** 設定を使って、Web アプリケーションのフォーム認証を設定します。

ASP.NET フォーム ベースの認証は、多くの要求を受け取るパブリック Web サーバー上にあるサイトまたはアプリケーションに適しています。この認証モードでは、オペレーティング システムが提供している認証方法に頼ることなく、アプリケーション レベルでのクライアント登録および認証を管理することができます。

***重要**・フォーム認証は、Web サーバーにユーザー名とパスワードをテキスト形式で送信します。ホームページ以外のアプリケーション内のすべてのページ、およびログオン ページには、Secure Sockets Layer (SSL) 暗号化を使用してください。*

タスク ***Web サイトのフォーム認証を有効化するには、次の手順に従います:***

1. **[ターゲット システムの構成]** の下のビュー リストにある **[IIS 構成]** をクリックします。
2. **[Web サイト]** エクスプローラーで、Web サイトを選択します。
3. **[セキュリティ > 認証アクセス]** の下にある **"フォーム認証"** 設定を **[はい]** に設定します。

IISROOTFOLDER サポートの追加

IISROOTFOLDER は、ターゲット システム上にある Web サーバーのルート ディレクトリを判断するのに利用される InstallShield ディレクトリ変数です。インストール プロジェクトで IIS 機能を使用していて、Web サイトが既に追加されている場合 IISROOTFOLDER は自動的に追加されます。

メモ: *[ファイル] ビューの IISROOTFOLDER ディレクトリに追加したファイルはすべて、ターゲット マシンの Web サーバーのルート ディレクトリにインストールされます。IIS がターゲットマシン上に存在しない場合、ファイルはルートフォルダーへコピーされます。*

IIS_WEBSITE_NAME プロパティ

IIS_WEBSITE_NAME プロパティは現在使用されていません。これらのプロパティが以前のプロジェクト バージョンから存在している場合、アップグレーダはこれらを自動的に処理します。アップグレーダは Web サイトを作成して、サイト番号フィールドを [IIS_WEBSITE_NAME] に設定します。新しい Web サイトには、任意のプロパティまたはハードコード化された番号を使用できます。

IIS_PORT_NUMBER プロパティ

IIS_PORT_NUMBER プロパティは現在使用されていません。これらのプロパティが以前のプロジェクト バージョンから存在している場合、アップグレーダはこれらを自動的に処理します。アップグレーダは Web サイトを作成して、ポート番号フィールドを [IIS_PORT_NUMBER] に設定します。新しい Web サイトを作成すると、任意のプロパティまたはハードコード化された番号を使用できます。

エンドユーザー インターフェイスを定義する

ヘルプ ライブラリのこのセクションでは、エンドユーザー インターフェイスの様々な要素を定義することができる、InstallShield のいくつかの機能について説明されています。インストール中に表示されるダイアログとビルボードからテキストまですべてが網羅されています。

ダイアログの使い方

インストールのユーザー インターフェイスは様々な意味で重要ですが、特に、エンドユーザーの入力や設定が通常ユーザー インターフェイスを通じて処理されるという点で非常に重要です。ユーザー インターフェイスで目的の画面を表示することが困難だったり、理解しにくいと、ユーザーは製品をインストールする際、問題を抱えることになります。インストールの作成手順や、エンドユーザーの操作を簡素化するために、InstallShield ではいくつかの定義済みダイアログの利用が可能です。

使用できるダイアログは提供されているものに限られていますが、それらの多くはカスタマイズでき、必要な外観や機能性を持たせることができます。たとえば、ダイアログそれぞれの上部にカスタマイズしたイメージを追加できるので、会社のロゴを使用して強い印象を与えることができます。

インストールにダイアログを追加する

タスク ***ダイアログをインストールに追加するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーで、追加するダイアログのチェック ボックスを選択します。

ダイアログのテーマ

プロジェクト: *ダイアログ テーマは Express プロジェクトで使用できます。*

ダイアログ テーマは、エンドユーザー ダイアログに統一感のとれた個性的な印象を与えることができる、あらかじめ定義されている 1 セットのイメージです。

[ダイアログ] ビューの "グローバル ダイアログ テーマ" 設定で選択したテーマ オプションを変更して、プロジェクトに提供されているテーマから任意のテーマを選択し、プロジェクトで使用されているすべての内部および外部ダイアログ (Setup.exe 初期化ダイアログを含む) に適用することができます。

メモ: *InstallShield では現在、独自のダイアログ テーマを作成することはできませんが、2 つのテーマが用意されています。詳細については、「ダイアログのテーマ」を参照してください。*

ダイアログ テーマの選択または変更

プロジェクト: *ダイアログ テーマは Express プロジェクトで使用できます。*

ダイアログ テーマを使用して、インストールのエンド ユーザー ダイアログの外観を変更することができます。1 つのプロジェクトにつき、1 つのテーマを選択することができます。

タスク ***プロジェクトに使用されているダイアログ テーマを変更するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ダイアログ]** をクリックします。
2. **[ダイアログ]** エクスプローラーをクリックします。グローバル ダイアログの設定が、右のペインに表示されます。
3. ***"グローバル ダイアログ テーマ"*** 設定で、使用するテーマを選択します。

選択されたテーマが、プロジェクトのダイアログに適用されます。

ダイアログのテーマ

プロジェクト: *InstallShield Premier および InstallShield では、Express Edition で提供されていない機能が提供されています。*

InstallShield には 2 つの異なるテーマが含まれています。

- Classic テーマ
- InstallShield Blue テーマ

これらのテーマのサンプルは、「エンドユーザー ダイアログ」セクションをご覧ください。

ダイアログのビットマップ イメージ

各エンドユーザー ダイアログには、以下で説明されているイメージの 1 つが含まれています。

スプラッシュ ビットマップ

スプラッシュ ビットマップ イメージは、[スプラッシュ ビットマップ] ダイアログで表示されます。このイメージは、465 ピクセル (幅) x 281 ピクセル (高さ) の .bmp または .jpg ファイルでなければなりません。

次のサンプル ダイアログ (Classic テーマと InstallShield Blue テーマ) では、デフォルトのスプラッシュ ビットマップ ファイルが表示されています。デフォルトのイメージは、独自のイメージ ファイルで置き換えることができます。

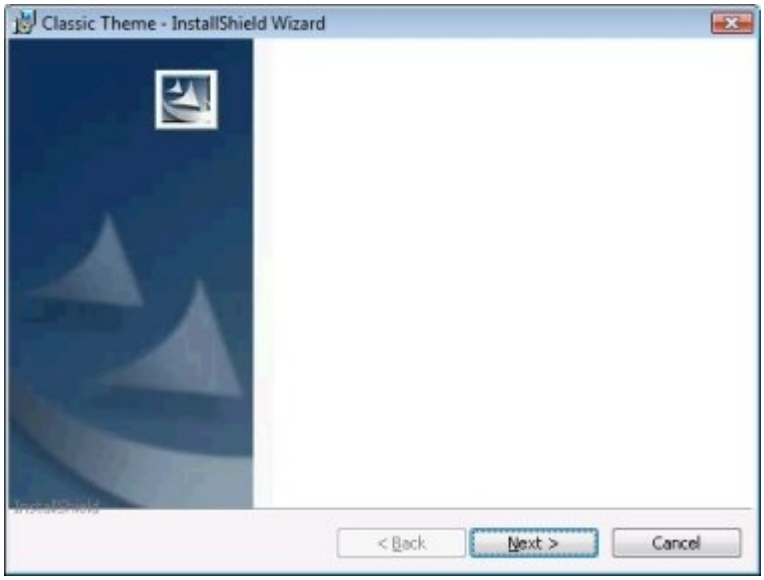


図 4-1: Classic テーマの [スプラッシュ ビットマップ] ダイアログ

図 4-2: InstallShield Blue テーマの [スプラッシュ ビットマップ] ダイアログ

ビットマップ イメージ

ビットマップ イメージは、[インストール - ようこそ] ダイアログと [セットアップの完了] ダイアログのバックグラウンドで表示されます。このイメージは、499 ピクセル (幅) x 312 ピクセル (高さ) の .bmp または .jpg ファイルでなければなりません。

各ダイアログのビットマップのイメージを変更したり、ビットマップ イメージを含むすべてのダイアログで使用するグローバル ダイアログ イメージを指定したりすることができます。

次のサンプル ダイアログ (Classic テーマと InstallShield Blue テーマ) では、デフォルトのビットマップ イメージが表示されています。イメージは 2 列に分かれていて、テキストの背後に白色の背景が含まれています。

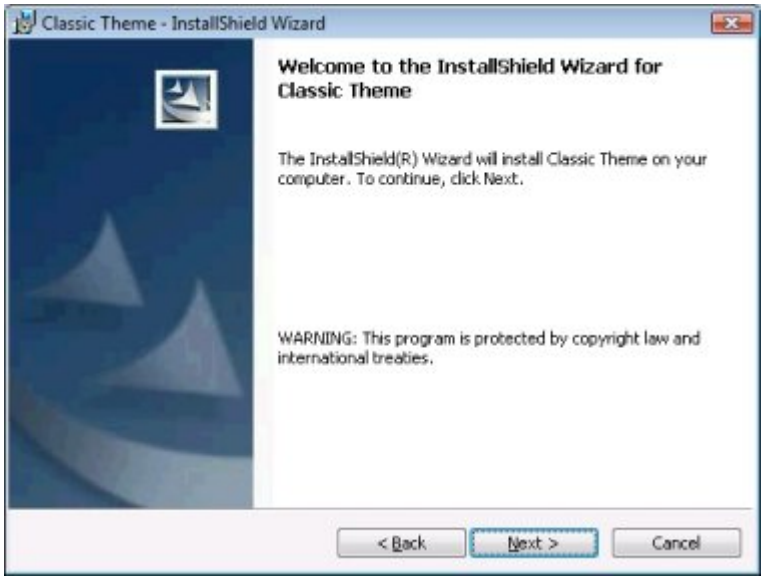


図 4-3: Classic テーマの [インストール - ようこそ] ダイアログ

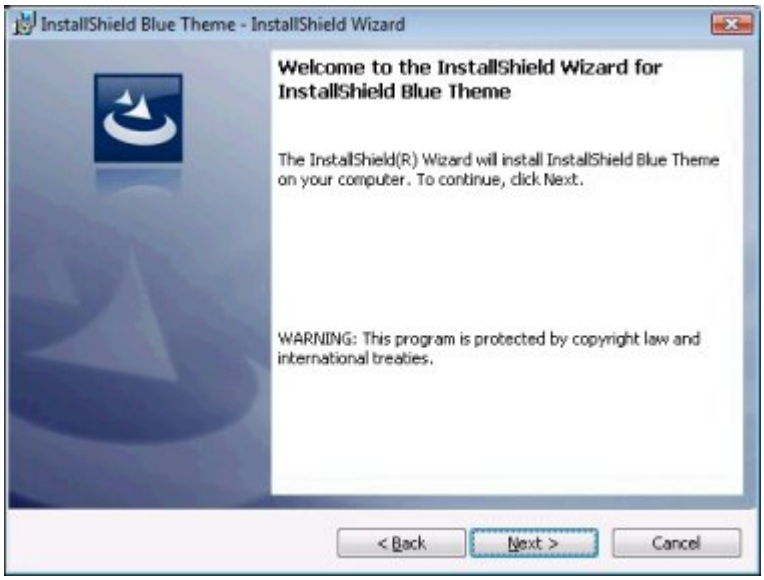


図 4-4: InstallShield Blue テーマの [インストール - ようこそ] ダイアログ

バナー ビットマップ

バナー ビットマップ イメージは、いくつかのダイアログでタイトル バーのすぐ下に表示されます ([使用許諾契約]、[Readme]、[ユーザー情報]、[インストール先のフォルダー]、[データベース フォルダー]、[セットアップの種類]、[カスタム セットアップ]、[インストールの準備完了]、[セットアップ進行状況])。このイメージは、499 ピクセル (幅) x 58 ピクセル (高さ) の .bmp または .jpg ファイルでなければなりません。

各ダイアログのバナー ビットマップのイメージを変更したり、バナー ビットマップを含むすべてのダイアログで使用するグローバル ダイアログ バナーを指定したりすることができます。

次のサンプル ダイアログ (Classic テーマと InstallShield Blue テーマ) では、バナー ビットマップが表示されています。イメージは、ダイアログのタイトル領域の右端にある画像です。

図 4-5: Classic テーマの [使用許諾契約書] ダイアログ

図 4-6: InstallShield Blue テーマの [使用許諾契約書] ダイアログ

[スプラッシュ ビットマップ] ダイアログのスプラッシュ ビットマップを変更する

タスク *[スプラッシュ ビットマップ] ダイアログのスプラッシュ ビットマップを変更するには、以下の手順に従います:*

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーで、[**スプラッシュ ビットマップ**] をクリックします。

3. **"スプラッシュ ビットマップ"** 設定で、ダイアログのイメージとして使用する .bmp または .jpg ファイルへのパスを入力します。また、省略記号 (...) ボタンを使用してファイルを参照することもできます。イメージは、465 (幅) x 281 (高さ) ピクセルである必要があります。

エンドユーザー ダイアログのビットマップ イメージを変更する

タスク ***エンドユーザー ダイアログのビットマップ イメージを変更するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ダイアログ]** をクリックします。
2. **[ダイアログ]** エクスプローラーで、ビットマップを変更するダイアログをクリックします。
3. **"ビットマップ イメージ"** 設定で、ダイアログのイメージとして使用する .bmp または .jpg ファイルへのパスを入力します。また、省略記号 (...) ボタンを使用してファイルを参照することもできます。イメージは、499 (幅) x 312 (高さ) ピクセルである必要があります。

重要: *また、"グローバル ダイアログ イメージ" プロパティ ([ダイアログ] ビューにある [ダイアログ] エクスプローラーをクリックしたとき開きます) を構成することで、すべてのエンドユーザー ダイアログに共通するビットマップ イメージを構成することもできます。特定のダイアログに関して "ビットマップ イメージ" 設定の値を変更した後に、"グローバル ダイアログ イメージ" 設定の値を変更すると、"ビットマップ イメージ" 設定の値は "グローバル ダイアログ イメージ" 設定の値で上書きされます。*

エンドユーザー ダイアログのバナー ビットマップを変更する

タスク ***エンドユーザー ダイアログのバナー ビットマップを変更するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ダイアログ]** をクリックします。
2. **[ダイアログ]** エクスプローラーで、ビットマップを変更するダイアログをクリックします。
3. **"バナー ビットマップ"** 設定で、ダイアログのイメージとして使用する .bmp または .jpg ファイルへのパスを入力します。また、省略記号 (...) ボタンを使用してファイルを参照することもできます。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。

重要: *また、"グローバル ダイアログ バナー" 設定 ([ダイアログ] ビューにある [ダイアログ] エクスプローラーをクリックしたとき開きます) を構成することで、すべてのエンドユーザー ダイアログに共通するバナー イメージを構成することもできます。特定のダイアログに関して "バナー ビットマップ" 設定の値を変更した後に、"グローバル ダイアログ バナー" 設定の値を変更すると、"バナー ビットマップ" 設定の値は "グローバル ダイアログ バナー" 設定の値で上書きされます。*

エンドユーザー ダイアログのグローバル ダイアログ イメージを変更する

タスク ***ビットマップ イメージを含むすべてのエンドユーザー ダイアログのビットマップ イメージを変更するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーをクリックします。
3. ***"グローバル ダイアログ イメージ"*** 設定で、ビットマップ イメージを含むすべてのダイアログでビットマップ イメージとして使用する .bmp または .jpg ファイルへのパスを入力します。また、省略記号 (...) ボタンを使用してファイルを参照することもできます。イメージは、499 (幅) x 312 (高さ) ピクセルである必要があります。

ヒント: *個々のダイアログにビットマップ イメージを設定することもできます。*

注意: *特定のダイアログに関して "ビットマップ イメージ" 設定の値を変更した後に、"グローバル ダイアログ イメージ" 設定の値を変更すると、"ビットマップ イメージ" 設定の値は "グローバル ダイアログ イメージ" 設定の値で上書きされます。*

エンドユーザー ダイアログのグローバル ダイアログ バナーを変更する

タスク ***ビットマップ イメージを含むすべてのエンドユーザー ダイアログのバナー ビットマップを変更するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーをクリックします。
3. ***"グローバル ダイアログ バナー"*** 設定で、ビットマップ バナーを含むすべてのダイアログでビットマップ バナーとして使用する .bmp または .jpg ファイルへのパスを入力します。また、省略記号 (...) ボタンを使用してファイルを参照することもできます。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。

ヒント: *個々のダイアログにバナー ビットマップを設定することもできます。*

注意: *特定のダイアログに関して "バナー イメージ" 設定の値を変更した後に、"グローバル ダイアログ バナー" 設定の値を変更すると、"バナー イメージ" 設定の値は "グローバル ダイアログ バナー" 設定の値で上書きされます。*

Windows Vista 以降のシステムの再起動を最小限にする

Windows ロゴ: *インストール終了後のシステム再起動は、エンドユーザーにとって不都合なものです。Windows ロゴ プログラムの要件の 1 つに、エンドユーザーがインストール完了後自動的にアプリケーションを閉じて再起動を行うことができるオプションを含まなくてはならないという項目があります。*

この要件をサポートするため、すべての Express プロジェクトには、デフォルトで MsiRMFilesInUse ダイアログが含まれます。インストール中に更新が必要なファイル (複数可) が他のアプリケーションによって使用中の場合、Windows Vista 以降のシステム上では [使用中のファイル (再起動マネージャー)] ダイアログが表示されます。ダイアログには、エンドユーザーが選択できる次の 2 つのオプションが含まれます。

- エンドユーザーは選択で、インストールの完了後、自動的にファイルを使用中のアプリケーションを閉じて、再起動することができる。
- エンドユーザーは、アプリケーションを閉じない選択ができる。インストールの終わりで再起動が必要。

エンド ユーザー エクスペリエンスを最適化するため、アプリケーションには再起動マネージャー API の利用が推奨されます。再起動マネージャーは、エンド ユーザーがアプリケーションを停止した時点から正確に、また効果的にこれを再開します。詳しい情報は、「About Restart Manager (再起動マネージャー)」および MSDN Web サイトで再起動マネージャーに関するその他の文書を参照してください。

[カスタム セットアップ] ダイアログのオプション

[カスタム セットアップ] ダイアログには、ターゲット システム、インストール中の機能、および Windows Installer のインストール オプションについての情報と緊密に統合された、洗練されたユーザー インターフェイスがあります。これによって、エンドユーザーがインストールを最大限に制御することができます。

このダイアログが提供する多くのオプションおよび情報は、下記で説明されているようにセットアップのデザインで設定された機能のプロパティによって決定されます。

テーブル 4-1 • [カスタム セットアップ] ダイアログのオプション

オプション	説明
アドバタイズ	アドバタイズ機能を利用すると、インストールが初回で実行された後、オン デマンドでファイルをインストールすることができます。[カスタム セットアップ] ダイアログで、機能をクリックすると、[必要な場合にインストール] オプションを選択して、その機能を後でインストールするように指定することができます。 ただし、このデフォルト オプションが表示されるのは、セットアップの作成者が機能の "アドバタイズ" 設定に [はい] を選択した場合のみです。
エンドユーザーに対して機能を隠す	機能の "可視" 設定 を [いいえ] に設定すると、[カスタム セットアップ] ダイアログに機能またはサブ機能は表示されず、エンドユーザーはインストール オプションを変更できなくなります。
すべてのサブ機能を表示する	機能の "可視" 設定はまた、ダイアログが最初に表示されたときに、サブ機能が展開されるかどうかを制御します。

テーブル 4-1 • [カスタム セットアップ] ダイアログのオプション (続き)

オプション	説明
機能の説明の表示	機能を選択したときに [カスタム セットアップ] ダイアログの下部に表示される説明は、機能の "説明" 設定から取得されます。
機能の順序を変更する	Custom Setup ダイアログで機能がエンドユーザーに対して表示される順序を変更することができます。機能が表示される順序は、[機能] ビューの順序から取得されます。詳細については、「[カスタム セットアップ] ダイアログで機能の順序を変える」を参照してください。
機能のインストールを必須にする	機能の "必須" 設定を [はい] に設定すると、エンド ユーザーに対して [**インストールしない**] オプションが表示されず、機能をインストールしなければならなくなります。

実行時に使用許諾契約書を表示する

使用許諾契約書のダイアログは、インストールのデフォルト ユーザー インターフェイスの一部として含まれています。使用許諾契約書を表示するには、リッチテキストファイル (.rtf) をこのダイアログに関連付けます。

タスク ***使用許諾契約書にテキストを追加するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーで、[**使用許諾契約書**] をクリックします。
3. ***"ライセンス ファイル"*** 設定をクリックして、参照 (...) ボタンをクリックして、使用する .rtf ファイルを参照します。

インストールからダイアログを削除する

タスク ***インストールからダイアログを削除するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ダイアログ**] をクリックします。
2. [**ダイアログ**] エクスプローラーで、削除するダイアログのチェック ボックスをクリアします。

実行時のテキストとメッセージを編集する

InstallShield では、インストール時に表示される文字列を完全にローカライズする機能が提供されています。[次へ] ボタンの文字列から機能の説明まですべてをカスタマイズできます。

InstallShield の [テキストとメッセージ] ビューには、インストール プロセス中に表示できるすべてのダイアログとメッセージが一覧表示されます。このビューにあるダイアログを 1 つクリックすると、そのダイアログのサンプル スクリーン ショットが、そのダイアログに属するすべての文字列を含む文字列テーブルと共に表示されま

す。同様に、このビューでメッセージを 1 つクリックすると、サンプル メッセージ ボックスとすべての関連する文字列を含む文字列テーブルが表示されます。このビューからはすべてのランタイム文字列にアクセスできます。

他の国のユーザーのためにローカライズすることは、インストールを作成する過程で大変重要です。InstallShield では、インストール プロジェクトにあるランタイム文字列のすべてをテキスト (.txt) ファイルにエクスポートして、翻訳作業の負担を大幅に軽減できます。文字列をエクスポートする場合、翻訳用に 1 つの共通のファイル タイプで送り、それをインストール プロジェクトにインポートし直してローカライズ済みのユーザー インターフェイスを取得することができます。

実行時の文字列を編集する

タスク ***実行時の文字列を編集するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**テキストとメッセージ**] をクリックします。
2. [**テキストとメッセージ**] エクスプローラーで、編集するダイアログまたはメッセージをクリックします。
3. 編集する文字列の値をダブルクリックし、新しい文字列を入力します。

ヒント: *ダイアログ テキストで Windows Installer のプロパティを使用することができます。Windows Installer のプロパティを使用するには、プロパティを角かっこで囲みます (例、*[INSTALLDIR]*)。*

テキストとメッセージ文字列にコメントを追加する

プロジェクトでテキストまたはメッセージ文字列にコメントを追加することができます。これらのコメントはエンドユーザーには表示されません。文字列を識別する目的でのみ使用します。

タスク ***文字列にコメントを追加するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**テキストとメッセージ**] をクリックします。
2. [**テキストとメッセージ**] エクスプローラーで、コメントを追加するダイアログまたはメッセージをクリックして、コメントを追加する文字列を見つけます。
3. 文字列の "**コメント**" の値をダブルクリックして、コメントを入力します。この値を見るために右へスクロールする必要がある場合もあります。

テキストとメッセージ文字列のフォントを変更する

インストールで使用されているダイアログとメッセージ ボックスのすべての文字列のデフォルト フォントとフォント サイズは、[一般情報] ビューで設定されます。InstallShield では、1 つ以上の特定の文字列のデフォルト フォントまたはフォント サイズをオーバーライドすることができます。

タスク ***特定の文字列のフォントとフォント サイズを変更するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**テキストとメッセージ**] をクリックします。
2. [**テキストとメッセージ**] エクスプローラーで、フォントを変更するダイアログまたはメッセージをクリックして、該当の文字列を見つけます。
3. フォントを変更する文字列の "**フォント**" の値をダブルクリックします。
4. 以下のいずれかを実行します。
   - [**フォント**] リストで、適切なフォントを選択します。
   - "**フォント**" 設定で、省略記号ボタン (...) をクリックします。[**フォント**] ダイアログ ボックスが開きます。選択した文字列のフォントとフォントの特徴 (サイズ、スタイル、色など) を選択し、[OK] をクリックします。

文字列のエクスポート

インストール プロジェクトで使用されているすべてのランタイム文字列を翻訳するタスクを軽減するため、InstallShield では文字列をテキスト (.txt) ファイルにエクスポートすることができます。この .txt ファイルを、翻訳されたテキストで更新することができる翻訳者に渡すことができます。.txt ファイルをインストール プロジェクトにインポートしなおすと、ローカライズされたユーザー インターフェイスが完成します。

タスク ***プロジェクト内のランタイム文字列をすべてエクスポートするには、以下の手順に従います:***

1. [**プロジェクト**] メニューで、[**文字列エントリのエクスポート**] をクリックします。**文字列テーブルのエクスポート ウィザード**が開きます。
2. ウィザードのパネルを完成して、文字列を .txt ファイルにインポートします。

文字列テーブルのインポート

インストールの文字列を含む .txt ファイルの翻訳が終わり、その .txt ファイルをプロジェクトにインポートすると、ローカライズ済みのユーザー インターフェイスが完成します。

タスク ***プロジェクトで翻訳済みのランタイム文字列をインポートするには、以下の手順に従います:***

1. [**プロジェクト**] メニューで [**文字列エントリのインポート**] をクリックします。**文字列テーブルのインポート ウィザード**が開きます。
2. ウィザードのパネルを完成して、文字列をインポートします。

ビルボードを表示する

ビルボードをプロジェクトに追加して、インストール処理中にエンド ユーザーに対して情報を提供できます。ビルボードは、エンド ユーザーと連絡を取ったり、広告、教育、およびエンターテインメントを提示するために使用することができます。たとえば、ビルボードを使ってインストール中の製品に含まれる新しい機能の概要や貴社の他の製品についての情報を提供できます。各ビルボードは、貴社のグラフィック担当者がファイル転送の外観を完全にカスタマイズできるファイルです。

ビルボード ファイルの種類

InstallShield では、ビルボード用に様々な種類のファイルがサポートされています:

- Adobe Flash アプリケーション ファイル (.swf)
- イメージ (.bmp、.gif、.jpg、および .jpeg)

Flash アプリケーション ファイルを表示するために必要な Adobe Flash Player がターゲット システムに存在しない場合、インストールはそれを検知して Flash ビルボードの代わりにイメージ ビルボードを表示します。このため、Flash ビルボードをプロジェクトに含める場合は、1 つ以上のイメージ ビルボードもプロジェクトに含めることが推奨されます。

メモ: *プロジェクトに複数のイメージ ビルボードを追加することができますが、Adobe Flash アプリケーション ファイル ビルボードの場合は 1 つだけしか追加できません。*

ビルボードの種類

InstallShield は、異なる種類のビルボードをサポートします。たとえば、インストールが全画面背景を使用し、ビルボードを前景に、また小さい進行状況ボックスを画面の右下に表示するスタイルがあります。別のスタイルでは、インストールがビルボードを表示する標準サイズのダイアログを表示します。このダイアログの下の部分に、進行状況バーが表示されます。

各ビルボード タイプの説明とサンプル スクリーンショットは、次のとおりです。

全画面表示、右下に小さい進行状況ボックスを表示する

全画面表示、右下に小さい進行状況ボックスを表示するタイプのビルボードは、インストールが標準エンドユーザー ダイアログを表示するときに、全画面の背景も表示します。ファイルの転送中、インストールが全画面背景を使用し、ビルボードを前画面に、また小さい進行状況ボックスを画面の右下に表示します。

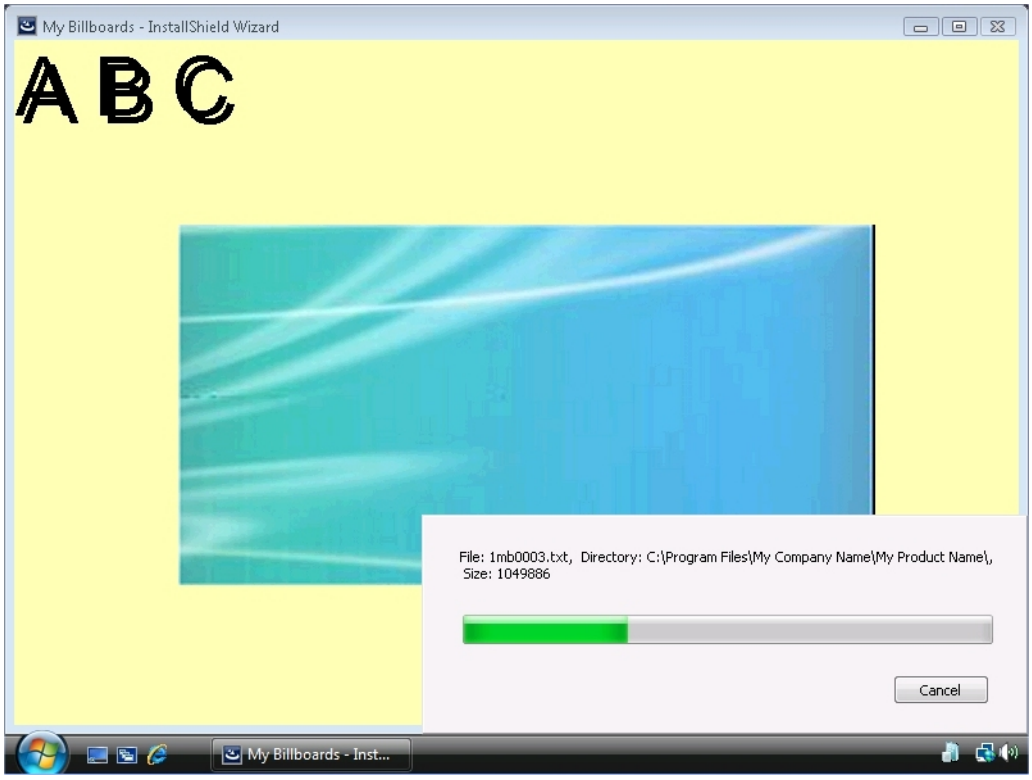


図 4-7: 全画面表示、右下に小さい進行状況ボックスを表示する

サンプル スクリーンショットでは、ビルボードは中央にある青緑色の長方形です。構成可能なビルボード設定の一部は次のように設定されています:

- 原点— 中央揃え
- タイトル — A B C
- フォント — 48 pt. Arial
- 背景色 — 黄色

ウィンドウ表示、標準の進行状況を表示する

ウィンドウ表示、標準の進行状況を表示するタイプののビルボードでは、ファイルの転送中、インストールはビルボードを表示する標準サイズのダイアログを表示します。このダイアログの下の部分に、進行状況バーが表示されます。このスタイルの場合、インストールは背景を表示しません。

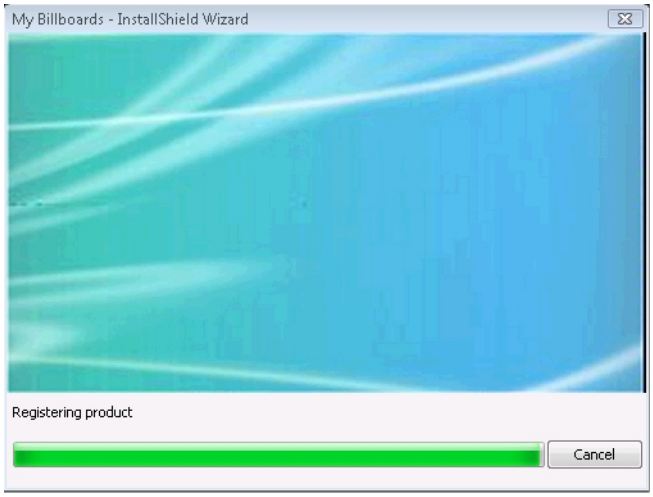


図 4-8: ウィンドウ表示、標準の進行状況を表示する

サンプル スクリーンショットでは、ビルボードは青緑色の長方形です。そのサイズは、幅が 544 ピクセルで、高さが 281 ピクセルです。

ウィンドウ表示、右下に小さい進行状況ボックスを表示する(ビルボードなし)

ウィンドウ表示、右下に小さい進行状況ボックスを表示する(ビルボードなし)タイプのビルボードでは、ファイル転送中にインストールは小さい進行状況ボックスを画面の右下に表示します。ビルボードまたは背景は一切表示しません。

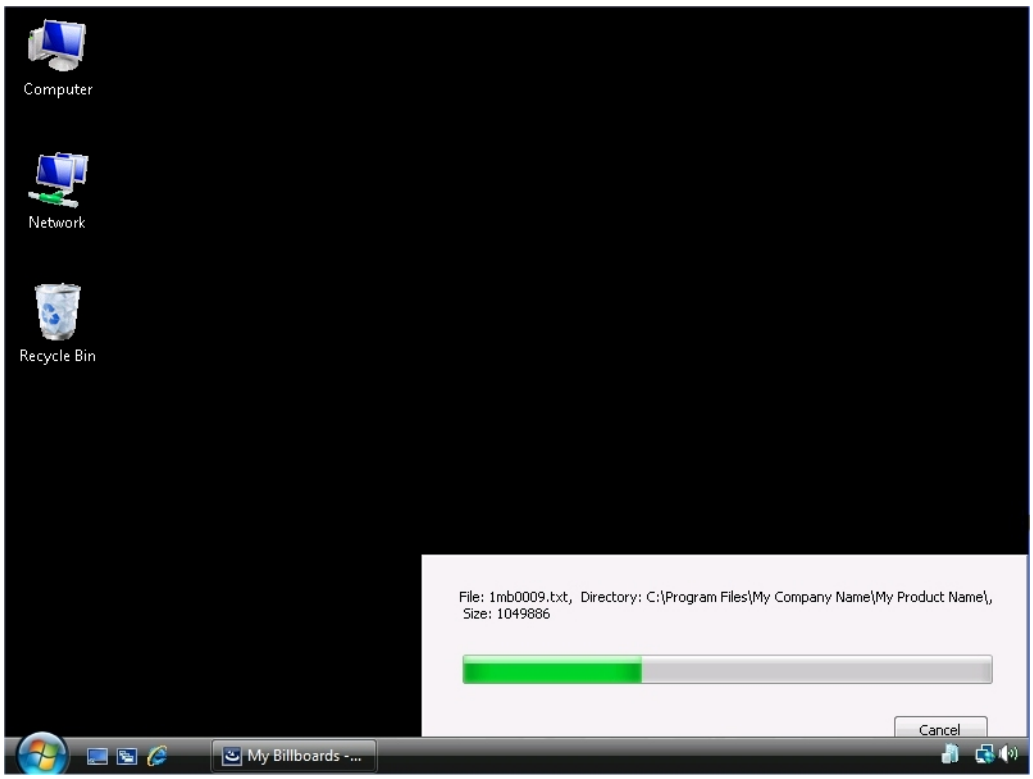


図 4-9: ウィンドウ表示、右下に小さい進行状況ボックスを表示する(ビルボードなし)

サンプル スクリーンショットに見られるように、進行状況バーが表示されますが、ビルボードは表示されません。黒色の背景は、エンド ユーザーのデスクトップです。

インストールで使用するビルボードの種類を指定する

InstallShield は、異なる種類のビルボードをサポートします。

タスク ***インストールで使用するビルボードの種類を指定するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ビルボード**] をクリックします。
2. 中央のペインで、[**ビルボード**] エクスプローラーをクリックします。右側のペインに **"ビルボードの種類"** 設定が表示されます。
3. **"ビルボードの種類"** 設定で、適切なビルボードの種類を選択します。

各ビルボード タイプのサンプルは、「ビルボードの種類」を参照してください。

Adobe Flash アプリケーション ファイル ビルボードを追加する

InstallShield を使って、ファイル転送処理中に Flash アプリケーション ファイル ビルボードを表示できます。Flash アプリケーション ファイルは、ビデオ、動画、音声、インタラクティブ インターフェイス、ゲーム、テキスト、その他の .swf ファイルがサポートするあらゆる要素で構成されます。Flash ビデオ ファイル (.flv) や MP3 オーディオ ファイルは .swf ファイルに埋め込んで、ファイル転送中にターゲット システム上のローカルで使用できるようにすることが推奨されます。.swf ファイルは Web サイト上に配置された外部ファイルを参照することが可能ですが、この外部実装ではエンド ユーザーがインターネットに接続されていることが必須となります。

タスク ***Adobe Flash アプリケーション ファイル ビルボードをインストールに追加するには、以下の手順を実行します。***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ビルボード**] をクリックします。
2. [**ビルボード**] エクスプローラーで、[**Adobe Flash アプリケーション ファイル(.swf)**] を右クリックしてから、[**新しいビルボード**] を選択します。新しいビルボード が **NewBillboard1** という名前で作成されます。
3. ビルボードの名前を入力します。この名前はインストールを作成するときにアイテムを識別するために使用されます。この名前はインストール時には表示されません。
4. 右側のペインで、ビルボードの設定を構成します。

メモ: *.swf ファイルの作成に使用した Flash またはその他のツールのバージョンがターゲット システムにインストールされている Flash Player よりも新しい場合、ターゲット システム上で一部の Flash 機能が予定どおりに動作しない可能性があります。*

イメージ ビルボードの追加

ファイル転送処理中に 1 つのイメージ ビルボードのみを表示したり、一連のイメージ ビルボードで、各ビルボードが特定の時間表示されるよう設計したりすることができます。InstallShield は、.bmp、.gif、.jpg、および .jpeg イメージ ファイルをサポートします。

メモ: *動画 .gif ファイルはサポートされていません。ビルボードで動画を使用したい場合は、Adobe Flashアプリケーション ファイル ビルボードの使用をご検討ください。*

タスク ***イメージ ビルボードをインストールに追加するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ビルボード**] をクリックします。
2. [**ビルボード**] エクスプローラーで、[**イメージ**] を右クリックしてから、[**新しいビルボード**] をクリックします。新しいビルボード が **NewBillboard1** という名前で作成されます。
3. ビルボードの名前を入力します。この名前はインストールを作成するときにアイテムを識別するために使用されます。この名前はインストール時には表示されません。
4. 右側のペインで、ビルボードの設定を構成します。

ビルボード設定の設定を構成する

Adobe Flash アプリケーション ファイル ビルボード、またはイメージ ビルボードをプロジェクトに追加するとき、その設定を構成する必要があります。

タスク ***ビルボードの設定を構成するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ外観のカスタマイズ**] の下にある [**ビルボード**] をクリックします。
2. 中央ペインの [**ビルボード**] エクスプローラーで構成するビルボードを選択します。右側のペインにビルボードが表示されます。
3. 必要に応じて設定を構成します。

ビルボードの各設定についての詳細は、「Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定」を参照してください。

リリースをビルドまたは起動せずにビルボードをプレビューする

InstallShield では、リリースをビルドおよび実行せずに、実行時にビルボードがどのように表示されるのかをプレビューできます。

ビルボードをプレビューすると、そのビルボードに現在構成されている背景色、位置、および関連設定を使ったビルボードの外観を確認できます。

タスク ***ビルボードをプレビューするには、以下の手順を実行します。***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ビルボード]** をクリックします。
2. 中央ペインの **[ビルボード]** エクスプローラーでプレビューするビルボードを右クリックしてから、**[ビルボードのプレビュー]** を選択します。

InstallShield が、実行時に表示されるビルボードのプレビューを表示します。

プレビューを止めるには、[プレビュー] ウィンドウにある [キャンセル] ボタンをクリックします。

ヒント: *ビルボードのプレビューを使うと、Flash またはイメージ ビルボードが、選択された異なるビルボード タイプではどのように表示されるのかを確認するのに特に便利です。ビルボードをプレビューし、ビルボード タイプの変更してから、再度ビルボードをプレビューすることができます。*

ビルボードの順番を設定する

イメージ ビルボードは、[ビルボード] ビューで表示されているのと同じ順序で、上から下に順番に表示されます。

タスク ***イメージ ビルボードが実行時に表示される順番を変更するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ビルボード]** をクリックします。
2. **[ビルボード]** エクスプローラーで、移動するビルボードを 1 つ右クリックし、**[上に移動]** または **[下に移動]** をクリックします。

すべてのビルボードが正しく並べ替えられるまで最後のステップを繰り返します。

ビルボードを含むインストールの実行時の動作

重要: *インストールにビルボードが含まれている場合、インストールには Setup.exe セットアップランチャーを含めなくてはなりません。セットアップランチャーが実行時にビルボードを表示するため、これが必須となります。[リリース] ビューにあるリリースについての Setup.exe タブでは、セットアップランチャーを使用するかどうかなどの情報を指定することができます。詳細については、「Setup.exe タブ」を参照してください。*

インストールに Flash ビルボードと 1 つ以上のイメージ ビルボードが含まれている場合、実行時のファイル転送処理中に Flash ビルボードまたはイメージ ビルボードのうち 1 つのビルボード タイプのみが表示されます。

- Flash Player がターゲット システムに存在する場合、インストールは Flash ビルボードを表示します。
- Flash Player が存在しない場合、インストールはイメージ ビルボードを表示します。

実行時の動作は、インストールが Flash ビルボードかイメージ ビルボードのどちらを表示するかによって、多少異なります:

- **インストールが Flash ビルボードを表示する場合** – ファイル転送が完了すると、インストールは Flash ビルボードに割り当てられた時間が経過するまで、それを表示し続けます。割り当てられた時間が経過すると、インストールはビルボードの表示を終了して、適切な [セットアップの完了] ダイアログを表示します。

  ファイル転送が Flash ビルボードに割り当てられた時間よりも長くかかった場合、インストールはファイル転送が終了するまで、Flash ビルボードを表示し続けます。

- **インストールがイメージ ビルボードを表示する場合** – ファイル転送が完了すると、その他のビルボードがスケジュールされていても、また現在のビルボードに割り当てられた時間が経過していなくても、インストールはイメージ ビルボードの表示を終了します。次に、インストールは適切な [セットアップの完了] ダイアログを表示します。

  ファイルの転送時間が、ビルドボードに割り当てられた時間を超える場合、インストールはファイル転送が終了するまでビルボードを表示し続けます。[ビルボード] ビューの "ビルボードのループ" 設定に [いいえ] が選択されている場合、インストールがファイルの転送を終了する前に最後のビルボードに到達したとき、インストールはファイルの転送が終了するまで最後のイメージ ビルボードを表示し続けます。次に、インストールは適切な [セットアップの完了] ダイアログを表示します。この設定に [はい] が選択されている場合、インストールがファイルの転送を終了する前に最後のビルボードに到達したとき、インストールは最初のビルボードから再び表示を開始します。必要な場合、ファイルの転送が終了して [セットアップの完了] ダイアログが表示されるまでループが継続します。

メモ: *.swf ファイルの作成に使用した Flash またはその他のツールのバージョンがターゲット システムにインストールされている Flash Player よりも新しい場合、ターゲット システム上で一部の Flash 機能が予定どおりに動作しない可能性があります。*

ビルボードの削除

タスク ***インストールからビルボードを削除するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ビルボード]** をクリックします。
2. **[ビルボード]** エクスプローラーで、削除するビルボードを右クリックして **[削除]** を選択します。

メンテナンスおよびアンインストールのためのインストールを作成する

InstallShield では、エンドユーザーがインストールを再実行して、プログラム機能の変更したり、アプリケーションの再インストールまたは削除をすることができます。エンドユーザーがコントロール パネルの [プログラムの追加と削除] でアプリケーションを選択すると、インストールによって以下を実行できるダイアログ ボックスが表示されます。

1. 以前にインストールされなかった機能の個別インストールおよびアンインストール。
2. 最初のインストールで選択された設定でアプリケーションを再インストール
3. アプリケーションのアンインストール

アプリケーションの変更、修正、アンインストールを行う場合、オペレーティング システムにがアプリケーションの存在を認知している必要があります。このため、インストールはオペレーティング システムにアプリケーションを登録し、従って簡単にメンテナンスまたはアンインストールができます。[一般情報] ビューで必要な情報を入力します。詳細については、「インストール情報を指定する」を参照してください。

InstallShield インストールでは、メンテナンス (変更および修正) は自動的に処理されます。また、アンインストールは自動的に処理されますが、唯一の例外は、アンインストール中それ自身で結果を取り消すか、または、アンインストール中のみ実行される別のカスタム アクションによってその影響を取り消させる必要があるカスタム アクションがあるということです。

製品が作成したレジストリデータの削除

デフォルトでは、製品のアンインストーラーは、インストール プログラムが作成したデータのみを削除します。

[レジストリ] ビューで特殊なアンインストール フラグを使用することができます。アンインストール フラグは、アンインストール時に削除するレジストリ データを制御します。特に、レジストリキーで [キー全体をアンインストール] フラグを使用すると、キーおよびそのすべての値とサブキーがアンインストール時に削除されます。詳細については、「レジストリ フラグ」を参照してください。

インストールのビルド、テスト、および配布

インストール プロジェクトの機能、ファイル、ショートカット、レジストリ エントリ、エンドユーザー ダイアログ、およびその他の要素を構成が完了すると、インストールのリリースを作成しビルドできます。リリースをビルドすると、インストールのコンテンツがパッケージされ、配布メディアにコピーして、必要に応じて配布または配置することができるディスク イメージが作成されます。

テストは安定したインストールを作成するために最も重要です。InstallShield では、リリースのエンドユーザー インターフェイスのみを部分的にテスト実行することが可能ですInstallShield のボタンをクリックするだけで、インストールを実行することもできます。この方法でインストールを実行した場合、エンド ユーザーのマシン上とまったく同様にインストールが実行します。ファイルはすべて転送され、ショートカットおよびレジストリ エントリが作成され、ユーザー インターフェイスが表示されます。

インストール作成の最後の手順は、指定場所への配布です。ネットワーク ドライブ、フロッピー ディスクまたはローカル ドライブの別の場所を指定できます。インストールを配布すると、インストールをビルドしたときに作成されたディスク イメージが、指定の場所にコピーされます。

リリースの構成とビルド

InstallShield でプロジェクトのデザインが完了すると、構成およびビルドしてエンドユーザーへ配布できるリリースを作成できます。リリースは、[リリース] ビューで設定したオプションに基づいてビルドされます。

リリースのビルド

ヒント: *以下の手順は、(Visual Studio との統合を行わず) InstallShield 内部でビルドされたリリースに適用します。Visual Studio 内部から InstallShield リリースをビルドする方法については、「Microsoft Visual Studio でリリースをビルドする」を参照してください。*

タスク ***リリースをビルドするには、以下の手順に従います:***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、ビルドするメディアの種類をクリックします。
3. ビルドの設定を編集します。
4. 選択したメディアの種類を右クリックして、**[ビルド]** をクリックします。

IsCmdBld.exe を使って、コマンドラインからリリースをビルドすることもできます。

InstallShield では、ビルドしたインストール パッケージをプロジェクトの場所に基づいてリリースの場所に配置します。ビルドに失敗してエラーが生成された場合、以前のインストール パッケージがあるときは、¥DiskImages¥Disk1 フォルダーに復元されます。

ヒント: *リリースをビルドする際、Windows エクスプローラーが Disk1 フォルダーまたはサブフォルダーを指定していないことを確認してください。Disk1 フォルダーが指定されていると、ビルド処理は終了しません。エクスプローラーがサブフォルダーにアクセスしようとすると、エラーが発生します。*

非圧縮のビルドを作成する場合、特定のオペレーティング システムの名前を付けた追加のフォルダーをインストールに含めることができます。これによって、特定のオペレーティング システム用のファイルが、別のオペレーティング システム対象のファイルと同じ名前を持つ場合にも対応することができます。

セットアップランチャーの作成

InstallShieldでは、インストールに Setup.exe セットアップランチャーを含めるかどうかを指定できます。Setup.exe セットアップランチャーは、次のような場合に必要です:

- 必要に応じて、自動的にターゲット システムの Windows Installer エンジンを更新またはインストールする。
- プロジェクトに InstallShield 前提条件が含まれている場合。
- プロジェクトに .NET Framework が含まれている場合。
- プロジェクトにビルボードが含まれている場合。

Setup.exe セットアップランチャーは、上記のシナリオを制御するブートストラップ アプリケーションです。

[リリース] ビューにあるリリースの Setup.exe タブでは、Setup.exe 起動ツールを使用するかどうかなどの情報を指定することができます。詳細については、「Setup.exe タブ」を参照してください。

Windows Installer と Setup.exe

Windows Installer がターゲット システムに存在しない可能性がある、または、インストールが特定のバージョンの Windows Installer でのみ使用可能な機能に依存している可能性がある場合、InstallShield では、Windows Installer をインストールする再配布可能ファイルをインストールに含めるオプションが提供されます。このオプションを選択すると、InstallShield はターゲット システム上で Windows Installer の存在を確認する Setup.exe 起動ツールを作成します。Windows Installer がインストールされていない場合や、または新しいバージョンをインストールする必要がある場合、 Setup.exe は Windows Installer インストールを起動してからインストール パッケージを起動します。

詳細については、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。

InstallShield 前提条件と Setup.exe

InstallShield 前提条件を含むプロジェクトでは、Setup.exe によってターゲット システムが InstallShield 前提条件を満たしているかどうかが確認されるため、Setup.exe が必須になります。条件が満たされた場合、Setup.exe は InstallShield 前提条件をインストールします。詳細については、「プロジェクトに含まれている InstallShield 前提条件を使って作業する」を参照してください。

.NET Framework と Setup.exe

.NET Framework を含むプロジェクトでは、Setup.exe がターゲット システムを検索して .NET Framework が存在するかどうかを確認するため、Setup.exe が必要です。適切なバージョンの .NET Framework が存在しない場合、Setup.exe によってそのバージョンがインストールされます。

.NET Framework の追加については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。

ビルボードと Setup.exe

ビルボードは、実行時に Setup.exe によって表示されるため、ビルボードを含むプロジェクトには、Setup.exe が必要です。

ビルボードに関する詳細については、「ビルボードを表示する」を参照してください。

セットアップランチャーのファイルのプロパティをカスタマイズする

InstallShield では、Setup.exe セットアップ起動プログラムのバージョン リソースにカスタム情報を使用できます。情報が、セットアップ起動プログラムの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンドユーザーが Setup.exe ファイルが右クリックして、[プロパティ] をクリックしたときに表示されます。

Setup.exe のプロパティを構成するための InstallShield の設定

次のテーブルは、Windows の [前提条件] ダイアログ ボックスに含まれている様々なプロパティと、それらを構成するのに使用できる InstallShield で対応する設定の一覧です。

メモ: *[プロパティ] ダイアログ ボックスは、Windows のバージョンによって異なります。たとえば、Windows 7 システムでは、バージョン リソース情報は [プロパティ] ダイアログ ボックスの [詳細] タブに表示されます。一方、Windows XP システムでは、バージョン リソース情報は、同じダイアログボックスの [バージョン] タブに表示されます。*

また、Windows の一部のバージョンは、[プロパティ] ダイアログ ボックスの一部の設定を表示しません。

テーブル 4-1 ・ Setup.exe のプロパティの情報ソース

Setup.exe プロパティ	Setup.exe プロパティを構成するための InstallShield 設定
会社名	[一般情報] ビューの "発行者" 設定
製品名	[一般情報] ビューの "製品名" 設定
製品バージョン	[一般情報] ビューの "製品バージョン" 設定に入力された製品バージョンに初期化します。
ファイル バージョン	[リリース] ビューで製品構成の "ファイル バージョン" 設定この設定が空白の場合、InstallShield は [一般情報] ビューの "製品バージョン" 設定を使います。 ***メモ:*** *ファイル バージョンは、常に 4 つのフィールドが含まれます。"ファイルバージョン" に 4 フィールドよりも少ないフィールドを指定すると、残りのフィールドには 0 が挿入されます。たとえば、ファイル バージョンとして 1.1 を指定すると、Setup.exe のバージョン リソースで使用されるファイル バージョンは 1.1.0.0 となります。*

テーブル 4-1・Setup.exe のプロパティの情報ソース (続き)

Setup.exe プロパティ	Setup.exe プロパティを構成するための InstallShield 設定
著作権情報	InstallShield は、提供されたカスタム値、またはデフォルトの InstallShield 著作権情報を使用します。 "著作権情報" フィールドの値に独自の値を使用するには、以下の手順に従います: 1. **[リリース]** ビューで、構成するリリースを選択します。 2. **"カスタム バージョンのプロパティを使用する"** 設定での **Setup.exe** タブで **[はい]** を選択します。 3. **"起動ツールの著作権"** 設定に、Setup.exe ファイルのプロパティ内の "著作権" フィールドに使用するテキストを入力します。 "カスタム バージョンのプロパティを使用する" 設定に [いいえ] を選択するか、"起動ツールの著作権" 設定を空白のままにすると、InstallShield はデフォルトの InstallShield 著作権情報を使用します。
ファイルの説明	InstallShield は、提供されたカスタム値またはデフォルトの InstallShield Setup.exe の説明を使用します。 "説明" フィールドの値に独自の値を使用するには、以下の手順に従います: 1. **[リリース]** ビューで、構成するリリースを選択します。 2. **"カスタム バージョンのプロパティを使用する"** 設定での **Setup.exe** タブで **[はい]** を選択します。 3. **"ファイルの説明"** 設定に、Setup.exe ファイルのプロパティ内の "説明" フィールドに使用するテキストを入力します。 "カスタム バージョンのプロパティを使用する" 設定に [いいえ] を選択すると、InstallShield はデフォルトの InstallShield Setup.exe の説明を使用します。[はい] を選択して、"説明" 設定を空白のままに残すと、InstallShield は [一般情報] ビューの "[概要情報ストリーム] のコメント" 設定を使用します。この設定が空白の場合、InstallShield はデフォルトの InstallShield Setup.exe の説明を使用します。
言語	InstallShield はこのフィールドに、プロジェクトで指定された言語を使用します。

Setup.exe のプロパティ ダイアログ ボックスのサンプル

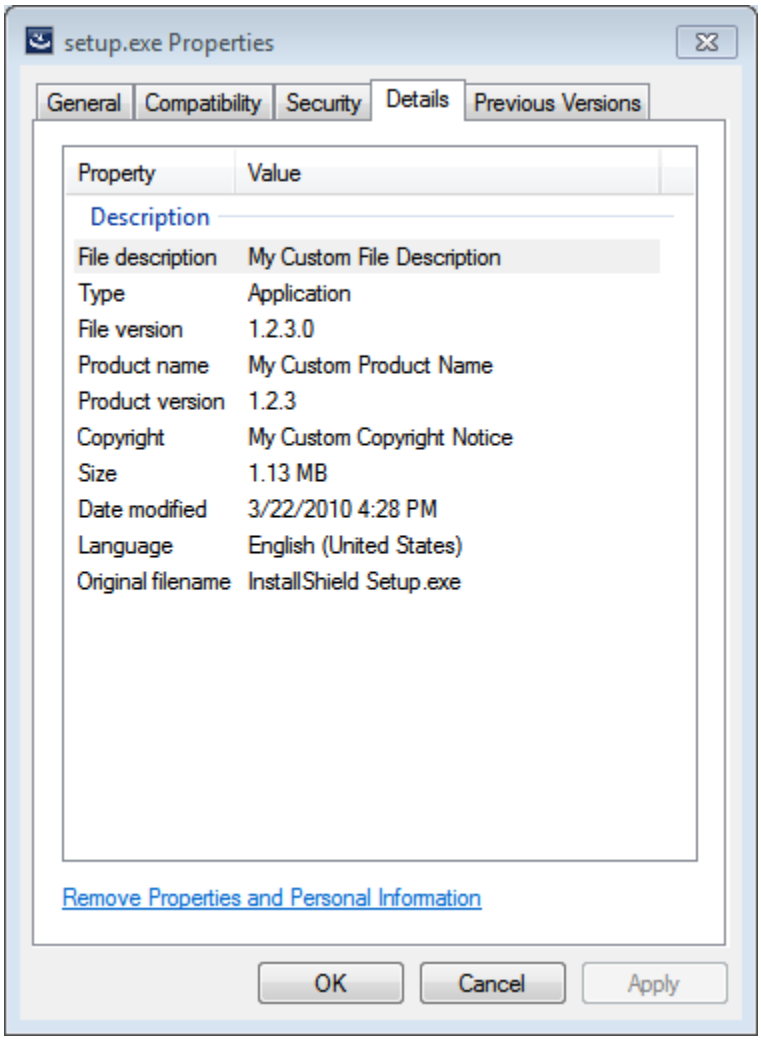


図 4-1: Windows 7 マシン上における Setup.exe のプロパティのサンプル

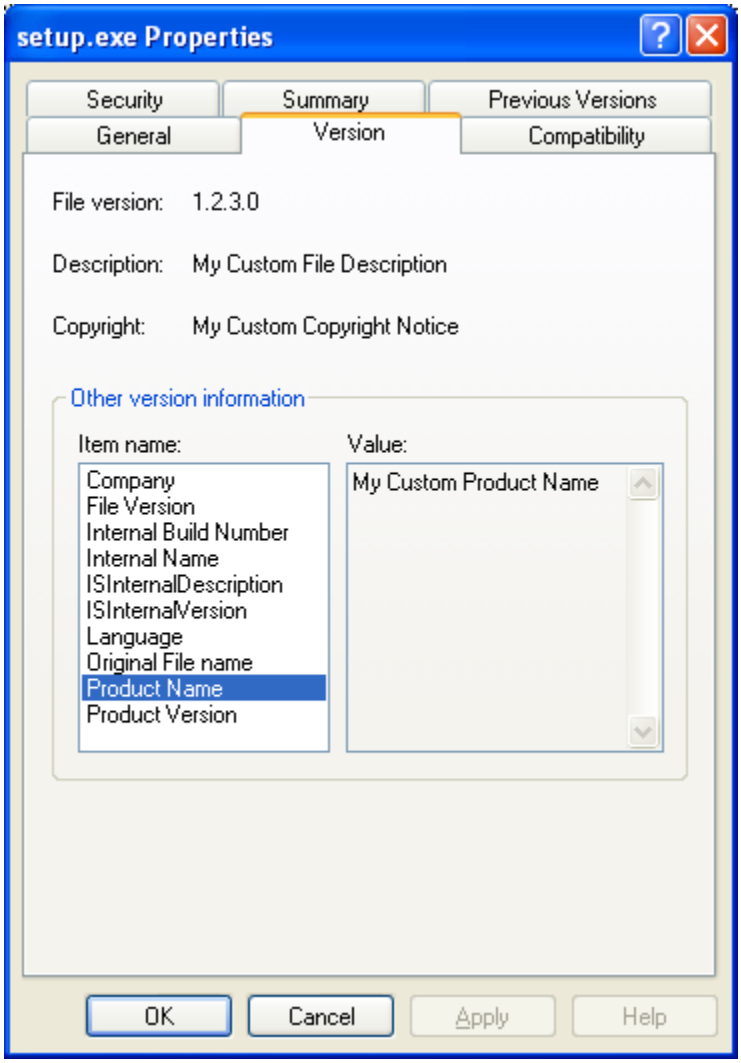


図 4-2: Windows XP マシン上における Setup.exe のプロパティのサンプル

64 ビット セットアップ ランチャーのビルド

InstallShieldでは、インストールに 64 ビット**Setup.exe** セットアップランチャーを含めるかどうかを指定できます。64 ビット **Setup.exe** セットアップ ランチャーは、インストールが 64 ビット システムをターゲットとする場合のみ必要です。デフォルトで、InstallShield は 32 ビット システムだけでなく 64 ビットシステム上の WOW64 上で実行可能な 32 ビット セットアップ ランチャーをビルドします。一部の 64 ビット ターゲット システム、たとえば Windows Server Core システム- は、WoW64 (32-bit Windows-on-Windows) をサポートしない場合があります。これらの 64 ビット ターゲット システムは 32 ビット セットアップランチャーを実行できません。

[**リリース**] ビューの [**全般**] タブで、64 ビット セットアップ ランチャーをビルドするための情報を指定できます。

64 ビット セットアップ ランチャーをビルドしたとき、対応する 64 ビット MSI パッケージがデフォルトでビルドされることはありません。64 ビット MSI パッケージをビルドするには、適切な 64 ビット値 (x64 または Intel64) を使って "テンプレート概要" を構成する必要があります。InstallShield Express Edition は、Template Summary プロパティを自動的に構成します。ピュア 64 ビット MSI パッケージは、インストール中に 32 ビット製品ファイルをコピーまたは 32 ビット カスタム アクションを実行することができません。これは、32 ビット バイナリを WOW64 サポートなしで 64 ビット ターゲット システム上にロードすることができないためです。

64 ビット MSI パッケージのビルドについての詳細は、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。

64 ビット セットアップ ランチャーを含む 32 ビット MSI パッケージをビルドする

MSI パッケージで 64 ビット セットアップ ランチャー オプションを含む 32 ビット パッケージのビルドが構成された場合、InstallShield は次の警告メッセージを表示します:

ISEXP: 警告 -7372: セットアップ ランチャーでは 64 ビットのビルドが構成されていますが、テンプレートの概要では 64 ビット MSI パッケージのビルドが構成されていません。

この警告は、64 ビット セットアップ ランチャーがピュア 32 ビット システム上での実行に失敗し、32 ビット MSI パッケージがピュア 64 ビット システム上でのインストールに失敗するために発生します。このエラーを回避するため、インストーラーがピュア 32 ビット システムおよび 64 ビット システム上の WOW64 をターゲットにする必要があるときは、64 ビット セットアップ ランチャーを使用しないでください。または、MSI パッケージで、ピュア 64 ビット システムをターゲットにする 64 ビット MSI パッケージを生成するように構成してください。

InstallShield 前提条件および 64 ビット セットアップ ランチャー

一般的に InstallShield 前提条件を含むプロジェクトでは、**Setup.exe** によってターゲット システムで InstallShield 前提条件が満たされているかどうかが確認されるため、**Setup.exe** が必須になります。条件が満たされている場合、**Setup.exe** が InstallShield 前提条件をインストールします。

64 ビット セットアップ ランチャーを使用する InstallShield 前提条件を含むプロジェクトのビルドを構成する場合、必ず前提条件のインストーラーが 64 ビットであり、ピュア x64 システムにインストールが可能で、前提条件がレジストリまたはフォルダーの場所で 64 ビットの場所を確認するように構成します。

場合によって、64 ビット セットアップ ランチャーを使う 32 ビット前提条件のインストーラーをビルドしたい場合、32 ビットの場所を確認する条件を構成します。前提条件に含まれるほとんどのレジストリ条件はデフォルトの場所を確認するように構成されています (32 ビット セットアップ ランチャーの場合は 32 ビットの場所、64 ビット セットアップ ランチャーの場合は 64 ビットの場所)。ただし、32 ビットの前提条件インストーラーは、32 ビット レジストリハイブの下にレジストリ キーを作成します。この問題を解決するためには、32 ビットの場所を確認するようにレジストリの場所を明示的に変更する必要があります。

ビルドのキャンセル

タスク ***既に起動したビルドをキャンセルするには、以下の手順に従います:***

ツールバー上の **[ビルドの中止]** ボタンをクリックします。

ビルド中に製品バージョンを変更する

InstallShield では、ビルド時にインストールの製品バージョンを変更する複数の方法があります。

- ISCmdBld.exe を使ってコマンドラインからインストールをビルドする場合、-y コマンドライン パラメーターを使って製品バージョンを指定します。詳細については、「IsCmdBld.exe」を参照してください。
- MSBuild または Team Foundation Server を使ってインストールをビルドする場合、InstallShield タスクで ProductVersion パラメーターを使います。詳細については、「Microsoft ビルド エンジン (MSBuild)」を参照してください。

リリースのビルド場所

インストールのディスク イメージ フォルダーは、リリースの場所にビルドされます。圧縮されていないアプリケーション ファイルの保存に必要な、すべての補足的なファイルやフォルダーは、ディスク イメージ フォルダーのサブフォルダーに配置されます。リリースの場所は、プロジェクトの場所のサブフォルダーです。

リリースは次のフォルダーにビルドされます:

<プロジェクトの場所>¥<プロジェクト名>¥Express¥<リリースの種類>¥DiskImages¥Disk1

デフォルトのプロジェクトの場所:

C:¥InstallShield 2021 Projects

たとえば、現在のプロジェクト名が CoolProject で、リリースのメディア タイプが CD-ROM であった場合、Disk1 フォルダーの場所は次のようになります:

C:¥InstallShield 2021 Projects¥CoolProject¥Express¥Cd_rom¥DiskImages

非圧縮のビルドを作成する場合、特定のオペレーティング システムの名前を付けた追加のフォルダーをインストールに含めることができます。これによって、特定のオペレーティング システム用のファイルが、別のオペレーティング システム対象のファイルと同じ名前を持つ場合にも対応することができます。

ビルド ログとレポート

リリースをビルドすると、ログとビルド レポートが毎回生成されます。ログには、ビルド プロセス中に出力ウィンドウに表示されるのと同じ情報が含まれます。ビルド レポートにはビルドの正確な概要、すべての機能、セットアップの種類、マージ モジュール、ダイナミック リンク、およびビルドに含められたファイルの一覧が含まれます。ログとレポートの両方が生成され、リリースをビルドするたびにタイム スタンプが追加されるため、セットアップの内容の手動検証や記録管理にこれらを使用できます。

クイック ビルド

インストールをテストするとき、ファイルに変更がなければインストールのすべてを頻繁にビルドする必要はありません。インストールの最初の完全ビルドを実行した後は、InstallShield の [クイック ビルド] オプションを使用することができます。[クイックビルド] オプションはインストールの .msi ファイル部分のみを再ビルドし、ビルド処理を短くすることができます。これは文字列またはダイアログをテストまたは修正するときに便利です。

メモ: *SingleImage メディア タイプでは [クイック ビルド] オプションはサポートされていません。これは .msi ファイルが* Setup.exe *ファイルに含まれているためです。また、WebDeployment メディア タイプでも [クイック ビルド] オプションはサポートされていません。これは .msi ファイルが .cab ファイルに含まれているためです。*

クイック ビルドを実行する

タスク ***クイック ビルドを実行するには、以下の手順に従います:***

[**ビルド**] メニューで [**クイックビルド**] をクリックします。

メモ: *SingleImage メディア タイプでは [クイック ビルド] オプションはサポートされていません。これは .msi ファイルが* Setup.exe *ファイルに含まれているためです。また、WebDeployment メディア タイプでも [クイック ビルド] オプションはサポートされていません。これは .msi ファイルが .cab ファイルに含まれているためです。*

コマンドラインを使ったビルド

IsCmdBld.exe を使って、コマンドラインからリリースをビルドすることができます。この方法を使ったインストールのビルドは、バッチ ファイルからビルドをするときに便利です。

IsCmdBld.exe は、デフォルトで次の場所にあります。

InstallShield Program Files フォルダー¥System

ISCmdBld.exe は、インストールされた場所から移動しないでください。

コマンドラインから IsCmdBld.exe へ渡されたパラメーターは、リリース タイプの設定をオーバーライドします。

コマンドラインからのビルド

タスク ***コマンドラインからインストールをビルドするには、以下の手順に従います:***

1. [コマンドライン プロンプト] ウィンドウを開きます。
2. ディレクトリを次のように変更します。

   InstallShield Program Files フォルダー¥System

3. コマンドライン実行可能ファイル (**IsCmdBld.exe**) の名前と必要なパラメーターを入力します。

次に、IsCmdBld.exe を実行してリリースをビルドするステートメントの例を示します。

IsCmdBld.exe -p "C:¥InstallShield 2021 Projects¥MyProject1.ise" -c COMP -e y

-p で始まる最初のパラメーターは、ビルドする InstallShield プロジェクト ファイル (.ise) ファイルへのパスです。2 番目のパラメーター -c COMP は、パッケージを単一ファイルに圧縮するかどうか指定します。最後のパラメーター -e y は、ビルドに Setup.exe を含めるように指定します。

.ini ファイルでコマンドライン ビルド パラメーターを渡す

コマンドラインを使ったビルド中に多くのパラメーターを渡す場合や、同じパラメーターを継続的に渡す場合、.ini ファイルを使うと便利な場合があります。以下は、IsCmdBld.exe を実行して、**MySetup.ini** ファイルで指定したパラメーターと共に リリースをビルドするステートメントの例を示します。

ISCmdBld.exe -i "C:¥InstallShield 2021 Projects¥MySetup.ini"

コマンドラインでパラメーターを渡すときと同じ情報を .ini ファイルに含める必要があります。このファイルには 4 つのセクションがあります:

- [Project]—このセクションには、製品構成の名前およびプロジェクト ファイル (.ise) へのパスの エントリを含めます。パッチをビルドしている場合、ビルド中のパッチ構成の名前のエントリを含めます。
- [Release]—このセクションには、圧縮の種類 (圧縮または非圧縮)、ビルド フラグ、Setup.exe の設定、およびリリース名などのリリース構成情報のエントリを含めます。
- [Mode]—このセクションには、Silent=yes (リリースのビルド中に、ビルド エラーまたは警告メッセージを抑制する場合) のような、使用可能なオプション エントリを含めます。このセクションでは、ログ ファイルを作成するかどうかを指定することもできます。
- [BuildLocation]—このセクションでは、リリースの出力場所をオプションで指定することができます。

一部のセクションは必須ではありません。コマンドラインから直接パラメーターを渡すことに関しては、サイレント ビルドやビルドの場所などの要件のためのパラメーターはオプションです。下の .ini ファイルの例では、これらのパラメーターは [Mode] セクションと [BuildLocation] セクションにあります。デフォルトを使用する場合は、.ini ファイルからこれらのエントリを省略してください。デフォルトでは、ログ ファイルは作成されず、インストールもサイレント モードで実行されません。また、リリースは [オプション] ダイアログ ボックスの [ファイルの場所] タブで指定されたプロジェクトの場所に作成されます。

サンプル .ini ファイル

次のテーブルは、サンプル .ini ファイルに含まれている 4 つのセクションからのサンプル エントリです。サンプル エントリは、各テーブルの最初の列に表示されています。その他の列は、対応するコマンドライン パラメーターと説明です。

[Project] セクションのエントリ

テーブル 4-2 ・ サンプル エントリ – [Project] セクション (.ini ファイル)

エントリ	対応する ISCmdBld.exe コマンドライン パラメーター	説明
Name="C:¥InstallShield 2021 Projects¥Othello.ise"	-p	.ise ファイルへのパス。
Product=Othello		プロジェクトの名前。このエントリを使って、[一般情報] ビューで指定された値をオーバーライドできます。

[Release] セクションのエントリ

テーブル 4-3 ・ サンプル エントリ – [Release] セクション (.ini ファイル)

エントリ	対応する ISCmdBld.exe コマンドライン パラメーター	説明
Configuration=COMP	-c	圧縮、非圧縮の違い。
Name=Othello Beta	-r	リリース名。
SetupEXE=yes	-e	Setup.exe ファイルを作成します。

[Mode] セクションのエントリ

テーブル 4-4 ・ サンプル エントリ – [Mode] セクション (.ini ファイル)

エントリ	対応する ISCmdBld.exe コマンドライン パラメーター	説明
Silent=yes	-s	サイレント モードで実行します。
MergeModulePath="C:¥..."	-o	マージ モジュール (.msm ファイル) のパスを検索します。 詳細については、「マージ モジュールを含むディレクトリを指定する」を参照してください。
PrerequisitePath="C:¥..."	-prqpath	InstallShield 前提条件ファイル (.prq) の検索パス。 詳細については、「InstallShield 前提条件を含むディレクトリを指定する」を参照してください。

Microsoft ビルド エンジン (MSBuild)

InstallShield は、.NET Framework に含まれている Microsoft ビルド エンジン (MSBuild) をサポートします。MSBuild サポートを利用して、Visual Studio がインストールされていないビルド ラボ環境で InstallShield プロジェクトと共に Visual Studio ソリューションをビルドすることができます。

概要

MSBuild は、Visual Studio に依存するビルドを削除するように設計された拡張ビルド フレームワークです。.NET Framework を使用すると、コマンドライン、または MSBuild のすべてのホストからプロジェクトまたはソリューションをビルドすることができます。MSBuild についての詳細は、「MSDN ライブラリ」を参照してください。

MSBuild のタスク

MSBuild の柔軟性および拡張性は、タスクと呼ばれる細かくグループ分けされた内部ビルド手順を通して制御されます。MSBuild と共に発送されるタスクの 1 つは Csc と呼ばれ、Visual C# プロジェクトからのコードをコンパイルすることができます。InstallShield は、InstallShield という名前の MSBuild タスクをインストールします。このタスクは、InstallShield プロジェクトと、プロジェクトのデフォルトのビルド ステップを提供するターゲット ファイルをビルドします。このカスタマイズされたタスクとターゲット ファイルによって、MSBuild は InstallShield プロジェクトを Visual Studio ソリューションの一部としてビルドするのに必要なすべてのアクションを実行することができます。

以下のテーブルは、InstallShield タスクのパラメーターの説明です。

テーブル 4-5 ・ MSBuild InstallShield タスク

パラメーター	種類	説明
InstallShieldPath	文字列	このパラメーターは、InstallShield アプリケーションを含むフォルダーへのパスを指定します。
プロジェクト	文字列	このパラメーターは、プロジェクト ファイル (.ise) の場所を指定します。
ProductConfiguration	文字列	このパラメーターは、リリースの製品構成を指定します。このパラメーターの値には、**Express** が入ります。
ReleaseConfiguration	文字列	このパラメーターは、リリース名を指定します。有効なオプションは次の通りです。 ・ カスタム ・ CD_ROM ・ DVD-10 ・ DVD-18 ・ DVD-5 ・ DVD-9 ・ SingleImage ・ WebDeployment

テーブル 4-5 • MSBuild InstallShield タスク (続き)

パラメーター	種類	説明
PatchConfiguration	文字列	このパラメーターは、InstallShield Premier または InstallShield でビルドされる パッチ構成の名前を指定します。 ***エディション***• *パッチの構成は、InstallShield Premier および InstallShield でポートされていますが、Express Edition ではサポートされていません。このため、PatchConfiguration パラメーターは Express Edition では使用されません。*
OutDir	文字列	このパラメーターは、出力先フォルダーとファイルの保存先フォルダーへの完全修飾パスを指定します。
MergeModulePath	String[ ]	このパラメーターは、プロジェクトで参照するマージ モジュール (.msm ファイル) を含むフォルダーを指定します(複数指定可)。 InstallShield では、マージ モジュールを含むフォルダーを指定するその他の方法も提供されています。詳細については、「マージ モジュールを含むディレクトリを指定する」を参照してください。
PrerequisitePath	String[ ]	このパラメーターは、プロジェクトで参照される InstallShield 前提条件ファイル (.prq ファイル) を含むフォルダーをセミコロンで区切って指定します(複数指定可)。 InstallShield では、InstallShield 前提条件ファイル ファイルを含むフォルダーを指定するその他の方法も提供されています。詳細については、「InstallShield 前提条件を含むディレクトリを指定する」を参照してください。
ReleaseFlags	String[ ]	このパラメーターを使用して、リリースに含めるリリース フラグを指定できます。複数のフラグはカンマで区切ります。 ***エディション***• *リリース フラグは InstallShield Premier および InstallShield でサポートされていますが、Express Edition ではサポートされていません。このため、ReleaseFlags パラメーターは Express Edition では使用されません。*

テーブル 4-5・MSBuild InstallShield タスク（続き）

パラメーター	種類	説明
PathVariables	ITaskItem [ ]	このパラメーターを使用して、プロジェクトのパス変数をオーバーライドすることができます。このパラメーターを使用して、パス変数へのパスを指定したり、PathVariable サブ要素を使用して、パス変数の名前の値を定義することができます。 同じ Visual Studio ソリューション フォルダー内にある姉妹プロジェクトのファイルを参照するには、VSSolutionFolder と呼ばれる定義済みのパス変数を使用します。詳細については、「Visual Studio ソリューションで VSSolutionFolder パス 変数を使用する」を参照してください。 MSBuild ITaskItem[ ] プロパティについての詳細は、MSDN ライブラリを参照してください。 ***エディション**・このパラメーターは、InstallShield Premier および InstallShield で使用できます。*
PreprocessorDefines	ITaskItem [ ]	このパラメーターを使用して、プロジェクトのプリプロセッサ定義を追加またはオーバーライドすることができます。このパラメーターを使用して、プリプロセッサ定義の定義を指定したり、Token サブ要素を使用して、プリプロセッサ定義の名前の値を定義することができます。 MSBuild ITaskItem[ ] プロパティについての詳細は、MSDN ライブラリを参照してください。 ***エディション**・このパラメーターは、InstallShield Premier および InstallShield で使用できます。*
OutputGroups	ITaskItem [ ]	このパラメーターは、Visual Studio からのプロジェクト出力グループを指定します。このパラメーターを使用して、プロジェクト出力グループのソースの場所へのパスを指定したり、次のリストにあるサブ要素を使用して、これらの追加の値を定義することができます。 ・ Name—プロジェクトの名前 ・ OutputGroup—プロジェクト出力グループの名前 ・ TargetPath—プロジェクト出力グループのターゲット パス（ソースの場所とは異なります） MSBuild ITaskItem[ ] プロパティについての詳細は、MSDN ライブラリを参照してください。

テーブル 4-5 ・ MSBuild InstallShield タスク (続き)

パラメーター	種類	説明
ビルド	文字列	このパラメーターは、実行するビルドのタイプを指定します。次のビルドのタイプから選択することができます。 ・ Complete—完全ビルドを生成するとき、この値を指定します。 ・ Tables—完全ビルドを既に実行していて、インストールの .msi ファイルの部分のみを再ビルドするとき、この値を指定します。このタイプのビルドはクイック ビルドと呼ばれます。このタイプのビルドは、SingleImage タイプまたは WebDeployment タイプのリリース構成では実行できません。 ***エディション・*** *Compile、TablesAndFiles、および UpgradeOnly は、InstallShield Premier および InstallShieldで提供されている追加の Build の値です。*
BuildCompressed	ブール値	このパラメーターは、リリースを単一ファイルに圧縮するか、または圧縮せずに複数ファイルとして残すかを指定します。
BuildSetupExe	ブール値	このパラメーターは、インストールと同時に Setup.exe ファイルを作成するかどうかを指定します。
ProductVersion	文字列	このパラメーターは、製品バージョンを指定します。これは、製品バージョンのビルド バージョン (3 番目のフィールド) を増加するときに、特に便利です。 有効な製品バージョン番号については、「製品バージョンを指定する」を参照してください。
PropertyOverrides	ITaskItem[ ]	このパラメーターを利用すると、Windows Installer プロパティの値をオーバーライドしたり、プロパティが存在しないとき、それを新規作成したりできます。このパラメーターを利用するには、値に新しいプロパティ値があり、かつメタデータ Property がそのプロパティの名前であるアイテムのプロパティ一覧を含めます。 このパラメーターは、InstallShield タスクの PropertyOverrides プロパティに InstallShieldPropertyOverrides ItemGroup パススルーとして露出されます。

テーブル 4-5 ・ MSBuild InstallShield タスク (続き)

パラメーター	種類	説明
RunMsiValidator	文字列	このパラメーターを使用して、.cub ファイルを使って、.msi パッケージを検証することができます。 ***エディション・*** *このパラメーターは、InstallShield Premier および InstallShieldで使用できます。*
RunUpgradeValidation	ブール値	このパラメーターは、アップグレードの検証を実行するかしないかを指定します。 ***エディション・*** *このパラメーターは、InstallShield Premier および InstallShieldで使用できます。*
StopOnFirstError	ブール値	このパラメーターは、一番最初のエラーが発生した時点で、ビルドを停止するかどうかを指定します。
MsiVersion	文字列	このパラメーターは、ターゲット マシン上でインストールが許可する Windows Installer の最小バージョンを指定します。
DotNetFrameworkVersion	文字列	このパラメーターは、ターゲット マシン上でインストールが許可する .NET Framework の最小バージョンを指定します。
DotNetUtilPath	文字列	Regasm.exe と InstallUtilLib.dll は .NET Framework の各バージョンに含まれるユーティリティです。このパラメーターは、.NET インストーラー クラスおよび COM Interop を含むリリースのビルド時に使用する、これらのファイルの 32 ビット バージョンを含むディレクトリのパスを指定します。これは、.NET Framework 再配布可能ファイルへのパスではありません。
Disk1Folder	出力文字列	このパラメーターは、出力ファイルの場所を指定します。
SummaryInfoComments	文字列	ビルド番号を含めるなど、ビルド時に **概要情報ストリーム** コメントを設定するには、このパラメーターを使用します。 このプロパティを使って追加されたコメントは、ビルド済みのインストーラーの **[プロパティ]** ダイアログ ボックスで参照することができます。
MSIPackageFileName	文字列	このパラメーターを使ってビルド時に MSI パッケージ ファイル名を指定します。.msi ファイルに使用するファイル名を指定します (ピリオドまたは拡張子は除きます)。

MSBuild スクリプト

MBuild は、1 つのファイル形式 (XML ビルド スクリプト) を理解します。XML ビルド スクリプトは、Visual C# および Visual Basic .NET プロジェクト (.csproj および .vbproj) のプロジェクト ファイル形式として使用されます。MSBuild はまた、ソリューション ファイルおよび Visual C++ プロジェクト ファイル フォーマット (.sln と .vcproj) を処理するための内部フックも備えています。

InstallShield の Visual Studio との統合では、MSBuild と互換性のある XML フォーマット プロジェクト ファイル (.isproj) が使用されています。これにより、MSBuild は、InstallShield プロジェクトを含む Visual Studio ソリューションをシームレスにビルドすることができます。

.isproj ファイルのカスタマイズ

InstallShield プロジェクトの変更を .isproj ファイルに反映させるには、.isproj ファイルの上部に PropertyGroup 要素または ItemGroup 要素 を追加するか、または既存の PropertyGroup または ItemGroup 要素を更新します。そのあと、InstallShield 関連のパラメーターを必要に応じて追加します。

.isproj ファイルからの、以下のサンプルコードは、次の処理を行う方法を説明します:

- 製品バージョンを設定する。
- 製品名を設定する。
- カスタム パブリック プロパティ MY_PROPERTY を *My Value* に設定する。
- InstallShield 前提条件に以下の検索パス、<ISProductFolder>¥SetupPrerequisites および <ISProjectFolder>¥MyCustomPrerequisites を指定する。

```
<PropertyGroup>
  <InstallShieldProductVersion>1.2.3</InstallShieldProductVersion>
</PropertyGroup>
<ItemGroup>
  <InstallShieldPropertyOverrides Include="新しい製品">
    <Property>ProductName</Property>
  </InstallShieldPropertyOverrides>
  <InstallShieldPropertyOverrides Include="値">
    <Property>MY_PROPERTY</Property>
  </InstallShieldPropertyOverrides>
  <InstallShieldPrerequisitePath Include="&lt;ISProductFolder&gt;¥SetupPrerequisites"/>
  <InstallShieldPrerequisitePath Include="&lt;ISProjectFolder&gt;¥MyCustomPrerequisites"/>
</ItemGroup>
```

MSBuild を使用して、コマンドラインからリリースをビルドする

メモ: *MSBuild を使って、InstallShield プロジェクトを含む Visual Studio ソリューションをビルドするには、.NET Framework 3.5 以降が必要です。*

MSBuild により、Visual Studio がインストールされていないマシン上で、簡単にコマンドラインからリリースをビルドすることができます。唯一マシンにインストールされている必要のあるコンポーネントは、.NET Framework と InstallShield です。マシン上に Visual Studio ソリューションのコピーを作成して、MSBuild を実行します。

タスク ***コマンドラインから MSBuild を使用するには、以下の手順に従います。***

1. [コマンドライン プロンプト] ウィンドウを開きます。
2. MSBuild.exe を含むディレクトリへ変更します:

   C:¥Windows¥Microsoft.NET¥Framework¥*Version Folder*¥

3. Visual Studio 統合プロジェクトのリリース ビルドを行うコマンドライン ステートメントを入力します。例:

   MSBuild.exe C:¥*My Visual Studio Solution を含むフォルダー*¥*My Solution*.sln /property:Configuration=Release

単一の自己展開型インストール ファイルを作成する

タスク ***単一の自己展開型インストール ファイルを作成するには、次の手順を実行します。***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、**SingleImage** をクリックします。
3. **[ビルド]** タブをクリックします。
4. **"圧縮"** 設定で、**[圧縮]** を選択します。
5. **"圧縮メディア"**設定で **[はい]** を選択します。
6. **[ビルド]** エクスプローラーで、**SingleImage** を右クリックし **[ビルド]** をクリックします。

SingleImage ビルド タイプでビルドしたリリースはすべて単一ファイルに圧縮されます。

メモ: *"セットアップランチャー" 設定で [はい] を選択したときに InstallShield が生成するファイルは、実行可能ファイル (.exe) です。実行可能ファイルは Windows Installer データベース (.msi) と、(適切な場合) Windows Installer および製品をインストールするのに必要なすべてのファイルも含みます。セットアップランチャーを含まない場合、生成される単一ファイルは、製品をインストールするのに必要なロジックおよびデータすべてを含む .msi フィルです。*

Windows Vista 以降のプラットフォームでのセットアップランチャーの必要実行レベルを指定する

InstallShield では、Windows Vista 以降のプラットフォーム上でインストール (セットアップランチャー、InstallShield 前提条件および .msi ファイル) を実行するための Setup.exe ファイルが必要とする最低実行レベルを指定することができます。プロジェクトの個別のリリースに対して設定することが可能です。

タスク ***リリースの必要実行レベルを指定するには、次の手順に従います。***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[リリース]** エクスプローラーで、設定を行うリリースを選択します。

3. **"必要実行レベル"** に適切な設定を行います。
4. **Setup.exe** タブをクリックします。

選択可能なオプションは、以下のとおりです。

- **管理者** — Setup.exe の実行には、管理者権限が必要です。管理者は、Setup.exe の実行を承認する必要があります。非管理者は、管理者としての認証が必要になります。
- **最高権限** — Setup.exe の実行には、管理者権限が推奨されます。管理者は、Setup.exe の実行を同意(コンセント)する必要があります。非管理者は、管理者権限を持たずに Setup.exe を実行します。
- **起動者** — Setup.exe の実行に、管理者権限は必要ありません。したがって、管理者権限を持たないユーザーも Setup.exe を実行することができます。Setup.exe は、資格情報または同意 (コンセント) を求める UAC メッセージを表示しません。デフォルトでは、これが設定されています。

"セットアップランチャー" 設定が [はい] に設定された場合、InstallShield は Windows アプリケーション マニフェストを Setup.exe 起動ツールに埋め込みます。このマニフェストは選択された実行レベルを指定します。Windows Vista よりも古いバージョンのオペレーティング システムでは、必要実行レベルは適用されません。

"セットアップランチャー" 設定が [いいえ] に設定されている場合、InstallShield は Windows アプリケーション マニフェストを Setup.exe 起動ツールに埋め込みません。

必要実行レベルを昇格することの利点は、Setup.exe を実行するための権限の昇格が、必要な場合 1 度で済むという点です。またこれらの権限はインストールに含まれるすべての前提条件および .msi ファイルにも適用できるため、承認を得るためのプロンプトを複数回にわたって行う必要がありません。たとえば、すべての前提条件のうち 2 つが管理者権限を必要とする場合、この設定を [管理者] に変更します。そうすることにより、インストール中、Windows Installer が Setup.exe ファイルを実行する前、プロンプトはエンドユーザーに対して一度のみ表示されます。ただし、権限を昇格して、インストールの終わりでもアプリケーションを起動する場合、この昇格された権限はアプリケーションに適用されますので注意してください。ほとんどの場合、Windows Vista 以降では、昇格された権限を使用したアプリケーションの実行は推奨されていません。

エンド ユーザーのインストール エクスペリエンスは、インストールが必要とする権限によってのみ実行された場合、安全性がより一層向上します。アプリケーションは、システム管理者のみによる実行が必須の場合を除き、最も低い権限で実行されることが理想的です。

インターネットで配布するインストールを作成する

ユーザーがソフトウェアを受け取る方法は急激に変化しています。インターネット テクノロジがそれほど発展していなく、高速のインターネット接続が導入される以前は、すべてのソフトウェアが、フロッピー ディスクや CD-ROM などのリムーバブル メディアで発送されていました。ソフトウェアは現在、インターネットから直接ダウンロードされることがほとんどです。この時間およびコストを節約できるソフトウェア配布プロセスを利用するため、より簡単にダウンロードおよびインストールできる方法で、インストールをパッケージする必要があります。

Web 用のインストールには、満たす必要のある条件がいくつかあります。

圧縮サイズ

現在、多くの人が、高速ケーブル モデムや DSL 回線を使用してインターネットに接続していますが、低速度モデムを使用している人も多くいます。アプリケーションのダウンロードに必要なオンライン時間が増加するため、パッケージ サイズは、低速の接続を使用している人にとって非常に重要です。

自己展開

多くのファイル圧縮ユーティリティは、アプリケーション ファイルを解凍するための特別なクライアント側アプリケーションが必要です。このため、ユーザー側のダウンロードとインストール プロセスが複雑なものになります。インストール プロセスをシンプルにするには、使用する圧縮ユーティリティが他のアプリケーションを必要としないように、自己展開できる必要があります。

デジタル署名

顧客がソフトウェアをダウンロードおよびインストールするときの安全を確保するため、アプリケーションパッケージをデジタルで署名できます。デジタル署名により、ソフトウェア作成者および/または企業がエンドユーザーに識別され、発行以来アプリケーション コードが変更または改ざんされていないことが保証されます。アプリケーションにデジタル署名を付加する詳しい方法については、「デジタル署名とセキュリティ」を参照してください。

使いやすさ

インターネット配布のインストールのパッケージにおいて最も重要なことは、使いやすさです。すべての顧客が、インストール ファイルを保存する場所を指定したり、コンピューターを検索してこれらのファイルを自分で探したりすることを望んでいるとは限りません。代わりに、セットアップが圧縮パッケージにシームレスに統合し、インストールの開始手順を 1 つだけにします。

プロキシ サーバーのサポート

特定のファイルがターゲット システム上で必要な場合のみ、インストールがそのファイルをダウンロードするように構成できます。たとえば、Windows Installer エンジン、.NET Framework、および一部の InstallShield 前提条件が、一部またはほとんどのターゲット システム上に既在する可能性があります。これらのファイルをインストールに埋め込む代わりに、必要なファイルだけを実行時にダウンロードするようにプロジェクトを構成することができます。こうすることで、インストール全体のサイズを抑えることができます。

エンド ユーザーがプロキシ サーバーを使ってインターネットにアクセスする場合で、インストールがファイルをダウンロードするように構成されているとき、インストールはダウンロード中に、Internet Explorer で手動で構成されたシステム プロキシ設定を使用します。これは、ターゲット システム上で別のブラウザーがデフォルトとして設定されている場合でも同じです。

InstallShield は、Internet Explorer の ″設定を自動的に検出する″ 設定をサポートしませんので、ご注意ください。(エンドユーザーが使用している Internet Explorer で、LAN 接続に対して [設定を自動的に検出する] チェックボックスが選択されているときに、インストールでファイルのダウンロードが必要な場合、ファイルのダウンロードができないため、インストールは失敗します。エンドユーザーが使用している Internet Explorer で、LAN 接続に対して [設定を自動的に検出する] チェック ボックスが選択されている可能性があるとき、ダウンロードされるように構成する代わりに、すべてのファイルをインストールに埋めこんだ方が良い場合はあります。ファイルが埋めこまれている場合、失敗は避けられます。)ただし、InstallShield は Internet Explorer の LAN 接続用にセットアップされた自動構成スクリプト機能をサポートします。

Web 配布リリースの作成

タスク ***Web 配布用インストールを作成するには、以下の手順の従います:***

1. 以下のいずれかを実行して、Web 配布ウィザードを起動します。
   - **[Web 配布ウィザード]** ボタンをクリックします。
   - **[ビルド]** メニューで、**[Web 配布ウィザード]** をクリックします。
   - CTRL+W を押す。
2. ウィザードのパネルで必要な情報を入力します。
3. ウィザードの **[概要]** パネルで、**[[完了] ボタンをクリック後にリリースをビルドする]** チェック ボックスを選択します。
4. **[完了]** をクリックしてリリースをビルドします。

[リリース] ビューから、指定の場所にリリースを配布することができます。詳細については、「フォルダーまたは FTP サイトにリリースを自動的に配布する」を参照してください。

デジタル署名とセキュリティ

インストールとアプリケーションにデジタル署名をすることで、インストールやアプリケーション内のコードが、発行時以来、改ざんまたは変更されていないことをエンドユーザーに保証することができます。

[署名] タブを使って、InstallShield がファイルに署名するときに使用するデジタル署名に関する情報 (証明機関より付与されたデジタル証明書ファイルを含む) を指定します。

[署名] タブでまた、ビルド時にデジタル署名をするインストール内のファイルを指定することもできます。InstallShield は、作業中のプロジェクトの種類に応じて、リリースに含まれる次の任意およびすべてのファイルに署名することができます。

- Windows Installer パッケージ (.msi ファイル)
- Setup.exe ファイル
- リリースの任意のファイル (アプリケーション ファイルを含む)

Windows ロゴ: *インストールのすべての実行可能ファイル (.exe、.dll、.ocx、.sys、.cpl、.drv、および .scr ファイル) は、Windows ロゴ プログラムに準拠するためにデジタル署名が必要です。*

[署名] タブにある設定に関する詳しい情報は、「[署名] タブ」を参照してください。

証明機関

証明機関とは、デジタル証明書 (デジタル ID とも呼ばれます) を発行、管理する VeriSign の ような組織です。証明機関は、指定された基準に基づいて要求側の ID を検証し、デジタル証明書を発行します。デジタル証明書を取得するには、証明機関に会社と製品に関する特定の情報を提供する必要があります。

証明機関の一覧については、MSDN Web サイトの「Microsoft Root Certificate Program Members」を参照してください。

SHA-1 と SHA-256 証明書の違い

InstallShield では、インストールおよびファイルをビルド時に署名する際、HA-256 または SHA-1 ハッシュ アルゴリズムを使ったデジタル証明書を使用できます。

SHA-1 はセキュリティの脆弱性があるため、SHA-256 の使用が推奨されます。Microsoft は、Windows では 2016 年 1 月以降に SHA-1 証明書を使って署名およびタイムスタンプが追加されているアイテムを信頼しないことを発表しました。さらに、証明書を発行する組織である証明機関では、SHA-1 証明書が段階的に廃止されます。したがって、InstallShield プロジェクトに含まれる任意の SHA-1 証明書は、SHA-256 証明書と差し替えることが推奨されます。最新情報および特定の詳細については、証明機関にお問い合わせください。

プロジェクトで、SHA-256 証明書を使った署名が構成されている場合、InstallShield はビルド時に署名を行うファイルの署名に SHA-256 ハッシュを使用します。プロジェクトで SHA-1 証明書を使った署名が構成されている場合、InstallShield は SHA-1 ハッシュを使用します。SHA-1 証明書を使用すると、SHA-1 の使用についてアラートするビルド警告 -7346 が発生します。

証明書ファイルまたは証明書ストアにある証明書を使ってデジタル署名を生成する

ファイルおよびインストールの署名に使用するデジタル署名情報を指定するとき、InstallShield では次のオプションから選択できます:

- 使用中のマシンにある .pfx 証明書ファイルを指定できます。
- 証明書を含む証明書ストアを参照できます。

オプション 1 ー.pfx ファイル

.pfx (Personal Information Exchange) ファイルを使って、インストールおよびアプリケーションにデジタル署名を行います。次のツールを利用して、.pvk ファイルと .spc ファイルから .pfx ファイルを作成することができます:

- PVK2PFX.exe—Windows Platform SDK の一部で、Microsoft Visual Studio 2005 にも含まれています。
- pvkimprt.exe—この PVK Digital Certificate Files Importer ツールは、Microsoft Web サイト (http://www.microsoft.com/downloads/details.aspx?FamilyID=F9992C94-B129-46BC-B240-414BDFF679A7&displaylang=EN) からダウンロードすることができます。

.pfx ファイルは通常、パスワードに関連付けられています。

オプション 2 ー 証明書ストアにある証明書

証明書ストアにデジタル証明書を格納する場合、使用する証明書を含む証明書ストアをプロジェクトで参照することができます。この方法で署名を行う場合、ストア名 (Personal、Trusted Root Certification Authorities、Enterprise Trust、Intermediate Certification Authorities)、ストアの場所 (ユーザー、マシン)、および特定の証明書を識別するためのサブジェクトといった情報を指定する必要があります。

ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。

デジタル署名にタイムスタンプを追加する

リリースのデジタル署名情報を指定すると、InstallShield はデフォルトで、ビルド時にデジタル署名にタイムスタンプを追加します。デジタル証明書のタイムスタンプは、ファイルが署名された日時を記録します。これによって、署名が行われた時点で証明書の期限が切れていないことを証明することができます。信頼されたタイムスタンプ サーバーからのタイムスタンプは、一般的に証明書の有効期限よりも長く、デジタル署名を有効な状態に保ちます。

InstallShield が使用するデフォルトのタイムスタンプ サーバーを変更する方法、またはタイムスタンプ機能を無効にする方法については、「デジタル署名のタイムスタンプ サーバーを変更する」を参照してください。

ビルド時にリリースとそのファイルにデジタル署名を行う

では、リリースに構成できるデジタル署名の設定があります。ビルド時に、InstallShield は構成した設定を使用して、インストール パッケージ、Setup.exe ファイル、および、リリースにある定義した基準を満たす他のすべてのファイルに署名します。

タスク *リリースとそのファイルのデジタル署名を構成するには、以下の手順に従います。*

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、署名するリリースをクリックします。
3. **[署名]** タブをクリックします。
4. 次の設定を適切に構成します。
   - **証明書 URL**
   - **デジタル証明書ファイル**—この設定の省略記号ボタン (...) をクリックします。[証明書の選択] ダイアログ ボックスが開いて、.pfx ファイルの場所を指定するか、証明書を含む証明書ストアについての情報を指定することができます。
   - **証明書パスワード**—ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。
   - **署名の説明**
5. **"出力ファイルに署名する"** 設定で、署名を行うファイルを指定します (Setup.exe、.msi パッケージ、その両方、またはどちらにも署名しない)。
6. **"パッケージ内のファイルに署名する"** 設定で、インストール内の追加ファイルに署名するかどうかを指定します。

   [はい] を選択する場合、**"パッケージ内のファイルに署名する"** 設定の下にある他の設定を使って、署名を行うファイルとファイル パターンおよび署名を行わない項目を指定します。

   署名しないファイルとファイル パターンは、すべての署名するファイルとファイル パターンをオーバーライドしますので注意してください。たとえば、**"含める"** 設定および **"除外する"** 設定に ***.exe** を指定すると、InstallShield は .exe ファイルに署名を行いません。

ヒント: [署名] タブにある設定に関する詳しい情報は、「[署名] タブ」を参照してください。

ビルド時に、InstallShield は [署名] タブで指定されたファイルに署名を行います。リリースが、マージ モジュールを含むインストール用の場合、ファイルは、マージ モジュールがマージされる前に署名されます。

リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する

InstallShield では、インストールに含まれている InstallShield 前提条件のランタイムの場所を指定できます。

タスク *リリースの InstallShield 前提条件の配置場所を指定するには、以下の手順に従います:*

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. 構成するリリースを選択します。
3. **Setup.exe** タブをクリックします。
4. **"InstallShield 前提条件の場所"** 設定で、適切なオプションを選択します。

提供されている各オプションについての詳細は、「Setup.exe タブ」を参照してください。

タスク *各 InstallShield 前提条件に対して異なる場所を指定するには、次の手順に従います:*

1. **[再配布可能ファイル]** ビューで、各 InstallShield 前提条件について適切な場所を指定します。詳細については、「特定の InstallShield 前提条件の実行時の場所を指定する」を参照してください。
2. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
3. 構成するリリースを選択します。
4. **Setup.exe** タブをクリックします。
5. **InstallShield"前提条件の場所"** 設定で、**[個々の選択に従う]** を選択します。

InstallShield 前提条件が別の前提条件の依存ファイルとしてプロジェクトに追加される場合、前提条件依存ファイルの場所は、それを必要とする前提条件の場所設定に従います。

InstallShield 前提条件を含むリリースのビルドで次の両方が当てはまるとき、1 つまたは複数のビルド エラーが発生する可能性があります。

- 前提条件が、Setup.exe から抽出されるとき、または (エンド ユーザーのコンピューターに Web からダウンロードされる代わりに) ソース メディアからコピーされるときの InstallShield 前提条件の場所を指定してください。
- 前提条件ファイルはコンピューターにはありません。

ビルドエラーを除去するには、プロジェクトから InstallShield 前提条件を削除するか、インターネットからコンピューターに InstallShield 前提条件をダウンロードするか、またはリリースの InstallShield 前提条件の場所をダウンロードオプションに変更してから、リリースを再ビルドします。

メディアの圧縮

[リリース] ビューで使用できる各リリースの種類 (例、CD_ROM、DVD、および SingleImage) には、"圧縮" と呼ばれる設定が含まれます。インストールのサイズを縮小できる圧縮アルゴリズムを使用する場合、この設定で [圧縮] を選択します。この設定内で、インストールのファイルが .cab ファイルに圧縮されます。

ファイルをリリース メディアに圧縮しない場合、"圧縮" 設定で [非圧縮] を選択します。

フロッピー ディスクの配布

直接にはサポートされていませんが、[カスタム] リリース タイプを使用して、フロッピーディスクでリリースを配布することができます。フロッピーディスクにリリースをビルドする場合の注意点は以下の通りです:

- **ディスク容量**—製品がインストールされる前に Windows Installer をインストールする場合、最初に Windows Installer をインストールしなくてはなりません。Windows Installer 再配布可能ファイルのサイズに関する要件のため、Windows Installer エンジンも配布する必要がある場合、フロッピーディスクでインストールを配布することは困難です。
- **ディスク分割**—InstallShield で作成され、Windows Installer サービスとインタラクトする .msi ファイルは複数のディスクに分割することはできません。また、.msi ファイルはインストールの最初のディスクに置く必要があります。したがって、1 つの圧縮された .msi ファイルにすべてのファイルを入れようとすると、1 枚のフロッピーディスクには収まらない場合があります。ただし、ファイルを圧縮しない場合は、インストールの一連のディスクに含めることができます。

フロッピー ディスクで配布するリリースをビルドする

タスク *フロッピーディスク配布用のリリースをビルドするには、以下の手順に従います:*

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、**[カスタム]** をクリックします。
3. **[ビルド]** タブをクリックします。
4. **"メディア サイズ"** 設定を **1.4** に設定します。
5. **"メディア サイズ単位"** 設定で、**MB** と入力します。
6. **"クラスタ サイズ"** 設定で、クラスタ サイズをディスク容量に応じてバイトで入力します (**512** または **1024**)。
7. **"圧縮"** 設定で、**[いいえ]** を選択します。
8. **"Autorun.inf ファイルの生成"** 設定で、**[いいえ]** を選択します。
9. **Setup.exe** タブをクリックします。
10. **"セットアップランチャー"** 設定で、**[いいえ]** を選択します。
11. **[ビルド]** エクスプローラーで、**[カスタム]** を右クリックし **[ビルド]** をクリックします。

注意: *複数ディスクのインストールをビルドする場合は、2 枚目以降のディスクのボリューム ラベルを設定する必要があります。ボリューム ラベルは DISK2 (2 枚目のディスク)、DISK3 (3 枚目のディスク) という具合に指定する必要があります。詳細については、「ボリューム ラベルの設定」を参照してください。*

ボリューム ラベルの設定

タスク ***CD-ROM または DVD-ROM のボリューム ラベルを設定するには、以下の手順に従います:***

CD-ROM または DVD-ROM 作成ソフトウェアの要件を調べてください。

タスク ***フロッピーディスクのボリューム ラベルを設定するには、以下の手順に従います:***

1. Windows エクスプローラーを開きます。
2. フロッピーディスクが入っているドライブを右クリックし、**[プロパティ]** をクリックします。
3. **[全般]** タブをクリックします。
4. **[ラベル]** ボックスで、適切なボリューム ラベルをします。

ヒント: *複数のディスクのインストールをビルドする場合は、二枚目以降のディスクのボリューム ラベルを指定します。ボリューム ラベルの指定方法は、使用する配布メディアによって異なります。ボリューム ラベルは DISK2 (2 枚目のディスク)、DISK3 (3 枚目のディスク) という具合に指定する必要があります。*

Windows Installer パッケージ (.msi) を出力として作成する

.msi ファイルを生成して、インストールをパッケージすることができます。

タスク ***.msi ファイルを InstallShield の出力として作成するには、以下の手順に従います:***

1. 必要なファイル、機能、レジストリ エントリ、ショートカットを含むインストールを作成します。
2. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
3. **[ビルド]** エクスプローラーで、ビルドするメディアの種類をクリックします。

   メモ: *WebDeployment リリース、または SingleImage リリース タイプを使用した場合、エンド ユーザーがアクセスできる .msi ファイルを生成することができません。*

4. **[ビルド]** タブをクリックします。
5. **"圧縮"** 設定で、**[非圧縮]** を選択します。このように設定すると、InstallShield はSetup.exe ファイルに .msi ファイルを埋め込みません。

6. **[ビルド]** ボタンをクリックします。

リリースがビルドされて、<プロジェクトの場所>¥<プロジェクト名>¥Express¥<リリースの種類>¥DiskImages¥Disk1 フォルダーに格納されます。.msi ファイルはこのフォルダーの中にあります。

Autorun

コンピューターのドライブにソフトウェア CD または DVD を挿入すると、マルチメディア ブラウザーが自動的に起動され、ソフトウェアを簡単にインストールできる場合があります。この Autorun 機能は、CD または DVD のルートレベルにあるテキスト ファイル (Autorun.inf) を使って実現されます。このファイルは、主に CD または DVD ブラウザーまたはインストールのどちらかを起動するのに使用されます。

CD または DVD の Autorun を有効にする

タスク ***ターゲット システムの CD または DVD ドライブに CD または DVD を挿入したときに Autorun を有効にするには、以下の手順を実行します:***

1. **Autorun.inf** という名前のテキスト ファイルを作成します。
2. Autorun.inf ファイルで、エンド ユーザーが CD または DVD をドライブに挿入した時に自動的に起動するファイルの名前を指定します。以下の構文を使用してください。

   [autorun]
   open = filename

   filename は、起動するファイルの名前です。たとえば、インストールに Setup.exe を含める場合、**filename** の場所に **Setup.exe** を指定します。
3. Autorun.inf ファイルを CD または DVD のルート レベルに配置します。

インストールのテストと実行

インストールを配布する前に、テストを行って、顧客に頼らずに問題を見つけることはとても重要です。InstallShield では、(ファイルをターゲット システムにコピーする手間をかけずに) エンドユーザー ダイアログをテストしたり、転送するファイルを含むインストール全体を実行したりすることができます。

インストールのテストと実行

リリースのテストは、インストールの開発プロセスで必ず必要です。リリースをテストするには、以下のいずれかの方法を使用します。

- **インストールの実行** —このオプションを選択すると、インストールがエンド ユーザーのマシン上とまったく同じ状態で実行されます。ファイルはすべて転送され、ショートカットおよびレジストリ エントリが作成され、エンドユーザー インターフェイスが表示されます。このタイプのテストを InstallShield インターフェイスから実行することはできますが、公にリリースの前に複数のクリーン マシーンでセットアップをテストすることをお勧めします。
- **インストールのテスト**—インストールをテストしたとき、エンド ユーザー インターフェイス要素のみが実行されます。ファイルは転送されず、コンピューターへの変更もありません。この規則の唯一の例外は、イン

ストールにカスタム アクションがある場合です。すべてのカスタム アクションはこのテスト中に実行されます。

タスク ***インストールをテストまたは実行するには、以下の手順に従います:***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[ビルド]** エクスプローラーで、テストまたは実行するリリースをクリックします。
3. ツールバー上の **[実行]** ボタンまたは **[ユーザー インターフェイスのテスト]** ボタンをクリックします。

サイレント モードでインストールを実行する

Setup.exe の起動時に進行状況バーを表示しない場合、/s コマンドライン パラメーターを使用することができます。たとえば、**Setup.exe /s** を入力すると、Setup.exe は起動しますが、ユーザー インターフェイスは表示されません。.msi パッケージもサイレントで実行する場合は、次のように /v パラメーターを使用して Setup.exe を通して /qn コマンドライン パラメーターを渡す必要があります:

Setup.exe /s /v/qn

条件インストール

InstallShield では、いくつかの基準に基づいて条件付インストールを行うことができる機能が提供されています。条件は [要件] ビューを使って指定できます。インストールを指定された条件に満たないマシンで実行すると、インストールは終了してアプリケーションはインストールされません。

インストールの配布

インストールを作成したあと、場合により、指定した場所へ配布する必要があります。これにはネットワーク ドライブ、CD、ローカル ドライブの別の場所、または FTP サイトが可能です。インストールを配布すると、インストールをビルドしたときに作成されたディスク イメージが、指定の場所にコピーされます。

フォルダーまたは FTP サイトにリリースを自動的に配布する

リリースのビルドとテストが完了すると、残る作業は、それを適切な場所に配布するのみです。リリースを適切な場所に手動でコピーすることもできますし、[リリース] ビューの [イベント] タブを使って、InstallShield がリリースを適切な場所 (ローカル/ネットワークの場所、または FTP サイト) へ自動的にコピーするように構成することもできます。

タスク ***InstallShield を構成して、リリースを特定の場所へ自動的に配布するようにするには、以下の手順に従います。***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[リリース]** エクスプローラーで、構成するリリースを選択します。
3. **[イベント]** タブをクリックします。

4. 設定を適切に構成します。[イベント] タブにある設定に関する詳しい情報は、「[イベント] タブ」を参照してください。

メモ: *インストールが 1 つのディスクのみで構成される場合は、リリースの保存場所に Disk1 フォルダーの内容がコピーされますが、フォルダーそのものはコピーされません。インストールが複数のディスクで構成される場合は、フォルダーおよびその内容がリリースの保存場所にコピーされます。*

ビルドを行う度にビルド エンジンが指定された場所にリリースをコピーするように設定するには、"ビルド後、配布する" 設定を [はい] に選択します。

5

アプリケーションのアップデート

アプリケーションのアップグレードをインストールすることは、アプリケーションのオリジナルリリースのインストールよりもはるかに一般的な操作です。この意味で、効率的で、信頼のおけるアップグレードを作成することは、とても重要なタスクです。

[アップデートの更新] は、異なるタイプのアップグレードについての確かなバックグラウンド情報が提供されており、一般的なパッチについての誤解が分かりやすく説明されています。"アプリケーションのアップデート" はまた、製品に最適なアップグレード ソリューションを判断するのに役に立ち、アップグレード、パッチを作成する手順を案内します。また、このセクションは、FlexNet Connect を利用してどのようにエンドユーザーに製品の新しいバージョンのリリースついて通知するかについても説明します。

アップグレードの概要

ソフトウェア アプリケーションの保守は、開発にかかったコストよりも高くかかる場合があります。そのため、効率的で信頼のおけるアップグレードを作成することは重要なタスクです。アプリケーションに対する堅牢なアップグレードの配布が可能かどうかは、オリジナル インストールパッケージがどのように構成され、配布されたかによって左右されます。

Windows Installer は 3 種類 (スモール アップデート、マイナー アップグレード、およびメジャー アップグレード) の製品アップグレードをサポートしています。アップグレードは、完全インストールまたはパッチとしてパッケージすることができます。パッチはアップグレードを実装するための 1 つの仕組みに過ぎません。ただし完全リリースとは違って、パッチはインストール済みのファイルを最新版に変更するために必要な部分のみをユーザーに配布します。

メジャーアップグレード

プロジェクト: *この情報は Express プロジェクトに適用します。*

メジャーアップグレードでは製品の変更規模が大きいため、パッケージコードだけでなく製品バージョン番号および製品コードの両方を変更する価値があります。たとえばバージョン 1.2 製品の 2.0 へのアップデートです。メジャーアップグレードは、以前のバージョンが存在しない場合は初回インストールと同様に作動します。以前のバージョンが存在する場合、メジャーアップグレードは通常それをアンインストールしてから新しいバージョンをインストールします。

メジャーアップグレードの動作の仕方

Windows Installer のヘルプで示唆されているように、メジャー アップグレードを行うにはインストールの最新バージョンの製品コードを変更する必要があります。

メモ: *インストールの製品コードは、[一般情報] ビューで設定できます。新しい製品コードは、一意である限り、特に制約はありません。*

最新インストールの製品コードを更新した場合、[アップグレード パス] ビューを使って、アップグレードを行う以前のすべてのバージョンについての情報を指定します。これは、製品の最も新しいインストール プロジェクトを使って行います。

メージャー アップグレードの作成についての詳細は、「完全インストール アップグレードの作成」を参照してください。

実行時のメージャーアップグレード

製品の以前のバージョンがターゲット マシン上に存在しない場合にエンド ユーザーがメジャー アップグレードを実行すると、初回インストールとしてインストールが行われます。

製品の以前のバージョンがターゲット マシン上に存在している状態でエンド ユーザーがメージャー アップグレードを実行すると、エンドユーザーはあたかも以前のインストールが存在しないマシン上に最新のアプリケーションをインストールするかのような印象を受けます。唯一の異なる点は、新しいリソースがインストールされる前に、インストールはまずターゲット マシンから古いバージョンのアプリケーションとそのリソースを削除するということです。この削除過程は、[セットアップ進行状況] ダイアログの進行状況バーで表示され、エンドユーザーはアンインストールの処理状況を直接見ることができます。以前のインストールの削除が完了すると、最新のインストールからのリソースがターゲットマシンにインストールされます。

このタイプのアップグレードは、つまり、完全アンインストールとそれに続くアプリケーションに関連付けられたリソースすべての再インストールからなります。したがって、エンドユーザーによって構成されたアプリケーションのためのすべてのデータはエンドユーザーのマシーンから完全に削除される可能性があります。エンドユーザー データの一部を残しておく必要がある場合、このデータのバックアップをとってから、新しいデータのインストールが完了した後それを置き換えるカスタム アクションを作成する必要があります。

マイナーアップグレード

エディション: *InstallShield Premier Edition または InstallShield を使って、完全インストールとしてパッケージされたマイナー アップグレード を作成することができます。InstallShield Premier Edition、InstallShield 、または InstallShield Express Edition を使って、QuickPatch プロジェクトとしてパッケージされたマイナー アップグレードを作成することができます。*

製品データベースおよびファイルへの変更規模が ProductCode プロパティを変更するほどではないけれども、ProductVersion プロパティを変更をするだけの価値があるとき、これをマイナー アップグレードといいます。つまり、マイナーアップグレードの場合、パッケージコードおよび製品バージョン番号両方とも以前のインストールパッケージと異なりますが、製品コードは変わりません。バージョン 1.1 の製品からバージョン 1.2 へアップデートはひとつの例です。マイナーアップグレードでは、通常、異なるバージョンの間でインストールの構成において重要な変更は行われません。フル インストールとしてパッケージされたマイナー アップグレード は、以前のバージョンが存在しない場合は初回インストールと同様に動作しますが、製品が既にインストールされている場合、その上からインストールを行います。既存のインストールをアップグレードする場合、マイナーアップグレードは基本的にバージョン 1.2 と 1.1 のアプリケーションの差分のみをインストールします。

スモール アップデート

エディション: *InstallShield Premier Edition または InstallShield 使って、スモール アップデートを作成できます。*

本質的にスモール アップデートは、インストール済みのアプリケーションのいくつかのファイルを変更するのに使用されるアップデート タイプで、一般的に小さいバグ修正の配布に使用されます。スモール アップデートには、ホットフィックスなどの製品バージョンの変更が必要なほど大きくはない変更が含まれます。スモールアップデートでは、パッケージコードの変更が必要です。

3.0 以前のバージョンの Windows Installer と共にインストールされるスモールアップデートの短所は、製品のより新しいバージョンのためのインストーラーをはじめとする、外部プログラムは、オリジナルバージョンとアップデートされたバージョンを区別することができないということです。また、3.0 以前のバージョンの Windows Installer は、スモールアップデートを正しい順序で適用することができないこともあります。

パッチの適用

エディション: *InstallShield Premier Edition または InstallShield 使って、標準パッチを作成できます。*

パッチは、以前のバージョンの Windows Installer インストールパッケージをアップデートしてアプリケーションをアップデートするための効率的な仕組みです。パッチはインストール済みのファイルを新しいファイルへ変更するのに必要な部分のみをカスタマーに配布します。パッチが優れている理由のひとつは、アップグレードを配布するために必要なアップグレードパッケージのサイズを、完全インストール パッケージのそれと比べて格段に小さく収めることが可能な点です。アップグレードパッケージをできるだけ小さくすることで、インターネットを通してアップグレードを簡単に配布することが可能になります。

但し、パッチが必ずしも最善のソリューションとは限らないので注意してください。たとえば、インストールを圧縮形式から非圧縮形式に変更する場合、またはその逆を行なう場合、アップグレードはパッチではなく完全インストールとしてパッケージしなくてはなりません。アップグレードの最適なパッケージ オプションの選び方については、「アップグレードのパッケージ オプション」を参照してください。

パッチはパッチパッケージ (.msp) ファイルの形式で配布され、これをエンドユーザーがインストール済みの製品に適用します。パッチパッケージは、インストールの旧バージョンをその数に関わらずアップデートすることができます。パッチパッケージには、指定した以前のバージョンをアップデートするための個別のトランスフォームおよび指示が含まれています。

パッチ作成の重要な側面は、パッチパッケージが作成されるパラメーターを定義するパッチ作成プロパティ (.pcp) ファイルの生成です。.pcp ファイルは特定のスキーマを持つデータベースです。

QuickPatch プロジェクト

プロジェクト: *以前のバージョンのインストールが次のいずれかのプロジェクト タイプで作成されている場合、初期のバージョンの製品を更新する QuickPatch を作成することができます。*

- *Express*
- *QuickPatch*

QuickPatch プロジェクトは、規模の小さいシングル アップグレードをユーザーへ配布したいインストール作成者へお勧めするプロジェクトの種類です。カスタム アクションの追加、.ini データの変更などのより広範囲におよび変更には通常、標準パッチが必要です。

QuickPatch は、そのカスタマイズの範囲が限られてはいますが、標準パッチの作成 (InstallShield Premier Edition および InstallShield で利用可能) に代わるシンプルなオーサリング方法です。基本的にどちらのパッチ作成方法も同じ配布タイプ (.msp と .exe ファイル) を作成します。

QuickPatch では、次のすべてを実行することができます。

- 元のインストールまたは以前の QuickPatch へ新しいファイルを追加する。
- 元のインストールのファイルを削除する。
- 以前の QuickPatch と共に追加されたファイルを削除する。
- 上記と同じ操作をレジストリ エントリで実行する。
- 元のインストールに含まれていたが、現在の QuickPatch プロジェクトには適用しないカスタム アクションを削除する。

QuickPatch プロジェクトの作成は常に、[新規 QuickPatch 作成] ウィザードから始めます。ウィザードを完了すると、QuickPatch プロジェクトに必要な基本条件をすべて満たすことができます。そのあと InstallShield で QuickPatch プロジェクトが開いたら、プロジェクト設定を構成することができます。

最適なアップグレード ソリューションの決め方

アップグレードのインストール作成は、そのタイプに関わらず、製品の以前のバージョンを持っていないターゲット システムを対象にするかどうかを決めるところから始まります。それが決まると、アップグレードのパッケージにどの種類の方法を使用するかを決めることができます。以下のテーブルは最適な方法を選ぶときに目安となる一般的な概要です。アップグレードのパッケージに関するテクニックについてより掘り下げた内容については、「アップグレードのパッケージ オプション」を参照してください。

テーブル 5-1 ・ Express プロジェクトの可能なアップグレード ソリューション

ターゲット システムの状況	必要なインストール の種類	アップデート パッケージの技法
ターゲット システムの中には以前のバージョンの製品がインストールされているものもあり、製品がまったくインストールされていないものもある。	ファイルのサイズが問題ではない場合、以下の両方を行うインストールを作成することも可能です。 ・ ターゲット システムに以前のバージョンが存在していない場合、初回インストールとして作動する。 ・ 既にターゲット システムにインストールされている場合、既存の製品をアップデートする。	完全インストール アップグレードを作成します。詳細については、「完全インストール アップグレードの作成」を参照してください。
ターゲット システムの中には以前のバージョンの製品がインストールされているものもあり、製品がまったくインストールされていないものもある。	以前のバージョンの製品をアップグレードを必要とするエンドユーザーのためのスモール インストールが必要な場合、2 つのインストールを分けて作成することができます。 ・ 初回インストールとして作動する完全インストール。 ・ 既にインストールされている製品の 1 つまたは複数の以前のバージョンをアップデートする小規模のインストール。このインストールにはバージョンとバージョンの間で変更があったデータのみが含まれているので、完全インストール パッケージの配布よりも狭い帯域幅を利用したアップグレードの配布が可能となります。	初回インストール用には、完全インストール アップグレードを作成します。詳細については、「完全インストール アップグレードの作成」を参照してください。 製品の以前のバージョンを持っているエンドユーザー向けには、完全インストール アップグレードの代わりに QuickPatch プロジェクトを作成することが可能な場合もあります。QuickPatch プロジェクトが適切かどうかを判断するには、「アップグレードのパッケージ オプション」 を参照してください。

テーブル 5-1 ・ Express プロジェクトの可能なアップグレード ソリューション (続き)

ターゲット システムの状況	必要なインストール の種類	アップデート パッケージの技法
すべてのターゲット システムが製品の以前のバージョンを持っている。	既にインストールされている製品の 1 つまたは複数の以前のバージョンをアップデートする小規模のインストールを作成することができます。このインストールにはバージョンとバージョンの間で変更があったデータのみが含まれているので、完全インストール パッケージの配布よりも狭い帯域幅を利用したアップグレードの配布が可能となります。	フル インストール アップグレードの代わりに QuickPatch プロジェクトの作成が可能かもしれません。QuickPatch プロジェクトが適切かどうかを判断するには、「アップグレードのパッケージ オプション」を参照してください。

アップグレードのパッケージ オプション

***プロジェクト・** この情報は、次のプロジェクトの種類に適用します :*

- *Express*
- *QuickPatch*

エンド ユーザーのマシン上にインストール済みの製品のバージョンを更新するインストールを作成する場合、アップグレードをパッケージする 2 つの選択肢があります。

- 以前のバージョンがインストールされている場合は既存製品をアップデートし、以前のバージョンが存在しない場合は初回インストールとして作動する完全インストールとして、アップグレードをパッケージすることが可能です。
- アップデートするバージョン間で変更されているデータ (.msi データおよびバイトレベルのファイル差分) のみを含む QuickPatch としてアップグレードをパッケージすることが可能です。

フルインストール パッケージ

メジャー アップグレードは、製品の以前のバージョンが存在しない場合に初回インストールと同様に動作します。以前のバージョンが存在する場合、メジャーアップグレードは通常それをアンインストールしてから新しいバージョンをインストールします。

QuickPatch パッケージ

QuickPatch パッケージを利用すると、アプリケーションのファイルを特定バージョンにアップデートするのに必要な、データベースの一部のみを配布することができます。このため、フル インストールとしてパッケージされたアップグレードに比べて規模の小さいパッケージを作成することが可能です。つまり、完全インストールパッケージの配布よりも狭い帯域幅を利用したアップグレードの配布が可能となります。

***メモ・** パッチはアップグレードの一種ではありません。パッチは簡単にいうと、省スペースでアップグレードを配布するためのメカニズムです。*

アップグレードの最適なパッケージオプションの決定方法

「最適なアップグレード ソリューションの決め方」トピックには、以前のバージョンの更新のために最も適切なパッケージの種類を判断するときに参考となるテーブルが掲載されています。QuickPatch がアップグレードのパッケージ方法として最も適切なメカニズムである場合もあります。しかし状況によって、QuickPatch の代わりにフル インストールとしてアップグレードをパッケージする方法が適切な場合もあります。次のテーブルは、ユーザーのニーズに最も適したアップグレードの種類を決定するお手伝いをします。アップグレードの要件のうち 1 つでも QuickPatch に適していない場合は、フル インストール アップグレードを作成する必要があります。

テーブル 5-2 ・ フル インストール アップグレードと QuickPatch の違い

アップグレードまたは QuickPatch の要件	フル インストール アップグレードを作成する	QuickPatch を作成する	メモ
ベース パッケージをアップグレードする多くの累積アップデートを適用できる機能	はい	可 (QuickPatch の簡素化を使用する場合)	QuickPatch の簡素化を使用しない場合、15 回以上パッチを適用することはできません。詳細については、「QuickPatch パッケージを簡素化するかどうかを指定する」を参照してください。
.msi パッケージ名の変更	はい	いいえ	.msi ファイルが Setup.exe インストールのランチャ に圧縮されていなければ、デフォルトのファイル名は ″製品名″ プロパティから取得されます。
エンドユーザーが以前のバージョンと最新のバージョンを同一マシンにインストールできるようにする	はい	いいえ	
新しいサブ機能を追加する	はい	いいえ	
ファイルの追加、変更、または削除	はい	はい	
レジストリデータの追加、変更、または削除	はい	はい	QuickPatch で追加する新規レジストリデータはすべて、既にオリジナルの製品に存在する機能に関連付ける必要があります。
ショートカットの追加、変更、または削除	はい	いいえ	

テーブル 5-2・フル インストール アップグレードと QuickPatch の違い（続き）

アップグレードまたは QuickPatch の要件	フル インストール アップグレードを作成する	QuickPatch を作成する	メモ
カスタム アクションの追加、変更、または削除	はい	オリジナルのベースインストールに含められたカスタム アクションのみ削除することができます。	
再配布可能ファイルの追加または削除	はい	いいえ	
ODBC リソースの追加、変更、または削除	はい	いいえ	
.ini ファイルの編集	はい	いいえ	
IIS Web サイトやコンポーネント サービスのようなサーバー設定を構成する	はい	いいえ	

製品の以前のバージョンをアップデートするのに利用するパッケージ タイプを判断するとき、以下の詳細についても考慮してください。

- ターゲット イメージが Windows Installer 1.2 以前で作成され、アップグレード イメージが Windows Installer 2.0 以降で作成された場合、アップグレードは QuickPatch ではなく完全インストールとしてパッケージする必要があります。このスキーマの違いを超えたパッケージに QuickPatch を作成すると、問題が発生します。
- アップグレードがターゲット システム上で 1 つまたは複数のファイルを別の場所に移動する場合、QuickPatch ではなくフル インストールとしてアップグレードをパッケージしなくてはなりません。エンドユーザーがターゲット システム上にあるファイルを移動するアップグレード用のパッチをインストールすると、問題が発生する可能性があります。たとえば、パッチが作動しない、システム修復が作動しない、後続のパッチが作動しない、またエンドユーザーが製品をアンインストールできない等。ワークアラウンドとして、古いロケーションにあるファイルを削除して新しいロケーションにファイルを追加する QuickPatch を作成することができます。
- インストールを圧縮形式から非圧縮形式に変更する場合、またはその逆を行なう場合、アップグレードは QuickPatch ではなく完全インストールとしてパッケージしなくてはなりません。このシナリオで QuickPatch を利用すると、ターゲット システムの修復が作動しない、後続の QuickPatch が動作しない、またはエンドユーザーが製品をアンインストールできない場合があります。
- ファイルを .cab ファイル間で移動させる必要がある場合、または .cab ファイル内でファイル順を変更する必要がある場合、アップグレードを QuickPatch ではなくフル インストールとしてパッケージしなくてはなりません。
- オリジナルインストールのファイル数が 32,767 を超えるとき、最新版のインストールに含まれるファイル数が 32,767 未満の場合、QuickPatch は失敗します。同様に、オリジナル インストールのファイル数が 32,767 未満のときに、最新版のインストールに含まれるファイル数が 32,767 を超える場合、QuickPatch は失敗します。どちらの場合も、アップグレードをフルインストールとしてパッケージする必要があります。

オリジナルインストールと、最新版インストールの両方のファイル数が 32,767 を超える (または両方のファイル数が 32,767 未満) 場合、アップデートを QuickPatch としてパッケージすることが可能です。

アップグレードおよび QuickPatch プロジェクトを使用する

アップグレードをフル インストールとしてパッケージする場合、まずインストール プロジェクトの最新版を開いて、必要に応じてファイルおよびレジストリ エントリを追加するなどの変更を加えます。

QuickPatch としてアップグレードをパッケージする場合、まず新しい QuickPatch プロジェクトを作成します。QuickPatch プロジェクトでは、QuickPatch を使ってどの以前のリリースをパッチするかを指定します。

エディション: *InstallShield Express Edition では、ターゲット マシン上に製品の以前のバージョンが存在する場合に、新しいバージョンをインストールする前に古いバージョンを削除するフル インストール パッケージとしてメジャー アップグレードを作成することができます。InstallShield Express Edition ではまた、QuickPatch パッケージとしてパッケージされたマイナー アップグレードを作成することもできます。スモール アップデートまたは標準パッチを作成する場合は、InstallShield Premier Edition または InstallShield のアップグレードを考慮してください。*

アップグレードおよび QuickPatch プロジェクトの作成方法については、このセクションにあるトピックを参照してください。

ファイルの上書き規則について

Windows Installer サービスは、デフォルトで、アップグレードが含むファイルがターゲット システムに既に存在するファイルを上書きするかどうかを判断するときいくつかのファイルの上書き規則を使用します。これらの規則は、REINSTALLMODE プロパティが o 設定を使用してターゲット システムの古いファイルを上書きインストールするときに適用されます。この動作を変更するには、o オプションを次の値の中から 1 つを使って置き換えます。

- p – ターゲット システム上に対応するファイルがない場合のみ再インストールします。
- e – ファイルが見つからないか、または、バージョンが古いか同じなとき再インストールします。
- d – ファイルが見つからないか異なるとき、再インストールします。
- a – バージョンに関わらず、すべてのファイルを再インストールします。

REINSTALLMODE の設定は、インストール時にすべての機能に適用されるので、個別に設定することはできません。また、REINSTALLMODE に a を含めるように設定すると、パッチの適用中にオリジナルのインストールソースを求めるプロンプトを引き起こす可能性があります。

パッケージ コード、製品バージョン、および製品コードのアップデート

いくつかの Windows Installer コードは製品を識別するのに役立ちます。

- **Package Code** – 概要情報ストリームの一部。特定のデータベースを認識します。パッケージコードは、Windows Installer プロパティではありません。同じパッケージコードを持つ 2 つの .msi データベースの内容は同じでなければなりません。したがって、ビルドごとにパッケージコードを変更する必要があります。

- **ProductVersion** — これは、製品バージョンを含む Windows Installer プロパティです。Windows Installer は、バージョンを比較する際、ProductVersion プロパティの最初の 3 つのフィールドのみを利用することに注意してください。たとえば、1.2.3.4 の製品バージョンの場合、4 は無視されます。(これは、ProductVersion 値の比較にのみ当てはまり、ファイル バージョンの比較には当てはまらないので注意してください。)
- **ProductCode** — これは、製品の GUID を含む Windows Installer プロパティです。たとえ ProductName プロパティの値が同じでも、Windows Installer は ProductCode GUID が異なる 2 つの製品は相互に関連が無いものとして扱います。
- **UpgradeCode** — これは、製品ファミリーを意味する GUID を含む Windows Installer プロパティです。UpgradeCode は、パッチ目的のために、関連製品ファミリーのさまざまなバージョンや言語すべてで統一されている必要があります。[アップグレード パス] ビューでアップグレードの UpgradeCode を設定することができます。

いづれのアップグレード タイプも、インストールされている製品を識別するため、[一般情報] ビューにあるパッケージ コード、製品バージョン、および製品コードの様々な組み合わせを変更する必要があります。アップグレード コードは、すべてのバージョンで 同じものを使用する必要があります。次のテーブルで、異なるタイプのアップグレードに各コードをいつ変更すべきかが分かります。

テーブル 5-3 · 異なる種類のアップグレードごとに変更する必要があるコード

	パッケージ コード	製品バージョン	製品コード	アップグレード コード
スモール アップデート	X			
マイナーアップグレード	X	X		
メジャー アップグレード	X	X	X	

完全インストール アップグレードの作成

***プロジェクト·** この情報は Express プロジェクトに適用します。*

完全インストール アップグレードが状況に適した最も有効なアップグレード ソリューションであると判断したならば、[アップグレード パス] ビューでアップグレードの作成を開始します。新しい Express プロジェクトを開始するか、製品の最新版のインストールを開いて、必要に応じてこれを変更することができます。

メジャー アップグレードは重要な機能の変更を意味し、製品コードおよび製品バージョンの変更が必要となります。これらの値は [一般情報] ビューでアップデートすることができます。

***メモ·** 製品コードを変更すると、ProductName 値が同じでも、Windows Installer は製品の最新版と以前のバージョンが関連しないもとのして処理します。製品の両方のバージョンを同じシステム上にインストール可能にするには、製品コードおよびメインとなるインストールディレクトリ (多くの場合、INSTALLDIR) の両方を変更するだけです。*

アップグレード パスは、メジャー アップグレードにのみ使用します。[アップグレードパス] ビューは、インストール全体をアップグレードするためのもので、一部のファイルのアップグレード(パッチ) に使用するものではありません。パッチ機能が必要な場合は、QuickPatch プロジェクト を作成することができます。

製品の以前のバージョンに対するアップグレードを作成するときに [アップグレード パス] ビューを使用しなかった場合、エンド ユーザーは新しいバージョンをインストールする前に古いバージョンを手動でアンインストールしなくてはならないことがあります。[アップグレードパス] ビューは、新らしいバージョンのインストールだけでなく、1 つまたは複数製品のアンインストールも処理することができます。たとえば、ユーザーのシステムに製品のバージョン 1.0 と 2.0 がインストールされているとして、バージョン 3.0 をリリースする場合、[アップグレードパス] ビューの [アップグレード パス] エクスプローラーに以前のバージョンを両方追加することによって、どちらか一方、または両方を削除することができます。各アップグレードパスは、[アップグレード パス] エクスプローラーに個別に入力する必要があります。

次の表は、最初のインストール バージョンと、2 つの製品アップデートの製品コードとアップグレード コードの関係を表しています。この情報に基づいてアップグレードパスを設定すると、バージョン 1,0 はバージョン 1,1 か 2,0 に、バージョン 1,1 は 2,0 にアップグレードします。

テーブル 5-4 ・ アップグレードのサンプル コード

製品バージョン	製品コード	アップグレード コード
1.0	{D73FFD24-6087-4D5A-82AB-1F6B1A680433}	{A00700B1-2345-6789-ABCD-EF0123456789}
1.1	{D73FFD24-6087-4D5A-82AB-1F6B1A680433}	{A00700B1-2345-6789-ABCD-EF0123456789}
2.0	{A1071CCE-11A1-11A2-B1A2-B3000D067A34}	{A00700B1-2345-6789-ABCD-EF0123456789}

InstallShield を使用したアップグレード作成に関する詳細は、InstallShield ヘルプラ イブラリ内の該当セクションを参照してください。

アップグレード中にグローバル アセンブリ キャッシュからアセンブリが削除されるのを防ぐ方法

***プロジェクト・** この情報は Express プロジェクトに適用します。*

デフォルトで、[アップグレード パス] ビューで作成されたアップグレードは、製品の新しいバージョンをインストールする前に古いバージョンを削除するように構成されます。言い換えると、RemoveExistingProducts アクションが InstallFinalize アクションの前にスケジュールされています。製品がグローバル アセンブリ キャッシュ (GAC) にインストールされているアセンブリを含んでいる場合、このデフォルト シーケンス動作により、アップグレードが適用された後にアセンブリが GAC から消えてしまうという問題が発生する場合があります。この問題は、Windows Installer の参照カウントがアセンブリに対して適切に実行されないという問題に起因しており、結果的に、アップグレードでアセンブリが削除された後再インストールされないという問題につながっています。

この Windows Installer の問題を回避するため、InstallShield を使って以前のバージョンが削除される前に製品の新しいバージョンをインストールするようにプロジェクトを構成することができます。

タスク ***製品の以前のバージョンが削除される前にアップグレードがインストールされるようにプロジェクトを構成して、GAC からアセンブリが削除されないようにするためには、次の手順を実行します。***

1. **[セットアップの編成]** の下にあるビュー リストにある **[アップグレード パス]** をクリックします。
2. **[アップグレード パス]** ペインで、**[アップグレード パス]** を選択します。
3. 右側にある [ヘルプ] ペインで、**[RemoveExistingProducts の並べ替え]** ボタンを選択します。

InstallShield は、InstallFinalize アクションの後に RemoveExistingProducts アクションが発生するように並べ替えます。

ヒント: *デフォルトの動作に戻すには、[RemoveExistingProducts の再シーケンス] ボタンを再度クリックします。*

Windows Installer 問題についての詳細は、「Microsoft ナレッジベース記事: 905238」を参照してください。

アップグレード項目の追加

プロジェクト: *この情報は Express プロジェクトに適用します。*

以前のバージョンの製品が既にリリースされている場合に、エンド ユーザーが以前のバージョンを手動でアンインストールしてから現在のバージョンをインストールしなくても、現在のバージョンに確実にアップグレードできるようにするには、[アップグレード パス] ビューにアップグレード情報を入力します。

タスク ***アップグレード項目を追加するには、次の手順を実行します。***

1. [**アップグレード パス**] ビューを開きます。
2. **[アップグレード パス]** エクスプローラーを右クリックし、**[新しいアップグレード パス]** を選択します。**[開く]** ダイアログ ボックスが開きます。
3. 以下のいずれかを実行します。
   - アップグレードするバージョンのインストールがシステムに存在する場合、.msi ファイルまたは .exe ファイルを参照してから **[開く]** を選択します。InstallShield は、現在のプロジェクトの以前のバージョンからアップグレード コードをコピーします。
   - アップグレードするバージョンのインストールがシステムに存在しない場合は、**[キャンセル]** を選択します。

新しいアップグレード項目が追加されます。

アップグレード プロパティの構成

プロジェクト: *この情報は Express プロジェクトに適用します。*

[アップグレード パス] ビューでアップグレード項目を追加した場合、構成可能ないくつかの設定があります。これらの各設定についての情報は、「アップグレード パスの設定」を参照してください。

アップグレード項目の削除

プロジェクト: *この情報は Express プロジェクトに適用します。*

タスク ***[アップグレード パス] ビューからアップグレード項目を削除するには、次の手順を実行します。***

1. [**アップグレード パス**] ビューを開きます。
2. [**アップグレード パス**] エクスプローラーで、アップグレード項目を右クリックして、[**削除**] を選択します。

パッチ時の考慮事項

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

以下は、QuickPatch プロジェクト作成のためのいくつかのガイドラインです。

圧縮されたインストールのパッチ

パッチの作成プロセスには、以前のインストールと最新のインストールの非圧縮リリースが必要です。インストールが圧縮されているインストールの場合は、リリースの管理用イメージを利用できます。

InstallShield は、QuickPatch プロジェクトで非圧縮のイメージを指定すると、自動的にユーザーのために管理イメージを作成します。

アップデート起動ツール (Update.exe) を作成する

InstallShield では、アップデートに Update.exe アップデート起動ツールを含めるかどうかを指定できます。Update.exe アップデート起動ツールは次のような場合に必要です。

- 必要に応じて、自動的にターゲット システムの Windows Installer エンジンを更新またはインストールする。
- 必要に応じて、.NET Framework をターゲット システムに自動的にインストールする。

Update.exe アップデート起動ツールは、上記のシナリオを制御するブートストラップ アプリケーションです。

InstallShield で Update.exe アップデート起動ツールを作成しないように構成した場合、.msp ファイルが作成されます。

詳細については、「QuickPatch パッケージに Update.exe アップデート起動ツールをビルドをするかどうかを指定する」を参照してください。

Windows Installer 3.0 (およびそれ以上) とパッチ

Windows Installer 3.0 以上には、多数のパッチ関連の強化点があります。パッチまたは QuickPatch が Windows Installer 3.0 以上を用いてターゲットマシンで適用される場合、これらの強化点を利用することができます。詳細は、次を参照してください:

- パッチ シーケンス
- パッチのアンインストール
- 非管理者パッチ

パッチ シーケンス

***プロジェクト**・この情報は、QuickPatch プロジェクトに適用します。*

InstallShield では、ターゲット マシンにパッチが提供された順番に関係なく、Windows Installer バージョン 3.0 以降がインストール済みの製品にスモール アップデート パッチを適用する順番を指定することができます。パッチ シーケンス データを使って、Windows Installer が同じパッチ ファミリー内でパッケージされた各アップグレードの適切な関係を認識できるようにします。従って、製品にパッチ 2 を適用した後にパッチ 1 を適用すると、パッチ 2 のファイルを上書きせずにパッチ 1 が登録されます。Windows Installer バージョン 3.0 以前では、パッチ シーケンスは無視され、すべてのスモール アップデート パッチは、ターゲット マシンに提供された順番で製品に適用されます。

Windows Installer 3.0 以降で利用可能なパッチシーケンス機能は、パッチ作成処理を簡素化します。次のセクションでは、その方法を説明します。

Windows Installer 3.0 以前のバージョンを使って適用されるパッチを作成する

Windows Installer 3.0 より以前のバージョンを利用して製品に適用することができるパッチを作成する必要がある場合、スモールアップデート以外の方法を利用することが推奨されます。スモール アップデートは製品バージョンを変更することはありません。したがって製品の最新版に利用されるインストーラーを含み、外部プログラムは、スモール アップデートを伴う製品およびスモール アップデートを伴わない製品の違いを認識しません。Windows Installer 3.0 以前のバージョンに限ったシナリオでは、こういったインストーラーの制限事項について考慮する必要があるため、以前の製品状態について様々な可能性をターゲットにしなくてはなりません。次のテーブルに図式化されたサンプル アプリケーションのライフサイクルで、それによって生じる複雑さを説明します。

テーブル 5-5・Windows Installer 3.0 以前のバージョンを使って適用されるパッチのサンプル アプリケーション ライフサイクル

アプリケーション パッケージ	製品バージョン	パッケージがターゲットにする以前のセットアップ
1.基本のインストール	1.0	—
2.マイナー アップグレード	1.1	1.0
3.マイナー アップグレード	1.2	1.0, 1.1

テーブル 5-5 • Windows Installer 3.0 以前のバージョンを使って適用されるパッチのサンプル アプリケーション ライフサイクル (続き)

アプリケーション パッケージ	製品バージョン	パッケージがターゲットにする以前のセットアップ
4.マイナー アップグレード	1.3	1.0、1.1、1.2、4
5.マイナー アップグレード	1.4	1.0、1.1、1.2、1.3
6.メジャー アップグレード	2.0	1.0, 1.1, 1.2, 1.3, 1.4

Windows Installer 3.0 を使って適用されるパッチを作成する

Windows Installer 3.0 以降で利用可能なパッチ シーケンス機能を使うと、スモール アップデートは製品バージョンを変更することはありませんが、製品の複数の異なるバージョン用に連続性を持たない 1 つのアップグレードを配布するスモール アップデートを安心して利用することができます。スモール アップデートとは違って、マイナー アップグレードは製品バージョンを変更します。マイナー アップグレードはまた、スモール アップデートパッチのシーケンス用フレームワークを形成します。製品のバージョン 1.1 用のスモール アップデートがバージョン 1.2 に適用された場合、インストーラーはスモール アップデートをターゲット システムに登録して、1.2 マイナー アップグレードが適用される前の段階と同じ要領でそれを適用します。

スモール アップデートパッチはまた、Windows Installer 3.0 以降を有効にし、製品に別のパッチが個別に適用または削除されても製品の有効な状態を保ちます。さらに、パッチ シーケンスを利用すると、ターゲット マシン上に存在する可能性のあるパッチの組み合わせすべてを考慮する必要なく、以前の比較的小さな製品状態の集まりからアップグレード パッケージを生成することができます。次のテーブルに図式化されたサンプル アプリケーション ライフサイクルは、この利点を説明します。

テーブル 5-6 • Windows Installer 3.0 を使って適用されるパッチのサンプル アプリケーション ライフサイクル

アプリケーション パッケージ	パッチ シーケンス番号	製品バージョン	パッケージがターゲットにする以前のセットアップ
1.基本のインストール	—	1.0	—
2.スモール アップデート	1	1.0	1.0
3.スモール アップデート	2	1.0	1.0
4.スモール アップデート	3	1.0	1.0
5.スモール アップデート	4	1.0	1.0

テーブル 5-6 · Windows Installer 3.0 を使って適用されるパッチのサンプル アプリケーション ライフサイクル (続き

アプリケーション パッケージ	パッチ シーケンス番号	製品バージョン	パッケージがターゲットにする以前のセットアップ
6.マイナー アップグレード	–	1.1	1.0

上のテーブル内のスモール アップデートのすべては、同じパッチ ファミリーに所属します。Windows Installer 3.0 以降はパッチ ファミリーを利用して、同じファミリー内でスモール アップデート パッチとその他すべてのパッチを比較し、各パッチをターゲット マシンに適用する順番を決定します。パッチ シーケンスは、パッチ パッケージ データベースの MsiPatchSequence テーブル に追加されます。このテーブルは、同じパッチ ファミリーをターゲットにするパッチの関係を定義します。

パッチのアンインストール

***プロジェクト**· この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以降はスモールアップデートまたはマイナーアップグレードのパッチアンインストールを サポートします。アンインストールするパッチは Windows Installer 3.0 以降を利用してインストールされたものでなくてはなりません。パッチがアンインストールされた時、製品はパッチがインストールされる前の状態に戻ります。以前のバージョンの Windows Installer では、パッチの削除を希望するエンドユーザーはパッチされた製品をアンインストールしてから、パッチを適用せずに製品を再インストールする必要がありました。この場合、削除しないパッチについては再び適用する必要があります。

Windows XP SP2 以降を実行中のシステムでは エンドユーザーは [プログラムの追加と削除]を使ってパッチをアンインストールすることができます。以前のバージョンの Windows で Windows Installer 3.0 以降を実行中のシステムでは、パッチはコマンドラインからアンインストールします。さらに詳しい情報は、Windows Installer ヘルプ ライブラリの「Uninstalling Patches」を参照してください。

すべてのパッチがアンインストール可能ではないため、パッチのアンインストールはデフォルトでは無効に設定されています。たとえば、パッチとしてパッケージされたメジャーアップグレードはアンインストールすることができません。エンドユーザーへ配布する前にパッチのアンインストールを充分テストすることが推奨されます。アンインストール不可能なパッチについてさらに詳しい情報は、Windows Installer ヘルプ ライブラリの「Uninstallable Patches」を参照してください。

非管理者パッチ

***プロジェクト**· この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以上では、管理者以外によるインストールが可能なパッチを作成することができます。非管理者によるパッチは次のすべての条件が満たされた場合に利用することができます。

- Microsoft Windows XP 以降のクライアント プラットフォームの場合、ターゲットマシンが Windows Installer 3.0 以降を実行中。サーバー プラットフォームはサポートしていない。
- アプリケーションは CD-ROM または DVD などのリムーバブル メディアからインストールされている。

- アプリケーションはマシンごとにインストールされている。

メモ: *ALLUSERS プロパティがコマンドラインで上書きされた場合、非管理者パッチは失敗します。*

- ベース インストールには、後に続くすべてのパッチの署名に使用される証明書を含まれていなくてはなりません。
- ベース インストールには、MsiPatchCertificate テーブルを含まれていなくてはなりません。このテーブルは非管理者が後に続くパッチを適用した際、そのデジタル署名を照合するために使用される署名者証明書を提供します。このテーブルには必要に応じて複数の証明書を含むことが可能で、あとに続くパッチは最低 1 つの証明書と照合することが可能でなくてはなりません。詳細については、「非管理者パッチのインストールを準備する」を参照してください。
- 非管理者パッチには、MsiDigitalCertificate テーブルが含まれていなくてはなりません。このテーブルには、署名済みパッチの署名証明書が含まれています。

前述の基準のどれにも当てはまらない場合、エンドユーザーはロックダウンされた環境でデジタル署名済みパッチをインストールすることができません。

非管理者パッチの一般的なシナリオは、コンピューター ゲーム業界で見られます。コンピューター ゲーム ユーザーの一部は子供で、自分のユーザープロファイルおよび HKEY_CURRENT_USER の下にあるレジストリ キー内のフォルダー以外のシステム領域にアクセスが不可能な場合があります。子供がマシンにインストールする内容およびアクセスできる範囲を、彼等の両親が管理者アクセスを利用してコントロールする例が考えられます。両親がすべてのアプリケーションをインストールした場合で、インストール済みのソフトウェアに対するパッチが利用可能なときに、前述の基準がすべて満たされた場合、子供は両親が付き添わなくても自分で非管理者パッチをダウンロード並びにインストールすることができます。

QuickPatch プロジェクトの作成および QuickPatch パッケージの適用

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

QuickPatch プロジェクトは、規模の小さいシングル アップグレードをユーザーへ配布したいインストール作成者へお勧めします。QuickPatch はカスタマイズ可能な範囲が限られてはいますが、[パッチのデザイン] ビューを使わないシンプルなパッチ作成方法として利用できます。基本的にどちらのパッチ作成方法も同じ配布タイプ (.msp と .exe ファイル) を作成します。

既存の .msi ファイルまたは既存の QuickPatch をパッチする QuickPatch プロジェクトを作成するには、新規 QuickPatch 作成ウィザードを利用します。ウィザードを完了すると、QuickPatch プロジェクトに必要な基本条件をすべて満たすことができます。

既存の QuickPatch をパッチする QuickPatch プロジェクトを作成する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

既存の QuickPatch プロジェクトをベースにして、新規 QuickPatch プロジェクトを作成することができます。これは、オリジナルアプリケーションまたは特定のパッチ済みバージョンに対してパッチをあてる必要があるエンドユーザすべてに配布することができるパッチを作成します。

注意: *[一般情報] ビューの [履歴] に表示されるリリースを開いて変更を加えた場合、最新プロジェクトは動作不可能となります。これは [履歴] にあるリリース間で共有されている中間データが不足または改定された可能性があるためです。つまり、既存の QuickPatch プロジェクトをパッチする QuickPatch プロジェクトを作成するたびに InstallShield は既存の QuickPatch プロジェクト (.ise ファイル) のモードを読み取り専用に変更します。*

タスク ***既存の QuickPatch をパッチする QuickPatch プロジェクトを作成するには、以下の手順を実行します。***

1. InstallShield で、パッチをあてる QuickPatch プロジェクト (.ise ファイル) を開きます。
2. **[ツール]** メニューで、**[QuickPatch の作成]** をクリックします。

新規 QuickPatch プロジェクトが InstallShield で開きます。

新規 QuickPatch 作成ウィザードを使用して、既存の QuickPatch に QuickPatch を作成することもできます。

QuickPatch パッケージを簡素化するかどうかを指定する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

QuickPatch プロジェクトの構成を行うとき、InstallShield で QuickPatch パッケージの作成を簡素化して、最もシンプルなパッケージをビルドするかどうか指定することができます。QuickPatch の簡素化を行うと、通常の QuickPatch パッケージに比べて新しいサブ機能とカスタム アクションの数が少ない QuickPatch パッケージを生成できます。

たとえば、QuickPatch プロジェクトに新しいファイルとレジストリ エントリが含まれていて、 が QuickPatch の簡素化を行わなかった場合、そのファイルとレジストリ エントリ用に新しいサブ機能が作成されます。さらに、特定の Windows Installer パッチ要件に対応するため、1 つまたは複数のビルド済み カスタム アクションが追加されます。これに対し、InstallShield が QuickPatch の簡素化を行う場合、ファイルまたはレジストリ エントリは既存する機能に追加されるため、特別なビルド済み InstallShield カスタム アクションは必要ありません。

メモ: *次のシナリオにおいて、InstallShield が QuickPatch パッケージの作成処理を簡素化することはできません。*

- *QuickPatch パッケージがインストール済みのファイルを削除する。*
- *QuickPatch パッケージがレジストリ キーを削除する、またはその名前を変更する。*
- *QuickPatch パッケージが、簡素化されていない通常の QuickPatch イメージをターゲットとする。つまり、[一般情報] ビューの [履歴] 領域で、簡素化を行わなかった QuickPatch のチェック ボックスを選択した場合、QuickPatch の簡素化はできないということです。1 つまたは複数の簡素化されていない QuickPatch イメージをターゲットとする簡素化された QuickPatch のビルドを試みると、ビルド警告が表示され、簡素化は行われません。*

また、15 未満の簡素化されていない累積 QuickPatch パッケージをベース .msi パッケージまたはメジャー アップグレード パッケージに適用できる点にご注意ください。制限数を超えると、パッチを適用中に エラー 2701 が発

生します。正確な制限は、パッケージの機能ツリーおよび簡素化されていない各 QuickPatch がそのツリーに追加するサブ機能の階層数によって決まります。Windows Installer の機能ツリー階層数の制限は 16 レベルです。

簡素化された QuickPatch パッケージには、新しいサブ機能が含まれないため、この制限がありません。

タスク ***QuickPatch パッチを簡素化するかどうかを指定するには、以下の手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[詳細]** タブをクリックします。
4. **"QuickPatch の簡素化"** 設定で、適切なオプションを選択します:
   - QuickPatch パッケージを簡素化する場合、**[はい]** を選択します。この値は、新しい QuickPatch プロジェクトのデフォルト値です。
   - QuickPatch パッケージを簡素化を選択しない場合、**[いいえ]** を選択します。

パッチのターゲット リリースを指定する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

タスク ***QuickPatch プロジェクトでどのリリースをパッチするのかを指定するには、次の手順を実行します。***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[履歴]** を選択します。
3. 中間リリース (ベース QuickPatch イメージと現在の QuickPatch イメージの中間となるリリース) の隣にあるチェック ボックスを選択またはクリアして、現在のプロジェクトでパッチするのかどうかを指定します。

メモ: *ベース QuickPatch イメージまたは現在のプロジェクトのチェック ボックスをクリアすることはできません。中間 QuickPatch プロジェクトのチェック ボックスをクリアすると、現在の QuickPatch プロジェクトを使って、アプリケーションの最新イメージとして中間イメージを持つマシンをアップグレードすることはできません。*

たとえば、インストールとしてバージョン 1.0 を配布した場合で、その後オリジナルインストールをバージョン 1.1 にアップグレードするパッチをリリースした場合。バージョン 1.2 QuickPatch を作成して、履歴で 1.1 QuickPatch のチェック ボックスが選択されていない場合、 1.2 QuickPatch は 1.0 リリースのアップグレードに利用できますが、1.1 リリースには利用できません。

QuickPatch を実行するカスタム アクションを指定する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

タスク ***QuickPatch の適用時に実行するカスタム アクションを指定するには、以下の手順を実行します。***

1. [**パッチの設定**] の下にあるビュー リストで、[**一般情報**] をクリックします。
2. [**一般情報**] エクスプローラーで [**カスタム アクション**] をクリックします。
3. カスタム アクションの隣にあるチェック ボックスを選択またはクリアして、QuickPatch のインストール中にそれらを実行するかどうかを指定します。

QuickPatch へファイルを追加する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

メモ: *QuickPatch プロジェクトは、追加する新しいファイルに新しいターゲットの場所を定義できるサポートを含みません。そのため、QuickPatch プロジェクトに新しいファイルを追加する場合、そのインストール場所は、元のインストールで定義されたフォルダーでなくてはなりません。*

タスク ***QuickPatch にファイルを追加するには、以下の手順を実行します。***

1. [**パッチ設定の定義**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**パッチするファイル**] エクスプローラーを右クリックし、[**新規ファイルの挿入**] をクリックします。[**開く**] ダイアログ ボックスが開きます。
3. 追加するファイルをクリックして [**開く**] をクリックします。[**パッチするファイル**] エクスプローラーにファイルが追加されます。
4. 新しいファイルをクリックしてから、その設定を構成します。

QuickPatch の識別情報を指定する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以降は、QuickPatch がアンインストール不可能な場合も、適用された各 QuickPatch についての [プログラムの追加と削除] エントリをターゲット システムに追加します。

タスク ***QuickPatch の識別情報を指定するには、以下の手順に従います:***

1. [**パッチの設定**] の下にあるビュー リストで、[**一般情報**] をクリックします。
2. [**一般情報**] エクスプローラーで、[**ビルドの設定**] を選択します。
3. [**識別**] タブをクリックします。

4. 個々の設定を構成します。

QuickPatch のアンインストールを有効にする

***プロジェクト・** この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以降は、スモール アップデートおよびマイナー アップグレード用の QuickPatch パッケージのアンインストールをサポートします。ただし、すべての QuickPatch パッケージがアンインストール可能という訳ではありません。標準パッチおよび QuickPatch パッケージのアンインストールについての詳細および、関連する制限事項については、「パッチのアンインストール」を参照してください。

タスク ***QuickPatch パッケージのアンインストールを有効にするには、以下の手順を実行します。***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[共通]** タブをクリックします。
4. **[パッチのアンインストールを許可する (Windows Installer 3.0 が必要)]** チェック ボックスを選択します。

QuickPatch パッケージをシーケンスする

***プロジェクト・** この情報は、QuickPatch プロジェクトに適用します。*

タスク ***QuickPatch パッケージのシーケンスを定義するには、以下の手順を実行します。***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[詳細]** タブをクリックします。
4. 以下のいずれかを実行します。
   - InstallShield が生成するデフォルトのパッチ シーケンスを使用するには、**"パッチ シーケンス エントリの作成"** プロパティを **[はい]** に設定します。
   - パッチ シーケンスを利用しない場合は、このプロパティを **[いいえ]** に設定します。

QuickPatch パッケージに署名する

***プロジェクト・** この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以上では、管理者以外によるインストールが可能なパッチを作成することができます。非管理者による QuickPatch パッケージは厳しい条件が満たされたときのみ利用することができます。たとえば、パッチが更新するベース インストールは、パッチ パッケージの署名に使用される証明書を含まなくてはなりません。満たされなければならない他の基準については、「非管理者パッチ」を参照してください。

タスク ***パッチパッケージを署名するには、以下を手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[デジタル署名]** タブをクリックします。
4. **[パッチ パッケージの署名]** チェック ボックスを選択します。
5. デジタル署名設定を構成します。

QuickPatch パッケージをパスワードで保護する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

セキュリティをさらに向上させるため、QuickPatch パッケージをパスワードで保護することができます。QuickPatch パッチをパスワードで保護すると、QuickPatch パッチを適用するエンド ユーザーは、アップデートを起動するために大文字小文字の区別があるパスワードを入力する必要があります。

タスク ***QuickPatch パッケージをパスワードで保護するには、以下の手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[詳細]** タブをクリックします。
4. **"起動ツールをパスワードで保護"** 設定で、**[はい]** を選択します。
5. **"起動ツールのパスワード"** 設定に、パッチで使用するパスワードを指定します。

メモ: *パスワードは大文字と小文字を区別します。*

QuickPatch パッケージに Update.exe アップデート起動ツールをビルドをするかどうかを指定する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

InstallShield では、InstallShield で、作成中の QuickPatch パッケージに Update.exe アップデート ランチャを作成するかどうかを指定できます。

必要に応じて自動的にターゲット システムで Windows Installer サービスを更新またはインストールする場合、Update.exe アップデート起動ツールが必要です。Update.exe 起動ツールが必要になる場合についての詳細は、「パッチ時の考慮事項」を参照してください。

InstallShield で Update.exe アップデート起動ツールを作成しないように構成した場合、.msp ファイルが作成されます。

タスク ***QuickPatch に Update.exe アップデート起動ツールを含めるかどうかを指定するには、以下の手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[共通]** タブをクリックします。
4. Update.exe を含める場合、**[Update.exe を作成する]** チェック ボックスを選択します。

   Update.exe を含めない場合、**[Update.exe を作成する]** チェック ボックスをクリアします。

ヒント: *[ビルドの設定] 領域にある [詳細] タブを利用しても、Update.exe を含めるかどうかを指定することができます。*

アップデート起動ツールのファイルのプロパティをカスタマイズする

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

InstallShield では、Update.exe アップデート起動ツールのバージョン リソースにカスタム情報を使用できます。この情報は、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。[プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックしたときに表示されます。

メモ: *[プロパティ] ダイアログ ボックスは、Windows のバージョンによって異なります。たとえば、Windows 7 システムでは、バージョン リソース情報は [プロパティ] ダイアログ ボックスの [詳細] タブに表示されます。一方、Windows XP システムでは、バージョン リソース情報は、同じダイアログボックスの [バージョン] タブに表示されます。*

また、Windows の一部のバージョンは、[プロパティ] ダイアログ ボックスの一部の設定を表示しません。

Update.exe ファイルの [プロパティ] ダイアログ ボックスと同じ概観を持つ Setup.exe ファイルの [プロパティ] ダイアログ ボックスのスクリーンショットを見るには、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。

タスク ***Update.exe 設定のデフォルトの InstallShield 値を独自のカスタム値でオーバーライドするには、以下の手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[詳細]** タブをクリックしてから、**[アップデート起動ツールの設定]** 領域を探します。
4. 以下の設定に、Update.exe ファイルのプロパティに使用する値を入力します。
   - 会社名
   - 製品名
   - 製品バージョン
   - 説明
   - 著作権情報

ヒント: *"製品バージョン" 設定は、Update.exe のファイル バージョンと製品バージョンを更新します。ファイル バージョンは、常に 4 つのフィールドで構成されます。4 フィールドよりも少ないフィールドを指定すると、残りのフィールドには 0 が挿入されます。たとえば、製品バージョンとして 1.1 を指定すると、Update.exe のバージョン リソースで使用されるファイル バージョンは 1.1.0.0 となります。*

アップデート起動ツールの設定を空白のままに残すと、InstallShield はデフォルトの InstallShield 値を使用します。

[パッチするファイル] エクスプローラーからファイルを削除する

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

タスク ***[パッチするファイル] エクスプローラーからファイルを削除するには、以下の手順に従います:***

1. **[パッチ設定の定義]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. 削除するファイルを右クリックして、**[削除]** をクリックします。

QuickPatch を使用したインストール済みファイルの変更と削除

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

タスク ***QuickPatch と共にインストールされたファイルを変更または削除するには、以下の手順に従います:***

1. [**パッチ設定の定義**] の下にあるビュー リストで、[**ファイル**] をクリックします。
2. [**パッチするファイル**] エクスプローラーを右クリックし、[**既存のファイルのパッチ**] をクリックします。[**ファイルの選択**] ダイアログ ボックスが開きます。
3. 変更または削除するファイルをクリックします。[**パッチするファイル**] エクスプローラーにファイルが追加されます。
4. [**パッチするファイル**] エクスプローラーに追加したばかりのファイルをクリックして、設定を構成します。

ヒント: *また、上記 3、4 の手順を踏まずに、直接ファイルまたはフォルダーを [**元のセットアップ ファイル**] から [**パッチするファイル**] へドラッグ アンド ドロップすることもできます。*

QuickPatch を使用したレジストリデータの追加、変更および削除

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

QuickPatch プロジェクトでレジストリデータを追加、変更または削除するときの手順は、基本的に、オリジナルのインストールにそれを行うときと同じ要領です。唯一異なる点は、QuickPatch プロジェクトのレジストリデータを追加する前に、[レジストリ] ビュー上部にある [ビューフィルター] リスト内で既存の機能を選択する必要があるという点です。QuickPatch では新規機能を追加することができないため、レジストリ データはすべて、既にオリジナルの製品に存在する機能に関連付ける必要があります。

ヒント: *QuickPatch プロジェクト用に変更を加えたレジストリの設定を変更するには、まずアイテムを右クリックしてから [取り消す] をクリックします。*

グローバル アセンブリ キャッシュのアセンブリをパッチする

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

Windows Installer 3.0 以上で、MsiPatchOldAssemblyFile および MsiPatchOldAssemblyName テーブルを利用すると、パッチパッケージでオリジナル インストール ソースのランタイム要求なしにグローバル アセンブリ キャッシュ (GAC) 内のアセンブリをパッチすることができます。デフォルトでは、QuickPatch パッケージをビルドすると InstallShield がこれらのテーブル用のエントリを自動的に生成します。(QuickPatch プロジェクトでこの自動生成を無効にするには、**"MsiPatchOldAssembly テーブルの生成"** プロパティを [いいえ] に設定します。QuickPatch の場合、このプロパティは [一般情報] ビューのビルドの設定アイテムの [詳細] タブにあります。

メモ: MsiPatchOldAssemblyFile *および* MsiPatchOldAssemblyName *テーブルのエントリを自動生成するためには、InstallShield はパッチを適用する .msi パッケージの最新バージョンへの書き込みアクセスを必要とします。*

InstallShield は、このパッケージをパッチ作成の前に変更します。InstallShield はパッケージを書き込み可能にしますが、それが不可能だった場合はビルド警告を生成し、テーブルエントリは作成されません。

*これらのテーブルエントリはターゲット システムが Windows Installer 3.0 以上を実行している場合のみ適用されます。システムが Windows Installer 2.0 を実行中の場合、パッチは実行しますがこれらのテーブルは無視され、パッチが GAC 内のファイルを更新する必要がある場合はオリジナルソースパッケージを要求します。*Update.exe *に Windows Installer 2.0 エンジンのみを含む場合でも、InstallShield は、これらのテーブル エントリを生成します。これは既に Windows Installer 3.0 以上がインストールされたターゲット システムがこれらのテーブル エントリを利用するためです。*

QuickPatch の適用

プロジェクト: *この情報は、QuickPatch プロジェクトに適用します。*

QuickPatch パッケージ (.msp) ファイルには、インストールされた 1 つまたは複数の製品バージョンをアップグレードするのに必要なトランスフォームおよび指示が含まれています。

QuickPatch パッケージを適用するには、Windows Installer v. 1.1 以降がシステムにインストールされている必要があります。またアップデートするパッケージは、管理イメージとして、またはローカルにインストールされている必要もあります。さらに別の要件として、既存のインストール パッケージが、QuickPatch パッケージと同じ権限を使って同じユーザー向けにインストールされている必要があります。たとえば製品がすべてのユーザー用としてインストールされている場合は、アップデートもすべてのユーザー用にインストールする必要があります。

QuickPatch パッチを適用する場合、Windows エクスプローラーで .msp ファイルをダブルクリックするか、またはファイルを右クリックして [開く] を選択するのが最も簡単です。マイナーアップグレードのパッチを適用すると、最初に開くダイアログは PatchWelcome ダイアログです。メジャー アップグレードのパッチを適用すると、インストールをスタンドアロンで実行したときに表示されるような完全ダイアログ シーケンスが表示されます。

MsiExec.exe /p オプションを使用して、コマンドラインからパッチを適用できます。次のステートメントを入力すると、X:¥Product Updates¥Build 36¥PatchForV1.msp にあるパッチパッケージが適用されます。

```
msiexec /p "X:¥Product Updates¥Build 36¥PatchForV1.msp"Update.exe
```

[一般情報] ビューの [ビルドの設定] 領域で、QuickPatch パッチ構成の [Update.exe の作成] チェック ボックスを選択すると、.msp ファイルが実行可能ファイルにラップされます。次のコマンド ライン指定すると、Update.exe によってパッチ パッケージが起動されます。

```
msiexec /p <path to .msp file> REINSTALL=ALL REINSTALLMODE=omus
```

QuickPatch をサイレント モードで適用する

サイレント モードで QuickPatch を適用する場合、2 つの方法があります。/qn コマンド ライン パラメーターを使って MsiExec.exe を起動するか、/s を Update.exe へ渡します。

サイレント モードで QuickPatch を適用する場合、考慮を要する重要な点があります。正しく動作させるには、QuickPatch を適用する際に、Windows Installer のプロパティ REINSTALL が **ALL** に、REINSTALLMODE が **omus** に設定されている必要があります。Update.exe は常にこれらのプロパティをコマンド ラインで設定するため、QuickPatch パッケージが Update.exe を使って適用された場合は、ユーザーが特別な操作を行う必要はありません。

完全なユーザー インターフェイスを使って QuickPatch パッケージを適用すると、インストールのデフォルトのダイアログの 1 つ、PatchWelcome が表示されます。ここには、正しいオプションを使って REINSTALL および REINSTALLMODE を設定するためのコントロール イベントが含まれています。ただしこのダイアログは、エンドユーザー インターフェイスが抑制されていると表示されないため、次のようにコマンドラインでプロパティを設定する必要があります。

msiexec /p <.msp ファイルへのパス> /qn REINSTALL=ALL REINSTALLMODE=omus

QuickPatch は既存する .msi データベースのキャッシュを変更しないため、REINSTALLMODE に **v** 設定を含める必要はありません。

FlexNet Connect を利用してエンドユーザーにアップグレードの通知をする

FlexNet Connect を利用して、Web に接続しているエンド ユーザーに対してアプリケーションのパッチ、アップデート、および製品情報が入手可能であることを自動的に通知します。

FlexNet Connect の実装

FlexNet Connect を利用してエンドユーザーに対して自動的にアップデートを通知する作業には、大きく分けて 2 つのサイクル (初期配布とアップデート配布) があります。アプリケーションの初期配布段階を完了した後、そのアプリケーションのアップデートを顧客に配布するたびにアップデート配布に関する一連の作業を行います。

アップデート配布

1. InstallShield を使ってアプリケーションのアップデートを作成します。
2. 新しい製品バージョンおよび製品コードを FlexNet Connect パブリッシャー サイト(Web ベースの管理ポータル)に登録します。
3. FlexNet Connect パブリッシャー サイトにアップデートをパブリッシュし、**[メッセージ ステータス]** を **[テスト]** に設定します。
4. アップデートをテストします。
5. FlexNet Connect パブリッシャー サイトにアップデートをパブリッシュし、**[メッセージ ステータス]** を **[アクティブ]** に設定します。

FlexNet Connect には様々なオプションがあり、完全ソリューションとして本製品と共に購入することもできますし、またはカスタマイズ ソリューションとして個別に購入することもできます。詳しい情報は、Revenera Web サイトをご覧ください。

6

追加のインストール オプション

InstallShield では、製品をインストールするインストール パッケージの作成を支援するだけでなく、最終インストール パッケージを強化する他のインストール オプションも提供されています。このセクションは、追加のインストール オプションに関する説明です。

ターゲット システムの要件を指定する

InstallShield で、ターゲット システムのインストール要件を簡単に設定することができます。たとえば、製品を適切に実行するために特定の OS またはアプリケーションが必要な場合、インストール プロジェクトで指定することができます。ターゲット システムが要件を満たさない場合、インストールが終了します。

製品のオペレーティング システム要件を指定する

InstallShield では、製品の要件として特定のオペレーティング システムを指定することができます。ターゲット システムが要件を満たさない場合、インストールが終了します。

タスク ***ターゲット システムのオペレーティング システム要件を指定するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[要件]** をクリックします。
2. 中央のペインで、**[システム ハードウェアの要件]** を選択します。
3. ″**OS バージョン**″ 設定で、省略記号ボタン (...) をクリックします。**[システム ハードウェアの要件]** ダイアログ ボックスが開きます。
4. 適切なオペレーティング システムの要件を指定します。

″OS バージョン″ 設定の値が更新されます。

ヒント: *製品のオペレーティング システムの要件を指定すると、実質的に製品をサポートしないオペレーティング システムが除外することになります。*

たとえば、最新 Windows オペレーティング システムのチェック ボックスのみを選択した場合、[要件] ページで選択しなかったオペレーティング システムを除外する起動条件が作成されます。この起動条件タイプでは、将来的にリリースされる Windows オペレーティング システムのバージョンが自動的にサポートされます。これは起動条件で除外されていないためです。

製品のプロセッサの要件を指定する

製品に特定のプロセッサ クラスが必要なとき、製品が互換性のないプロセッサを持つシステム上にインストールされないようにすることができます。ターゲット システムのプロセッサが指定された最低要件に満たない場合、インストールが終了します。

タスク ***ターゲット システムのプロセッサ要件を指定するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**要件**] をクリックします。
2. 中央のペインで、[**システム ハードウェアの要件**] を選択します。
3. **"プロセッサ"** 設定で、適切なプロセッサを選択します。

製品の RAM 要件を指定する

製品を適切に実行するために一定の容量の RAM が必要なとき、容量が満たないシステム上に製品がインストールされないようにすることができます。ターゲット システムに必要最低限度の容量がない場合、インストールが終了します。

タスク ***ターゲット システムの RAM 要件を指定するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**要件**] をクリックします。
2. 中央のペインで、[**システム ハードウェアの要件**] を選択します。
3. **"RAM"** 設定で、適切な値を選択します。

製品の画面の解像度要件を指定する

製品を適切に実行するために特定の画面解像度が必要なとき、画面解像度が満たないシステム上に製品がインストールされないようにすることができます。インストールが実行されるマシンの解像度が指定した解像度を下回る場合、インストールは終了します。

タスク ***ターゲット システムの画面解像度要件を指定するには、以下の手順に従います:***

1. ビュー リストの [**セットアップ要件およびアクションの定義**] の下にある [**要件**] をクリックします。
2. 中央のペインで、[**システム ハードウェアの要件**] を選択します。
3. **"画面解像度"** 設定で、適切な値を選択します。

カラー深度の要件を指定する

製品を適切に実行するために特定のカラー深度が必要なとき、カラー深度が満たないシステム上に製品がインストールされないようにすることができます。ターゲット システムのカラー深度が選択された最低条件に満たないとき、インストールが終了します。

タスク *ターゲット システムのカラー深度要件を指定するには、以下の手順に従います:*

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[要件]** をクリックします。
2. 中央のペインで、**[システム ハードウェアの要件]** を選択します。
3. **"カラー深度"** 設定で、適切な値を選択します。

製品のソフトウェアの要件を指定する

製品を適切に実行するために特定のソフトウェアが必要なとき、そのソフトウェアを持たないシステム上に製品がインストールされないようにすることができます。必須のソフトウェアがないシステムでインストールが実行されると、不足しているソフトウェアの説明がエラー メッセージで表示され、インストールが終了します。

[要件] ビューにある [システム ソフトウェアの要件] エクスプローラーには、Adobe Reader や .NET Framework などのソフトウェアのビルトイン要件の条件が含まれています。これらのビルドイン要件を追加することもできますし、要件を自分で構成することもできます。

タスク *既存のソフトウェアの要件の 1 つをプロジェクトに追加するには、以下の手順に従います:*

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[要件]** をクリックします。
2. **[システム ソフトウェアの要件]** エクスプローラーで、プロジェクトに追加するソフトウェアの条件を見つけます。

   条件ステータスを編集するには、隣に適切なアイコンが表示されるまで条件の隣にある設定をクリックします。

   テーブル 6-1 · システム ソフトウェア要件の条件のアイコン

アイコン	説明
✔	ソフトウェア条件はインストールが続行するのに True でなくてはなりません。
✘	ソフトウェア条件はインストールが続行するのに False でなくてはなりません。
☐	ソフトウェア条件は確認されず、インストールに影響も与えません。

*ヒント: 新しいソフトウェア条件を既存の要件リストに追加するには、**[システム ソフトウェア要件]** エクスプローラーを右クリックして、**[新規条件の作成ウィザード (システム ウィザード)]** をクリックします。この後、システム検索ウィザードが開き、それを使ってカスタム要件を作成することができます。詳細については、「システム検索ウィザード」を参照してください。*

Windows Installer プロパティを使用する

Windows Installer プロパティを使用して、インストール全体で共通パスおよびユーザー情報の変数を使用できます。これらのプロパティは、ダイアログ テキスト、.ini ファイルの変更、カスタム アクション、レジストリ エントリで使用できます。プロパティの一覧については、「Windows Installer プロパティ リファレンス」をご覧ください。

これらのプロパティを使用するとき、プロパティを角かっこで囲みます。たとえば、ダイアログでインストールディレクトリを表示する場合、[テキストとメッセージ] ビューで、次のように入力することができます。

このプロジェクトは [INSTALLDIR] にインストールされます。

この文は、エンドユーザーに対して次のように表示されます。

このプログラムは C:¥Program Files¥*会社名*¥*製品名* にインストールされます。

プロパティの種類

Windows Installer のプロパティには 4 つの一般的な種類があります。

- パブリック
- プライベート
- 制限付きパブリック
- 必要

メモ: これらのカテゴリの一部は重複しています。たとえば ProductCode プロパティは、必須のプライベート プロパティです。

パブリック プロパティ

パブリック プロパティの名前には、大文字だけが使用されます。たとえば INSTALLDIR はパブリック プロパティです。パブリック プロパティはコマンドラインで指定することが可能で、インストールの起動に利用したり、ユーザー インターフェイスを利用して選択するのに利用できます。

メモ: インストールのユーザー インターフェイスからの値を、インストールでターゲット システムを変更する時点まで保持するのはパブリック プロパティだけです。エンドユーザーに表示されるダイアログのプロパティの値を設定する場合、その値をファイルまたはレジストリに書き込むようにするには、パブリック プロパティを使用するようにしてください (例 MY_PUBLIC_PROPERTY)。

プライベート プロパティ

プライベート プロパティには、その名前に最低小文字が 1 つ使用され、ユーザー インターフェイスから変更することはできません。たとえば ProgramFilesFolder はプライベート プロパティです。プライベートプロパティの値は、コマンドラインから設定できないので、エンドユーザーが制御することはできません。

制限付きパブリック プロパティ

制限付きパブリックプロパティでは、ネットワーク管理者がシステム管理者のみが変更できるパブリックプロパティを定義することができます。これにより管理者は、ネットワークの他のユーザーがセットアップを変更することを心配せずに、即座に設定を変更できます。詳細については、「パブリック プロパティが制限付きパブリック プロパティである必要があることを指定する」を参照してください。

必須プロパティ

Windows Installer サービスは、すべての Windows Installer インストールで必要な 5 つのプロパティに依存しています。デフォルトで、これらのプロパティは InstallShield を使用して作成したインストールすべてに含まれます。

- ProductCode
- ProductLanguage
- Manufacturer
- ProductVersion
- ProductName

条件

多くのプロパティは、インストールが起動されるまで設定されません。こうしたプロパティには、ターゲット システムからの情報が与えられます。たとえば、VersionNT プロパティは、インストールが起動されるまで設定されません。オペレーティング システムが Windows 2000 以降の場合、このプロパティは、ターゲット マシンで実行されている Windows のバージョンに設定されます。

実行時に設定したプロパティは、インストールの起動条件を作成するとき使用することができます。Windows XP のみに製品をインストールする場合は、条件付き論理を使ってエンド ユーザーのシステムを確認し、すべての条件を満たした場合に製品をインストールするように設定することができます。

Windows Installer プロパティ リファレンス

Windows Installer のプロパティを使用して、Windows Installer サービスでインストールの特定部分を参照することができます。これらのプロパティは、カスタム アクション、.ini ファイル キーワード値、およびダイアログ テキストで使用できます。プロパティの詳しい設定方法については、「Windows Installer プロパティを使用する」をご覧ください。

Windows Installer プロパティの次のカテゴリはこのトピックに説明されています。リンクをクリックすると直接カテゴリに移動できます。

- 特別なフォルダーとファイルのプロパティ
- 機能のインストールに関するプロパティ
- その他の構成可能なプロパティ
- ユーザーが提供する情報

- 製品固有のプロパティ
- インストーラーが設定するシステム フォルダー
- インストーラーにより設定されるオペレーティング システムのプロパティ
- インストーラーにより設定されるハードウェア プロパティ
- インストーラーにより更新されるステータス プロパティ
- 日付と時刻のプロパティ

メモ: インストール用のプロパティとパス変数の違いに注意してください。パス変数は山かっこ (<>) で囲みます。どちらもディレクトリを表しますが、Windows Installer プロパティは実行時に使用できるのに対し、パス変数は、セットアップを設計、作成するときにソースファイルを指すためにだけ使用できます。

特別なフォルダーとファイルのプロパティ

特別なフォルダーのプロパティは、ターゲット システムに格納されたり、インストールされるファイルの場所を定義します。ファイルのプロパティは特定のファイルを参照します。

インストールでフォルダー プロパティを使用するには、プロパティを角かっこ ([]) で囲みます。

テーブル 6-2 · フォルダーのプロパティ

プロパティ名	説明
INSTALLDIR	このプロパティには、機能内のファイルのデフォルトのインストール先が含まれます。詳細については、「デフォルトの製品インストール先フォルダー (INSTALLDIR) の設定」を参照してください。

テーブル 6-2 • フォルダーのプロパティ (続き)

プロパティ名	説明
SETUPEXEDIR	SETUPEXEDIR プロパティには、 Setup.exe へのパスが含まれています。たとえば、Setup.exe へのパスが C:¥MySetups¥MyApp¥Setup.exe とすると、SETUPEXEDIR の値は C:¥MySetups¥MyApp になります。 ***メモ:*** *SETUPEXEDIR の使用には 2 つの制限があります。* • *SETUPEXEDIR は Setup.exe によって設定されます。エンドユーザーが .msi パッケージを直接実行した場合、SETUPEXEDIR は設定されません。これに対処するため、SETUPEXEDIR を使うインストールと SourceDir を使うインストールの 2 つの実装をインストールに含めることができます。SETUPEXEDIR の存在をテストし、存在しない場合は条件付きで SourceDir 実装を使用することができます。* • *SETUPEXEDIR はアンインストール時に設定されない場合があります。エンドユーザーが Setup.exe を実行することによってアンインストールをトリガーすると、SETUPEXEDIR は設定されます。アンインストールが [プログラムの追加と削除] から実行されると、SETUPEXEDIR は設定されません。* 非圧縮インストールでは、SourceDir と SETUPEXEDIR は同じ値を持つので注意が必要です。
SETUPEXENAME	プロジェクトがビルドされるときに作成される、セットアップランチャー ファイルの名前を識別します。セットアップランチャー ファイルの名前が変更された場合、インストールは実行時に、このプロパティ値を更新します。以下のパスは、このファイルへの完全パスを示します: [SETUPEXEDIR]¥[SETUPEXENAME]

テーブル 6-2・フォルダーのプロパティ（続き）

プロパティ名	説明
SourceDir	このプロパティは、すべてのソース ファイルが保存されているルート ディレクトリを格納します。 ***注意***・*リリースがセットアップランチャーに圧縮された .msi パッケージを含み、.msi パッケージがローカル マシンにキャッシュされない場合、SourceDir はインストール中に参照されますが、アンインストールまたはメンテナンス中には参照されません。これらの状況では、セットアップ イメージはインストール後に削除される一時的な場所から実行されます。* SETUPEXEDIR は、ディレクトリ識別子 SourceDir の代わりに使用できます。SourceDir を使用したとき、実行中の .msi パッケージの場所がポイントされるという問題が起こることがあります。圧縮インストールの場合、.msi パッケージは一時的な場所にストリームされて、そこから実行されます。このため、SourceDir の値はエンドユーザーのマシン上の一時的な場所になり、その値が期待していた値とは異なる場合もあります。
TARGETDIR	TARGETDIR プロパティは、管理インストール時に Windows Installer パッケージ（非圧縮データ ファイル）がコピーされる場所を表します。

機能のインストールに関するプロパティ

以下のセクションでは、機能のインストールに関するプロパティが説明されています。これらのプロパティを利用して、エンドユーザーは機能をどのようにインストールするかを指定することができます。

テーブル 6-3・ 機能のインストールに関するプロパティ

プロパティ名	説明
ADDDEFAULT	ADDDEFAULT プロパティは、デフォルトの構成でインストールされる機能のリストをコンマで区切って格納します。このプロパティの値を ALL に設定すると、ユーザーはすべての機能をデフォルトの設定でインストールできます。
ADDLOCAL	このプロパティは、ローカルにインストールされる機能のリストをカンマで区切って格納します。各機能には、インストール用に選択された機能がローカルにインストールされるか、ソース メディアから実行されるかを決めるための Remote Installation プロパティがあります。
ADDSOURCE	このプロパティは、ソース メディアから実行される機能のリストをカンマで区切って格納します。このプロパティが ALL に設定される場合、すべての機能がソース メディアから実行されます。

テーブル 6-3・（続き)機能のインストールに関するプロパティ

プロパティ名	説明
ADVERTISE	このプロパティは、アドバタイズされる機能のリストをカンマで区切って格納します。
REINSTALL	このプロパティは、再インストールされる機能のリストをカンマで区切って格納します。REINSTALL が ALL に設定されていると、ユーザーのシステムに既にインストールされているすべての機能が再インストールされます。
REINSTALLMODE	このプロパティには、再インストールの種類を指定する文字列が含まれます。これらのオプションは、Msiexec.exe /f コマンドラインパラメーターで使用可能な値に対応します。このプロパティの詳細は、Windows Installer ヘルプ ライブラリの「REINSTALLMODE Property」プロパティ トピックを参照してください。 REINSTALLMODE は、常に REINSTALL と一緒に設定されます。
ReinstallModeText	このプロパティには、ユーザーが MaintenanceType ダイアログで [修復] を選択したときに設定される再インストール オプションが含まれます。ReinstallModeText の値はコントロール イベント ReinstallMode に引数として渡されます。このコントロール イベントは、REINSTALLMODE プロパティで使用可能なオプションと同じオプションを受け付けます。 ReinstallModeText は定義済みの Windows Installer プロパティではありません。
REMOVE	このプロパティは、削除される機能のリストをコンマで区切って格納します。REMOVE が ALL に設定されていると、すべての機能が削除されます。
COMPADDLOCAL	このプロパティは、ローカルにインストールされるコンポーネント ID のリストをカンマで区切って格納します。最小のディスク容量を使用するコンポーネントの機能がインストールされます。
COMPADDSOURCE	このプロパティは、ソース メディアから実行されるコンポーネント ID のリストをカンマで区切って格納します。最小のディスク容量を使用するコンポーネントの機能がインストールされます。
PATCH	パッチをインストールするとき、このプロパティにはパッチ パッケージへの完全パスが含まれます。

その他の構成可能なプロパティ

以下のセクションでは、その他各種の構成可能なプロパティについて説明します。

テーブル 6-4・追加の構成可能プロパティ

プロパティ名	説明
ACTION	このプロパティは、実行するシーケンス ([インストール]、[アドバタイズ] または [管理]) を指定します。使用できる値は INSTALL、ADVERTISE および ADMIN です。ACTION プロパティは、インストーラー起動に使用されたコマンドラインに基づいて、自動的に設定されます。
ALLUSERS	Windows Installer が、マシンごとまたはユーザーごとのインストールを試みるかどうかを指定します。 ALLUSERS の値が 1 に設定されている場合、Windows Installer はマシンごとのインストールを試みます。マシンごとのインストールの場合、ショートカット、レジストリ エントリなどの構成情報は All Users のプロファイルに格納されます。 ・ Windows Vista 以降のシステムでは、[ユーザー アカウント制御] が有効にされていて、ユーザーが管理者権限を持たないとき、製品をインストールするためには管理資格情報が必要です。 ・ 他のシステムでは、ユーザーが管理者権限を持たないとき、インストールはエラー メッセージを表示して終了します。 ***プロジェクト・** すべての Express プロジェクトで、このプロパティのデフォルトは 1 に設定されています。詳細については、「ユーザーごとのインストールとマシンごとのインストールの違い」を参照してください。* ALLUSERS の値が設定されていなく、値が空の文字列 (“”) の場合、Windows Installer はユーザーごとのインストールを実行し、構成情報はユーザーの個人プロファイルに格納されます。 ALLUSERS の値が 2 に設定されている場合、Windows Vista 以降のシステムで、Windows Installer はマシンごとのインストールを試みます。以前のプラットフォームでは、ユーザーに権限がある場合、Windows Installer は、マシンごとのインストールを試みます。それ以外の場合、Windows Installer はユーザーごとのインストールを実行します。
ARPAUTHORIZEDCDFPREFIX	このプロパティは、アプリケーションの更新チャネルの URL を格納します。
ARPCOMMENTS	このプロパティには、[プログラムの追加と削除] で表示されるこの製品に関するコメントが含まれます。

テーブル 6-4・追加の構成可能プロパティ（続き）

プロパティ名	説明
ARPCONTACT	このプロパティには、電子メール アドレスや電話番号などのサポート連絡先情報が含まれます。
ARPINSTALLLOCATION	このプロパティには、アプリケーションのプライマリ フォルダーへの完全修飾パスが格納されます。ARPINSTALLLOCATION は、"プロパティの設定" タイプのカスタム アクションを使って INSTALLDIR の値に設定することができます。
ARPNOREPAIR	このプロパティが 1 に設定されている場合、プログラム ウィザード に [修復] ボタンは表示されません。
ARPREADME	製品の Readme ファイルの完全修飾パスまたは URL を保持します。
ARPSIZE	このプロパティは、アプリケーションの予測サイズをキロバイト単位で格納します。
ARPSYSTEMCOMPONENT	このプロパティを 1 に設定すると、[プログラムの追加と削除] パネルにプログラムが表示されなくなります。
ARPURLINFOABOUT	このプロパティは、アプリケーションのホームページまたは作成者のホームページの URL を格納します。
ARPURLUPDATEINFO	このプロパティは、アプリケーションの更新情報の URL を格納します。
ARPNOMODIFY	このプロパティを設定すると、デフォルトで製品の修正を可能にする [プログラムの追加と削除] の機能が無効にされます。
ARPNOREMOVE	このプロパティを設定すると、製品の削除を可能にする [プログラムの追加と削除] の機能が無効にされます。
AVAILABLEFREEREG	このプロパティは、アプリケーションに必要な追加の空きレジストリ領域の大きさをキロバイト単位で設定可能にします。
CCP_DRIVE	このプロパティは、競争力のあるアップグレード行う特定の製品のインストールディスクのルートパスを保持します。
DISABLEADVTSHORTCUTS	このプロパティは、アドバタイズされたショートカットの作成を無効にします。
DISABLEMEDIA	この設定は、インストーラーがソース リストにメディア情報を追加できないようにします。
DISABLEROLLBACK	このプロパティを 1 に設定すると、インストーラーがインストールプロセス中に変更または削除されたファイルのコピーを保存するロールバック スクリプトを作成できないようになります。

テーブル 6-4・追加の構成可能プロパティ（続き）

プロパティ名	説明
EXECUTEACTION	このプロパティは、ExecuteAction アクションにより開始されたトップレベルのアクションを設定します。
EXECUTEMODE	このプロパティは、インストーラーの実行モードを設定します。[None] という値は、システムが変更されていないことを示します。デフォルト値である [Script] は、システムへの変更はすべてスクリプトを通じて実行されることを意味します。
INSTALLLEVEL	このプロパティは、機能の値に対応する値を維持します。インストールされる機能のレベルが INSTALLLEVEL プロパティと同じであるかそれより小さい場合、機能がインストールされます。これは、[標準] または [カスタム] などの異なるセットアップの種類に使用されます。
LOGACTION	ログされるアクション名のリストはセミコロンで区切ります。
MSIINSTALLPERUSER	このプロパティは、Windows Installer によって、パッケージが現在のユーザーに対してのみインストールされることを示します。 ・ ALLUSERS プロパティが 2 に設定されていて、MSIINSTALLPERUSER に空の文字列 (""") が設定されている場合、Windows Installer は、マシンごとにインストールを実行します。 ・ ALLUSERS プロパティが 1 に設定されていて、MSIINSTALLPERUSER に 1 が設定されている場合、Windows Installer は、ユーザーごとにインストールを実行します。 このプロパティは、Windows Installer 5 および Windows 7、または Windows Server 2008 R2 で使用できます。以前のバージョンの Windows Installer と Windows は、このプロパティを無視します。 詳細は、次を参照してください: ・ ユーザーごとのインストールとマシンごとのインストールの違い ・ MSIINSTALLPERUSER プロパティ (MSDN Web サイト)
Privileged	このプロパティは、ユーザーが管理者である場合、またはアプリケーションが管理者が割り当てたアプリケーションである場合に、程度の高い権限でインストールを実行します。
PROMPTROLLBACKCOST	このプロパティは、ディスク容量不足のためにインストールが続行できなくなった場合の処理を指定します。ユーザー インターフェイス レベルに応じて、ユーザーからの入力なしで自動的にロールバックを行うか、あるいはロールバックを無効にして続行することをユーザーにたずねるか指定できます。

テーブル 6-4・追加の構成可能プロパティ（続き）

プロパティ名	説明
PRIMARYFOLDER	このプロパティで指定したフォルダーが、インストールの「プライマリ」フォルダーになります。このフォルダーに対するパスは、PrimaryVolumePath プロパティ、PrimaryVolumeSpaceAvailable プロパティ、PrimaryVolumeSpaceRequired プロパティ、および PrimaryVolumeSpaceRemaining プロパティの値を決定するために使用されます。
REBOOT	このプロパティは、インストールが完了した後で再起動を強制または抑止することができます。使用できる値は次のとおりです。 ・ F — インストールが完了したときに再起動を強制します。 ・ S — ForceReboot アクションによって発生する再起動を除くすべての再起動を抑制します。 ・ R — Windows Installer アクションによって発生するすべての再起動を抑制します。
ROOTDRIVE	このプロパティは、管理モードでは、最初に見つかった書き込み可能なネットワーク ドライブにデフォルトのドライブを設定します。その他のモードでは、使用可能な最大量のディスク容量を持つ書き込み可能ローカル ドライブに、デフォルトのドライブを設定します。
SCRIPTFILE	このプロパティは、インストール中に実行されるすべての操作を含むスクリプト ファイルの場所を定義します。
SEQUENCE	このプロパティは、テーブルでアクションが実行される順序を示す .msi データベース テーブルを指定します。
SHORTFILENAMES	管理者モードで、このプロパティは、短いファイル名のみが使用されるように設定できます。
LIMITUI	このプロパティを設定すると、ユーザー インターフェイス レベルが基本に制限されます。カスタム ユーザー インターフェイスをインストーラーに組み込まれた UI と対話させない場合に便利です。
DefaultUIFont	このプロパティは、デフォルトのフォントを指定するために、TextStyle テーブルで見つかったあらかじめ定義されたスタイルの1つに設定します。このプロパティが設定されない場合、インストーラーはシステム フォントを使用します。そのため、フォーマットが壊れることがあります。

ユーザーが提供する情報

以下のセクションでは、エンドユーザーが入力する情報について説明します。エンド ユーザーが入力する情報には、ユーザー名、会社名、または言語などが含まれます。

テーブル 6-5・ユーザーが提供する情報

プロパティ名	説明
AdminProperties	AdminProperties は、管理インストール中にプロパティ セットのリストを保持します。これらのプロパティは外部 (ユーザー名) または内部 (このページの他のプロパティ) のいずれかになります。
COMPANYNAME	このプロパティは、インストールを実行するエンドユーザーの組織名を格納します。この情報は、[ユーザー情報] ダイアログから取り出されます。
ISX_SERIALNUM	このプロパティは、エンド ユーザーが [カスタマー情報] ダイアログの "シリアル番号" フィールドに入力したシリアル番号を格納します。
UserLanguageID	このプロパティは、エンドユーザーのデフォルトの言語 ID を保持します。
USERNAME	このプロパティは、インストールを実行中のエンドユーザーの名前を格納します。エンドユーザーの名前は [ユーザー情報] ダイアログから取得されます。
ProductLanguage	このプロパティは、数値で表されたこの製品の言語 ID を格納します。

製品固有のプロパティ

Property テーブルで設定できる製品固有のプロパティに関する情報は以下のとおりです。これらの種類のプロパティの例としては、テクニカル サポートの電話番号、製品名、シリアル番号があります。

テーブル 6-6・製品固有のプロパティ

プロパティ名	説明
ARPHELPLINK	このプロパティは、テクニカル サポートのインターネット アドレスを保持します。 この値は、[プログラムの追加と削除] の "サポート URL" プロパティに設定されます。セットアップをグローバル化するためには、文字列テーブル エントリを指定する必要があります。
ARPHELPTELEPHONE	このプロパティは、テクニカル サポートの電話番号を保持します。 この値は、[プログラムの追加と削除] の "サポート 電話番号" プロパティに設定されます。セットアップをグローバル化するためには、文字列テーブル エントリを指定する必要があります。

テーブル 6-6・製品固有のプロパティ（続き）

プロパティ名	説明
ProductCode	ProductCode は製品の特定のバージョンの GUID です。言語バージョンやリリース バージョンが異なれば、この ID も異なります。このプロパティは、[**一般情報**] ビューで設定されます。
ProductName	このプロパティは、製品名を格納します（例、InstallShield）。このプロパティは、[**一般情報**] ビューで設定されます。
ProductState	インストーラーは製品のインストールされた状態にこのプロパティを設定します。このプロパティは、以下の 4 つの値のいずれか 1 つを保持することができます。 -1 – 製品はインストールおよびアドバタイズされていません。 1 – 製品はアドバタイズされていますが、インストールされていません。 2 – 製品は別のユーザー用にインストールされています。 5 – 製品はインストールされており、現在のユーザーが使用できます。
ProductVersion	ProductVersion プロパティは、メジャー、マイナー、およびビルド バージョン番号を *AA.BB.CCCC* の形式で格納します。このプロパティは、[一般情報] ビューで設定されます。
Manufacturer	製品メーカーの名前を格納します。 この値は、[プログラムの追加と削除] の "発行者" プロパティに設定されます。セットアップをグローバル化するためには、文字列テーブル エントリを指定する必要があります。
DiskPrompt	このプロパティは、ディスクを入力するように指示するメッセージ ボックスに表示される文字列を保持します。"Disk 1" のように、ディスクのラベルに印刷されている追加情報に空のテキストを含めることも必要です。
DiskSerial	DiskSerial プロパティはこのリリースの内部シリアル番号に設定する必要があります。
ComponentDownload	このプロパティは、文字列識別子（GUID）により製品をダウンロードする URL を保持します。
LeftUnit	このプロパティは、数字の左側に単位を表示します。この構造が要求される言語に必要です。
UpgradeCode	これは、すでにインストールされている製品の関連セットを検索するために使用される GUID です。

テーブル 6-6・製品固有のプロパティ（続き）

プロパティ名	説明
IsAdminPackage	このプロパティは、現行のインストール パッケージが管理インストールを通じて作成されている場合、1 に設定されます。このプロパティは、事後管理インストールを検出するために使用できます。

インストーラーが設定するシステム フォルダー

以下のプロパティは、エンド ユーザーのシステムの多くのフォルダーへの完全修飾パスを保持します。

テーブル 6-7・システム フォルダーのプロパティ

プロパティ名	説明
AppDataFolder	このプロパティは、現在のユーザーのアプリケーション データ フォルダーへの完全パスを保持します。
CommonAppDataFolder	このプロパティは、All Users アプリケーション データ フォルダーへの完全修飾パスを保持します。
CommonFilesFolder	このプロパティの値は、32 ビット Common Files フォルダーへの完全修飾パスです。
DesktopFolder	このプロパティは、現在のユーザーの [デスクトップ] フォルダーへの完全パスを保持するために使用します。セットアップがすべてのユーザーに対して実行されているときに、ALLUSERS プロパティが設定されている場合、DesktopFolder プロパティにはすべてのユーザーのデスクトップ フォルダーへの完全パスが保持されます。
FavoritesFolder	FavoritesFolder プロパティは現在のユーザーの Favorites フォルダーへの完全パスを保持します。
FontsFolder	このプロパティは、Fonts フォルダーへの完全パスを保持します。
PersonalFolder	このプロパティは、現在のユーザーの個人フォルダーへの完全パスを保持します。
ProgramFilesFolder	このプロパティは、現在のユーザーの Program Files フォルダーへの完全パスを保持します。
ProgramMenuFolder	このプロパティは、現在のユーザーの [**プログラム**] メニューへの完全パスを保持するために使用されます。セットアップがすべてのユーザーに対して実行されているときに、ALLUSERS プロパティが設定されている場合、ProgramMenuFolder プロパティにはすべてのユーザーの [プログラム] メニューへの完全パスが保持されます。
SendToFolder	このプロパティは、現在のユーザーの SendTo フォルダーへの完全パスを保持します。

テーブル 6-7・システム フォルダーのプロパティ (続き)

プロパティ名	説明
StartMenuFolder	このプロパティは、現在のユーザーの [スタート] メニュー フォルダーまでの完全パスを保持するために使用されます。セットアップがすべてのユーザーに対して実行されているときに、ALLUSERS プロパティが設定されている場合、StartMenuFolder プロパティにはすべてのユーザーの [スタート] メニュー フォルダーへの完全パスが保持されます。
StartupFolder	このプロパティは、現在のユーザーの [スタートアップ] フォルダーへの完全修飾パスを保持するために使用されます。セットアップが All Users に対して実行されているときに、ALLUSERS プロパティが設定されている場合、StartupFolderプロパティにはすべてのユーザーの [スタート] メニューへの完全パスが保持されます。
SystemFolder	このプロパティは、32-bit System フォルダーまでの完全パスを保持します。
TempFolder	このプロパティは、Temp フォルダーまでの完全パスを保持します。
TemplateFolder	このプロパティは、現在のユーザーの Template フォルダーへの完全パスを保持します。
WindowsFolder	このプロパティは、ユーザーの Windows フォルダーへの完全パスを保持します。
WindowsVolume	このプロパティは Windows がインストールされているドライブに設定されます。

インストーラーにより設定されるオペレーティング システムのプロパティ

以下のプロパティは、インストーラーにより実行時に設定されます。ターゲット システム上の環境変数を参照します。

テーブル 6-8・オペレーティング システムのプロパティ

プロパティ名	説明
AdminUser	このプロパティは、ユーザーが管理者権限を持っている場合、インストール時にインストーラによって設定されます。
ComputerName	このプロパティは、インストールが実行されているコンピューターの名前を格納します。このプロパティは、インストーラーの初期設定時に Windows API の **GetComputerName** の呼び出しによって設定されます。

テーブル 6-8・オペレーティング システムのプロパティ (続き)

プロパティ名	説明
LogonUser	このプロパティは、インストールを実行するユーザーの名前を格納します。このプロパティは、Windows API の GetUserName の呼び出しによって設定されます。
OLEAdvtSupport	このプロパティは、ターゲット システムで COM を通したインストール オン デマンドをサポートしている場合、初期化中にインストーラーによって設定されます。
ServicePackLevel	オペレーティング システムのサービスパックがインストールされている場合、このプロパティは、そのアップデートの数値を格納します。
SharedWindows	このプロパティは、ターゲット システムで共有ウィンドウが使用されているときに設定されます。
ShellAdvtSupport	このプロパティは、ターゲット システムで機能のアドバタイズをサポートしている場合、初期化中にインストーラーによって設定されます。このプロパティは、Windows 98 または Windows 98 以降で、また Internet Explorer 4.01 がインストールされている場合はそれ以前のシステムでも自動的に設定されます。
SystemLanguageID	このプロパティは、ターゲット システムのデフォルトの言語 ID を保持します。この値は、初期化時に GetSystemDefaultLangID を呼び出すことによって、インストーラーが定義します。
TerminalServer	このプロパティは、ターゲット システムが Windows Terminal Server を装備したサーバーの場合、初期化時にインストーラーによって設定されます。
TTCSupport	このプロパティは、ターゲット システムが True Type のフォントコレクション (TTC) をサポートする場合、初期化時にインストーラーによって設定されます。JPN - 932、Taiwan - 950、China - 936、Korea - 949、Hong Kong - 950 システムは、TTC をサポートします。
Version9X	このプロパティは、Windows 95 と 98 オペレーティング システムのバージョン番号を整数として格納します。以下の数式が整数を判別するために使用されます: (MajorVersion * 100) + MinorVersionWindows 95 では Version9X は 400 に、Windows 98 では 410 に、Windows ME では 490 に設定されます。Version9X は Windows NT ベースのシステムでは設定されません。
VersionDatabase	このプロパティは、インストール中に使用されるデータベースのバージョン番号を格納します。

テーブル 6-8・オペレーティング システムのプロパティ（続き）

プロパティ名	説明
VersionNT	このプロパティは、Windows NT ベースのオペレーティング システムのバージョン番号を整数として格納します。以下の数式が整数を判別するために使用されます: (MajorVersion * 100) + MinorVersion特定のオペレーティング システムの VersionNT プロパティについては、Windows Installer ヘルプ ライブラリを参照してください。
WindowsBuild	このプロパティは、実行されているオペレーティング システムのビルド番号を格納します。
MsiNTProductType	このプロパティは、ターゲットマシンで実行されている NT オペレーティング システムの種類を格納します。このプロパティには、Windows Installer バージョン 2.0 が必要です。
MsiNTSuiteBackOffice	このプロパティは Microsoft BackOffice コンポーネントがインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuiteDataCenter	このプロパティは、Windows 2000 Datacenter Server がインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuiteEnterprise	このプロパティは、Windows 2000 Advanced Serverがインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuiteEnterprise	このプロパティは、Windows 2000 Advanced Serverがインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuiteSmallBusiness	このプロパティは Microsoft Small Business Server がインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuiteSmallBusinessRestricted	このプロパティは、制限クライアントライセンス付き Microsoft Small Business Server がインストールされている場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNTSuitePersonal	このプロパティは オペレーティング システムが Workstation Personal の場合に 1 に設定されます。他の場合、このプロパティは設定されていません。
MsiNetAssemblySupport	このプロパティは、オペレーティング システムで .NET Framework アセンブリをサポートするときに設定されます。他の場合、このプロパティは設定されていません。このプロパティには、Windows Installer バージョン 2.0 が必要です。

テーブル 6-8・オペレーティング システムのプロパティ（続き）

プロパティ名	説明
MsiWin32AssemblySupport	このプロパティは、オペレーティング システムで Win32 アセンブリをサポートするときに設定されます。他の場合、このプロパティは設定されていません。このプロパティには、Windows Installer バージョン 2.0 が必要です。

インストーラーにより設定されるハードウェア プロパティ

以下のプロパティは、実行時にインストーラーによって設定され、エンド ユーザーのシステムの特定のハードウェア プロファイルの設定を格納します。

テーブル 6-9・ハードウェアのプロパティ

プロパティ名	説明
Alpha	このプロパティは、プロセッサ レベルの数値を格納し、Alpha プロセッサでセットアップが実行される場合にのみ定義されます。（このプロパティは、Windows Installer バージョン 1.0 でのみサポートされています。）
BorderSide	このプロパティは、ウィンドウの横の境界線の幅をピクセル単位で指定します。
BorderTop	このプロパティは、ウィンドウの上の境界線の幅をピクセル単位で指定します。
CaptionHeight	このプロパティは、キャプション領域の高さをピクセル単位で指定します。
ColorBits	このプロパティは、各ピクセルの隣接カラービットの数（つまり、ユーザーのモニターのカラー深度）を格納します。たとえば、ユーザーのモニターが 256 色を使用している場合、ColorBits は 8 に設定されます。
Intel	このプロパティは、プロセッサ レベルの数値を格納し、Intel 32-bit プロセッサでセットアップが実行される場合にのみ定義されます。
PhysicalMemory	このプロパティは、インストールされている物理メモリの大きさをメガバイト単位で格納します。
ScreenX	このプロパティは、画面の幅をピクセル単位で定義します。
ScreenY	このプロパティは、画面の高さをピクセル単位で定義します。
TextHeight	このプロパティは、テキスト文字の高さを設定します。

テーブル 6-9・ハードウェアのプロパティ（続き）

プロパティ名	説明
VirtualMemory	このプロパティには、使用可能なページ ファイル領域の大きさがメガバイト単位で格納されます。

インストーラーにより更新されるステータス プロパティ

以下のプロパティは、インストーラーにより実行時に設定されます。これらのプロパティの値は、インストールの状態に関係ありません。

テーブル 6-10・ステータスのプロパティ

プロパティ名	説明
AFTERREBOOT	このプロパティは、ForceReboot アクションにより再起動された直後にインストーラーによって 1 に設定されます。
CostingComplete	このプロパティは、コスティングが始まるとすぐに 1 に設定され、コスティングが完了すると 0 に設定されます。
RollbackDisabled	インストーラーは、RollbackDisabled プロパティをロールバックが無効になるたびに設定します。デフォルトでは、このプロパティは設定されていません。
インストール済み	このプロパティは、製品がすでにインストールされているかどうかを判別します。
OutOfDiskSpace	このプロパティは、インストールのターゲットになるドライブに十分なディスク容量がない場合に True に設定されます。それ以外の場合は False に設定されます。
OutOfNoRbDiskSpace	このプロパティは、インストール時のターゲット ディスクに十分な空きディスク容量がない場合、およびロールバック機能がオフになっている場合に True に設定されます。すべてのターゲット ディスクに十分な空き容量がある場合は False に設定されます。
Preselected	このプロパティは、機能が事前に選択されているかどうかを判別し、選択されていない場合は選択ダイアログを表示しません。
PrimaryVolumePath	インストーラーは、このプロパティに PRIMARYFOLDER プロパティで指定されたパスを設定します。
PrimaryVolumeSpaceAvailable	インストーラーは、このプロパティに PRIMARYFOLDER プロパティで指定されたボリューム上で使用可能なバイト総数（512 バイト単位）を表す文字列を設定します。
PrimaryVolumeSpaceRequired	このプロパティは、現在選択している機能に必要なディスク容量の合計をバイト数で表した文字列（512 バイト単位で表示）を格納します。

テーブル 6-10・ステータスのプロパティ（続き）

プロパティ名	説明
PrimaryVolumeSpaceRemaining	インストーラーは、選択されている機能がすべてインストールされた場合に、システムで使用可能な残りのバイト数を 512 バイト単位で表した文字列に、このプロパティを設定します。
Resume	このプロパティは、保留されたセットアップからインストールを続行するときにユーザーに表示されるテキスト文字列を格納します。
UpdateStarted	このプロパティは、システムへの変更がインストール処理の結果として行われる場合に設定されます。
ReplacedInUseFiles	このプロパティは、現在使用中のファイルが上書きされる場合に設定されます。再起動が必要かどうかをチェックするためのカスタム アクションがこのプロパティを使用します。
NOUSERNAME	このプロパティを 1 に設定すると、インストーラーは USERNAME プロパティを設定しません。デフォルトでは、このプロパティは設定されず、USERNAME プロパティはレジストリから設定されます。
NOCOMPANYNAME	このプロパティを 1 に設定すると、インストーラーは COMPANYNAME プロパティを設定しません。デフォルトでは、このプロパティは設定されず、COMPANYNAME プロパティはレジストリから設定されます。

日付と時刻のプロパティ

テーブル 6-11・日付と時刻のプロパティ

プロパティ名	説明
Date	このプロパティは、現在の日を保持します。
Time	このプロパティは、現在の時間を保持します。

パブリック プロパティが制限付きパブリック プロパティである必要があることを指定する

制限付きパブリック プロパティを利用すると、ネットワーク管理者は、システム管理者または昇格された権限を持つ個人によってのみ変更が可能なパブリック プロパティを定義することができます。これにより管理者は、ネットワークで承認されていないユーザーがインストールを改ざんする可能性があることを心配することなく、即座に設定を変更できます。

Windows Installer は、いくつかのパブリック プロパティを制限付きパブリック プロパティとして見なします。制限付きパブリック プロパティの全一覧については、Windows Installer ヘルプ ライブラリの「Restricted Public Properties」を参照してください。

追加のパブリック プロパティを含める場合、それらを SecureCustomProperties プロパティに追加する必要があります。

以下のようなタスクを InstallShield ユーザー インターフェイスから実行した場合、InstallShield は自動的に該当するプロパティを SecureCustomProperties プロパティに追加します:

- [要件] ビューでプロジェクトに追加した起動条件でパブリック プロパティを使用すると、そのパブリック プロパティは SecureCustomProperties プロパティに追加されます。
- プロパティを使って設定したコントロールを含むダイアログの場合、このプロパティは SecureCustomProperties プロパティに追加されます。

これにより、カスタム パブリック プロパティを [ユーザー インターフェイス] シーケンスで設定してから、[実行] シーケンスに渡すことができるようになります。

実行時のシリアル番号検証を実装する

インストールの一部として、エンドユーザーが有効なシリアル番号を入力しなくてはならないようにできます。これは、カスタム アクション .dll ファイル と [カスタマー情報] ランタイム ダイアログを使用して設定できます。

タスク ***製品のシリアル番号検証を実装するには、以下の手順に従います:***

1. ビュー リストの **[セットアップ外観のカスタマイズ]** の下にある **[ダイアログ]** をクリックします。
2. **[ダイアログ]** エクスプローラーで、**[カスタマー情報]** をクリックします。
3. **"シリアル番号を表示"** フィールドの値を **[はい]** に変更します。
4. **"シリアル番号のテンプレート"** フィールドで、製品のシリアル番号に使用するフォーマットを指定します。たとえば、次のように入力します。

   ??????-####-??????????

   以下は、このフィールドについてのガイドラインです:

   - 疑問符 (?) を英数文字にあたる部分に入力します。
   - 数記号 (#) を数字にあたる部分に入力します。
   - ダッシュ (-) を使って、文字列をグループに分けします。ダッシュ (-) は、1 つの文字グループがそこで終わり、次のグループがそこから始まるということを示すシリアル番号内のブレーク ポイントを意味します。
5. **"シリアル番号の検証 DLL"** フィールドに、シリアル番号検証に使用する DLL の場所を入力します。または、[参照 (...)] ボタンをクリックして、この .dll ファイルを指定します。

   ヒント: *シリアル番号の検証 .dll ファイルのサンプル コードは、InstallShield Program Files フォルダー ¥Samples¥WindowsInstaller¥ValidateSerialNumber ディレクトリにあります。詳細については、「シリアル番号を検証するカスタム アクションを使用する」を参照してください。*

6. **"検証関数"** フィールドで、検証を実行する .dll ファイル関数の名前を入力します。
7. **"正常時の戻り値"** フィールドに 0 以外の数値を設定します。
8. **"再試行回数制限"** 設定で、エンドユーザーがシリアル番号を再入力することができる回数を指定します。
9. **"All Users オプションを表示"** フィールドを適切に構成します。

エンドユーザーが [ユーザー情報] ダイアログでシリアル番号を入力すると、.dll ファイルに渡されて検証が行われます。シリアル番号が正常に検証されると、インストールが続行されます。シリアル番号が無効な場合、"再試行回数制限" フィールドで許可する回数に従って、インストールは終了します。

メモ: *InstallShield は、シリアル番号 .dll ファイル カスタム アクションをユーザー定義のカスタム アクションの後にシーケンスします。つまり、[ユーザー情報] ダイアログの後に独自のカスタム アクションを追加すると、インストールはこのカスタム アクションを、シリアル番号 .dll ファイル カスタム アクションの前に起動します。*

インストール中とインストール後のシリアル番号へのアクセス

エンド ユーザーのシリアル番号を格納する Windows Installer プロパティは、ISX_SERIALNUM です。[カスタマー情報] ダイアログの "シリアル番号の表示" 設定で [はい] を選択すると、このダイアログに "シリアル番号" フィールドが含まれて、Windows Installer がプロパティを実行時にエンド ユーザーによって入力されたシリアル番号に設定します。

インストール中

インストール中、この VBScript サンプルにあるようなコードを使用して ISX_SERIALNUM の値にアクセスできます。

```
' シリアル番号の値を取得
Dim sSerialNo
sSerialNo= Session.Property("ISX_SERIALNUM")

' 表示する。
MsgBox sSerialNo
```

重要: *インストール後に ISX_SERIALNUM の値にアクセスできるようにするには、インストールが、ターゲット システムのどこかに ISX_SERIALNUM 値を格納する必要があります。たとえば、[レジストリ] ビューで、**[ISX_SERIALNUM]** を使う値を追加するレジストリ キーを作成します。*

インストール後

インストール完了後、インストールによって ISX_SERIALNUM プロパティの値がターゲット システムに書き込まれている場合、シリアル番号にアクセスすることが可能です。たとえば、インストールがその値をレジストリにか書き込んだ場合、そのシリアル番号のレジストリをチェックするコードを作成できます。

1

InstallShield プロジェクトを DevOps でビルドする

InstallShield では、DevOps における InstallShield プロジェクトのビルドがサポートされています。InstallShield プロジェクトをビルドするための、次の機能が提供されています:

- Azure DevOps ビルド拡張
- InstallShield StandAlone Build と Docker

Azure DevOps ビルド拡張

InstallShield では、Azure DevOps ビルド拡張がサポートされています。この拡張を使って、Azure DevOps Pipelines で InstallShield プロジェクトをビルドすることができます。

タスク ***Azure DevOps Pipelines で InstallShield プロジェクトのインストールをビルドするには、次の手順に従います。***

1. InstallShield ビルド拡張 を Azure DevOps 組織に追加およびインストールします。
   Install InstallShield SAB タスク および InstallShield Build タスク は Build Pipeline タスクで使用できます。
2. タスクを構成します。
3. リポジトリに格納されている InstallShield プロジェクトをビルドします。

Install InstallShield SAB タスク

Install InstallShield SAB タスクは、タスクで構成されている InstallShield バージョンに基づいて InstallShield Standalone Build セットアップを Microsoft がホストするマシン上にインストールして、タスクで提供されているライセンス サーバーの詳細を使ってライセンス サーバーを構成します。

InstallShield SAB のインストール タスクを構成するには、「Install InstallShield SAB Task」サポート技術情報の記事 (英語) を参照してください。

InstallShield Build タスク

InstallShield Build タスクは、InstallShield SAB がエージェント上にインストール済みかどうかをチェックし、インストールされていない場合は、InstallShield SAB をインストールして InstallShield プロジェクトのビルドをトリガします。

InstallShield ビルド タスクを構成するには、「InstallShield Build Task」サポート技術情報の記事 (英語) を参照してください。

InstallShield StandAlone Build と Docker

InstallShield プロジェクトは、Docker コンテナを使ってビルドすることもできます。Docker コンテナを使って InstallShield プロジェクトをビルドするには、Docker イメージをダウンロードしてから Docker イメージをビルドします。詳細については、InstallShield StandAlone Build with Docker」サポート技術情報の記事 (英語) を参照してください。

2

InstallShield と外部アプリケーションの統合

InstallShield の重要な側面のひとつは、Microsoft Source Control Interface に準拠するソース管理ソフトウェアとの共存および統合だけでなく、Visual Studio(TM) .NET の様な他のソフトウェア開発ツールとどのように共存、および統合するかという点です。

このセクションにあるヘルプ トピックで、InstallShield での外部アプリケーション サポートの範囲について各詳細を参照してください。

Microsoft Visual Studio との統合

InstallShieldでは、Microsoft Visual Studio でインストール プロジェクトを直接作成することができます。InstallShield では、Microsoft Visual Studio 内部からインストールの作成、変更、ビルドを実行することができます。

統合機能

InstallShield は Visual Studio シェル内部に完全統合されます。統合のユニークな機能には以下のようなものがあります。

- InstallShield のすべてのナビゲーションは Solution Explorer 内に表示されます。
- 各 InstallShield ビューは個別のウィンドウに表示されるのでスクロールする必要は無く、並列表示オプションも用意されています。
- InstallShield を Visual Studio の外部で実行することができます。
- 他の Visual Studio プロジェクトへのダイナミック リンクがあり、出力グループの新しいコンテンツがインストールに含まれます。

統合の利点

InstallShield の統合インストール オーサリング ソリューションを利用するその他の利点は、以下の通りです:

- Visual Studio ユーザー インターフェイスから離れることなく、使い慣れたナビゲーション、レイアウト オプションを使用しながら、インストールの作成やカスタマイズが可能です。

- インストールはソリューションが作成される度にソース ファイルが更新され、自動的に最新の状態が保たれます。
- インストールは、たとえばデバッグ、リリース、ビルド ディレクトリから自動的に含まれたソース ファイルといったソリューションの [ビルド構成] を反映します。
- .NET プロパティと依存関係は自動的にスキャンして、インストールに含めることができます。

メモ: *Visual Studio 内部から直接 InstallShield プロジェクトを作成、編集、およびビルドするには、Visual Studio 2005 以降が必要です。InstallShield を Visual Studio 2003 以前と統合することはできません。*

Visual Studio の統合は 1 度に InstallShield の 1 バージョンとのみ可能です。システムで最後にインストールまたは修復された InstallShield のバージョンが Visual Studio の統合に使用されます。

Microsoft Visual Studio で InstallShield プロジェクトを作成する

InstallShield は Microsoft Visual Studio と統合されています。ソリューションとして Visual Studio ワークスペース内から、InstallShield インストールを作成することができます。

タスク ***Microsoft Visual Studio 内で、InstallShield プロジェクトを作成するには、以下の手順を実行します。***

1. [**ファイル**] メニューで、[**新規**] をポイントして [**プロジェクト**] をクリックします。[**新規プロジェクト**] ダイアログ ボックスが開きます。
2. *Visual Studio 2010 の場合:* [**インストールされたテンプレート**] ボックスで [**InstallShield Express プロジェクト**] をクリックします。次に、適切なプロジェクトの種類を選択します。

   以前のバージョンの Visual Studio の場合: [**プロジェクトの種類**] ボックスで [**InstallShield Express プロジェクト**] をクリックします。次に、[**テンプレート**] ボックスで適切なプロジェクトの種類を選択します。
3. 必要に応じて名前とプロジェクトの場所の設定を構成します。
4. [**OK**] をクリックします。

Microsoft Visual Studio で InstallShield プロジェクトを開く

InstallShield は Microsoft Visual Studio と統合されています。ソリューションとして Visual Studio ワークスペース内から、InstallShield インストールを開くことができます。

タスク ***Microsoft Visual Studio 内で InstallShield プロジェクトを開くには、以下の手順を実行します。***

1. [**ファイル**] メニューで [**開く**] をポイントして [**プロジェクト**] をクリックします。[**プロジェクトを開く**] ダイアログ ボックスが開きます。
2. 目的の InstallShield ファイルを参照します。
3. [**開く**] をクリックします。

メモ: InstallShield Professional または InstallShield Express 3.x バージョン以前を使って作成したファイルは開けません。

Visual Studio ソリューションで VSSolutionFolder パス 変数を使用する

ハイレベルなベース ディレクトリを参照する、VSSolutionFolder と呼ばれる新しい定義済みパス変数をプロジェクトで使用できます。このサポートを使うと、InstallShield プロジェクトで Visual Studio ソリューション フォルダー内にある姉妹プロジェクトのファイルへのスタティック リンクを含めることができます。異なるマシン上のプロジェクトで作業を行う場合、VSSolutionFolder パス変数を使用するスタティック リンクは、姉妹プロジェクトのファイルへの正しいパスを参照することができます。

VSSolutionFolder パス変数は、InstallShield が Visual Studio ソリューション内で開かれたときに自動的に定義されます。また、MSBuild を使って InstallShieldプロジェクトを含むソリューションをビルドするときにも、自動的に定義されます。ただし、Visual Studio ソリューションなしで InstallShieldプロジェクトを開いた場合、VSSolutionFolder が自動的に定義されることはありません。たとえば、InstallShieldプロジェクトを、Visual Studio を開かずに InstallShieldインターフェイスで直接開いた場合、VSSolutionFolder は定義されません。同様に、コマンドライン ツール IsCmdBld.exe や、MSBuild で .isproj ファイルを使用する場合、VSSolutionFolder は定義されません。IsCmdBld.exe を使って InstallShieldプロジェクトのリリースをビルドするには、-L コマンドライン パラメーターを使って、VSSolutionFolder の値を設定します。このプロパティは、デフォルトのターゲット ファイルが使用されたとき、ItemGroup InstallShieldPathVariableOverrides として公開されます。

InstallShieldプロジェクトで VSSolutionFolder パス変数を含むパスを持つソース ファイルを含み、それを VSSolutionFolder パス変数がサポートされていない環境でビルドすると、次のようなビルド エラーが発生する可能性があります:

- -6103: ファイル <VSSolutionFolder>¥MyFile.exe が見つかりません
- -6271: ファイル <VSSolutionFolder>¥MyFile.exe が見つかりませんでした。このファイルの MsiFileHash テーブルをビルド中にエラーが発生しました。指定した場所にファイルが存在することを確認します。

Visual Studio ソリューションにリファレンスを追加する

[ファイル] ビューを使って、インストール プロジェクトに Visual Studio のリファレンスを追加します。

タスク ***リファレンスを追加するには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にある **[ソリューション エクスプローラー]** で、**[ファイル]** をダブルクリックします。
2. **[ソース コンピューターのフォルダー]** ペインにある Visual Studio Solution のアイテムには、現在のソリューションに含まれているすべてのプロジェクトのサブアイテムがあります。プロジェクトをクリックしてそのプロジェクトの出力グループを表示します。出力グループは **[ソース コンピューターのファイル]** ペインに表示されます。
3. インストール プロジェクトへ出力リファレンスを追加するには、**[インストール先コンピューターのフォルダー]** ペインのターゲット フォルダーへ出力をドラッグ アンド ドロップします。

InstallShield ツールバーまたはコマンドを Visual Studio ツールバーに追加する

InstallShield ツールバーおよび個別のツールバー コマンドボタンを Microsoft Visual Studio ワークスペースに追加することができます。

InstallShield ツールバーを追加する

タスク ***InstallShield ツールバーを追加するには、以下の手順を実行します。***

1. ツールバーの任意の場所を右クリックしてツールバーオプションを表示します。
2. ツールバーを選択して Visual Studio ワークスペースの上部に追加します。

InstallShield ツールバー コマンド ボタンを追加する

タスク ***個別のコマンドボタンをツールバーへ追加するには、以下の手順を実行します。***

1. ツールバーの任意の場所を右クリックしてツールバーオプションを表示します。
2. リストの下から [**カスタマイズ**] を選択します。[**カスタマイズ**] ダイアログ ボックスが開きます。
3. [**コマンド**] タブをクリックします。
4. [**カテゴリ**] リストで、カテゴリを選択して、特定のカテゴリ内で利用可能なコマンドを表示します。
5. [**コマンド**] リストで、コマンドボタンをクリックして、それをツールバーへドラッグします。

Microsoft Visual Studio でリリースをビルドする

Visual Studio 内での InstallShield プロジェクトのリリースのビルドは、InstallShield でのリリースのビルドと異なります。Visual Studio 内でリリースをビルドするとき、インストール プロジェクトを含むソリューション全体をビルドするオプションと、インストール プロジェクトのみをビルドするオプションがあります。

リリースをビルドする前に、リリースをソリューション構成と関連付ける必要があります。Visual Studio の構成マネージャーを使って、リリースをソリューション構成と関連付けます。

タスク ***Visual Studio 内部でリリースをビルドするには、以下の手順に従います:***

1. Visual Studio の構成マネージャーを使って、リリースを適切なソリューション構成にマップします。
   a. [**ビルド**] メニューで [**構成マネージャー**] をクリックします。

      別の方法として、[**標準**] ツールバーで [**ソリューション構成**] リストの中から [**構成マネージャー**] を選択します。

   b. [**アクティブ ソリューション構成**] リストから構成を選択します。
   c. [**プロジェクト コンテキスト**] ボックスで、プロジェクト構成を [**構成**] 列にある適切なリリースにマップします。

メモ: [ビルド] チェック ボックスの選択がクリアされているときにソリューションをビルドすると、選択解除されたプロジェクト構成はビルドされません。

d. **[閉じる]** ボタンをクリックします。

2. **[ビルド]** メニューで、該当するコマンドを選択します:

テーブル 2-1 ・ [ビルド] メニューのコマンド

コマンド	説明
ビルド ソリューション	構成マネージャーで、ソリューションに含まれている各プロジェクトに対して何がマップされているかに従って全ソリューションをビルドします。
ビルド *[インストール プロジェクト名]*	構成マネージャーの [構成] 列で指定されているリリースに従って、インストール プロジェクトのみをビルドします。このオプションは、[ソリューション エクスプローラー] でインストール プロジェクトが選択されているときに有効です。

ヒント: ソリューション全体をビルドするには、CTRL+SHIFT+B も使用できます。

.NET アセンブリをプロジェクトに追加する

InstallShield では、.NET アセンブリ ファイルを機能に追加することによって、インストール プロジェクトに .NET アセンブリを追加することができます。

タスク ***プロジェクトに .NET アセンブリを追加するには、以下の手順を実行します。***

1. **[アプリケーション データの指定]** の下にある **[ソリューション エクスプローラー]** で、**[ファイル]** をダブルクリックします。
2. 機能に .NET アセンブリを追加 します。
3. ファイルを右クリックしてから、**[プロパティ]** をクリックします。**[プロパティ]** ダイアログ ボックスが開きます。
4. **[COM と .NET の設定]** タブをクリックします。
5. **[ビルド時にスキャン]** リストから **[依存関係およびプロパティ]** を選択します。

ビルド時に、InstallShield が .NET アセンブリをスキャンして、必要に応じてその依存関係およびプロパティを追加します。

Web サービスまたは Web アプリケーションからプロジェクト出力を追加する

InstallShield には、拡張 Web サービスサポートが搭載されています。Web サービスまたは Web アプリケーションプロジェクトからプロジェクト出力を追加すると、プロジェクトを Web サービスとして追加するよう求めるメッセージが InstallShield によって表示されます。[いいえ] を選択すると、プロジェクト出力は通常通り追加されます。[はい] を選択すると、InstallShield は次の操作を行います。

1. IISROOTFORDER というインストール先フォルダーを作成。
2. 次の Visual Studio プロジェクト出力を IISROOTFOLDER に配布。

   [コンテンツ ファイル] は [IISROOTFOLDER]{VSIPProjectName} に移動

   [Primary Output] は [IISROOTFOLDER]{VSIPProjectName}¥bin″ に移動
3. ターゲットに [IISROOTFOLDER]{VSIPProjectName} を持つ IISVirtualDirectory を作成。

.NET Framework サポートをインストール プロジェクトへ追加する

アプリケーションのインストールでターゲット システムに .NET Framework サポートが必要とされる場合、このサポートをインストール プロジェクトに追加できます。詳細については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。

Microsoft Visual Studio Team Foundation Server との統合

Microsoft Visual Studio Team Foundation Server (TFS) は、製品のチーム開発においてタスクを共同作業で進めることができるツールとテクノロジのセットです。 は、Team Foundation Serve との統合をサポートします。InstallShield では、Team Foundation Server との統合をサポートします。統合による主要な機能は以下のとおりです:

- **ソース管理**—ソース管理エクスプローラーを使って、InstallShield プロジェクトを Team Foundation のバージョン管理と統合することで、InstallShield プロジェクトと Visual Studio ソリューションへの変更を管理できます。
- **自動ビルド**—Team Foundation ビルドを使って、InstallShield プロジェクトと Visual Studio ソリューションを定期的にコンパイル、テスト、およびデプロイできます。インストールはソリューションが作成される度にソース ファイルが更新され、自動的に最新の状態が保たれます。
- **プロジェクト管理**—InstallShield プロジェクトと Visual Studio ソリューションのバグ、タスク、およびプロジェクト ドキュメントなどの作業項目を管理します。プロジェクト ステータスは、チーム全体で Team System Web Access 内部、および Team Explorer 内部から使用できます。

統合の要件

InstallShield を Team Foundation Server と統合するには、InstallShield プロジェクトの作成、アップデート、またはビルドを行う各マシンに InstallShield をインストールします。つまり、InstallShield プロジェクトの作成およびアップデートを行う各マシンに InstallShield のインストールが必要です。Team Foundation Server に格納されている InstallShield プロジェクトのビルド エージェントとして指定されているマシン上にも をインストールする必要があります。InstallShield のライセンスの詳細については、InstallShield 使用許諾契約書 (EULA) を参照してください。

一部の種類のプロジェクトおよびソリューションをビルド エージェントがビルドするために、ビルド マシンに追加のソフトウェアをインストールすることが必要な場合もあります。たとえば、C++ プロジェクトをビルドする場合、C++ コンパイラやその他の依存関係も必要なため、ビルドマシンに Visual Studio をインストールする必要があります。

複数のビルド エージェントを使って Team Foundation Server プロジェクトをビルドする場合、特定のビルド タグを InstallShield が搭載されているマシン上の任意のエージェントに割り当てることができます。また、InstallShield プロジェクト用に作成された各ビルド定義に特殊なビルド タグを適用することもできます。こうすることで、InstallShield がインストールされているビルド マシンによってのみ InstallShield インストールのビルドに使用されます。ビルド タグの作成およびエージェント、ならびにビルド定義への割り当てについての詳細は、Visual Studio Team Foundation Server ドキュメントを参照してください。

64 ビットのビルド マシン上でビルドをキューに配置する場合、InstallShield.Tasks.dll ファイル (32 ビット ファイル) をロードするのに 32 ビット バージョンの MSBuild が使用されるように InstallShield プロジェクトのビルド定義を構成してしてください。そうしなければ、InstallShield.Tasks.dll ファイルをロードできなかったことを通知するビルド エラーが発生します。32 ビット バージョンの MSBuild を選択するには、Team Explorer でビルド定義の [プロセス] タブをクリックします。次に、[詳細] ノードの下にある "MSBuild プラットフォーム" 設定で [x86] を選択します。32 ビットのビルド マシンを使用している場合、"MSBuild プラットフォーム" 設定では [自動] または [x86] のどちらかを選択できます。

InstallShield と Visual Stuio が搭載された同じマシン上に Team Explorer をインストールした場合、Visual Studio で開かれている InstallShield プロジェクト内から Team Explorer を使用できます。これで、次のようなタスクを行うことができます:

- InstallShield プロジェクトでの作業中にソース管理エクスプローラーを使用。
- InstallShield プロジェクトと Visual Studio ソリューションのビルドを構成。
- 新しいビルドをキューに配置。

InstallShield プロジェクトを含むソリューションのビルドをキューに配置すると、ビルドされたインストールはドロップ フォルダー内の Install サブフォルダーにコピーされます。InstallShield ビルドが Team Foundation ビルドの下で実行されていることを検出すると、インストールはソリューションの最終出力の場所 (OutDir) にコピーされます。つまり、これはバイナリ ディレクトリで、Team Foundation ビルド プロセスの終わりに、ドロップ フォルダーにコピーされます。

InstallShield プロジェクトを Team Explorer に追加する

InstallShield と Visual Studio が搭載されている同じマシン上に Team Explorer をインストールした場合、Team Explorer を使って、InstallShield プロジェクト (.ise および .isproj ファイル) を Team Foundation Server に追加できます。

Visual Studio 内から InstallShield プロジェクトを作成したときに、プロジェクトをソース管理エクスプローラーに追加するには、[新規プロジェクト] ダイアログ ボックスで **[ソース管理に追加]** チェック ボックスを選択します。既存の InstallShield プロジェクトをソース管理エクスプローラーに追加するには、その他のファイルを追加するのと同じ要領でプロジェクトを追加します。

3

リファレンス

InstallShield のリファレンス情報は次のセクションに分類されます。

テーブル 3-1 ・リファレンス セクション

セクション	説明
メニュー、ツールバー、および ウィンドウのリファレンス	メニュー、ツールバー、およびウィンドウを含む InstallShield ユーザー インターフェイスの様々なコンポーネントについて説明します。
ダイアログ ボックス リファレンス	InstallShield で表示される各ダイアログ ボックスについてのリファレンス情報を含みます。
ウィザード リファレンス	InstallShield で利用できる各ウィザードの詳しい情報を提供します。
ビュー リファレンス	InstallShield で表示される各ビューについて説明します。
エラーと警告	インストールの作成、ビルド、および実行する際に起こりうるエラー コードおよび警告についての情報を提供します。このセクションには、プロジェクトを InstallShield 製品の以前のバージョンから最新バージョンに移行するときに起こりうるエラーと警告についてのリファレンス情報も含まれています。
InstallShield カスタム アクション リファレンス	InstallShield で使用できるカスタム アクションそれぞれについて説明します。
コマンドライン ツール	リリースのビルドおよびインストールの実行などのタスクを実行するためにコマンドラインから使用できるツールを紹介します。
エンドユーザー ダイアログ	InstallShield の [ダイアログ] ビューで提供されているすべてのエンドユーザー ダイアログのリファレンスとして使用できます。

メニュー、ツールバー、および ウィンドウのリファレンス

このセクションは、メニュー、ツールバー、およびウィンドウを含む InstallShield ユーザー インターフェイスの様々なコンポーネントについて説明します。

メニュー

InstallShield のメニューは、InstallShield ユーザー インターフェイスの上部のメニューバーにあります。メニューには、それぞれ、コマンドのリストがあります。一部のコマンドはその隣にアイコンを持っており、コマンドとアイコンを手早く関連付けることができます。

InstallShield の各メニューは、このセクションで説明されています。

- ファイル
- 編集
- 表示
- 移動
- プロジェクト
- ビルド
- ツール
- ヘルプ

[ファイル] メニュー

次の表は、[ファイル] メニューのコマンド一覧と関連キーボード ショートカットおよびアイコンの一覧です。

テーブル 3-1 ・ファイル メニュー コマンド

コマンド	ショートカット	アイコン	説明
新規作成	Ctrl + N		新規インストール プロジェクトを作成します。
開く	Ctrl + O		既存のインストール プロジェクトを開きます。
閉じる			現在のプロジェクトを閉じます。
保存	Ctrl + S		現在のプロジェクトを保存します。

テーブル 3-1 • ファイル メニュー コマンド (続き)

コマンド	ショートカット	アイコン	説明
名前を付けて保存			現在のプロジェクトファイルに新しい名前を付けて、新しい場所に保存することができます。
1、2、3、4			最後にアクセスしたプロジェクトの 1 つを開きます。
終了			現在のプロジェクトを閉じて InstallShield を終了します。

[編集] メニュー

次のテーブルは、[編集] メニューのコマンド一覧と関連キーボード ショートカットの一覧です。

テーブル 3-2 • [編集] メニュー コマンド

コマンド	ショートカット	説明
Undo	Ctrl + Z	最後に実行されたアクションを元に戻します。
Cut	Ctrl + X	選択中のテキストをクリップボードにカットします。
Copy	Ctrl + C	選択中のテキストをクリップボードにコピーします。
Paste	Ctrl + V	クリップボードの内容を貼り付けます。
Find	Ctrl + F	[ファイル] ビューでインストール プロジェクトに追加したフォルダーおよびファイルを検索することができる、[検索] ダイアログ ボックスを開きます。詳細については、「プロジェクトでファイルとフォルダーを検索する」を参照してください。

[表示] メニュー

次の表は、[表示] メニューのコマンド一覧と関連キーボードショートカットおよびアイコンの一覧です。

テーブル 3-3 • [表示] メニュー コマンド

コマンド	ショートカット	アイコン	説明
出力ウィンドウ			出力ウィンドウの表示を切り替えます。
ビュー リスト			[ビュー リスト] を切り替えます。
ビュー バー	F4		ビュー バーを切り替えます。
ヘッダー バー			ヘッダーバーを切り替えます。

テーブル 3-3 • [表示] メニュー コマンド (続き)

コマンド	ショートカット	アイコン	説明
ツールバー			ツールバーを切り替えます。
ステータス バー			ステータス バーを切り替えます。
プロジェクト アシスタント			プロジェクト アシスタントを表示します。

[移動] メニュー

次の表は、[移動] メニューのコマンド一覧と関連キーボードショートカットおよびアイコンの一覧です。一部のビューに関するコマンドは、InstallShield で開いたプロジェクトによっては、[移動] メニューからは利用できません。

テーブル 3-4 • [移動] メニュー コマンド

コマンド	ショートカット	アイコン	説明
前のビュー	Alt + 上向き矢印		ビュー リストに表示されている現在のビューの 1 つ上のビューに移動します。
次のビュー	Alt + 下向き矢印		ビュー リストで表示されている現在のビューの 1 つ下のビューに移動します。
Back	Alt + 左向き矢印		ビューを選択する履歴から最後に訪れたビューへ移行します。ビュー履歴に複数のエントリがある場合、これは何回でも使用することが可能です。
進む	Alt + 右向き矢印		ビューを選択する履歴から最後に訪れたビューへ移行します。最初に [戻る] をクリックしたビューに到達するまで続けられます。
スタート ページ			スタートページに移動します。
ヘルプ			[ヘルプ] ビューに移動します。
プロジェクト アシスタント			プロジェクト アシスタントを表示します。
セットアップの編成			[一般情報]、[機能]、[セットアップの種類]、[アップデート通知] または [アップグレード パス] ビューを表示することができます。

テーブル 3-4 • [移動] メニュー コマンド (続き)

コマンド	ショートカット	アイコン	説明
アプリケーション データの指定			[ファイル] ビュー、[ファイルと機能] ビュー、[再配布可能ファイル] ビュー、または [依存関係] ビューを表示することができます。
ターゲット システムの構成			[ショートカット/フォルダー]、[レジストリ]、[ODBC リソース]、[INI ファイルの変更]、[ファイル拡張子]、[環境変数]、[IIS 構成] または [コンポーネント サービス] ビューを表示することができます。
セットアップ外観のカスタマイズ			[ダイアログ]、[ビルボード] または [テキストとメッセージ] ビューを表示できます。
セットアップ要件およびアクションの定義			[必要条件]、[カスタム アクション] または [セットアップ ファイル] ビューを表示できます。
リリースの準備			[リリース] ビューを表示することができます。

[プロジェクト] メニュー

次のテーブルは [プロジェクト] メニュー コマンドの一覧です。

テーブル 3-5 • [プロジェクト] メニュー コマンド

コマンド	説明
Visual Studio デプロイメント プロジェクト インポート ウィザード	Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートできるウィザードを起動します。
スタティック スキャンの実行	スタティック スキャン ウィザードを起動します。
ダイナミック スキャンの実行	ダイナミック スキャン ウィザードを起動します。
文字列エントリのインポート	文字列テーブルのインポート ウィザードを起動します
文字列エントリのエクスポート	文字列テーブルをエクスポート ウィザードを起動します。
設定	Msi ログファイル設定ダイアログ ボックスを起動します。

[ビルド] メニュー

次の表は、[ビルド] メニューのコマンド一覧と関連キーボードショートカットおよびアイコンの一覧です。

テーブル 3-6 · [ビルド] メニューのコマンド

コマンド	ショートカット	アイコン	説明
ビルド	F7		リリースをデフォルトの設定でビルドするか、既にリリースをビルド済みの場合は、最後に保存した設定でリリースを再ビルドします。
クイック ビルド	Shift + F7		インストールの .msi ファイル部分のみを再ビルドし、完全再ビルドに比べて変更点を素早く確認することができます。
ビルドの停止	Ctrl + Break		現在のビルドプロセスをキャンセルします。
テスト	Ctrl + T		システムを変更することなくインストールのユーザー インターフェイス部分を実行することができます。すべてのカスタム アクションが実行されます。
実行	Ctrl + F5		IDE を終了せずに完成したインストールを実行することができます。
アンインストール			前回実行されたリリースをアンインストールします。
Web の配布ウィザード	Ctrl + W		Web の配布ウィザードが起動し、Web ページから起動するインストールを作成することができます。

[ツール] メニュー

次の表は、[ツール] メニューのコマンド一覧と関連キーボードショートカットおよびアイコンの一覧です。

テーブル 3-7 · [ツール] メニュー コマンド

コマンド	アイコン	説明
リリース フォルダーを開く		Windows Explorer を起動し、リリース フォルダーを開きます。
QuickPatch の作成		すべての既存プロジェクトを保存して閉じるようプロンプトしてから、QuickPath ウィザードを起動します。
アップデートの確認		サービスパックおよび InstallShield への他のアップデートを確認します。アップデーが可能な場合、ダウンロードおよびインストールを選択できます。

テーブル 3-7 • [ツール] メニュー コマンド (続き)

コマンド	アイコン	説明
再配布可能ファイル ダウンローダー		再配布可能ファイル ダウンローダー ウィザードを起動して、サードパーティ再配布可能ファイル、マージ モジュール、前提条件、その他のファイルをローカルコンピューターに素早くダウンロードすることができます。

[ヘルプ] メニュー

次の表は、[ヘルプ] メニューのコマンド一覧と関連キーボードショートカットおよびアイコンの一覧です。

テーブル 3-8 • [ヘルプ] メニュー コマンド

コマンド	アイコン	説明
目次		InstallShield ヘルプライブラリの [目次] タブを表示します。
索引		InstallShield ヘルプ ライブラリの [索引] タブを表示します。
Search		InstallShield ヘルプ ライブラリの [検索] タブを表示します。
サポート セントラル		Web のサポート セントラルを表示します。
InstallShield コミュニティ		Web の [コミュニティ] を表示します。
リリース ノート		InstallShield リリースノートを表示します。
Revenera Web サイト		Revenera の Web サイトに接続します。
[ヘルプ] ビュー		[ヘルプ] ビューを表示します。
InstallShield Express について		バージョン情報を表示して InstallShield を登録できる [バージョン情報] ダイアログ ボックスを表示します。

ツールバー

InstallShield ユーザー インターフェイスでは標準ツールバーを使用することができます。標準ツールバーを使用して、頻繁に使用するメニュー コマンドに素早くアクセスすることができます。必要に応じて、カスタム ツールバーを作成したり、標準ツールバーをカスタマイズしたりすることができます。すべてのツールバーはサイズや配置を変更できるほか、固定したり取り外したりすることができます。

標準ツールバー

次のテーブルは標準ツールバージョンにあるすべてのボタンの説明です。

テーブル 3-9 · 標準ツールバーのボタン

ボタン	名前	説明
	新しいプロジェクト	プロジェクトタイプを選択して新しいプロジェクトを始める [新規プロジェクト] ダイアログ ボックスを起動します。
	開く	[開く] ダイアログ ボックスを起動します。[開く] ダイアログ ボックスでは、既存のインストール プロジェクトを開くことができます。
	保存	現在のプロジェクトファイルを保存します。
	ビューバー	インターフェイスの左側に表示されるビューバーを表示、または非表示します。
	ビュー リスト	InstallShield IDE で使用できるすべてのビューを表示するビュー リストを表示または非表示にします。 *メモ· また、F4 を押して、ビューリストを表示または非表示にすることもできます。*
	前のビュー	ビュー リストで表示されている通り、現在のビューの 1 つ上のビューを表示します。
	次のビュー	ビュー リストで表示されている通り、現在のビューの 1 つ下のビューを表示します。
	Back	ビューを選択した履歴から、最後に訪れたビューを表示します。ビュー履歴に複数のエントリがある場合、このボタンは何回でもクリックすることが可能です。
	進む	ビューを選択した履歴から最後に訪れたビューを表示します。初めに [戻る] ボタンをクリックしたビューに画面が変わるまで、このボタンをクリックし続けることができます。
	ビルド	リリースをデフォルトの設定でビルドするか、既にリリースをビルド済みの場合は、最後に保存した設定でリリースを再ビルドします。
	ビルドの停止	現在のビルドプロセスをキャンセルします。

テーブル 3-9 ・ 標準ツールバーのボタン (続き)

ボタン	名前	説明
	実行	完成したインストール プロジェクトを実行します。 ***メモ・[インストール前にアンインストールする]*** *チェック ボックスを選択すると、InstallShield はインストールを再実行する前に製品をアンインストールします。このチェック ボックスは、[オプション] ダイアログ ボックスの [プリファレンス] タブから使用できます。*
	アンインストール	前回実行されたリリースをアンインストールします。
	テスト	システムに何の変更も加えることなく、インストール プロジェクトのユーザーインターフェイスの部分を実行できます。すべてのカスタム アクションが実行されます。
	Web の配布ウィザード	Web の配布ウィザードが起動し、Web ページから起動するインストールを作成することができます。
	リリース フォルダーを開く	Windows Explorer を起動し、現在のリリースの DISK1フォルダーで開きます。リリースがない、または、リリースばまだビルドされていない場合、Windows Explorer がデフォルトのプロジェクトの場所に開きます。
	[ヘルプ] ビュー	InstallShield について多くの質問の答えが見つかる IDE ヘルプビューを表示します。

出力ウィンドウ

[出力] ウィンドウは、プロジェクトをビルドすると InstallShield の画面下部に表示されます。また、プロジェクト変換中にプロジェクトに関する情報も表示します。出力ウィンドウには次のようなタブがあります。

テーブル 3-10 ・ 出力ウィンドウのタブ

タブ	説明
出力	配布出力情報を保存し、ビルド出力を表示します。これには、テキストファイルとして保存された出力ファイルへのリンクがアクティブになります。
タスク	プロジェクト ビルド時のエラーや警告メッセージの説明を表示します。エラーおよび警告コードは、ナレッジベース の各項目にリンクしています。

ダイアログ ボックス リファレンス

このセクションでは、InstallShield ユーザー インターフェイスで使用できる各ダイアログ ボックスについて説明します。

- [.NET 1.1/2.0 コア言語] ダイアログ ボックス
- [.NET 1.1/2.0 言語パック] ダイアログ ボックス
- [MIME の種類を追加] ダイアログ ボックス
- [アプリケーション拡張子マッピング] ダイアログ ボックス
- [アプリケーションのマッピング] ダイアログ ボックス
- [インストール先ファイルの参照] ダイアログ ボックス
- [ディレクトリの参照/INSTALLDIR の設定] ダイアログ
- [ファイルの参照] ダイアログ ボックス
- ショートカットのターゲットを参照するダイアログ ボックス
- [タイトル ターゲットの参照] ダイアログ ボックス
- [証明書の選択] ダイアログ ボックス
- [条件ビルダー] ダイアログ ボックス
- [コンテンツ ソース パス] ダイアログ ボックス
- [カスタム エラー] ダイアログ ボックス
- [依存関係] ダイアログ ボックス
- [ダイアログのイメージ] ダイアログ ボックス
- [セットアップのデジタル署名] ダイアログ ボックス
- [レジストリ データの編集] ダイアログ ボックス
- [エラー マッピングのプロパティ] ダイアログ ボックス
- [ファイルの詳細] ダイアログ ボックス
- [ファイルのプロパティ] ダイアログ ボックス
- ファイル 削除の [プロパティ] ダイアログ ボックス
- [フォルダーのプロパティ] ダイアログ ボックス
- [InstallShield 前提条件のプロパティ] ダイアログ ボックス
- [Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックス
- [マージ モジュールの構成可能な値] ダイアログ ボックス
- [マージ モジュールのプロパティ] ダイアログ ボックス
- [MIME の種類] ダイアログ ボックス
- [MSI 値] ダイアログ ボックス

- [複数行文字列値] ダイアログ ボックス
- [新規プロジェクト] ダイアログ ボックス
- [オプション] ダイアログ ボックス
- [出力] ダイアログ ボックス
- ファイルとディレクトリの [アクセス許可] ダイアログ ボックス
- レジストリ キーの [アクセス許可] ダイアログ ボックス
- [アイコンの選択] ダイアログ ボックス
- [INSTALLDIR の設定] ダイアログ ボックス/[DATABASEDIR の設定] ダイアログ ボックス
- [設定] ダイアログ ボックス
- [システム ハードウェア要件] ダイアログ ボックス
- [マージ モジュール検索パスのアップデート] ダイアログ ボックス
- [Express プロジェクト名のアップグレード] ダイアログ ボックス

[.NET 1.1/2.0 コア言語] ダイアログ ボックス

[.NET 1.1/2.0 コア言語] ダイアログ ボックスを使って、配布する .NET コア言語を選択します。これは、.NET 1.1 コア再配布可能ファイルのインストール中に使用される言語です。

".NET Framework バージョン" 設定でバージョン 2.0 を選択すると、このバージョンの再配布可能ファイルにはすべての言語が含まれているため、言語オプションはすべて選択されて無効となります。

使用中のシステムに特定の言語がインストールされていない場合、このダイアログ ボックス内でその言語のチェック ボックスは無効です。

ヒント: *.NET Framework 再配布可能ファイルの 1 つ以上の言語バージョンを使用中のシステムにダウンロードするには、[その他の言語をダウンロードする] ボタンをクリックします。再配布可能ファイル ダウンローダ ウィザードが起動して、1 つ以上の再配布可能ファイルを使用中のシステムにダウンロードできます。*

[.NET 1.1/2.0 言語パック] ダイアログ ボックス

[.NET 1.1/2.0 言語パック] ダイアログ ボックスを使って、ターゲット システムにインストールする .NET 言語パックに対応する言語を選択します。このダイアログ ボックスは、[リリース] ビューで、リリースの [.NET/J#] タブにある ".NET 1.1/2.0 言語パック" 設定で省略記号ボタン (...) をクリックすると開きます。

使用中のシステムに特定の言語パックがインストールされていない場合、このダイアログ ボックス内でその言語のチェック ボックスは無効です。

ヒント: *1 つ以上の言語パックを使用中のシステムにダウンロードするには、[その他の言語をダウンロードする] ボタンをクリックします。再配布可能ファイル ダウンローダ ウィザードが起動して、言語パックを使用中のシステムにダウンロードできます。*

[MIME の種類を追加] ダイアログ ボックス

[MIME の種類を追加] ダイアログ ボックスを使って、ファイル拡張子名とプログラム、又はそれらのファイルを処理するインタプリタ間のマッピングを追加又は変更します。このダイアログ ボックスは、[MIME の種類] ダイアログ ボックスの [追加] または [編集] ボタンをクリックすると開きます。

テーブル 3-1 • [MIME の種類を追加] ダイアログ ボックスの設定

設定	説明
ファイル名拡張子	ファイル名拡張子を入力します(例、**.abc**)。これはスタティック ファイル名拡張子です。 実行可能ファイルにワイルドカードのアプリケーション マッピングを使用するには、アスタリスク (*) を入力します。
MIME タイプ	MIME の種類を入力します (例、**application/octet-stream**)。

[アプリケーション拡張子マッピング] ダイアログ ボックス

[アプリケーション拡張子マッピング] ダイアログ ボックスを使って、ファイル拡張子名とプログラム、またはファイルを処理するインタプリタ間のマッピングを追加または変更します。このダイアログ ボックスは[アプリケーション マッピング] ダイアログ ボックスで [追加] をクリックした時に表示されます。

テーブル 3-2 • [アプリケーション拡張子マッピング] ダイアログ ボックスの設定

設定	説明
拡張子	アプリケーション (例、*.abc*) に関連付けられたファイル名の拡張子を入力します。 実行可能ファイルにワイルドカードのアプリケーション マッピングを使用するには、アスタリスク (*) を入力します。
実行可能ファイル	パスを入力するか、[参照] をクリックして [実行可能ファイルの選択] ダイアログ ボックスを立ち上げます。ここでマップするプロジェクトの実行可能ファイルを指定します。実行可能ファイルの名前 (.exe or .dll) を入力、又は [参照] ボタンを使ってファイルを検索します。実行可能ファイルは Web サーバーのローカル ハード ドライブに配置します。
動詞	[動詞] セクションには、どの HTTP 動詞をアプリケーションにパスすべきかを指定できます。 • **すべての動詞** – すべての動詞を含める場合、このオプションを選択します。すべてのリクエストをアプリケーションに渡します。 • **最大数** – このオプションを選択して、アプリケーションに渡す HTTP 動詞を指定することができます。動詞をカンマで分けます。

テーブル 3-2 • [アプリケーション拡張子マッピング] ダイアログ ボックスの設定 (続き)

設定	説明
スクリプト エンジン (IIS 6 以前のみ)	アプリケーションを実行許可なしにディレクトリ内で実行させたい時、このチェック ボックスを選択します。基本的にこの設定はインタープリタへマップされた ASP や IDC といったスクリプトベースのアプリケーション用のものです。 スクリプト マップされたアプリケーションを実行するためには、実行許可プロパティで "スクリプトのみ" 又は "スクリプトと実行可能ファイル" オプションを選択します。スクリプトマップされたアプリケーションのみを実行する場合は [スクリプトのみ] オプションを選択します。スクリプト マップされたアプリケーションと実行可能ファイル (.exe and .dll) を実行する場合は [スクリプトと実行可能ファイル] を選択します。 この設定は IIS 6 以前に適用します。IIS 7 は、この設定を無視します。
そのファイルの存在を確認 (IIS 6 以前のみ)	Web サーバーに対して、要求されたスクリプト ファイルが存在していることと、要求しているユーザーがそのスクリプト ファイルへのアクセス許可を持っていることを確認するよう指示するには、このチェック ボックスを選択します。 スクリプト ファイルが存在しない場合、もしくはエンドユーザーがアクセス権を持っていない場合はブラウザーに適切な警告メッセージが表示され、スクリプト エンジンは呼び出されません。このオプションは、たとえば Perl インタープリタのような、スクリプトがアクセス不能の場合に CGI レスポンスを送らない CGI 以外の実行可能ファイルにマップされたスクリプトに使うと便利です。 ***メモ:*** *スクリプトがサーバーとスクリプトエンジンによって 2 度開かれているので、このチェック ボックスが選択されているときにはパフォーマンスが低下します。* この設定は IIS 6 以前に適用します。IIS 7 は、この設定を無視します。

[アプリケーションのマッピング] ダイアログ ボックス

[アプリケーションのマッピング] ダイアログ ボックスを使って、ファイル拡張子名とそのファイルを処理するアプリケーションとの間のマッピングを編集または削除することができます。

[アプリケーションのマッピング] ダイアログ ボックスは、[IIS 構成] ビュー内から利用することができます。このダイアログ ボックスを開くには、エクスプローラーで Web サイト、アプリケーション、または仮想ディレクトリをクリックします。次に、"アプリケーションのマッピング" 設定で省略記号ボタン (...) をクリックします。

メモ: [動詞]列にアスタリスク(*)が表示されたときは、指定した拡張子にすべての動詞が使用されます。

テーブル 3-3 · [アプリケーションのマッピング] ダイアログ ボックスの設定

設定	説明
追加	ファイル名拡張子とプログラム、またはこれらのファイルを処理するインタープリタ間とのマッピングを追加するには、このボタンをクリックします。そうすると、[アプリケーション拡張子マッピング] ダイアログ ボックス が開きます。
編集	既存するアプリケーション マッピングを編集するには、マッピングを選択して、このボタンをクリックします。
削除	既存するアプリケーション マッピングを削除するには、拡張子を選択して、このボタンをクリックします。

[インストール先ファイルの参照] ダイアログ ボックス

セットアップにカスタム アクションを含め、"ソースの場所" の値として [製品と共にインストール] を選択すると、"ファイル名" プロパティを選択したときにこのダイアログが表示されます。カスタム アクションで使用するファイルが含まれる機能を選択し、ファイルを探して [開く] をクリックします。

[ディレクトリの参照/INSTALLDIR の設定] ダイアログ

このダイアログ ボックスを使用して、ディレクトリの参照、新規作成、名前の変更、または削除を行います。

ダイアログ ボックスの設定

インストール先ディレクトリ

このフィールドには、現在使用できるすべてのインストール先ディレクトリが一覧表示されます。このフィールドで、ディレクトリの選択、 作成 、 名前の変更 、または 削除 を行うことができます。

ディレクトリの選択

タスク ***ディレクトリを選択するには、次の操作を実行します。***

1. 選択するディレクトリをクリックします。
2. [OK] をクリックします。

ディレクトリの新規作成

タスク ***新しいディレクトリを作成するには、次の操作を実行します。***

1. ディレクトリまたは **[インストール先コンピューター]** を選択し、INSERT を押します。選択したフォルダーまたはインストール先コンピューターの下にディレクトリが作成されます。
2. ディレクトリ名を入力します。
3. 必要であれば ディレクトリ識別子 を入力します。

ディレクトリ名の変更

タスク ***ディレクトリの名前を変更するには、次の操作を実行します。***

1. ディレクトリまたは **[インストール先コンピューター]** を選択し、F2 を押します。
2. 新しいディレクトリ名を入力します。定義済みのディレクトリ名は変更できないことにご注意ください。
3. 必要に応じて、ディレクトリ識別子を変更して、ディレクトリの新しい名前と統一させます。

ディレクトリの削除

タスク ***ディレクトリを選択するには、次の操作を実行します。***

ディレクトリを選択して DELETE を押します。定義済みディレクトリは削除できないことにご注意ください。

ディレクトリを削除すると、選択したディレクトリの下にあるすべてのサブディレクトリも一緒に削除されます。

ディレクトリ識別子

[ディレクトリ識別子] ボックスを使用して、ディレクトリに分かりやすい名前を付けることができます。

メモ: *ディレクトリ識別子は、有効な MSI 識別子 である必要があります。機能の場合、ディレクトリ識別子はすべて大文字である必要があります。*

[ファイルの参照] ダイアログ ボックス

[ファイルを開く] ダイアログ ボックスは、[リリース] ビューで、リリースの [署名] タブにある "含めるパターンとファイル" 設定または "除外するパターンとファイル" 設定で省略記号ボタン (...) をクリックすると表示されます。[ファイルを参照] ダイアログ ボックスで、プロジェクトのスタティック ファイルに署名を行うかどう

かを指定できます。また、ワイルドカード文字としてアスタリスク (*) を使用することもできます。ワイルドカード文字の使用は、プロジェクトに動的にリンクされたファイル含め、特定のパターンに一致するすべてのファイルに署名を行う場合、特に便利です。

テーブル 3-4 ・ [ファイルの参照] ダイアログ ボックスの設定

設定	説明
署名するファイルを選択	このボックスは、"含めるパターンとファイル" 設定で省略記号ボタン (...) をクリックすると表示されます。 このダイアログ ボックスでは、**[ファイルの種類を表示する]** 一覧で選択したファイルの種類に一致するプロジェクトにあるすべての静的に含められたファイルが一覧表示されます。いくつかのデフォルト ファイル パターン (*.dll など) も一覧表示されます。 ビルド時に InstallShield で署名するプロジェクト内のファイルの種類に対応するファイルとファイル パターンのチェック ボックスを選択します。
署名をスキップするファイルを選択	このボックスは、"除外するパターンとファイル" 設定で省略記号ボタン (...) をクリックすると表示されます。 このダイアログ ボックスでは、**[ファイルの種類を表示する]** 一覧で選択したファイルの種類に一致するプロジェクトにあるすべての静的に含められたファイルが一覧表示されます。いくつかのデフォルト ファイル パターン (*.dll など) も一覧表示されます。 ビルド時に、あるファイルとファイル パターンが InstallShield によって署名されるのを避ける場合、該当するファイルとファイル パターンのチェック ボックスを選択します。
ファイルの種類を表示する	この一覧を使用して、**[署名するファイルを選択]** ボックスまたは **[署名をスキップするファイルを選択]** ボックスで表示されるファイルの種類をフィルターします。

Windows ロゴ・ *インストールのすべての実行可能ファイル (.exe、.dll、.ocx、.sys、.cpl、.drv、および .scr ファイル) は、Windows ロゴ プログラムに準拠するためにデジタル署名が必要です。*

このダイアログ ボックスで [OK] をクリックすると、"含めるパターンとファイル" 設定の下に新しい "含める" 設定、または "除外するパターンとファイル" 設定の下に新しい "除外する" 設定が追加されます。

署名しないファイルとファイル パターンは、すべての署名するファイルとファイル パターンをオーバーライドしますので注意してください。たとえば、"含める" 設定および "除外する" 設定に *.exe を指定すると、InstallShield は .exe ファイルに署名を行いません。

ショートカットのターゲットを参照するダイアログ ボックス

[ショートカット ターゲットの参照] ダイアログ ボックスを利用して、ショートカットのターゲットを指定できます。

テーブル 3-5 • [ショートカットのターゲット] ダイアログ ボックスのオプション

オプション	説明
検索先	このボックスには、現在使用できるすべてのインストール先ディレクトリがリストされます。
ファイルリスト	このリストには、[検索先] ボックスで識別されたディレクトリで見つかったファイルが表示されます。
ファイル名	ショートカットのターゲットを、選択したインストール先ディレクトリの特定のファイルにする場合、このボックスにそのファイル名を入力します。
ファイルの種類	このリストを使って、ファイル リストに表示されるファイルの種類をフィルターできます。

[タイトル ターゲットの参照] ダイアログ ボックス

InstallShield は、スタート画面上のデスクトップ アプリのタイルの外観を構成することができます。[タイル ターゲットの参照] ダイアログ ボックスを使って、タイルの外観を構成するアプリの .exe ファイルを参照して選択します。

テーブル 3-6 • [タイル ターゲットの参照] ダイアログ ボックスの設定

設定	説明
検索先	外観を構成するアプリの .exe ファイルを含むディレクトリを参照します。このリストからフォルダーを選択すると、プライマリ ペインにそのフォルダーのコンテンツが表示されます。プライマリ ペインで .exe ファイルをクリックして選択します。
ファイル名	プライマリ ペインで .exe ファイルを選択すると、ここにそのファイル名が表示されます。

[証明書の選択] ダイアログ ボックス

プロジェクトでリリースのデジタル署名情報を構成するとき、[証明書の選択] ダイアログ ボックスを使って、ファイルに署名を行うために使用する証明書を指定します。InstallShield では、次のオプションから選択できます。

- 使用中のマシンにある、署名に使用する .pfx 証明書ファイルを指定できます。

- 署名に使用する証明書を含む証明書ストアを参照できます。

[証明書の選択] ダイアログ ボックスへのアクセス

[証明書の選択] ダイアログ ボックスへのアクセス方法は、リリースまたは QuickPatch パッケージのどちらの証明書情報を指定するかによって異なります。

タスク ***リリースの [証明書の選択] ダイアログ ボックスにアクセスするには、次の手順に従います:***

1. **[リリースの準備]** の下のビュー リストにある **[リリース]** をクリックします。
2. **[リリース]** エクスプローラーで、構成するリリースを選択します。
3. ***"デジタル証明書情報"*** 設定で、省略記号ボタン (...) をクリックします。

タスク ***QuickPatch プロジェクトで [証明書の選択] ダイアログ ボックスにアクセスするには、次の手順に従います:***

1. **[パッチの設定]** の下にあるビュー リストで、**[一般情報]** をクリックします。
2. **[一般情報]** エクスプローラーで、**[ビルドの設定]** を選択します。
3. **[デジタル署名]** タブをクリックします。
4. ***"デジタル証明書情報"*** 設定の横にある **[参照]** ボタンをクリックします。

[証明書の選択] ダイアログ ボックスの設定

テーブル 3-7 ・ [証明書の選択] ダイアログ ボックスの設定

設定	説明
ファイル (.pfx) を使用	.pfx ファイルを使ってビルド時にリリースに署名を行うには、このオプションを選択します。次に、.pfx の場所を指定します。ファイルへのパスを入力するか、省略記号ボタン (...) を使ってファイルの場所を参照します。
証明書ストアを使用	ビルド時にリリースにデジタル署名を行うために使用する証明書を含む証明書ストアを参照する場合、このオプションを選択してから、このオプションの下にあるサブ設定に値を入力します。
証明書ストア名	使用する証明書を含む証明書ストアの名前を選択します。選択可能なオプションは以下のとおりです: ・ Personal ・ 信頼されたルート証明機関 ・ エンタープライズの信頼 ・ 中間証明機関 この設定は、**[証明書ストアを使用]** オプションを選択すると有効になります。
証明書ストアの場所	使用する証明書を含む証明書ストアの場所を選択します。選択可能なオプションは以下のとおりです: ・ ユーザー ・ マシン この設定は、**[証明書ストアを使用]** オプションを選択すると有効になります。
証明書サブジェクト	使用する証明書のサブジェクトを入力するか、マシン上で使用可能な証明書のリストから選択します。 この設定は、**[証明書ストアを使用]** オプションを選択すると有効になります。
詳細表示	証明書の全般的な情報や、セキュリティ、証明書パスなどの情報が **[詳細表示]** オプションに一覧表示されます。
署名ダイジェスト	署名ダイジェストのハッシュ アルゴリズムを選択する、または証明書のハッシュに基づいて InstallShield が自動的に選択します。選択可能なオプションは以下のとおりです: ・ 証明書のハッシュに従う ・ SHA-1 ・ SHA-256 ・ デュアル (SHA-1 & SHA-256)

[条件ビルダー] ダイアログ ボックス

[条件ビルダー] ダイアログ ボックスを使用して、プロジェクトの機能とカスタム アクションのオペレーティング システム条件とソフトウェア条件を作成することができます。さらに、カスタム アクションに機能条件を作成することもできます。

タスク ***[条件ビルダー] ダイアログ ボックスを起動するには、以下のいずれかを実行します。***

機能またはカスタム アクションの **"条件"** 設定で、省略記号ボタン (...) ボタンをクリックします。

メモ: *[常にインストール] 機能の条件は作成できません。*

ダイアログ ボックスの設定

[オペレーティング システム] タブ

[オペレーティング システム] タブでは、機能のインストールまたはカスタム アクションの実行を行うために、ターゲット システムが指定のオペレーティング システムを使用していることを条件として設定することができます。ターゲット システムが指定のオペレーティング システムを使用していない場合、機能はインストールされず、カスタム アクションも起動されません。

[ソフトウェア] タブ

[ソフトウェア] タブで、ソフトウェア要件の条件を作成できます。このタブで指定されたソフトウェアがターゲット システムで見つからない場合、関連付けられた機能はインストールされず、カスタム アクションも起動されません。

ソフトウェア要件の設定

ソフトウェアがターゲット システムに存在するかどうかに基づいてソフトウェア要件の条件を設定します。ソフトウェアがターゲット システムに*存在する*ときに、カスタム アクションの実行または機能のインストールを行う場合、チェック ボックスをチェック済みの状態 ☑ に設定しますチェック ボックスをこの状態にするには、このボックスに緑色のチェックマークが表れるまで繰り返しクリックします。

ソフトウェアがターゲット システムに*インストールされていない*ときカスタム アクションの実行または機能のインストールを行う場合、ソフトウェアの前のチェック ボックスに赤の "X" を設定します (☒)。チェック ボックスをこの状態にするには、このボックスに赤の X が表われるまで繰り返しクリックします。

空のチェック ボックス (☐) は、条件がソフトウェアにまったく影響されないことを示します。ソフトウェア条件を無効にするには、チェック ボックスが空になるまで繰り返しクリックします。

[機能] タブ

メモ: *[機能] タブは、カスタム アクションに条件を作成するときのみ表示されます。*

[機能] タブでは、カスタム アクションの条件に対する機能要件を選択できます。カスタム アクションは選択された機能がインストールされるよう設定されている場合のみ実行されます。

[コンテンツ ソース パス] ダイアログ ボックス

[コンテンツ ソース パス] ダイアログ ボックスは、[IIS 構成] ビューで Web サイトの "コンテンツ ソース パス (ローカル または UNC)" 設定にある UNC ボタンをクリックすると開きます。このダイアログ ボックスには、以下の設定が含まれます:

テーブル 3-8 · [コンテンツ ソース パス] ダイアログ ボックスの設定

設定	説明
Universal Naming Convention コンテンツ ソース パス	IIS Web サイトに含まれるファイルの UNC パスを指定します。例: **¥¥server¥share**

[カスタム エラー] ダイアログ ボックス

[カスタム エラー] ダイアログ ボックスは、Web サイト、アプリケーション、または IIS 仮想ディレクトリ用にカスタマイズ可能な HTTP エラーをすべてを一覧表示します。カスタム エラーは、URLまたはサーバー上のファイルへのポインターのいずれかです。

タスク ***エラーメッセージを構成するには、以下の手順を実行してください。***

1. **[カスタム エラー]** ダイアログ ボックスで 1 つまたは複数のエラーを選択します。
2. **[編集]** をクリックします。**[エラー マッピングのプロパティ]** ダイアログ ボックスが開きます。
3. メッセージの種類を選択し、必要に応じて、ファイルまたは URL を指定します。
4. [OK] をクリックします。

[デフォルト値に設定] ボタンをクリックして、選択したエラーをデフォルト設定に戻すことができます。

[依存関係] ダイアログ ボックス

[ファイル] ビューの **[インストール先コンピューターのファイル]** ペインのアイテムを右クリックして、**[ビルド時に依存関係をスキャン]** を選択すると、[依存関係] ダイアログ ボックスに依存関係の一覧が表示されます。

ダイアログ ボックスはアセンブリ DLL の結果を表示します。Microsoft Visual Studio からダイアログを起動した場合、ダイアログ ボックスはプロジェクト出力の結果を表示します。Visual Studio の外で InstallShield から起動した場合、[ビルド時に依存関係をスキャン] はプロジェクト出力で無効になります。

メモ· *これはポータブル実行可能ファイル (例、EXE、DLL、OCX) と、([ファイルのプロパティ] ダイアログ ボックス内の"COM と .NET の設定" タブを使用して)"ビルド時に .NET スキャン" プロパティに "依存関係とプロパティ" が設定されている場合のみ有効です。探している依存関係、またはその内の一つが見つからなかった場合は赤いアイコンが表示されます。*

ダイアログ ボックスの設定

依存関係

このセクションにはすべての依存関係が一覧で表示され、各ファイルの横にはチェック ボックスがあります。ビルドから依存関係を除外する場合、ファイル横のチェック ボックスの選択を解除します。[OK] をクリックしてダイアログを閉じます。

メモ: ビルド時に検出された新たな依存関係 ([依存関係] ダイアログ ボックスを閉じた後で追加されたファイル) はビルドに追加されます。

[ダイアログのイメージ] ダイアログ ボックス

[ダイアログ イメージ] ダイアログ ボックスを使用して、インストールのダイアログに表示するイメージ (.bmp、.gif、.jpg、または .ibd) を追加します。

全画面イメージ

外部ダイアログの全画面背景になるグラフィック ファイルを参照します。外部ダイアログは、インストールの最初または最後に表示されるダイアログで、InstallWelcome および SetupCompleteSuccess(インストール成功時に最後に表示されるダイアログ)などがあります。全画面のイメージ サイズは、499x312 ピクセルです。

バナーイメージ

内部ダイアログの上部で実行されるグラフィックファイルを参照します。内部ダイアログはインストール ダイアログの最初と最後の間に表示され、[使用許諾契約] ダイアログと [カスタム セットアップ] ダイアログを含みます。バナーのイメージ サイズは 499x58 ピクセルです。

[セットアップのデジタル署名] ダイアログ ボックス

[セットアップのデジタル署名] ダイアログ ボックスは、プロジェクト アシスタントの [インストールのビルド] ページにある [セットアップにデジタル署名する] をクリックすると表示されます。このダイアログ ボックスで、オブジェクト内のコードが発行以来変更または破損されていないことをエンドユーザーに対して保証できます。

テーブル 3-9 ・ [セットアップのデジタル署名] ダイアログ ボックスの設定

設定	説明
セットアップにデジタル署名する	インストールにデジタル署名を行う場合、このチェック ボックスを選択します。 このチェック ボックスを選択すると、このダイアログの他の設定が有効にされます。
証明書 URL	完全修飾 URL を入力します (例、**http://www.mydomain.com**)。この URL は、エンドユーザーが製品や組織、会社に関しての情報を得るためのリンク先として、デジタル署名の中で使用されます。

テーブル 3-9 • [セットアップのデジタル署名] ダイアログ ボックスの設定 (続き)

設定	説明
デジタル証明書情報	リリースに署名を行うために使用するデジタル証明書を指定するには、この設定の横にある省略記号 (...) ボタンをクリックします。[証明書の選択] ダイアログ ボックスが開いて、.pfx ファイルの場所を指定するか、証明書を含む証明書ストアについての情報を指定することができます。 詳細については、「[証明書の選択] ダイアログ ボックス」を参照してください。
パスワード	使用する .pfx にパスワードがある場合、それを入力します。InstallShield はパスワードを暗号化して、プロジェクト (.ise) ファイルに保存します。 ビルド時に InstallShield はパスワードを使って .pfx でファイルの署名を行います。証明書がパスワードで保護されているのにもかかわらず、この設定に何も入力しなかった場合、.pfx ファイルを使った署名が失敗します。 ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。

ヒント: *[リリース] ビューの [署名] タブでは、ビルド時に、インストールのどの部分に対してデジタル署名を行うかを指定できます。InstallShield は、作業中のプロジェクトの種類に応じて、リリースに含まれる次の任意およびすべてのファイルに署名することができます。*

- *Express プロジェクトの Windows Installer パッケージ (.msi ファイル)*
- *Express プロジェクトの* Setup.exe *ファイル*
- *リリースの任意のファイル (アプリケーション ファイルを含む)*

詳細については、「ビルド時にリリースとそのファイルにデジタル署名を行う」を参照してください。

Windows ロゴ: *インストールのすべての実行可能ファイル (.exe、.dll、.ocx、.sys、.cpl、.drv、および .scr ファイル) は、Windows ロゴ プログラムに準拠するためにデジタル署名が必要です。*

[レジストリ データの編集] ダイアログ ボックス

このダイアログ ボックスでは、インストール プロジェクトでレジストリ データを編集できます。このダイアログ ボックスを起動するには、[レジストリ] ビューの値を右クリックをして [変更] を選択します。

テーブル 3-10 ・ [レジストリ データの編集] ダイアログ ボックスの設定

設定	説明
値名	値名は、このダイアログ ボックスで読み取り専用です。値の名前を変更する場合、このダイアログ ボックスを終了して、名前を変更する値を選択し、F2 キーを押します。
値データ	ターゲットマシンで表示するのと同様に、このレジストリ値に対してデータを入力します。

[エラー マッピングのプロパティ] ダイアログ ボックス

[エラー マッピングのプロパティ] ダイアログ ボックスは、IIS 仮想ディレクトリ エラーコードとそのデフォルトプロパティを表示します。[メッセージの種類] リストを利用して メッセージに URL またはポインターにサーバー上のファイルを設定することができます。メッセージに URL を設定する場合、URLを完全な形で指定します。エラーメッセージにファイルポインターを設定する場合、エラーメッセージとして使用するファイルを参照、または、そのファイルの完全パスを入力します。これには、システムには既に存在していて、作成中のプロジェクトには存在しないファイルを含みます。

[ファイルの詳細] ダイアログ ボックス

システム検索ウィザードの [詳細] ボタンをクリックして、このダイアログ ボックスにアクセスします。このボタンは、検索方法を定義して検索するファイルを指定するとアクティブになります。このダイアログ ボックスでは次の詳細を指定することにより、検索を拡張できます：

テーブル 3-11 ・ [デバイス ファイル] ダイアログ ボックスの設定

設定	説明
最小バージョン	ファイルがターゲット システムに存在し、バージョンが入力した値より高ければ、検索は成功です。 ***メモ:*** *署名テーブルの MinVersion プロパティは、"より大きい" バージョンのファイルのみを検索します。従って、1.00.11 を検索する場合、MinVersion プロパティの値には 1.00.10 を入れる必要があります。Windows Installer 資料にはこれが違った形で説明されています。*
最大バージョン	ファイルがターゲット システムに存在し、バージョンが入力した値と等しいか、または低ければ検索は成功です。

テーブル 3-11 • [デバイス ファイル] ダイアログ ボックスの設定 (続き)

設定	説明
最小の日付	チェック ボックスを選択して、最小日で検索します。ファイルがターゲット システムに存在し、日付が入力した値と等しいか、または大きければ検索は成功です。
最大の日付	チェック ボックスを選択して、最大日で検索します。ファイルがターゲット システムに存在し、日付が入力した値と等しいか、または小さければ、検索は成功です。
最小サイズ	ファイルがターゲット システムに存在し、サイズが (バイトで)指定した値と等しいか、または大きければ、検索は成功です。
最大サイズ	ファイルがターゲット システムに存在し、サイズが (バイトで)指定した値と等しいか、または小さければ、検索は成功です。
Languages	[参照 (…)] ボタンをクリックして [言語] ダイアログ ボックスを表示します。検索の条件として複数言語を選択することができます。リストされた言語から最低 1 つマッチすれば、検索は成功です。

メモ: *編集フィールドに入力する情報はオプションです。フィールドは空白で残すこともできます。*

[ファイルのプロパティ] ダイアログ ボックス

ファイルの [プロパティ] ダイアログ ボックスでは、そのファイルがターゲット システムにインストールされるときに設定される様々な属性を決定します。

[プロパティ] ダイアログ ボックスで構成される設定で、ダイナミック リンク ファイルを設定することはできません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。

タスク ***[ファイルのプロパティ] ダイアログ ボックスを開くには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. ファイルを右クリックして、**[プロパティ]** を選択します。

このダイアログ ボックスでは次のタブが利用できます。

- 全般
- COM と .NET の設定
- 詳細

[全般] タブ

[プロパティ] ダイアログ ボックスの [全般] タブを使って、ファイルをターゲット システムにインストールするときに様々なプロパティを上書きすることができます。

ダイナミック リンク ファイルのファイルのプロパティは設定できません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。

テーブル 3-12 • [ファイルのプロパティ] ダイアログ ボックスの [全般] タブにある設定

設定	説明
場所	この読み取り専用フィールドは、このファイルが含まれるコンポーネントの保存先に従って、ターゲット システムにこのファイルをインストールするディレクトリを表示します。
フォント タイトル	フォントをインストールしている場合、ここでフォント タイトルを **FontTitle (FontType)** の形式で指定できます (例、**Roman (All res)**)。フォントがシステムに登録されている場合、InstallShield がフォント名を提供します。 .ttf または .ttc フォントに対してフォント タイトルを指定しないでください。Windows Installer では、埋め込みフォント名を読み取って登録を行います。また、InstallShield では、.ttf または .ttc ファイル用にこのフィールドに **[ファイルから読み込むタイトル]** とマークします。 さらに詳しい情報は、Windows Installer ヘルプ ライブラリの「ICE07」を参照してください。

テーブル 3-12 • [ファイルのプロパティ] ダイアログ ボックスの [全般] タブにある設定 (続き)

設定	説明
システム属性のオーバーライド	このファイルを、開発システムでこのファイルに現在設定されている同じシステムのプロパティでインストールするには、このチェック ボックスをクリアします。 ファイルの 1 つ以上のプロパティをオーバーライドするには、**[システム属性のオーバーライド]** チェック ボックスを選択してから、以下のチェック ボックスから 1 つ以上を選択します。 • **読み取り専用**—Windows Installer がファイルをインストールするときに、これを読み取り専用にする場合、このチェック ボックスを選択します。 • **隠しファイル** — Windows Installer がファイルをインストールするときに、これを隠しファイルにする場合、このチェック ボックスを選択します。 • **ファイル ハッシュの使用** — このオプションは、バージョン指定されていないファイルにのみ使用できます。Windows Installer はファイルのハッシュを使用してバージョン指定されていないファイルの不要なコピーを避けたり、検出したりできます。Windows Installer が既存ファイルをアップグレードするかどうか判断するときに、インストールに含まれるファイル ハッシュとターゲット システム上の対応するファイルのファイル ハッシュとを比較する場合、このチェック ボックスを選択します。 • **システム** — Windows Installer がファイルをシステム ファイルとしてインストールする場合は、このチェック ボックスを選択します。 • **重要** — このファイルがコンポーネントの操作に必須であることを示す場合、このチェック ボックスを選択します。何らかの理由によって重要ファイルがインストールされないと、コンポーネントもインストールされません。必須ファイルがインストールされない場合、通常表示される [中止]、[再試行]、および [無視] ボタンではなく、[再試行] および [キャンセル] オプションを含むエラーメッセージがエンドユーザーに対して表示されます。(これによって、エンド ユーザーはそのファイルをインストールしないでインストール処理を完了できます。)
システムのサイズを変更する	″サイズ″ 設定は、**″システム サイズのオーバーライド″** 設定がクリアされたとき、ファイルのサイズを表示します。 Windows Installer がインストールの必要ディスク容量を計算するときに、選択されたファイルの実際のサイズを無視して、指定された値をそのファイルのサイズとして判断するように設定するには、このチェック ボックスを選択してから ″サイズ″ 設定で適切な値をバイト単位で入力します。

テーブル 3-12 • [ファイルのプロパティ] ダイアログ ボックスの [全般] タブにある設定 (続き)

設定	説明
システムのバージョンを変更する	ファイルのバージョンが指定されている場合、**"システムバージョンの上書き"** 設定がクリアされたときに、"バージョン" 設定にはファイルのバージョンが表示されます。 Windows Installer が選択されたファイルの実際のバージョン番号を無視して、その代わりに指定された番号をバージョン番号と判断するように設定するには、このチェック ボックスを選択してから、"バージョン" 設定に適切なバージョン番号を入力します。実行時、このファイルがターゲット システム上にあるファイルと同じ名前および同じターゲット場所を持つ場合、Windows Installer がターゲット システムのファイルを現在のインストールに含まれているバージョンでアップデートするか、ファイルをそのままに残すかを決定するとき、指定されたバージョン番号を使用します。 たとえば、プロジェクトに含まれるファイルのバージョンが 2.0.0.0 で、上書きバージョンを 3.0.0.0 と入力した場合、Windows Installer は、ターゲット システム上のファイル バージョンが 3.0.0.1 以降の場合は置換しますが、3.0.0.0 以前の場合は置換しません。 Windows Installer が既存ファイルを上書きするかどうかを判断する方法についての詳細は、「ターゲット マシン上でファイルを上書きする」を参照してください。 ファイルの最大バージョン番号は、65535.65535.65535.65535 です。
システムの言語を変更する	Windows Installer が選択されたファイルの言語を無視して、その代わりに指定された言語を使用するように設定する場合は、このチェック ボックスを選択してから [言語] ボックスの言語識別コードに 10 進数値を入力します。実行時、このファイルがターゲット システム上にあるファイルと同じ名前および同じターゲット場所を持つ場合、Windows Installer がファイルを現在のインストールに含まれているバージョンでアップデートするか、ファイルをそのままに残すかを決定するとき、インストールに含まれているファイルの言語が指定された言語であると判断します。 Windows Installer が既存ファイルを上書きするかどうかを判断する方法についての詳細は、「ターゲット マシン上でファイルを上書きする」を参照してください。 フォントは埋め込み言語 ID リソースを持たないため、Font ファイルは言語 ID と共に作成しません。フォントファイルには、このエントリを空白で残します。
アクセス許可	ファイルのアクセス許可を設定するには、このボタンをクリックします。

[COM と .NET の設定] タブ

[COM と .NET 設定] タブを使って、インストールでファイルの COM および .NET プロパティを設定することができます。このタブは、InstallShield によりファイルがポータブル実行可能であると判別された場合にのみ使用できます。[オプション] ダイアログ ボックスの [ファイルの拡張子] タブで、ポータブル実行可能ファイルとして扱うファイルを指定することができます。

ダイナミック リンク ファイルの COM および .NET の設定を構成することはできません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。

テーブル 3-13 ・ [ファイルのプロパティ] ダイアログ ボックスの [COM と .NET の設定] タブにある設定

設定	説明
登録タイプ	このプロパティを使ってファイルの登録方法を指定することができます。ファイルの種類別に、次のオプションから選ぶことができます。 ・ **なし**—選択ファイルをターゲット マシンに保存ない場合、このオプションを選択します。このオプションは、すべてのファイルに対してデフォルトの設定になっています。 ・ **COM 情報の抽出**—セットアップ中に、InstallShield によりすべての COM 登録データをファイルから取り出してシステムに登録する場合は、このオプションを選択します。COM オブジェクト登録の際は、このオプションをお勧めします。 ・ **自己登録**—実行するファイルが自己登録をサポートしている場合、このオプションを選択できます。自動登録は、Windows Installer で COM 情報を取り出すことによりファイルを登録または登録解除する場合に比べて信頼性に欠けます。
ビルド時にスキャン	ビルド時に、.NET 依存関係、またはプロパティについてスキャンするには、次のオプションから 1 つを選んでください。 ・ **なし**—.NET 依存関係またはプロパティのスキャンが不要な場合はこれを選びます。 ・ **プロパティのみ**—.NET プロパティのみスキャンします。 ・ **依存関係とプロパティ**—.NET 依存関係とプロパティの存在をスキャンして確認します。インストール プロジェクトに足りない依存関係およびプロパティを追加します。
アプリケーション ファイル	プロジェクトがビルド時にスキャンされる際、このプロパティが使われます。スキャナーはその他の情報と共にこのプロパティを使用して、アセンブリの [ファイルアプリケーション] プロパティ の値を決定します。
Installer クラス	インストール時にアセンブリの Install、Commit、Rollback、および Uninstall メソッドが適切なタイミングで呼び出されるようにするには、このオプションを選択します。
COM Interop	このアセンブリで .NET COM interop が使用できるようにする場合はこの選択肢を選びます。インストール時に、COM オブジェクトでアセンブリを呼び出すレジストリ エントリがターゲット システムに作成されます。

[詳細] タブ

[ファイルのプロパティ] ダイアログ ボックスの [詳細] タブで、ファイルの登録方法とファイルをインストールするオペレーション システムを指定することができます。

ダイナミック リンク ファイルのファイルでこれらの設定を構成することはできません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。

テーブル 3-14 ・ [ファイルのプロパティ] ダイアログ ボックスの [詳細] タブにある設定

設定	説明
ターゲット オペレーティング システム	場合によって、ターゲット システムのオペレーティング システムに合わせて、異なるバージョンの機能をインストールする必要があります。**"ターゲット オペレーティング システム"** 設定を使って、ファイルをインストールするオペレーティング システムを指定できます。詳細については、「製品のオペレーティング システム要件を指定する」を参照してください。 すべてのサポート対象のプラットフォームにファイルをインストールするには、[**すべてのオペレーティング システム**] チェック ボックスを選択します。デフォルトではこのオプションが選択されています。ファイルが対象にする特定のオペレーティング システムを選択するには、このボックスをクリアします。
インストール/アンインストールのプロパティ	[インストール/アンインストールのプロパティ] エリアでは、インストーラーが、インストールおよびアンインストールに関してこのファイルをどのように処理するか指定できます。有効なオプションは次のとおりです: ・ **パーマネント**—このファイルを永久にターゲット システム上に残す場合、このオプションを選択します。インストーラーは、アンインストール中にこのファイルを削除しません。 ・ **共有**—[共有] チェック ボックスを選択すると、インストーラーがこのファイルの参照カウント ("refcount") を行うようにすることができます。 ファイルが共有にマークされると、参照カウントが存在しない場合には Windows Installer により参照カウントが作成され、存在する場合には増分されます。Windows Installer は、すべての共有ファイルのカウントが個別に保持しますが、標準の参照カウントは以下のレジストリ キーの下に保存されます。 HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥SharedDLLs このカウントは、ファイルがアンインストールされると減少されます。

テーブル 3-14 • [ファイルのプロパティ] ダイアログ ボックスの [詳細] タブにある設定 (続き)

設定	説明
ファイルの上書きプロパティ	**[ファイルの上書きプロパティ]** 領域では、ファイルがターゲット システムに既に存在する場合、インストーラーがこのファイルをどのように処理するか指定できます。 **Windows Installer バージョン規則 (推奨)**—このオプションを選択すると、ターゲット システムに既に存在するファイルを置き換えするかどうかを判別に Windows Installer バージョン規則が使われます。Windows Installer は次の規則を強制します。 • **バージョン付きファイル**—ターゲット マシン上のファイルの方がインストールされるバージョンより新しい場合でも、常に最新バージョンのファイルが保持されます。さらに、バージョン指定されていないファイルではなく、バージョン指定されたファイルが保持されます。 • **ファイル言語**—他の要素が同じ場合、インストールと同じ言語を持つファイルが他言語のバージョンより優先的に保持されます。この規則に対する例外はファイルが複数言語だった場合にのみ適用されます。単一言語のバージョンのファイルに対して複数言語のファイルが保持されます。 • **日付**—ターゲットマシン上に既に存在するファイルの変更日がそのファイルの作成日以降であった場合、ファイルは上書きされません。この規則によってアップグレードまたは再インストール中にユーザー環境設定ファイルが消去されるのを防ぐことができます。 **上書きしない**—このオプションを選択すると、(ターゲット システムに存在する場合) ファイルは、ファイルのバージョンに関係なく、決して上書きされません。 **常に上書き**—このオプションを選択すると、(ターゲット システムに存在する場合) ファイルは、ファイルのバージョンに関係なく、常に上書きされます。

ファイル 削除の [プロパティ] ダイアログ ボックス

ファイル 削除の [プロパティ] ダイアログ ボックスには、ターゲット システムから削除するように選択されたファイルまたはフォルダーの情報が表示されます。このファイルとフォルダーの削除機能は、アプリケーションによって作成されるファイルの削除など、インストールが追跡を行わない処理に使用すると便利です。

タスク ***[ファイルの削除プロパティ] ダイアログ ボックスを開くには、以下の手順に従います:***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[ファイル]** をクリックします。
2. **[インストール先コンピューターのフォルダー]** ペインで、構成したいファイル削除項目を含むフォルダーをクリックします。
3. **[インストール先コンピューターのファイル]** ペインで、構成するファイル削除項目を右クリックしてから、**[プロパティ]** をクリックします。

   InstallShield では、削除するように構成されたファイルまたはフォルダーを識別するために赤い X 印のアイコンが使用されます。

ファイルとフォルダーの削除項目をプロジェクトに追加する方法については、「ターゲット システムからファイルとフォルダーを削除する」を参照してください。

テーブル 3-15 ・ [プロパティ] ダイアログ ボックスの設定

設定	説明
場所	この設定は、削除するように構成されている選択済みの項目の場所を示します。
フォルダーが空の場合は削除する	選択された削除項目が、ターゲット システム上で空白の場合に削除するフォルダーである場合、このオプションを選択します。
ファイルをフォルダーから削除する	選択された削除項目が ″場所設定″識別されるフォルダー内にあるファイルで、これをターゲット システムから削除する場合は、このオプションを選択します。
ファイル名	この設定は、[**ファイルをフォルダーから削除する**] オプションを選択すると使用できます。 変更するファイルの名前を指定します。
削除のスケジューリング	ターゲット システムから選択したファイルまたはフォルダーを削除するタイミングを指定します。ファイルまたはフォルダーに関連付けられた機能が、次の 1 つの方法で処理されるときに削除処理が発生します: ・ **機能のインストール** – 選択された項目の機能がインストールされるとき、フォルダー削除項目の場合は、そのフォルダーが空の場合に、選択された項目が削除されます。 ・ **機能のアンインストール** – 選択された項目のコンポーネントがアンインストールされるとき、フォルダー削除項目の場合は、そのフォルダーが空の場合に、選択された項目が削除されます。 ・ **機能のインストールおよびアンインストール** – 選択された項目の機能がインストール、またはアンインストールされるときに、フォルダー削除項目の場合は、そのフォルダーが空の場合に、選択された項目が削除されます。

[フォルダーのプロパティ] ダイアログ ボックス

このダイアログ ボックスには、InstallShield で定義されたフォルダーの種類と場所についての情報と、セットアップのビルド時のファイルの動的取得方法を決定するオプションを用意されています。

このダイアログからは、次の 2 つのタブが使用できます。

- 全般
- ファイルのリンク

[全般] タブ

[フォルダーのプロパティ] ダイアログ ボックスの [一般] パネルには、選択ディレクトリに置いたファイル数、およびそれらのファイルに必要な全体のディスク容量などの情報が表示されます。このパネルの情報は編集できません。[フォルダープロパティ] パネルを表示するには、[ファイル] ビューでインストール先フォルダーを右クリックして、[プロパティ] を選択します。

[ファイルのリンク] タブ

[フォルダーのプロパティ] ダイアログ ボックスの [ファイルのリンク] タブでは、ダイナミック フォルダーを指定できます。このダイアログ ボックスは、[ファイル] ビューのインストール先フォルダーを右クリックして、[ダイナミック ファイル リンク] を選択することでアクセスできます。

テーブル 3-16 · [フォルダーのプロパティ] ダイアログ ボックスの [ファイルのリンク] タブにある設定

設定	説明
ソース フォルダー	動的にリンクするフォルダーへのフルパスを入力するか、[参照] ボタンをクリックしてファイルを指定します。
サブフォルダーを含める	各サブフォルダー内のファイルにダイナミック リンクを持たせるには、このチェック ボックスを選択します。 InstallShield で、サブフォルダー内にあるダイナミック リンクを持つファイルにコンポーネントがどう作成されるかについては、「ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する」を参照してください。
すべてのファイルの自己登録	ダイナミック リンクのすべてのファイルを自己登録する場合、このチェック ボックスを選択します。

テーブル 3-16・[フォルダーのプロパティ] ダイアログ ボックスの [ファイルのリンク] タブにある設定 (続き)

設定	説明
ベスト プラクティス メソッドを使って作成	ダイナミック リンクがあるファイルのコンポーネントを作成するとき、ベスト プラクティスを順守することを指定する場合、このチェック ボックスを選択します。コンポーネント作成のベスト プラクティスに従うと、選択と除外のフィルター基準を満たすすべてのファイルに対して、次のタスクがビルド時に実行されます: ・ ダイナミック リンクがあるフォルダーにある各ポータブル実行可能 (PE) ファイルについてコンポーネントが別々に作成されます。各 PE ファイルは、そのコンポーネントのキー ファイルです。 ・ ダイナミック リンクのルート レベルにあるすべての非 PE ファイルがリンクを含むコンポーネントに追加されます。 ・ ダイナミック リンクにサブフォルダーが含まれている場合、サブフォルダー内にあるすべての非 PE ファイルに新しいコンポーネントが作成されます。ダイナミック リンクに複数のサブフォルダーが含まれている場合、各サブフォルダー内のすべての非 PE ファイルにコンポーネントが 1 つずつ別々に作成されます。 ダイナミック リンクがあるファイルのコンポーネントを作成するとき、ベスト プラクティスを順守しないことを指定する場合、このチェック ボックスをクリアします。コンポーネント作成メソッドでは、選択と除外のフィルター基準を満たすすべてのファイルに対して、次のタスクがビルド時に実行されます: ・ ファイルの種類に関わらず、ダイナミック リンクがあるソース フォルダーのルート レベルにあるすべてのファイルにコンポーネントが 1 つ作成されます。 ・ ダイナミック リンクに 1 つまたは複数のサブフォルダーが含まれている場合、ファイルの種類に関わらず、各サブフォルダーのすべてのファイルにコンポーネントが 1 つずつ作成されます。サブフォルダーのコンポーネント内にある最初のダイナミック リンクが付いたファイルが、そのコンポーネントのキー ファイルです。 すべての新しいダイナミック リンクについて、このチェック ボックスがデフォルトで選択されています。 ***ヒント・** コンポーネントは、個別にインストール可能な最小の製品構成単位です。InstallShield Express Edition では、コンポーネントが自動で作成されます。詳細については、「インストールの基本」を参照してください。* 詳細については、「ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する」を参照してください。
すべてのファイルを含める	インストール内のリンクがあるディレクトリの全コンテンツを含める場合、このオプションを選択します。

テーブル 3-16 • [フォルダーのプロパティ] ダイアログ ボックスの [ファイルのリンク] タブにある設定 (続き)

設定	説明
次のワイルド カード パターンを基にファイルを選択/除外する	ファイルの種類を選択または除外する場合、このオプションを選択します。先頭にアスタリスク (*) をつけて、拡張子を "選択" または "除外" フィールドに入力します。複数のエントリはカンマで区切ります。 たとえば、すべての画像ファイルがサウンド ファイルと共に 1 つのフォルダーの中にあるとき、画像ファイルのみダイナミック リンクを付加する場合、ダイナミック リンクがあるフォルダーに .bmp ファイルと .ico ファイルのみを含めるように指定することができます。これを行うには、以下の例のように、選択パターンにアスタリスク (*) を使用します: ***.bmp、*.ico** 特定のファイルを選択または除外する場合、選択また除外のパターン ボックスに完全なファイル名を入力します。

メモ: *ダイナミック ファイル リンクの制限の詳細については、「ダイナミック ファイル リンク」を参照してください。*

[InstallShield 前提条件のプロパティ] ダイアログ ボックス

[InstallShield 前提条件プロパティ] ダイアログ ボックスは、[再配布可能ファイル] ビューで選択された InstallShield 前提条件を右クリックしてから [プロパティ] をクリックすると開きます。このダイアログ ボックスでは、選択された InstallShield 前提条件の場所を指定することができます。詳細については、「リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する」を参照してください。

ビルドの場所

このリストから適切なオプションを選択します。次の 3 つのオプションがあります。

- **Web からダウンロードする**—プロジェクトに含まれるすべての InstallShield 前提条件ファイルを、必要に応じて各前提条件の InstallShield 前提条件ファイル (.prq) で指定された URL からダウンロードする場合、このオプションを選択します。
- **Setup.exe から抽出する**—実行時に抽出されるように InstallShield 前提条件ファイルを Setup.exe に圧縮する場合、このオプションを選択します。
- **ソース メディアからコピーする**—InstallShield 前提条件ファイルをソース メディアのルート ディレクトリに格納する場合、このオプションを選択します。

ヒント: *[Setup.exe から抽出する] オプションまたは [ソース メディアからコピーする] オプションを選択してから、コンピューター上で利用できない InstallShield 前提条件を含むリリースをビルドした場合、前提条件が必要とする各ファイルについて 1 つまたは複数のビルド エラーが生成されます。これらのビルドエラーを回避するには、インターネットからコンピューターへ InstallShield 前提条件のダウンロードを行なうか、リリースをビルドする前にプロジェクトから削除します。*

[Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックス

[一般情報] ビューにある [MSI ログの作成] 設定で [参照 (...)] ボタンをクリックすると、[Windows Installer 4 ログ記録オプション] ダイアログ ボックスが表示されます。このダイアログ ボックスを使うと、Windows Installer 4.0 がインストールをログ記録するかどうかについて、コマンド ラインの使用やレジストリでのログ パラメーターの構成の必要なしに、プロジェクト全体を通して指定することができます。このダイアログ ボックスを使用して、ログ記録されるメッセージの種類をカスタマイズすることもできます。

テーブル 3-17 • [Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックス

オプション	説明
いいえ	インストールはログ記録されません。これがデフォルトの値です。
はい (MsiLoggingは voicewarmupx のデフォルト値に設定されます)	InstallShield によって **voicewarmupx** のデフォルト値が MsiLogging プロパティに挿入されます。 Windows Installer 4.0 が搭載されたターゲット システム上でインストールが実行されると、次の処理が行なわれます。 • voicewarmupx のデフォルト ログ モードに従って、インストーラーがログ ファイルを作成します。 • インストーラーが、MsiLogFileLocation プロパティに、ログ ファイルのパスを挿入します。 • SetupCompleteSuccess、SetupCompleteError および SetupInterrupted ダイアログに **[Windows Installer ログを表示]** チェック ボックスが追加されます。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。 以前のバージョンの Windows Installer はこの設定を無視します。以前のバージョンの Windows Installer を実行中のシステム上で表示される実行時ダイアログでは、**[Windows Installer ログを表示]** チェック ボックスは表示されません。

テーブル 3-17 • [Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックス (続き)

オプション	説明
カスタム MsiLogging 値	InstallShield は、このボックスに指定された値を MsiLogging プロパティに挿入します。 Windows Installer 4.0 が搭載されたターゲット システム上でインストールが実行されると、次の処理が行なわれます。 • このボックスに指定されたカスタム値に基づいて、インストーラーがログ ファイルを作成します。 • インストーラーが、MsiLogFileLocation プロパティに、ログ ファイルのパスを挿入します。 • SetupCompleteSuccess、SetupCompleteError および SetupInterrupted ダイアログに **[Windows Installer ログを表示]** チェック ボックスが追加されます。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。 以前のバージョンの Windows Installer はこの設定を無視します。以前のバージョンの Windows Installer を実行中のシステム上で表示される実行時ダイアログでは、**[Windows Installer ログを表示]** チェック ボックスは表示されません。

重要: MsiLogFileLocation *プロパティは、読み取り専用のため、ログ ファイルの場所の設定および変更には使用できません。*

[マージ モジュールの構成可能な値] ダイアログ ボックス

構成可能な再配布可能ファイルは、マージ モジュールまたは ModuleConfiguration テーブルで少なくとも 1 つの行を持ち、ModuleSubstitution テーブルで少なくとも 1 行によって参照されるオブジェクトです。これによって再配布可能ファイルの値を変更することができます。

[マージ モジュールの構成可能な値] ダイアログ ボックスの表示

[再配布可能ファイル] ビュー内で構成可能なマージ モジュールまたはオブジェクトを選択すると、[マージ モジュール構成可能値] ダイアログ ボックスが表示されます。選択された再配布可能ファイルがオブジェクトの場合、[マージ モジュールの構成可能な値] ダイアログ ボックスは、オブジェクト ウィザードが終了したとき表示されます。

構成可能なマージ モジュールまたはオブジェクトを右クリックして、**[マージ モジュールの構成]** を選択することもできます。

ダイアログ ボックスの設定

ダイアログ ボックスには、構成可能値を変更できるグリッドが含まれています。左の列には構成可能値の名前が含まれています。右の列には、値のオプションが含まれています。ドロップダウンメニューから新しい値を選択することができます。値に関する情報については、再配布可能ファイル ベンダーのマニュアルを参照してください。

使用する値を指定したあと、OK をクリックして新しい値を保存します。これらの値は、プロジェクトをビルドするとき、.msi パッケージをビルドするために使用されます。

デフォルトに戻す

このボタンをクリックして、再配布可能ファイルのデフォルトの設定を復元します。再配布可能ファイル内の構成可能な値はすべてデフォルトに戻されます。

[マージ モジュールのプロパティ] ダイアログ ボックス

[マージ モジュールのプロパティ] ダイアログ ボックスは、[再配布可能ファイル] ビューで選択されたマージ モジュールまたは再配布可能ファイルを右クリックすると表示されます。

ダイアログ ボックスの設定

GUID

マージ モジュールの一意の GUID を表示します。

作成者

マージ モジュールの作成者を表示します。

バージョン

マージ モジュールのバージョンを表示します。

Destination

マージ モジュールのファイルの保存先を指定します。デフォルト設定 (**マージ モジュールのデフォルトのインストール先を使用する**) の使用をお勧めします。

[MIME の種類] ダイアログ ボックス

[MIME の種類] ダイアログ ボックスを使って、ファイル名拡張子と、Web サーバーからブラウザまたはメール クライアントにスタティック ファイルとして提供される対応コンテンツの種類とのマッピングを追加、編集、または削除します。

[MIME の種類] ダイアログ ボックスは、[IIS 構成] ビュー内から利用することができます。このダイアログ ボックスを開くには、エクスプローラーで Web サイト、アプリケーション、または仮想ディレクトリをクリックします。次に、"MIME の種類" 設定で省略記号ボタン (...) をクリックします。

テーブル 3-18 ・ [MIME の種類] ダイアログ ボックスの設定

設定	説明
追加	ファイル名拡張子と、Web サーバーからブラウザまたはメール クライアントにスタティック ファイルとして提供される対応コンテンツの種類とのマッピングのためのエントリを追加します。これによって、[MIME の種類を追加] ダイアログ ボックス が開きます。

テーブル 3-18・[MIME の種類] ダイアログ ボックスの設定 (続き)

設定	説明
編集	既存する MIME の種類を編集するには、MIME の種類を選択して、このボタンをクリックします。
Delete	既存する MIME の種類を削除するには、MIME の種類を選択して、このボタンをクリックします。

[MSI 値] ダイアログ ボックス

[MSI 値] ダイアログ ボックスで、表示された Windows Intaller パッケージの Registry テーブルにあるプライマリ キーを作成または変更できます。詳細については、「Registry のプライマリ キーを指定する」を参照してください。

[複数行文字列値] ダイアログ ボックス

[複数行の文字列値] ダイアログ ボックスでは、選択したレジストリ値を各ヌル区切り文字列に行をタイプすることで変更することができます。

タスク *[複数行の文字列値] ダイアログ ボックスに複数行の文字列値を入力するには:*

1. レジストリ値をどのように変更するかを選択します。次のオプションから選択できます。
   - 後に追加する
   - 前に追加
   - Replace
2. ダイアログ ボックスのグリッド部分で、各ヌル区切り文字列の行を入力するか変更します。グリッドを右クリックしてコンテキストメニューを表示するか、次の動作に関連した以下のグリッドで次のキーを押します。

   テーブル 3-19・グリッドで文字列を入力するためのコンテキストメニュー コマンド

アクション	ショートカット キー
文字列の追加	Ins
文字列の名前変更	F2
文字列の削除	Delete
上へ移動	
下へ移動	

3. 各ヌル区切り文字列の行入力が終わったら、[OK] をクリックします。文字列は自動的に連結されます。

メモ: 文字列にはスペースだけを含めることはできますが、空白にしたり、文字列の区切り文字である [~] を使うことはできません。

[新規プロジェクト] ダイアログ ボックス

InstallShield で新規プロジェクトを作成するとき、[新規プロジェクト] ダイアログ ボックスが開きます。このダイアログ ボックスではプロジェクトの種類を選択したり、プロジェクトに名前を付けたり、プロジェクト ファイルの場所を指定したりできます。プロジェクト タイプを選択してから [OK] をクリックすると、InstallShield インストール開発環境 (IDE) が開きます。

メモ: Express プロジェクトを選択すると、プロジェクト アシスタントが起動され、プロジェクトの作成をお手伝いします。

テーブル 3-20 · [新規プロジェクト] ダイアログ ボックスの設定

設定	説明
プロジェクト名	このフィールドにプロジェクト名を入力します。
プロジェクト言語	インストール プロジェクトで利用する言語を選択します。
場所	場所を入力するか、[参照] をクリックしてプロジェクトの場所に移動します。表示されているデフォルトのプロジェクトの場所を変更するには、[オプション] ダイアログ ボックスの [ファイルの場所] タブにある [プロジェクトの場所] パスを変更します。
[プロジェクト名] サブフォルダーにプロジェクトファイルを作成	このオプションを選択すると、InstallShieldは、プロジェクトフォルダーの下にプロジェクトの名前を持つサブフォルダーを作成します。

***エディション:** InstallShield の Premier Edition と InstallShield Professional Edition では、新規プロジェクトを作成するとき、追加の機能が提供されています。たとえば、Premier Editionでは同インストール プロジェクトで複数言語を使用することができます。*

[オプション] ダイアログ ボックス

[オプション] ダイアログ ボックスを使うと、InstallShield インストール開発環境 (IDE) におけるプロジェクト作成と作業環境を指定することができます。これらのグローバル設定を表示および編集するには、[ツール] メニューで [オプション] を選択します。

[オプション] ダイアログ ボックスは、複数のタスク関連タブで構成されています:

- 全般

- ファイルの場所
- プリファレンス
- マージモジュール
- 品質
- アップデート
- .NET
- [ファイル] ビュー
- ファイル拡張子
- 前提条件

[全般] タブ

[オプション] ダイアログ ボックスの [全般] タブでは、ヘルプ システムとビルド エラーのプリファレンスを指定します。

テーブル 3-21 ・ [全般] タブの設定

設定	説明
ヘルプ ウィンドウを常に手前に表示する	ヘルプ ウィンドウを IDE の手前に固定したい場合は、このチェック ボックスを選択します。IDE 内をクリックしたときにヘルプウィンドウを背景に表示する場合は、このボックスのチェックを外します。
最初のエラーの発生時にビルド プロセスを停止	ビルドエラーが発生したときにビルド プロセスを中止するには、このチェック ボックスを選択します。

[ファイルの場所] タブ

[オプション] ダイアログ ボックスの [ファイルの場所] タブでは、プロジェクト ファイルのデフォルト ディレクトリを設定します。

テーブル 3-22 ・ [ファイルの場所] タブの設定

設定	説明
プロジェクトの場所	このフィールドには、インストール プロジェクト ファイルの保存場所を参照するか、そのパスを入力します。 この場所は、新しいインストール プロジェクトのデフォルトフォルダーになります。プロジェクト (.ise) ファイル、インストール パッケージ (.msi ファイル)、およびディスク イメージ ファイルなどのソース ファイルやリリース ファイルはすべて、[プロジェクトの場所] のサブフォルダーに保管されています。 デフォルトでは、プロジェクトはハードドライブのルート階層にある *InstallShield 2021 Projects* フォルダーに格納されます。

[プリファレンス] タブ

[オプション] ダイアログ ボックスの [プリファレンス] タブでは、コマンドの実行およびプロジェクトの再ロードのためのプリファレンスを指定します。

テーブル 3-23 ・ [プリファレンス] タブの設定

設定	説明
インストール前にアンインストールする	[ビルド] メニューからインストールを再試行する前に、自動的に製品ををアンインストールする場合、このチェック ボックスを選択します。
起動時に最後に開いたプロジェクトを再ロードする	InstallShield を起動したときに、最後に開いたプロジェクトを自動的に再ロードするように設定するには、このボックスを選択します。

[マージ モジュールのオプション] タブ

[オプション] ダイアログ ボックスの [マージ モジュールのオプション] タブでは、マージ モジュールの場所とファイル検索のプリファレンスを指定します。

テーブル 3-24 ・ [マージ モジュールのオプション] タブの設定

設定	説明
マージ モジュールの場所 (現在のユーザー) とマージ モジュールの場所 (All Users)	マージ モジュール (.msm ファイル) を格納する場所へのパスを入力します。パスを追加するには、次の例のようにカンマで区切ります。 C:¥MergeModules,C:¥My Files¥MergeModules 以下の例のように、パスにはパス変数を使用できます: <ISProductFolder>¥Modules¥i386,<ISProjectFolder>¥MyCustomModules パスには環境変数を使用できます。 All Users オプションは、ユーザー設定を簡単に更新できないシステム アカウントでコマンドラインのビルドを実行している場合に使用できます。 InstallShield では、マージ モジュールを含むフォルダーを指定するその他の方法も提供されています。詳細については、「マージ モジュールを含むディレクトリを指定する」を参照してください。 ***メモ:*** *[すべてのユーザー] オプションを構成するためには、InstallShield を管理者権限で実行する必要があります。管理者権限を持っていない場合、このオプションは無効です。詳細については、「InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い」を参照してください。*

テーブル 3-24 • [マージ モジュールのオプション] タブの設定 (続き)

設定	説明
一致するファイルは同じバージョンでなければならない	ファイルをプロジェクトに追加したとき、InstallShield はマージ モジュールを検索し、同じファイルを含むモジュールがないかどうかを確認します。一致するファイルがあった場合はそれを通知します。 一致するファイルがバージョンも同じものでなくてはならない場合、このチェック ボックスを選択します。
一致するファイルはすべて同じインストール先を持つ	ファイルをプロジェクトに追加したとき、InstallShield はマージ モジュールを検索し、同じファイルを含むモジュールがないかどうかを確認します。一致するファイルがあった場合はそれを通知します。 一致するファイルのインストール先も同じでなくてはならない場合、このチェック ボックスを選択します。

[.NET] タブ

[オプション] ダイアログ ボックスの [.NET] タブでは、.NET プロジェクトの環境設定を行います。また、ここでは .NET Framework と共に含まれるユーティリティ Regasm.exe および InstallUtilLib.dll ファイルの場所も指定します。これらのユーティリティは COM interop と .NET カスタム アクションで利用されます。

テーブル 3-25 • [.NET] タブの設定

設定	説明
デフォルトのビルド時 .NET スキャンファイル設定	このリストでは、プロジェクトに追加される新規ポータブル実行可能ファイルの "ビルド時の .NET スキャン" 設定の構成方法を選択します。
.NET Framework ファイルの場所	Regasm.exe と InstallUtilLib.dll は .NET Framework の各バージョンに含まれるユーティリティです。.NET インストーラー クラスおよび COM Interop を含むリリースのビルド時に使用する、これらのファイルのバージョンを含むディレクトリのパスを指定します。 • **32 ビットの場所** — Regasm.exe および InstallUtilLib.dll のパスを入力またはその場所を参照します。 • **64 ビットの場所** — InstallShield を 32 ビット システム上で使用している場合、このオプションは無効です。InstallShield を 64 ビット システムで使用している場合、Regasm.exe および InstallUtilLib.dll の場所を参照するか、そのパスを入力します。 ***メモ:*** *これらのオプションを構成するためには、InstallShield を管理者権限で実行する必要があります。管理者権限を持っていない場合、このオプションは無効です。詳細については、「InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い」を参照してください。*

[ファイル ビュー] タブ

[オプション] ダイアログ ボックスの [ファイル ビュー] タブで、InstallShield のビューの様々な領域でどの列を表示するかなど、ユーザー設定を指定することができます。

このタブで変更を行うとき、プロジェクトを一度閉じてから開くと変更点が有効になります。

テーブル 3-26 ・ [ファイル ビュー] タブの設定

設定	説明
サイズ	InstallShield の次の領域に [サイズ] 列を含めるには、このチェックボックスを選択します。 ・ [ファイルと機能] ビュー ・ [ファイル] ビュー ・ プロジェクト アシスタントの [アプリケーション ファイル] ページ
リンク先	InstallShield の次の領域に [リンク先] 列を含めるには、このチェックボックスを選択します。 ・ [ファイルと機能] ビュー ・ [ファイル] ビュー ・ プロジェクト アシスタントの [アプリケーション ファイル] ページ
更新日時	InstallShield の次の領域に [変更日] 列を含めるには、このチェックボックスを選択します。 ・ [ファイルと機能] ビュー ・ [ファイル] ビュー ・ プロジェクト アシスタントの [アプリケーション ファイル] ページ
Destination	プロジェクト アシスタントの [アプリケーション ファイル] ページに [インストール先] 列を含めるには、このチェックボックスを選択します。
バージョン	InstallShield の次の領域に [バージョン] 列を含めるには、このチェックボックスを選択します。 ・ [ファイルと機能] ビュー ・ [ファイル] ビュー ***重要:*** *[バージョン] チェック ボックスを選択すると、前述のビューのパフォーマンスが低下します。*
ファイル キー	[ファイル] ビューに [キー] 列を含めるには、このチェックボックスを選択します。

テーブル 3-26 • [ファイル ビュー] タブの設定 (続き)

設定	説明
表示するファイルの最大数	InstallShield の次の領域でダイナミック ファイル リンクに表示するファイルの最大数を指定します。 • [ファイルと機能] ビュー • [ファイル] ビュー • プロジェクト アシスタントの [アプリケーション ファイル] ページ デフォルト値は 1000 です。ダイナミック ファイル リンクに指定された数以上のファイルが含まれている場合、ファイル リストの先頭項目に ***リストが切り捨てられました*** と表示されます。

[ファイルの拡張子] タブ

[オプション] ダイアログ ボックスの [ファイルの拡張子] タブでは、[ポータブル実行可能ファイル] (PE ファイル) のプリファレンスを指定しますInstallShield は、プロジェクトにコンポーネントを作成するとき、この PE ファイル一覧を参照します。(コンポーネントは、個別にインストール可能な最小の製品構成単位です。InstallShield Express Edition では、コンポーネントが自動で作成されます。詳細については、「インストールの基本」を参照してください。)例:

- [ファイル] ビューの **[インストール先コンピューターのフォルダー]** ペインで PE ファイルをフォルダーに追加すると、新しいコンポーネントが作成され、そのコンポーネントのキー ファイルとして設定されます。
- ダイナミック ファイル リンクを作成するときベスト プラクティス メソッドを使用すると、ビルド時に、ダイナミック リンクがあるフォルダー内にある各 PE ファイルについてコンポーネントが別々に作成されます。各 PE ファイルは、そのコンポーネントのキー ファイルです。[ベスト プラクティス] ダイナミック ファイル リンクについては、「ダイナミック リンクがあるファイルの適切なコンポーネント作成方法を判別する」を参照してください。

以下は、このタブで提供されている設定です:

テーブル 3-27 • [ファイルの拡張子] タブの設定

設定	説明
ポータブル実行可能ファイルの拡張子	このボックスには、InstallShield に PE ファイルとして認識させるファイル拡張子を入力します。拡張子をコンマで分けます。以下は、このボックスのデフォルトのエントリです: AX,EXE,DLL,OCX,VXD,CHM,HLP,TLB

[前提条件] タブ

[オプション] ダイアログ ボックスの [前提条件] タブで、InstallShield 前提条件に関するプリファレンスを設定することができます。

テーブル 3-28 ・ [前提条件] タブの設定

設定	説明
前提条件ファイルの場所 (現在のユーザー) と前提条件ファイルの場所 (すべてのユーザー)	InstallShield 前提条件ファイル (.prq ファイル) を格納するフォルダのパスを指定します。 複数の場所を指定する場合、次の例のように、各パスをカンマで区切ります: C:¥Prerequisites,C:¥My Files¥Prerequisites 以下の例のように、パスにはパス変数を使用できます: <ISProductFolder>¥SetupPrerequisites,<ISProjectFolder>¥MyCustomPrerequisites All Users オプションは、ユーザー設定を簡単に更新できないシステム アカウントでコマンドラインのビルドを実行している場合に使用できます。 InstallShield では、InstallShield 前提条件ファイル ファイルを含むフォルダーを指定するその他の方法も提供されています。詳細については、「InstallShield 前提条件を含むディレクトリを指定する」を参照してください。 ***メモ・** [すべてのユーザー] オプションを構成するためには、InstallShield を管理者権限で実行する必要があります。管理者権限を持っていない場合、このオプションは無効です。詳細については、「InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い」を参照してください。*

[出力] ダイアログ ボックス

[出力] ダイアログ ボックスは、ファイル システム エディターでプロジェクト出力グループについての情報を表示します。このダイアログ ボックスは、Microsoft Visual Studio で作成された InstallShield プロジェクトで使用できます。

タスク ***[出力] ダイアログ ボックスにアクセスするには、以下のいずれかを実行します。***

- プロジェクト出力グループが**ファイル システム エディター**で選択されたとき、**[プロパティ]** ウィンドウで **"出力"** プロパティを選択します。
- **[ファイル]** ビューの **[ソース コンピューターのファイル]** ペインでアイテムを右クリックし、**[プロジェクト出力の解決]** を選択します。
- [ファイル] ビューの **[インストール先コンピューターのファイル]** ペインでアイテムを右クリックし、**[プロジェクト出力の解決]** をクリックします。

メモ: 複数のプロジェクト出力グループが選択された場合、情報は最初の選択されたグループに対してのみ表示されます。

ダイアログ ボックスの設定

ターゲット名

選択されたプロジェクト出力グループのファイル名をターゲット コンピューターで表示されるとおりに表示します。このフィールドは読み取り専用です。

ソース パス

開発コンピューター上でのプロジェクト出力グループ ファイルへのパスを表示します。このフィールドは読み取り専用です。

ファイルとディレクトリの [アクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックスで、ロックダウンされた環境で製品を実行するエンドユーザーのために、ファイルとフォルダーを保護するための設定を構成することができます。ファイルまたはフォルダーのアクセス許可を特定のグループとユーザーに割り当てることができます。たとえば、管理者グループに特定のファイルについての [読み取り]、[書き込み]、および [削除] アクセス許可を割り当てることができますが、別のグループのすべてにユーザーについては [読み取り] 許可のみ割り当てることができます。

プロジェクトの [一般情報] ビューにある "ロックダウンの設定方法" 設定の選択に従って、InstallShield は ISLockPermissions テーブルまたは LockPermissions テーブルのどちらかにアクセス許可データを追加します。詳細については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

次のテーブルは、[アクセス許可] ダイアログ ボックスの各領域についての説明です。

テーブル 3-29 ・ [アクセス許可] ダイアログ ボックスの各領域

領域	説明
名前	このグリッドで、ドメイン名およびユーザー名を任意に組み合わせて入力できます。エントリを追加するには、グリッド上で右クリックして [新規] をクリックします。同じコンテキストメニューを利用して、エントリを変更または削除することができます。 現在のユーザーのドメインを指定するには、"ドメイン" フィールドに [%USERDOMAIN] を選択します。現在インストールを実行中のユーザーを指定するには、"ユーザー" フィールドに [LogonUser] を選択します。ローカル システム上におけるユーザー アカウントのアクセス許可を設定するには、"ドメイン" フィールドを空白のままにします。"ドメイン" または "ユーザー" フィールドには、実行時に設定される任意の Windows Installer プロパティを入力できます (例、[MYPROPERTY])。 [一般情報] ビューの "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションを選択した場合、"ユーザー" フィールドには、よく使用されるセキュリティ識別子 (SID) の一覧が表示されます。ほとんどの SID は、[従来型の Windows Installer 処理] オプションではサポートされていません。 [カスタム InstallShield 処理] オプションは、"ユーザー" フィールドに一覧されているすべての SID の翻訳された名前をサポートします。従来型のオプションで、非英語システム上で翻訳された名前を使ってアクセス許可を設定すると、インストールが失敗する可能性があります。 ***ヒント:*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*
アクセス許可	[名前] 領域で名前を選択し、[アクセス許可] ボックスにあるチェックボックスを選択またはクリアして、対応するファイルまたはフォルダーのアクセス許可を構成します。アクセス許可を選択すると、[詳細] ボタンをクリックして、他の関連するアクセス許可と詳細設定を指定することができます。

テーブル 3-29 • [アクセス許可] ダイアログ ボックスの各領域 (続き)

領域	説明
アクセス拒否	[アクセス許可] ボックスで選択するアクセス許可を明示的に拒否するには、このチェック ボックスを選択します。 このチェック ボックスは、[一般情報] ビューにある "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションが選択されている場合のみ利用できます。[従来型の Windows Installer 処理] オプションは、この動作をサポートしません。 ***ヒント:*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*

[特殊なアクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックスで [詳細設定] ボタンをクリックすると、[特殊なアクセス許可] ダイアログ ボックスが開きます。次のテーブルは、[特殊なアクセス許可] ダイアログ ボックスの各領域についての説明です。

テーブル 3-30 • [特殊なアクセス許可] ダイアログ ボックスの各領域

領域	説明
特殊なアクセス許可	このボックスで、設定するアクセス許可のチェック ボックスを選択します。
これらのアクセス許可を子オブジェクトに適用する	フォルダーのアクセス許可を構成するとき、すべてのフォルダーのサブフォルダーとファイルにアクセス許可を適用するには、このチェック ボックスを選択します。 このチェック ボックスは、[一般情報] ビューにある "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションが選択されている場合のみ利用できます。[従来型の Windows Installer 処理] オプションは、この動作をサポートしません。 ***ヒント:*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*

レジストリ キーの [アクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックスで、ロックダウンされた環境で製品を実行するエンドユーザーのために、レジストリ キーを保護するための設定を構成することができます。レジストリ キーのアクセス許可を特定のグループとユーザーに割り当てることができます。たとえば、管理者グループに特定のレジストリ キーについての [読み取り]、[書き込み]、および [削除] アクセス許可を割り当てることができますが、別のグループのすべてにユーザーについては [読み取り] 許可のみ割り当てることができます。

プロジェクトの [一般情報] ビューにある "ロックダウンの設定方法" 設定の選択に従って、InstallShield は ISLockPermissions テーブルまたは LockPermissions テーブルのどちらかにアクセス許可データを追加します。詳細については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

次のテーブルは、[アクセス許可] ダイアログ ボックスの各領域についての説明です。

テーブル 3-31 ・ [アクセス許可] ダイアログ ボックスの各領域

領域	説明
名前	このグリッドで、ドメイン名およびユーザー名を任意に組み合わせて入力できます。エントリを追加するには、グリッド上で右クリックして [新規] をクリックします。同じコンテキストメニューを利用して、エントリを変更または削除することができます。 現在のユーザーのドメインを指定するには、"ドメイン" フィールドに [%USERDOMAIN] を選択します。現在インストールを実行中のユーザーを指定するには、"ユーザー" フィールドに [LogonUser] を選択します。ローカル システム上におけるユーザー アカウントのアクセス許可を設定するには、"ドメイン" フィールドを空白のままにします。"ドメイン" または "ユーザー" フィールドには、実行時に設定される任意の Windows Installer プロパティを入力できます (例、[MYPROPERTY])。 [一般情報] ビューの "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションを選択した場合、"ユーザー" フィールドには、よく使用されるセキュリティ識別子 (SID) の一覧が表示されます。ほとんどの SID は、[従来型の Windows Installer 処理] オプションではサポートされていません。 [カスタム InstallShield 処理] オプションは、"ユーザー" フィールドに一覧されているすべての SID の翻訳された名前をサポートします。従来型のオプションで、非英語システム上で翻訳された名前を使ってアクセス許可を設定すると、インストールが失敗する可能性があります。 ***ヒント・*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*

テーブル 3-31 • [アクセス許可] ダイアログ ボックスの各領域 (続き)

領域	説明
アクセス許可	[名前] 領域で名前を選択し、[アクセス許可] ボックスにあるチェック ボックスを選択またはクリアして、対応するレジストリ キーのアクセス許可を構成します。アクセス許可を選択すると、[詳細] ボタンをクリックして、他の関連するアクセス許可と詳細設定を指定することができます。
アクセス拒否	[アクセス許可] ボックスで選択するアクセス許可を明示的に拒否するには、このチェック ボックスを選択します。 このチェック ボックスは、[一般情報] ビューにある "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションが選択されている場合のみ利用できます。[従来型の Windows Installer 処理] オプションは、この動作をサポートしません。 ***ヒント:*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*

[特殊なアクセス許可] ダイアログ ボックス

[アクセス許可] ダイアログ ボックスで [詳細設定] ボタンをクリックすると、[特殊なアクセス許可] ダイアログ ボックスが開きます。次のテーブルは、[特殊なアクセス許可] ダイアログ ボックスの各領域についての説明です。

テーブル 3-32 • [特殊なアクセス許可] ダイアログ ボックスの各領域

領域	説明
特殊なアクセス許可	このボックスで、設定するアクセス許可のチェック ボックスを選択します。
これらのアクセス許可を子オブジェクトに適用する	レジストリ キーのアクセス許可を構成するとき、すべてのキーのサブキーにアクセス許可を適用するには、このチェック ボックスを選択します。 このチェック ボックスは、[一般情報] ビューにある "ロックダウンの設定方法" 設定で [カスタム InstallShield 処理] オプションが選択されている場合のみ利用できます。[従来型の Windows Installer 処理] オプションは、この動作をサポートしません。 ***ヒント:*** *カスタム InstallShield 処理オプションと、従来型の Windows Installer 処理オプションについての詳細は、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。*

[アイコンの選択] ダイアログ ボックス

[アイコンの選択] ダイアログ ボックスを使うと、.ico ファイル、.dll ファイル、.exe ファイルのアイコンを参照することができます。これによりアイコンファイルの場所とファイルの中のアイコンのインデックスを指定するプロセスが簡単になりました。

タスク ***アイコンを選択するには、以下の手順に従います:***

1. **[ファイル名]** ボックスにファイルへのパスを入力して ENTER を押すか、または **[フォルダーを開く]** をクリックしてファイルを参照し、表示するファイルを探します。ファイルの中のすべてのアイコンがリストに表示されます。
2. アイコンファイルが見つかったら、マウスまたはカーソルキーを使って使用するアイコンを選択することができます。**[大きいアイコンで表示する]** オプションと **[小さいアイコンで表示する]** オプションを使用して、リストのアイコンのプレビューを表示し、標準の 32x32 サイズおよび 16x16 サイズでどのように表示されるか見ることができます。

メモ: *[大きいアイコンで表示する] ラジオ ボタンと [小さいアイコンで表示する] ラジオ ボタンは、選択されたアイコンのインデックスに影響しません。各サイズでアイコンをプレビューする目的でのみ利用されます。*

[INSTALLDIR の設定] ダイアログ ボックス/ [DATABASEDIR の設定] ダイアログ ボックス

このダイアログ ボックスを使用して、ディレクトリの参照、新規作成、名前の変更、または削除を行います。[OK] をクリックすると、使用中のダイアログに応じて、INSTALLDIR または DATABASEDIR に選択されたディレクトリが設定されます。

ダイアログ ボックスの設定

インストール先ディレクトリ

このフィールドには、現在使用できるすべてのインストール先ディレクトリが一覧表示されます。このフィールドで、ディレクトリの選択、 作成 、 名前の変更 、または 削除 を行うことができます。

ディレクトリの選択

タスク ***ディレクトリを選択するには、次の操作を実行します。***

1. 選択するディレクトリをクリックします。
2. [OK] をクリックします。

ディレクトリの新規作成

タスク ***新しいディレクトリを作成するには、次の手順を実行します。***

1. ディレクトリ名を入力します。
2. 必要であれば ディレクトリ識別子 を入力します。

ディレクトリ名の変更

タスク ***ディレクトリの名前を変更するには、次の操作を実行します。***

1. ディレクトリまたは**インストール先コンピューター**を選択して F2 を押すか、右クリックして **[名前の変更]** をクリックします。
2. 新しいディレクトリ名を入力します。定義済みのディレクトリ名は変更できないことにご注意ください。
3. 必要に応じて、ディレクトリ識別子を変更して、ディレクトリの新しい名前と統一させます。

ディレクトリの削除

タスク ***ディレクトリを選択するには、次の操作を実行します。***

ディレクトリを選択して DELETE キーを押すか、ディレクトリを右クリックして **[削除]** をクリックします。定義済みディレクトリは削除できないことにご注意ください。

メモ: *ディレクトリを削除すると、選択したディレクトリの下にあるすべてのサブディレクトリも一緒に削除されます。*

ディレクトリ識別子

[ディレクトリ識別子] フィールドを使用して、使いやすい名前を付けることができます。たとえば、ProgramFilesFolder¥MyProgram¥Graphics¥Jpg というディレクトリがあった場合、ディレクトリ パスを識別するために、**JPG** という名前を使用することができます。InstallShield インターフェイスで、次のようなパスが表示されます。

{JPG} [ProgramFilesFolder]MyProgram¥Graphics¥Jpg

メモ: *ディレクトリ識別子は、有効な MSI 識別子 である必要があります。*

[設定] ダイアログ ボックス

[設定] ダイアログ ボックスでは、InstallShield が現在開いているプロジェクトのインストールをビルドする方法について、その環境を設定することができます。

タスク ***設定ダイアログ ボックスを開くには、以下の手順を実行します。***

[ビルド] メニューで **[設定]** をクリックします。

[設定] ダイアログ ボックスでは次のタブが利用できます:

- [MSI ログ ファイル] タブ (Windows Installer プロジェクトのみ)

[MSI ログファイル] タブ

[設定] ダイアログ ボックスにある [MSI ログ ファイル] タブを使って、InstallShield 内からインストールを実行中に利用可能なログ ファイルを作成することができます。さらに、このタブではログファイルに書き込まれる情報の種類を選択することもできます。

テーブル 3-33 · [MSI ログ ファイル] タブの設定

設定	説明
MSI ログ ファイル オプション	ログファイルで表示するメッセージの種類のチェック ボックスを選択します。チェック ボックスを選択するとき、対応する引数が [MsiExec.EXE コマンドライン引数] ボックスに追加されます。
MsiExec.EXE コマンドライン引数	インストールの実行時に MsiExec.exe に渡すコマンドラインの引数を入力します。チェック ボックスが MSI ログファイルのオプション領域から選択されると、対応するコマンドラインパラメーターが、このボックスに追加されます。
ログファイル	インストール実行時に作成されるログファイルの名前を入力します。

[システム ハードウェア要件] ダイアログ ボックス

[システム ハードウェア要件] ダイアログ ボックスでは、製品のオペレーティング システム要件を設定できます。ターゲット システムが設定したオペレーティング システムの要件を満たさない場合、製品のインストールは許可されません。

このダイアログ ボックスオペレーティング システムの要件を設定するとき、次のいずれかを実行します。

- 製品にオペレーティング システム条件を設定しない場合、**[すべてのオペレーティング システム]** チェック ボックスを選択します。
- サポート対象のオペレーティング システムのリストに制限を加える場合、**[すべてのオペレーティング システム]** チェック ボックスをクリアします。それから、製品でサポートされているオペレーティング システムのチェック ボックスを選択します。

ヒント· *製品のオペレーティング システムの要件を指定すると、実質的に製品をサポートしないオペレーティング システムが除外することになります。*

たとえば、最新 Windows オペレーティング システムのチェック ボックスのみを選択した場合、[要件] ページで選択しなかったオペレーティング システムを除外する起動条件が作成されます。この起動条件タイプでは、将来的にリリースされる Windows オペレーティング システムのバージョンが自動的にサポートされます。これは起動条件で除外されていないためです。

[マージ モジュール検索パスのアップデート] ダイアログ ボックス

このダイアログ ボックスは、**[マージ モジュールを参照]** コマンドを使用して、インストール プロジェクトにマージ モジュールを追加した場合に表示されます。オプション ダイアログでマージ モジュール検索パス (マージ モジュールの場所フィールド) に追加されるパスが表示されます。

[OK] をクリックしてアクションを承認します。

[Express プロジェクト名のアップグレード] ダイアログ ボックス

InstallShield の最新バージョンにインポートする InstallShield Express 2.x プロジェクトを選択した後、作成する新規プロジェクトファイルのパスや名前を入力するプロンプトが表示されます。

完全なファイル名を入力するか、[参照] ボタンをクリックしてファイルを作成するフォルダーに移動します。

ウィザード リファレンス

InstallShield は、インストール プロジェクトの作成を支援する多くのウィザードを装備しています。

プロジェクト: *すべてのウィザードが、すべてのプロジェクトの種類で使用できるわけではありません。*

InstallShield で利用できるウィザードは次の通りです。

- 新規 QuickPatch 作成ウィザード
- DirectX オブジェクト ウィザード
- ダイナミック スキャン ウィザード
- 文字列テーブルのエクスポート ウィザード
- REG ファイルのインポート ウィザード
- 文字列テーブルのインポート ウィザード
- 再配布可能ファイル ダウンローダー ウィザード
- スタティック スキャン ウィザード
- システム検索ウィザード
- Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード
- Visual Studio デプロイメント プロジェクト インポート ウィザード
- Web の配布ウィザード

新規 QuickPatch 作成ウィザード

QuickPatch プロジェクトは、規模の小さいシングル アップデートをエンドユーザーへ配布したいインストール作成者へお勧めします。QuickPatch プロジェクトの作成は [新規 QuickPatch 作成] ウィザードで始めます。

新規 QuickPatch 作成ウィザードには、次の関連パネルがあります。

- ようこそ
- プロジェクト名
- パッチするリリース
- 完了

[ようこそ] パネル

QuickPatch プロジェクトは、規模の小さいシングル アップデートをエンドユーザーへ配布したいセットアップ作成者へお勧めします。新しい QuickPatch の作成ウィザードを実行すると、2 つの種類の配布可能ファイル (*.msp と *.exe ファイル) が作成されます。

QuickPatch プロジェクトの作成は常に、[新規 QuickPatch 作成] ウィザードから始めます。ウィザードに従うと、QuickPatch プロジェクトに必要な基本条件をすべて満たすことができます。

続行するには [次へ] をクリックしてください。

メモ: ウィザードの [キャンセル] または [終了] をクリックすると、QuickPatch プロジェクトは未完成となり、QuickPatch プロジェクトを InstallShield で開くことはできません。

[プロジェクト名] パネル

QuickPatch の作成には、元のインストールまたは Setup.exe ファイルのどの部分を変更あるいはパッチするのかについてセットアップ作成者の知識が重要なので、作成者自身がオリジナル インストールへのパスを指定する必要があります。元のセットアップは新規パッチのベースとなるセットアップです。新規 QuickPatch 作成ウィザードは、このインストールのみを更新することができるパッチを作成します。

このパネルでは次の設定を構成します。

テーブル 3-1 ・ [プロジェクト名]パネルの設定

設定	説明
QuickPatch プロジェクト名	QuickPatch プロジェクトの名前を入力するか、[参照] ボタンをクリックして既存プロジェクトを開きます。場合によって、既存プロジェクトの上書きをするのかどうかを確認するプロンプトが表示されます。
パッチするビルド済み MSI パッケージまたは Setup.exe	ビルド済み MSI パッケージまたは Setup.exe の場所を指定するか、[参照] ボタンをクリックして既存セットアップを指定します。新規 QuickPatch 作成ウィザードは、このセットアップのみを有効にすることができるパッチを作成します。

[パッチのリリース] パネル

このパネルでは、パッチをビルドする元となるプロジェクトのリリースを選択します。ビルド済みのリリース 1 つのみを選択することができます。外部リリースを指定するオプションもあります。このオプションを有効にするため、関連付けられたラジオボタンをクリックしてリリースを参照します。

[完了] パネル

このウィザードパネルは、新規 QuickPatch 作成ウィザードが無事に完了した後に表示されます。このパネルが表示されると、プロジェクトの管理インストールがまもなく起動されます。管理インストールが完了した後、InstallShield は ウィザード パネルに入力された情報を使って QuickPatch プロジェクトを作成します。

DirectX オブジェクト ウィザード

DirectX オブジェクト ウィザードを使って、インストール プロジェクトに Microsoft DirectX 9c 再配布可能ファイルを含み、いくつかのオプションを設定することができます。

メモ: DirectX オブジェクトに含めるファイルについては、「DirectX 9.0 オブジェクトを含める」を参照してください。

DirectX 9c の ManagedDX コンポーネント (オプション) は、.NET Framework 1.1 以降がシステムに インストールされている必要があります。

DirectX オブジェクト ウィザードには、次の関連するウィザードパネルがあります。

- ようこそ
- オブジェクトの設定
- 概要

DirectX についての詳細は、最新の DirectX SDK またはマイクロソフトの MSDN Web サイトを参照してください。

[ようこそ] パネル

DirectX オブジェクト ウィザードを使って、インストール プロジェクトに Microsoft DirectX 9c 再配布可能ファイルを含み、いくつかのオプションを設定することができます。

メモ: DirectX オブジェクトに含めるファイルについては、「DirectX 9.0 オブジェクトを含める」を参照してください。

DirectX 9c の ManagedDX コンポーネント (オプション) は、.NET Framework 1.1 以降がシステムに インストールされている必要があります。

DirectX についての詳細は、最新の DirectX SDK またはマイクロソフトの MSDN Web サイトを参照してください。

[オブジェクトの設定] パネル

[オブジェクト設定] パネルでは、次のオプションを設定することができます:

テーブル 3-2 · [オブジェクトの設定] パネルのオプション

オプション	説明
DirectX ファイルを Disk1 フォルダー内のフォルダーに配置する	このチェック ボックスを選択すると、ビルド時に作成中のインストールに対して DirectX フォルダーが作成され、リリースの Disk1 フォルダーに配置されます。 このチェック ボックスをクリアすると、DirectX 再配布可能ファイルが .msi ファイルにストリームされます。単一実行可能インストールを作成している場合、ファイルが .msi ファイルに含められるようにこのチェック ボックスをクリアにします。 オブジェクトに含める DirectX ファイルについては、「DirectX 9.0 オブジェクトを含める」を参照してください。

テーブル 3-2 • [オブジェクトの設定] パネルのオプション (続き)

オプション	説明
DirectX 9 のインストールを開始する前に Microsoft DirectX EULA を表示する	デフォルトでは、Microsoft DirectX EULA が DirectX 9 のインストールが始まる前にエンド ユーザーに対して表示されます。EULA が表示されないようにするには、このチェック ボックスをクリアします。

DirectX についての詳細は、最新の DirectX SDK またはマイクロソフトの MSDN Web サイトを参照してください。

[概要] パネル

[概要] パネルでは、[オブジェクトの設定] パネル で構成したオプションを確認することができます。設定を使用して、プロジェクトに DirectX 9 再配布可能ファイルを追加するには [完了] をクリックしてください。

ダイナミック スキャン ウィザード

ダイナミック スキャン ウィザードは、実行可能ファイルの実行中にシステムを監視する使い易いツールです。ウィザードでは実行可能ファイルが必要とする可能性のある .dll および .ocx ファイルのリストが表示され、それぞれをプロジェクトに含めるかどうかを指定できます。ウィザードを使ってプロジェクトに既に含まれている実行可能ファイルをスキャンするか、スキャン処理を開始する前にウィザードの [実行可能ファイルの指定] パネルを使ってスキャンする新しい実行可能ファイルを選択してプロジェクトに追加することができます。

タスク ***ダイナミック スキャン ウィザードを起動するには、次の手順を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[依存関係]** をクリックします。
2. **[ダイナミック スキャンの実行]** ボタンをクリックします。

ダイナミック スキャン ウィザードには、次のような関連パネルがあります。

- ようこそ
- ファイルのフィルター
- 実行プログラムの指定
- アプリケーション ファイルの指定
- アプリケーションの起動
- アプリケーションの実行中
- ファイルの選択
- スキャン結果
- ダイナミック スキャン ウィザードの完了

[ようこそ] パネル

ダイナミック スキャン ウィザードを使って、アプリケーションの依存ファイルを簡単にプロジェクトに追加できます。スキャナーを使用する前に、対象の実行可能ファイルをプロジェクトに追加することをお勧めします。

ウィザードの使用を開始するには、[次へ] をクリックします。

[ファイルのフィルター] パネル

ダイナミック スキャン ウィザードは、インストールに追加する必要のないファイルを依存関係としてリストする場合があります。例えば、ターゲット マシン上に既存する一般的なシステムファイルは通常、再インストールする必要がありません。スキャナーを実行した時にこれらのファイルがプロジェクトに追加されることを回避するには、[ファイルのフィルター] パネルで **[ファイルのフィルター]** チェック ボックスを選択します。

スキャンから除外するファイル リストをカスタマイズする方法については、「依存関係スキャナーでファイルをフィルターする」を参照してください。

[実行可能ファイルの指定] パネル

[実行可能ファイルの指定] パネルでは、プロジェクトに既に含まれている実行可能ファイル、またはまだ追加されていない実行可能ファイルをスキャンするかどうかを選択します。

テーブル 3-3 · [実行可能ファイルの指定] パネルの設定

設定	説明
マイプロジェクトから実行ファイルを選択する	スキャンする実行可能ファイルがプロジェクトに既に追加されている場合は、このオプションを選択します。
新しい実行可能ファイルを選択する	スキャンする実行可能ファイルが現在プロジェクトに含まれていない場合は、このオプションを選択します。

[アプリケーション ファイルの指定] パネル

[アプリケーション ファイルの指定] パネルでは、スキャンする特定の実行可能ファイル (.exe) を選択します。さらに、ファイルや作業フォルダーのコマンドラインパラメーターを指定することができます。

テーブル 3-4 · [アプリケーション ファイルの指定] パネルの設定

設定	説明
アプリケーション	スキャンする実行可能ファイルを指定します。プロジェクトに既に含まれているファイルをスキャンする場合は、プロジェクトに既存する実行可能ファイルのリストから選択します。まだプロジェクトの一部になっていないファイルをスキャンする場合は、ファイルへのパスを入力するか、[参照] ボタンをクリックしてファイルへ移動します。

テーブル 3-4 • [アプリケーション ファイルの指定] パネルの設定 (続き)

設定	説明
コマンド ライン	実行可能ファイルに渡すコマンドライン パラメーターを入力します。これらのパラメーターはスキャン処理中のみ使用され、ウィザードの終了後は使用されません。
作業用フォルダー	このアプリケーションに対する作業フォルダーへのパスを入力するか、[参照] ボタンをクリックしてディレクトリに移動します。デフォルトでは、ディレクトリは、スキャンするアプリケーションが格納されているフォルダーと同じフォルダーに設定されています。

アプリケーションの起動

ダイナミック スキャン ウィザードによるアプリケーションのスキャンを実行する前に、該当するアプリケーションの起動が必要です。ウィザードによるアプリケーションの起動後、アプリケーションのメニュー項目や機能が使用可能になっているか、できるだけ多くを試してみてください。これは、依存関係ファイルの場所を見つけてプロジェクトに追加するのに役立ちます。

[次へ] をクリックして、アプリケーションを起動し、依存関係のスキャンを開始します。

[アプリケーションが実行中です] パネル

[アプリケーションが実行中です] パネルはアプリケーションを実行している間、表示されます。アプリケーションを終了してから、[完了] ボタンをクリッすると、スキャン結果が表示されます。スキャン結果を確認しなければ、ファイルは追加されません。

[ファイル選択] パネル

[ファイルの選択] パネルには、プロジェクトに追加する必要がある可能性の高いファイルおよびマージ モジュールのリストが表示されます。このパネルを使って、インストールに含めるファイルおよびマージ モジュールを選択できます。詳細については、「依存関係スキャナー結果の確認」を参照してください。

テーブル 3-5 • [ファイル選択] パネルの設定

設定	説明
File	該当するチェック ボックスを選択して、プロジェクトに追加するファイルおよびマージ モジュールを指定します。
すべて選択解除	すべてのチェック ボックスをクリアする場合は、このボタンをクリックします。その後、プロジェクトに追加するファイルまたはモジュールに対応する各チェック ボックスを手作業で選択できます。
すべて選択	すべてのチェック ボックスを選択する場合は、このボタンをクリックします。その後、プロジェクトに追加するファイルまたはモジュールに対応する各チェック ボックスを手作業でクリアできます。

[スキャン結果] パネル

[スキャン結果] パネルには、ウィザードが識別した依存関係の中からプロジェクトに追加することが選択されたファイルが表示されます。

これらの依存関係をプロジェクトに追加するには、[次へ] ボタンをクリックします。依存関係を追加しないでウィザードを終了するには、[キャンセル] ボタンをクリックします。再び依存関係の可能性があるファイルのリストを表示して、それらを追加または削除するには、[戻る] ボタンをクリックします。

[ダイナミック スキャン ウィザード完了] パネル

ダイナミック スキャン ウィザードが [ダイナミック スキャン ウィザードの完了] パネルを表示する段階で、ウィザードは実行可能ファイルの依存関係をプロジェクトに追加済みです。まだプロジェクトに含まれていない実行可能ファイルのスキャンを選択した場合、その .exe ファイルも追加されます。

[完了] をクリックすると、ウィザードを閉じて InstallShield に戻ります。

文字列テーブルのエクスポート ウィザード

[文字列テーブルをエクスポート] ウィザードを使うと、文字列テーブルをテキスト (.txt) ファイルにエクスポートして、翻訳してから InstallShield にインポートし直すことができます。[文字列テーブルのエクスポート] ウィザードには、次のような関連パネルがあります。

- ようこそ
- ファイル名
- ウィザードの終了

[ようこそ] パネル

セットアップのランタイム文字列を翻訳する最も簡単な方法は、テキスト (.txt) ファイルにエクスポートし、翻訳することです。翻訳が完了したら、インストールのローカライズされたバージョンのセットアッププロジェクトに、これらの文字列を再びインポートすることができます。文字列テーブルのエクスポートウィザードは、テキストファイルへの文字列すべてのエクスポートの手順を示します。

[ファイル名] パネル

このパネルでは、文字列すべてのエクスポート先となるテキストファイルの場所およびファイル名を指定できます。

テーブル 3-6 ・ [ファイル名]パネルの設定

設定	説明
ファイル名	文字列すべてのエクスポート先となるテキストファイルのパスおよびファイル名を入力するか、または [参照] ボタンをクリックしてこのファイルを指定します。

[ウィザードの完了] パネル

この時点で、直前のパネルで指定したファイルに、文字列のすべてが送られています。[キャンセル] をクリックして変更を破棄するか、[完了] をクリックして変更を保存し IDE に戻ります。

REG ファイルのインポート ウィザード

REG ファイルのインポート ウィザードを利用して、既存のレジストリデータ (.reg) を InstallShield プロジェクトにインポートすることができます。このレジストリデータは、セットアッププロジェクト中にターゲット システムのレジストリに追加されます。

レジストリファイルインポートウィザードには、次のような関連パネルがあります。

- ようこそ
- レジストリ ファイルのインポート
- 競合オプションのインポート
- インポート処理状況

[ようこそ] パネル

REG ファイルのインポート ウィザードを利用して、既存のレジストリデータ (.reg) を InstallShield プロジェクトにインポートすることができます。このレジストリデータは、セットアッププロジェクト中にターゲット システムのレジストリに追加されます。

メモ: *レジストリ データをインポートする前に、このデータの追加先として正しい機能が選択されているか確認してください。機能を指定するには、ウィザードをキャンセルして、[レジストリ] ビューの [機能] リストから該当する機能を選択します。*

[次へ] をクリックすると、.reg ファイルのインポートを開始します。

注意: *InstallShield は、Regedit のエクスポートによって作成された .reg ファイル、またはそのフォーマットに厳密に従ったファイルからのみインポートすることができます。また、InstallShieldは複数行のレジストリ値もサポートしていません。*

[レジストリ ファイルのインポート] パネル

このパネルでは、インポートする .reg ファイルを指定できます。

テーブル 3-7 • [レジストリ ファイルのインポート] パネルの設定

プロパティ	説明
レジストリ ファイル	インポートする .reg ファイルのパスを入力するか、[参照] ボタンをクリックしてこのファイルに移動します。 ***注意***• *InstallShield は、Regedit のエクスポートによって作成された .reg ファイル、またはそのフォーマットに厳密に従ったファイルからのみインポートすることができます。また、InstallShieldは複数行のレジストリ値もサポートしていません。*

[競合オプションのインポート] パネル

インポートする .reg ファイルに、セットアップのレジストリ データと競合する設定が含まれている場合は、競合をどのように解決するかを選択できます。

テーブル 3-8 • [競合オプションのインポート]パネルの設定

設定	説明
レジストリ データを上書きする	インポートしている REG ファイル内のデータで、すでにインストール プロジェクトにある競合データを上書きする場合、このオプションを選択します。
レジストリ データを上書きしない	インポート中に競合が発生した場合に、すでにインストール プロジェクトにあるデータを保持する場合、このオプションを選択します。競合しないレジストリデータがすべてインポートされます。

[インポート進行状況] パネル

このパネルには、REG ファイルのインポートの進行状況が表示されます。[キャンセル] をクリックしてインポートを中止するか、ウィザードにより REG ファイルのインポートが終了するまで待ち、[完了] をクリックし IDE に戻ります。

文字列テーブルのインポート ウィザード

文字列テーブルのインポートウィザードを使用して、InstallShield に文字列テーブルをインポートします。このウィザードは、翻訳した文字列を InstallShield に再びインポートするタスクの手順を説明します。[文字列テーブルのインポート] ウィザードには、次のような関連パネルがあります。

- ようこそ

- ファイル名
- ウィザードの終了

[ようこそ] パネル

セットアップのローカライズ版を作成している場合、ランタイム文字列のすべてを変換用にエクスポートする方法がとても簡単です。文字列は一度変換すると、セットアッププロジェクトにインポートできます。文字列テーブルのインポートウィザードでは、インポートするすべての変換済みランタイム文字列を含むテキスト (.txt) ファイルを指定できます。

文字列テーブルをインポートする場合、既存の文字列のすべては、インポートしている文字列により上書きされます。したがって、変換バージョンをインポートする前に、オリジナルの文字列のコピーを必ず作成してください。

[ファイル名] パネル

このダイアログでは、インポートする特定のテキスト (.txt) ファイルを選択できます。

テーブル 3-9 · [ファイル名]パネルの設定

設定	説明
ファイル名	インポートする文字列を含むテキスト (.txt) ファイルのパスおよびファイル名を入力するか、または [参照] ボタンをクリックしてこのファイルに移動します。

[次へ] ボタンをクリックして、これらの文字列をプロジェクトにインポートします。既存の文字列は上書きされます。

[ウィザードの完了] パネル

この時点で、指定したテキストファイル内に含まれる文字列は、セットアップ プロジェクトにインポートされています。変更を保存せずにウィザードを終了する場合、[キャンセル] ボタンをクリックします。[完了] をクリックすると、変更を保存して IDE に戻ります。

再配布可能ファイル ダウンローダー ウィザード

再配布可能ファイル ダウンローダー ウィザードを利用して、サードパーティ再配布可能ファイル、マージ モジュールモジュール、前提条件、その他のファイルをローカルマシンに素早くダウンロードすることができます。再配布可能ファイル ダウンローダ ウィザードは、リリース ウィザードの [.NET ランタイム オプション] パネル、または [リリース] ビュー (".NET 1.1/2.0 コア言語" 設定および ".NET 1.1/2.0 言語パック" 設定) から起動することができます。

タスク ***再配布可能ファイル ダウンローダー ウィザードを利用するには、以下の手順に従います:***

1. **[ダウンロードするファイルの選択]** パネルで、ダウンロードする再配布可能ファイル/前提条件を選択し、**[次へ]** をクリックして **[進行状況]** パネルへ移動します。
2. 進行状況バーが完了したら、**[次へ]** をクリックします。

最終パネルには、ウィザードがマシンへダウンロードした再配布可能ファイル/前提条件のリストが表示されます。**[完了]** をクリックしてウィザードを終了します。

スタティック スキャン ウィザード

スタティック スキャン ウィザードを利用して、プロジェクト含まれているファイルをスキャンして必要な依存関係があるかどうかをチェックすることができます。このウィザードはプロジェクトにあるすべての .exe、.dll、.ocx、.sys、.com、.drv、.scr、および .cpl ファイルをスキャンし、検出した依存関係をインストールに追加します。

プロジェクトに追加された新規ファイルは、ファイルが依存している同じ機能に追加されるので、インストールが必要なときに確実に実行されることを確認します。

タスク ***スタティック スキャン ウィザードを起動するには、次の操作を実行します。***

1. **[アプリケーション データの指定]** の下にあるビュー リストで、**[依存関係]** をクリックします。
2. **[スタティック スキャンの実行]**ボタンをクリックします。

スタティック スキャン ウィザードには、次のような関連パネルがあります:

- ようこそ
- ファイルのフィルター
- スキャンの進行状況
- ファイルの選択
- スキャン結果
- スタティック スキャン ウィザードの終了

[ようこそ] パネル

スタティック スキャン ウィザードを利用して、プロジェクト含まれているファイルをスキャンして必要な依存関係があるかどうかをチェックすることができます。このウィザードはプロジェクトにあるすべての .exe、.dll、.ocx、.sys、.com、.drv、.scr、および .cpl ファイルをスキャンし、検出した依存関係をインストールに追加します。

[次へ] ボタンをクリックすると、プロジェクトに含まれるファイルの依存関係を調べるスキャンを開始します。

[ファイルのフィルター] パネル

スタティック スキャン ウィザードは、インストールに追加する必要のないファイルを依存関係としてリストする場合があります。例えば、ターゲット マシン上に既存する一般的なシステムファイルは通常、再インストールする必要がありません。スキャナーを実行した時にこれらのファイルがインストールに追加されることを回避するには、[ファイルのフィルター] パネルで **[ファイルのフィルター]** を選択します。

スキャンから除外するファイル リストをカスタマイズする方法については、「依存関係スキャナーでファイルをフィルターする」を参照してください。

[スキャンの進行状況] パネル

スタティック スキャン ウィザードがプロジェクトに含まれるすべての .exe、.dll、.ocx、.sys、.com、.drv、.scr、および .cpl ファイルをスキャンして依存関係を調べている最中に [スキャンの進行状況] パネルが表示されます。スキャン結果を確認しなければ、ファイルは追加されません。

[ファイル選択] パネル

[ファイルの選択] パネルには、プロジェクトに追加する必要がある可能性の高いファイルおよびマージ モジュールのリストが表示されます。このパネルを使って、インストールに含めるファイルおよびマージ モジュールを選択できます。詳細については、「依存関係スキャナー結果の確認」を参照してください。

テーブル 3-10 ・ [ファイル選択] パネルの設定

設定	説明
File	該当するチェック ボックスを選択して、インストールに追加するファイルを指定します。各種アイコンが示す意味は次のとおりです。 ・ : このアイコンは、ファイルがシステムまたはドライバー ファイルであることを示します。このタイプのファイルは通常インストールの一部として再配布されるものではなく、インストール先コンピューターの実行不能の原因となる可能性があります。これらのファイルが必要かどうか、含める前に確認してください。 ・ : このアイコンは、システムファイルではないことを示します。インストールにこのファイルを含めるには、アイコンの左側のチェック ボックスを選択してください。 ・ : このアイコンはマージ モジュールを示します。指定のマージ モジュールをプロジェクトに追加するには、アイコンの左側のチェック ボックスを選択します。 ・ : これは、インストール プロジェクトの .NET アセンブリのアイコンです。 ・ : このアイコンは、スタティック スキャナーが検出したものの、既にプロジェクトに含まれているファイルを識別します。

テーブル 3-10 • [ファイル選択] パネルの設定 (続き)

設定	説明
すべて選択解除	すべてのチェック ボックスをクリアする場合は、このボタンをクリックします。その後、プロジェクトに追加するファイルまたはモジュールに対応する各チェック ボックスを手作業で選択できます。
すべて選択	すべてのチェック ボックスを選択する場合は、このボタンをクリックします。その後、プロジェクトに追加するファイルまたはモジュールに対応する各チェック ボックスを手作業でクリアできます。

[スキャン結果] パネル

[スキャン結果] パネルには、ウィザードが識別した依存関係の中からプロジェクトに追加することが選択されたファイルが表示されます。

これらの依存関係をプロジェクトに追加するには、[次へ] ボタンをクリックします。依存関係を追加しないでウィザードを終了するには、[キャンセル] ボタンをクリックします。再び依存関係の可能性があるファイルのリストを表示して、それらを追加または削除するには、[戻る] ボタンをクリックします。

[スタティック スキャン ウィザードの完了] パネル

スタティック スキャン ウィザードが [スタティック スキャン ウィザードの完了] パネルを表示する段階で、ウィザードは選択された依存関係をプロジェクトに追加済みです。

[完了] をクリックすると、ウィザードを閉じて InstallShield に戻ります。

システム検索ウィザード

InstallShield の [要件] ビューでは、インストール プロジェクトのターゲット システム要件を指定できます。システム ソフトウェア要件を指定するとき、システム検索ウィザードを実行しなくてはなりません。[システム検索ウィザード] は、インストール前にターゲットのシステム上にある特定のファイル、フォルダー、レジストリ キーまたは .ini 値を探す Windows Installer 機能を提供します。

タスク ***システム検索ウィザードが起動するには、次の手順を実行します。***

1. ビュー リストの **[セットアップ要件およびアクションの定義]** の下にある **[要件]** をクリックします。
2. エクスプローラーで、項目を右クリックしてから **[新規起動条件の作成] (システム検索ウィザード)]** を選択します。

システム検索ウィザードには、以下のパネルがあります。

- ようこそ
- 検索する対象を指定してください
- 検索方法

- この値の処理方法を指定してください

[ようこそ] パネル

システム検索ウィザードの [ようこそ] パネルこのウィザードを使って、プロジェクトにシステム検索を追加または変更することができます。システム検索を追加または変更するには [次へ] をクリックします。

検索する対象を指定してくださいパネル

システム検索ウィザードの **[検索内容]** パネルを使って、検索するアイテムの種類、およびターゲット システム内で検索する場所を指定します。選択可能なオプションは以下のとおりです:

- ファイル パス(フォルダーを検索)
- フォルダーパス、全てのドライブを検索する
- フォルダー パス、特定のフォルダーを検索する
- フォルダー パス(特定のファイルを検索)
- ファイル パス(レジストリ エントリで指定)
- フォルダー パス(レジストリ エントリで指定)
- レジストリ エントリ
- ファイル パス(.ini ファイル値で指定)
- フォルダー パス(.ini ファイル値で指定)
- .ini ファイル値

検索方法パネル (システム検索メソッドの定義)

[検索内容] パネルには、検索をカスタマイズするための設定があります。このパネルの設定は、前のパネルで選択した検索の種類によって異なります。

テーブル 3-11 • 検索をカスタマイズする

検索の種類	構成が必要な対応する設定
ファイル パス(フォルダーを検索)	ファイルの詳細とその検索場所を指定します。 詳細については、「[ファイルの詳細の指定] パネル (ファイル検索オプション)」を参照してください。
フォルダーパス、全てのドライブを検索する	検索するフォルダー名およびサブフォルダーの階層数を指定します。 詳細については、「検索方法パネル (フォルダー検索オプション)」を参照してください。

テーブル 3-11・検索をカスタマイズする (続き)

検索の種類	構成が必要な対応する設定
フォルダー パス、特定のフォルダーを検索する	フォルダー名と、それをターゲット システム上で探す場所を指定します。 詳細については、「検索方法パネル(特定のフォルダーオプション)」を参照してください。
フォルダー パス(特定のファイルを検索)	ファイルの詳細とその検索場所を指定します。 詳細については、「[ファイルの詳細の指定] パネル (ファイル検索オプション)」を参照してください。
ファイル パス(レジストリ エントリで指定)	レジストリ エントリの詳細を指定します。 詳細については、「検索方法パネル (レジストリ検索オプション)」を参照してください。
フォルダー パス(レジストリ エントリで指定)	レジストリ エントリの詳細を指定します。 詳細については、「検索方法パネル (レジストリ検索オプション)」を参照してください。
レジストリ エントリ	レジストリ エントリの詳細を指定します。 詳細については、「検索方法パネル (レジストリ検索オプション)」を参照してください。
ファイル パス(.ini ファイル値で指定)	.ini ファイル値についての詳細を指定します。 詳細については、「検索方法パネル (.ini ファイル検索 オプション)」を参照してください。
フォルダー パス(.ini ファイル値で指定)	.ini ファイル値についての詳細を指定します。 詳細については、「検索方法パネル (.ini ファイル検索 オプション)」を参照してください。
.ini ファイル値	.ini ファイル値についての詳細を指定します。 詳細については、「検索方法パネル (.ini ファイル検索 オプション)」を参照してください。

[ファイルの詳細の指定] パネル (ファイル検索オプション)

[ファイル パス(フォルダーを検索)] タイプのシステム検索を構成する場合、または **[フォルダー パス(特定のファイルを検索)]** タイプの検索を構成する場合、**[ファイルの詳細とその検索場所を指定]** パネルには、次の設定が表示されます:

テーブル 3-12 • [ファイルの詳細とその検索場所を指定] パネルの設定

設定	説明
ファイル名	探すファイルまたはアプリケーションの完全名と拡張子を入力します。 ***メモ:*** *ファイル名を入力すると、[詳細] ボタンが有効化されます。特定のバージョン、作成日、サイズ、または言語を含む検索に限定したい場合は、このボタンをクリックします。詳細については、「[ファイルの詳細] ダイアログ ボックス」を参照してください。*
検索先	ターゲット システム上でWindows Installer を検索する場所を指定します。
検索するサブフォルダーの最大数	ターゲット システム上で検索するファイルのサブ フォルダーの最大数を指定します。

検索方法パネル (フォルダー検索オプション)

[フォルダー パス(すべてのドライバーを検索)] タイプのシステム検索を構成する場合、**[検索方法]** パネルには、次の設定が表示されます:

テーブル 3-13 • 検索方法パネルの設定

設定	説明
フォルダー名	Windows Installer が検索するフォルダーの完全名と拡張子を入力します。
検索先	ターゲット システム上でWindows Installer を検索する場所を指定します。
検索するサブフォルダー数	ターゲット システム上で Windows Installer が検索するサブフォルダーの階層数を指定します。

検索方法パネル(特定のフォルダーオプション)

[フォルダー パス(特定のフォルダーを検索)] タイプのシステム検索を構成する場合、**[検索方法]** パネルには、次の設定が表示されます:

テーブル 3-14 • 検索方法パネルの設定

設定	説明
フォルダー名	検索するフォルダーの完全名と拡張子を入力します。

テーブル 3-14 • 検索方法パネルの設定 (続き)

設定	説明
検索先	ターゲット システム上でWindows Installer を検索する場所を指定します。完全パスを指定するか、前回の検索で使用したパスを選択できます。 完全パスを指定するには、既存ディレクトリを選択して [参照] ボタンをクリックするか、新しいパスを作成します。 前回の検索からのパスを指定する場合、一覧から選択します。リストが空白の場合、プロジェクトにその他の検索が含まれていないことを示します。
検索するサブフォルダー数	ターゲット システム上で Windows Installer が検索するサブフォルダーの階層数を指定します。

検索方法パネル (レジストリ検索オプション)

[**フォルダー パス(レジストリ エントリで指定)**] タイプのシステム検索、または**レジストリ エントリ** タイプのシステム検索を構成する場合、[**検索方法**] パネルに次の設定が表示されます:

テーブル 3-15 • 検索方法パネルの設定

設定	説明
レジストリ ルート	ターゲット システム上で Windows Installer が検索するレジストリ ルートを指定します。
レジストリ キー	検索する項目と関連したレジストリキーを正確に入力します。例えば、HKEY_LOCAL_MACHINE¥SOFTWARE¥Adobe¥Acrobat をターゲット システムで検索したい場合、この設定に**SOFTWARE¥Adobe¥Acrobat** のように入力します。 ***ヒント:*** *Windows レジストリ エディターから正しいキー名をコピーして構文が正しいことを確認してください。*
レジストリ値 (オプション)	特定のレジストリ値を検索するには、Windows レジストリ エディターに表示されるレジストリ値を正確に入力します。 ***メモ:*** *この設定が空白の場合、システム検索はレジストリ キーのデフォルト値を探します。*

テーブル 3-15・検索方法パネルの設定（続き）

設定	説明
レジストリの 64 ビットの部分を検索	64 ビット ターゲット システムは、通常 2 つの HKEY_LOCAL_MACHINE¥Software キーを持ちます: ・ HKLM¥Software – 64 ビット アプリケーション用 ・ HKLM¥Software¥Wow6432Node – 32 ビット アプリケーション用 Windows Instaler が 64 ビット ターゲット システム上でレジストリの 64 ビット 部分をチェックする場合は、このチェックボックスを選択します。

検索方法パネル（.ini ファイル検索 オプション）

[**ファイル パス(.ini ファイル値で指定)**] タイプのシステム検索、または.**ini ファイル値** タイプのシステム検索を構成する場合、[**検索方法**] パネルに次の設定が表示されます:

テーブル 3-16・検索方法パネルの設定

設定	説明
INI ファイル名	.ini ファイル名をターゲット システムに表示されるとおりに指定します。このファイルは、Windows フォルダーに既存するファイルです。
INI セクション名	Windows Installer で検索する値を含む .ini ファイルのセクション名を指定します。
INI キー名	Windows Installer で検索する値を含むキーの名前を指定します。
行全体を読み込む	Windows Installer が .ini ファイルの行全体を読み込む場合は、このチェック ボックスを選択します。

この値の処理方法を指定してくださいパネル

[**値の処理方法**] パネルで、検索条件が検出された場合、または検出されなかった場合にインストールを続行するかどうかを指定します。

このパネルでは、検索条件に合致しなかったためにインストールが中断された場合に表示するテキストを入力することもできます。

Visual Basic .NET、Visual C++ .NET および C# .NET 用の Visual Studio .NET ウィザード

Visual Studio .NET ウィザードは新しい InstallShield インストール プロジェクトを作成して、それを Microsoft Visual Studio .NET ソリューションへ追加します。

メモ: *Visual Studio .NET ウィザードは、システムに Microsoft Visual Studio .NET がインストールされている場合のみ使用可能です。*

InstallShield インストール プロジェクトを作成中の場合、Visual Studio .NET ウィザードは以下の処理を行います。

- 新しい InstallShield プロジェクトを作成して ([新規プロジェクト] ダイアログ ボックスで指定されたファイル名) 、ソリューション (.sln ファイル) に追加します。
- [ビルド時にスキャン] オプションが [依存関係とプロパティ] ([オプション] ダイアログ ボックスの .NET タブ)に設定されている場合、ビルド時にすべての依存関係をプロジェクトへ追加します。
- ソリューションにある各プロジェクトから InstallShield プロジェクトにプライマリ出力を追加します。
- リリース設定をアップデートしてダウンロードで .NET Framework の適切なバージョンを配布します。

ウィザードを起動するには、[新規プロジェクト] ダイアログ ボックス の該当するアイコンをクリックします。

[ようこそ] パネル

Visual Studio .NET ウィザード は、InstallShield インストール プロジェクトを Microsoft Visual Studio .NET ソリューションに追加します。

Visual Studio .NET ウィザードは、システムに Microsoft Visual Studio .NET がインストールされている場合のみ使用可能です。

[次へ] をクリックしてソリューションを選択してください。

[ソリューション] パネル

[ソリューション] パネルでは、InstallShield インストール プロジェクトを追加する Visual Studio .NET ソリューションへのパスを入力または参照します。

[完了] をクリックして、InstallShield プロジェクトを作成し、選択済みのソリューションを追加します。

Visual Studio デプロイメント プロジェクト インポート ウィザード

Visual Studio デプロイメント プロジェクト インポート ウィザードを使って、Visual Studio セットアップまたはマージ モジュール プロジェクト (.vdproj) を InstallShield プロジェクト (.ise) にインポートすることができます。複数の Visual Studio プロジェクトを InstallShield プロジェクトにインポートする場合、このウィザードを繰り返し使用することができます。

重要: *InstallShield プロジェクトにインポートする Visual Studio セットアップ またはマージ モジュール プロジェクトに 1 つ以上のプロジェクト出力が含まれている場合、その InstallShield は、Visual Studio セットアップまたはマージ モジュール プロジェクトおよびそのプロジェクトのすべての依存関係を含む、同じ Visual Studio ソリューションに含まれていなくてはなりません。*

タスク *Visual Studio デプロイメント プロジェクト インポート ウィザードを起動するには、以下の手順の 1 つを行います:*

- InstallShield を Visual Studio に統合しないで使用している場合: [**プロジェクト**] メニューから [**Visual Studio デプロイメント プロジェクト インポート ウィザード**] をクリックします。
- InstallShield を Visual Studio 内部から使用している場合: InstallShield ツールバーから、[**Visual Studio デプロイメント プロジェクト インポート ウィザード**] をクリックします。

このウィザードには、以下のパネルがあります:

- ようこそ
- プロジェクト ファイル
- オプション
- 概要

[ようこそ] パネル

Visual Studio デプロイメント プロジェクト インポート ウィザードを使って、Visual Studio セットアップまたはマージ モジュール プロジェクト (.vdproj) を InstallShield プロジェクト (.ise) にインポートすることができます。複数の Visual Studio プロジェクトを InstallShield プロジェクトにインポートする場合、このウィザードを繰り返し使用することができます。

重要: *InstallShield プロジェクトにインポートする Visual Studio セットアップ またはマージ モジュール プロジェクトに 1 つ以上のプロジェクト出力が含まれている場合、その InstallShield は、Visual Studio セットアップまたはマージ モジュール プロジェクトおよびそのプロジェクトのすべての依存関係を含む、同じ Visual Studio ソリューションに含まれていなくてはなりません。*

ウィザードの使用を開始するには、[次へ] をクリックします。

[プロジェクトファイル] パネル

[プロジェクト ファイル] パネルを使って、インポートする Visual Studio プロジェクト (.vdproj) を指定します。プロジェクトには、セットアップ プロジェクトまたはマージ モジュール プロジェクトを選択できます。

[オプション] パネル

Visual Studio デプロイメント プロジェクト インポート ウィザードを使って、プロジェクト出力、ファイル、レジストリ キー、ファイル拡張子、カスタム アクション、ターゲット システム検索、および前提条件を Visual Studio プロジェクトから InstallShield プロジェクトにインポートできます。[オプション] パネルでは、適切な場合、どのプロパティを Visual Studio プロジェクトから InstallShield プロジェクトにインポートするのかも選択できます。

メモ: *オプションのチェック ボックスを選択すると、InstallShield プロジェクト内の既存の値が Visual Studio プロジェクトで構成された値で上書きされます。たとえば、[製品名] チェック ボックスを選択すると、InstallShield プロジェクトの "製品名" 設定の値が Visual Studio プロジェクトで設定された値で上書きされます。*

テーブル 3-17 ・ インポートで使用できるオプション

オプション	説明
製品名	インポートしている Visual Studio プロジェクトで構成された ProductName プロパティを InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 製品名は、InstallShield の [一般情報] ビューで構成されます。 ***プロジェクト:*** *このプロパティは、マージ モジュール プロジェクトには適用しません。*
製品バージョン	インポートしている Visual Studio プロジェクトで構成された ProductVersion プロパティの値を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 製品バージョンは、InstallShield の [一般情報] ビューで構成されます。

テーブル 3-17・インポートで使用できるオプション (続き)

オプション	説明
INSTALLDIR	インポートしている Visual Studio プロジェクトのアプリケーション ファオルダに構成された DefaultLocation プロパティの値を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 このチェック ボックスを選択すると、InstallShield は [一般情報] ビューの INSTALLDIR 設定の値を、Visual Studio プロジェクトで構成されたパスでアップデートします。 ***メモ:*** *Visual Studio では、アプリケーション フォルダーに複数のフォーマットされたプロパティが含まれたディレクトリ パス (例、[ProgramFilesFolder][Manufacturer]¥[ProductName]) を指定できます。Visual Studio プロジェクトは、実行時にディレクトリ カスタム アクションを使用して、パスを解決します。ただし、InstallShield では、この種類のディレクトリ パスはサポートされていません。したがって、InstallShield はパスを変換処理中に解決し、パスの* INSTALLDIR *プロパティを使用します。* ***エディション:*** *InstallShield Premier および InstallShield には 64 ビット サポートが含まれていますが、Express Edition には含まれていません。* *Visual Studio プロジェクトの DefaultLocation プロパティが* **[ProgramFiles64Folder]** *のような 64 ビットの場所を使用する場合で、Visual Studio デプロイメント プロジェクト インポート ウィザードのこのパネルで INSTALLDIR チェック ボックスを選択すると、InstallShield Express Edition はそのフォルダーに対応する 32 ビットの場所を使用します (例、* **[ProgramFilesFolder]***)。InstallShield Premier および InstallShield は 64 ビットの場所を使用します。*

テーブル 3-17・インポートで使用できるオプション (続き)

オプション	説明
[プログラムの追加と削除] のプロパティ	Visual Studio プロジェクトで構成された"プログラムの追加と削除" プロパティ (AddRemoveProgramsIcon、Manufacturer、Description、ManufacturerUrl、Author、SupportUrl、および SupportPhone) を InstallShield プロジェクトで使用する場合は、このチェック ボックスを選択します。 このチェック ボックスを選択すると、InstallShield は [一般情報] ビューにある次の設定の値を、Visual Studio プロジェクトで構成された値でアップデートします。 ・ 表示アイコン (Visual Studio プロジェクトでは、AddRemoveProgramsIcon プロパティ) ・ 発行者 (Visual Studio プロジェクトでは、Manufacturer プロパティ) ・ [プログラムの追加と削除] のコメント (Visual Studio プロジェクトでは、Description プロパティ) ・ 発行元/製品 URL (Visual Studio プロジェクトでは、ManufacturerUrl プロパティ) ・ サポート連絡先 (Visual Studio プロジェクトでは、Author プロパティ) ・ サポート URL ・ サポート電話番号 ***プロジェクト・*** *このプロパティは、マージ モジュール プロジェクトには適用しません。*

テーブル 3-17・インポートで使用できるオプション (続き)

オプション	説明
[概要情報ストリーム] のプロパティ	インポートしている Visual Studio プロジェクトで構成された Summary Information Stream プロパティ (Title、Subject、Keywords、および TargetPlatform) を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 このチェック ボックスを選択すると、InstallShield は [一般情報] ビューにある次の設定の値を、Visual Studio プロジェクトで構成された値でアップデートします。 ・ タイトル ・ サブジェクト ・ キーワード ・ "テンプレートの概要" 設定の "プロセッサ" の種類 (Visual Studio プロジェクトでは、TargetPlatform プロパティ) ***エディション***・*TargetPlatform 値のインポートは InstallShield Premier および InstallShield でサポートされています。* ***プロジェクト***・*このプロパティは、マージ モジュール プロジェクトには適用しません。*
製品コード	インポートしている Visual Studio プロジェクトで構成された ProductCode プロパティの値を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 製品コードは、InstallShield の [一般情報] ビューで構成されます。 ***プロジェクト***・*このプロパティは、マージ モジュール プロジェクトには適用しません。*
アップグレード コード	インポートしている Visual Studio プロジェクトで構成された UpgradeCode プロパティの値を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 アップグレード コードは、InstallShield の [一般情報] ビューで構成されます。 ***プロジェクト***・*このプロパティは、マージ モジュール プロジェクトには適用しません。*

テーブル 3-17 • インポートで使用できるオプション (続き)

オプション	説明
すべてのユーザー	インポートしている Visual Studio プロジェクトで構成された InstallAllUsers プロパティの値を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 このチェック ボックスを選択すると、InstallShield は [一般情報] ビューの ALLUSERS 設定の値を、Visual Studio プロジェクトの InstallAllUsers プロパティで構成された値に基づいてアップデートします。 ***プロジェクト***• *このプロパティは、マージ モジュール プロジェクトには適用しません。*
プロジェクト言語	インポートしている Visual Studio プロジェクトの Localization プロパティで選択された言語を InstallShield プロジェクトで使用するには、このチェック ボックスを選択します。 言語は、InstallShield の [一般情報] ビューの "セットアップ言語" 設定で構成されます。 このチェック ボックスを選択していて、Visual Studio プロジェクトの言語が InstallShield プロジェクトの言語と一致しなかった場合、InstallShield はプロジェクト内の既存の文字列エントリの値を Visual Studio プロジェクトの言語のデフォルト文字列エントリ値と置換します。たとえば、このチェック ボックスを選択していて、InstallShield プロジェクトの言語がスペイン語で、Visual Studio プロジェクトの言語がドイツ語であった場合、InstallShield はプロジェクト内のスペイン語のランタイム文字列をデフォルトのドイツ語翻訳で置換します。このため、[一般情報] ビュー内の "発行者" 設定を更新している場合など、文字列エントリの値を編集した後に Visual Studio プロジェクトの言語をインポートすると、InstallShield は "発行者" 設定の値およびその他の文字列の値を、デフォルトのドイツ語の文字列エントリの値に置換します。 ***プロジェクト***• *このプロパティは、マージ モジュール プロジェクトには適用しません。*

[概要] パネル

[概要] パネルでは、Visual Studio デプロイメント プロジェクト インポート ウィザードで指定した設定を確認できます。設定のいずれかを変更する場合は、該当するパネルが表示されるまで [戻る] ボタンをクリックします。プロジェクトをインポートするには、[完了] ボタンをクリックします。

Web の配布ウィザード

The Web Deployment Wizard enables you to build a setup for Web deployment.このウィザードを使うと、セットアップにリンクする Web ページを自動的に作成できるほか、Windows Installer エンジンを含めるか指定したり、デジタル署名やセキュリティ情報を提供することができます。このウィザードの関連パネルを以下に示します。

- ようこそ
- リンク タイプ
- Windows Installer エンジンのオプション
- Windows Installer の場所
- 詳細設定
- Internet Explorer をターゲットするためのデジタル署名とセキュリティ
- 概要

[ようこそ] パネル

The Web Deployment Wizard enables you to build a setup for Web deployment.このウィザードを使うと、セットアップにリンクする Web ページを自動的に作成できるほか、Windows Installer エンジンを含めるか指定したり、デジタル署名やセキュリティ情報を提供することができます。

[リンク タイプ] パネル

[リンクの種類] パネルでは、Web ページに配置するリリースへのハイパーリンクの種類を選択します。リンクには、インストールを自動的に開始する One-Click Install テクノロジを使用したリンクと、エンド ユーザーがインストールを実行または保存するよう指示されるダイアログを表示するリンクがあります。

One-Click Install リンクを作成する場合は、[はい] を、エンド ユーザーがリンクをクリックしたときに標準の[保存/実行] ダイアログを表示する場合は [いいえ] を選択します。

[Windows Installer エンジン オプション] パネル

[Windows Installer エンジンのオプション] パネルでは、作成中のインストールに Windows Installer インストールを含めるかどうかを指定します。次のオプションを設定できます。

テーブル 3-18 • [Windows Installer エンジン オプション] パネルの設定

設定	説明
Windows Installer 3.1 エンジンを含める—Windows 2000 SP3 またはそれ以降	インストールに Windows Installer 3.1 エンジンを含める場合は、このチェック ボックスを選択します。
ランチャ警告を抑制する	オペレーティング システムに、インストールでアップデートできない Windows Installer サービスがある場合にエンドユーザーに表示される警告を非表示にすることができます。

テーブル 3-18 • [Windows Installer エンジン オプション] パネルの設定 (続き)

設定	説明
Windows Installer の再起動を遅延する	インストールが完了する前に Windows Installer エンジン インストールが再起動を必要とする場合に、これを遅延するには、このチェック ボックスを選択します。 必要時に、Windows Installer エンジンをインストールまたは更新した直後にシステムが再起動できるようにするには、このチェック ボックスをクリアします。

[Windows Installer の場所] パネル

セットアップに Windows Installer を含めるよう選択した場合、インストールがセットアップエンジンを探す場所として 2 つのオプションがあります。

テーブル 3-19 • [Windows Installer の場所] パネルの設定

設定	説明
以下に指定した Web からエンジンをロードする	インストールが、必要に応じて Windows Installer エンジン再配布可能ファイルを指定された URL からダウンロードする場合は、このオプションを選択します。デフォルトの URL (http://www.installengine.com/Msiengine30) は、便宜上、Revenera が維持しています。 このオプションを選択すると、インストールのサイズを縮小できる利点があります。ただし、エンド ユーザーは Windows Installer エンジンが必要なときにインターネットに接続できなければならないこと、また、エンジンのインストールをダウンロードするのに時間がかかる可能性があるという欠点があります。
Setup.exe からエンジンを抽出する	選択した Windows Installer エンジン再配布可能ファイル(複数可) を Setup.exe に圧縮し、実行時に必要に応じて抽出するには、このオプションを選択します。 このオプションを選択すると Setup.exe のサイズが増大しますが、エンジンをダウンロードする必要がないため、インストールがより早く完了する利点があります。また、このオプションにはインターネット接続が不要です。

[詳細設定] パネル

[詳細設定] パネルでは 2 つの追加オプションを使用できます。

テーブル 3-20 • [詳細設定] パネルの設定

設定	説明
メディア サイズの最適化	このオプションが選択されていると、Setup.exe のサイズが縮小されます。ただし、これを選択すると、ビルドプロセスの所要時間は長くなります。

テーブル 3-20・[詳細設定] パネルの設定 (続き)

設定	説明
ターゲットマシンにメディアをコピーする	.msi ファイルやその他のインストール ファイルをターゲット システムにキャッシュする場合、このチェック ボックスを選択して、キャッシュの場所を指定します。ファイルのキャッシュを行わない場合は、このチェック ボックスをクリアします。 .msi ファイルやその他のインストール ファイルをターゲット システム上にキャッシュすると、アプリケーションのメンテナンスや修復の際に便利です。
次のパスワードで Setup.exe をパスワードで保護する	インストールをパスワードで保護して、エンド ユーザーがプロジェクトをインストールするときに有効なパスワードの入力を求めるには、このチェック ボックスを選択して、有効なパスワードを入力します。

[Internet Explorer 対象のデジタル署名とセキュリティ] パネル

[デジタル署名とセキュリティ] パネルでは、InstallShield がファイルに署名するときに使用するデジタル署名に関する情報 (証明機関より付与されたデジタル証明書ファイルを含む) を指定します。

テーブル 3-21・[Internet Explorer 対象のデジタル署名とセキュリティ] パネルの設定

設定	説明
Internet Explorer のメディアに署名する	インストール パッケージにデジタル署名を行うには、このチェック ボックスを選択してから、このパネル内のその他の設定を完了します。
URL	完全修飾 URL を入力します (例、**http://www.mydomain.com**)。この URL は、エンドユーザーが製品や組織、会社に関しての情報を得るためのリンク先として、デジタル署名の中で使用されます。
デジタル証明書情報	リリースに署名を行うために使用するデジタル証明書を指定するには、この設定の横にある [参照] ボタンをクリックします。[証明書の選択] ダイアログ ボックスが開いて、.pfx ファイルの場所を指定するか、証明書を含む証明書ストアについての情報を指定することができます。 詳細については、「[証明書の選択] ダイアログ ボックス」を参照してください。

テーブル 3-21 • [Internet Explorer 対象のデジタル署名とセキュリティ] パネルの設定 (続き)

設定	説明
証明書パスワード	使用する .pfx にパスワードがある場合、それを入力します。InstallShield はパスワードを暗号化して、プロジェクト (.ise) ファイルに保存します。 ビルド時に InstallShield はパスワードを使って .pfx でファイルの署名を行います。証明書がパスワードで保護されているのにもかかわらず、この設定に何も入力しなかった場合、.pfx ファイルを使った署名が失敗します。 ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。

[概要] パネル

[概要] パネルには、Web 配布のために、それまでのパネルで提供した設定情報が表示されます。これらの設定を見て、内容が正確か確認してください。選択したオプションによって、[完了] をクリックすると Web 配布ウィザードが終了してリリースがビルドされる場合と、ただウィザードが終了する場合があります。

ビュー リファレンス

InstallShield インストール開発環境 (IDE) は、あらゆる範囲にわたる機能をまとめた複数のビューで構成されます。[ビュー リファレンス] セクションは、InstallShield インターフェイスの各ビューについて説明します。

[セットアップの編成] ビュー

何かを構築する場合、最初にしっかりした基礎を築く必要があります。インストールのベースは、[一般情報] ビューを通したアプリケーション情報の指定、[機能] ビューでの機能の作成、[セットアップの種類] ビューでのセットアップの種類の指定、および [アップデート通知] ビューでのアップデート通知の準備を行なうことによって形成されます。[セットアップの編成] の下にある各ビューについての説明は次の通りです。

テーブル 3-1 ・ [セットアップの編成] ビューの下にあるビュー

表示	説明
一般情報	[一般情報] ビュー には、プロジェクト、会社、および製品に関する基本情報が表示されます。新しいインストール プロジェクトを作成する場合、プロジェクトの [一般情報] ビューの設定を構成する必要があります。InstallShield は、デフォルトの設定を使用して新しいプロジェクトを作成しますが、独自のニーズに合ったデータをプロジェクトに含めるためには、独自の値を設定することをお勧めします。
機能	機能は、セットアップの構成要素です。これらは、プログラムファイル、ヘルプファイル、クリップアートなど、アプリケーションの特徴的な部分をエンド ユーザーに表示します。機能やサブ機能は、[機能] ビューで作成できます。
セットアップの種類	[セットアップの種類] では、製品の異なる構成がエンド ユーザーに提供されます。これらの構成は、製品の実行のために必須ではないサイズが大きな機能を配布する場合に役立ちます。
アップグレード パス	以前のバージョンのアプリケーションを配布済みで、エンド ユーザーのシステムにインストールされている以前のバージョンを現在のバージョンに更新する場合、[アップグレード パス] ビューでアップグレード情報を表示することができます。
アップデート通知	[アップデート通知] ビューでは、元のインストールの FlexNet Connect を有効にすることができます。 ***注意:*** *プロジェクトで自動アップデート通知を有効にすると、作成中のインストールに約 600 KB のファイルが追加されます。FlexNet Connect を動作させるためには、これらのファイルをアプリケーションと共に配布しなくてはなりません。サーバーの帯域幅の制限や、その他の理由のためにインストールにこれらのファイルを含むことができない場合、[いいえ] を選択して自動アップデート通知を無効にすることもできます。ただし、元のインストールの自動通知が有効でない場合、エンドユーザーにアップデートを配布するときに FlexNet Connect を利用することはできません。したがって [いいえ] を選択すると、将来的に自動アップデート通知機能を活用することができなくなります。*

[一般情報] ビュー

[一般情報] ビューでは、セットアップ、会社、およびインストールしているアプリケーションの基本情報を見ることができます。このビューに入力する情報には、参照だけに使用する情報、Windows ロゴ要件を満たすために使用する情報、さらにセットアップの基本的なインストールの設定に関する情報があります。

新しいインストール プロジェクトを作成する場合、プロジェクトの [一般情報] ビューの設定を構成する必要があります。InstallShield は、デフォルトの設定を使用して新しいプロジェクトを作成しますが、独自のニーズに合ったデータをプロジェクトに含めるためには、独自の値を設定することをお勧めします。

[一般情報] ビューは、次の要素で構成されます:

- ボタンの列
- 設定を表示するグリッド

次のテーブルは、[一般情報] ビューに表示されるボタンについて説明します。

テーブル 3-2 ・ [一般情報] ビューのコントロール

コントロールの名前	アイコン	説明
カテゴリー別		カテゴリごとに設定を並べ替えます。
アルファベット順		設定をアルファベット順に並べ替えます。

[一般情報] ビューの各設定についての説明は、「一般情報の設定」を参照してください。

一般情報の設定

[一般情報] ビューの設定は、次のメイン カテゴリに分かれています:

- 全般
- 概要情報ストリーム
- プログラムの追加と削除
- ソフトウェア識別タグ

[全般] の設定

[一般情報] ビューの "全般" 領域では、製品名や製品バージョンなどの詳細を指定します。この領域には、以下の設定があります。

テーブル 3-3 ・ 全般設定

設定	説明
製品名	製品の名前を入力します。製品名がどのように使用されるかについては、「製品名の指定」を参照してください。

テーブル 3-3 · 全般設定 (続き)

設定	説明
製品バージョン	製品のバージョン番号を入力します。バージョンには、数値のみを使用できます。一般的なフォーマットは *aaa.bbb.ccccc* または *aaa.bbb.ccccc.ddddd* で、*aaa* はメジャー バージョン番号、*bbb* はマイナー バージョン番号、*ccccc* はビルド番号、および *ddddd* はバージョン番号を示します。*aaa* と *bbb* の最大値は 255 です。*ccccc* と *ddddd* の最大値は、65,535 です。 4 番目のフィールド (*ddddd*) を含めることもできますが、インストールは異なる製品バージョンを区別するときに製品バージョンのこの部分を無視します。詳細については、「製品バージョンを指定する」を参照してください。 リリースに Setup.exe が含まれる場合、指定した製品バージョンが Setup.exe の [プロパティ] ダイアログ ボックスに表示されます。詳細については、「セットアップ ランチャーのファイルのプロパティをカスタマイズする」を参照してください。
製品コード	この製品を固有に識別する GUID を入力します。InstallShield使って、自動的に異なる GUID を作成するには、この設定の **[新しい GUID の生成]** ボタン ({...}) をクリックします。 このコードは製品を一意的に識別するため、リリースを既に配布している場合は製品コードの変更はお薦めできません。 詳細については、「製品コードの設定」を参照してください。
アップグレード コード	この製品を固有に識別する GUID を入力します。InstallShield使って、自動的に異なる GUID を作成するには、この設定の **[新しい GUID の生成]** ボタン ({...}) をクリックします。 製品 GUID は、アンインストールまたはメンテナンスを元のインストールと関連付けるのに使用されます。新しい GUID は、(既存のプロジェクトのコピーを含む) 作成した各新規プロジェクトに対して自動的に生成されます。プロジェクトの製品 GUID を変更すると、以前の GUID を回復することはできません。このため、プロジェクトの製品 GUID の変更は、通常必要なく、変更する場合は注意が必要です。 詳細については、「アップグレード コードを設定する」を参照してください。

テーブル 3-3 · 全般設定 (続き)

設定	説明
INSTALLDIR	Windows Installer プロパティ INSTALLDIR の値を指定します。この値は、実行時に製品のファイルのほとんどがインストールされるインストール先ディレクトリを示します。次は、デフォルトの値です: [ProgramFilesFolder]*会社名¥製品名* パスをハードコード化する代わりに、パスの一部としてディレクトリ プロパティを入力することができます。ディレクトリ プロパティを選択するには、この設定で省略記号ボタン (...) をクリックします。ここで、適切なディレクトリをリストから選択するか、定義済みディレクトリ内に新しいディレクトリを作成できます。円記号を使用して、サブディレクトリの下位レベルを [ProgramFilesFolder]MyApp¥Bin のように区切ります。 ***Windows ロゴ·*** *Windows ロゴ プログラムに準拠するためには、製品のデフォルトのインストール先が Program Files フォルダーのサブフォルダー ([ProgramFilesFolder]) である必要があります。これは、システムのロケールやユーザー設定によって異なることがあります。* 詳細については、「デフォルトの製品インストール先フォルダー (INSTALLDIR) の設定」を参照してください。

テーブル 3-3 • 全般設定 (続き)

設定	説明
ロックダウンの設定方法	ロックダウン環境で製品を実行するエンド ユーザー向けに、ファイル、フォルダー、およびレジストリ キーを保護するためのアクセス許可の種類を選択します。選択可能なオプションは以下のとおりです: • **カスタム InstallShield 処理**— InstallShield は、プロジェクトにISLockPermissions テーブルとカスタム アクションを追加して、ターゲット システム上のアクセス許可を設定します。このオプションがデフォルト値です。 • **従来型の Windows Installer 処理** — InstallShield は、.msi データベースの LockPermissions テーブルを使って、製品のアクセス許可情報を格納します。 多くの場合、従来型の Windows Installer 処理よりも、カスタム InstallShield 処理の方が有利です。例: • カスタム オプションを使うと、従来のオプションではサポートされていない、多くのよく知られているセキュリティ識別子 (SID) を使用できます。 • 従来型のオプションとは違い、カスタム オプションでは、多くのよく知られている SID に翻訳されたユーザー名を使用できます。従来型のオプションで、非英語システム上で翻訳された名前を使ってアクセス許可を設定すると、インストールが失敗する可能性があります。 • カスタム オプションを使うと、指定するアクセス許可を特定のユーザーまたはグループが所持することを拒否できます。従来型の処理で、これは不可能です。つまり、従来型の処理の場合、特定のアクセス許可を設定することのみが可能で、アクセス許可を拒否することはできません。 この機能はプロジェクト全体に反映される設定で、プロジェクトに含まれるファイル、フォルダー、およびレジストリ キーに設定する新しいアクセス許可すべてに適用します。プロジェクトで既にいくつかのアクセス許可を構成済みの場合にこの設定の値を変更する場合、InstallShield では、既存アクセス許可に別の処理方法を使用するかどうかを指定できます。 この設定の構成方法については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

テーブル 3-3 · 全般設定 (続き)

設定	説明
DATABASEDIR	これはすべてのアプリケーションのデータベース ファイルのデフォルトのインストール先フォルダーを指定します。 ***メモ:*** *アプリケーションのデータベース ファイルのインストール先フォルダーを指定するとき、閉じ角かっこ (]) または円記号 (¥) の前後にスペースを入れることはできません。たとえば、次のようなパスは有効ではありません。* [ProgramFilesFolder] ¥MyApp¥Bin [ProgramFilesFolder]MyApp¥ Bin DATABASEDIR などのインストーラー フォルダー プロパティを使用すると、デフォルト値が指定されるので、注意してください。エンド ユーザーは、コマンドラインで Msiexec.exe を起動する時にプロパティを設定したり、[データベース フォルダー] ダイアログで機能に対して新しいインストール先フォルダーを選択することにより、この値を変更できます。
デフォルトフォント	インストールがユーザー インターフェイスに使用するフォントを選択するには、この設定で省略記号ボタン (...) をクリックします。フォントを指定しなかった場合、インストーラーはデフォルトのシステムフォントを使用します。

テーブル 3-3 · 全般設定 (続き)

設定	説明
ALLUSERS	プロジェクトで、Windows Installer の ALLUSERS プロパティをどう設定するかを指定します。ALLUSERS プロパティは、Windows Installer がマシンごとのインストールを実行するか、またはユーザーごとのインストールを実行するかを指定します。有効なオプションは次のとおりです: · **ALLUSERS=""（ユーザーごとインストール）**—Windows Installer はユーザーごとインストールを実行し、構成情報はユーザーの個人プロファイルに格納されます。 · **ALLUSERS=1（マシンごとインストール）**—ALLUSERS の値が 1 に設定されている場合、Windows Installer はマシンごとのインストールを試みます。マシンごとのインストールの場合、ショートカット、レジストリ エントリなどの構成情報は All Users のプロファイルに格納されます。 Windows Vista 以降のシステムでは、[ユーザー アカウント制御] が有効にされていて、ユーザーが管理者権限を持たないとき、製品をインストールするためには管理資格情報が必要です。 他のシステムでは、ユーザーが管理者権限を持たないとき、インストールはエラー メッセージを表示して終了します。 · **ALLUSERS=2（ユーザーごとまたはマシンごとインストール）**—Windows Vista 以降のシステムで、Windows Installer はマシンごとのインストールを試みます。以前のプラットフォームでは、ユーザーに権限がある場合、Windows Installer は、マシンごとのインストールを試みます。それ以外の場合、Windows Installer はユーザーごとのインストールを実行します。 すべての新規プロジェクトのデフォルト値は、ALLUSERS=1（マシンごとインストール）です。 ***メモ·** ここで ALLUSERS に指定する値は、実行時に上書きされる場合があります。詳細については、「ユーザーごとのインストールとマシンごとのインストールの違い」を参照してください。*

テーブル 3-3 • 全般設定 (続き)

設定	説明
MSI ログ記録の作成	Windows Installer 4.0 以降でインストールをログ記録するかどうかを指定するには、この設定で省略記号ボタン (...) をクリックして、[Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックスを起動します。このダイアログ ボックスで Windows Installer がインストールのログ記録を行うかどうかを指定します。このダイアログ ボックスを使用して、ログ記録されるメッセージの種類をカスタマイズすることもできます。 この設定には 3 種類の値から選択することができます。 • **いいえ** — インストールはログ記録されません。これがデフォルトの値です。 • **はい**—**voicewarmupx** のデフォルト値が MsiLogging プロパティに挿入されます。 • **カスタム** — [Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックスで指定した値が MsiLogging プロパティに挿入されます。 この設定の値が [はい] または [カスタム] の場合に、インストールが Windows Vista 以降または Windows Server 2008 以降において、Windows Installer 4.0 以降を使って実行されたとき、以下の処理が行われます: • インストーラーが、適切なログ記録モード (″MSI ログ記録の作成″ 値が [はい] の場合は **voicewarmupx**、または [Windows Installer 4.0 以降のログ記録オプション] ダイアログ ボックスで指定した任意のカスタム値) に従って、ログ ファイルを作成します。 • インストーラーが、MsiLogFileLocation プロパティに、ログ ファイルのパスを挿入します。 • SetupCompleteSuccess、SetupCompleteError および SetupInterrupted ダイアログに **[Windows Installer ログを表示]** チェック ボックスが追加されます。エンドユーザーがこのチェック ボックスを選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。 ″MSI ログの作成″ 設定は、Windows Vista 以降のシステムまたは Windows Server 2008 以降のシステム上で Windows Installer 4.0 以降を使って実行するインストールに適用します。Windows Installer の古いバージョンを実行する以前のシステム上では、実行時ダイアログで **[Windows Installer ログを表示する]** チェック ボックスは表示されません。 ログ記録されるメッセージの種類をカスタマイズする方法についてなど、詳しい情報は「Windows Installer インストールをログ記録するかどうかを指定する」を参照してください。

テーブル 3-3 • 全般設定 (続き)

設定	説明
高速インストール	サイズが大きい Windows Installer パッケージをインストールするのに要する時間を短縮するには、次のオプションから 1 つ以上選択することを考慮してください: • このインストールでシステム復元ポイントを保存しない • ファイル コスティングのみを実行して、その他のコスト チェックをスキップする • 進行状況メッセージの頻度を減らす この設定は、Windows Installer プロパティ MSIFASTINSTALL を構成します。このプロパティは、コマンドラインで設定可能です。 Windows Installer 5 で、この設定がサポートされています。以前のバージョンの Windows Installer はこれを無視します。

概要情報ストリームの設定

Windows Installer データベースは COM 構造化ストレージとして実装され、通常、COM 構造化ストレージ ファイルには Summary Information Stream が含まれています。Summary Information Stream には、会社や、インストール中のソフトウェアに関する情報が含まれます。

[一般情報] ビューの [概要情報ストリーム] 領域には、次の設定があります。

テーブル 3-4 • 概要情報ストリームの設定

設定	説明
タイトル	作成するデータベースの種類を指定します。製品インストールの場合、デフォルト値は **インストール データベース** です。これが推奨される値です。 入力した値は、[プロパティ] ダイアログ ボックスの [概要] タブに使用され、Windows Installer データベースを右クリックしてから [プロパティ] をクリックすると表示されます。
サブジェクト	製品の名前を入力します。 入力した値は、[プロパティ] ダイアログ ボックスの [概要] タブに使用され、Windows Installer データベースを右クリックしてから [プロパティ] をクリックすると表示されます。
作成者	会社名を指定します。 入力した値は、[プロパティ] ダイアログ ボックスの [概要] タブに使用され、Windows Installer データベースを右クリックしてから [プロパティ] をクリックすると表示されます。
キーワード	製品の Windows Installer データベースを説明するキーワードを指定します。 入力した値は、[プロパティ] ダイアログ ボックスの [概要] タブに使用され、Windows Installer データベースを右クリックしてから [プロパティ] をクリックすると表示されます。

テーブル 3-4・概要情報ストリームの設定 (続き)

設定	説明
[概要情報ストリーム] のコメント	製品に関する任意のコメントを入力します。この設定の通常の値は、次の通りです: **このインストーラー データベースには、MyProduct をインストールするために必要なロジックとデータが含まれています。** 入力した値は、[プロパティ] ダイアログ ボックスの [概要] タブに使用され、Windows Installer データベースを右クリックしてから [プロパティ] をクリックすると表示されます。
スキーマ	この設定を使って、インストール パッケージに必要な Windows Installer の最小バージョンを識別する整数を指定します。Windows Installer 2.0 を最小バージョンとする場合、**200** と入力します。Windows Installer 3.0 を最小バージョンとする場合、**300** と入力します。Windows Installer 3.1 を最小バージョンとする場合、**301** と入力します。Windows Installer 4.5 を最小バージョンとする場合、**405** と入力します。 エンド ユーザーのシステム上にある Windows Installer のバージョンが、"スキーマ" 設定で指定された最小要件よりも古い場合 (たとえば、インストールが Windows Installer 4.5 の機能を使用するため、スキーマの値が 405 に設定されているが、エンド ユーザーの Windows Installer バージョンが 3.1 の場合)、インストール時にエラーメッセージが表示され、インストールは途中で終了します。 "スキーマ" 設定で入力した値は、Windows Installer データベースの Page Count Summary プロパティに使用されます。
管理者特権が必要	インストールが管理者権限を必要とするかどうかを指定します。デフォルトの設定は [はい] です。 [いいえ] を設定した場合、InstallShield は Word Count Summary プロパティの 3 ビット目を設定して、.msi パッケージのインストールにシステム権限が必要でないことを示します。[いいえ] が選択された状態で、適切な権限をもたずに .msi パッケージがタスクを実行しようとすると、Windows Installer によって実行時エラーが表示される場合があります。 この設定は、Windows Vista 以降のシステムまたは Windows Server 2008 以降のシステム上で Windows Installer 4.0 以降を使って実行するインストールに適用します。以前のバージョンの Windows Installer と Windows は、この設定を無視します。 エンド ユーザーのインストール エクスペリエンスは、インストールが必要とする権限によってのみ実行された場合、安全性がより一層向上します。アプリケーションは、システム管理者のみによる実行が必須の場合を除き、最も低い権限で実行されることが理想的です。

[プログラムの追加と削除] の設定

コントロール パネルの [プログラムの追加と削除] (最新バージョンの Windows では [プログラム]) には、サポートへのリンク、連絡先の電話番号、製品のアップデート情報、および製品の製造元に関する情報をエンド ユーザーに提供できます。インストールの構成の仕方によって、エンド ユーザーはボタンをクリックしてインストールの削除、修復、変更を選択することができます。プロジェクトでこの情報を指定するには、[一般情報] ビューにある [プログラムの追加と削除] の設定を構成します。

テーブル 3-5 ・ [プログラムの追加と削除] の設定

設定	説明
[プログラムの追加と削除] エントリの表示	コントロール パネルの [プログラムの追加と削除] でアプリケーションのエントリを表示するかどうかを示します。選択可能なオプションは以下のとおりです: • **はい**—製品のエントリが、ターゲット システムの [プログラムの追加と削除] で表示されます。これがデフォルトの値です。 • **いいえ**—製品のエントリが、ターゲット システムの [プログラムの追加と削除] で表示されません。エンドユーザーが [プログラムの追加と削除] を使って、製品の削除やメンテナンスを行ったり、サポート情報を表示できなくなります。このオプションを選択した場合、InstallShield はその他の "プログラムの追加と削除" 設定を無効にします。
アイコンの表示	[プログラムの追加と削除] で製品のエントリに使用するアイコンファイルへ (.ico または .exe) の、開発システム上でのパスを入力します。パスとファイル名を手作業で入力する代わりに、この設定の省略記号ボタン (...) をクリックして、ファイルを参照できます。
変更ボタンの無効	[プログラムの追加と削除] で、製品の [変更] ボタンを無効にするかどうかを指定します。[変更] ボタンは、エンドユーザーが製品のインストール後、インストール オプションを変更できるボタンです。エンドユーザーは必要に応じて機能を追加または削除できます。
削除ボタンの無効	[プログラムの追加と削除] で、製品の [削除] ボタンを無効にするかどうかを指定します。[削除] ボタンを使用すると、エンドユーザーは 1 つのボタンをクリックするだけで製品を削除することができます。この場合、アンインストーラーはコンパクトなユーザー インターフェイスで実行します。 エンド ユーザーが [削除] ボタンをクリックして製品を削除すると、プロジェクトの [ユーザー インターフェイス] シーケンス内のアクションが実行されます。
修復ボタンの無効	[プログラムの追加と削除] で、製品の [修復] ボタンを無効にするかどうかを指定します。[修復] ボタンを使うと、ファイルが削除されていた、または壊れていた場合、エンドユーザーは Windows Installer の修復オプションを実行することができます。

テーブル 3-5 · [プログラムの追加と削除] の設定 (続き)

設定	説明
発行者	製品を作成した会社の名前を指定します。この情報は、[プログラムの追加と削除] で製品のエントリに表示されます。入力した値は、Windows Installer Manufacturer プロパティに格納されます。 ***Windows ロゴ·*** *Windows ロゴ プログラムの要件に準拠するには、発行元の指定が必要です。*
発行元/製品 URL	会社または製品の一般的な URL を入力してください (例、**http://www.installshield.com**)。 Windows の一部のバージョンでは、[サポート情報] ダイアログ ボックスに表示される発行元の名前は、この URL へのハイパーリンクです。[サポート情報] ダイアログ ボックスは、エンド ユーザーが [プログラムの追加と削除] で製品のエントリ用のサポート情報ハイパーリンクをクリックすると表示されます。
サポート連絡先	エンドユーザーのテクニカルサポート窓口となる担当者または部門の名前を入力します。 Windows の一部のバージョンでは、この情報は、[プログラムの追加と削除] で製品のエントリ用の [サポート情報] ダイアログ ボックスで表示されます。
サポート URL	エンド ユーザーが製品のテクニカル サポート情報を参照できる URL を入力します。この URL は、[プログラムの追加と削除] で製品のエントリに表示されます。 ***Windows ロゴ·*** *Windows ロゴ プログラムの要件に準拠するには、有効な URL の入力が必要です。*
サポート電話番号	製品のテクニカル サポートの電話番号を入力します。 Windows の一部のバージョンでは、この情報は、[プログラムの追加と削除] で製品のエントリ用の [サポート情報] ダイアログ ボックスで表示されます。
README	製品の Readme ファイルの名前、またはパスを入力します。その代わりに、有効な URL を指定してインターネット上の Readme ファイルのリンクを表示することもできます。 Windows の一部のバージョンでは、この情報は、[プログラムの追加と削除] で製品のエントリ用の [サポート情報] ダイアログ ボックスで表示されます。 ***メモ·*** *リストから選択したディレクトリ識別子およびファイルまたはフォルダー名の間に円記号 (¥) を使用する必要はありません。たとえば、***[INSTALLDIR]MyFolder¥Readme.txt** *が有効なパス フォーマットです。*

テーブル 3-5 • [プログラムの追加と削除] の設定 (続き)

設定	説明
製品アップデート URL	エンド ユーザーが製品のアップデート情報を参照、または最新バージョンをダウンロードできる URL を指定します。 Windows の一部のバージョンでは、この情報は、[プログラムの追加と削除] で製品のエントリ用の [サポート情報] ダイアログ ボックスで表示されます。
[プログラムの追加と削除] のコメント	製品についてのコメントを入力します。この情報は、[プログラムの追加と削除] で製品のエントリに表示されます。

ソフトウェア識別タグの設定

[一般情報] ビューの [ソフトウェア識別タグ] 領域を使って、インストールに ISO/IEC 19770-2 ソフトウェア識別タグを含めるかどうかを指定できます。タグが含まれている場合、この領域で、[一般情報] ビューの他の領域でまで指定されていない識別情報を指定することもできます。

詳細については、「製品のソフトウェア識別タグを含める」を参照してください。

テーブル 3-6 • ソフトウェア識別タグの設定

設定	説明
ソフトウェア識別タグを使用	インストールに、ISO/IEC 19770-2 ソフトウェア識別タグを含めるかどうか指定します。[はい] を選択した場合、[一般情報] ビューのこの領域にある、他のタグ関係の設定を使って、[一般情報] ビューの他の領域でまで指定されていない識別情報を指定することができます。
ソフトウェア エンタイトルメントが必須である	ソフトウェアが正しく運用されていることを判断するために、製品に対応するソフトウェア エンタイトルメントがあることを条件とするかどうかを指定します。 一般的に、有料ソフトウェアの場合、この設定には [はい] を選択し、無償ソフトウェアである場合には、この設定に [いいえ] を選択します。
一意な ID	この特定の製品の特定のバージョンを識別する一意の ID を入力します。 InstallShield使って、自動的に異なる GUID を作成するには、この設定の **[新しい GUID の生成]** ボタン (...) をクリックします。 InstallShield では、タグ ファイルの名前の一部として入力した値 (*TagCreatorID_UniqueID*.swidtag) が使用されますので注意してください。したがって、入力する ID に、ファイル名には使えない文字は使用できません。
タグ作成者	タグを作成した組織の名前を指定します。

テーブル 3-6・ソフトウェア識別タグの設定 (続き)

設定	説明
タグ作成者 ID	タグを作成した組織の登録 ID を入力します。この ID は、同じ作成者名を持ち、かつ、異なる国に属する法的組織が複数あるとき、それらを区別するために有益です。 登録 ID の表示規則は、次のとおりです: **regid.YYYY-MM.ReversedDomainName,division** 登録 ID ビューは、次の部分から構成されます: ・ regid.—文字列 *regid* は、XML 部分がソフトウェア識別タグの登録 ID であることを示します。この文字列の後には、必ずピリオド (.) が必要です。 ・ YYYY-MM.—登録 ID のこの部分は、ドメイン名がタグ作成者によって丸 1 ヶ月所有された最初の年月 (YYYY) および (MM) を示します。たとえば、タグを作成するとき、ドメイン名を 1999 年 2 月 15 日に購入している場合、登録 ID のこの部分に **1999-03** を使用します。つまり、ドメイン名を完全な 1 ヶ月間所有した最初の年月は 1999 年 (1999) の 3 月 (03) となります。年と月は、必ずダッシュで区切ります。 ・ ReversedDomainName—この部分は、ソフトウェア識別タグを作成している組織のドメイン名を逆さにしたものです。たとえば、revenera.com がドメイン名であるとすると、その逆さ表示は: **com.revenera** ・ ,division – この部分はオプションで、コンマ (,) で始まり、追加文字列が続きます。組織内の異なる部署や部門によって区別するのに役立つ文字列を入力できます。登録 ID に、このオプションの区別を使用しない場合は、コンマも追加文字列も不要です。 InstallShield では、タグ ファイルの名前の一部として入力した値 (*TagCreatorID_UniqueID*.swidtag) が使用されますので注意してください。したがって、入力する ID に、ファイル名には使えない文字は使用できません。
ソフトウェア作成者	ソフトウェアを作成した組織の名前を指定します。 この設定はオプションです。この設定を空白のままにしておくと、ソフトウェア作成者の名前に "タグ作成者" 設定の値が使用されます。

テーブル 3-6・ソフトウェア識別タグの設定 (続き)

設定	説明
ソフトウェア作成者 ID	ソフトウェアを作成した組織の登録 ID を入力します。この ID は、同じ作成者名を持ち、かつ、異なる国に属する法的組織が複数あるとき、それらを区別するために有益です。 この設定はオプションです。この設定を空白のままにしておくと、ソフトウェア作成者 ID に "タグ作成者 ID" 設定の値が使用されます。 登録 ID の表示規則は、次のとおりです: **regid.YYYY-MM.ReversedDomainName,division** 登録 ID ビューは、次の部分から構成されます: ・ regid.—文字列 *regid* は、XML 部分がソフトウェア識別タグの登録 ID であることを示します。この文字列の後には、必ずピリオド (.) が必要です。 ・ YYYY-MM.—登録 ID のこの部分は、ドメイン名がタグ作成者によって丸 1 ヶ月所有された最初の年月 (YYYY) および (MM) を示します。たとえば、タグを作成するとき、ドメイン名を 1999 年 2 月 15 日に購入している場合、登録 ID のこの部分に **1999-03** を使用します。つまり、ドメイン名を完全な 1 ヶ月間所有した最初の年月は 1999 年 (1999) の 3 月 (03) となります。年と月は、必ずダッシュで区切ります。 ・ ReversedDomainName—この部分は、ソフトウェア識別タグを作成している組織のドメイン名を逆さにしたものです。たとえば、revenera.com がドメイン名であるとすると、その逆さ表示は: **com.revenera** ・ ,division – この部分はオプションで、コンマ (,) で始まり、追加文字列が続きます。組織内の異なる部署や部門によって区別するのに役立つ文字列を入力できます。登録 ID に、このオプションの区別を使用しない場合は、コンマも追加文字列も不要です。
ソフトウェア ライセンサー	ソフトウェアの著作権を所有する組織の名前を入力します。 この設定はオプションです。この設定を空白のままにしておくと、ソフトウェアライセンサーの名前に "タグ作成者" 設定の値が使用されます。

テーブル 3-6・ソフトウェア識別タグの設定 (続き)

設定	説明
ソフトウェア ライセンサー ID	ソフトウェアの著作権を所有する組織の登録 ID を入力します。この ID は、同じライセンサー名を持ち、かつ、異なる国に属する法的組織が複数あるとき、それらを区別するために有益です。 この設定はオプションです。この設定を空白のままにしておくと、ソフトウェア ライセンサー ID に "タグ作成者 ID" 設定の値が使用されます。 登録 ID の表示規則は、次のとおりです: **regid.YYYY-MM.ReversedDomainName,division** 登録 ID ビューは、次の部分から構成されます: ・ regid.—文字列 *regid* は、XML 部分がソフトウェア識別タグの登録 ID であることを示します。この文字列の後には、必ずピリオド (.) が必要です。 ・ YYYY-MM.—登録 ID のこの部分は、ドメイン名がタグ作成者によって丸 1 ヶ月所有された最初の年月 (YYYY) および (MM) を示します。たとえば、タグを作成するとき、ドメイン名を 1999 年 2 月 15 日に購入している場合、登録 ID のこの部分に **1999-03** を使用します。つまり、ドメイン名を完全な 1 ヶ月間所有した最初の年月は 1999 年 (1999) の 3 月 (03) となります。年と月は、必ずダッシュで区切ります。 ・ ReversedDomainName—この部分は、ソフトウェア識別タグを作成している組織のドメイン名を逆さにしたものです。たとえば、revenera.com がドメイン名であるとすると、その逆さ表示は: **com.revenera** ・ ,division – この部分はオプションで、コンマ (,) で始まり、追加文字列が続きます。組織内の異なる部署や部門によって区別するのに役立つ文字列を入力できます。登録 ID に、このオプションの区別を使用しない場合は、コンマも追加文字列も不要です。

[機能] ビュー

機能およびサブ機能は、アプリケーションとセットアップ用の関連アクセサリのパッケージ化に対してほとんど無制限の柔軟性を与えます。機能は、[機能] ビューで作成および設定できます。

機能は、エンド ユーザーから見たプリケーションの構成要素です。つまり、ヘルプ ファイルや製品パッケージの一部など製品の特殊機能を表し、エンド ユーザーの選択によってインストールしたりアンインストールしたりできます。アプリケーション全体は特定の目的を実行する機能ごとに分ける必要があります。

複雑なアプリケーションを容易にレイアウトするために、InstallShield では機能とサブ機能の両方が作成できます。サブ機能は機能をさらに分割したものです。機能は、ユーザーが選択してインストールできる、アプリケーションまたはアプリケーション スイートに含まれた自己完結型の要素である必要があります。そのため、アプリケーションの各部分をある "親" 機能のサブ機能として構成するのが、最も合理的な方法と言えます。機能はすべて可視的であることを想定するため、エンド ユーザーは機能のどの部分をインストールするかを Custom Setup ダイアログで選択できるようになります。

デフォルトの機能である [常にインストール] は名前を変更したり、プロジェクトから削除したり、サブフォルダーを追加したりすることはできません。この機能には、セットアップの一部としてインストールが必要なファイルが含まれています。

作成できるサブ機能は最大15までですが、編成のためにデザインを可能な限り単純にしておくことをお勧めします。

機能の設定

機能の設定は以下のとおりです。

テーブル 3-7 ・ 機能の設定

設定	説明
説明	機能の簡単な説明を入力します。機能の説明は、エンド ユーザーが機能またはサブ機能をクリックしたときに [カスタム セットアップ] ダイアログに表示されます。
リモート インストール	機能のデフォルトのインストール状態を示します。有効なオプションは次のとおりです: ・ **ソースを優先する**—この機能のファイルは、デフォルトで CD-ROM またはネットワークの場所といったソース メディアから直接実行されます。 ・ **ローカルを優先する**—この機能のファイルは、デフォルトでターゲット システム上にインストールされます。 ・ **親を優先する**—サブ機能のデフォルト状態は、親機能のそれと同じです。このオプションは、機能がサブ機能である場合にのみ使用できます。 詳細については、「機能の ″リモート インストール″ 設定を設定する」を参照してください。
Visible	機能の ″表示″ プロパティを設定することによって、[カスタム セットアップ] ダイアログでの機能の表示方法を指定します。次のオプションから選択してください。 ・ **閉じて表示する**—デフォルトでサブ機能が閉じた状態で、[カスタム セットアップ] ダイアログの機能が表示されます。 ・ **展開して表示する**—デフォルトでサブ機能が展開された状態で [カスタム セットアップ] ダイアログの機能が表示されます。 ・ **非表示**—機能は、エンド ユーザーに対して [カスタム セットアップ] ダイアログに表示されません。 この設定によって、機能がインストールされるかどうかについて、直接影響を及ぼすことはありません。非表示にした機能は、自動的にすべてインストールされるというわけではなく、インストールが必要な機能の場合は選択解除できず、インストールするべきではない機能の場合は選択できないようになります。 詳細については、「機能をエンドユーザーへ表示する」を参照してください。

テーブル 3-7・機能の設定 (続き)

設定	説明
アドバタイズ	この機能の適切なアドバタイズ オプションを選択します。選択可能なオプションは以下のとおりです: ・ **アドバタイズを許可する**—エンド ユーザーは、[カスタム セットアップ] ダイアログで、この機能のアドバタイズ オプションを選択できます。アドバタイズは許可されますが、インストール実行時のデフォルトのオプションではありません。 ・ **アドバタイズを優先する**—この機能はデフォルトでアドバタイズされます。エンドユーザーは、Custom Setup ダイアログの機能の "アドバタイズ" オプションを変更できます。 ・ **アドバタイズを許可しない**—この機能のアドバタイズを許可しない場合は、このオプションを選択します。エンドユーザーは、[カスタム セットアップ] ダイアログで機能をアドバタイズできません。 ・ **サポートされていない場合アドバタイズ不可**—アドバタイズは、Internet Explorer 4.01 以降がインストールされたシステムでのみ機能します。この条件を満たさないターゲット システムでは、アドバタイズは許可されません。ターゲット システムがアドバタイズをサポートする場合、アドバタイズは許可されます。 詳細については、「機能のアドバタイズ」を参照してください。
Required	ターゲット システム上で機能が必要かどうかを示します。機能が必要な場合、エンド ユーザーは、[カスタム セットアップ] ダイアログでこれを選択解除することができません。
条件	この設定は、この機能に条件が設定されているかどうかを示します。1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。詳細については、「機能の条件を設定する」を参照してください。 条件を追加すると、InstallShield によって、その条件の機能設定のグリッドに新しい行が追加されます。 条件を追加するか、既存の条件を編集または削除するには、"条件" 設定で省略記号ボタン (...) をクリックします。
Comments	機能に対する変更の履歴を残したり、将来の参照のためにコメントを入力します。機能のコメントはプロジェクトファイルにのみ保存され、インストールでは一切使用されません。
ビルド時にマージする REG ファイル	ビルド時に .reg ファイルを機能のレジストリ エントリとマージする場合、.reg ファイルの完全パスを入力します。その代わりに、この設定で省略記号ボタン (...) をクリックして、.reg ファイルを参照することもできます。
キー名	この読み取り専用設定は、機能を代表するのに使用される Windows Installer キーを表示します。このキーをカスタム アクションで使用して、機能にアクセスできます。

[セットアップの種類] ビュー

[セットアップの種類] ビューから、インストール中に表示されるセットアップの種類、セットアップの種類の名前および各セットアップの種類に関連付けられる機能を設定できます。以下は、セットアップの種類が通常どのように定義されているかを説明しています。

テーブル 3-8・セットアップの種類

セットアップの種類	説明
標準	[標準] セットアップ タイプは通常、プログラムの大部分の機能を含んでいます。たとえば、セットアップにマルチメディア チュートリアルが含まれている場合、それは標準セットアップの一部として含まれます。
最小	このセットアップの種類は、通常アプリケーションを実行するために必要な機能だけを含みます。このセットアップ タイプはディスクの容量に制限があるノートブック コンピューターなどのユーザーのためにデザインされています。
カスタム	このセットアップの種類では、インストールする機能をユーザーが選択することができます。もちろん、必要な機能が確実にインストールされるようにマークされている必要があります。しかし、オンラインヘルプのような機能はインストールする必要がないかもしれません。エンドユーザーは、不要な機能をインストールするかどうか選択できます。

[セットアップの種類] ビューでセットアップの種類を選択すると、以下の設定を構成できます:

テーブル 3-9・セットアップの種類

設定	説明
説明	このセットアップの種類の説明を入力します。説明は、[セットアップの種類] ダイアログでエンド ユーザーに表示されます。 複数行にわたる説明を入力するには、エスケープ シーケンス ¥n を使います。たとえば、InstallShield¥nExpress と入力した場合、[セットアップの種類] ダイアログで説明が次のように表示されます: InstallShield Express

[アップグレード パス] ビュー

[アップグレード パス] ビューで、製品の更新方法を特定する情報を指定することができます。たとえば、現在のバージョンを更新する製品について、以前のバージョンのアップグレードコード (製品 GUID) や、最小/最大バージョン、言語識別子、またその他の情報を提供することができます。

[アップグレード パス] ビューを使うと、メジャー アップグレードが適用されたときにターゲット システムのグローバル アセンブリ キャッシュ (GAC) からアセンブリが削除されるという問題を避けることもできます。詳細については、「アップグレード中にグローバル アセンブリ キャッシュからアセンブリが削除されるのを防ぐ方法」を参照してください。

アップグレード パスの設定

アップグレード パスの設定を指定すると、エンドユーザーが以前に配布された Windows Installer を使うバージョンの製品をインストールしている場合や、インストールされているバージョンが現在のリリースと違う製品コードやバージョン番号を持つ場合、エンドユーザーのシステムにどのような影響を与えるか設定することができます。各アップグレード エントリのプロパティの説明を以下に示します。

テーブル 3-10 • アップグレード パスの設定

設定	説明
アップグレード コード	この設定には、アプリケーションのバージョンを置換する製品バージョンのアップグレード コード GUID の値が含まれています。現在のバージョンのインストールが開始されると、Windows Installer エンジンはターゲット システムを検索して、指定されたアップグレード コードを探します。一致するアップグレード コードが、他のアップグレード プロパティと一緒に検出されると、Windows Installer は新しいバージョンをインストールしてターゲット システムをアップグレードします。 システムの .msi ファイルを選択することにより、アップグレード パス ツリーにアップグレード エントリを追加 すると、その .msi ファイルのアップグレードコードは自動的に "アップグレード コード" 設定に書き込まれます。アップグレードする製品バージョンの .msi ファイルがシステム上にない場合は、この設定にアップグレードコードを入力することができます。 また、省略記号ボタン (...) を使用して、システム上の .msi ファイルまたは .msi ファイルが含まれる Setup.exe ファイルを参照して、"アップグレード コード" 設定に値を入力することもできます。 ***注意•*** *アップグレード コードは、Windows Installer テクノロジを使ってインストールされた以前のバージョンの製品と同じである必要があります。*

テーブル 3-10 • アップグレード パスの設定 (続き)

設定	説明
最小バージョン	″最小バージョン″ 設定は、Windows Installer によるアップグレード対象のインストール済みバージョン検索を定義します。以前にインストールされたバージョンのアップグレード コードは、″アップグレード コード″ 設定に指定された値とまず一致している必要があります。一致するアップグレード コードが検出されると、指定の最小/最大バージョンや、そのバージョンを含めるか除外するか (″最小バージョンを含める″ と ″最大バージョンを含める″ 設定で指定) によって検索を絞り込むことができます。 この設定には、現在のバージョンの更新対象となる製品の最小バージョンが含まれています。設定には次の方法のいずれかで書き込むことができます: • システムの .msi または Setup.exe ファイルを参照してアップグレード パス ツリーに アップグレード エントリを追加すると、InstallShield は ″最小バージョン″ 設定にある .msi ファイルのバージョンを使用します。 • また、この設定に、0.00.0000 または 00.00.0000 (ゼロはバージョン番号を表す) の形式で最小バージョンを入力することもできます。 ***注意***• *バージョンの設定 (最小バージョン、最大バージョン、または両方) の値は、最低でも 1 つ指定する必要があります 。*
最小バージョンを含める	アップグレードに、″最小バージョン″ 設定で指定した最小バージョンを含めるかどうか指定します。 Windows Installer がターゲット システム上にインストールされている以前のバージョンの製品の最小バージョンを検索対象に含める場合は [はい] を選択します。
最大バージョン	″最大バージョン″ 設定は、Windows Installer によるアップグレード対象のインストール済みのバージョン検索を定義します。以前にインストールされたバージョンのアップグレード コードは、″アップグレード コード″ 設定に指定された値とまず一致している必要があります。一致するアップグレード コードが検出されると、指定の最小/最大バージョンや、そのバージョンを含めるか除外するか (″最小バージョンを含める″ と ″最大バージョンを含める″ 設定で指定) によって検索を絞り込むことができます。 この設定には、現在のバージョンの更新対象となる製品の最大バージョンが含まれています。設定には次の方法のいずれかで書き込むことができます: • システムの .msi または Setup.exe ファイルを参照してアップグレード パス ツリーに アップグレード エントリを追加すると、InstallShield は ″最大バージョン″ 設定にある .msi ファイルのバージョンを使用します。 • また、この設定に、0.00.0000 または 00.00.0000 (ゼロはバージョン番号を表す) の形式で最大バージョンを入力することもできます。 ***注意***• *バージョンの設定 (最小バージョン、最大バージョン、または両方) の値は、最低でも 1 つ指定する必要があります 。*

テーブル 3-10・アップグレード パスの設定 (続き)

設定	説明
最大バージョンを含める	アップグレードに、"最大バージョン" 設定で指定した最大バージョンを含めるかどうか指定します。 Windows Installer がターゲット システム上にインストールされている以前のバージョンの製品の最大バージョンを検索対象に含める場合は [はい] を選択します。
言語識別子 (ID)	"言語 ID" 設定は、Windows Installer によるアップグレード対象のインストール済みバージョンの検索をさらに詳しく定義します。以前にインストールされたバージョンのアップグレード コードは、"アップグレード コード" 設定に指定された値とまず一致している必要があります。一致するアップ グレードコードが検出されると、指定した言語や、それを含めるか除外するか ("言語検索基準" 設定で指定) などによって検索を絞り込むことができます。 この設定は、アップグレード対象の製品バージョン用の 10 進数の 言語 ID を識別します。 ***アップグレード対象の製品バージョン用の 10 進数の 言語 ID を提供するには、以下の手順に従います:*** 1. 省略記号ボタン (...) をクリックします。**[言語の選択]** ダイアログ ボックスが開きます。 2. このアップグレードで更新する各言語バージョンのチェック ボックスを選択します。 3. [OK] をクリックします。 別の方法として、この設定に 10 進数値 (複数可) を入力することもできます。二ヶ国語以上を入力する場合は、言語 ID をコンマ (,) で区切ります。 ***メモ:*** *システムの .msi ファイル参照して選択する方法で、[アップグレード パス] エクスプローラーに アップグレード エントリを追加すると、その .msi ファイルの 10 進数値による言語の値がデフォルトで自動的に "言語 ID" 設定に使用されます。*
言語検索基準	Windows Installer でターゲット システム上にインストールされている以前の製品バージョンを検索するときに、"言語識別子" 設定で指定されている言語識別子を検索対象に含めるかどうかを指定します。すべての言語を含める場合、この設定を空白のままにします。
削除エラーを無視する	ターゲット システムに既存する製品の以前のバージョンを削除できなかった場合に、Windows Installer が現在のバージョンのインストールを続行するかどうかを指定します。 [いいえ] を選択すると、ターゲット システムに既存する製品の以前のバージョンを削除できなかった場合に、Windows Installer は現在のバージョンのインストールを中止します。

テーブル 3-10 • アップグレード パスの設定 (続き)

設定	説明
機能の状態を移行する	アップグレード時に、インストール済みの製品バージョンから新しいバージョンに機能選択の状態を移行するかどうかを指定します。製品の前のバージョンをインストールしたときに、エンド ユーザーが特定の機能をインストールするように選択している場合があります。"機能の状態を移行する" 設定で [はい] を選択すると、製品の新しいバージョンにアップグレード中、以前に選択された機能がデフォルトで選択されます。 たとえば、元の製品に **Feature1**、**Feature2**、および**Feature3** が含まれていて、エンド ユーザーが Feature1 と Feature3 のみをインストールした場合、アップグレードが実行されたとき、インストールで **Feature1** と **Feature3** がデフォルトで選択されます。デフォルトで **Feature2** は選択されないままです。

[アップデート通知] ビュー

FlexNet Connect を利用して、Web に接続しているエンド ユーザーに対してアプリケーションのパッチ、アップデート、および製品情報が入手可能であることを自動的に通知します。[アップデート通知] ビューでは、元のインストールの自動通知を有効にすることができます。

FlexNet Connect の利用は簡単です。FlexNet Connect を有効にすると、InstallShield は Software Manager をインストールに含めます。このデスクトップ ツールはアプリケーションと一緒に発送されるので、エンドユーザーは、最新のアップデートを確認するツールとして利用することができます。エンドユーザーにアップデートを公開する場合、FlexNet Connect パブリッシャー サイトという名前の Web ベースの管理ポータルを使用します。

FlexNet Connect には様々なオプションがあり、完全ソリューションとして本製品と共に購入することもできますし、またはカスタマイズ ソリューションとして個別に購入することもできます。詳しい情報は、Revenera Web サイトをご覧ください。

[アプリケーション データの指定] ビュー

アプリケーション データには、インストールに追加するすべてのファイルが含まれます。ファイルは、[ファイル] ビューを使って、再配布可能ファイルを通して、または依存関係のスキャンを行って追加できます。

テーブル 3-11 • [アプリケーション データの指定] ビューの下にあるビュー

表示	説明
ファイル	[ファイル] ビューは、インストール プロジェクトにファイルを追加する主な方法です。このビューは、Windows エクスプローラーと同様に機能するため、インストール プロジェクトにファイルをドラッグ アンド ドロップすることができます。
ファイルおよび機能	ファイルに追加する機能は [ファイル] ビューで選択できますが、インストールにファイルを追加した後で、ファイルに関連付ける機能を変更するような場合には、[ファイルおよび機能] ビューで変更する必要があります。

テーブル 3-11 • [アプリケーション データの指定] ビューの下にあるビュー (続き)

表示	説明
再配布可能ファイル	再配布可能ファイル (InstallShield 前提条件、マージ モジュールおよびオブジェクト) を利用すると、インストール プロジェクトに個別の機能を追加することができます。このタイプの機能の例としては、Visual Basic のランタイム DLL および Microsoft C のランタイム ライブラリなどがあります。
依存関係	[依存関係] ビューには、依存関係をインストールに追加する 3 種類のスキャナーが含まれています。これらのスキャナーは、Visual Basic プロジェクトスキャナー、既にプロジェクトに含まれているファイルをスキャンしてその依存関係を追加するスタティックスキャナー、およびあらゆる依存関係について実行中のアプリケーションをスキャンするダイナミックスキャナーです。

[ファイル] ビュー

ファイルは、インストールの中心となります。インストール プログラムの主な目的は、ファイルをソース メディアからターゲットマシンのインストール先に転送することです。[ファイル] ビューは、4 つのペインに分けられます: 左の 2 つのペインにはフォルダーが含まれ、右の 2 つのペインには、これらのフォルダー内のファイルが表示されます。上記の 4 つのペインを [機能] リストと呼びます。ファイルをインストールに追加する前に、ファイルを関連付ける機能を選択する必要があります。ボックスに表示される機能は、ファイルが追加される機能です。

テーブル 3-12 • [ファイル] ビューにあるペイン

ペイン	説明
ソース コンピューターのフォルダー (左上)	**[ソース コンピューターのフォルダー]** ペインは、Windows エクスプローラーの左のペインに似ています。このペインには、ローカルまたはネットワーク上にあるフォルダーが含まれます。このペインで、インストールに追加するファイルを含むフォルダーに移動できます。
ソース コンピューターのファイル (右上)	**[ソース コンピューターのファイル]** ペインには、**[ソース コンピューターのフォルダー]** ペインで現在選択されているフォルダーに含まれるファイルが表示されます。このペインから、インストール先ペインの 1 つにファイルをドラッグできます。下部のペインにドロップしたファイルは、インストール プロジェクトに追加されます。

テーブル 3-12 • [ファイル] ビューにあるペイン (続き)

ペイン	説明
インストール先コンピューターのフォルダー (左下)	**[インストール先コンピューターのフォルダー]** ペインには、プロジェクトに追加済みのフォルダーだけでなく、定義済みのインストール先一覧が含まれています。 インストールがターゲット システム上に作成するカスタム フォルダーを追加するには、このビューでフォルダーを右クリックしてから、[新しいフォルダー] をクリックします。 一部の定義済みフォルダーは、デフォルトで非表示になっています。これらのフォルダーについての詳細は、「[ファイル] ビューで定義済みフォルダーを表示する」を参照してください。 ファイルとフォルダーをターゲット システム上の 64 ビットの場所に追加する処理については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。
インストール先コンピューターのファイル (右上)	**[インストール先コンピューターのファイル]** ペインには、現在選択されているインストール先のフォルダーに追加されているファイルがすべて表示されます。このペインでファイルを右クリックすると、ファイルの切り取り、コピー、貼り付け、削除、またはプロパティの編集を行うことができます。

次の 3 つのいずれかの方法を実行して、インストール プロジェクトにファイルを追加できます。各方法を以下に説明します。

テーブル 3-13 • プロジェクトへファイルを追加する方法

メソッド	説明
ドラッグアンドドロップ	ファイルをインストール プロジェクトに追加する最も簡単な方法は、[ファイル] ビューでペインを使用することです。このビューの上部の 2 つのペインは、機能的に Windows エクスプローラーと同じです。下部の 2 つのペインは、ファイルの宛先を示します。上部のペインから、下部のペインのインストール先フォルダーにソースファイルをドラッグするだけです。 ***メモ:*** *[ソースフォルダー] から [インストール先フォルダー] にフォルダーをドラッグアンドドロップすると、フォルダーをそのフォルダーのダイナミックリンクにするかどうかたずねるダイアログが表示されます。*
ダイナミックファイルリンク	ファイルをインストール プロジェクトに追加する別の方法は、全体のフォルダーの内容、またはフォルダーの特定のファイルにリンクすることです。この方法を使用すると、インストールのファイルを含む、ローカルまたはネットワーク上の特定のフォルダーを指定できます。インストールをビルドするたびに、選択されたフォルダーの内容が機能に追加されます。また、ワイルドカードを使用して、インストールに追加するファイルをフィルタリングできます。

テーブル 3-13 • プロジェクトへファイルを追加する方法 (続き)

メソッド	説明
依存関係 スキャン	ファイルをインストール プロジェクトに追加する 3 番目の方法は、[依存関係] ビューを使用することです。このビューには、インストール プロジェクトまたは実行中のアプリケーション内のすべての依存関係ファイルをスキャンして、それらをインストールに追加できるウィザードが含まれています。これらのウィザードは、[ツール] メニューから起動することもできます。

メモ• [ファイル] ビューのいずれかのペインを更新するには、更新するペインをクリックして、F5 を押します。

インストール先フォルダー

インストールにファイルを追加するには、インストール先フォルダーにそのファイルを配置します。次の定義済みインストール先フォルダーが、デフォルトとして [ファイル] ビューに表示されます。それぞれはダイナミックです。つまり、ハードコード化されたパスは必要ありません。その代わり、各インストール先のフォルダーの値は、ターゲット マシンのオペレーティング システムから取得されます。

一部の定義済みフォルダーは、デフォルトで非表示になっています。これらの非表示フォルダーについての詳細は、「[ファイル] ビューで定義済みフォルダーを表示する」を参照してください。

以下のプロパティは、エンド ユーザーのシステムの多くのフォルダーへの完全修飾パスを保持します。

テーブル 3-14 • 定義済みのインストール先フォルダー

プロパティ	説明
AdminToolsFolder	管理ツールのあるフォルダーを指します。
AppDataFolder	このプロパティは、現在のユーザーのアプリケーション データ フォルダーへの完全パスを保持します。デフォルトで、このプロパティは「none」になっています。
CommonAppDataFolder	これは、すべてのユーザーのアプリケーション データを含むフォルダーへの完全パスです。共通パスは、次のとおりです。 C:¥ProgramData
CommonFiles64Folder	このプロパティの値は、64 ビット Common Files フォルダーへの完全修飾パスです。 ターゲット システム上の 64 ビットの場所にインストールする処理については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。
CommonFilesArmFolder	このプロパティの値は、Arm¥Arm64 Common Files フォルダーへの完全修飾パスです。
CommonFilesFolder	このプロパティの値は、Common Files フォルダーへの完全修飾パスです。

テーブル 3-14・定義済みのインストール先フォルダー (続き)

プロパティ	説明
DATABASEDIR	このプロパティには、インストールのデータベース ファイルのインストール先が格納されます。[一般情報] ビューで DATABASEDIR の初期値を設定できます。また、[変更] ボタンを表示すると、エンドユーザーが実行時に [データベース フォルダー] ダイアログでこの値を修正できるようになります。
DesktopFolder	このプロパティは、現在のユーザーの [デスクトップ] フォルダーへの完全パスを保持するために使用します。セットアップが All Users に対して実行されているときに、ALLUSERS プロパティが設定されている場合、DesktopFolder プロパティには All Users デスクトップ フォルダーへの完全パスが保持されます。
FavoritesFolder	FavoritesFolder プロパティは現在のユーザーの[お気に入り] フォルダーまでの完全パスを含みます。
FontsFolder	このプロパティは、Fonts フォルダーへの完全パスを保持します。
GlobalAssemblyCache	このプロパティは、グローバル アセンブリ キャッシュ (GAC) への完全パスを含みます。
INSTALLDIR	このプロパティはインストールの保存先を格納します。[一般情報] ビューで、INSTALLDIR の初期値を設定することができます。
IISROOTFOLDER	このプロパティは、ターゲット システムの Web サーバーのルート ディレクトリを格納します。インストール プロジェクトで IIS 機能を使用していて、Web サイトが既に追加されている場合 IISROOTFOLDER は自動的に追加されます。 ***メモ:*** *[ファイル] ビューの IISROOTFOLDER ディレクトリに追加したファイルはすべて、ターゲット マシンの Web サーバーのルート ディレクトリにインストールされます。IIS がターゲットマシン上に存在しない場合、ファイルはルートフォルダーへコピーされます。*
LocalAppDataFolder	ローカルに保存されたアプリケーションデータの場所。このプロパティの通常の値は、次の通りです: C:¥Users¥*UserName*¥AppData¥Local
MyPicturesFolder	このプロパティは、現在のユーザーの [マイ ピクチャ] フォルダーへの完全パスを保持します。
PersonalFolder	このプロパティは、現在のユーザーの個人フォルダーへの完全パスを保持します。
PrimaryVolumePath	このプロパティは、インストールのプライマリ フォルダーを示す、PRIMARYFOLDER で指定されたパスに設定されます。

テーブル 3-14・定義済みのインストール先フォルダー (続き)

プロパティ	説明
ProgramFiles64Folder	このプロパティは、64 ビット Program Files フォルダーまでの完全パスを保持します。 ターゲット システム上の 64 ビットの場所にインストールする処理については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。
ProgramFilesArmFolder	このプロパティは、Arm¥Arm64 Proguram Files フォルダーへの完全修飾パスを保持します。
ProgramFilesFolder	このプロパティは、32 ビット Program Files フォルダーまでの完全パスを保持します。
ProgramMenuFolder	このプロパティは、現在のユーザーの [プログラム] メニューへの完全パスを保持するために使用されます。インストールが All Users に対して実行されているときに、ALLUSERS プロパティが設定されている場合、ProgramMenuFolder プロパティには All Users [プログラム] メニューへの完全パスが保持されます。
SendToFolder	このプロパティは、現在のユーザーの SendTo フォルダーへの完全パスを保持します。
StartMenuFolder	このプロパティは、現在のユーザーの [スタート] メニューへの完全修飾パスを保持するために使用されます。インストールが All Users に対して実行されているときに、ALLUSERS プロパティが設定されている場合、StartMenuFolder プロパティには All Users [プログラム] メニューへの完全修飾パスが保持されます。
StartupFolder	このプロパティは、現在のユーザーの [スタートアップ] フォルダーへの完全修飾パスを保持するために使用されます。セットアップが All Users に対して実行されているときに、ALLUSERS プロパティが設定されている場合、StartupFolder プロパティには All Users [プログラム] メニューへの完全パスが保持されます。
System16Folder	このプロパティは、システムの 16 ビット DLL を含むフォルダーまでの完全パスを保持します。
System64Folder	このプロパティは、64 ビット Windows System フォルダーまでの完全パスを保持します。 ターゲット システム上の 64 ビットの場所にインストールする処理については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。
SystemArmFolder	このプロパティは、Arm¥Arm64 Windows システム フォルダーへの完全修飾パスを保持します。
SystemFolder	このプロパティは、32 ビット Windows System フォルダーまでの完全パスを保持します。

テーブル 3-14・定義済みのインストール先フォルダー (続き)

プロパティ	説明
TempFolder	このプロパティは、Temp フォルダーまでの完全パスを保持します。
TemplateFolder	このプロパティは、現在のユーザーの [テンプレート] フォルダーへの完全パスを保持します。
WindowsFolder	このプロパティは、Windows フォルダーへの完全パスを保持します。
WindowsVolume	このプロパティは、Windows フォルダーのボリュームを保持します。このプロパティは Windows がインストールされているドライブに設定されます。

[ファイルおよび機能] ビュー

ファイルを機能に関連付けるのに最適なタイミングは、セットアップ プロジェクトにこれらのファイルを初めて追加するときです。ファイルを追加した後、そのファイルが関連付けられた機能を変更する場合は、[ファイルと機能] ビューを通してのみ行なうことができます。

タスク ***ファイルが関連付けられた機能を変更するには:***

1. 現在ファイルを含む機能をクリックして、その機能に関連付けられている全ファイルのリストを表示します。
2. 移動するファイルを選択して、新しい機能にドラッグします。

メモ: *ドラッグ アンド ドロップではなく、大きなグループとしてまたは個別に、ファイルを切り取って貼り付けることもできます。*

[再配布可能ファイル] ビュー

[再配布可能ファイル] ビューには、InstallShield に含まれているすべての InstallShield 前提条件、マージ モジュール、およびオブジェクトが含まれています。

InstallShield 前提条件

InstallShield 前提条件 は、製品が必要とする製品、またはテクノロジ フレームワークのためのインストールです。既存の InstallShield 前提条件は、どれでもインストール プロジェクトに追加することができます。

プロジェクトに InstallShield 前提条件を含めると、複数のインストールを連鎖することができるため、1 度に 1 つの実行シーケンスのみしか実行できない Windows Installer 制限をバイパスできます。Setup.exe セットアップランチャーは、連鎖を管理するブートストラップ アプリケーションとしての役割を果たします。

エディション: *InstallShield Premier および InstallShield には、InstallShield 前提条件エディターが含まれています。これを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。*

InstallShield では、次の 2 つのタイプの InstallShield 前提条件がサポートされています:

- **セットアップ前提条件** – この種類の前提条件のインストールは、インストールの実行の前に実行されます。
- **機能前提条件** – この種類の前提条件は、1 つまたは複数の機能に関連付けられています。機能前提条件は、前提条件を含む機能がインストールされたときに、その前提条件がシステム上に既にインストールされていない場合にインストールされます。したがって、機能の条件がターゲット システム上で満たされていない場合、またはエンド ユーザーが機能のインストールを行わないことを選択した場合、その機能はインストールされません。その結果、インストールされる別の機能にも機能前提条件が関連付けられていない限り、関連付けられた機能前提条件はいずれもインストールされません。

マージ モジュール

マージ モジュール (.msm ファイル) には、個別機能をインストールするために必要なロジックとファイルのすべてが含まれています。たとえば、多くのアプリケーションには Microsoft Visual Basic ランタイム .dll が必要です。機能にファイルを含めてインストール要件を調べる必要はなく、プロジェクトに含まれる機能の 1 つに Visual Basic 仮想マシン マージ モジュールを添付するだけでこれを実行できます。

メモ: *[再配布可能ファイル] ビューに含まれているマージ モジュールの多くは Microsoft またはその他のサードパーティによるものです。InstallShieldでは、これらのモジュールを無料配布することによって、インストール プロジェクトの作成を支援します。ただし、サードパーティが作成したモジュールに存在する問題を InstallShield が修正したり直すことはできません。サード パーティが作成したモジュールに関する問題は、ベンダーへお問い合わせください。*

オブジェクト

オブジェクトには、マージ モジュール同様、個別の機能をインストールするために必要なロジックとファイルがすべて含まれています。InstallShield に含まれる DirectX オブジェクトなどのオブジェクトは、ウィザードを使ってカスタマイズする必要があります。オブジェクトをインストールに追加すると、すぐに該当するカスタマイズ ウィザードが開きます。オブジェクトを追加時点でカスタマイズすることも、ウィザードをいったんキャンセルし、後でオブジェクトを右クリックして [オブジェクトの設定変更] を選択することによってカスタマイズすることもできます。

ライブ再配布可能ファイルギャラリー

多くの再配布可能ファイルはサイズが大きいため、プロジェクトで利用可能なものでも InstallShield のインストールと同時にコンピューターへ追加されない場合があります。その場合も、これらの再配布可能ファイルはインターネットからコンピューターへダウンロードすることができます。

構成可能マージ モジュール

構成可能な再配布可能ファイルは、マージ モジュールまたは ModuleConfiguration テーブルで少なくとも 1 つの行を持ち、ModuleSubstitution テーブルで少なくとも 1 行によって参照されるオブジェクトです。これによって再配布可能ファイルの値を変更することができます。[再配布可能ファイル] ビューで構成可能モジュールを選択した場合、表示される [マージ モジュール構成可能値] ダイアログ ボックスでモジュール追加時にそれを構成することができます。マージ モジュールを後でカスタマイズするには、それを右クリックして **[マージ モジュールの構成]**を選択します。

[再配布可能ファイル] ビューを使って作業する

[再配布可能ファイル] ビューは、次の要素で構成されます:

- ボタン行とその他のコントロール
- グループ ボックス領域 (ボタン行の下)
- 再配布可能ファイルの一覧
- 選択された再配布可能ファイルついての情報を表示する、詳細ペイン

次の表では、[再配布可能ファイル] ビューに表示されるすべてのボタンと、その他のコントロールについて説明します。

テーブル 3-15 ・ [再配布可能ファイル] ビューのコントロール

コントロールの名前	アイコン	説明
すべてのグループを展開する		グループを使って行を階層形式に編成する場合に、グループに含まれるすべての行を表示します。
すべてのグループを折りたたむ		グループを使って行を階層形式に編成する場合に、グループに含まれるすべての行を隠します。
詳細の表示		提供されている再配布可能ファイルの一覧から選択された InstallShield 前提条件、マージ モジュール、またはオブジェクトの詳細を表示するビュー内の右側にあるペインの表示/非表示を切り替えます。この詳細ペインで、再配布可能ファイルがインストールするファイルなどの情報を確認することができます。
グループ ボックスの表示		このビューのボタン行の下にある [グループ ボックス] 領域の表示/非表示を切り替えます。
リフレッシュ		再配布可能ファイルのリストを更新します。
文字列のフィルター条件を入力		この検索ボックスで指定した文字列に従って、[再配布可能ファイル] ビューに表示される再配布可能ファイルをダイナミックにフィルターします。このボックスに文字列を入力すると、それを含まない行すべてが非表示となります。
[再配布可能ファイル] ビュー ヘルプ		[再配布可能ファイル] ビューのヘルプを表示します。
グループ化する列のヘッダーをここにドラッグ		このグループ ボックス領域を使って、ビュー内の行をグループ分けします。ビューでは、列のヘッダーをグループ ボックスにドラッグ アンド ドロップするだけで複数階層にグループ化することができます。ビュー内の行は、グループ ボックスの配列に従って階層構造で表示されます。 詳細については、「様々なビューで、[グループ ボックス] 領域を使って作業する」を参照してください。

次の表は、[再配布可能ファイル] ビューの各列について説明します。

テーブル 3-16 ・ [再配布可能ファイル] ビューの列

列	説明
チェック ボックス	この列には、各再配布可能ファイルのチェック ボックスが表示されます。再配布可能ファイルをプロジェクトに追加するには、そのチェック ボックスを選択します。
タイプ/状態	この列には、再配布可能ファイルの種類を示すアイコンと、再配布可能ファイルの状態が表示されます。各アイコンの詳細については、「再配布可能ファイルギャラリーを管理する」を参照してください。
名前	この列には、各再配布可能ファイルの名前が表示されます。
バージョン	この列には、再配布可能ファイルのバージョン番号が表示されます。
種類	この列には、再配布可能ファイルの種類がリストされます。
場所	この列には、再配布可能ファイルが使用中のマシン上でローカルにインストール済みか、ダウンロードが必要であるかが示されます。詳細については、「再配布可能ファイルをコンピューターにダウンロードする」を参照してください。

[依存関係] ビュー

あるファイルが、他のファイルの関数に依存してタスクを実行することがよくあります。ただし、インストールプロジェクトにアプリケーションファイルを含める際に、*「依存関係」*と呼ばれる他のファイルに気が付かない場合があります。InstallShieldでは、これらのファイルを見つけて作業を行う次のスキャン ウィザードが用意されています。これらのスキャナーには、[依存関係] ビューからアクセスできます:

テーブル 3-17 ・ InstallShield の依存関係スキャナー

スキャンのオプション	関数
スタティック スキャンの実行	スタティック スキャン ウィザードは、プロジェクトのポータブル実行可能ファイル (例、.exe、.ocx、.com、.tlb、.hlp、および .chm) を探し、必要な依存関係を検出します。
ダイナミック スキャンの実行	ダイナミック スキャン ウィザードは、実行可能ファイルが実行中にシステムを監視して、実行可能ファイルで必要となる可能性がある .dll または .ocx ファイルを確認します。

スタティックおよびダイナミック スキャン ウィザードを使用する場合、InstallShield を使ってスタティックまたはダイナミック スキャンを実行するとき必ず自動的に選択または除外されるファイルを指定することができます。詳細については、「依存関係スキャナーでファイルをフィルターする」を参照してください。

[ターゲット システムの構成] ビュー

インストールでは、ターゲット システムの変更が必ず発生します。最も単純なインストールでは、単にファイルがコピーされるだけです。より複雑なインストールでは、レジストリの変更、.ini ファイルの編集、ショートカットの作成、その他の変更が行われます。InstallShield には、詳細設定用に次のようなサブビューがあります。

テーブル 3-18 ・ [ターゲット システムの構成] ビューの下にあるビュー

表示	説明
ショートカット/フォルダー	ショートカットを使用すると、インストールしたアプリケーションにすばやくアクセスできます。ショートカットおよびフォルダーは、デスクトップ上や [スタート] メニュー、その他さまざまな場所に作成することができます。
レジストリ	[レジストリ] ビューでは、レジストリ エントリを作成したり、既存のレジストリ データをインストールにインポートすることができます。
ODBC リソース	[ODBC リソース] ビューでは、ドライバー、トランスレーターおよびデータ ソースをアプリケーションの 1 つまたは複数の機能に追加できます。インストールされた ODBC リソースを選択するか、または新しいリソースをリストに追加し、それらを機能に関連付けた後、その属性をカスタマイズします。
INI ファイルの変更	このビューを使って、ターゲット システム上の .ini ファイルを編集します。
ファイル拡張子	製品にファイル拡張子を関連付ける必要がある場合は、[ファイル拡張子] ビューで関連付けを作成できます。
環境変数	ターゲット システムの環境変数を作成または変更したり、ターゲット システムから環境変数を削除する場合、[環境変数] ビューでそのプロパティを定義付けることができます。
インターネット インフォメーション サービス	[IIS 構成] ビューを利用すると、ターゲット システム上での IIS Web サイト、アプリケーション、および仮想ディレクトリの作成並びに管理が可能です。
コンポーネント サービス	[コンポーネント サービス] ビューでは、インストール パッケージ用の COM+ サーバー アプリケーションとコンポーネントを管理できます。
サービス	[サービス] ビューを使って、インストール中にインストール、およびアンインストール中に削除するサービスの情報を指定します。

[ショートカット/フォルダー] ビュー

[ショートカット/フォルダー] ビューは、ターゲット システム上のアプリケーションへのショートカットを作成するためのシンプルかつビジュアルな方法を提供します。このビューには、ショートカットとサブフォルダーを作成できる一組の定義済みのインストール先フォルダーを表示する [ショートカット] エクスプローラーがあります。InstallShield には標準で次のショートカットのインストール先があります。

テーブル 3-19・ショートカットの標準インストール先

ショートカットのインストール先	説明
[プログラム] メニューと [スタートアップ]	[プログラム] と [スタートアップ] フォルダーは、[スタート メニュー] の中にあります。[プログラム メニュー] フォルダーは、製品のショートカットをインストールする業界標準の場所となっています。[スタートアップ] フォルダーには、Windows が起動するときに常に起動する必要があるアイテムだけを配置します。
SendTo	[送る] フォルダーは、ユーザーがファイルを右クリックしたときに利用できます。このコンテキスト メニューには [送る] と呼ばれるコマンドが含まれています。このフォルダーにプログラムのショートカットを作成した場合、エンドユーザーは任意のファイルをクリックして製品に送ることができます。 たとえば、エンド ユーザーがメモ帳で HTML ページを開けるようにしたいとします。[ショートカット/フォルダー] ビューの [送る] フォルダーに、メモ帳へのショートカットを作成すると、エンド ユーザーは HTML ファイルを右クリックしたときに [送る] メニューからメモ帳をクリックすることができます。このページのソースファイルがメモ帳で開かれます。 ***メモ:*** *Send to で作成されたショートカットは、インストールを実行中のユーザーの Send to に設定されます。これは Windows Installer が [送る] プロパティを現在のユーザーの [送る] フォルダーへの絶対パスに設定するためです。* *Send to フォルダーのショートカットはアドバタイズできません。*
デスクトップ	[デスクトップ] フォルダーには、エンド ユーザーのデスクトップのショートカットが含まれています。デスクトップ フォルダー上にショートカットを作成すると、エンドユーザーのデスクトップに製品のアイコンが表示されます。 ***メモ:*** *デスクトップはショートカットの配置場所としては一番わかりやすい場所ですが、ショートカットを多く配置しすぎるとエンド ユーザーのデスクトップが混雑する可能性があります。*

新しいショートカットを追加すると、このビューの [ショートカット] エクスプローラーに追加されたショートカットが新しいノードとして追加されます。新しいショートカット ノードは、そのショートカットがターゲット システム上に既に存在するファイルに割り当てられていない場合、ショートカットの "アイコン ファイル" 設定で選択されたアイコン (実行時に、ショートカットがターゲット システムに追加されたときに使用されます) と同じアイコンと共に表示されます。

既に存在するファイルへのショートカットの場合、アイコン ファイルは実行時まで判明しません。したがって、[ショートカット] エクスプローラーでは、ターゲット システム上で実行時に使用されるアイコンの代わりに、各ショートカットに次のアイコンが表示されます。

アイコン ファイルがアイコンを含まない場合、このアイコンが [ショートカット] エクスプローラーのショートカットに使用されます。

メモ: *動的にリンクされたファイルへのショートカットを作成することはできません。詳細については、「ダイナミック ファイル リンクの制限事項」を参照してください。*

ショートカットとフォルダーについて構成可能な各設定についての詳細は、以下を参照してください:

- ショートカットの設定
- フォルダーの設定

ショートカットの設定

[ショートカット/フォルダー] ビューで、ショートカットの設定は次のメイン カテゴリで構成されています:

- 概観
- 動作
- 全般

概観

ショートカットの ″外観″ 設定を使って、ショートカットの説明やアイコンなどの詳細を指定します。

テーブル 3-20 ・外観の設定

設定	説明
表示リソース	複数言語アプリケーション用のインストールを準備中で、ショートカットの表示名用 .mui ファイルと言語ニュートラルなポータブル実行可能ファイルを切り離す場合、以下の設定を使用します: ・ **表示リソース** – 複数言語ユーザー インターフェイス (MUI) マニフェストを含む DLL ファイルを参照するには、この設定で省略記号ボタン (...) をクリックします。 DLL ファイルを参照してから選択するか、″表示リソース DLL ″ 設定でパスとファイル名を手入力した場合、InstallShield はこの設定にパスとファイル名をリストします。″表示リソース インデックス″ 設定で指定されたリソース インデックスもリストします。 ・ **表示リソース DLL** – MUI マニフェストを含む DLL ファイルのパスとファイル名を手動で指定する場合は、それを入力します。ハードコード化されたディレクトリ パスの代りに、Windows Installer ディレクトリ プロパティ (例、**[INSTALLDIR]MyResource.dll**) を含めることができます。 ″表示リソース″ 設定で省略記号ボタンをクリックして DLL ファイルを参照した場合、InstallShield は ″表示リソース DLL″ 設定で [#filekey] 形式を使って DLL ファイルを識別します。 ・ **表示リソース インデックス** – ショートカットの表示名インデックスを指定します。この値に、負の数値は指定できません。 ***メモ:*** *DLL を指定する場合、″表示リソース インデックス″ 設定の値も入力しなくてはなりません。* これらの設定により、言語依存ファイルをすべて含む .mui ファイルから、言語ニュートラルなポータブル実行可能ファイルを分離することができるため、後で追加言語のリソースを追加する際、アプリケーションの再コンパイルや再リンクの必要がなくなります。 Windows Vista 以降および Windows Server 2008 以降は、表示リソースをサポートします。それ以前のシステムは、これを無視します。
説明	ショートカットの説明を入力します。入力するテキストは、エンド ユーザーがショートカットの上にマウス ポインターを置いたときにツールヒントとして表示されます。また、ショートカットの [プロパティ] ダイアログ ボックスの ″説明″ フィールドにも表示されます。

テーブル 3-20・外観の設定 (続き)

設定	説明
説明リソース	複数言語アプリケーション用のインストールを準備中で、ショートカットの説明用 .mui ファイルと言語ニュートラルなポータブル実行可能ファイルを切り離す場合、以下の設定を使用します: ・ **説明リソース** – 複数言語ユーザー インターフェイス (MUI) マニフェストを含む DLL ファイルを参照するには、この設定で省略記号ボタン (...) をクリックします。 DLL ファイルを参照してから選択するか、"説明リソース DLL " 設定でパスとファイル名を手入力した場合、InstallShield はこの設定にパスとファイル名をリストします。"説明リソース インデックス" 設定で指定されたリソース インデックスもリストします。 この設定に値が含まれている場合、"説明" 設定の値は無視されます。この設定を空白のままにしておくと、Windows Installer は "説明" 設定の値を使用します。 ・ **説明リソース** – DLL MUI マニフェストを含む DLL ファイルのパスとファイル名を手動で指定する場合は、それを入力します。ハードコード化されたディレクトリ パスの代りに、Windows Installer ディレクトリ プロパティ (例、**[INSTALLDIR]MyResource.dll**) を含めることができます。 "説明リソース" 設定で省略記号ボタンをクリックして DLL ファイルを参照した場合、InstallShield は "説明リソース DLL" 設定で [#filekey] 形式を使って DLL ファイルを識別します。 ・ **説明リソース インデックス** – ショートカットの説明インデックスを指定します。この値に、負の数値は指定できません。 ***メモ:*** *DLL を指定する場合、"説明リソース インデックス" 設定の値も入力しなくてはなりません。* これらの設定により、言語依存ファイルをすべて含む .mui ファイルから、言語ニュートラルなポータブル実行可能ファイルを分離することができるため、後で追加言語のリソースを追加する際、アプリケーションの再コンパイルや再リンクの必要がなくなります。 Windows Vista 以降および Windows Server 2008 以降は、説明リソースをサポートします。それ以前のシステムは、これを無視します。

テーブル 3-20・外観の設定 (続き)

設定	説明
アイコン	ショートカットに表示するアイコンを指定するには、以下の設定を使用します: ・ **アイコン** – 作成するショートカットのアイコンを含むファイルを指定します。アイコン リソースを含む .ico ファイル、または実行可能ファイル (.dll または .exe) を指定する必要があります。この設定で表示されている省略記号ボタン (...)ファイルをクリックしてファイルを選択するか、"アイコン ファイル" 設定でパスとファイル名を手動で入力した場合、この設定にアイコン パスとインデックスがリストされます。 ・ **アイコン ファイル** – アイコンを含むファイルのパスとファイル名を手動で指定する場合は、それを入力します。 ・ **アイコン インデックス** – 指定したアイコン ファイルに 1 つ以上のアイコン リソースがある場合、この設定にインデックスを入力します。 負の数以外の整数を指定すると、実行可能ファイルのアイコン リソースの順番が参照されます。たとえば、0 はファイル内の最初のアイコン、1 は 2 番目のアイコン、2 は 3 番目のアイコンを参照します。 負の数字は特定のリソース ID を参照するために使用します。たとえばアイコン インデックス -12 は、リソース ID が 12 のアイコンを示します。

動作

ショートカットの "動作" 設定を使って、ターゲットやキーボード ショートカットなどの詳細を指定します。

テーブル 3-21・動作の設定

設定	説明
ターゲット	エンド ユーザーがショートカットを起動したときに起動される、ターゲット システム上にあるファイルへのパスを入力します。ハードコード化されたディレクトリ パスではなく、Windows Installer ディレクトリ プロパティ (例、[INSTALLDIR]File.exe) を使います。値をテ入力する代りに、省略記号ボタン (...) をクリックしてショートカット ターゲットを参照することもできます。

テーブル 3-21・動作の設定 (続き)

設定	説明
引数	このショートカット用のコマンドライン引数を入力します。引数が、ターゲット システム上でショートカットの [プロパティ] ダイアログ ボックスの "ターゲット" 値に追加されます。これらの引数は、他のコマンドライン引数と同様に機能します。たとえば、1 つのファイルを実行可能ファイルにリンクさせたり、またはコマンドライン引数を渡して実行可能ファイルをサイレントに実行するように設定できます。 ***メモ:*** *InstallShield では構文の確認が行われないので、構文が正しいことを確認してください。* ***ヒント:*** *選択されたファイル名の引数には* **%1** *を使用してください。たとえば、エンドユーザーが* C:¥File.ext *を右クリックして、このショートカットの引数が* –p %1 *と想定します。この場合、コマンドラインの引数は* –p C:¥File.ext *になります。場合によっては、スペースを含むファイル名を正しく処理するために、"%1" のように %1 引数を引用符で囲む必要があります。*
作業ディレクトリ	ショートカット ターゲットの作業ディレクトリを入力するか、省略記号ボタン (...) をクリックしてディレクトリを選択または作成します。 指定したディレクトリは、ターゲット システム上でショートカットの [プロパティ] ダイアログ ボックスの **"作業フォルダー"** フィールドに表示されます。作業ディレクトリは、製品が使用する現在のディレクトリであるだけでなく、標準の [ファイルを開く] および [ファイルの保存] ダイアログ ボックスで表示されるデフォルトのディレクトリでもあります。 ***プロジェクト:*** *たとえば、作業ディレクトリを* Files *という* INSTALLDIR *のサブディレクトリに設定する場合は、一覧から [INSTALLDIR] を選択して、その末尾に* Files *を追加します。設定が完了すると、[INSTALLDIR]Files と表示されます。*

テーブル 3-21 • 動作の設定 (続き)

設定	説明
ホット キー	この設定は、ショートカットに割り当てられたキーボード ショートカットの 10 進数値を含みます。キーボード ショートカットを製品のショートカットに割り当てると、エンド ユーザーは適切なホット キーを押してショートカットを起動できます。 InstallShield を使って、キーボード ショートカットの 10 進数値を自動的に計算するには、この設定で省略記号ボタン (...) をクリックします。 詳細については、「ショートカットにアクセスできるキーボード ショートカットを指定する」を参照してください。 ***注意:*** *ターゲット システム上の既存のキーボード ショートカットと競合する可能性があるため、ショートカットのキーボード ショートカットを構成することは避けることをお推めします。*
実行	エンド ユーザーがショートカットを起動したときに、ターゲット ファイルが使用するウィンドウのスタイルを選択します。選択可能なオプションは以下のとおりです: • **標準ウィンドウ** – ファイルを標準サイズのウィンドウで起動します。 • **最大ウィンドウ** – プログラムを全画面表示で起動します。 • **最小ウィンドウ** – ファイルが最小ウィンドウで起動し、タスクバーだけに表示されます。
管理者として実行	”管理者として実行” へのショートカットの設定を選択します。 選択可能なオプションは以下のとおりです: • **はい**– [管理者として実行] へのショートカットの設定を有効化します。 • **いいえ**– [管理者として実行] へのショートカットの設定を行ないません。 ***メモ:*** *ショートカットを [管理者として実行] を使って設定するためには、インストーラーを昇格モードで実行する必要があります。* *アドバタイズ ショートカットおよびインターネット ショートカットの場合、[管理者として実行] は無効化されます。*

全般

ショートカットの "全般" 設定を使って、ショートカットを含む機能などの詳細を指定します。

テーブル 3-22 ・ 全般設定

設定	説明
機能	この設定は、ショートカットが関連付けられている機能をすべて示します。
コメント	このショートカットに関するコメントを入力します。コメントは参考のためにプロジェクトファイルに保存され、エンド ユーザーへは表示されません。

フォルダーの設定

[ショートカット/フォルダー] ビューに作成する新しいフォルダーには、以下の設定を構成できます。

テーブル 3-23 ・ フォルダーの設定

設定	説明
説明	フォルダーの説明を入力します。InstallShield がこのフォルダーを [常にインストール] 機能に追加します。したがって、このフォルダーは、それに含まれるショートカットがインストールされない場合でも常にインストールされます。

[タイル構成] の設定

タイルの構成の設定は、次のメイン カテゴリに分かれています:

- 全般
- 外観

全般

タイル構成の "全般" 設定を使って、タイルに関連付けられている .exe ファイルのパスを確認します。

テーブル 3-24 ・ 全般設定

設定	説明
ターゲット	タイルに関連付けられている .exe ファイルのパス (たとえば、**[INSTALLDIR]File.exe**) が表示されます。

外観

タイル構成の ″外観″ 設定を使って、タイルの背景色などの詳細を指定します。

テーブル 3-25 ・外観の設定

設定	説明
テキスト色	中サイズのタイルでアプリケーション名を表示する際に使う前景テキスト色を選択します。選択可能なドロップダウン オプション: ・ **明色**—中サイズのタイルに表示するテキストの色を明色 (白いテキスト) に設定します。 ・ **暗色**—中サイズのタイルに表示するテキストの色を暗色 (黒いテキスト) に設定します。 ***メモ・*** *中サイズのタイル (150x150) のみ、タイル上に名前を表示できるスペースがあります。また、**″テキスト色″** 設定を反映させるためには、**″中タイルに名前を表示する″** 設定で **[はい]** を選択する必要があります。*
背景色	タイルの背景色に使う色を選択します。いくつかの定義済みの色を提供するドロップダウン リストから選択するか、省略記号ボタン (...) を使ってその他の色を選択、または RGB 16 進文字列の色を指定できます。
小さいタイル イメージ	小サイズ (70x70) タイルの背景イメージへの相対パスを指定します。 ***重要・*** *カスタム イメージを選択する場合、小タイル イメージ (70x70) および中タイル イメージ (150x150) の両方のイメージを指定し、手動でプロジェクトに追加する必要があります。1 つだけしか指定しなかった場合、Windows の規定のスタイルがアプリケーション アイコンおよび背景がタイルに適用されます。また、サイズが固定されたイメージ、ハイコントラストなイメージ、またはローカライズされたイメージを使用する場合、手動で resources.pri ファイルを作成して、アプリケーションの .exe ファイルおよびその .visualelementsmanifest.xml ファイルと同じディレクトリに配置しなくてはなりません。イメージの使い方についての詳細、または resources.pri ファイルの作成方法については、次の MSDN 記事を参照してください: Desktop App Tiles on the Start Screen*

テーブル 3-25・外観の設定 (続き)

設定	説明
普通サイズのタイル イメージ	中サイズ (150x150) タイルの背景イメージへの相対パスを指定します。 ***重要・*** *カスタム イメージを選択する場合、小タイル イメージ (70x70) および中タイル イメージ (150x150) の両方のイメージを指定し、手動でプロジェクトに追加する必要があります。1 つだけしか指定しなかった場合、Windows の規定のスタイルがアプリケーション アイコンおよび背景がタイルに適用されます。また、サイズが固定されたイメージ、ハイコントラストなイメージ、またはローカライズされたイメージを使用する場合、手動で resources.pri ファイルを作成して、アプリケーションの .exe ファイルおよびその .visualelementsmanifest.xml ファイルと同じディレクトリに配置しなくてはなりません。イメージの使い方についての詳細、または resources.pri ファイルの作成方法については、次の MSDN 記事を参照してください: Desktop App Tiles on the Start Screen*
中タイルに名前を表示する	アプリケーション名を中サイズ (150x150) タイルに表示するかどうかを指定します。選択可能なオプションは以下のとおりです: ・ **はい**—中サイズ タイルが使用された場合、タイル上にアプリケーションの名前を表示します。 ・ **いいえ**—タイル上にアプリケーション名を表示しません。 ***メモ・*** *この設定は、小サイズ タイルの使用時には効果がありません。*

[レジストリ] ビュー

InstallShield の [レジストリ] ビューによって、エンド ユーザーのレジストリを変更する作業が簡単になりました。Windows のレジストリ エディターと似た要領で、このビューを使用してキーおよび値を作成します。

さらに、"一般情報" 設定 に指定した値に基づいて、特定のレジストリ エントリも自動的に作成されます。これらの「情報キー」は、Windows ロゴガイドラインで必要となります。

インストール プロジェクトのレジストリ データすべてを表示するには、ビューの上部にある [機能] リストから [すべてのアプリケーションデータ] を選択します。このオプションが選択されている場合、レジストリデータを変更または削除することはできますが、新しいキーまたはバリューを追加することはできません。

ヒント・ *64 ビット インストールを作成していて、レジストリ エントリを 64 ビット レジストリの場所 (HKEY_LOCAL_MACHINE¥Software¥Wow6432Node ではなく、HKEY_LOCAL_MACHINE¥Software の下) にインストールしたい場合、そのエントリを SOFTWARE (64 ビット) ノード、またはそのサブノードに追加してください。64 ビット インストールを 32 ビット ターゲット システムで実行することはできません。詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。*

64 ビット システム上で 32 ビット アプリケーションによるレジストリの表示を確認するには、32 ビット バージョンのレジストリ エディター (SysWOW64 フォルダーの regedit.exe ファイル) を起動します。

[ODBC リソース] ビュー

システム構成のより複雑な領域の 1 つに、ODBC ドライバー、データソース名 (DSN)、トランスレーターの設定があります。ODBC リソースは、すべての必須属性とともにシステムに正しく登録されている必要があります。また、ドライバーおよびトランスレーターの場合、セットアップ DLL などの必要なファイルをインストールする必要があります。このプロセスは、[ODBC リソース] ビューで大幅に簡略化できます。このビューでは、開発システムにインストールされているドライバー、データソース、トランスレーターのチェックマークを外すことができます。

すべての ODBC ドライバー、データソース名 (DSNs)、およびシステムに登録されているトランスレーターは、[ODBC リソース] ビューで表示されます。DSN は、関連ドライバーの「子」として表示されます。ツリーを展開すると、既存の ODBC リソースがすべて表示されます。セットアップ プロジェクトのほとんどのデータと同様に、ODBC リソースも機能に関連付ける必要があります。機能がインストールされると、ODBC リソースは、全体の機能の一部としてインストールされます。

ヒント: *[ODBC リソース] ビューは、ODBC 関連リソースのインストール専用です。コア ODBC ファイルをインストールするには、[再配布可能ファイル] ビューで MDAC マージ モジュールを選択します。*

ODBC リソースの設定

ODBC リソース属性、および利用できる値のユニバーサル リストはありません。インストール用に指定するものがわからない場合は、ファイルのベンダーまたは作成者にお問い合わせください。ただし以下に示すように、ODBC リソースの各種類で共通する属性がいくつかあります。

メモ: *独自の ODBC 属性を追加するには、[ODBC リソース] ビューの最終行をクリックしてプロパティと値を指定します。*

ドライバー

以下の必須属性のほかに、ドライバーにはツリー上で表示する名前が必要です。この名前は、ドライバーの説明として登録されます。ドライバーの名前は、ローカライズできません。つまり、[テキストとメッセージ] ビューを介してこの名前を翻訳することはできないので、システムの言語に関係なく同じ名前を使用する必要があります。

テーブル 3-26 · ODBC ドライバー属性

属性	予期される値
ドライバー DLL	開発システム上にある DLL ファイルで、ODBC ドライバーとして機能するファイルへのパスを入力するか、省略記号ボタン (...) をクリックして、ファイルを参照します。
セットアップ DLL	開発システム上のセットアップ DLL へのパスを入力するか、省略記号ボタン (...) をクリックして、ファイルを参照します。ドライバー ファイルもセットアップ DLL である場合、この設定を空白のまま残します。

データ ソース

以下の必須属性のほかに、DSN にはツリー上で表示する名前が必要です。この名前は、DSN の説明として登録されます。DSN の名前は、ローカライズできません。つまり、[テキストとメッセージ] ビューを介してこの名前を翻訳することはできないので、システムの言語に関係なく同じ名前を使用する必要があります。

テーブル 3-27 • ODBC データソース属性

属性	予期される値
登録	システム データ ソースまたはユーザー データ ソースのどちらを使用するか指定します。 • **システム データ ソース** – データ ソースは、システムのすべてのユーザーが使用できます。 • **ユーザー データ ソース** – データ ソースは、現在のユーザーにだけ登録されます。

トランスレーター

以下の必須属性のほかに、トランスレーターにはツリー上で表示する名前が必要です。この名前は、トランスレーターの説明として登録されます。トランスレーターの名前は、ローカライズできません。つまり、[テキストとメッセージ] ビューを介してこの名前を翻訳することはできないので、システムの言語に関係なく同じ名前を使用する必要があります。

テーブル 3-28 • ODBC トランスレーター属性

属性	予期される値
トランスレーター DLL	開発システム上にある DLL ファイルで、トランスレーターとして機能するファイルへのパスを入力するか、省略記号ボタン (...) をクリックして、ファイルを参照します。
セットアップ DLL	開発システム上のセットアップ DLL へのパスを入力するか、省略記号ボタン (...) をクリックして、ファイルを参照します。

メモ: *トランスレーターに属性を追加することはできません。*

[INI ファイルの変更] ビュー

ターゲット システム上の .ini ファイル、特に Boot.ini といった、オペレーティング システムが使用する .ini ファイルを編集することは危険です。InstallShield では、.ini ファイルの変更を InstallShield インターフェイス上で正しく行なうことができます。.ini ファイルの編集には、以下に説明する 3 つのタスクがあります。

テーブル 3-29 ・ .ini ファイルの編集に関するタスク

タスク	説明
.ini ファイルの追加	.ini ファイル変更作成の最初の手順は、編集するファイルへのリファレンスを作成することです。リファレンスを作成するには、編集するファイルの名前および場所を知っている必要があります。指定した場所にファイルがない場合、ファイルを変更することはできません。.ini ファイルをインポートすることもできます。
.ini ファイルのセクションを指定する	編集する.ini ファイルを指定したら、次の手順で変更するファイルのセクションを指定できます。.ini ファイルは、セクションに分割され、それぞれのセクションにはキーワードがあります。セクションは、[SectionName] のように角かっこ [ ] で囲まれて識別されます。
.ini ファイルのキーワードとその値を指定する	INI ファイル キーワードは、INI ファイルの最下位の編成レベルです。これらのキーワードには、アプリケーションの終了から次回の起動までの間に保持しておく必要のあるデータを格納します。

.ini ファイルの変更で構成可能な各設定の詳細については、次を参照してください:

- .ini ファイルの設定
- .ini ファイル キーワードの設定

.ini ファイルの設定

インストールに .ini ファイルの変更を含める場合、ターゲットおよび機能の設定を行なう必要があります。これらを設定するには、[INI ファイル変更] ビューを開いて INI ファイル エクスプローラーで .ini ファイルを選択します。

テーブル 3-30 ・ .ini ファイルの設定

設定	説明
機能	この .ini ファイルの変更を関連付ける機能を選択します。 選択された機能がインストールされると、.ini ファイルが変更されます。選択された機能がインストールされていない場合、変更されません。

テーブル 3-30・.ini ファイルの設定 (続き)

設定	説明
ターゲット	ターゲット システムにある .ini ファイルの保存場所を指定してください。パスをハードコード化するのではなく、ドロップダウン リストから Windows Installer フォルダーのプロパティを選択します。 ***ヒント・*** *サブフォルダーとフォルダーのプロパティを円記号で区切らず、下位レベルのサブフォルダーを円記号で区切ります (例、*[ProgramFilesFolder]MyCompany¥Subdirectory*)。*

.ini ファイル キーワードの設定

キーワードは、.ini ファイルの最下位の編成レベルです。キーワードには、アプリケーションの終了から次に起動するまでの間に保持しておく必要のあるデータを格納します。キーワードの設定について、以下に説明します。

テーブル 3-31・.ini ファイル キーワードの設定

設定	説明
アクション	実行するアクションを選択します。有効なオプションは次のとおりです: ・ **古い値を置き換える** – 既存の値は、[データ値] 設定に入力された値で置換されます。値が存在しない場合は、インストーラーによって作成されます。 ・ **上書きしない** – キーワードが既に存在しない場合は、.ini ファイルにその値が追加されます。.ini ファイルにキーワードが存在する場合、キーワードは変更されません。 ・ **タグを追加する**–インストールは既存の値に新しい値を追加します。新しい値と既存の値はカンマで区切ります。タグの追加先となるキーワードが存在しない場合、変更されません。 ・ **値全体を削除する** – キーワードおよびその値は両方とも .ini ファイルから削除されます。指定したキーワードが存在しない場合、キーワードは変更されません。このオプションを選択する場合は、"データ値" 設定を空白に残します。 ・ **タグを削除する** – キーワードの複数値はタグとして認識されます。タグはコンマで区切られています。キーワードの値からタグを外すには、このオプションを選択します。"データ値" 設定で、削除するタグを入力してください。
データ値	キーワードの値を入力します。値を追加または付加する場合、新しい値を入力します。タグを削除する場合、削除するタグを入力します。キーワードと値を削除する場合、この設定を空白のまま残します。 キーワードの値として Windows Installer のプロパティを使用できます。Windows Installer のプロパティを使用するには、プロパティを角かっこで囲みます (例、[INSTALLDIR])。

[ファイルの拡張子] ビュー

ファイルの拡張子を使用すると、特定のファイルの種類を製品にリンクさせることができます。このファイルをダブルクリックすると、製品が起動して、そのファイルを開きます。テキスト (.txt) ファイルを開くと、そのテキストファイルの内容を表示するためメモ帳を起動するように、オペレーティング システムに指示するメッセージを送信していることになります。製品やそのファイルに同じような機能を提供する場合、ファイル拡張子関連付けを作成します。

ファイルの関連付けの構成可能な各設定についての詳細は、「ファイル拡張子設定」を参照してください:

ファイル拡張子設定

ファイル拡張子設定では、関連するファイル タイプの詳細情報を指定できます。これらの設定にアクセスするには、[ファイル拡張子] ビューを開いて、[ファイル拡張子] エクスプローラーから編集するファイル拡張子を選択します。

各ファイル拡張子設定について、以下に説明します。

テーブル 3-32 • ファイル拡張子設定

設定	説明
機能	このファイル タイプと関連付ける機能を選択します。選択した機能がインストールされている場合、ファイル タイプが関連付けられます。
ファイル	ファイル拡張子を関連付ける実行可能ファイルをリストから選択します。インストール プロジェクトに必要なファイルをまだ追加していない場合、ドロップダウン リストには表示されません。 ***重要:*** *ダイナミック リンク ファイルにファイル拡張子の関連付けを行うことはできません。関連付けを構成する場合、スタティック ファイルとしてプロジェクトに含めなくてはなりません。*
コマンド	Open 動詞のコンテキスト メニューに表示するテキストを入力します。この設定を空白のままにすると、この種類のファイルのコンテキスト メニューに最初のコマンドとして [開く(O)] が表示されます。MyProduct の編集(E) と表示する場合、この設定に「**MyProduct の編集(&E)**」と入力します。
引数	このタイプのファイルを開いたとき、製品に渡すコマンドライン引数を入力します。引数として **%1** を使用すると、現在選択されているファイル名を引数の一部として渡すことができます。たとえば、**-p %1** は、**-p C:¥MyFile.ext** と解決されます。
ProgID	ファイル拡張子と関連付ける製品の progID を入力します。
ProgID の説明	この説明は、ターゲット システムのレジストリで progID のデフォルト値として登録されます。

テーブル 3-32・ファイル拡張子設定 (続き)

設定	説明
アイコン	この種類のファイルに表示するアイコンを指定するには、以下の設定を使用します: ・ Icon—.ico ファイル、またはアイコン リソースを含む .ico または .exe ファイルを参照するには、この設定で省略記号ボタン (...) をクリックします。 ファイルを参照してから選択するか、"アイコン ファイル" 設定でパスとファイル名を手入力した場合、InstallShield はこの設定にアイコン パスとインデックスをリストします。 ・ アイコン ファイル — アイコンを含むファイルのパスとファイル名を手動で指定する場合は、それを入力します。 ・ アイコン インデックス — 指定したアイコン ファイルに 1 つ以上のアイコン リソースがある場合、この設定にインデックスを入力します。 負の数以外の整数を指定すると、実行可能ファイルのアイコン リソースの順番が参照されます。たとえば、0 はファイル内の最初のアイコン、1 は 2 番目のアイコン、2 は 3 番目のアイコンを参照します。 負の数字は特定のリソース ID を参照するために使用します。たとえばアイコン インデックス -12 は、リソース ID が 12 のアイコンを示します。

[環境変数] ビュー

環境変数は、インストール プログラムと共にターゲット システム上に設定できる名前と値の組み合わせで、アプリケーションおよび実行されているその他のプログラムによってアクセスできます。

InstallShield を使うと、インストール プログラムを使ってターゲット システム上に環境変数を作成、設定 (または変更) したり、システム上の環境変数を削除することができます。環境変数を指定するには、[ビュー リスト] にある [環境変数] ビューを開きます。

環境変数の設定

環境変数の設定を行なうと、ターゲット システム上にある既存の変数の変更方法を指定したり、新しい変数を作成したりできます。各環境変数設定について、次に説明します。

テーブル 3-33・環境変数の設定

設定	説明
機能	この環境変数と関連付ける機能を選択します。 環境変数と関連付けられた機能がインストールされると、ターゲット システム上で環境変数の作成、変更、または削除が行われます。 ***メモ:*** *この環境変数がアプリケーションのインストール時に確実に作成、変更または削除されるようにするには、[常にインストール] を選択します。環境変数を特定の機能のみと関連付ける場合は、その機能を選択します。*
値	この環境変数のパスまたは値を入力します。[INSTALLDIR]bin—などのように、定義済みフォルダーをこの値の一部に使用できます。 複数のパスを入力する場合は、パスをセミコロン (;) で区切ります。 ***メモ:*** *"インストール時" 設定に [削除する] を選択すると、"値" 設定に入力された値はすべてクリアされて、"値" 設定が読み取り専用となります。*
インストール時	関連付けられた機能がターゲット システムにインストールされたときに発生する動作を指定します。選択可能なオプションは以下のとおりです: ・ **作成する** — 指定された環境変数がターゲット システムに既存しない場合、インストールはその環境変数を作成して値を設定します。 ・ **設定する** — "配置" 設定と組み合わせて使用して、既存の環境変数の値を設定します。このオプションを選択した場合で、ターゲット システムに環境変数が存在しないときは、インストールは環境変数を作成して、その値を設定します。ターゲット システムに環境変数が存在する場合、インストールはその値を設定します。 ・ **削除する** — インストールは、指定された環境変数をターゲット システムから削除します。 ***メモ:*** *"インストール時" 設定に [削除する] を選択すると、"値" 設定に入力された値はすべてクリアされて、"値" 設定が読み取り専用となります。*

テーブル 3-33・環境変数の設定（続き）

設定	説明
Placement	指定された環境変数の既存値に相対して、"値" フィールドに配置する値を指定します。選択可能なオプションは以下のとおりです: ・ **後に追加する** – インストールは、"値" 設定に入力された新しい値を既存する値の終わりに追加します。 ・ **前に追加する** – インストールは、"値" 設定に入力された新しい値を既存する値の始まりに追加します。 ・ **置換する** – インストールは、指定された環境変数の値を "値" 設定に入力された新しい値で置換します。 ***メモ:*** *"インストール時" 設定で [作成] を選択し、かつ指定された環境変数がターゲット システムに存在した場合、"配置" 設定は新しい値が既存の環境変数値に追加された状態か、または既存の環境変数値を置き換えた状態かを指定します。ただし、この場合において、指定された環境変数がターゲット システムにない場合、その値は作成され、"配置" 設定は無視されます。*
アンインストール時	環境変数が関連付けられた機能がアンインストールされたとき、その環境変数を更新するかどうかを示します。選択可能なオプションは以下のとおりです: ・ **削除する**–"配置" 設定に [前に追加] または [後に追加] が選択されていて、アンインストール時にターゲット システム上の環境変数値に指定された値が含まれる場合、その値のみが既存する変数の値から削除されます。 "配置" 設定で [置換する] を選択した場合で、アンインストール時に条件（ターゲット システム上の値が指定された値と一致する、またはターゲット システム上の値が空白である）のどちらかが True 評価されたとき、環境変数全体が削除されます。 その他すべての状況下では、環境変数とその値がターゲット システムにそのまま残ります。 ・ **削除しない**–環境変数と（存在する場合）その値がターゲット システムにそのまま残ります。
種類	環境変数名がシステム変数または環境変数のどちらの種類であるかを示します。選択可能なオプションは以下のとおりです: ・ **システム** – インストールは、指定されたシステム環境変数を作成、変更または削除します。 ・ **ユーザー** – インストールは、エンド ユーザー環境から環境変数を作成、変更または削除します。指定された環境変数は、インストール時にログオンしているエンド ユーザー以外は使用できません。

≫ [IIS 構成] ビュー

[IIS 構成] ビューでは、新しい IIS Web サイト、アプリケーション、および仮想ディレクトリを作成および管理することができます。

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

[IIS 構成] ビューの設定は、次のメイン領域に分かれています:

- Web サイトの設定
- "アプリケーション" と "仮想ディレクトリ" の設定

次のテーブルは、[IIS 構成] ビューの設定の上に表示されるボタンについて説明します。これらのボタンは、中央ペインで Web サイト、アプリケーション、仮想ディレクトリが選択されている場合に表示されます。

テーブル 3-34 • [IIS 構成] ビューのコントロール

コントロールの名前	アイコン	説明
カテゴリー別		カテゴリごとに設定を並べ替えます。
アルファベット順		設定をアルファベット順に並べ替えます。

Web サイトの設定

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含ま

れています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

[IIS 構成] ビューの Web サイト項目を利用して Web サイトの追加および削除を行い、またシステム全体を通して Web サーバーを構成することができます。

Web サーバー の設定

[IIS 構成] ビューで [Web サイト] エクスプローラーを選択すると、次の Web サーバーの設定が表示されます。

テーブル 3-35 ・ Web サーバー の設定

設定	説明
IIS を構成した後、Web サーバーを再起動する (IIS 6 以前のみ)	インストールで IIS の変更がシステムに加えられたとき、毎回インストールの完了時に IIS を再起動する場合、[はい] を選択します。アプリケーションによっては、IIS の再起動が必要です。 この設定は IIS 6 以前に適用します。IIS 7 は、この設定を無視します。

テーブル 3-35 • Web サーバー の設定 (続き)

設定	説明
SSIEnableCmdDirective レジストリ値	HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥W3SVC¥Parameters レジストリ キーの SSIEnableCmdDirective レジストリ値をターゲット システムでどのように設定するかを指定します。SSIEnableCmdDirective レジストリ値は、サーバー側インクルード (SSI) の #exec CMD ディレクティブがシェル コマンドの実行に使用されることを Web サーバーが許可するかどうかを制御します。有効なオプションは次のとおりです: • 無視する – ターゲット システム上の SSIEnableCmdDirective レジストリ値を変更しません。デフォルトでは、これが設定されています。 • FALSE (0)–ターゲット システム上の SSIEnableCmdDirective レジストリ値を 0 に設定します。これにより、サーバー側インクルードの #exec CMD ディレクティブがシェル コマンドの実行に使用されるを防ぐことができます。この値を選択すると、IIS Web サーバーに #exec CMD ディレクティブに依存するアプリケーションが存在した場合、インストール プロジェクトの Web サイトおよび仮想ディレクトリがインストールされたあと、これらのアプリケーションが誤作動を起こす可能性があります。 • TRUE (1)–ターゲット システム上の SSIEnableCmdDirective レジストリ値を 1 に設定します。これにより、サーバー側インクルードの #exec CMD ディレクティブがシェル コマンドの実行で使用できるようになります。 FALSE または TRUE オプションを選択すると、値 (FALSE の場合 0、TRUE の場合 1) がINSTALLSHIELD_SSI_PROP プロパティに格納されます。 セキュリティに関する懸念により、デフォルトの SSIEnableCmdDirective 値は FALSE (0) になっています。FALSE (0) 値により、エンドユーザーによって承認されていないサーバー側での実行可能ファイルの実行を防ぐことができます。 ***メモ:*** *製品のインストール中に SSIEnableCmdDirective レジストリ値が変更された場合でも、ターゲット システムから製品がアンインストールされるときに、SSIEnableCmdDirective レジストリ値が変更されることはありません。* 詳細については、「Web サーバーで CMD コマンドが SSI #exec ディレクティブに使用されるのを許可するかどうかを指定する」を参照してください。

メモ: *上記の Web サーバーの設定は、IIS アイテム (Web サイト、アプリケーション、または仮想ディレクトリ) が何もインストールされなかった場合、ターゲット システムで更新されません。*

[Web サイト] の設定

エクスプローラーで Web サイトを選択すると、多くの設定が表示されます。Web サイトの設定は、いくつかのメイン カテゴリで構成されています:

- 識別
- 全般

- ホーム ディレクトリ
- アプリケーションの設定
- セキュリティ
- 詳細

識別の設定

[IIS 構成] ビューの Web サイトにある [識別] 領域には、次の設定があります:

テーブル 3-36 • Web サイトの識別の設定

設定	説明
名前	Web サイトの名前を入力します。
IP アドレス	特定の IP アドレスをターゲットする場合に入力します。 代りに、デフォルト値のアステリスク (*) のままに残すこともできます。この設定の値をアステリスクまたは空白にすると、現在使用中でない任意の IP アドレスが使用されます。
TCP ポート	IIS Web サイト用の TCP ポート設定は、ターゲットマシン上でサービスが実行されているポート番号を示します。IIS Web サーバーには複数の Web サイトをホストできるバージョンもあります。それぞれの Web サイトは固有のポート番号に関連付けられています。 ターゲット システムのポート番号の指定が難しい場合、この設定で 0 を入力します。 Web サイトがインストールされたとき、Web サイトが使用するポート番号とサイト番号についての詳細は、「TCP ポート番号とサイト番号の構成」を参照してください。
ホスト ヘッダー名	インストール中にインストールされる IIS Web サイトを識別するホスト ヘッダー名を指定するには、このボックスにその名前を入力します。例: **www.mycompany.com** ホスト ヘッダー (ドメイン名とも呼ばれます) を利用して、複数の Web サイトを Web サーバー上の IP アドレスに割り当てることができます。
サイト番号	"サイト番号" 設定は Web サイトが作成される場所のパスに含まれる番号を示します (例、w3svc/3)。デフォルト値は 0 です。 ターゲット システムのサイト番号の指定が難しい場合、この設定で 0 を入力します。 Web サイトがインストールされたとき、Web サイトが使用するポート番号とサイト番号についての詳細は、「TCP ポート番号とサイト番号の構成」を参照してください。

[全般] の設定

[IIS 構成] ビューの Web サイトにある [全般] 領域には、次の設定があります:

テーブル 3-37 • Web サイトの全般の設定

設定	説明
機能	Web サイトが関連付けられている機能を選択します。 選択された機能がインストールされるとき、Web サイトとそのアプリケーションおよび仮想ディレクトリのすべてがインストールされます。選択された機能がインストールされない場合、Web サイトとそのアプリケーションおよび仮想ディレクトリはインストールされません。
ASP.NET バージョン	Web サイトの ASP.NET バージョンを設定するには、完全なバージョン番号を入力するか、一覧からそれを選択します。 たとえば、バージョン 2 の ASP.NET を指定する場合、**2.0.50727** と入力します。バージョン 1.1 の ASP.NET を指定するには、**1.1.4322** と入力します。 Web サイトの ASP.NET バージョンを指定すると、IIS はそのバージョン番号をすべての Web サイトのアプリケーションに使用します。 ***重要***• *インストールが Windows Vista 以降のシステムで実行される可能性がある場合、ASP.NET バージョンを指定しないほうがよい場合があります。また、バージョン 3 の ASP.NET を指定すると、実行時にエラーが発生しますので注意してください。詳細については、「Web サイトまたはアプリケーションの ASP.NET バージョンを設定する」を参照してください。*
アンインストール時に削除する	選択した Web サイトをアンインストール中に削除するかどうかを指定します。 詳細については、「Web サイト、アプリケーション、および仮想ディレクトリのアンインストール」を参照してください。
デフォルト ドキュメント	Web サイトのデフォルト ページの名前を入力します。複数のページを指定するには、名前をコンマで区切ります。 Web サイトは、ブラウザーのリクエストでドキュメント名が指定されない場合に、デフォルト ページとして使用されます。

[ホーム ディレクトリ] の設定

[IIS 構成] ビューの Web サイトにある [ホーム ディレクトリ] 領域には、次の設定があります:

テーブル 3-38 • Web サイトのホーム ディレクトリの設定

設定	説明
コンテンツ ソース パス (ローカルまたは UNC)	この設定は、Web サイトのファイルを格納するローカル パスまたはネットワーク ディレクトリのパスを識別します。 • Web サイトのコンテンツがターゲット システムに存在する場合、この設定の省略記号ボタン (...) をクリックして、ローカル パスを指定します。[ディレクトリの参照] ダイアログ ボックスが開きます。このダイアログ ボックスでは、Windows Installer プロパティ (たとえば、[IISROOTFOLDER]) を選択するか、または新しいプロパティを作成できます。 デフォルトでは、これらのファイルは IISROOTFOLDER に格納されています。 • Web サイトのコンテンツがターゲット システムに存在する場合、この設定の [UNC] ボタンをクリックして、ネットワークの場所を指定します。参考例: ¥¥server¥share ***ヒント:*** *各 Web サイトには一意の物理パスが必要です。一意のパスは、Web サイトが Windows Vista 以降または Windows Server 2008 以降のシステムにインストールされる場合、特に重要です。詳細については、「IIS サポートの実行時要件」を参照してください。*
スクリプトソースへのアクセス	読み取りまたは書き込み許可のどちらかが設定されている場合にエンドユーザーがソース コードへアクセスできるようにするかどうかを指定します。ソース コードには ASP アプリケーションのスクリプトが含まれます。
読み取りアクセス	エンドユーザーが Web サイトに読み取りアクセスできるようにするかどうかを指定します。
書き込みアクセス	エンドユーザーが Web サイトに書き込みアクセスできるようにするかどうかを指定します。これにより、エンド ユーザーがターゲット マシン上の Web サイトのプロパティを変更できるようになります。
ディレクトリの参照	エンド ユーザーが、この Web サイトの下にあるすべての仮想ディレクトリとサブディレクトリを参照できるかどうかを指定します。
アクセスのログ記録	ログ ファイルに、この Web サイトへのアクセスを記録するかどうかを指定します。この Web サイトのログ記録が有効な場合のみ、アクセスが記録されます。
このリソースにインデックスを付ける	Microsoft インデックス サービスが Web サイトの全文インデックス上に このディレクトリを含むようにするかどうかを指定します。 この設定は IIS 6 以前に適用します。IIS 7 は、この設定を無視します。

アプリケーションの設定

[IIS 構成] ビューの Web サイトにある [アプリケーション] 領域には、次の設定があります:

テーブル 3-39 • Web サイトのアプリケーションの設定

設定	説明
アプリケーション マッピング	ディレクトリのアプリケーションのマッピングをカスタマイズするには、この設定で省略記号ボタン (...) をクリックします。これによって [アプリケーションのマッピング] ダイアログ ボックスが開きます。ここではファイル名拡張子とそれらのファイルを処理するアプリケーションとの間のマッピングを編集または削除することができます。
MIMEタイプ	選択した Web サイトに MIME の種類を追加、変更、または削除するには、この設定で省略記号ボタン (...) をクリックします。[MIME の種類] ダイアログ ボックス が開きます。ここで、ファイル名拡張子と、ターゲット システム上の Web サーバーからブラウザまたはメール クライアントにスタティック ファイルとして提供される対応コンテンツの種類とのマッピングを追加、編集、または削除します。
セッション タイムアウト(分)	サーバーが自動的に終了する前にセッションをアイドル状態に保つ時間を分単位で指定します。エンド ユーザーがこのタイムアウト期間中にページを更新したりページを要求しないとセッションが終了します。デフォルト値は 20 分です。
ASP スクリプト タイムアウト(秒)	.asp ページでスクリプトが実行でき、Windows イベントログ記録に書き込むことができる時間を秒数で指定します。このプロパティの最小値は 1 秒で、デフォルト値は 90 秒です。

セキュリティの設定

[IIS 構成] ビューで Web サイトを選択すると、[セキュリティ] 領域にいくつかのセキュリティ関連の設定が表示されます。[セキュリティ] 領域では、ユーザー ID を検証できるように Web サーバーを構成することができます。ユーザーを認証することで、認証されていないユーザーによる制限された内容への Web (HTTP) 接続を不可能にします。詳細については、IIS マニュアルを参照してください。

[セキュリティ] 領域の設定は、いくつかのカテゴリに分かれています:

- 匿名接続
- 認証済みアクセス
- セキュリティで保護された通信

[匿名接続] 領域の設定は次のとおりです:

テーブル 3-40 ・ [セキュリティの設定] 領域にある [匿名接続] の設定

設定	説明
匿名アクセスを有効にする	ユーザーが匿名アクセスできるようにするかどうかを指定します。匿名接続を許可する場合、適切な Windows ユーザー アカウント情報も入力します。 エンド ユーザーがコンテンツにアクセスする前に Web サーバーによる ID の確認を必要としない場合、この設定に [いいえ] を選択します。
IIS による匿名パスワードの制御	匿名パスワード設定がターゲット システム上の Windows での設定と自動的に統一されるようにするかどうかを指定します。匿名アカウント用に入力したパスワードが Windows のパスワードと異なる場合、匿名認証は正しく動作しません。 ***メモ・*** *パスワードの同期化はリモート コンピューター上の匿名アカウントではなく、ローカル コンピューター上で定義された匿名ユーザー アカウントと共に使用しなくてはなりません。*
匿名ユーザー名	匿名接続を有効にする場合、匿名アカウントの名前を入力します。
匿名パスワード	[IIS による匿名パスワードの制御] 設定で [いいえ] を選択した場合は、匿名ユーザー アカウントのパスワードを入力します。パスワードは Windows 内でのみ利用されます。匿名ユーザーはユーザー名とパスワードを使ってログオンしません。

[認証済みアクセス] 領域の設定は次のとおりです:

テーブル 3-41 ・ [セキュリティの設定] 領域にある [認証済みアクセス] の設定

設定	説明
基本認証	基本の認証メソッドを使って、Web サイトにアクセスするエンド ユーザーのユーザー名とパスワード情報を収集するかどうかを指定します。 ***重要・*** *基本の認証方法では、ネットワークを介してユーザー名とパスワードが送信されるときに情報が暗号化されません。ネットワーク監視ツールを利用する無法なエンド ユーザーによって、ユーザー名およびパスワードが傍受される可能性があります。*

テーブル 3-41 • [セキュリティの設定] 領域にある [認証済みアクセス] の設定 (続き)

設定	説明
統合 Windows 認証	統合 Windows 認証を有効化するかどうかを指定します。統合 Windows 認証は、ユーザー ID を確認するためにユーザーのブラウザーと暗号化されたやり取りを行います。 統合 Windows 認証が有効になっている場合、Web サーバーは次の条件下でのみそれを利用します。 • 匿名アクセスは無効です。 • Windows ファイルシステム アクセス許可が設定されいているため、匿名アクセスは拒否されます。エンド ユーザーは制限されたコンテンツへ接続を確立する前に Windows ユーザー名とパスワードを提示するよう求められます。
フォーム認証	このオプションを **[はい]** に設定してフォーム認証を有効化します。ASP.NET フォーム ベースの認証は、多くの要求を受け取るパブリック Web サーバー上にあるサイトまたはアプリケーションに適しています。この認証モードでは、オペレーティング システムが提供している認証方法に頼ることなく、アプリケーション レベルでのクライアント登録および認証を管理することができます。 ***重要:*** *フォーム認証は、Web サーバーにユーザー名とパスワードをテキスト形式で送信します。ホームページ以外のアプリケーション内のすべてのページ、およびログオン ページには、Secure Sockets Layer (SSL) 暗号化を使用してください。*

[セキュリティで保護された通信] 領域の設定は、次の通りです。

テーブル 3-42 • [セキュリティの設定] 領域にある [セキュリティで保護された通信] の設定

設定	説明
SSL 証明書	ターゲット システムにインストールする必要があるサーバー証明書を指定するには、この設定で省略記号ボタン (...) をクリックしてから、適切なセキュリティ証明書ファイル (.cer または .pfx) を選択します。プロジェクトに 1 つまたは複数の証明書が既に含まれている場合は、リストから証明書を選択できます。 .cer ファイルが Binary テーブルに格納されます。 証明書のインストールが構成されていない場合、この設定は空白です。
SSL 証明書のパスワード	指定した証明書にパスワードがある場合、それをこの設定に入力します。

詳細設定

[IIS 構成] ビューの Web サイトにある [詳細] 領域には、次の設定があります。

テーブル 3-43 • Web サイトの詳細の設定

設定	説明
カスタム エラー	Web サーバー エラーが発生したときにクライアントに送信される HTTP エラーをカスタマイズするには、この設定で省略記号ボタン (...) を選択します。[カスタム エラー] ダイアログ ボックスが開き、ここで 1 つまたは複数の HTTP エラーで表示するページを指定できます。 管理者は汎用 HTTP 1.1 エラー、IIS が提供する詳細なカスタム エラー ページ、またはインストールに含めた独自のカスタム エラーを使用できます。

"アプリケーション" と "仮想ディレクトリ" の設定

エディション: *InstallShield Express Edition では、1 インストールにつき 1 つの Web サイトをインストールできます。*

InstallShield の以下のエディションでは、1 インストールにつき複数の Web サイトを作成することができます。これらのエディションでは、IIS アプリケーション プールや Web サービス拡張を管理することもできます:

- *InstallShield Premier*
- *InstallShield*

さらに InstallShield Premier Edition には、既存の IIS Web サイトをチェックして Web サイト、その仮想ディレクトリ、アプリケーション、およびアプリケーション プールの設定に関するデータを記録する IIS スキャナーも含まれています。その IIS データを InstallShield Premier Edition の [IIS の構成] ビューにインポートできます。IIS データをプロジェクトにインポートしてから、必要に応じて、[IIS 構成] ビューを使って IIS の設定を変更できます。

[IIS] ビューで、Web サイトにアプリケーションまたは仮想ディレクトリを追加することができます。このビューでアプリケーションまたは仮想ディレクトリを選択すると、多くの設定が表示されます。この設定は、いくつかのメイン カテゴリで構成されています:

- 全般
- 仮想ディレクトリ
- アプリケーションの設定
- セキュリティ
- 詳細

[全般] の設定

[IIS 構成] ビューの アプリケーションまたは仮想ディレクトリにある [全般] 領域には、次の設定があります。

テーブル 3-44 • アプリケーションまたは仮想ディレクトリにおける全般の設定

設定	説明
名前	アプリケーションまたは仮想ディレクトリの名前を入力します。
ASP.NET バージョン	***メモ:*** *この設定はアプリケーションで使用できますが、仮想ディレクトリには無効です。* アプリケーションの ASP.NET バージョンを設定するには、完全なバージョン番号を入力するか、一覧からそれを選択します。 たとえば、バージョン 2 の ASP.NET を指定する場合、**2.0.50727** と入力します。バージョン 1.1 の ASP.NET を指定するには、**1.1.4322** と入力します。 Web サイトの ASP.NET バージョンを指定すると、IIS はそのバージョン番号をすべての Web サイトのアプリケーションに使用します。 ***重要:*** *インストールが Windows Vista 以降のシステムで実行される可能性がある場合、ASP.NET バージョンを指定しないほうがよい場合があります。また、バージョン 3 の ASP.NET を指定すると、実行時にエラーが発生しますので注意してください。詳細については、「Web サイトまたはアプリケーションの ASP.NET バージョンを設定する」を参照してください。*
デフォルト ドキュメント	アプリケーションまたは仮想ディレクトリのデフォルト ページの名前を入力します。複数のページを指定するには、名前をコンマで区切ります。 アプリケーションまたは仮想ディレクトリは、ブラウザーのリクエストでドキュメント名が指定されない場合に、デフォルト ページとして使用されます。

[仮想ディレクトリ] の設定

[IIS 構成] ビューの アプリケーションまたは仮想ディレクトリにある [仮称ディレクトリ] 領域には、次の設定があります。

テーブル 3-45 • アプリケーションまたは仮想ディレクトリにおける仮想ディレクトリの設定

設定	説明
コンテンツ ソース パス (ローカルまたは UNC)	この設定は、アプリケーションまたは仮想ディレクトリのデフォルト ファイルを格納するローカル パスまたはネットワーク ディレクトリ パスを識別します。 • アプリケーション、または Web ディレクトリのコンテンツがターゲット システムに存在する場合、この設定の省略記号ボタン (...) をクリックして、ローカル パスを指定します。[ディレクトリの参照] ダイアログ ボックスが開きます。 デフォルトでは、これらのファイルは IISROOTFOLDER に格納されています。 • アプリケーションまたは仮想ディレクトリのコンテンツがターゲット システムに存在する場合、この設定の [UNC] ボタンをクリックして、ネットワークの場所を指定します。参考例: ¥¥server¥share ***ヒント:*** *各アプリケーションまたは仮想ディレクトリには一意の物理パスが必要です。一意のパスは、仮想ディレクトリが Windows Vista 以降または Windows Server 2008 以降のシステムにインストールされる場合、特に重要です。詳細については、「IIS サポートの実行時要件」を参照してください。*
スクリプトソースへのアクセス	読み取りまたは書き込み許可のどちらかが設定されている場合にエンドユーザーがソース コードへアクセスできるようにするかどうかを指定します。ソース コードには ASP アプリケーションのスクリプトが含まれます。
読み取りアクセス	エンドユーザーがアプリケーションまたは Web サイトに読み取りアクセスできるようにするかどうかを指定します。
書き込みアクセス	エンドユーザーがアプリケーションまたは Web サイトに書き込みアクセスできるようにするかどうかを指定します。これにより、エンド ユーザーがターゲットマシン上のアプリケーションまたは仮想ディレクトリのプロパティを変更できるようになります。
ディレクトリの参照	エンド ユーザーが、このアプリケーションまたは Web サイトの下にあるすべての仮想ディレクトリとサブディレクトリを参照できるかどうかを指定します。
アクセスのログ記録	ログ ファイルに、このアプリケーションまたは Web サイトへのアクセスを記録するかどうかを指定します。アクセス数は、ログ記録が有効化されている場合のみ記録されます。

テーブル 3-45・アプリケーションまたは仮想ディレクトリにおける仮想ディレクトリの設定 (続き)

設定	説明
このリソースにインデックスを付ける	Microsoft のインデックス サービスを許可して、このアプリケーションまたは仮想ディレクトリを全文インデックスに含めるかどうかを指定します。 この設定は IIS 6 以前に適用します。IIS 7 は、この設定を無視します。

アプリケーションの設定

[IIS 構成] ビューの アプリケーションまたは仮想ディレクトリにある [アプリケーションの設定] 領域には、次の設定があります。

テーブル 3-46・アプリケーションまたは仮想ディレクトリにおけるアプリケーションの設定

設定	説明
アプリケーション名	選択した仮想ディレクトリをアプリケーションと関連付けるには、アプリケーションの名前を指定します。 ***メモ:*** *この設定は仮想ディレクトリで使用できますが、アプリケーションには無効です。* *仮想ディレクトリが InstallShield 2009 以前で作成されていて、InstallShield の現在のバージョンにアップグレードされている場合、この設定が表示されます。それ以外の場合、この設定は含まれません。*
アプリケーション マッピング	ディレクトリのアプリケーションのマッピングをカスタマイズするには、この設定で省略記号ボタン (...) をクリックします。これによって [アプリケーションのマッピング] ダイアログ ボックスが開きます。ここではファイル名拡張子とそれらのファイルを処理するアプリケーションとの間のマッピングを編集または削除することができます。
MIMEタイプ	選択したアプリケーションまたは仮想ディレクトリの MIME の種類を追加、変更、または削除するには、この設定で省略記号ボタン (...) をクリックします。[MIME の種類] ダイアログ ボックス が開きます。ここで、ファイル名拡張子と、ターゲット システム上の Web サーバーからブラウザまたはメール クライアントにスタティック ファイルとして提供される対応コンテンツの種類とのマッピングを追加、編集、または削除します。
セッション タイムアウト(分)	サーバーが自動的に終了する前にセッションをアイドル状態に保つ時間を分単位で指定します。エンド ユーザーがこのタイムアウト期間中にページを更新したりページを要求しないとセッションが終了します。デフォルト値は 20 分です。
ASP スクリプト タイムアウト(秒)	.asp ページでスクリプトが実行でき、Windows イベントログ記録に書き込むことができる時間を秒数で指定します。このプロパティの最小値は 1 秒で、デフォルト値は 90 秒です。

テーブル 3-46・アプリケーションまたは仮想ディレクトリにおけるアプリケーションの設定 (続き)

設定	説明
実行アクセス許可	選択したアプリケーションまたは仮想ディレクトリに対して許可する、プログラムの実行レベルを指定します。選択可能なオプションは以下のとおりです: ・ なし – HTML やイメージ ファイルといったスタティック ファイルのみアクセス可能です。 ・ スクリプトのみ – ASP スクリプトなどのスクリプトのみ実行可能です。 ・ スクリプトと実行可能ファイル – すべてのファイル タイプにアクセスまたはそれらを実行できます。
アプリケーション保護	*メモ・この設定はアプリケーションで使用できますが、仮想ディレクトリには無効です。* 保護レベルを指定します: ・ 高 – アプリケーションは、別の処理とは切り離された単独の処理で実行します。 ・ 中 – アプリケーションはその他のアプリケーションと共に切り離されたプール プロセスで実行します。 ・ 低 – アプリケーションは Web サービスと同じ処理で実行します。 この設定は IIS 5 以前に適用します。それ以降のバージョンはこの設定を無視します。

セキュリティの設定

[IIS 構成] ビューでアプリケーションまたは仮想ディレクトリを選択すると、[セキュリティ] 領域にいくつかのセキュリティ関連の設定が表示されます。[セキュリティ] 領域では、ユーザー ID を検証できるようにアプリケーションまたは仮想ディレクトリ を構成することができます。ユーザーを認証することで、認証されていないユーザーによる制限された内容への Web (HTTP) 接続を不可能にします。詳細については、IIS マニュアルを参照してください。

[セキュリティ] 領域の設定は、次のカテゴリに分かれています:

- 匿名接続
- 認証済みアクセス

[匿名接続] 領域の設定は次のとおりです:

テーブル 3-47 ・ [セキュリティの設定] 領域にある [匿名接続] の設定

設定	説明
匿名アクセスを有効にする	ユーザーが匿名アクセスできるようにするかどうかを指定します。匿名接続を許可する場合、適切な Windows ユーザー アカウント情報も入力します。 エンド ユーザーがコンテンツにアクセスする前に Web サーバーによる ID の確認を必要としない場合、この設定に [いいえ] を選択します。
IIS による匿名パスワードの制御	匿名パスワード設定がターゲット システム上の Windows での設定と自動的に統一されるようにするかどうかを指定します。匿名アカウント用に入力したパスワードが Windows のパスワードと異なる場合、匿名認証は正しく動作しません。 ***メモ・*** *パスワードの同期化はリモート コンピューター上の匿名アカウントではなく、ローカル コンピューター上で定義された匿名ユーザー アカウントと共に使用しなくてはなりません。*
匿名ユーザー名	匿名接続を有効にする場合、匿名アカウントの名前を入力します。
匿名パスワード	[IIS による匿名パスワードの制御] 設定で [いいえ] を選択した場合は、匿名ユーザー アカウントのパスワードを入力します。パスワードは Windows 内でのみ利用されます。匿名ユーザーはユーザー名とパスワードを使ってログオンしません。

[認証済みアクセス] 領域の設定は次のとおりです:

テーブル 3-48 ・ [セキュリティの設定] 領域にある [認証済みアクセス] の設定

設定	説明
基本認証	基本の認証メソッドを使って、アプリケーションまたは仮想ディレクトリにアクセスするエンド ユーザーのユーザー名とパスワード情報を収集するかどうかを指定します。 ***重要・*** *基本の認証方法では、ネットワークを介してユーザー名とパスワードが送信されるときに情報が暗号化されません。ネットワーク監視ツールを利用する無法なエンド ユーザーによって、ユーザー名およびパスワードが傍受される可能性があります。*

テーブル 3-48 • [セキュリティの設定] 領域にある [認証済みアクセス] の設定 (続き)

設定	説明
統合 Windows 認証	統合 Windows 認証を有効化するかどうかを指定します。統合 Windows 認証は、ユーザー ID を確認するためにユーザーのブラウザーと暗号化されたやり取りを行います。 統合 Windows 認証が有効になっている場合、Web サーバーは次の条件下でのみそれを利用します。 • 匿名アクセスは無効です。 • Windows ファイルシステム アクセス許可が設定されいているため、匿名アクセスは拒否されます。エンド ユーザーは制限されたコンテンツへ接続を確立する前に Windows ユーザー名とパスワードを提示するよう求められます。

詳細設定

[IIS 構成] ビューの アプリケーションまたは仮想ディレクトリにある [詳細] 領域には、次の設定があります。

テーブル 3-49 • アプリケーションまたは仮想ディレクトリにおける詳細の設定

設定	説明
カスタム エラー	Web サーバー エラーが発生したときにクライアントに送信される HTTP エラーをカスタマイズするには、この設定で省略記号ボタン (...) を選択します。[カスタム エラー] ダイアログ ボックスが開き、ここで 1 つまたは複数の HTTP エラーで表示するページを指定できます。 管理者は汎用 HTTP 1.1 エラー、IIS が提供する詳細なカスタム エラー ページ、またはインストールに含めた独自のカスタム エラーを使用できます。

[コンポーネント サービス] ビュー

[コンポーネント サービス] ビューでは、インストール パッケージ用の COM+ サーバー アプリケーションとコンポーネントを管理できます。

InstallShield のコンポーネント サービスに関する次の情報をお読みください。

- COM+ システム アプリケーション以外のみプロジェクトに追加することができます。したがって、InstallShield は [コンポーネント サービス] ビュー の [COM+ アプリケーション] エクスプローラーの下に、COM+ システム アプリケーション以外のみを表示します。
- ローカル マシンにインストールされている COM+ アプリケーションのみが、[コンポーネント サービス] ビューに表示され、プロジェクトに追加することが可能です。

[コンポーネント サービス] ビューで COM+ アプリケーションを選択するとき、以下のタブが表示されます:

- インストール
- 全般
- セキュリティ

- ID
- アクティベーション
- キュー
- 詳細
- ダンプ
- プール/再利用

[インストール] タブの設定は、InstallShield 固有の設定です。その他のタブにある設定は、コントロール パネルにある [コンポーネント サービス] 管理ツールにある設定に類似しています。[インストール] タブにある各設定についての詳細は、「[インストール] タブ」を参照してください。その他のタブにある設定については、「コンポーネント サービス」ヘルプを参照してください。

エディション: *InstallShield Premier Edition と InstallShield Professional Edition では、COM+ サーバー アプリケーションとアプリケーション プロキシ両方を管理することができる追加機能が [コンポーネント サービス] ビューで提供されています。COM+ アプリケーション プロキシはサーバー アプリケーション属性のサブセットで構成され、これはクライアント コンピューターからアプリケーションが存在するマシンへのリモート アクセスを可能にします。*

[インストール] タブ

[インストール] タブは、[コンポーネント サービス] ビューで COM+ アプリケーションを選択したときに表示されるタブの 1 つです。

テーブル 3-50 · [インストール] タブの設定

設定	説明
機能	選択した COM+ アプリケーションを含める機能を選択します。COM+ アプリケーションを新しい機能に追加するには、まず [機能] ビューで機能を作成してから、この一覧でそのチェック ボックスを選択します。
ビルド時にクライアントマシンから COM+ 設定をリフレッシュする	[コンポーネント サービス] ビューでは、ローカル マシンのコンポーネント サービスで使用できる COM+ の設定が表示されます。 プロジェクトで表示されている COM+ の設定をローカル マシンのコンポーネント サービスで提供されている設定で更新する場合、このチェック ボックスを選択します。リリースをビルドしたとき、そのつど設定がリフレッシュされます。
ロールと共にユーザー ID をインストールする	選択した COM+ アプリケーションをローカル マシンの COM+ アプリケーションについて構成したユーザー ID とロールと共にインストールする場合、このチェック ボックスを選択します。

テーブル 3-50 • [インストール] タブの設定 (続き)

設定	説明
InstallFinalize アクションの後でインストールする	選択した COM+ アプリケーションに、グローバル アセンブリ キャッシュ (GAC) にインストールする必要がある .NET アセンブリが含まれている場合、このチェック ボックスを選択します。このチェック ボックスを選択すると、ISComponentServiceFinalize アクションは選択した COM+ アプリケーションを InstallFinalize アクションの後でインストールします。Windows Installer は、InstallFinalize が実行されるまでスクリプト内のセッションで加えられた変更をコミットしません。
Destination	COM+ アプリケーションのデフォルトのインストール先は、 [ProgramFilesFolder]COMPlus Applications¥{UID} です。 COM+ ファイルを別の場所にインストールする場合、ターゲット先を選択します。指定するインストール先が一覧にない場合、**[ディレクトリ エントリを参照、作成、または変更しま]** オプションを選択します。

エディション: *InstallShield Premier Edition と InstallShield Professional Edition では、COM+ サーバー アプリケーションとアプリケーション プロキシ両方を管理することができる追加機能が [コンポーネント サービス] ビューで提供されています。COM+ アプリケーション プロキシはサーバー アプリケーション属性のサブセットで構成され、これはクライアント コンピューターからアプリケーションが存在するマシンへのリモート アクセスを可能にします。*

[サービス] ビュー

エディション: *InstallShield Express Edition は、サービスのインストール中にサービスをインストール、およびアンインストール中にサービスを削除するサポートを含みます。また、オプションでインストール後にサービスを開始、システム起動時に毎回自動的にサービスを開始、または (サービス コントロール マネージャを通してサービスが要求されたときに) オンデマンドでサービスを開始するサポートが含まれています。*

InstallShield Premier および InstallShield は、サービスにさらなる柔軟性を提供します。これらのエディションでは、インストールまたはアンインストール中にサービスを開始、停止、または削除することができます。これらのエディションではまた、Windows Installer 5 で利用可能な拡張サービス カスタマイズ オプションを構成することもできます。さらに、InstallShield Premier および InstallShield では、ターゲット システムに既存するサービスを構成することもできます。

[サービス] ビューを使って、インストール中にインストール、およびアンインストール中に削除するサービスの情報を指定します。サービスを追加するには、まず [ファイル] ビューを使ってプロジェクトにサービス実行可能ファイルを追加します。次に、[サービス] ビューで [サービス] ノードを右クリックしてから [サービスの追加] を選択します。次に、構成を行うサービスのサービス名を指定します。ここで入力する名前は、サービスの [プロパティ] ダイアログ ボックスに表示される名前と一致しなくてはなりません。

メモ: *Windows Installer はドライバー サービスをサポートしていないので、サービスは単一実行可能ファイル (.exe) である必要があります。*

サービスの設定を構成するにあたっては、サービスの技術的な詳細についてよく知っておく必要があります。

サービスに構成可能な各設定についての詳細は、「[サービス] ビューの設定」を参照してください。

[サービス] ビューの設定

エディション: *InstallShield Express Edition は、サービスのインストール中にサービスをインストール、およびアンインストール中にサービスを削除するサポートを含みます。また、オプションでインストール後にサービスを開始、システム起動時に毎回自動的にサービスを開始、または (サービス コントロール マネージャを通してサービスが要求されたときに) オンデマンドでサービスを開始するサポートが含まれています。*

InstallShield Premier および InstallShield は、サービスにさらなる柔軟性を提供します。これらのエディションでは、インストールまたはアンインストール中にサービスを開始、停止、または削除することができます。これらのエディションではまた、Windows Installer 5 で利用可能な拡張サービス カスタマイズ オプションを構成することもできます。さらに、InstallShield Premier および InstallShield では、ターゲット システムに既存するサービスを構成することもできます。

ヒント: *[サービス] ビューの上部にある [ビュー フィルター] を使って、ビューに表示するサービス データを含む機能を選択します。ビューフィルターは、プロジェクトの機能およびサブ機能の階層をリスト表示します。機能を選択すると、その機能に含まれるすべてのサービスが表示されます。*

[サービス] ビューの [サービス] エクスプローラーでサービスを選択して、次の設定を構成することができます。

インストールの設定

[インストールの設定] 領域を使って、サービスの表示名や説明などの情報、およびサービスの開始条件を指定します。

テーブル 3-51 ・ インストールの設定

設定	説明
ターゲット	この設定は、サービスを含む実行可能ファイルを指定します。これを変更するには、この設定の省略記号ボタン (...) をクリックします。
表示名	サービス コントロール マネージャーにおける、このサービスの表示名を入力します。この設定を空白のままにすると、サービスの名前 ([詳細設定] ノードの下にあるサービスのサブノードの名前に使用されているテキスト) が使用されます。

テーブル 3-51・インストールの設定 (続き)

設定	説明
説明	サービスの説明を入力します。この説明は、サービスがインストールされる時にターゲット システムに登録され、サービス コントロール マネージャーの [説明] 列に表示されます。また、サービスの [プロパティ] ダイアログ ボックスの [一般] タブにある [説明] ボックスにも表示されます。
サービスの種類	インストールするサービスの種類を選択します。選択可能なオプションは以下のとおりです: ・ 独自のプロセスで実行するWin32 ・ プロセスを共有するWin32 WIN32_OWN_PROCESS タイプのサービスには、1 つのサービスのコードが含まれます。WIN32_SHARE_PROCESS タイプのサービスには、1 つ以上のサービスのコードが含まれており、コードの共有が可能です。
デスクトップと対話する	サービスがデスクトップとインタラクトするかどうかを指定します。サービスにユーザー インターフェイスが含まれている場合は、[はい] を選択します。 [はい] を選択した場合、"ユーザー名" 設定は空白のままでなくてはなりません。これは、サービスが組み込み LocalSystem アカウントで実行するようにインストールされるためです。
開始の種類	サービスの開始時期を指定してください。選択可能なオプションは以下のとおりです: ・ **自動** – サービスはシステムが開始したときに自動的に開始します。 ・ **オン デマンド** – サービスは、サービス コントロール マネージャーを通して要求されたときに開始します。 ・ **無効** – サービスを開始することはできません。 一部のサービスは、別の開始タイプ (オペレーティング システムの初期化中、またはオペレーティング システム ローダによる) をサポートします。ただし、Windows Installer はこれらのサポートを含まないため、これらのオプションは "開始の種類" 設定では使用できません。
エラーコントロール	サービスの開始に失敗した場合に、サービス コントロール マネージャーが行う適切な操作を選択します。選択可能なオプションは以下のとおりです: ・ エラーをログに記録してから作業を継続する ・ エラーをログに記録してから、メッセージを表示して作業を継続する ・ エラーをログに記録して再起動する

テーブル 3-51・インストールの設定 (続き)

設定	説明
失敗時にインストールを中止する	サービスがターゲット システムにインストールできない場合は、インストール全体を失敗とするかどうかを指定します。デフォルト値は [いいえ] です。 ***メモ:*** *この設定で [はい] を選択して、エンド ユーザーがサイレントまたは基本の UI モードでインストールを実行する場合、ターゲット システムには Windows Installer 3 以降が必須となります。*
ロード順序グループ	このサービスがメンバーであるロード順グループがある場合、その名前を入力します。
依存関係	このサービスが必要とするサービス、またはロード順グループを入力してください。このサービスを開始する前に、システムは必要なサービス、またはロード順グループから最低 1 つのメンバーの開始を試行します。 複数の依存関係はコンマ (,) で区切ります。 サービス コントロール マネージャーがサービスとロード順グループを区別できるように、各ロード順グループ名の前に SC_GROUP_IDENTIFIER (通常は、プラス記号 (+)) を付ける必要があります。
ユーザー名	サービスにログオンするアカウントを入力してください。ローカル システム アカウントの下にサービスをインストールするには、この設定を空白のままに残します。(Microsoft は単一ユーザーの権限を偽装するサービスのインストールを推奨しません。) サービス タイプが**独自のプロセスで実行する Win32** の場合、入力する値には次の形式を使用します: **ドメイン名¥ユーザー名** サービスがビルトイン ドメインの下にログ記録される場合、次の形式を使用します: **.¥UserName**
Password	このサービスのパスワードを入力してください。"ユーザー名" 設定が空白の場合 (つまり、サービスがローカル システム アカウントの下でログオンをされているとき) は、この設定を空白に残してください。ユーザー名を指定していない限りパスワードは使用されません。
開始パラメーター	サービスの実行に必要なコマンドライン パラメーターまたはプロパティを入力します。
インストール時に開始	インストール中にサービスを開始するかどうかを指定します。 [はい] を選択する場合、"引数" 設定でサービスに渡す引数を指定します。

テーブル 3-51 • インストールの設定 (続き)

設定	説明
引数	サービスに渡す引数を指定します。複数の引数はコンマ (,) で区切ります。

[セットアップ外観のカスタマイズ] ビュー

セットアップの外観は、競合他社に差をつける上で非常に重要です。セットアップの外観や動作は、以下に示すビューで簡単にカスタマイズできます。

テーブル 3-52 • [セットアップ概観のカスタマイズ] ビューの下にあるビュー

表示	説明
ダイアログ	[ダイアログ] ビューでは、セットアップ過程で表示されるダイアログを選択することができます。また、選択したダイアログにカスタムバナーやイメージを追加して、その外観をカスタマイズすることができます。
ビルボード	ビルボードとは、インストールのファイル転送段階で、指定された時間表示されるイメージまたは Adobe Flash アプリケーション ファイルです。ビルボードは、エンド ユーザーと連絡を取ったり、広告、教育、およびエンターテインメントを提示するために使用することができます。たとえば、ビルボードを使ってインストール中の製品に含まれる新しい機能の概要や貴社の他の製品についての情報を提供できます。各ビルボードは、貴社のグラフィック担当者がファイル転送の外観を完全にカスタマイズできるファイルです。
テキストとメッセージ	セットアップで使用される文字列は、[テキストとメッセージ] ビューからアクセスできます。このビューでは、ダイアログに表示されるテキストやエンドユーザーに対して表示されるエラーメッセージを変更することができます。

[ダイアログ] ビュー

セットアップのユーザー インターフェイスは様々な意味で重要ですが、特に、エンドユーザーの入力や設定が通常ユーザー インターフェイスを通じて処理されるという点で非常に重要です。ユーザー インターフェイスで目的の画面を表示することが困難だったり、理解しにくいと、ユーザーは製品をインストールする際、問題を抱えることになります。インストールの作成手順や、エンドユーザーの操作を簡素化するために、InstallShield ではいくつかの定義済みダイアログの利用が可能です。

使用できるダイアログは提供されているものに限られていますが、それらの多くはカスタマイズでき、必要な外観や機能性を持たせることができます。たとえば、ダイアログそれぞれの上部にカスタマイズしたイメージを追加できるので、会社のロゴを使用して強い印象を与えることができます。

両方のダイアログ テーマのサンプル スクリーン ショットおよび各ダイアログのカスタマイズに関する情報について、次を参照してください:

- [スプラッシュ] ダイアログ
- [インストール – ようこそ] ダイアログ

- [使用許諾契約] ダイアログ
- Readme ダイアログ
- [ユーザー情報] ダイアログ
- [インストール先のフォルダー] ダイアログ
- [データベース フォルダー] ダイアログ
- [セットアップの種類] ダイアログ
- [カスタム セットアップ] ダイアログ
- [インストールの準備完了] ダイアログ
- [セットアップ進行状況] ダイアログ
- [セットアップの完了] ダイアログ

プロジェクトのすべてのダイアログに影響するグローバル設定に関する情報については、「すべてのエンドユーザー ダイアログのためのグローバル ダイアログ設定」を参照してください。

[ビルボード] ビュー

ビルボードをプロジェクトに追加して、インストール処理中にエンド ユーザーに対して情報を提供できます。ビルボードは、エンド ユーザーと連絡を取ったり、広告、教育、およびエンターテインメントを提示するために使用することができます。たとえば、ビルボードを使ってインストール中の製品に含まれる新しい機能の概要や貴社の他の製品についての情報を提供できます。各ビルボードは、貴社のグラフィック担当者がファイル転送の外観を完全にカスタマイズできるファイルです。

プロジェクトに 1 つ以上のビルボードを追加すると、実行時に Windows Installer がシステムに追加している変更について[セットアップの進行状況] ダイアログ がレポートを提示するのと同時に、ビルボードが表示されます。ビルボードが表示される時間および場所は、[ビルボード] ビューで設定を構成することによって制御できます。

[ビルボード] ビューの設定についての詳細は、次の項目を参照してください:

- ビルボード設定 – これらは、プロジェクト共通のビルボードの設定です。
- Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定 – これらの設定は、中央ペインで Flash ビルボードまたはイメージ ビルボードをクリックすると、[ビルボード] ビューの右側のペインに表示されます。

ビルボード設定

[ビルボード] ビューの中央ペインで [ビルボード] エクスプローラーをクリックすると、右側のペインに以下の設定が表示されます。これらは、プロジェクト共通のビルボードの設定です。

テーブル 3-53 • ビルボード設定

設定	説明
ビルボードの種類	インストールに使用するビルボードの種類を選択します。選択可能なオプションは以下のとおりです: • **全画面表示、右下に小さい進行状況ボックスを表示** – インストールが標準エンドユーザー ダイアログを表示するときに、全画面の背景も表示します。ファイルの転送中、インストールが全画面背景を使用し、ビルボードを前画面に、また小さい進行状況ボックスを画面の右下に表示します。 • **ウィンドウ表示、標準の進行状況を表示する** – ファイルの転送中、インストールはビルボードを表示する標準サイズのダイアログを表示します。このダイアログの下の部分に、進行状況バーが表示されます。このスタイルの場合、インストールは背景を表示しません。 • **ウィンドウ表示、右下に小さい進行状況ボックスを表示する(ビルボードなし)** – インストールは、ファイル転送中にインストールは小さい進行状況ボックスを画面の右下に表示します。 さらに詳しい情報、ならびに各ビルボード タイプのサンプル スクリーンショットは、「ビルボードの種類」を参照してください。
ビルボードのループ	インストールがファイルの転送が完了するまでイメージ ビルボードをループして、適切な [セットアップの完了] ダイアログを表示するかどうかを指定します。 この設定で [いいえ] を選択して、ビルボードに割り当てた時間よりもファイルの転送に時間がかかった場合、インストールはファイルの転送が終了するまで、最後のイメージ ビルボードを表示し続けます。 この設定で [はい] を選択して、ビルボードに割り当てた時間よりもファイルの転送に時間がかかった場合、インストールは最初のビルボードから再び表示します。必要な場合、ループはファイルの転送が終了するまで継続します。 この設定のデフォルト値は [いいえ] です。 この設定は、Adobe Flash アプリケーション ファイル ビルボードには効果がありません。

Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定

Flash またはイメージ ビルボードの設定では、表示するファイル、その表示時間、および画面上の表示位置を決定します。これらの設定にアクセスするには、[ビルボード] ビューを開いて、[ビルボード] エクスプローラーから構成するビルボードを選択します。

テーブル 3-54 • Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定

設定	説明
ファイル名	以下のいずれかを実行します。 • **Adobe Flash アプリケーション ファイル ビルボードの場合** – 選択されたビルボードに使用する Flash アプリケーション ファイル (.swf) へのパスを入力するか、省略記号 (...) ボタンをクリックしてファイルを参照します。 Flash アプリケーション ファイルは、ビデオ、動画、音声、インタラクティブ インターフェイス、ゲーム、テキスト、その他の .swf ファイルがサポートするあらゆる要素で構成されます。Flash ビデオ ファイル (.flv) や MP3 オーディオ ファイルは .swf ファイルに埋め込んで、ファイル転送中にターゲット システム上のローカルで使用できるようにすることが推奨されます。.swf ファイルは Web サイト上に配置された外部ファイルを参照することが可能ですが、この外部実装ではエンド ユーザーがインターネットに接続されていることが必須となります。 • **イメージ ビルボードの場合** – 選択されたビルボードに使用するイメージ ファイル (.bmp、.gif、.jpg、または .jpeg) へのパスを入力するか、省略記号 (...) ボタンをクリックしてファイルを参照します。 動画 .gif ファイルはサポートされていませんので、ご注意ください。ビルボードで動画を使用したい場合は、Adobe Flashアプリケーション ファイル ビルボードの使用をご検討ください。 ***メモ:*** *.swf ファイルの作成に使用した Flash またはその他のツールのバージョンがターゲット システムにインストールされている Flash Player よりも新しい場合、ターゲット システム上で一部の Flash 機能が予定どおりに動作しない可能性があります。*
期間	ビルボードが表示される時間を、秒単位で入力してください。入力する数字は、1 から 32767 まで (9 時間と少し) でなくてはなりません。 期間が実行時の動作に及ぼす影響は、インストールが Flash ビルボードかイメージ ビルボードのどちらを表示するかによって異なります。詳細については、「ビルボードを含むインストールの実行時の動作」を参照してください。

テーブル 3-54 • Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定 (続き)

設定	説明
元の場所	画面上でビルボードを表示する場所を選択します。選択可能なオプションは以下のとおりです: • 右上 • 左上 • 右下 • 左下 • 中央揃え X および Y 座標は、この点から測定されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
X 軸	ビルボードの横位置を "原点" 設定で選択した場所に相対して変更するには、距離をピクセルで指定します。たとえば、ビルボードの原点が左下で、X 座標値が 100 の場合、画面の左側から 100 ピクセル離れたビルボードの左側に表示されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
Y 軸	ビルボードの縦位置を "原点" 設定で選択した場所に相対して変更するには、距離をピクセルで指定します。たとえば、ビルボードの原点が左下で、Y 座標値が 100 の場合、画面の下側から 100 ピクセル離れたビルボードの下側に表示されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
効果	このビルボードの移動効果を選択します。画面に表示されて、指定した時間の経過後に非表示にするだけではなく、移動効果では、ビルボードとビルボードの変更がよりスムーズに行われます。 ***メモ:*** *この設定は、イメージ ビルボードに適用されますが、Adobe Flash アプリケーション ファイル ビルボードには適用されません。* *また、この設定は [ビルボードの種類] 設定で、****[ウィンドウ表示、標準の進行状況を表示する]*** *オプションまたは、****[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*

テーブル 3-54 • Adobe Flash アプリケーション ファイル ビルボードとイメージ ビルボードの設定 (続き)

設定	説明
背景色	この設定には、現在選択されているビルボードの背景色が表示されます。この色を変更するには、省略記号ボタン (...) をクリックします。背景色に希望する色を選択、またはカスタム色を定義できる [色] ダイアログ ボックスが表示されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
タイトル	背景の左上に表示される、このビルボードのタイトルを入力します。 ***ヒント:*** *インストールが実行時にタイトルとして表示できる最大文字数は、そのフォント、フォント サイズ、フォントの属性、およびこの設定に指定するタイトル文字列の長さによって異なります。また、ターゲット システムのスクリーン解像度にも左右されます。このため、長いタイトルを指定する場合は、異なるスクリーン解像度を使ってビルボードをプレビューして、タイトル全体が実行時に表示されるかどうかをテストしてください。* ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
背景スタイル	使用する背景のスタイルを選択します。選択可能なオプションは以下のとおりです: • **グラデーション** – 背景は濃い色から薄い色へ徐々に変化します。 • **純色** – 背景は単一色で表示されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*
フォント	省略記号 (...) ボタンをクリックして、選択されたビルボードの背景にあるタイトルに使用するフォントを選択します。ターゲットマシンに指定したフォントがない場合、デフォルトのシステム フォントが使用されます。 ***メモ:*** *この設定は、"ビルボードの種類" 設定で* ***[全画面表示、右下に小さい進行状況ボックスを表示する]*** *オプションが選択されている場合のみ有効です。*

[テキストとメッセージ] ビュー

通常、ユーザーがアプリケーションを使用するにあたって最初に経験するのは、ソフトウェアのインストールです。セットアッププロセスの操作が難しかったり、ダイアログのメッセージがわかりにくかったりすると、第一印象が悪くなることもあります。そうならないようにするために、セットアップで表示されるテキストを、すべてカスタマイズすることができます。また、すべてのランタイム文字列をエクスポートして翻訳し、グローバル化されたセットアップ バージョンの IDE に新しい言語をインポートすることもできます。

[テキストとメッセージ] ビューには、インストール プロセス中に表示できるすべてのダイアログとメッセージが一覧表示されます。これらのダイアログの 1 つをクリックすると、そのダイアログの画像、およびそのダイアログに関連付けられているすべての文字列が表示されます。このビューからはすべてのランタイム文字列にアクセスできます。

[セットアップ要件およびアクションの定義] ビュー

カスタム アクションとセットアップ要件は、どちらも Windows Installer サービスでサポートされていない機能を実行するために役立つ方法です。カスタム アクションを使って、実行可能ファイルを起動、スクリプトを実行、または DLL から関数を呼び出すことができます。要件を設定することで、ソフトウェアのハードウェア要件に一致しないマシンに対して製品がインストールされないようにできます。

テーブル 3-55 • [セットアップ要件およびアクションの定義] ビューの下にあるビュー

表示	説明
セットアップ要件	特定のハードウェアまたはオペレーティング システム要件が満たされなければアプリケーションが正常に動作しないような場合には、これらの要件を満たさないマシンにはアプリケーションがインストールされないようにすることができます。
カスタム アクション	カスタム アクションでは、Windows Installer が本来サポートしていない機能をインストールに追加することができます。これらのアクションは、Readme ファイルの表示からレジストリ エントリの消去などあらゆる用途に使用できます。
セットアップ ファイル	インストール時に、特定のファイルをセットアップで使用できるようにする場合、これらのファイルを [セットアップファイル] ビューに追加することができます。

[要件] ビュー

[要件] ビューでは、アプリケーションが正しく動作するために必要なソフトウェアと特定のマシン環境を指定することができます。システムまたはソフトウェアの必要条件が満たされない場合、インストールは終了します。ハードウェアまたはソフトウェアの必要条件に合わないシステムにはアプリケーションをインストールできないので、必要条件を自由に設定してください。

システム ハードウェア要件

次のテーブルには可能なオプションが記述されています。

テーブル 3-56・システム要件の条件

条件	説明
OS バージョン	オペレーティング システム要件を指定または変更するには、この設定の省略記号ボタン (...) をクリックします。 この設定のデフォルト値は (任意の OS バージョン) で、製品が特定のオペレーティング システムを必要としないことを示します。 この設定に特定のオペレーティング システムがリストされている場合、インストールはそれらのオペレーティングシステム上で実行可能となります。リストされていないオペレーティング システムでインストールが実行されると、インストールが終了して、製品はインストールされません。
プロセッサ	製品に最低限必要なプロセッサを選択します。
RAM	製品を実行するのに最低限必要な RAM を選択します。
画面の解像度	製品に最低限必要な画面解像度を選択します。
カラー深度	製品に最低限必要なカラー深度を選択します。

システム ソフトウェア要件

ソフトウェア要件の指定についての詳細は、「製品のソフトウェアの要件を指定する」を参照してください。

[カスタム アクション] ビュー

Windows Installer は多機能ですが、必要な機能をすべて備えているわけではありません。そのため Microsoft では、カスタム アクションを使用して、Windows Installer で直接サポートされていない機能をインストールに柔軟に追加できるようにしています。

InstallShield は、DLL 関数を呼び出すか、実行可能ファイルを起動するか、または VBScript あるいは JScript を使用するカスタム アクションをサポートします。カスタム アクションの使用が推奨されるシナリオは、例えば次のような場合です。

- インストール実行時にエンド ユーザーが入力したシリアル番号を検証する
- 製品がインストールされているかどうかを確認する
- インストール後にターゲット マシンを再起動する

[カスタム アクション] ビューでカスタム アクションを追加する場合、使用可能なシーケンスの 1 つにそのカスタム アクションをスケジュールしなくてはなりません。たとえば、インストール中、[セットアップの完了] ダイアログが表示された直後に特定のカスタム アクションを起動する場合があります。その場合、カスタム アクションを [カスタム アクション] ビューの **[セットアップの完了] ダイアログの後**項目に追加します。アクションをシーケンスに追加した後、必要に応じてスケジュールし直すこともできます。詳細については、「カスタム アクションを起動するタイミングの変更」を参照してください。

[カスタム アクション] ビューでカスタム アクションを選択すると、その設定を構成することができます。詳しくは、次を参照してください:

- MSI DLL カスタム アクションの設定
- DLL カスタム アクション設定
- 実行可能ファイル カスタム アクションの設定
- VBScript カスタム アクションの設定
- JScript カスタム アクションの設定

カスタム アクションの各種類の比較は、「カスタム アクションを使用する」を参照してください。

MSI DLL カスタム アクションの設定

Windows Installer DLL (MSI DLL) カスタム アクションを [カスタム アクション] ビューを使ってプロジェクトに追加する場合、その設定を構成する必要があります。

テーブル 3-57 • MSI DLL カスタム アクションの設定

設定	説明
ソースの場所	カスタム アクションに使用されるファイルのある場所を指定します: • **ファイル システムの参照**—このカスタム アクションに使用されるファイルはシステム上にあります。 • **製品と一緒にインストール**—ファイルはターゲット システムにインストールされます。 カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分にスケジュールするかによって、このオプションが使用できない場合もあります。たとえば、インストールの [ファイル転送の前] 部分でカスタム アクションをスケジュールした場合、カスタム アクションはファイルがターゲット システムにインストールされる前に実行されるため、[製品と一緒にインストール] オプションは使用することはできません。
ファイル名	"ソースの場所" 設定で [ファイルシステムの参照] を選択した場合、"ファイル名" 設定で省略記号ボタン (...) をクリックして、カスタム アクションに使用する DLL ファイルを参照します。インストールをビルドすると、このファイルは自動的にインストールに組み込まれます。カスタム アクションはインストール中に実行されますが、インストールのあとファイルは削除されます。 "ソースの場所" 設定で [製品と一緒にインストール] を選択した場合、"ファイル名" 設定で省略記号ボタン (...) をクリックして、[ファイル] ビューを使ってプロジェクトに追加したファイルの一覧からファイルを指定できます。ターゲット システムにインストールされているファイルが、ターゲット システムのインストール先から起動されます。

テーブル 3-57・MSI DLL カスタム アクションの設定 (続き)

設定	説明
関数名	Windows Installer DLL から呼び出すエントリポイント関数の名前を入力します。 関数は以下ののシグネチャで定義しなくてはなりません: UINT _stdcall *FunctionName* (MSIHANDLE hInstall) {...} 関数名には異なる名前を使用できますが、戻り値の型、呼び出し規則、および単一のパラメーターには、前述のシグネチャで指定された型を使用する必要があります。関数シグネチャで使用される MSIHANDLE データ型は、実行中のインストールへのハンドルです。
スクリプト内実行	選択したカスタム アクションをトリガーするシーケンスの反復を選択します。各オプションの詳細については、「アクションの実行のオプション」を参照してください。 この設定は、すべてのシーケンスに適用するわけではありません。たとえば、**初期化の後 (最初のダイアログの前)** の下にカスタム アクションを追加した場合、そのアクションは即時実行としてスケジュールされます。遅延、ロールバック、コミット実行としてスケジュールすることはできません。
アクションの待機	カスタム アクションが終了するまでインストール、アンインストール、またはメンテナンスを待機してから、残りのインストール処理を行うかどうかを指定します。 [いいえ] を選択すると、カスタム アクションを実行中にインストール、アンインストール、またはメンテナンスが続行します。
終了コードを無視する	Windows Installer がこのカスタム アクションの戻り値を無視して、インストール、アンインストール、またはメンテナンスを続行するかどうかを指定します。
Comments	このカスタム アクションに関するコメントを入力します。入力したコメントは参照する目的でのみ使用され、エンドユーザーに表示されることはありません。
条件	この設定を使って、選択したカスタム アクションを実行するためには True 評価が必要な 1 つ以上の条件を指定できます。たとえば、ターゲット システムで特定のオペレーティング システムまたは最小システム要件をチェックする条件を作成できます。実行時に条件が True 評価されると、カスタム アクションが実行します。1 つ以上の条件が False 評価されると、カスタム アクションは実行しません。 1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。 条件を追加すると、InstallShield によって "条件" 設定の下に新しい設定が追加されます。この新しい設定は、追加された条件の条件ステートメントを表示します。 条件を編集または削除するには、"条件" 設定で省略記号ボタン (...) をクリックします。

DLL カスタム アクション設定

DLL カスタム アクションを [カスタム アクション] ビューに追加する場合、その設定を構成する必要があります。

テーブル 3-58 • DLL カスタム アクションの設定

設定	説明
ソースの場所	カスタム アクションに使用されるファイルのある場所を指定します: • **ファイル システムの参照**—このカスタム アクションに使用されるファイルはシステム上にあります。 カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分にスケジュールするかによって、このオプションが使用できない場合もあります。たとえば、カスタム アクションをインストールの [ファイル転送の後] にスケジュールした場合、このソースの場所オプションは該当しない場合があります。 • **製品と一緒にインストール**—ファイルはターゲット システムにインストールされます。 カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分にスケジュールするかによって、このオプションが使用できない場合もあります。たとえば、インストールの [ファイル転送の前] 部分でカスタム アクションをスケジュールした場合、カスタム アクションはファイルがターゲット システムにインストールされる前に実行されるため、[製品と一緒にインストール] オプションは使用することはできません。 MSI DLL カスタム アクションは、標準 DLL カスタム アクションよりもソースの場所オプションにおける柔軟性に優れています。詳細については、「Windows Installer DLL カスタム アクション」を参照してください。
ファイル名	″ソースの場所″ 設定で [ファイルシステムの参照] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、カスタム アクションに使用する DLL ファイルを参照します。インストールをビルドすると、このファイルは自動的にインストールに組み込まれます。カスタム アクションはインストール中に実行されますが、インストールのあとファイルは削除されます。 ″ソースの場所″ 設定で [製品と一緒にインストール] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、[ファイル] ビューを使ってプロジェクトに追加したファイルの一覧からファイルを指定できます。ターゲット システムにインストールされているファイルが、ターゲット システムのインストール先から起動されます。
関数名	DLL から呼び出す関数名を入力します。関数によりパラメーターが受け入れられる必要がありますが、この設定にパラメーターを一覧表示する必要はありません。たとえば、使用する関数が以下のプロトタイプを持つ場合、関数名に MyFunction と入力します。 CHAR WINAPI MyFunction(HWND, LPSTR , LPSTR, LPSTR, LPSTR); 詳細については、「Classic DLL カスタム アクション関数プロトタイプ」または「新しい DLL カスタム アクション関数プロトタイプ」を参照してください。

テーブル 3-58 • DLL カスタム アクションの設定 (続き)

設定	説明
関数の署名	DLL 関数が使用するフォーマットを選択します。選択可能なオプションは以下のとおりです: • **クラシック**—このフォーマットの場合、エントリポイント関数の正確なプロトタイプが必要です。このフォーマットは、InstallShield Express の古いバージョンで使用されています。詳細については、「Classic DLL カスタム アクション関数プロトタイプ」を参照してください。 • **新規作成**—このフォーマットを使って、現在実行中の .msi データベースのハンドルを取得することができます。データベースのハンドルを取得すると、任意の Windows Installer API を呼び出すことができます。詳細については、「新しい DLL カスタム アクション関数プロトタイプ」を参照してください。 この種類の DLL のプロトタイプは、MSI DLL カスタム アクションが必要とするシグネチャと同じシグネチャを使用します。プロジェクトに追加する新しいカスタム アクションにこの種類のシグネチャを使用する場合、標準 DLL よりも柔軟なスケジュール オプションが提供されている MSI DLL カスタム アクションを使用することをお勧めします。詳細については、「Windows Installer DLL カスタム アクション」を参照してください。
スクリプト内実行	選択したカスタム アクションをトリガーするシーケンスの反復を選択します。各オプションの詳細については、「アクションの実行のオプション」を参照してください。 この設定は、すべてのシーケンスに適用するわけではありません。たとえば、**初期化の後 (最初のダイアログの前)** の下にカスタム アクションを追加した場合、そのアクションは即時実行としてスケジュールされます。遅延、ロールバック、コミット実行としてスケジュールすることはできません。
アクションの待機	カスタム アクションが終了するまでインストール、アンインストール、またはメンテナンスを待機してから、残りのインストール処理を行うかどうかを指定します。 [いいえ] を選択すると、カスタム アクションを実行中にインストール、アンインストール、またはメンテナンスが続行します。
終了コードを無視する	Windows Installer がこのカスタム アクションの戻り値を無視して、インストール、アンインストール、またはメンテナンスを続行するかどうかを指定します。
Comments	このカスタム アクションに関するコメントを入力します。入力したコメントは参照する目的でのみ使用され、エンドユーザーに表示されることはありません。

テーブル 3-58 • DLL カスタム アクションの設定 (続き)

設定	説明
条件	この設定を使って、選択したカスタム アクションを実行するためには True 評価が必要な 1 つ以上の条件を指定できます。たとえば、ターゲット システムで特定のオペレーティング システムまたは最小システム要件をチェックする条件を作成できます。実行時に条件が True 評価されると、カスタム アクションが実行します。1 つ以上の条件が False 評価されると、カスタム アクションは実行しません。 1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。 条件を追加すると、InstallShield によって "条件" 設定の下に新しい設定が追加されます。この新しい設定は、追加された条件の条件ステートメントを表示します。 条件を編集または削除するには、"条件" 設定で省略記号ボタン (...) をクリックします。

実行可能ファイル カスタム アクションの設定

.exe ファイルを起動するカスタム アクションを [カスタム アクション] ビューに追加する場合、その設定を構成する必要があります。

テーブル 3-59 • 実行可能ファイル カスタム アクションの設定

設定	説明
ソースの場所	カスタム アクションに使用されるファイルのある場所を指定します: • **ファイル システムの参照**—このカスタム アクションに使用されるファイルはシステム上にあります。 • **製品と一緒にインストール**—ファイルはターゲット システムにインストールされます。カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分でスケジュールするかによっては、[製品と一緒にインストール] オプションが使えない場合もあります。たとえば、インストールの [ファイル転送の前] 部分でカスタム アクションをスケジュールした場合、カスタム アクションはファイルがターゲット システムにインストールされる前に実行されるため、[製品と一緒にインストール] オプションは使用することはできません。 • **ファイルがターゲット システムに存在する**—ファイルがターゲット システム上に既存します。 ***ヒント:*** *[ターゲット システムに存在するファイル] を選択し、"ファイルの場所" 設定を* [SUPPORTDIR] *に設定すると、[セットアップ ファイル] ビューに追加したファイルをカスタム アクションとして起動することができます。*

テーブル 3-59・実行可能ファイル カスタム アクションの設定 (続き)

設定	説明
ファイル名	″ソースの場所″ 設定で [ファイルシステムの参照] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、カスタム アクションに使用する .exe ファイルを参照します。インストールをビルドすると、このファイルは自動的にインストールに組み込まれます。カスタム アクションはインストール中に実行されますが、インストールのあとファイルは削除されます。 ″ソースの場所″ 設定で [製品と一緒にインストール] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、[ファイル] ビューを使ってプロジェクトに追加したファイルの一覧からファイルを指定できます。ターゲット システムにインストールされているファイルが、ターゲット システムのインストール先から起動されます。 この設定は、″ソースの場所″ 設定で次のオプションの 1 つが選択されている場合に表示されます: ・ ファイルシステムの参照 ・ 製品と一緒にインストール
ファイルの場所	起動する実行可能ファイルを含むターゲット システム上にあるフォルダーを選択するか、省略記号ボタン (...) をクリックして、フォルダーを選択または作成します。 たとえば、Notepad.exe を起動する場合、″ファイルの場所″ 設定に **[SystemFolder]** と入力し、″ファイル名とコマンドライン″ 設定に **Notepad.exe** と入力します。 この設定は、″ソースの場所″ 設定で [ファイルがターゲット システムに存在する] が選択されている場合のみ表示されます。
コマンド ライン	実行可能ファイルに渡すコマンドライン パラメーターを入力します。たとえば、実行可能ファイルとして Notepad を選択した場合、コマンドラインでファイル名を渡して実行可能ファイルが起動されるときにファイルを開くことができます。この場合、**E:/Readme.txt** と入力すると、Readme ファイルを起動できます。長いファイル名は引用符で囲んでください。 この設定は、″ソースの場所″ 設定で次のオプションの 1 つが選択されている場合に表示されます: ・ ファイルシステムの参照 ・ 製品と一緒にインストール
ファイル名とコマンドライン	ファイルの名前を入力します。ファイル名の前にはディレクトリを付けません。これは ″ファイルの場所″ 設定に表示されます。ファイル名の後、実行可能ファイルに送る任意のコマンドラインを入力します。 たとえば、Notepad.exe を起動する場合、″ファイルの場所″ 設定に **[SystemFolder]** と入力し、″ファイル名とコマンドライン″ 設定に **Notepad.exe** と入力します。 この設定は、″ソースの場所″ 設定で [ファイルがターゲット システムに存在する] が選択されている場合のみ表示されます。

テーブル 3-59・実行可能ファイル カスタム アクションの設定 (続き)

設定	説明
スクリプト内実行	選択したカスタム アクションをトリガーするシーケンスの反復を選択します。各オプションの詳細については、「アクションの実行のオプション」を参照してください。 この設定は、すべてのシーケンスに適用するわけではありません。たとえば、**初期化の後 (最初のダイアログの前)** の下にカスタム アクションを追加した場合、そのアクションは即時実行としてスケジュールされます。遅延、ロールバック、コミット実行としてスケジュールすることはできません。
アクションの待機	カスタム アクションが終了するまでインストール、アンインストール、またはメンテナンスを待機してから、残りのインストール処理を行うかどうかを指定します。 [いいえ] を選択すると、カスタム アクションを実行中にインストール、アンインストール、またはメンテナンスが続行します。
終了コードを無視する	Windows Installer がこのカスタム アクションの戻り値を無視して、インストール、アンインストール、またはメンテナンスを続行するかどうかを指定します。
Comments	このカスタム アクションに関するコメントを入力します。入力したコメントは参照する目的でのみ使用され、エンドユーザーに表示されることはありません。
条件	この設定を使って、選択したカスタム アクションを実行するためには True 評価が必要な 1 つ以上の条件を指定できます。たとえば、ターゲット システムで特定のオペレーティング システムまたは最小システム要件をチェックする条件を作成できます。実行時に条件が True 評価されると、カスタム アクションが実行します。1 つ以上の条件が False 評価されると、カスタム アクションは実行しません。 1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。 条件を追加すると、InstallShield によって ″条件″ 設定の下に新しい設定が追加されます。この新しい設定は、追加された条件の条件ステートメントを表示します。 条件を編集または削除するには、″条件″ 設定で省略記号ボタン (...) をクリックします。

VBScript カスタム アクションの設定

VBScript カスタム アクションを [カスタム アクション] ビューに追加する場合、その設定を構成する必要があります。

テーブル 3-60 • VBScript カスタム アクションの設定

設定	説明
ソースの場所	カスタム アクションに使用されるファイルのある場所を指定します: • **ファイル システムの参照**—このカスタム アクションに使用されるファイルはシステム上にあります。 • **製品と一緒にインストール**—ファイルはターゲット システムにインストールされます。カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分でスケジュールするかによっては、[製品と一緒にインストール] オプションが使えない場合もあります。たとえば、インストールの [ファイル転送の前] 部分でカスタム アクションをスケジュールした場合、カスタム アクションはファイルがターゲット システムにインストールされる前に実行されるため、[製品と一緒にインストール] オプションは使用することはできません。
ファイル名	″ソースの場所″ 設定で [ファイルシステムの参照] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、カスタム アクションに使用する VBScript ファイルを参照します。インストールをビルドすると、このファイルは自動的にインストールに組み込まれます。カスタム アクションはインストール中に実行されますが、インストールのあとファイルは削除されます。 ″ソースの場所″ 設定で [製品と一緒にインストール] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、[ファイル] ビューを使ってプロジェクトに追加したファイルの一覧からファイルを指定できます。ターゲット システムにインストールされているファイルが、ターゲット システムのインストール先から起動されます。
関数名	.vbs ファイルで関数を定義した場合、実行する関数名を入力します。これは、VBScript の後で実行されます。VBScript 関数は、引数をカスタム アクションから取得できません。さらに、カスタム アクションは最初にスクリプトを実行してから、指定されている関数がある場合それを呼び出すので、関数は不要です。
スクリプト内実行	選択したカスタム アクションをトリガーするシーケンスの反復を選択します。各オプションの詳細については、「アクションの実行のオプション」を参照してください。 この設定は、すべてのシーケンスに適用するわけではありません。たとえば、**初期化の後 (最初のダイアログの前)** の下にカスタム アクションを追加した場合、そのアクションは即時実行としてスケジュールされます。遅延、ロールバック、コミット実行としてスケジュールすることはできません。

テーブル 3-60 • VBScript カスタム アクションの設定 (続き)

設定	説明
アクションを待機する	カスタム アクションが終了するまでインストール、アンインストール、またはメンテナンスを待機してから、残りのインストール処理を行うかどうかを指定します。 [いいえ] を選択すると、カスタム アクションを実行中にインストール、アンインストール、またはメンテナンスが続行します。
終了コードを無視する	Windows Installer がこのカスタム アクションの戻り値を無視して、インストール、アンインストール、またはメンテナンスを続行するかどうかを指定します。
Comments	このカスタム アクションに関するコメントを入力します。入力したコメントは参照する目的でのみ使用され、エンドユーザーに表示されることはありません。
条件	この設定を使って、選択したカスタム アクションを実行するためには True 評価が必要な 1 つ以上の条件を指定できます。たとえば、ターゲット システムで特定のオペレーティング システムまたは最小システム要件をチェックする条件を作成できます。実行時に条件が True 評価されると、カスタム アクションが実行します。1 つ以上の条件が False 評価されると、カスタム アクションは実行しません。 1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。 条件を追加すると、InstallShield によって "条件" 設定の下に新しい設定が追加されます。この新しい設定は、追加された条件の条件ステートメントを表示します。 条件を編集または削除するには、"条件" 設定で省略記号ボタン (...) をクリックします。

JScript カスタム アクションの設定

JScript カスタム アクションを [カスタム アクション] ビューに追加する場合、その設定を構成する必要があります。

テーブル 3-61 • JScript カスタム アクションの設定

設定	説明
ソースの場所	カスタム アクションに使用されるファイルのある場所を指定します: • **ファイル システムの参照**—このカスタム アクションに使用されるファイルはシステム上にあります。 • **製品と一緒にインストール**—ファイルはターゲット システムにインストールされます。カスタム アクションをインストール、アンインストール、またはメンテナンスのどの部分でスケジュールするかによっては、[製品と一緒にインストール] オプションが使えない場合もあります。たとえば、インストールの [ファイル転送の前] 部分でカスタム アクションをスケジュールした場合、カスタム アクションはファイルがターゲット システムにインストールされる前に実行されるため、[製品と一緒にインストール] オプションは使用することはできません。
ファイル名	″ソースの場所″ 設定で [ファイルシステムの参照] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、カスタム アクションに使用する JScript ファイルを参照します。インストールをビルドすると、このファイルは自動的にインストールに組み込まれます。カスタム アクションはインストール中に実行されますが、インストールのあとファイルは削除されます。 ″ソースの場所″ 設定で [製品と一緒にインストール] を選択した場合、″ファイル名″ 設定で省略記号ボタン (...) をクリックして、[ファイル] ビューを使ってプロジェクトに追加したファイルの一覧からファイルを指定できます。ターゲット システムにインストールされているファイルが、ターゲット システムのインストール先から起動されます。
関数名	.js ファイル内で関数を定義した場合、JScript から呼び出す関数の名前を入力します。JScript 関数は、引数をカスタム アクションから取得できません。 カスタム アクションは最初にスクリプトを実行し、指定されている関数がある場合それを呼び出すので、関数を持っている必要はありません。
スクリプト内実行	選択したカスタム アクションをトリガーするシーケンスの反復を選択します。各オプションの詳細については、「アクションの実行のオプション」を参照してください。 この設定は、すべてのシーケンスに適用するわけではありません。たとえば、**初期化の後 (最初のダイアログの前)** の下にカスタム アクションを追加した場合、そのアクションは即時実行としてスケジュールされます。遅延、ロールバック、コミット実行としてスケジュールすることはできません。

テーブル 3-61 • JScript カスタム アクションの設定 (続き)

設定	説明
アクションの待機	カスタム アクションが終了するまでインストール、アンインストール、またはメンテナンスを待機してから、残りのインストール処理を行うかどうかを指定します。 [いいえ] を選択すると、カスタム アクションを実行中にインストール、アンインストール、またはメンテナンスが続行します。
終了コードを無視する	Windows Installer がこのカスタム アクションの戻り値を無視して、インストール、アンインストール、またはメンテナンスを続行するかどうかを指定します。
Comments	このカスタム アクションに関するコメントを入力します。入力したコメントは参照する目的でのみ使用され、エンドユーザーに表示されることはありません。
条件	この設定を使って、選択したカスタム アクションを実行するためには True 評価が必要な 1 つ以上の条件を指定できます。たとえば、ターゲット システムで特定のオペレーティング システムまたは最小システム要件をチェックする条件を作成できます。実行時に条件が True 評価されると、カスタム アクションが実行します。1 つ以上の条件が False 評価されると、カスタム アクションは実行しません。 1 つ以上の条件を指定するには、この設定で省略記号ボタン (...) をクリックします。 条件を追加すると、InstallShield によって "条件" 設定の下に新しい設定が追加されます。この新しい設定は、追加された条件の条件ステートメントを表示します。 条件を編集または削除するには、"条件" 設定で省略記号ボタン (...) をクリックします。

[セットアップのファイル] ビュー

セットアップまたはサポート ファイルは、アプリケーションのインストール処理の間だけターゲット システムで利用できるファイルです。インストールが始まると、セットアップファイルとサブフォルダーはターゲット システムのテンポラリディレクトリにコピーされ、インストールが完了したときに削除されます。サポートディレクトリ (SUPPORTDIR) はダイナミックファイルの場所を表し、ターゲット システムごとに異なります。また、同じシステムでもインストールするたびに異なります。

[セットアップファイル] ビューはまた Disk1 ノードを含んでいており、ソース メディアのルートにファイルを追加することができます。

[リリースの準備] ビュー

セットアップ プロジェクト作成の最後の手順では、インストールのビルドとテストを行います。InstallShield は、InstallShield 内からインストールをテストできる機能、および様々な選択可能なメディタイプを提供します。

テーブル 3-62 • [リリースの準備] ビューの下にあるビュー

表示	説明
リリース	[リリース] ビューでは、配布メディアの構成/ビルド/テストのほか、それをエンドユーザーに配布することができます。

[リリース] ビュー

リリースをビルドすると、インストール プロジェクトからすべての情報が収集され、サポートされている Windows プラットフォームに製品をインストールすることができる Windows Installer インストール パッケージ (.msi ファイル) にコンパイルされます。

InstallShield でプロジェクトの設計が完了したら、テスト用にリリースをビルドすることができます。そして、最終的には、顧客に配布することができます。[リリース] ビューには、リリースのビルド方法を指定できる設定が含まれています。[リリース] ビューの設定に関する詳しい情報は、次を参照してください。

- Express の設定
- メディアの種類
- [ビルド] タブ
- Setup.exe タブ
- [署名] タブ
- [.NET/J#] タブ
- [インターネット] タブ
- [イベント] タブ

リリースをビルドすると、プロジェクトからすべての情報が収集され、サポートされている Windows プラットフォームに製品をインストールすることができるパッケージにコンパイルされます。

Express の設定

[リリース] ビューの中央ペインにある Express アイテムには、すべてのプロジェクト リリースに適用する次の設定があります。

テーブル 3-63 • Express 設定

設定	説明
パッケージ コード	[一般情報] ビューで入力済みのデフォルト パッケージ コードをオーバーライドするには、新しい GUID を入力します。InstallShield使って、自動的に異なる GUID を作成するには、この設定の **[新しい GUID の生成]** ボタン ([...]) をクリックします。 ***メモ:*** *"パッケージ コードの生成" 設定で [はい] が選択されている場合、このパッケージ コードは無視されます。*
パッケージ コードの生成	リリースがビルドされるたびに InstallShield が新しいパッケージ コードを生成するかどうかを指定します: • **はい** – ビルド時に InstallShield が新しいパッケージ コードを生成して、それを .msi パッケージに含みます。 • **いいえ** – ビルド時に InstallShield は新しいパッケージ コードを生成しません。"パッケージ コード" 設定にパッケージ コードを入力した場合、そのパッケージ コードが使用されます。リリースにパッケージ コードを指定しなかった場合、最初にプロジェクトを作成したときに InstallShield によって生成されたパッケージ コードが使用されます。 パッケージ コードは "概要情報ストリーム" の一部であり、特定のデータベースを識別します。同じパッケージコードを持つ 2 つの .msi データベースの内容は同じでなければなりません。そのため、パッケージ コードはビルドごとに変更しなくてはなりません。
MSI パッケージ ファイル名	InstallShield がビルド時に生成する .msi ファイルに使用するファイル名を指定します (ピリオドまたは拡張子は除きます)。この設定が空白の場合、製品名が使用されます。 このフィールドに Property テーブルからのプロパティの値を含めることもできます。詳細は「セットアップ ファイル名」を参照してください。

テーブル 3-63・Express 設定 (続き)

設定	説明
セットアップ ファイル名	InstallShield がビルド時に生成するセットアップランチャー ファイルに使用するファイル名を指定します(.exe ファイル拡張子は除きます)。この設定が空白の場合、InstallShield はデフォルト値 *Setup* を使用し、セットアップランチャーの名前は Setup.exe となります。 このフィールドに、Property テーブルからのプロパティの値を含めることもできます。たとえば、次の任意のプロパティを入力します: setup[ProductVersion] setup[CustomVersion] setup[ProductCode] setup[ProductCode][ProductVersion] たとえば、**setup[ProductVersion]** と入力した場合、セットアップ名が setup14.10.1234.exe となります。
64 ビット セットアップ ランチャー	64 ビット セットアップ ランチャーと 32 ビット セットアップ ランチャーのどちらをビルドするのかを指定します。 [はい] を選択して、64 ビット セットアップ ランチャーをビルドします。 [いいえ] を選択して、32 ビット セットアップ ランチャーをビルドします。詳細については、「64 ビット セットアップ ランチャーのビルド」を参照してください。

メディアの種類

メディア タイプ、またはリリース タイプは、インストールを配布する際に使用する、ディスクの種類やその他の電子メディアを表します。使用可能なリリースの種類を以下に示しますが、最も一般的なリリースの種類は CD-ROM です。

テーブル 3-64・リリース タイプ

リリースタイプ	説明
CD-ROM	インストールを CD で配布する場合は、このオプションを選択します。メディアのこの種類の最大サイズは、CD 一枚につき 650 MB です。
カスタム	配布しているメディアがメディアの種類の一覧の中にない場合は、このオプションを選択します。たとえば、Zip ディスクでインストールを配布できます。
DVD-5	インストールを 4.38 GB DVD で配布する場合は、このオプションを選択します。
DVD-9	インストールを 7.95 GB DVD で配布する場合は、このオプションを選択します。
DVD-10	インストールを 8.75 GB DVD で配布する場合は、このオプションを選択します。
DVD-18	インストールを 15.83 GB DVD で配布する場合は、このオプションを選択します。

テーブル 3-64・リリース タイプ (続き)

リリースタイプ	説明
SingleImage	インストールをネットワーク上に配置する場合は、このオプションを選択します。このメディアの種類には、サイズの制限はありません。
WebDeployment	Web 上で配布するようにデザインされたインストールを作成しているとき、このオプションを選択することをお勧めします。このメディアの種類には、サイズの制限はありません。WebDeployment セットアップには、特定の設定が関連付けられています。

インストールで複数のディスクを必要とする場合、InstallShield は自動的に必要な数のディスクに分割します。

***注意**・複数ディスクのインストールをビルドする場合は、2 枚目以降のディスクのボリューム ラベルを設定する必要があります。*

[ビルド] タブ

[ビルド] タブでは、リリースのパッケージ方法を構成します。

テーブル 3-65・[ビルド] タブの設定

設定	メディアの種類	説明
リリースの場所	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	リリースのビルドを開始するトップレベルのディレクトリのパスを入力するか、省略記号ボタン (...) をクリックして場所を参照します。
メディア サイズ	**カスタム、CD-ROM、DVD**	この設定は、現在選択されているメディアのサイズを表示します。この設定はカスタムのメディアを除き、すべてのメディアに対して読み取り専用です。10 進法の区切り文字はピリオド (.) です。リリースをビルドすると、この設定で指定された値以下のディスク イメージ フォルダーにファイルが分割されます。 "メディア サイズ単位" 設定は、"メディア サイズ" 設定の値がメガバイト単位とギガバイト単位のどちらであるかを示します。
メディア サイズ単位	**カスタム、CD-ROM、DVD**	この設定は、現在のメディアのユニット サイズを表示します。たとえば、CD はメガバイト単位で、DVD はギガバイト単位で容量が計測されます。この設定の変更は、カスタム メディア タイプを使用する場合のみ可能です。

テーブル 3-65・[ビルド] タブの設定 (続き)

設定	メディアの種類	説明
クラスタ サイズ	**カスタム、CD-ROM、DVD**	クラスタとは、ディスク上の空き領域の最小単位です。このサイズはバイト単位で計測され、各ファイルに対して割り当てられる領域の最小値です。クラスタサイズを 1024 バイトに設定し、ファイルが 2000 バイトの場合、そのファイルには、ディスクで 2 クラスタ必要です。ファイルが 1000 バイトの場合は 1 クラスタのみ必要です。ただし、クラスタにある残りの領域を、他のファイルが使用することはできません。 あらかじめ定義されたメディアタイプのクラスタ サイズは変更できません。カスタム メディア タイプを構成する場合を除き、この設定は読み取り専用です。
圧縮	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	製品のデータ ファイルを圧縮するかどうかを指定します: ・ **圧縮** — InstallShield が製品のすべてのデータ ファイルを .cab ファイルに圧縮します。 ・ **非圧縮** — InstallShield が製品のファイルを .cab ファイルに圧縮しません。 ***メモ:*** *ビルド プロセスの出力は、ビルドするメディアの種類 (CD-ROM、DVD-5、または SingleImage など) や、圧縮方法 (″圧縮″ 設定で指定)、およびセットアップランチャーを含めるかどうか (Setup.exe タブにある ″セットアップランチャー″ 設定で指定) によって異なります。* *たとえば、″圧縮″ 設定で [圧縮] を選択し、″セットアップランチャー″ 設定で [はい] を選択して、SingleImage リリースをビルドした場合、データ ファイルが .cab ファイルに圧縮され、.cab ファイルが Setup.exe ファイルにストリームされます。″圧縮″ 設定に [非圧縮] を選択し、″セットアップランチャー″ 設定に [いいえ] を選択して、CD-ROM リリースをビルドした場合、データ ファイルは圧縮されずに、.msi ファイルを含むフォルダーのサブフォルダーに配置されます。*

テーブル 3-65 • [ビルド] タブの設定 (続き)

設定	メディアの種類	説明
Cab 最適化の種類	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	″圧縮″ 設定で [圧縮] を選択した場合、このリリースの .cab ファイルをビルドするときに使用する圧縮の種類も指定します。選択可能なオプションは以下のとおりです: • LZX – InstallShield は LZX 圧縮を使って、製品のデータ ファイルを .cab ファイルに圧縮します。このオプションを選択すると、最もサイズの小さい .cab ファイルが作成されますが、実行時に .cab ファイルからデータが抽出されるとき、最も時間がかかります。 • MSZIP – InstallShield は MSZIP 圧縮を使って、製品のデータ ファイルを .cab ファイルに圧縮します。デフォルトでは、これが設定されています。 • なし – .cab ファイルが作成されるとき、圧縮は行われません。 ***重要:*** *圧縮を行うと、一般的に圧縮ファイルのサイズが減少します。ただし、ビルドプロセスの完了までの時間が長くなることがあります。圧縮されるファイルのサイズとその数によって、LZX 圧縮とビルドに数時間かかる場合もあります。したがって、LZX オプションを選択した場合、ビルドが完了するまでの所要時間をできるだけ短縮できるよう、ビルド マシンに最新のハードウェアを搭載することが推奨されます。*
Autorun.inf を作成する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	インストールを CD-ROM または DVD-ROM で配布するときに、Autoplay 機能をサポートする場合、[はい] を選択します。インストールを自動再生するための指示が含まれた Autorun.inf というテキスト ファイルがディスク イメージ フォルダーのルートに作成されます。 このファイルを編集して AutoPlay オプションを追加したり、コマンド ライン パラメーターを MsiExec.exe、Setup.exe、または Update.exe に渡したりできます。

テーブル 3-65・[ビルド] タブの設定 (続き)

設定	メディアの種類	説明
未使用のディレクトリを保持する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	このリリースをビルドしたとき、.msi ファイルの Directory テーブルから未使用のディレクトリを削除するかどうかを指定します。選択可能なオプションは以下のとおりです: ・ **いいえ** — Directory テーブルの Directory 列に一覧されているディレクトリが .msi ファイル内の既知の場所で参照されていない場合、InstallShield はビルド時に作成する .msi ファイルの Directory テーブルからそのディレクトリを削除します。これはマージ モジュールがマージされてから削除されますが、.msi ファイルに存在するディレクトリのみが削除の対象となります。したがって、マージ モジュールの Directory テーブルに新しい未使用のディレクトリが含まれている場合、そのディレクトリはインストールに追加されます。 ・ **はい** — InstallShield がビルド時に作成する .msi ファイルの Directory テーブルにあるディレクトリは、そのまま保持されます。 デフォルト値は [いいえ] です。 ***メモ:*** *特定の条件下では、定義済みディレクトリが解決されないためにインストールが失敗する場合があります。Directory テーブルから未使用ディレクトリを削除することで、無駄なエラーを回避することができます。したがって、この設定に [いいえ] を選択することが推奨されます。*

Setup.exe タブ

メモ: *Windows Installer 再配布可能ファイルをインストールに含めるときの情報ついては、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」をご覧ください。*

Setup.exe タブでは、Setup.exe ファイルの設定を構成します。また、Windows Installer 3.1 以前の再配布可能ファイルをインストールに含めるかどうかも指定することができます。Windows Installer 再配布可能ファイルについての詳細は、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。

テーブル 3-66・Setup.exe タブの設定

設定	メディアの種類	説明
セットアップランチャー	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe セットアップ ランチャーを作成するかどうかを指定します。 セットアップランチャーが必要になるシナリオについては、「セットアップランチャーの作成」をご覧ください。

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
必要実行レベル	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	″必要実行レベル″ 設定を使用して、インストールの Setup.exe ファイルが Windows Vista プラットフォーム上でインストール (セットアップ ランチャー、すべての InstallShield 前提条件、および .msi ファイル) を実行するときに必要な最低権限レベルを指定できます。選択可能なオプションは、以下のとおりです。 ・ **管理者** — Setup.exe の実行には、管理者権限が必要です。管理者は、Setup.exe の実行を承認する必要があります。非管理者は、管理者としての認証が必要になります。 ・ **最高権限** — Setup.exe の実行には、管理者権限が推奨されます。管理者は、Setup.exe の実行を承認する必要があります。非管理者は、管理者権限を持たずに Setup.exe を実行します。 ・ **起動者** — Setup.exe の実行に、管理者権限は必要ありません。したがって、管理者権限を持たないユーザーも Setup.exe を実行することができます。Setup.exe は、資格情報または同意 (コンセント) を求める UAC メッセージを表示しません。デフォルトでは、これが設定されています。 InstallShield は Windows アプリケーション マニフェストを Setup.exe ランチャに埋め込みます。このマニフェストは選択された実行レベルを指定します。Windows Vista よりも古いバージョンのオペレーティング システムでは、必要実行レベルは適用されません。 必要実行レベルを昇格することの利点は、Setup.exe を実行するための権限の昇格が、必要な場合 1 度で済むという点です。またこれらの権限はインストールに含まれるすべての前提条件および .msi ファイルにも適用できるため、承認を得るためのプロンプトを複数回にわたって行う必要がありません。たとえば、すべての前提条件のうち 2 つが管理者権限を必要とする場合、この設定を [管理者] に変更します。そうすることにより、インストール中、Windows Installer が Setup.exe ファイルを実行する前、プロンプトはエンドユーザーに対して一度のみ表示されます。ただし、権限を昇格して、インストールの終わりでもアプリケーションを起動する場合、この昇格された権限はアプリケーションに適用されますので注意してください。ほとんどの場合、Windows Vista 以降では、昇格された権限を使用したアプリケーションの実行は推奨されていません。 詳細については、「インストール中におけるユーザー アカウント制御のプロンプトの数を最小化する」を参照してください。

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
MSI エンジンを含める	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Windows Installer 3.1 の再配布可能ファイルを含めるかどうかを指定します。 ***メモ:*** *この設定は、Windows Installer 3.1 の再配布可能ファイルに適用します。様々なバージョンの Windows Installer 再配布可能ファイルをプロジェクトに追加する異なる方法についての詳細は、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。*
MSI 3.1 エンジンの場所	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	***メモ:*** *この設定は、Windows Installer 3.1 の再配布可能ファイルに適用します。様々なバージョンの Windows Installer 再配布可能ファイルをプロジェクトに追加する異なる方法についての詳細は、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。* Windows Installer エンジンのインストーラーの場所を指定します。選択可能なオプションは以下のとおりです: ・ **ソース メディアからコピーする** – 選択した Windows Installer エンジン インストーラーをソース メディアのルート上に残します。このオプションは、すべてのファイルを Setup.exe に圧縮している WebDeployment または SingleImage メディア タイプでは使用できません。 ・ **Setup.exe からエンジンを抽出する** – Windows Installer エンジン インストーラーを Setup.exe に圧縮し、実行時に抽出します。 ・ **Web からエンジンをダウンロードする** – 指定した URL から必要に応じて Windows Installer エンジン インストーラーをダウンロードします。この設定を選択した場合は、必ず URL 設定を適切な Web の場所に設定するか、デフォルト値をそのままにしておきます。
MSI 3.1 エンジン URL	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	この設定は、必要に応じてセットアップランチャーが Windows Installer 3.1 エンジンをダウンロードする場所を指定します。このプロパティは、現在のリリースの "MSI 3.1 エンジンの場所" 設定が [Web からエンジンをダウンロードする] に設定されている場合のみ適用されます。URL にファイル名を指定しないでください。 ***メモ:*** *便宜上、デフォルトの URL のは Revenera が運営する サイトとなっています。3.1 エンジンはこの場所からダウンロードすることができます。*

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
MSI エンジンの再起動を遅延する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	*メモ・この設定は、Windows Installer 3.1 以前の再配布可能ファイルに適用します。様々なバージョンの Windows Installer 再配布可能ファイルをプロジェクトに追加する異なる方法についての詳細は、「Windows Installer 再配布可能ファイルをプロジェクトに追加する」を参照してください。* アプリケーションのインストールが完了するまで、ターゲット システム上での Windows Installer エンジンのインストールまたはアップデートに関連した再起動を遅延するかどうかを指定します。 必要な場合、再起動を遅延するには [はい] を選択します。[いいえ] を選択すると、Windows Installer エンジンがインストールまたは更新された直後 (アプリケーションのインストール前) に、必要に応じてシステムが再起動します。
MSI をローカルにキャッシュ	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	現在のビルドについて、.msi ファイルまたは他のインストール ファイルをターゲット システムにキャッシュするかどうか指定します。 ・ [はい] — .msi ファイルおよびその他のンストール ファイルがターゲット システム上にキャッシュされ、アプリケーションのメンテナンスや修復のときに使用されます。現在のリリースの "キャッシュ パス" 設定は、ファイルをキャッシュする場所を指定します。 ・ [いいえ] — .msi ファイルおよびその他のインストール ファイルはターゲット システム上にキャッシュされません。 *メモ・この設定は、ターゲットシステム上で .msi ファイルが Setup.exe ファイルと同じフォルダーに存在しないリリースに対して有効です。*
キャッシュ パス	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	キャッシュされた .msi ファイルおよび他のキャッシュされたインストール ファイルを保存するエンド ユーザーのシステム上の場所を指定します。**C:¥CachedFiles** などハードコード化された値を入力することもできますが、一覧から選択したインストール先変数を使ってファイルをパスにキャッシュすることをお勧めします。デフォルト値は **[LocalAppDataFolder]ダウンロードされたインストール** です。 *メモ・この設定は、現在のリリースの "MSI をローカルにキャッシュ" 設定が [はい] に設定されている場合にのみ使用されます。*

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
最短初期化時間	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	エンドユーザーがこのリリースを実行した時に、インストールが初期化ダイアログを表示する最短時間 (秒) を指定します。 InstallShield は、この設定で指定した値を Setup.ini ファイルの SplashTime キー名の値として使用します。 インストールの初期化中、デフォルトで、初期化ダイアログが表示されます。スプラッシュ画面が使用されている場合、それもこのときに表示されます。この設定で最短初期化時間を指定すると、初期化ダイアログとスプラッシュ画面が少なくとも指定された秒数の間表示されます。初期化の時間が指定された時間を超えた場合、ダイアログとスプラッシュ画面は、インストールが初期化を完了するまで続けて表示されます。初期化が指定された時間以内に終了した場合、インストールは指定された時間が経過するまでダイアログとスプラッシュ画面を表示し、それからインストールを続行します。
ランチャをパスワードで保護	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	インストールをパスワードで保護するには、[はい] を選択してから ″起動ツールのパスワード″ 設定でパスワードを入力します。 この設定は、″セットアップランチャー″ 設定で [はい] が選択されている場合のみ利用できます。
ランチャ パスワード	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	インストールを保護するパスワードを入力します。パスワードは大文字と小文字を区別します。 この設定は、″起動ツールをパスワードで保護″ 設定で [はい] が選択されている場合のみ利用できます。
カスタム バージョンのプロパティを使用する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe のデフォルトの著作権情報とファイルの説明を独自の著作権情報とファイルの説明でオーバーライドするかどうかを指定します。[はい] を選択した場合、″起動ツールの著作権″ 設定と ″ファイルの説明″ 設定に独自の情報を入力します。 著作権情報と説明が、セットアップランチャーの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンドユーザーが Setup.exe ファイルが右クリックして、[プロパティ] をクリックしたときに表示されます。 詳細については、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
ランチャの著作権	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe のデフォルトの著作権情報を製品の著作権情報でオーバーライドするには、製品の著作権情報を入力します。このとき、"カスタム バージョンのプロパティを使用する" 設定で [はい] も選択してください。 著作権情報が、セットアップランチャーの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンドユーザーが Setup.exe ファイルが右クリックして、[プロパティ] をクリックしたときに表示されます。 詳細については、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。
ファイルの説明	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe のデフォルトのファイルの説明をオーバーライドするには、適切な説明を入力します。このとき、"カスタム バージョンのプロパティを使用する" 設定で [はい] も選択してください。 説明が、セットアップランチャーの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Setup.exe ファイルが右クリックして、[プロパティ] をクリックしたときに表示されます。 詳細については、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。
ファイル バージョン	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe のデフォルトファイル バージョンを独自のファイル バージョンでオーバーライドするには、適切なファイルバージョン番号を入力します。このとき、"カスタム バージョンのプロパティを使用する" 設定で [はい] も選択してください。 ファイル バージョンが、セットアップランチャーの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Setup.exe ファイルが右クリックして、[プロパティ] をクリックしたときに表示されます。 詳細については、「セットアップランチャーのファイルのプロパティをカスタマイズする」を参照してください。

テーブル 3-66・Setup.exe タブの設定 (続き)

設定	メディアの種類	説明
InstallShield 前提条件の場所	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	[再配布可能ファイル] ビューで選択された InstallShield 前提条件がある場所を指定します。選択可能なオプションは以下のとおりです: ・ **個々の選択に従う**—[再配布可能ファイル] ビューで各 InstallShield 前提条件のプロパティに指定した場所を使用します。 ・ **Web からダウンロードする**—プロジェクトに含まれるすべての InstallShield 前提条件ファイルを、必要に応じて各前提条件の InstallShield 前提条件 (.prq) ファイルで指定された URL からダウンロードします。このオプションは、[再配布可能ファイル] ビューで各 InstallShield 前提条件のプロパティに指定した場所をオーバーライドします。 このオプションは、インストールがインターネットからダウンロードされるパッケージサイズとダウンロード時間を最小化したい場合に推奨されます。InstallShield 前提条件は、適切なバージョンが既にターゲット マシンにある場合はダウンロードされません。 ・ **Setup.exe から抽出する** — InstallShield 前提条件ファイルを Setup.exe に圧縮して、必要に応じて実行時に抽出します。このオプションは、[再配布可能ファイル] ビューで各 InstallShield 前提条件のプロパティに指定した場所をオーバーライドします。 このオプションは、インストール全体が Setup.exe に完全に含まれていなければならない場合に選択します。[Web からダウンロードする] オプションを選択すると、インストールが小さくなりダウンロード時間も短縮されますが、[Setup.exe から抽出する] オプションは完全に独立したインストールを提供します。 ・ **[ソース メディアからコピーする]**— InstallShield 前提条件ファイルをソース メディアに格納します。このオプションは、[再配布可能ファイル] ビューで各 InstallShield 前提条件のプロパティに指定した場所をオーバーライドします。 InstallShield 前提条件が別の前提条件の依存ファイルとしてプロジェクトに追加される場合、前提条件依存ファイルの場所は、それを必要とする前提条件の場所設定に従います。 ***ヒント・*** *[Setup.exe から抽出する] オプションまたは [ソース メディアからコピーする] オプションを選択してから、コンピューター上で利用できない InstallShield 前提条件を含むリリースをビルドした場合、前提条件が必要とする各ファイルについて 1 つまたは複数のビルド エラーが生成されます。これらのビルドエラーを回避するには、[再配布可能ファイル] ビューを使ってインターネットからコンピューターへ InstallShield 前提条件のダウンロードを行うか、リリースをビルドする前にプロジェクトから削除します。* 詳細については、「リリース レベルでの InstallShield 前提条件のランタイムの場所を指定する」を参照してください。

[署名] タブ

[署名] タブを使って、InstallShield がファイルに署名するときに使用するデジタル署名に関する情報 (証明機関より付与されたデジタル証明書ファイルを含む) を指定します。[署名] タブでまた、ビルド時にデジタル署名をするインストール内のファイルを指定することもできます。

テーブル 3-67 ・ [署名] タブの設定

設定	メディアの種類	説明
証明書 URL	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	完全修飾 URL を入力します (例、**http://www.mydomain.com**)。この URL は、エンドユーザーが製品や組織、会社に関しての情報を得るためのリンク先として、デジタル署名の中で使用されます。
デジタル証明書情報	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	リリースに署名を行うために使用するデジタル証明書を指定するには、この設定で省略記号ボタン (...) をクリックします。[証明書の選択] ダイアログ ボックスが開いて、.pfx ファイルの場所を指定するか、証明書を含む証明書ストアについての情報を指定することができます。 詳しくは、「[証明書の選択] ダイアログ ボックス」を参照してください。
証明書パスワード	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	使用する .pfx にパスワードがある場合、それを入力します。InstallShield はパスワードを暗号化して、プロジェクト (.ise) ファイルに保存します。 ビルド時に InstallShield はパスワードを使って .pfx でファイルの署名を行います。証明書がパスワードで保護されているのにもかかわらず、この設定に何も入力しなかった場合、.pfx ファイルを使った署名が失敗します。 ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。
出力ファイルに署名する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	署名を行うファイルを指定します。選択可能なオプションは以下のとおりです: ・ なし — インストールに署名を行わない場合、このオプションを選択します。 ・ Setup.exe — Setup.exe ファイルに署名する場合、このオプションを選択します。 ・ [ソース メディアからコピーする] — Setup.exe ファイルと Windows Installer パッケージ (.msi) に署名する場合、このオプションを選択します。 ・ Windows Installer パッケージ—Windows Installer パッケージ (.msi) に署名する場合は、このオプションを選択します。

テーブル 3-67 • [署名] タブの設定 (続き)

設定	メディアの種類	説明
署名の説明	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	″署名出力ファイル″ 設定で指定されたファイルに使用する署名の説明を指定します。ここで指定した説明は、UAC (ユーザー アカウント制御) ボックスの「プログラム名:」ラベルの右側に表示されます。UAC ダイアログ ボックスは、エンド ユーザーが署名されたファイルを起動したとき、昇格された権限が必要な場合に開きます。 この設定を空白のままに残すと、InstallShield は UAC (ユーザー アカウント制御) ボックスの「プログラム名:」ラベルの右側に表示される説明として、ファイル名を拡張子なしで使用します。″パッケージ″ 設定およびそのサブ設定で [ファイルに署名] を選択して、パッケージ内のファイルに署名を行った場合、InstallShield はビルド時に署名されるパッケージ内のファイルの UAC ダイアログ ボックスにこの説明を使用しません。
パッケージ内のファイルに署名する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	リリースに含まれる任意のファイルに署名するかどうかを指定します。 [はい] を選択する場合、″含めるパターンとファイル″ 設定と ″除外するパターンとファイル″ 設定を使って、署名を行うファイルを示します。 ***Windows ロゴ*** *• インストールのすべての実行可能ファイル (.exe、.dll、.ocx、.sys、.cpl、.drv、および .scr ファイル) は、Windows ロゴ プログラムに準拠するためにデジタル署名が必要です。*
既に署名されているファイルに署名する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	プロジェクトにデジタル署名されているファイルが既に存在する場合、InstallShield で、既存のデジタル署名を [署名] タブで指定したデジタル署名で置き換えるかどうかを指定します。これは、″含めるパターンとファイル″ 設定と ″除外するパターンとファイル″ 設定で指定された要件を満たすファイルにのみ適用されますので注意してください。 • 既にファイルに含められている既存のデジタル署名情報の代わりに、[署名] タブで指定するデジタル署名情報を使ってファイルに署名をする場合は、[はい] を選択します。 • 既に署名されているファイルについて、既存のデジタル署名情報をそのまま残す場合は、[いいえ] を選択します。 デフォルト値は [いいえ] です。

テーブル 3-67 • [署名] タブの設定 (続き)

設定	メディアの種類	説明
元の場所にあるファイルに署名する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	InstallShield で署名をするとき、元のファイルにも署名をするか、またはリリースに組み込まれるファイルのみ署名するかを指定します。 • 各ファイルの一時コピーに署名して、その署名された一時コピーを使ってリリースをビルドする場合、[いいえ] を選択します。ここで [いいえ] を選択すると、元のファイルは変更も署名もされませんので注意してください。 • InstallShield で元のファイルに署名する場合、[はい] を選択します。 デフォルト値は [いいえ] です。
含めるパターンとファイル	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ビルド時にデジタル署名するファイルおよびファイル パターンを指定するには、以下の手順の 1 つを行います。 • たとえば *.dll のようなファイル パターンだけでなく、現在プロジェクトに含まれているすべてのスタティック ファイルの一覧から、1 つ以上のファイル名またはファイル パターンを選択するには、この設定で省略記号ボタン (...) をクリックします。[ファイルを参照] ダイアログ ボックスが開き、1 つ以上のパターンおよびファイルを選択できます。アイテムの選択が完了すると、"次のパターンとファイルを含める" 設定の下に 1 つ以上の新しい "含める" 設定が追加されます。 • ファイル名またはパターンを手入力するには、この設定の [追加] ボタンをクリックします。InstallShield が "次のパターンとファイルを含める" 設定の下に新しい "含める" 設定を追加します。この新しい設定を使って、ファイル名またはパターンを指定できます。

テーブル 3-67 • [署名] タブの設定 (続き)

設定	メディアの種類	説明
含める	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ビルド時にデジタル署名するファイルまたはファイル パターンを指定します。次の事項に注意してください。 • ワイルドカード文字の指定には、アスタリスク (*) を使用します。 たとえば、すべての .exe ファイルに署名する場合、*.exe のように指定します。**.exe** ワイルドカード文字の使用は、プロジェクトに動的にリンクされたファイル含め、特定のパターンに一致するすべてのファイルに署名を行う場合、特に便利です。 • 署名しないファイルとファイル パターンは、すべての署名するファイルとファイル パターンをオーバーライドしますので注意してください。たとえば、"含める" 設定および "除外する" 設定に ***.exe** を指定すると、InstallShield は .exe ファイルに署名を行いません。 ファイルまたはファイル パターンを削除するには、この設定で [削除] ボタンをクリックします。 別のファイルまたはファイル パターンを追加するには、"含めるパターンとファイル" 設定を使います。
除外するパターンとファイル	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ビルド時にデジタル署名しないファイルおよびファイル パターンを指定するには、以下の手順の 1 つを行います。 • たとえば *.dll のようなファイル パターンだけでなく、現在プロジェクトに含まれているすべてのスタティック ファイルの一覧から、1 つ以上のファイル名またはファイル パターンを選択するには、この設定で省略記号ボタン (...) をクリックします。[ファイルを参照] ダイアログ ボックスが開き、1 つ以上のパターンおよびファイルを選択できます。アイテムの選択が完了すると、"次のパターンとファイルを除外する" 設定の下に 1 つ以上の新しい "除外する" 設定が追加されます。 • ファイル名またはパターンを手入力するには、この設定の [追加] ボタンをクリックします。InstallShield が "除外するパターンとファイル" 設定の下に新しい "除外する" 設定を追加します。この新しい設定を使って、ファイル名またはパターンを指定できます。

テーブル 3-67 • [署名] タブの設定 (続き)

設定	メディアの種類	説明
除外する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ビルド時にデジタル署名しないファイルまたはファイル パターンを指定します。次の事項に注意してください。 • ワイルドカード文字の指定には、アスタリスク (*) を使用します。 たとえば、.drv ファイルはどれも署名しない場合 ***.drv** のように指定します。 ワイルドカード文字の使用は、プロジェクトに動的にリンクされたファイルが含まれていて、特定のパターンに一致するすべてのファイルに署名を行わない場合、特に便利です。 • 署名しないファイルとファイル パターンは、すべての署名するファイルとファイル パターンをオーバーライドしますので注意してください。たとえば、"含める" 設定および "除外する" 設定に ***.exe** を指定すると、InstallShield は .exe ファイルに署名を行いません。 ファイルまたはファイル パターンを削除するには、この設定で [削除] ボタンをクリックします。 別のファイルまたはファイル パターンを追加するには、"除外するパターンとファイル" 設定を使います。

[.NET/J#] タブ

.NET/J# タブで、.NET Framework 1.0、1.1、または 2.0 の 32 ビット バージョンのサポートを追加することができます。また、プロジェクトに J# のサポートも追加することができます。

メモ• *プロジェクトに別のバージョンの .NET Framework 再配布可能ファイルを含める場合、[再配布可能ファイル] ビューを使って、Microsoft .NET Framework 用の適切な InstallShield 前提条件をプロジェクトに追加します。*

詳細については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。

テーブル 3-68・.NET/J# タブの設定

設定	メディアの種類	説明
.NET Framework の場所	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	.NET Framework ランタイムのある場所を指定します。.NET Framework は、.NET 機能を使用するアプリケーションには必須です。有効なオプションは次のとおりです: ・ **ソース メディアからコピーする** – .NET Framework ランタイムがソース メディアのルートに残されます。このオプションは、すべてのファイルを Setup.exe に圧縮している WebDeployment または SingleImage メディア タイプには適用できません。 ・ **Setup.exe から抽出** – .NET Framework ランタイムが Setup.exe に圧縮され、実行時に抽出されます。 ・ **Web からダウンロード** – 指定の URL から (必要に応じて) .NET Framework のランタイムがダウンロードされます。このオプションを選択した場合、"NET と J# Framework URL" 設定の値を適切な Web の場所に設定する必要があります。 ・ **含めない**–選択したリリースに、.NET Framework 1.0、1.1、または 2.0 の再配布可能ファイルは含まれません。
.NET Framework バージョン	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	インストールと共に配布する .NET Framework のバージョンを選択します。 ***メモ:*** *プロジェクトに .NET Framework 3.5、3.0 SP1、3.0、2.0 SP1 再配布可能ファイルを含めるには、[再配布可能ファイル] ビューで Microsoft .NET Framework 用の適切な InstallShield 前提条件をプロジェクトに追加します。* *詳細については、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」を参照してください。*
.NET 1.1/2.0 コア言語	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	"NET Framework バージョン" 設定で .NET バージョン 1.1 を選択した場合、配布する .NET コア言語を 1 つ指定できます。これは、.NET 1.1 コア再配布可能ファイルのインストール中に使用される言語です。バージョン 2.0 を選択すると、すべての言語オプションは、このバージョンの再配布可能ファイルに含まれているので、選択され無効にされます。 コア言語を変更するには、この設定の省略記号ボタン (...) をクリックします。

テーブル 3-68 • .NET/J# タブの設定 (続き)

設定	メディアの種類	説明
Dotnetfx.exe へ渡すコマンドライン	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	*メモ• この設定は、"NET Framework バージョン" 設定で .NET 1.1 が選択されている場合のみ適用されます。* マイクロソフトの DotNetFx.exe ファイルに渡すコマンド ラインを入力します。
.NET 1.1/2.0 言語パック	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	.NET 言語パックを含めるには、この設定の省略記号ボタン (...) をクリックします。ビルド マシンにインストールしたマイクロソフト言語パックに応じて利用できるオプションが異なります。省略記号ボタンをクリックすると、追加で使用可能な言語パックをダウンロードできます。
言語パックへ渡すコマンドライン	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	*メモ• この設定は、"NET Framework バージョン" 設定で .NET 1.1 が選択されている場合のみ適用されます。* インストールに含まれるすべての Microsoft LangPack.exe ファイルに送るコマンド ラインを入力します。
[.NET のオプション] ダイアログを表示する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	Setup.exe で、ターゲット システムに .NET Framework をインストールするかどうかをエンドユーザーに確認する [はい/いいえ] メッセージ ボックスを表示するかどうかを示します。この設定はインストールに .NET Framework を含めるかどうかを決定するものではありません。エンドユーザーにインストールする選択肢を与えるかどうかを指定するだけです。 • [いいえ] – 必要な場合、エンドユーザーにメッセージ ボックスを表示せずに .NET Framework をインストールします。 • [はい] – .NET オプション ダイアログ ボックスを表示し、エンドユーザーが .NET Framework をインストールするかどうかを指定することができます。 [はい] を選択すると、メッセージ ボックスが実行時にターゲット システムで表示されます。メッセージ ボックスは、インストールはオプションで Microsoft .NET Framework を使用することを通知し、インストールするかどうかをたずねます。

テーブル 3-68・.NET/J# タブの設定 (続き)

設定	メディアの種類	説明
.NET Frameworkのインストール時に完全なユーザー インターフェイスを表示する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	NET Framework インストール用の完全インターフェイスを表示するかどうかを指定します。 [はい] を選択すると、dotnetfx.exe がターゲット コンピューターに .NET Framework をインストールするとき、Microsoft .NET Framework (英語) セットアップ ウィザードが表示されます。ウィザードは .NET Framework インストールの進行状況を表示します。 [いいえ] を選択すると、dotnetfx.exe がターゲット コンピューターに .NET Framework をインストールするとき、InstallShield Wizard が表示されます。InstallShield Wizard は .NET Framework インストールの進行状況を表示します。
.NET ビルド構成	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	この設定には、.NET ソリューションのビルド構成の名前が含まれます。InstallShield はこの設定を使用して、プロジェクト出力ファイル (デバッグやリリースなど) の場所を決定しますC# および VB.NET プロジェクト ウィザードは、新しいプロジェクト作成時やプロジェクト ソリューション変更時に自動的にこのフィールドを作成します。
.NET と J# Framework URL	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	インストールが .NET Framework ランタイムと J# 再配布可能ファイル (含まれている場合) をダウンロードする場所を指定します。この URL では、ファイル名を指定する必要はありません。 この設定は、現在のリリースについて、".NET Framework の場所" 設定または "J# 再配布可能ファイルの場所" 設定 (含まれている場合) が [Web からダウンロードする] に設定されている場合のみ必要です。 まず、インストールが、InstallShield ファイル Dotnetfx.exe をダウンロードして実行します。次いで、マイクロソフト Web サイトから Dotnetredist.exe をダウンロードします。この動作は Dotnetfx.exe にハードコード化されているため、ホストする場所を問いません。

テーブル 3-68・.NET/J# タブの設定 (続き)

設定	メディアの種類	説明
J# 再配布可能ファイルの場所	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	J# 再配布可能ファイルの場所を指定します。.NET Framework は、.NET 機能を使用するアプリケーションに必須です。選択可能なオプションは以下のとおりです: ・ **ソース メディアからコピーする** – J# 再配布可能ファイルをソース メディアのルート上に残します。このオプションは、すべてのファイルを Setup.exe に圧縮している WebDeployment または SingleImage メディア タイプには適用できません。 ・ **Setup.exe からを抽出する** – J# 再配布可能ファイルを Setup.exe に圧縮して実行時に抽出します。 ・ **Web からダウンロード** – ″.NET および J# Framework URL″ 設定で指定した URL から (必要に応じて) J# 再配布可能ファイルをダウンロードします。 ・ **含めない** – 選択したリリースに J# 再配布可能ファイルを含めません。 ***メモ:*** *使用される J# バージョン番号は、.NET Framework に選択したバージョン番号と一致します。これらの再配布可能ファイルから J# 1.1、および .NET 2.0 をインストールすることはできません。*
J# 再配布可能ファイルへ渡すコマンドライン	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	vjredist.exe に渡すコマンド ライン パラメーターを指定します。有効なコマンド ライン パラメーターについては Microsoft サポートに問い合わせてください。
J# オプションダイアログを表示する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ターゲット システムに J# をインストールするかどうかをエンドユーザが指定できるダイアログをインストールで表示するかどうを指定します。 ***メモ:*** *.NET Framework 1.1 もインストールに含まれている場合、ダイアログは、.NET Framework 1.1 もインストールされることを通知します。*
インストールをサイレントで実行する場合、J# をインストール	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	(インストールがサイレントで実行される場合など) J# オプション ダイアログが表示されない場合にターゲット システムに J# をインストールするかどうかを指定します。

[インターネット] タブ

[インターネット] タブでは、WebDeployment タイプのリリースについての Web に関連する情報を指定します。このタブは他のリリース タイプでは無効にされています。

テーブル 3-69 ・ [インターネット] タブの設定

設定	説明
One-Click Install の生成	One-Click Install を作成するかどうかを指定します。One-Click Install は、最初のユーザーインターフェイスが HTML ページのインストール プログラムです。エンド ユーザーが Web ページにアクセスして [インストール] ボタンをクリックすると、インストール ファイルがターゲット システムにダウンロードされ、起動します。 この設定を [はい] に設定した場合、"One-Click HTML ベース名" 設定と "One-Click .cab ベース名" 設定にファイル名を指定する必要があります。
One-Click HTML ベース名	InstallShield がビルド時に生成する HTML ファイルのベース ファイル名を指定します。指定したベース名の末尾には .htm ファイル名拡張子が付けられ、生成されたファイルは現在のリリースの場所の Disk1 フォルダーに作成されます。 この設定は、"One-Click Install の生成" 設定の値が [はい] の場合にのみ有効です。
One-Click .cab ベース名	ビルド プロセスによって One-Click Install インストールに生成されるキャビネット ファイル (.cab) のベース ファイル名を指定します。 ここで指定された名前に .cab ファイル名拡張子が追加されます。生成されたファイルは、現在のリリースの Disk1 フォルダーに作成されます。 この設定で指定した名前は、インストールにデジタル署名を行なった場合、実行時にデジタル証明書に表示されます。 この設定は、"One-Click Install の生成" 設定の値が [はい] の場合にのみ有効です。

[イベント] タブ

[イベント] タブでは、ビルド時またはオンデマンドでリリースをフォルダーまたは FTP サイトに自動的に配布できるようにする設定を構成することができます。

エディション・ *InstallShield Premier Ediion の [イベント] タブには、ビルド プロセスの様々な段階で実行するコマンドを指定できる追加の設定が提供されています。たとえば、製品のデータ ファイルを .cab ファイルに格納する*

場合、InstallShield が .msi パッケージと .cab ファイルをビルドした後に実行するコマンドを指定できます。このイベントは、.msi パッケージにデジタル署名が行われ、Setup.exe ファイルにストリームされる前に発生します。

テーブル 3-70 • [イベント] タブの設定

設定	メディアの種類	説明
フォルダーにコピー	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	リリースを自動的にフォルダーへ配布する場合、その場所を指定します。コピーされたフォルダーと同じ名前の既存のフォルダーは上書きされますが、フォルダーは削除されません。この設定でパスを入力するか、省略記号ボタン (...) をクリックして場所を参照して、フォルダーのパスを指定します。 ***ヒント•*** *[リリース] ビューでリリースを右クリックして [配布] をクリックしたとき、そのつど指定されたフォルダーにリリースに関連するすべてのファイルがコピーされます。* *ビルドを行う度にビルド エンジンが指定されたフォルダーにリリースをコピーするように設定するには、"ビルド後、配布する" 設定を [はい] に選択します。* ***メモ•*** *この設定でフォルダーを指定し、[イベント] タブで FTP ロケーションも指定した場合、リリースは FTP ロケーションにのみコピーされます。*
FTP ロケーション	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	リリースを特定の FTP サーバーに自動的に配布できるようにするには、その場所の FTP URL を指定します。プロジェクトの各リリースに、異なる FTP ロケーションを指定することができます。 ***メモ•*** *FTP デフォルト フォルダー以外のパスにリリースを配布する場合は、ダブル スラッシュ (//) を使います。たとえば、リリースを、FTP の URL が* ftp://ftp.mydomain.com *であるW* myproduct *という名前のルートレベル フォルダーへ配布する場合、FTP の場所に* ftp://ftp.mydomain.com//myproduct *と入力します。* ***ヒント•*** *[リリース] ビューでリリースを右クリックして [配布] をクリックしたとき、そのつど指定された FTP ロケーションにリリースに関連するすべてのファイルがコピーされます。* *ビルドを行う度にビルド エンジンが指定された場所にリリースをコピーするように設定するには、"ビルド後、配布する" 設定を [はい] に選択します。*

テーブル 3-70 • [イベント] タブの設定 (続き)

設定	メディアの種類	説明
FTP サイト ユーザー名	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	指定する FTP ロケーションへのアップロードにユーザー名が必要な場合、ユーザー名を入力します。 ***メモ:*** *プロジェクト内のリリースの 1 つにユーザー名を入力すると、そのユーザー名が同じプロジェクトと他の Express プロジェクトの他のリリースにも使用されます。*
FTP サイト パスワード	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	指定する FTP ロケーションへのアップロードにパスワードが必要な場合、パスワードを入力します。 ***メモ:*** *プロジェクト内のリリースの 1 つにパスワードを入力すると、そのパスワードが同じプロジェクトと他の Express プロジェクトの他のリリースにも使用されます。*
ビルドの後、配布する	カスタム、CD-ROM、DVD、SingleImage、WebDeployment	ビルドする度に、ビルド エンジンが自動的に指定した場所へリリースを配布するようにするかどうかを指定します。 ***ヒント:*** *ビルドの実行ごとではなく、オンデマンドでリリースを素早く配布するには、リリースを右クリックして、[配布] をクリックします。[配布] コマンドは、リリースが少なくとも 1 度ビルドされている場合のみ利用することができます。*

QuickPatch プロジェクト

QuickPatch プロジェクトは、規模の小さい単一のアップグレードをエンド ユーザーに配布したいインストール作成者へお勧めする特殊な Windows Installer ベースのプロジェクトです。QuickPatch プロジェクトごとに、独自のビューがあります。

[パッチ設定の定義] ビュー

[パッチ設定の定義] ビューは、QuickPatch プロジェクトを作成または開いたときに利用できます。QuickPatch プロジェクトを開く、または作成した後、[パッチ設定の定義] ビューで基本的な設定の一部を構成することができます。このビューを利用して、QuickPatch を構成します。

次のテーブルは、[パッチ設定の定義] ビューの下にあるビューについて説明します。

テーブル 3-71 ・ [パッチ設定の定義] ビューの下にあるビュー

表示	説明
一般情報	製品およびオリジナル インストールについての情報を入力します。現在のプロジェクトがパッチするリリースを指定し、プロジェクトに含めるカスタム アクションを選択します。
ファイル	パッチにファイルを追加して、ファイル パッチ オプションを指定します。
レジストリ	どのレジストリ値が追加、変更、または削除されたかを調べます。

[一般情報] ビュー

[一般情報] ビューには、QuickPatch プロジェクトに関する基本的な情報が含まれています。このビューでは製品プロパティ、ビルド設定、パッチ履歴、およびカスタム アクションを表示し、構成することもできます。

製品のプロパティ

[一般情報] ビューで [製品のプロパティ] をクリックすると、以下が表示されます。

テーブル 3-72 ・ 製品のプロパティの設定

設定	説明
元のセットアップ パス	ここでは、製品名とオリジナル インストール イメージへのパスが表示されます。パス ボックスは読み取り専用です。
製品バージョン	[オリジナル セットアップのバージョン] ボックスには、オリジナル インストールの製品バージョンが表示されます。このボックスは読み取り専用です。[QuickPatch バージョン] ボックスで、QuickPatch プロジェクトの製品バージョンを変更することができます。このボックスへの入力はオプションです。
FlexNet Connect の統合	FlexNet Connect がオリジナル インストールで有効にされていて、FlexNet Connect が製品を識別するために使用するカスタム バージョン番号を指定する場合 (Windows Installer が使用する標準バージョン スキームを使わずにという意味で)、このボックスで作成中の QuickPatch に新しいカスタム バージョン番号を入力します。

ビルドの設定

[一般情報] ビューで [ビルドの設定] をクリックすると、以下のタブが表示されます。

- [共通] タブ
- [識別] タブ

- [デジタル署名] タブ
- [詳細] タブ

[共通] タブ

[一般情報] ビューで [ビルドの設定] をクリックすると、いくつかのタブが表示されます。[共通] タブには、QuickPatch のためのよく利用されるビルドの設定があります。

ビルドの場所

パッチ ファイルをビルドする場所を指定、または既存フォルダーを検索します。

ランチャの設定

ここでは、次のようなランチャー設定を構成することができます。

テーブル 3-73・構成可能なランチャー設定

設定	説明
Update.exe の作成	現在の QuickPatch パッケージに Update.exe アップデート ランチャを作成する場合、このチェック ボックスを選択します。 Update.exe アップデート ランチャーが必要になる場合についての詳細は、「パッチ時の考慮事項」を参照してください。
Windows Installer 3.1 エンジンを含める	このチェック ボックスを選択して Windows Installer 3.1 エンジン をパッチパッケージに含めます。
.NET Framework を含める	このチェック ボックスを選択して、.NET Framework をパッチパッケージに含めます。

パッチのアンインストール

アプリケーション全体と他の QuickPatch パッケージをアンインストールして再度インストールし直す手間を掛けずにパッチをアンインストールできるようにする場合、**[パッチのアンインストールを許可する (Windows Installer 3.0 が必要)]** チェック ボックスを選択します。QuickPatch パッケージのアンインストールは一定の条件下のみで動作しますのでご注意ください。たとえば、バージョン 3.0 よりも前の Windows Installer はアプリケーションから QuickPatch のみを削除することができません。さらに詳しい情報は、Windows Installerヘルプの「パッチの削除」を参照してください。

[識別] タブ

QuickPatch プロジェクトの [一般情報] ビューで [ビルドの設定] をクリックすると、いくつかのタブが表示されます。[識別] タブには、表示文字列の設定があります。表示文字列は、[プログラムの追加と削除] で表示されるパッチに関する情報に使用されます。メタデータは、ターゲットマシンの適用されたパッチを検索しカタログする Windows Installer 3.0 以降の API によっても使用されます。

メモ: パッチのメタデータはパッチ (.msp) ファイルに直接格納され、.msi パッケージには格納されません。.msp ファイルのコンテンツはローカライズされません。したがって、文字列テーブルのエントリは、以下のメタデータの設定には使用することができません。

テーブル 3-74 ・ パッチのアンインストールの設定

設定	説明
表示名	パッチの名前を指定します。
サポート URL	エンドユーザーにテクニカルサポートを提供する URL を指定します。
説明	パッチの簡単な説明を指定します。
製造元の名前	アプリケーションの製品メーカーの名前を指定します。
ターゲット製品名	アプリケーションまたはターゲット アプリケーション スイートの名前を指定します。
分類	アップグレードのカテゴリーを指定します。たとえば、クリティカルアップデート、ホットフィックス、および、サービスパック等。

[デジタル署名] タブ

QuickPatch プロジェクトの [一般情報] ビューで [ビルドの設定] をクリックすると、いくつかのタブが表示されます。[デジタル署名] タブで、パッチにデジタル署名をするときの設定を指定することができます。

テーブル 3-75 ・ [デジタル署名] タブの設定

設定	説明
パッチパッケージに署名する	QuickPatch パッケージにデジタル署名を行う場合、このチェック ボックスを選択します。
Update.exe に署名する	Update.exe パッケージにデジタル署名を行う場合、このチェック ボックスを選択します。
証明書 URL	完全修飾 URL を入力します (例、**http://www.mydomain.com**)。この URL は、エンドユーザーが製品や組織、会社に関しての情報を得るためのリンク先として、デジタル署名の中で使用されます。
デジタル証明書情報	リリースに署名を行うために使用するデジタル証明書を指定するには、この設定の横にある [参照] ボタンをクリックします。[証明書の選択] ダイアログ ボックスが開いて、.pfx ファイルの場所を指定するか、証明書を含む証明書ストアについての情報を指定することができます。 詳細については、「[証明書の選択] ダイアログ ボックス」を参照してください。

テーブル 3-75 • [デジタル署名] タブの設定 (続き)

設定	説明
Password	使用する .pfx にパスワードがある場合、それを入力します。InstallShield はパスワードを暗号化して、プロジェクト (.ise) ファイルに保存します。 ビルド時に InstallShield はパスワードを使って .pfx でファイルの署名を行います。証明書がパスワードで保護されているのにもかかわらず、この設定に何も入力しなかった場合、.pfx ファイルを使った署名が失敗します。 ストアにパスワード付きでインポートされた証明書をプロジェクトで使用するように構成すると、ビルド時、InstallShield がプロジェクトのファイルに署名を行うときに、Windows がパスワードをプロンプトします。Windows が使用する強力なキー保護のため、InstallShield がパスワードを暗号化サービス プロバイダーに提供することはできません。
署名の説明	(必要な場合) パッチ パッケージおよび Update.exe ファイルに使用する署名の説明を指定します。ここで指定した説明は、UAC (ユーザー アカウント制御) ボックスの「プログラム名:」ラベルの右側に表示されます。UAC ダイアログ ボックスは、エンド ユーザーが署名されたファイルを起動したとき、昇格された権限が必要な場合に開きます。 この設定を空白のままに残すと、InstallShield は UAC (ユーザー アカウント制御) ボックスの「プログラム名:」ラベルの右側に表示される説明として、ファイル名を拡張子なしで使用します。

[詳細] タブ

[一般情報] ビューで [ビルドの設定] をクリックすると、いくつかのタブが表示されます。[詳細] タブには、QuickPatch 用に構成することができるビルドの設定がすべてあります。

ビルドの場所

[ビルドの保存場所] 領域では、次の設定を行います。

テーブル 3-76 • ビルドの場所に関する設定

プロパティ	説明
ビルドの場所	パッチ ファイルをビルドする場所を指定、または既存フォルダーを検索します。
Update.exe の作成	現在の QuickPatch パッケージに Update.exe アップデート ランチャを作成するかどうかを指定します。 Update.exe アップデート ランチャーが必要になる場合についての詳細は、「パッチ時の考慮事項」を参照してください。

テーブル 3-76・ビルドの場所に関する設定 (続き)

プロパティ	説明
置換するパッチ GUID のリスト	現在の QuickPatch で、1 つまたは複数の以前のインストール済みのパッチを置換するには、このプロパティにそれらのパッチのパッチ GUID をカンマで区切って設定します。例: {C86838C9-DEDC-4451-B96F-94AFB9460F15},{C8633E5B-AC44-45d8-B487-C68B3B1F60D6} [履歴] ビューに QuickPatch プロジェクトが複数あっても、このプロパティの設定は通常、*必要ありません*。ただし QuickPatch プロジェクトが以前の QuickPatch プロジェクトに追加されたファイルを上書きしない場合、このプロパティの設定が必要な場合もあります。 置換するパッチの GUID が分からない場合、このプロパティをクリックしてから、省略記号 (...) ボタンをクリックして参照します。パッチ (.msp または .exe ファイル) を選択すると、対応する GUID がこのプロパティに InstallShield によって追加されます。
新規 UpdatedImage フォルダーを作成	この設定を利用して、既存の UpdatedImage フォルダーと共に QuickPatch パッケージをビルドすることが可能になります。このオプションを利用するには、このパッケージを少なくとも 1 度ビルドしている必要があります (UpdatedImage フォルダーが必要な為) 。このオプションを [いいえ] に設定すると、既存の UpdatedImage フォルダーからパッケージをビルドします。この方法で UpdatedImage フォルダーにある MSI パッケージを微調整して、その .msi データを QuickPatch プロジェクトで利用することができます。このオプションを [はい] (デフォルト値) にすると、パッケージをビルドする度に UpdatedImage フォルダーに .msi ファイルが再生成されます。
MsiPatchOldAssembly テーブルの生成	MsiPatchOldAssemblyFile および MsiPatchOldAssemblyName テーブルのエントリを自動的に生成するかどうかを指定します。これらのエントリを利用すると Windows Installer 3.0 以降で実行されるパッチ パッケージが元のインストール ソースを実行時に要求することなく、GAC (グローバル アセンブリ キャッシュ) のアセンブリをパッチすることができます。詳細については、「グローバル アセンブリ キャッシュのアセンブリをパッチする」を参照してください。
パッチ シーケンスのエントリ作成	QuickPatch にパッチ シーケンシングを使用するかどうかを指定します。パッチ シーケンスを参照して、もう使われていないパッチ、入れ替わったパッチ、または製品に既に適用済みのパッチを見ることができます。シーケンスは、パッチがターゲットマシンに提供された順番に関係なく、どの順番で Windows Installer バージョン 3.0 以上がインストールされた製品にパッチを適用するかを指定します。バージョン 3.0 以前の Windows Installer の場合、パッチ シーケンスは無視され、すべてのパッチは、ターゲットマシンに提供された順番で製品に適用されます。

テーブル 3-76・ビルドの場所に関する設定 (続き)

プロパティ	説明
QuickPatch の簡素化	QuickPatch パッケージの簡素化を行い、最もシンプルなパッケージをビルドするかどうかを指定します。デフォルトの値は [はい] です。 QuickPatch の簡素化を行うと、通常の QuickPatch パッケージに比べて新しいサブ機能とカスタム アクションの数が少ない QuickPatch パッケージを生成できます。 たとえば、QuickPatch プロジェクトに新しいファイルとレジストリ エントリが含まれていて、 が QuickPatch の簡素化を行わなかった場合、そのファイルとレジストリ エントリ用に新しいサブ機能が作成されます。さらに、特定の Windows Installer パッチ要件に対応するため、1 つまたは複数のビルド済み カスタム アクションが追加されます。これに対し、InstallShield が QuickPatch の簡素化を行う場合、ファイルまたはレジストリ エントリは既存する機能に追加されるため、特別なビルド済み InstallShield カスタム アクションは必要ありません。 ***メモ:*** *次のシナリオにおいて、InstallShield が QuickPatch パッケージの作成処理を簡素化することはできません。* - *QuickPatch パッケージがインストール済みのファイルを削除する。* - *QuickPatch パッケージがレジストリ キーを削除する、またはその名前を変更する。* - *QuickPatch パッケージが、簡素化されていない通常の QuickPatch イメージをターゲットとする。つまり、[一般情報] ビューの [履歴] 領域で、簡素化を行わなかった QuickPatch のチェック ボックスを選択した場合、QuickPatch の簡素化はできないということです。1 つまたは複数の簡素化されていない QuickPatch イメージをターゲットとする簡素化された QuickPatch のビルドを試みると、ビルド警告が表示され、簡素化は行われません。*
ランチャをパスワードで保護	QuickPatch パッチをパスワードで保護するには、[はい] を選択してから "ランチャ パスワード" 設定でパスワードを入力します。QuickPatch パッチをパスワードで保護すると、QuickPatch パッチを適用するエンド ユーザーは、アップデートを起動するために大文字小文字の区別があるパスワードを入力する必要があります。 この設定は Update.exe ファイルを使用する QuickPatch パッチのみに適用します。
ランチャ パスワード	アプリケーションを保護するパスワードを入力します。パスワード保護をアクティブにするには、"ランチャをパスワードで保護" 設定で [はい] を選択する必要があります。 パッチをパスワードで保護すると、パッチを適用するエンド ユーザーは、アップデートを起動するために大文字小文字の区別があるパスワードを入力する必要があります。 この設定は Update.exe ファイルを使用するパッチのみに適用します。

Windows Installer エンジン

ここでは、Windows Installer エンジンのための次のような設定を構成することができます。

テーブル 3-77 • Windows Installer エンジンの設定

プロパティ	説明
Windows Installer 3.1 エンジンを含める	Windows Installer エンジンをパッチパッケージに含めるかどうか指定します。
エンジンの場所	次のオプションから選択してください。 • **Web からエンジンをダウンロードする** – Windows Installer エンジンのインストールが必要な場合、実行時にダウンロードされます。 • **Update.exe からエンジンを抽出する** – ビルドは Windows Installer エンジンを Update.exe ファイルにストリームし、顧客に配布する単一ファイルを生成します。エンジンは、実行時に Update.exe ファイルから抽出され必要に応じてインストールされます。 • **ソース メディアからコピーする**–ビルドは、Windows Installer エンジンを Update.exe ファイルと同じディレクトリにコピーします。
Windows Installer 3.1 エンジン URL	エンジン場所の URL を指定します。Update.exe ファイルは、この場所を、実行時に、エンジンをダウンロードする場所として使います。便宜上、デフォルトの URL の場所は Revenera が運営するライブ サイトとなっています。

Microsoft .NET Framework

ここでは、Microsoft .NET Framework エンジンのための次のような設定を構成することができます。

テーブル 3-78 • Microsoft .NET Framework の設定

プロパティ	説明
ビルドに含める	パッチに Microsoft .NET Framework を含めるかどうかを指定します。
エンジンの場所	次のオプションから選択してください。 • **Web からエンジンをダウンロードする** – .NET Framework エンジンのインストールが必要な場合、実行時にダウンロードされます。 • **Update.exe からエンジンを抽出する**–ビルドは.NET Framework を Update.exe ファイルにストリームし、顧客に配布するシングルファイルを生成します。エンジンは、実行時に Update.exe ファイルから抽出され必要に応じてインストールされます。 • **ソース メディアからコピーする** – ビルドは、.NET Framework を Update.exe ファイルと同じディレクトリにコピーします。

テーブル 3-78 • Microsoft .NET Framework の設定 (続き)

プロパティ	説明
エンジン URL	.NET Framework の場所の URL を指定します。Update.exe ファイルは、この場所を、実行時に、.NET Framework をダウンロードする場所として使います。便宜上、デフォルトの URL の場所は Revenera が運営するライブ サイトとなっています。

アップデート起動ツールの設定

この領域では、Update.exe 起動ツールにおける次の設定を構成することができます:

テーブル 3-79 • アップデート起動ツールの設定

プロパティ	説明
会社名	Update.exe のデフォルトの会社名を独自の会社名でオーバーライドするには、ここに会社名を入力します。 会社名が、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックすると表示されます。 詳細については、「アップデート起動ツールのファイルのプロパティをカスタマイズする」を参照してください。
製品名	Update.exe のデフォルトの製品名を独自の製品名でオーバーライドするには、ここに製品名を入力します。 製品名が、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックすると表示されます。 詳細については、「アップデート起動ツールのファイルのプロパティをカスタマイズする」を参照してください。
製品バージョン	Update.exe のデフォルトの製品バージョンを独自の製品バージョンでオーバーライドするには、ここに製品バージョンを入力します。 製品バージョンが、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックすると表示されます。 詳細については、「アップデート起動ツールのファイルのプロパティをカスタマイズする」を参照してください。

テーブル 3-79・アップデート起動ツールの設定（続き）

プロパティ	説明
説明	Update.exe のデフォルトの説明を独自の説明でオーバーライドするには、適切な説明を入力します。 説明が、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックしたときに表示されます。 詳細については、「アップデート起動ツールのファイルのプロパティをカスタマイズする」を参照してください。
著作権情報	Update.exe のデフォルトの著作権情報を製品の著作権情報でオーバーライドするには、製品の著作権情報を入力します。 著作権情報が、アップデート起動ツールの [プロパティ] ダイアログ ボックスに表示されます。この [プロパティ] ダイアログ ボックスは、エンド ユーザーが Update.exe ファイルを右クリックしてから、[プロパティ] をクリックすると表示されます。 詳細については、「アップデート起動ツールのファイルのプロパティをカスタマイズする」を参照してください。

テーブル 3-79・アップデート起動ツールの設定 (続き)

プロパティ	説明
必要実行レベル	″必要実行レベル″ 設定を使用して、インストールの Update.exe ファイルが Windows Vista プラットフォーム上でインストール (セットアップランチャー、すべての InstallShield 前提条件、および .msi ファイル) を実行するときに必要な最低権限レベルを指定できます。選択可能なオプションは、以下のとおりです。 ・ **管理者** — Update.exe の実行には、管理者権限が必要です。管理者は、Setup.exe の実行を承認する必要があります。非管理者は、管理者としての認証が必要になります。 ・ **最高権限** — Update.exe の実行には、管理者権限が推奨されます。管理者は、Setup.exe の実行を承認する必要があります。非管理者は、管理者権限を持たずに Setup.exe を実行します。 ・ **起動者** — Update.exe の実行に、管理者権限は必要ありません。したがって、管理者権限を持たないユーザーも Update.exe を実行することができます。Update.exe は、資格情報または同意 (コンセント) を求める UAC メッセージを表示しません。 ・ **前のセットアップ マニフェストを使用** — Update.exe マニフェストは前のセットアップで指定された必要実行レベルと同じレベルを使用します。デフォルトでは、これが設定されています。 [Update.exe を作成する] チェック ボックスが選択されている場合、InstallShield は Update.exe 起動ツールにアプリケーション マニフェストを埋め込みます。このマニフェストは選択された実行レベルを指定します。Windows Vista よりも古いバージョンのオペレーティング システムでは、必要実行レベルは適用されません。 [Update.exe を作成する] チェック ボックスがクリアされている場合、InstallShield は Update.exe 起動ツールに Windows アプリケーション マニフェストを埋め込みません。 詳細については、「インストール中におけるユーザー アカウント制御のプロンプトの数を最小化する」を参照してください。

履歴

[一般情報] ビュー内の [一般情報] エクスプローラーにある [履歴] 項目は QuickPatch プロジェクトの概要です。すべての関連リリースの一覧を表示し、現在の QuickPatch プロジェクトでどのリリースをパッチするのかを指定することができます。

注意・ *[履歴] に表示されるリリースを開いて変更を加えた場合、最新プロジェクトは動作不可能となります。これは [履歴] にあるリリース間で共有されている中間データが不足または改定された可能性があるためです。*

メモ・ *[履歴] でどのリリースをパッチするかを指定する場合、それを識別するために InstallShield は各リリースに関連する特定のバージョンを検索します。*

カスタム アクション

[一般情報] ビュー内の [一般情報] エクスプローラーにある [カスタム アクション] 項目は、パッチを作成しているオリジナル インストール プロジェクトで定義されているカスタム アクションのリストです。現在の QuickPatch プロジェクトで実行するカスタム アクションを指定することができます。

[ファイル] ビュー

[ファイル] ビューで、QuickPatch のファイルを管理をしたり、オリジナル インストールに使われているファイルのバージョン番号、言語および他の情報を見たりすることができます。[パッチするファイル] および [元のセットアップ ファイル] エクスプローラー内のすべてのファイルはキーファイル順にリストされています。

[元のセットアップ ファイル] エクスプローラーの中のファイルをクリックすると、ファイル名、インストール先、バージョン番号、および他の関連する情報が右のペインに表示されます。このエクスプローラーから [パッチするファイル] エクスプローラーにファイルをドラッグ アンド ドロップすることができます。[パッチするファイル] エクスプローラーでは、QuickPatch が元のインストールに適用されたときに追加、変更または削除されるファイルがすべて表示されます。

新規ファイルの設定

[ファイル] ビュー内の [パッチするファイル] エクスプローラーで新規ファイルをクリックすると、右のペインで次のような設定を構成することができます。

ファイルの設定

テーブル 3-80 ・ [ファイルの設定] 領域にある設定

設定	説明
パッチセットアップに追加するファイルを指定します。	オリジナル インストールに含まれるファイルに変更を行うとき、パッチが適用されるときにターゲット システムにインストールするファイルの最新バージョンを指定します。
自己登録	追加するファイルが自己登録 DLL または OCX ファイルの場合、このチェック ボックスを選択します。 ***メモ・** このオプションは .exe または .tlb ファイルの自己登録を行いません。*
COM 情報の抽出	ファイルが COM サーバーの場合で、パッチのビルド時にファイルから COM 情報を自動的に抽出するには、このチェック ボックスを選択します。

統合の設定

ファイルの保存先、および、このファイルと関連付ける機能を選択します。

メモ: このインストール状態には機能の関連付けが必要です。QuickPatch プロジェクトへ新規データを追加する場合、それを機能と関連付ける必要があります。新規データは対応する機能がインストールされている場合のみインストールされます。

変更/削除されたファイルの設定

[ファイル] ビュー内の [パッチするファイル] エクスプローラーでオリジナル インストールに追加したファイルをクリックすると、次のような設定を構成することができます。

アップデートされたファイル

テーブル 3-81 ・[アップデートされたファイル] 領域の設定

設定	説明
ファイルの最新バージョンを指定する	オリジナル インストールに含まれるファイルに変更を行うとき、パッチが適用されるときにターゲット システムにインストールするファイルの最新バージョンを指定します。
ファイル全体として含める	選択したファイルを、バイト レベルのファイルの差分だけではなく、1 つのファイルとしてパッチに含める場合、このチェック ボックスの選択を考慮してください。その場合、InstallShield は Windows Installer が選択されたファイルの実際のバージョン番号をすべて無視し、またターゲット システムのファイルを QuickPatch パッケージに含まれるバージョンでアップデートするかしないかを決定するときに、選択されたファイルがバージョン 1.0.0.0 であると想定するようにパッチを構成します。 ターゲット システムのファイルを QuickPatch パッケージに含まれるバージョン指定がされていない対応する新しいファイルで常に上書きする場合、バージョンが指定されていないファイルでこのチェック ボックスを選択すると便利です。Windows Installer ファイルの上書き規則では、バージョンが指定されているファイルが、バージョンが指定されていないファイルに優先します。したがって、このチェック ボックスを選択すると、Windows Installer はバージョンが指定されていないファイルを QuickPatch パッケージに含まれるファイルでアップデートします。これは、QuickPatch に含まれるファイルのバージョンが指定されていると判断するためです。 ***重要・*** *Windows Installer ファイルの上書き規則によると、ターゲット システム上のファイルの方がインストールされるバージョンより新しい場合でも、常に最新バージョンのファイルが保持されます。このため、ターゲット システム上のファイルのバージョンが指定されている場合、一部の状況において、ターゲット システム上のファイルがアップデートされない可能性があるため、このチェック ボックスの選択は避けたほうが良い場合があります。たとえば、ターゲット システム上のファイル バージョンが 1.1.0.0 で、QuickPatch プロジェクトのファイル バージョンが 2.0.0.0 の時にこのチェック ボックスを選択すると、実行時、ターゲット システム上のファイルはアップデートされません。これは、QuickPatch パッケージのファイルの実際のバージョン番号がターゲット システム上のファイルのバージョン番号よりも新しいにも関わらず、Windows Installer が QuickPatch のファイル バージョンをターゲット システム上のファイル バージョンよりも古い 1.0.0.0 と判断したためです。*

テーブル 3-81 • [アップデートされたファイル] 領域の設定 (続き)

設定	説明
COM 情報の抽出	ファイルが COM サーバーの場合で、パッチのビルド時にファイルから COM 情報を自動的に抽出するには、このチェック ボックスを選択します。このチェック ボックスは、ベース .msi パッケージの Class または TypeLib テーブルに COM 情報を含むパッチ ファイルのデフォルトとして選択されています。 ***メモ:*** *パッチを行う既存のファイルを指定するとき、InstallShield は自動的に、そのファイルが自己登録型であるかどうかを検出します。元にファイルが自己登録型である場合、パッチに含まれるファイルも自己登録型に設定されます。*

元のファイルの情報

[オリジナル ファイル情報] では、オリジナル インストールのオリジナル ファイルについての情報を見ることができます。

このボックスをチェックして、ファイルをセットアップからパッチで削除する

パッチが適用されるとき、ターゲット マシンからファイルを削除する場合、このチェック ボックスを選択します。

[レジストリ] ビュー

QuickPatch プロジェクトの [レジストリ] ビューでは、現在ソースマシンに存在する項目をビジュアル表示、またはインストール先/ターゲット システム上でパッチプロジェクトがビルドされ、適用された後にパッチを行う項目をビジュアル表示します。[インストール先コンピューターの [レジストリ] ビュー] ペインおよび [インストール先コンピューターのレジストリ データ] ペインは、元のインストールのレジストリ エントリが予め挿入されています。

インストール先コンピューターのレジストリ データペインに表示される各項目は、値タイプとデータの現在の状況を示すアイコンの隣に表示されます。変更されたデータ値のアイコンは水色(ターコイズ)の鉛筆があります。新規データ値のアイコンは右上に赤い星印と共に表示されます。ターゲット システムから削除されるレジストリ値のアイコンは、赤い X 印が付いています。以下のサンプルアイコンと説明を参照してください。

テーブル 3-82 • QuickPatch プロジェクトの [レジストリ] ビューのアイコン

アイコン	説明
	このアイコンは、QuickPatch が適用されるときターゲット システムに追加される新規 DWORD 値を示します。
	このアイコンは、QuickPatch が適用されるときターゲット システムで更新される変更済みの DWORD 値を示します。
	このアイコンは、以前のインストールからターゲット システム上にある既存 DWORD 値を示します。
	このアイコンは、ターゲット システム上に存在し、QuickPatch プロジェクトが適用されるとき削除される DWORD 値を示します。

エラーと警告

次のトピックは、インストール作成作業中に起こる可能性のあるエラーおよび警告についての情報を提供します。

テーブル 3-1・エラーおよび警告トピックのリスト

トピック	説明
ビルド エラーと警告	ビルド処理中に発生する可能性があるエラーと警告をリストします。
アップグレード エラーと警告	以前のバージョンの InstallShield で作成されたプロジェクトを現在のバージョンにアップグレードしたときに発生する可能性があるエラーと警告をリストします。
Windows Installer ランタイム エラー	実行時に発生する可能性がある Windows Installer エラーをリストします。このエラーは、一部の種類の機能をサポートするために InstallShield プロジェクトに自動的に追加されるビルトイン InstallShield カスタム アクションに関連しています。
Setup.exe ランタイム エラーと警告	Setup.exe の実行時に発生する可能性があるエラーをリストします。
Visual Studio プロジェクトのインポート エラーと警告	次のどちらかの処理を行ったときに発生する可能性のあるエラーと警告をリストします: ・ Visual Studio セットアッププロジェクト (.vdproj) を InstallShield プロジェクト (.ise) に変換。 ・ Visual Studio セットアップまたはマージ モジュール プロジェクト (.vdproj) を InstallShield プロジェクト (.ise) にインポート。

Windows Installer の実行時エラーの詳細については、Windows Installer ヘルプ ライブラリの Windows Installer Error Messages を参照してください。

ビルド エラーと警告

下記の表は、ビルド エラーや警告が発生した際のトラブルシューティングについて説明します。

テーブル 3-2・ビルド エラーと警告

エラー	説明	トラブルシューティングのヒント
-32000	ビルドがユーザーによってキャンセルされました。	このエラーが発生するのは、ビルド プロセスの途中でビルドが終了した場合のみです。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7347	InstallShield では、.spc および .pvk ファイルを使った署名がサポートされていません。プロジェクト内のデジタル証明書情報を .pfx 証明書または証明書ストアにある証明書への参照と入れ換えてください。	このエラーは、.spc および .pvk ファイルを使ったデジタル署名がサポートされている InstallShield の以前のバージョンで、これらのファイルをプロジェクトに追加していた場合、そのプロジェクトを InstallShield 2021 Express Edition で開いてリリースまたはパッチをビルドしようとしたときに発生します。 このエラーを解決するには、.spc の参照をリリースまたは QuickPatch プロジェクトから削除します。これを、.pfx 証明書または証明書ストアにある証明書への参照と入れ換えることができます。詳しくは、次を参照してください: • デジタル署名とセキュリティ • ビルド時にリリースとそのファイルにデジタル署名を行う • QuickPatch パッケージに署名する
-7346	リリースに含まれるファイルが SHA-1 証明書を使って署名されています。Windows は、2016 年 1 月以降に SHA-1 証明書を使って署名されているファイルを信頼しません。	この警告は、ビルド時に SHA-1 証明書を使ってリリースにデジタル署名を行うようにプロジェクトが構成されている場合に発生します。 SHA-1 はセキュリティの脆弱性があるため、SHA-256 の使用が推奨されます。したがって、InstallShield プロジェクトに含まれる任意の SHA-1 証明書は、SHA-256 証明書と置き換えることが推奨されます。 InstallShield でリリースに署名するための SHA-1 証明書を SHA-256 証明書に置き換えるには、[リリース] ビューの [署名] タブを使って、現在の証明書への参照を SHA-256 証明書と置き換えます。 詳細については、「デジタル署名とセキュリティ」を参照してください。
-7256	ソフトウェア識別タグ ファイルにデジタル署名するためには、.NET Framework 3.5 のインストールが必要です。	プロジェクトにソフトウェア識別タグを含めるように構成した場合、ビルドするリリースに .pfx ファイルを使用してリリースにデジタル署名を行うと、InstallShield はビルド時に作成するタグにデジタル署名を行います。ビルド マシンに .NET Framework 3.5 がインストールされていない場合、このビルド警告が表示されます。 このビルド警告を解決するためには、ビルド マシンに .NET Framework 3.5 をインストールしてください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7255	ソフトウェア識別タグファイルのデジタル署名には、.pfx ファイルのみ使用できます。	プロジェクトにソフトウェア識別タグを含めるように構成した場合、ビルドするリリースに証明書ストアにある証明書を使ったデジタル署名を構成すると、InstallShield がこのビルド警告を生成します。 このビルド警告を解決するためには、証明書ストアにある証明書を .pfx ファイルに切り替えてください。デジタル署名付きのタグが不要な場合、この警告は無視できます。
-7236	[一般情報] ビューの %1 設定の値が無効であるため、InstallShield は、ソフトウェア識別タグを作成できませんでした。以下の文字が含まれていないことを確認してください: ¥ / : * ? ″ < > |	InstallShield は、タグファイル (*TagCreatorID_UniqueID*.swidtag) の名前の一部として、入力済みの ″一意な ID″ および ″タグ作成者 ID″ 設定の値を使用します。したがって、入力する ID に、ファイル名には使えない文字は使用できません。 この問題を解決するためには、警告メッセージで通知された設定に有効な値を入力してください。 詳細については、「製品のソフトウェア識別タグを含める」を参照してください。
-7235	[一般情報] ビューの %1 設定が空白であるため、InstallShield は、ソフトウェア識別タグを作成できませんでした。	ソフトウェア識別タグを作成するには、[一般情報] ビューにあるいくつかの設定に値が必要です。この警告を解決するには、警告メッセージで通知された設定に適切な値を入力するか、または ″ソフトウェア識別タグの使用″ 設定で [いいえ] を選択します。 詳細については、「製品のソフトウェア識別タグを含める」を参照してください。
-7234	テンプレート ファイル Swidtag.xml を開くことができなかったため、InstallShield はソフトウェア識別タグを作成できませんでした。	InstallShield はタグ ファイルをビルドする際に、以下のいずれかの場所にインストールされている Swidtag.xml ファイルを使用します: *InstallShield Program Files フォルダー*¥Support¥0409 *InstallShield Program Files フォルダー*¥Support¥0411 この問題を解決するには、ロックが解除されている Swidtag.xml ファイルが存在することを確認してください。 詳細については、「製品のソフトウェア識別タグを含める」を参照してください。
-7222	メディアにサポート言語が含まれていません。メディアに最低 1 つのサポート言語を選択しなくてはなりません。	このエラーはメディアにサポート言語がない場合のみ発生します。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7211	圧縮済みネットワーク イメージ Setup.exe をビルドしています。その他のビルドの種類は、InstallShield の評価モードでは使用できません。	InstallShield を評価モードで使用している (つまり、アクティベートしていない) 場合、圧縮済み Setup.exe ファイルをビルドできますが、その他のリリースの種類をビルドすることはできません。Windows Installer ベースのリリースを作成すると、.msi データベースが常に Setup.exe ファイル内に組み込まれます。 評価モードで非圧縮リリースをビルドしようとすると、非圧縮オプションは無視されて、InstallShield によってビルド警告が表示されます。 InstallShield のアクティベーションを行うと、評価モードの制限が解除されます。
-7209	現在の Windows Installer パッケージ名には Unicode 文字が含まれているため、圧縮済みネットワーク イメージ Setup.exe をビルドする際にランタイム エラーが発生する可能性があります。	このビルド警告は、以下の条件が当てはまる場合に発生します: • .msi パッケージの名前に、ビルド マシンの ANSI コード ページでサポートされていない Unicode 文字が含まれている。 • リリースにセットアップランチャー (Setup.exe file) が含まれている。 この警告は、Windows Installer パッケージが Setup.exe からターゲット システム上の一時保管場所に抽出された場合に、エラー 1152 が発生する可能性があることを通知します。 この警告を解決するには、マシンの ANSI コード ページでサポートされていない Unicode 文字を含まないように .msi パッケージの名前を変更します。そのためには、[リリース] ビューにある Express ノードをクリックすると表示される "MSI パッケージ ファイル名" 設定を使用します。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7208	このパッケージの現在の場所は、Windows Installer が .cab ファイルを抽出しようとしたときにランタイム エラーの原因となる Unicode 文字を含みます。	この警告は、以下の条件が当てはまる場合に発生します: ・ InstallShield がビルドしているリリースのパスにビルド マシンの ANSI コード ページでサポートされていない Unicode 文字が含まれている。 ・ 製品のプログラム ファイルが .cab ファイルに圧縮されるようにリリースを構成している。 エンド ユーザーがビルド マシンの ANSI コード ページでサポートされていない Unicode 文字を含まないパスの場所からリリースを起動する場合は、この警告を無視できます。 エンド ユーザーがターゲット システムの ANSI コード ページでサポートされていない Unicode 文字を含むパスからインストールを起動しようとすると、Windows Installer が .cab ファイルを抽出しようとしたときにランタイム エラー 1311 が発生します。
-7185	%1 トランスレーション (文字列 ID %2) には、コードページ %3 で使用できない文字が含まれています。	このエラーは、指定言語に入力された文字列の値に、その言語のコードページでは使用できない文字が含まれているとき発生します。このエラーを解決するには、適切なコードページから文字を使用するように、[文字列エディター] ビューで文字列の値を編集します。 ***エディション・*** *InstallShield Premier および InstallShield では、.msi データベースに UTF-8 エンコードを使用するリリースをビルドできます。UTF-8 エンコードは、すべての言語の文字を同時にサポートするため、エンド ユーザーに表示するテキストおよびファイル名とレジストリ キーの両方で、たとえば日本語とドイツ語、またはロシア語とポーランド語のように自由に言語を組み合わせて使用できます。組み合わされた言語は、ターゲット システムの現在の言語設定に関わらず正しく表示されます。*

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7184	%1 列 (%2 テーブル) に、コードページ %3 で使用できない文字が含まれています: "%4"	このエラーは、プロジェクトの一部に、1 つまたは複数のターゲット言語のコードページで使用できない文字が含まれているときに発生します。 このエラーを解決するには、適切なコードページからの文字を使用するように、エラー メッセージで説明されたデータを変更します。たとえば、エラー メッセージが Shortcut テーブルについて記述している場合、その文字列を [ショートカット/フォルダー] ビューで変更してみてください。 ***エディション•*** *InstallShield Premier および InstallShield では、.msi データベースに UTF-8 エンコードを使用するリリースをビルドできます。UTF-8 エンコードは、すべての言語の文字を同時にサポートするため、エンド ユーザーに表示するテキストおよびファイル名とレジストリ キーの両方で、たとえば日本語とドイツ語、またはロシア語とポーランド語のように自由に言語を組み合わせて使用できます。組み合わされた言語は、ターゲット システムの現在の言語設定に関わらず正しく表示されます。*
-7128	setup.exe.manifest の埋め込みエラー。	このエラーは、Setup.exe ランチャでのアプリケーション マニフェストの埋め込みに問題があるときに発生します。アプリケーション マニフェストでは、Windows Vista 以降のプラットフォーム上でインストール (セットアップランチャー、InstallShield 前提条件および .msi ファイル) を実行するためのインストールの Setup.exe ファイルによって必要とされる最低特権レベルを指定します。 このエラーは、InstallShield Supportフォルダーからテンプレート マニフェスト ファイルが不足している場合にも発生します。また、Setup.exe テンプレートが InstallShield Redist¥Language Independent¥i386 フォルダーから不足している場合にも発生します。 このエラーを解決するには、InstallShield の修復を実行してください。
-7124	MSBuild を起動できませんでした。.NET Framework %1 が、システムの '%2' にインストールされていることを確認してください。Visual Studio 内から InstallShield を実行するとき、Visual Studio 2005 以降が必要です。	MSBuild を使って、InstallShield プロジェクトを含む Visual Studio ソリューションをビルドするには、開発マシンに .NET Framework 3.5 以降が必要です。また、Visual Studio 2005 以降を使用しなくてはなりません。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7113	%1 の言語サポートは、この Edition には含まれていません。	このエラーは、ひとつのプロジェクトでデフォルトの数を超える数の言語を有効化した場合、Premier Edition 以外では 2 言語以上を使ってビルドしたときに発生します。(最初のエラーで停止するよう設定されていない限り) ビルドは成功しますが、デフォルトの数を超えるの言語はビルドされません。制限数以内の最初の 2 つの言語はビルドされます。%1 はこのリリースに含まれていない言語の名前で置換されます。このエラーを解決するには、リリースのビルドに InstallShield Premier Edition を使ってください。
-7084	VBScript カスタム アクション %1 は有効な VBS ファイルをポイントしていません。	この警告メッセージは、インストールに VBScript カスタム アクションを追加するとき、カスタム アクションに指定されたファイルが VBScript ではないときに表示されます。このエラーを解決するには、[カスタム アクション] ビューで指定されたカスタム アクションに対して適切なタイプのファイルを選択します。
-7076	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7075	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7074	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7073	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7072	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7071	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7069	XML 読み取りオブジェクト (MSXML3.dll) を作成することができませんでした。このファイルがシステムに登録されていることを確認してください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7064	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7063	コンポーネント %2 のファイル %1 を Windows Fonts フォルダーにインストール中ですが、Font テーブルに対応するレコードがありません。Font はターゲット システムに適切に登録されません。	この警告を解決するには、[ファイル] ビューを開いて、メッセージの中で指摘のあったフォントを削除し、そのあと Fonts フォルダーにもう一度そのフォントを追加します。InstallShield は Font テーブルに対応するレコードを自動的に追加します。
-7062	クライアント マシンから COM+ に関する設定をリフレッシュ中です。クライアント マシンから COM+ の設定をリフレッシュ中に、エラーが発生しました。	このビルドエラーは、例外エラーが、[コンポーネント サービス] ビューで選択された COM+ アプリケーションで起きたときに発生します。COM+ 設定をリフレッシュするオプションが、ビルド プロセスの最中に提供されます。ビルドエラー -7062 は、このオプションを選択して、リフレッシュ プロセスが失敗したとき発生します。このエラーを解決するには、[コンポーネント サービス] ビューを開いて COM+ アプリケーションを選択し、インストール タブでビルド時にクライアント マシンから COM+ 設定をリフレッシュするチェック ボックスをクリアします。
-7061	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7060	ライセンス要件に従い、InstallShield では MSI 3.0 Beta からMSI 3.0 エンジン再配布可能ファイルを提供する必要があります。instmsi.exe を MSI 3.0 Beta フォルダーから ¥redist¥Language Independent¥i386¥MSI3.0¥instmsiw.exe へコピーしてください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7059	"最新のパッチ バージョン" プロパティをアップデートすることができません。従って、このプロジェクトからの将来のパッチ ビルドは適切に配列されない可能性があります。プロジェクトが読み取り専用になっていないことを必ず確認してください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7058	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7057	この配布に FlexNet Connect が有効になっていますが、アップデートを確認するための URL が指定されていません。これは 'CD-ROM' 配布構成に必要です。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7053	"インターネット" 配布構成が有効になっていますが、"インストール先" フィールドが作成されていません。配布を実行したときにエラーが発生します。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7052	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7051	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7050	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7049	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7048	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7045	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7044	メイン アプリケーション マニフェストをリリース固有の場所へコピー中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7042	プライマリ アプリケーション アセンブリのソースの場所を決定中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7041	デフォルトのデプロイメント マニフェストを初期化しています。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7040	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7038	リリースの場所にベースのデプロイメント フォルダーを作成中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7037	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7036	プライマリ アプリケーション アセンブリのロード中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7035	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7030	ビルドが必要なリリースをロード中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7029	ビルドエンジンが 'CE-ROM' リリースタイプ用の設定をロードすることができません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7028	ビルドエンジンが 'ファイル共有' リリースタイプ用の設定をロードすることができません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7027	ビルドエンジンが 'インターネット' リリースタイプ用の設定をロードすることができません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7026	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-7025	COM+ コンポーネント サーバーファイル '%1' が、システムで見つかりません。コンポーネント '%2' はターゲット システムにインストールされません。COM+ アプリケーションがインストールされた後に、コンポーネントを手動でインストールする必要があります。	%1 は、不足している COM+ DLL を参照し、%2 は ProgID を参照します。この警告を解決するためには、システムの適切な場所に DLL を追加するか、プロジェクトから COM+ コンポーネントを削除します。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7024	J# をビルドに含むには、.NET 1.1 も含める必要があります。	J# には、.NET Framework 再頒布可能パッケージ バージョン 1.1 が必要ですが、インストールに含まれていません。エンドユーザーのシステム上に .NET Framework 1.1 があることが確実に分かっている場合は、この警告メッセージを無視してください。.NET Framework 1.1 を持っていない可能性がある場合、それをインストールに追加します。.NET Framework がエンドユーザーのシステム上にない場合、J# をインストールの一部としてインストールする際、問題が起きる可能性があります。
-7023	内部ビルド エラー。	このエラーは、マージ モジュールが適切に構成されていない、または破損しているときに発生します。プロジェクトにマージ モジュールが複数存在するときに、どちらがエラーの原因になっているか分からない場合、まず最初にその問題になっているマージ モジュールを特定する必要があります。方法として、まず、プロジェクトからマージ モジュールのどちらか 1 つを削除し、リリースをビルドしてみます。エラーが続けて起こるようであれば、その問題になっているマージ モジュールを特定したことになります。そうでない場合、別のマージ モジュールを削除して、またビルドします。エラーの原因になっているマージ モジュールの特定ができた時点で、再インストールしてみます。エラーがそれでも解決されないようであれば、作成者に問い合わせて最新のバージョンを取得してください。
-7012	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-7011	%1 が GlobalAssemblyCache へインストールされるためにはストロング ネームが必要です。	ストロング ネームが使われている場合のみ、Global Assembly Cache にアセンブリをインストールすることができます。ストロング ネームには、アセンブリの簡易テキスト名、バージョン番号、カルチャ情報(利用可能な場合)、パブリック キーおよびデジタル署名が使われます。Global Assembly Cache でストロング ネームのアセンブリを使って、複数の異なるバージョンの .dll ファイルを同名で持つアセンブリを複数バージョン持つことができます。Global Assembly Cache に保存されているアセンブリはコンピューター上で異なるアプリケーション間で共有することができます。このエラー メッセージを解決するとき、2 つのオプションがあります。 • アセンブリを他のアプリケーションと共有することが明示的に要求されていない場合、アセンブリのターゲット インストール先を Global Assembly Cache 以外の場所に変えます。COM interop がアセンブリにアクセスできるようにするためには、アセンブリを Global Assembly Cache にインストールする必要はありませんので注意してください。 • Global Assembly Cache にインストールするためにストロング ネームが使われているアセンブリを選択します。アセンブリは、1 度作成されると、ストロング ネームで署名できませんので注意してください。アセンブリは、作成時のみストロング ネームで署名できます。
-6654	ファイル %1 はコンポーネント %2 のキーファイルではありません。製品とともにインストールされる標準 DLL の機能を呼び出すときに、そのアクションが遅延実行としてスケジュールされる場合、呼び出される DLL はコンポーネントのキーファイルでなければなりません。	このエラーは、以下のいずれかが True のとき発生します。 • ファイルが PE として認識されなかったため、InstallShield は このファイルをコンポーネントのキーファイルとして設定しませんでした。 • ファイルがダイナミック(動的)にリンクされているため、InstallShield は このファイルをコンポーネントのキーファイルとして設定しませんでした。動的にリンクされたファイル をコンポーネントのキーファイルに設定することはできません。 このビルド エラーを解決するには、次のいずれか 1 つを実行します。 • [オプション] ダイアログ ボックスの [ファイル拡張子] タブで必ず DLL が PE ファイルとしてリストされていることを確認してください。 • 動的にリンクされたフォルダーからのファイルを除外して、そのファイルをダイナミックリンクを張らずにプロジェクトに追加します。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6653	インストールの機能 %1 には、1600 以上のコンポーネントが含まれています。1 機能につき、含めることのできる最大コンポーネント数は 1600 です。	このエラーを解決するには、エラーメッセージで示された機能をサブ機能に分けて、機能のファイルをこれらのサブ機能に再割り当てします。
-6651	ビルドしているセットアップに 32,767 を超えるファイルがあります。セットアップ パッケージを適当な MSI スキーマに自動的に切り替えます。	この警告は、インストール パッケージに含まれるファイルが 32,767 ファイルを超える場合に表示されます。InstallShield は自動的に大きなパッケージ スキーマを適用します。 ***注意**· パッケージで大きなパッケージ スキーマを使う場合、パッチに問題があります。パッチをビルドする場合、エラー「プライマリ トランスフォームを生成できません。」がトリガーされます。*
-6647	ファイルを '%1' から '%2' へ移動できません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6646	マージ モジュール '%2' の '%1' プロパティが NULL に設定されています。このプロパティは NULL を受け付けません。プロパティを NULL にすることはできません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6645	%1 はロードに失敗しました。エラー: %2 エラーの説明: %3	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6641	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6640	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6639	マージ モジュール '%1' は次のマージ モジュールのどれか 1 つをセットアップへ含むことを必要とします: %2	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6638	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6637	*"C:¥RegTest¥SingleSlash1"=dword: 2* のインポート中、コンポーネント NewComponent1 の無効レジストリ データ (イタリック体の情報が無効データを示します)。	ここに示した無効データは円記号を 2 つではなく 1 つしか使わなかった例です。別の例では *dword* ではなく *DWORD* を入力しています。
-6636	キーファイル %1 および %2 が File テーブルで見つかりました。大文字小文字に関わらず、キーが同一名の場合、ファイルがターゲット システムにインストールされたとき予期しない結果が招くことがあります。これは、キャビネット ファイルに圧縮されたファイルがファイルキーを使って名前が付けられるためです。この問題を解決するためには、圧縮セットアップまたはマージ モジュールをビルドする場合、ファイルキーの 1 つをキャビネット ファイルで一意になるように変更します。[ダイレクト エディター] ビューでファイル キー名を変更することができます。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6635	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6634	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6633	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6632	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6624	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6623	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6622	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6620	キー同期用の参照パッケージ '%1' を開くことができませんでした。	wwwwwwwwwwwwwwwwwwwwこのエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6619	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6618	動的に作成されたコンポーネントへのキーパスの設定中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6617	ファイル '%1' を同期マップへ追加中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6616	コンポーネント '%1' へのキーパスの設定中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6615	ファイル '%1' の File.FileName をアップデートすることができませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6613	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6612	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6611	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6558	InstallExecuteSequence テーブルのカスタム アクション %1 はインストールされたファイルから実行されます。カスタム アクションを問題なく実行するには、ソース ファイルがローカルにインストールされたことを確認する条件が必要な場合があります。	インストールされたファイルから実行されるカスタム アクションを問題なくに実行するには、条件ステートメントを使って確実にファイルをローカルにインストールしなくてはなりません。 ・ カスタム アクションが RemoveFiles の前にシーケンスされている場合　機能がローカルにインストールされている場合のみアクションを実行します。 ・ カスタム アクションが RemoveFiles とInstalLFiles の間にシーケンスされている場合　機能がローカルにインストールされている場合のみアクションを実行します。アンインストールではアクションを実行しません。 ・ カスタム アクションが InstallFiles の後にシーケンスされている場合　機能がローカルにインストールされる場合のみアクションを実行します。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6557	<InstallShield>¥Support¥NetCF.ini ファイルから指定されたプラットフォーム/プロセッサの .NET Compact Framework が見つかりませんでした。	.NET Compact Framework が正しい場所にあるか確認してください。
-6556	セットアップへ .NET Compact Framework .cab ファイルを含む際にエラーが発生しました。	このエラーは次の理由で発生した可能性があります: ・ *InstallShield Program Files フルダー*¥Support¥.NetCR.ini file ファイルで指定された場所に .NET Compact Framework ファイルが見つかりませんでした。 ・ CEDefault.ico ファイルが *InstallShield プログラム ファイル フォルダー*¥Program¥04xx フォルダーにありません。 ・ ISCEInstall テーブルへの書き込みエラー。
-6555	<InstallShield>¥Support¥CFAppMgr.ini ファイルを一時保管場所にコピーすることができません。	CFAppMgr.ini ファイルが *InstallShield Program Files フォルダー*¥Support フォルダーにあることを確認してください。このファイルがない場合、InstallShield セットアップを修復モードで起動してください。一時ファイルをすべて削除してシステム上の空きスペースを増やしてください。
-6553	プロセッサの .NET Compact Framework .cab ファイルが見つかりませんでした。	*InstallShield Program Files フォルダー*¥Support¥NetCF.ini ファイルを開き、このプロセッサの .NET CF 選択を構成して問題を解決してください。
-6551	NewComponent1 コンポーネントのレジストリレコードである Registry1 の名前列と値列が空です。Windows Installer では、空の文字列にデフォルト値の HKEY_LOCAL_MACHINE¥New Key #1が設定されます。"デフォルト" 値を "設定しない" に設定する場合、[レジストリ] ビューで "存在しない場合はインストールする、存在する場合はアンインストールする" をそのキーに設定します。キーがターゲットマシン上に既に存在する場合は、(デフォルト) 値は変更されません。	この警告は、プロジェクトに追加されて、デフォルトのキーが手動で設定されなかったすべてのレジストリで表示されます。"デフォルト" 値を "設定しない" に設定するには、[レジストリ] ビューで "存在しない場合はインストールする、存在する場合はアンインストールする" をそのキーに設定する必要があります。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6550	%1 はグローバル アセンブリ キャッシュにインストールできる有効な .NET アセンブリではありません (%1 には完全パスを差し替える)。	有効なファイルのみグローバル アセンブリ キャッシュにインストールされます。インストール用に選択したファイルが有効なアセンブリであることを確認してください。
-6530	アップグレード項目 [2] のセットアップ [1] 処理エラー。これは有効なセットアップではありません。	このエラーは [アップグレード] ビューにアップグレード エントリが存在し、指定された以前のインストールが指定の場所に存在しない場合に発生します。これは、以前のインストールが [アップグレード] ビューでされたために移動されたか、または以前のインストールを含むフォルダーが権限に関する問題でアクセスできないために起きます。また、これエラーは、参照された以前のインストール (Setup.exe または .msi) が有効な Windows Installer ベースのインストールではない場合も起きます。
-6525	InstallExecuteSequence テーブルのカスタム アクション %1 はインストールされたファイルから実行されます。%2 アクションの後にシーケンスしなければなりません。カスタム アクションが適切にシーケンスされ、ベース アクションがシーケンスに存在することを確認してください。	このエラーは、セットアップの最中にインストールされたファイルから実行するカスタム アクションをオーサリングしたときに発生します。カスタム アクションはインストールの After Setup Complete Success ダイアログの部分で起動します。さもなければ、Windows Installer がカスタム アクションを実行したときにファイルがターゲット システムに存在しません。
-6499	シャローフォルダー構造設定は複数ディスクリリースでは利用しないで下さい。	シャローフォルダー構造設定は [リリース] ビューの [リリース] プロパティグリッドから設定することができます。エラー -6499 が発生した場合、シャローフォルダー構造オプションを [いいえ] に変更して標準ビルドフォルダー構造を作成するようにします。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6479	内部ビルド エラー。	プロジェクト内の破損したマージ モジュールがエラーの原因となっています。InstallShield と共に出荷されたサードパーティ マージ モジュール以外、または Microsoft Visual Studio .NET 2003 のベータ版と共に出荷されたマージ モジュールでエラーメッセージが報告されています。 このエラーを解決するため問題の原因となっているマージ モジュールを判断し、それをプロジェクトから削除します。マージ モジュールを見つけるには、[再配布可能ファイル] ビューでプロジェクトからすべてのマージ モジュールを削除します。そしてプロジェクトへ 1 つずつマージ モジュールを追加し、エラー -6479 が発生するまでリリースをビルドします。これがエラーの原因となっているマージ モジュールです。 エラーの原因となっているマージ モジュールを判断したら、そのモジュールのベンダーへ詳細を問い合わせてください。マージ モジュールが Microsoft 固有モジュールの場合、Visual Studio の最新サービスパックをダウンロードしてそのアップデートを取得します。マージ モジュールが Visual Studio .NET 固有モジュールの場合、Visual Studio .NET の最新サービスパックから取得したものであることを確認します。
-6478	FON ファイルはフォントタイトルを持たなくてはなりません。%2 機能の %1 フォント	ファイルの [フォントタイトル] プロパティを設定します。これを行なうには、[ファイル] ビューでファイルを右クリックしてプロパティを選択し、[フォント タイトル] フィールドにタイトルを入力します。フォントタイトルがない場合は、インストール時に実行エラーが発生します。
-6274	Setup.exe が次の Disk1 の場所に見つかりませんでした: %1。に見つかりません。ビルドテーブルのみを実行している場合は、リリースの前の完全ビルドが完了しているか確認してください。	このエラーは、セットアップが Setup.exe からスタンプを取得しようとしたにもかかわらず、Setup.exe が Disk1 フォルダーに存在しなかった場合に発生します。これは完全ビルドを途中でキャンセルしてから [ビルドテーブルのみ] オプションを実行した場合に発生します。
-6271 を入力します。	ファイル %1 が見つかりません。このファイルの MsiFileHash テーブルをビルド中にエラーが発生しました。指定した場所にファイルが存在することを確認します。	指定した場所に識別されたファイルが存在することを確認します。 ファイルに定義済みフォルダー VSSolutionFolder が含まれる場合は、「Visual Studio ソリューションで VSSolutionFolder パス変数を使用する」を参照してください。
-6270	アイコンテーブルのレコード %1 が %2 文字制限を超えています。そのためビルドでデータベースを維持できません。	レコードが制限文字数内であることを確認してください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6269 を入力します。	ディレクトリ %1 を %2 へコピー中にエラーが発生しました。ソース ディレクトリのパスが正しいことを確認してください。	ソース ディレクトリのパスを確認してください。
-6268 を入力します。	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6267	cab ファイル %1 を %2 の場所に抽出中にエラーが発生しました。	この場所が正確か確認してください。
-6266 を入力します。	%1 の CAB 抽出エンジンを作成および初期化中にエラーが発生しました	このエラーは、1 つ以上のファイルが登録されていない場合に発生することがあります。すべてのファイルが登録されていることを確認してください。
-6265 を入力します。	機能 '%1' をビルド中にエラーが発生しました。この機能はプロジェクトの別の機能と同じ名前で、大文字小文字が違います。標準プロジェクトでは機能名を重複できません。機能の 1 つの名前を変更してこのエラーを修正してください。	機能の 1 つを削除するか名前を変更してください。
-6264 を入力します。	%1 テーブルのレコードは文字列 ID %2 を列 %3 に使用していますが、この文字列は空白で、列はヌル不可能です。	列で使用する文字列を指定してください。
-6260 を入力します。	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6259	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6258	ファイル「%1」からデジタル署名情報を抽出中にエラーが発生しました。IDE に提供されたデジタル署名情報が正しいか確認してください。	デジタル署名情報が正しいことを確認してください。[リリース] ビューの [署名] タブで、デジタル署名情報にアクセスすることができます。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6257	PRODUCT_NAME がプロジェクトで定義されていることをビルドエンジンが検出しました。実行時にインストールは IDE で定義された製品名値ではなく、この値が製品名として表示します。	この警告メッセージは、ID PRODUCT_NAME が文字列テーブルで定義されているときに発生します。この文字列テーブル ID は [一般情報] ビューで設定した製品名値よりも優先されます。このエラーの解決法については、ナレッジベース 記事 Q109136、「ERRDOC: Build Error -6257」を参照してください。
-6255	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6254	ファイル %1 の MsiFileHash テーブルをビルド中にエラーが発生しました	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6253	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6252	内部ビルドエラー	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6251	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6250	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6248	機能 %2 の 依存関係 %1 が見つかりませんでした	このファイルは機能のキーファイルに必要です。このファイルが機能のキーファイルと同じディレクトリのビルドシステムにあることを確認してください。 依存関係が必要ない場合は、機能の .NET 依存関係スキャンをオフにすることができます。依存関係スキャンをオフにするには、機能の "ビルド時に .NET をスキャン" プロパティを [なし] か [プロパティのみ] に設定します。
-6247	.NET Framework 再配布可能ファイル %1 がシステムに見つかりません。%2	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6245	プロジェクトに含まれる 1 つ以上の機能に、.NET Framework を必要とする .NET プロパティが含まれています。リリースに .NET Framework を含めることをお勧めします。	最初に [リリース] ビューでメディアの種類を選択して .NET Framework をプロジェクトに追加します。メディアタイプを選択したとき、ビルドプロパティが表示されます。".NET Framework の場所" プロパティで適切なオプションを選択してください。 ・ ソース メディアからコピーする—このオプションを選択すると、.NET Framework のランタイムがソース メディアのルートに残されます。このオプションは、すべてのファイルを Setup.exe に圧縮している Web メディア タイプまたはネットワーク イメージ メディア タイプでは使用できません。 ・ Setup.exe から抽出—このオプションを選択すると、.NET Framework ランタイムが Setup.exe に圧縮され、実行時に抽出されます。 ・ Web からダウンロード—このオプションを選択すると、(必要に応じて) 指定の URL から .NET Framework ランタイム がダウンロードされます。このオプションを選択した場合、.NET Framework URL プロパティを適切な Web の場所に設定する必要があります。 ".NET Framework バージョン" プロパティでは、インストールと共に配布する .NET バージョンを選択します。
-6244	プロジェクトに含まれる 1 つ以上の機能に、.NET Framework を必要とする .NET プロパティが含まれています。ただし .NET Framework を検出することができません。%2	http://www.installshield.com/products/dotnet.asp から InstallShield .NET アップデートをダウンロードしてください。 Visual Studio リリースの候補がある場合は、dotnetfx.exe を Windows Component Update CD の dotnetframework ディレクトリ から *InstallShield Program Files フォルダー*¥Redist¥0409¥i386 にコピーできます。
-6243	InstallUtilLib.dll がシステムに見つかりません。	このファイルはインストーラーカスタム アクションに必要です。セットアップを実行し直して、Microsoft .NET Framework 再配布可能ファイルをインストールし直す必要があります。このファイルは http://www.installshield.com/products/dotnet.asp からダウンロードするか、(Visual Studio リリースの候補があれば)、Windows Feature Update CD の dotnetframework ディレクトリから dotnetfx.exe を *InstallShield Program Files フォルダー*¥Redist¥0409¥i386 にコピーすることができます。
-6242	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6241	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6240	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6239	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6238	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6237	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6236	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6235	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6234	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6233	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6232	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6231	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6230	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6229	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6228	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6227	機能 %1 の .NET スキャンが失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6226	Visual Studio .NET ソリューションを開いています。	ソリューションが指定の場所に存在するか確認してください。 さらに、Visual Studio がビルド マシンにインストールされていることを確認してください。
-6225 を入力します。	Visual Studio .NET プロジェクト出力 "%1" を解決しています	プロジェクトがソリューションに存在するか確認してください。さらに、そのプロジェクトに指定された出力があるか確認してください。
-6224 を入力します。	機能 %2 のマージ モジュール %1 を処理しています	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6223	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6222	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6221	機能 %2 から Visual Studio .NET プロジェクト出力 "%1" を解決できませんでした	プロジェクトがソリューションに存在するか確認してください。さらに、そのプロジェクトに指定された出力があるか確認してください。
-6219 を入力します。	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6218	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6217	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6216	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6215	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6214	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6213	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6212	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 • ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6211	機能 %1 のインストール先は GlobalAssemblyCache ですが、キーファイルは .NET アセンブリではありません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6208	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6207	セットアップ ファイル %s をビルド中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6205	機能 '%1' をビルド中にエラーが発生しました。機能のインストール先ディレクトリ %2 が directory テーブルにありません。機能のインストール先を有効な場所に変更してください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6204	機能 %2 に %1 をインポート中にエラーが発生しました。指定の場所にファイルが存在することと、ファイルが有効な REG ファイルであるかどうかを確認してください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6203	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6202	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6201	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6200	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6199	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6197	%1 を %2 へストリーム中に、エラーが発生しました。.msi パッケージが、現在別のプロセスで使用されていないことを確認してください。	.msi パッケージが現在別のプロセスで使用されていないことを確認してください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6196 を入力します。	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6195	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6194	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6193	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6192	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6191	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6190	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6189	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6188	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6187	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6186	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6185	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6184	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6183	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6182	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6181	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6180	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6179	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6178	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6177	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6176	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6175	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6174	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6173	ファイル %1 の名前を %2に変更中にエラーが発生しました。	interm フォルダーを含む製品構成のドライブに十分な空き容量があるか確認してください (たとえば、¥MySetups¥Your Project Name-27¥Product Configuration 1¥Interm)。
-6172 を入力します。	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6171	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6170	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6169	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6168	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6167	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6166	概要情報ストリームの Word Count Summary プロパティを更新中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6165	テーブル %1 をエクスポート中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6164	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6163	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6162	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6161	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6160	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6158	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6156	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6155	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6154	ビルドがユーザーによってキャンセルされました。	このエラーが発生するのは、ビルド プロセスの途中でビルドが終了した場合のみです。
-6153	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6152	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6151	ターゲットデータベースを保存できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6150	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6149	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6148	指定されたテーブルにレコードを挿入できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6147	テーブルの指定ターゲット フィールドを、更新できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6146	指定されたテーブルで、新規レコードを作成できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6145	テーブルの指定列から値を取り出せませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6144	テーブル %1 のデータベース ビューを開くことができませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6143	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6142	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6141	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6140	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6139	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6137	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6136	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6135	指定された機能が、セットアップの種類と関連付けられていませんでした。	指定された機能を、セットアップの種類と関連付けてください。
-6134	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6133	現在の製品構成で指定されたプロパティを更新中にエラーが発生しました。	製品構成の指定プロパティを調べて、それが有効であるか確認してください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6131	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6130	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6129	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6128	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6127	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6126	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6125	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6124	未定義の例外。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6123	未定義の例外。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6122	ビルドする機能が 1 つもありませんでした。	ビルドには、少なくとも 1 つの機能を含める必要があります。
-6121	ビルドしたマージ モジュールをモジュールフォルダーにコピーする際に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6120	リソースリンカが、指定された DLL のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6118	リソース コンパイラが、DLL のリンクに必要な指定 RES ファイルのビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6116	次のプロジェクトからの rc ファイルのエクスポートに失敗しました: %1。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6115	ISScript.msi から IDriver.exe のバージョンを取り出せませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6114	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6113	増分ビルド中に、エラーが発生しました。	[テーブルのみをビルドする] および [テーブルをビルドしてファイルを更新する] を選択する前に、前のメディアがビルドされたか確認し、また前のメディアが削除されていないことを確認してください。
-6112	Setup.exe にストリームしているファイルの削除でエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6111	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6110	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6109	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6108	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6107	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6106	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6105	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6104	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6103	ファイル %1 が見つかりませんでした。	指定した位置にファイルが存在すること確認してください。 ファイルに定義済みフォルダー VSSolutionFolder が含まれる場合は、「Visual Studio ソリューションにリファレンスを追加する」を参照してください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6102	″%1″ に一致するダイナミックファイルを検索中にエラーが発生しました。	指定したダイナミックファイルのフラグが有効であることを確認してください。
-6101	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6100	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6099	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6098	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6097	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6096	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6095	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6094	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6093	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6092	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6091	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6090	コントロール '%2' に指定されたスクロール可能テキスト ファイル '%1' が存在しません。	このビルド エラーは [使用許諾契約] ダイアログ ボックス用に (1) が指定されたスクロール可能テキスト コントロール、または [Readme] ダイアログ ボックスと (2) が無効な .rtf ファイルをポイントしてることが原因となっています。 このエラー メッセージを解決するには、[ダイアログ] ビューを開き、[ダイアログ] ノードを展開してから [使用許諾契約] ノードを選択します。"ライセンス ファイル" プロパティを ¥Redist¥0409¥Eula.rtf に設定します。Readme ノードを選択します。"Readme ファイル" プロパティを ¥Redist¥0409¥Readme.rtf に設定します。 別の方法として、[ライセンスファイル] プロパティまたは [Readme ファイル] プロパティを適切なパスを使って独自のファイルに設定することもできます。ビルダーがこのプロパティに指定されているパスでファイルを見つけることができる限り、エラーは発生しないはずです。
-6088	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6087	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6086	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6085	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6084	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6083	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6082	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6081	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6080	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6079	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6078	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6077	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6076	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6075	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6074	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6073	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6072	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6071	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6070	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6069	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6068	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6067	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6066	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6065	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6064	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6063	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6062	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6061	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6060	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6059	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6058	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6057	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6056	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6055	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6054	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6053	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6052	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6051	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6050	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6049	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6048	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6047	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6046	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6045	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6044	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6043	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6042	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6041	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6040	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6039	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6038	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6037	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6036	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6035	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6034	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6033	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6032	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6031	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6030	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6029	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6028	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6027	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6026	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6025	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6024	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6023	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6022	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6021	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6020	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6019	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6018	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6017	ビルドは COM 情報を抽出できませんでした。管理者として実行していることを確認してください。	InstallShield で COM 情報を抽出するためには、管理者権限が必要です。詳細については、「InstallShield を管理者権限を使って、または管理者権限を持たずに起動する違い」を参照してください。
-6016	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6015	次のファイルの短いファイル名を作成中にエラーが発生しました: %1。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6014	機能のショートカットがアイコンリソースを参照していないため、無効です。	これを解決するには、アイコンファイルとアイコンインデックスを、ショートカットに指定してください。
-6013	指定された機能の条件が無効です。	[機能] ビューから、指定機能の [条件] プロパティを修正してください。
-6012	言語 %1 のコードページを設定できませんでした。	指定した言語用のコードページをインストールしてください。
-6011	文字列テーブルを開けませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6009	Internet Explorer の Web メディアの作成中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6008	Internet Explorer の Web メディアの作成中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6007	指定された非圧縮セットアップファイルの Disk1 へのコピー中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6006	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6005	パッケージ定義ファイルを作成中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6004	Msi エンジンのデジタル証明書の Setup.exe へのストリーム中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6003	"%1" の Setup.exe へのストリーム中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6002	オブジェクトのカスタム ビルド セットアップを実行中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-6001	ISObjectProperty の前処理中に、エラーが発生しました。すべての項目に有効な IncludeInBuild タグが付いているか、確認してください。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-6000	補足マージ モジュール情報の収集中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5099	ビルド レポートを作成中にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5098	Directory テーブルに循環参照が含まれています。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5097	_Validation テーブルにカスタムレコードを書き込めませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5096	標準テーブルリストを読み出せませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5095	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5094	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5093	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5092	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5091	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5090	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5089	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5088	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5087	最初のエラー発生時に中止しました。	ユーザーが、[ツール] の [オプション] メニューで [最初のエラーの発生時にビルドプロセスを停止] を選択しています。エラーが発生したため、ビルドを中止しました。
-5086	警告をエラーとして処理します。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5085	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5084	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5083	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5082	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5081	Setup.ini に書き込めませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5080	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5079	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5078	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5077	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5076	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5075	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5074	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5072	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5071	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5070	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5069	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5068	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5066	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5065	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5064	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5063	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5062	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5061	指定されたファイル名は、既に存在します。	機能の "ソースの場所" プロパティを使うと、このエラーは発生しません。
-5060	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5059	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5058	マージ モジュールの依存関係を取得できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5057	マージ モジュールカタログを取得できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5056	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5054	ファイル "%1" のサイズを判別できませんでした。	指定したファイルが存在することを確認してください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5053	ファイル "%1" が見つかりませんでした。	指定したファイルが存在することを確認してください。
-5052	ボリューム %1 の空き容量を判別できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5051	ファイル "%1" から読み取り専用の属性を削除できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5050	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5049	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5048	ファイル "%1" を作成できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5047	ディレクトリ %1 を作成できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5046	以前のビルドレポートを保存できませんでした。	このビルド エラーは、リリースが存在するが、対応するリリース フォルダー内の Reports フォルダーが存在しない時に発生します。この状況は一般的に、Reports フォルダーを手動で削除しているが、ディスク イメージ フォルダーを削除していない場合に発生します。この状況はまた、プロジェクトを InstallShield の以前のバージョンからアップグレードしている場合にも発生します。 このエラーを解決するには、[リリース フォルダーを開く] ボタンをクリックします。現在のリリース フォルダーが開きます。そのフォルダー内のすべてのファイルを削除します。その後、リリースを再ビルドします。
-5045	以前のビルドログを保存できませんでした。	このビルド エラーは、リリースが存在するが、対応するリリース フォルダー内の LogFiles フォルダーが存在しない時に発生します。この状況は一般的に、LogFiles フォルダーを手動で削除しているが、ディスク イメージ フォルダーを削除していない場合に発生します。この状況はまた、プロジェクトを InstallShield の以前のバージョンからアップグレードしている場合にも発生します。 このエラーを解決するには、[リリース フォルダーを開く] ボタンをクリックします。現在のリリース フォルダーが開きます。そのフォルダー内のすべてのファイルを削除します。その後、リリースを再ビルドします。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5044	ディレクトリ %1 を削除できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5043	ボリュームが、存在しません。	[リリース] ビューを調べて、指定したボリュームが存在することを確認してください。
-5042	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5041	機能または機能のインストール先の指定に、文字列 ID ″%1″ が使用されています。	インストール先の指定に、文字列 ID は使えません。この文字列 IDを使っている機能を検索し、ディレクトリ識別子を使ってインストール先を変更してください。
-5040	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5039	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5038	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5037	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5036	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5033	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5025	パッケージを保存できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5024	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5019	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5018	論理ディスクに、機能が 1 つも含まれていません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5017	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5016	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5015	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5014	アイコン "%1" をビルド中にエラーが発生しました。指定されたアイコンキーがアイコンテーブルに見つかりませんでした。	このエラーについての情報は、ナレッジベース記事 Q105625 を参照してください。
-5013	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5011	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5010	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5009	スキーマ概要ストリームは、200 以上である必要があります。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5008	この 32 ビット パッケージを 64 ビット データに含めることはできません。64 ビット データは、マージ モジュールに含まれている可能性があります。	プロジェクトに 64 ビットの場所にインストールするように構成された 1 つ以上のファイル、フォルダー、またはレジストリ エントリが含まれていない場合、InstallShield がビルド時に 32 ビット .msi パッケージを作成しようとします。ただし、32 ビット .msi パッケージに 1 つ以上の 64 ビット コンポーネントを含む 64 ビット マージ モジュールを含めることはできません。 このビルドエラーを解決するために、次のオプションを考慮してください: ・ 64 ビット マージ モジュールを 32 ビット版に置換する。 ・ 64 ビット パッケージを生成するため、プロジェクトの 64 ビットの場所に 1 つ以上のファイル、フォルダー、またはレジストリ エントリを追加する。ただし、64 ビット パッケージを 32 ビット ターゲット システムで実行することはできません。 詳細については、「32 ビットと 64 ビットのターゲット システム両方をサポートする難しさについて」を参照してください。
-5007	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-5006	"%1" を保存できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5005	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5004	プロジェクトを書き込みアクセスで開けませんでした。	プロジェクトが IDE で既に開かれていないか確認してください。
-5003	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5002	内部ビルド エラー。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-5001	setup.ini をコピーできませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4701	One-Click Install Web ページをビルド中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4354	ビルドが、機能のファイルから COM 情報を抽出できませんでした。	ファイルが自動登録されること、および自動登録プロセスに問題がないか確認してください。
-4349	動的に抽出された COM 機能の %1 情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4348	動的に抽出された COM 機能の %1 情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4347	動的に抽出された COM 機能の情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4346	動的に抽出された COM 機能の情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4345	動的に抽出された COM 機能の情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4344	動的に抽出された COM 機能の情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4343	動的に抽出された COM 機能の情報のビルドに失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

テーブル 3-2・ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-4340	COM 抽出モジュールの初期化に失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4303	指定された機能でファイルキーを同期させる際、予期しないエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4302	機能 %2 でファイルキー %1 を同期させる際に、競合が発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4301	キー同期用の .msi ファイルが見つかりませんでした。	指定の MSI ファイルが存在することを確認してください。
-4092 を入力します。	MSI データベース %1 を開く際に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4075	ファイルが見つかりません。機能 '%2' のモジュール '%1' をマージ中にエラーが発生しました。	マージ モジュール検索パスにマージ モジュールが存在することを確認してください。
-4072	%1 の依存関係の取得中に、エラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-4006	ディレクトリの全ファイルの削除に失敗しました	ビルド出力先ディレクトリ (Disk1) 内の使用中のファイルを閉じます。 Msidb.exe が開いている場合は、閉じてください。
-3876	Summary Stream の不正なテンプレート %1 を無視しています。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-3713	関数ブロックは "モジュール::関数" の形式である必要があります。	関数ブロックを "モジュール::関数" の形式に変更してください。
-3114	アプリケーションパス %1 は、関連機能 %2のインストール先フォルダーを含んでいません。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。 アプリケーション パスはレジストリ エントリでもあるため、次のようなレジストリ パスが作成されていてパス エントリがない場合、このエラーが発生する可能性があります。 HKEY_LOCAL_MACHINE¥Microsoft¥Windows¥CurrentVersion¥App Paths これを訂正するには、パスエントリを設定します。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-3028	指定されたテーブルの文字列 ID の置換中に、エラーが発生しました。	指定されたテーブルで、必要な文字列 ID が空欄にされています。文字列 ID を指定してください。
-3016	Binary テーブル %1 をパッケージへ追加する際にエラーが発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-2200	ファイルを上書きできません。	指定のファイルが読み取り専用でないことを確認してください。
-1531 を入力します。	ディスクに指定されたサイズが、この機能には小さすぎます。	リリース ウィザードで、[メディアのフォーマット] 設定を大きなサイズに変更してください。
-1530	ディスクに指定されたサイズが、このファイルには小さすぎます。	リリース ウィザードで、[メディアのフォーマット] 設定を大きなサイズに変更してください。
-1527	プロジェクトにファイルが含まれていません。	この警告は、プロジェクト中に 1 つもファイルが入っていない場合に発生します。[ファイル] ビューからプロジェクトにファイルを追加すると、このエラーは発生しません。
-1505	CAB ファイルを MSI パッケージに追加できませんでした。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-1501	"%1" を "%2" に圧縮できませんでした。	IsCmdBld.exe が必ず相対ディレクトリで実行されるようにしてください。それにはエラーメッセージで指定されたファイルを見つけてそのディレクトリで IsCmdBld.exe を起動します。
-1027	%1 の署名に失敗しました。	現在のリリースに提供されているデジタル書名情報 (デジタルキーおよびパスワード) を確認してください。
-1024	ファイルが見つかりません。Binary テーブルにファイルをストリームできません。	指定したファイルが存在することを確認してください。
-1014 を入力します。	ディレクトリの名前を変更できません。	Windows エクスプローラーまたは DOS プロンプトがリリース出力フォルダー (Disk1) のサブフォルダーまたは Interm フォルダーをポイント中かロックしている可能性があります。現在のディレクトリを変更してください。Disk1 フォルダー内の開かれているファイルを閉じてください。Msidb.exe が開いている場合は、閉じてください。
-1013	指定されたファイルは、他のプログラムによって使われています。	現在このファイルを使っているアプリケーションを閉じて、ビルドプロセスを再実行してください。

テーブル 3-2 · ビルド エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-1009	ディスク容量が不足しているか、またはターゲットドライブが見つかりません。	ビルドターゲット上のディスク領域を増やすか、ビルドのターゲットを新規に選んでください。ターゲットドライブを指定できない場合は、新しいターゲットを選択するか、あるいはターゲットドライブのアクセス許可が正しく設定されているか確認してください。
-1001	MSI データベースを開く際にエラーが発生しました	MSI エンジンが破損している可能性があります。エンジンの修復、または再インストールが必要です。このエラーの情報について、ナレッジベース記事 Q105892 を調べるか、テクニカル サポート にご連絡ください。
-1000	製品構成が無効です。ディレクトリの作成に失敗しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。

アップグレード エラーと警告

次のテーブルは、プロジェクトを InstallShield の以前のバージョンから現在のバージョンにアップグレードしたときに発生する可能性のあるエラーと警告について説明します。また、問題解決のためのヒントも提供します。

テーブル 3-3 · アップグレードエラーと警告

エラー	説明	トラブルシューティングのヒント
-72	Config.sys はサポートしなくなりました。	Config.sys への変更はサポートされていません。これらの変更は移行されません。 ***メモ·*** *IInstallShield Program Files フォルダー¥Samples フォルダーに、Config.sys を変更するカスタム アクション例があります。*
-73	Autoexec.bat は現在サポートされていません。	Autoexec.bat への変更はサポートされていません。これらの変更は移行されません。 ***メモ·*** *環境変数は、[環境変数] ビュー から変更できます。*
-74	アンインストール アイコンは現在サポートされていません。	詳細については、「アンインストール ショートカットの作成」を参照してください。
-75	スプッシュ画面をサポートしなくなりました。	[スプラッシュ] スクリーンは [スプラッシュ] ダイアログに置換されました。

テーブル 3-3 • アップグレードエラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-77	ロゴ ビットマップは現在サポートされていません。	セットアップのこの部分は移行されません。
-78	背景色は現在サポートされていません。	セットアップのこの部分は移行されません。
-248	ファイルグループ %1 により、Windows NT3.51 がターゲット OS としてマークされています。NT4 へマイグレートします。	Windows Installer アーキテクチャでは NT3.51 のオペレーティング システムをサポートしていないため、指定されたファイルグループが NT3.51 機能として移行されない場合があります。作成された新しい機能は Windows NT4 に指定されます。
-249	ターゲットディレクトリ %1 を直接、移行できませんでした。機能 %3 のインストール先フォルダー %2 への移行	指定されたディレクトリを 1 つのまとまりとして移行できませんでした。現在、指定されたインストール先フォルダーに、移行されたターゲットディレクトリの内容が含まれています。
-250	不正なインストール先 <SRCDIR>のファイルグループを[INSTALLDIR]に変換しました。	INSTALLDIR は、すべてのセットアップのファイルに対するデフォルトのインストール先フォルダーです。
-251	<SRCDISK> はサポートされていません。インストール先を[INSTALLDIR]に変換しています。	SRCDISK は新しい InstallShield でサポートされていません。INSTALLDIR は、すべてのセットアップのファイルに対するデフォルトのインストール先フォルダーです。
-405	レジストリハイブ %1 はサポートされていません。このキーは無視されます。	指定されたレジストリハイブは Windows Installer でサポートされていません。したがって、そのハイブに含まれている情報は移行されませんでした。
-406	レジストリの種類 %1 はサポートされていません。このエントリは無視されます。	指定されたレジストリの種類は Windows Installer でサポートされていません。したがって、そのエントリに含まれている情報は移行されませんでした。

テーブル 3-3 • アップグレードエラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-407	古いプロジェクトでは、アンインストールのためのマークがこのレジストリ エントリに付いていませんでした。これは次のバージョンでアンインストールされます。	指定されたレジストリ エントリは、アンインストール時に、製品の残りの部分と一緒にアンインストールされます。
-535	再起動オプションは移行されません。「カスタム アクション」ビューで再起動アクションを追加してください。	インストール中のシステム再起動はカスタム アクションを通じて処理されます。詳細については、「カスタム アクション ギャラリー」を参照してください。
-536	個々のファイルグループの選択はサポートされていません。	ファイルグループが存在しないため、この設定は移行できません。
-537	Welcome ダイアログが必要です	この必須ダイアログはセットアップに追加されています。
-538	SetupComplete ダイアログが必要です	この必須ダイアログはセットアップに追加されています。
-541	Temp ファイルは現在サポートされていません	一時ファイルの使用は InstallShield でサポートされていません。これらのファイルは移行されません。[セットアップファイル] ビューを使用して、インストール中にターゲット システムで使用するファイルを追加します。
-542	Disk1 ファイルは現在サポートされていません。	これらの変更は移行されません。
-544	LicenseAgreement ダイアログの .txt ファイルは、Express 3.0 で使用できるよう .rtf ファイルに変換されました。	InstallShield では、LicenseAgreement ダイアログにリッチ テキスト (.rtf) ファイルが必要です。テキスト (.txt) ファイルは InstallShield によって .rtf に変換されました。
-545	ReadMeFileBrowser ダイアログの .txt ファイルは、Express 3.0 で使用できるよう .rtf ファイルに変換されました。	InstallShield では、ReadMeFileBrowser ダイアログにリッチ テキスト (.rtf) ファイルが必要です。テキスト (.txt) ファイルは InstallShield によって .rtf に変換されました。

テーブル 3-3 • アップグレードエラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
-601	拡張子 %1 を移行できません。	指定された拡張子を移行することができません。この拡張子を [ファイル拡張子] ビューでもう一度作成してください。
-602	最初に見つかったファイル %1 を拡張子 %2 のバイナリ ファイルとして使用します。	指定されたファイルは拡張子のバイナリに移行されました。

Windows Installer ランタイム エラー

次のテーブルは、インストールの実行時に発生する可能性のある Windows Installer エラーの一覧です。このエラーは、異なる機能をサポートするために InstallShield プロジェクトに自動的に追加されるビルトイン InstallShield カスタム アクションに関連しています。カスタム アクションの一覧は、「InstallShield カスタム アクション リファレンス」を参照してください。

ヒント: 一部のランタイム エラーについての詳細 (解決策を含む) は、ナレッジ ベースを参照してください。

テーブル 3-4 • Windows Installer ランタイム エラー

エラー番号	メッセージ	トラブルシューティング情報
27500	このセットアップには、IIS 仮想ルートの設定に IIS (Internet Information Server) が必要です。IIS がインストールされていることを確認してください。	このエラーは、プロジェクトに [IIS 構成] ビューで構成された IIS データが含まれている場合に発生します。 このエラーが発生した場合、ターゲット システムに IIS がインストールされていることを確認してください。
27501	このセットアップでは、IIS Virtual Roots の設定に管理者権限が必要です。	このエラーは、プロジェクトに [IIS 構成] ビューで構成された IIS データが含まれている場合に発生します。 このエラーは、IIS 仮想ディレクトリの設定に必要な管理者権限がない状態で、IIS 仮想ディレクトリを設定するインストールを実行したときに発生します。

テーブル 3-4・Windows Installer ランタイム エラー（続き）

エラー番号	メッセージ	トラブルシューティング情報
27508	COM+ アプリケーション [2] のインストール中にエラーが発生しました。[3]	このエラーは、プロジェクトに [コンポーネント サービス] ビューで構成された COM+ アプリケーションが含まれる場合に発生します。 このエラーは、COM+ コンポーネント DLL の依存関係が不足している場合、またはCOM+ アプリケーションの設定に誤りがある場合に発生します。このエラーを解決するためには、プロジェクトに依存関係を追加するか、プロジェクトに含まれる COM+ アプリケーションを確認して、必要な変更を加えてください。
27509	COM+ アプリケーション [2] のアンインストール中にエラーが発生しました。[3]	このエラーは、プロジェクトに [コンポーネント サービス] ビューで構成された COM+ アプリケーションが含まれる場合に発生します。 このエラーは、ランタイムがシステムから COM+ アプリケーションを削除するのに失敗した場合に発生します。
27510	COM+ アプリケーション [2] のインストール中にエラーが発生しました。Microsoft(R) .NET クラス ライブラリをロードできませんでした。.NET サービス コンポーネントの登録を行うためには、Microsoft(R) .NET Framework のインストールが必要です。	このエラーは、プロジェクトに [コンポーネント サービス] ビューで構成された COM+ アプリケーションが含まれる場合に発生します。 このエラーの発生を防ぐためには、プロジェクトへの .NET Framework の追加を考慮してください。詳しくは、「.NET Framework 再配布可能ファイルをプロジェクトへ追加する」をご覧ください。 その他、プロジェクト アシスタントの [要件] ページを使って、プロジェクトに .NET Framework 要件を追加することもできます。ターゲット システムに .NET Framework の適切なバージョンが搭載されていない場合、インストールはエラー メッセージを表示して製品のインストールを行いません。
27517	このインストールには、COM+ アプリケーションをインストールするための管理者権限が必要です。管理者としてログオンし、このインストールを再実行してください。	このエラーは、プロジェクトに [コンポーネント サービス] ビューで構成された COM+ アプリケーションが含まれる場合に発生します。 このエラーは、COM+ アプリケーションのインストールに必要な管理者権限がない状態で、COM+ アプリケーションを含むインストールを実行したときに発生します。

テーブル 3-4・Windows Installer ランタイム エラー (続き)

エラー番号	メッセージ	トラブルシューティング情報
27555	オブジェクト [2] にアクセス許可を適用しようとしたときに、エラーが発生しました。システム エラー: [3] ([4])	このエラーは、プロジェクトのファイル、フォルダ、またはレジストリ キーの保護にカスタム InstallShield 処理を使用した場合に発生する場合があります。この機能に関する詳細については、「ロックダウン環境におけるファイル、フォルダー、およびレジストリ キーのセキュリティ保護」を参照してください。

Setup.exe ランタイム エラーと警告

以下は、Setup.exe を実行した時に発生する可能性のあるエラーの一覧表です。Setup.exe を起動するために **CreateProcess()** を呼び出すときに、これらの戻り値をトラップすることができます。ただし、Setup.exe が MsiExec.exe からのエラー コードを戻すことはありません。

メモ: *エラーが発生した時に表示される文字列は、ご使用のオペレーティング システムが英語版ではないときは、英語で表示されない場合があります。*

テーブル 3-5・Setup.exe ランタイム エラーと警告

エラー	説明	トラブルシューティングのヒント
-1	セットアップがユーザーによってキャンセルされました	Esc キーまたは [キャンセル] ボタンを押さずにセットアップをもう一度実行してください。
1150	セットアップが互換性のない Windows のバージョンを検出しました。[OK] をクリックし、Windows 95、Windows NT 4.0、またはそれ以降のバージョンでセットアップを再起動してください。	Windows Installer は Windows NT 4.0 またはそれ以降、および Windows 9x と互換性があります。ご使用の Windows のバージョンを確認し、必要な場合は互換性のあるバージョンにアップグレードしてください。
1151	一時ディレクトリへの書き込みエラー	一時ディレクトリへ書き込むには、環境変数 TEMP が設定されている必要があります。一時フォルダーが存在していて、セットアップを格納するのに十分なディスク容量があることを確認してください。一時フォルダーにファイルがある場合は、それらを削除して Setup.exe を再実行してください。
1152	一時ディレクトリへの <ファイル名> の抽出エラー	Temp フォルダーが書き込み可能であることを確認してください (上のエラーを参照)。Temp フォルダーが書き込み可能な場合は、セットアップ中のファイルが壊れている可能性があります。壊れているファイルがないことを確認し、Setup.exe を再実行してください。

テーブル 3-5・Setup.exe ランタイム エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
1153	セットアップ初期化ファイルの読み取りエラー	Setup.ini ファイル は、Setup.exe と同じフォルダーに配置しなくてはなりません。そうでない場合、適切な場所に Setup.ini を移動させます。
1154	インストーラーが <パス> で見つかりません。	Windows Installer が正しくインストールされていないか、古いバージョンが使用されている可能性があります。必要に応じて再インストールします。
1155	ファイル <ファイル名> が見つかりません。	.msi ファイルがあることを確認してください。.msi ファイルがある場合は、それが Setup.exe と同じフォルダー内にあることを確認してください。
1157	MsiExec.exe の起動に失敗しました	ターゲット プラットフォームにWindows Installer の正しいバージョンを配布していることを確認してください。 MsiExec.exe コマンドライン引数の構文を確認してください。
1158	文字列の作成エラー	Setup.ini の文字列がすべて有効であることを確認してください。詳細は InstallShield 使用許諾契約書を参照してください。
1201	セットアップは、<フォルダー> に <容量> KB の空き容量が必要です。空き容量を増やして再度実行してください。	ターゲット位置のディスク容量が不充分です。セットアップをインストールするドライブに 10 MB 以上の空き容量があることを確認してください。
1202	このインストールを完了するための十分な権限がマシンのすべてのユーザーにありません。管理者としてログオンし、このインストールを再実行してください。	このインストールを完了するためには、管理者権限が必要です。
1208	<言語> の ANSI コードページがシステムにインストールされていないので、選択された言語でセットアップを実行できません。セットアップを実行し、他の言語を選択してください。	セットアップを実行し、他の言語を選択してください。

テーブル 3-5・Setup.exe ランタイム エラーと警告 (続き)

エラー	説明	トラブルシューティングのヒント
1603	一般的な Windows Installer エンジンのエラーSetup.ini に必要なディスク容量を増やし、再度実行してください。	InstMsiW.exe (Windows NT および 2000 用) または InstMsiA.exe (Windows 9x 用) を実行して Windows Installer をもう 1 度インストールしてください。
1614	ファイルをダウンロード中にエラーが発生しました	ターゲットマシンの接続状況とディスク容量をチェックして、ダウンロードを再試行してください。ターゲットマシンに Internet Explorer 3,02 以降がインストールされていることを確認してください。
1627	ファイルを保存できません	指定のファイルを保存できませんでした。ターゲットマシンに対して十分な権限がない、接続が失われた、スペース不足など、さまざまな理由が考えられます。ターゲットディレクトリにアクセスできることを確認してください。

Visual Studio プロジェクトのインポート エラーと警告

このテーブルは、以下のどちらかの処理を行ったときに発生する可能性のあるエラーと警告をリストします:

- Visual Studio セットアッププロジェクト (.vdproj) を InstallShield プロジェクト (.ise) に変換。

- Visual Studio セットアップまたはマージ モジュール プロジェクト (.vdproj) を InstallShield プロジェクト (.ise) にインポート。

テーブル 3-6 • Visual Studio プロジェクトのインポート エラーと警告

エラー/警告番号	説明	トラブルシューティングのヒント
-9000	不明な例外が発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-9001	不明な COM 例外が発生しました。	このエラーの情報について ナレッジ ベースを調べるか、テクニカル サポートにご連絡ください。
-9002	プロジェクト %1 をロード中にエラーが発生しました。	このエラーは、Visual Studio セットアップまたはマージ モジュール プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9003	プロジェクト %1 を作成中にエラーが発生しました。	このエラーは、InstallShield が Visual Studio セットアップ プロジェクトから Express プロジェクトを作成できないときに発生します。InstallShield は、プロジェクトを Visual Studio プロジェクト ファイル (.vdproj) を含んでいるフォルダーに作成しようとします。 このエラーを解決するには、Visual Studio プロジェクト ファイルを含むフォルダーが読み取り専用になっていないかどうかを確認し、読み取り専用の場合、それを書き込み可能にします。
-9004	プロジェクト %1 を保存中にエラーが発生しました。	このエラーは、InstallShield が Visual Studio セットアップ プロジェクトから Express プロジェクトを作成できないときに発生します。InstallShield は、プロジェクトを Visual Studio プロジェクト ファイル (.vdproj) を含んでいるフォルダーに保存しようとします。 このエラーを解決するには、Visual Studio プロジェクト ファイルを含むフォルダーが読み取り専用になっていないかどうかを確認し、読み取り専用の場合、それを書き込み可能にします。
-9005	プロジェクトを変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9006	プロジェクトの種類を判別中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9007	セクション '%1' を解析中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9008	セクション %1 (%2 の下) に開き大かっこがありません。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9009	セクション %1 (%2 の下) に閉じ大かっこがありません。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9010	無効なエントリ %1 が見つかりました (セクション %1 (%2 の下))。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。
-9011	プロパティ %1 をセクション %2 に追加中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9012	セクション %1 をセクション %2 に追加中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9013	セクション % 1が存在しません。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9014	製品のプロパティを変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9015	ファイルを変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9016	ファイルの種類 %1 が無効です。ファイル %2 は変換されませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9017	ファイル %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9018	機能を変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9019	機能 %1 を変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9020	フォルダーを変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9021	フォルダー %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9022	フォルダーの種類 %1 が無効です。フォルダー %2 は変換されませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9023	カスタム アクションを変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9024	カスタム アクション %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9025	カスタム アクションの種類 %1 が無効です。カスタム アクション %2 は変換されませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9026	ソース ファイル %1 (カスタム アクション %2) のコンポーネントが見つかりませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9027	ファイルの種類を変換中にエラーが発生しました。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9028	ファイル拡張子 %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9029	ファイル拡張子の種類 %1 が無効です。ファイル拡張子 %2 は変換されませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9030	動詞 %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9031	動詞の種類 %1 が無効です。動詞 %2 は変換されませんでした。	このエラーは、Visual Studio プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9032	ファイル 拡張子 %1 のコマンドが指定されていません。ファイル拡張子は変換されませんでした。	この警告は、Visual Studio のファイルの種類エディターで指定されたファイルの種類の Command プロパティが空白のまま残されたとき発生します。この警告は、変換処理中に InstallShield がファイル拡張子を変換できなかったことをアラートします。 この問題を解決するには、Visual Studio でファイル タイプのコマンドを指定してから、Visual Studio セットアップ プロジェクトを InstallShield プロジェクトに変換します。そうでない場合、InstallShield の [ファイル拡張子] ビューでファイル拡張子を追加および構成することができます。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9033	コマンド %1 (ファイル拡張子 %2) のコンポーネントが見つかりませんでした。ファイル拡張子は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9034	レジストリを変換中にエラーが発生しました。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9035	レジストリ キー %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9036	レジストリ キーの種類 %1 が無効です。レジストリ キー %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9037	レジストリ値 %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9038	レジストリ値の種類 %1 が無効です。レジストリ値 %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9039	ショートカットを変換中にエラーが発生しました。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9040	ショートカット %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9041	ショートカットの種類 %1 が無効です。ショートカット %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9042	ショートカットのターゲット %1 が無効です。ショートカット %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9043	起動条件を変換中にエラーが発生しました。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9044	起動条件 %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9045	起動条件の種類 %1 が無効です。起動条件 %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9046	ロケータを変換中にエラーが発生しました。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9047	ロケーター %1 を変換中にエラーが発生しました。種類: %2	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9048	ロケータの種類 %1 が無効です。ロケータ %2 は変換されませんでした。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9049	製品のアイコンを変換中にエラーが発生しました。	このエラーは、Visual Studio セットアップ プロジェクト ファイルが破損している場合、または InstallShield が Visual Studio セットアップ プロジェクト ファイルを正しく読み込めないときに発生する可能性があります。このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9050	プロジェクトに言語 '%1' は InstallShield に含まれていません。言語は変換されませんでした。	この警告は、言語のサポートを含む Visual Studio プロジェクトを変換するとき、InstallShield にその言語のビルトイン サポートがないとき発生します。
-9052	ファイルの種類 '%1' は拡張子が指定されていません。ファイルの種類は変換されませんでした。	この警告は、Visual Studio のファイルの種類エディターで指定されたファイルの種類の Extensions プロパティが空白のまま残されたとき発生します。この警告は、インポートまたは変換処理中に InstallShield がファイル拡張子を構成できなかったことをアラートします。 この問題を解決するには、Visual Studio でファイルの種類に 1 つ以上のファイル拡張子を指定してから、Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートします。これ以外の方法としては、Visual Studio プロジェクトを変換またはインポートした後に InstallShield の [ファイルの拡張子] ビューでファイル拡張子を追加または構成できます。

テーブル 3-6 ・ Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9053	起動条件 '%1' で条件が指定されていません。起動条件は変換されませんでした。	この警告は、Visual Studio の起動条件エディターで指定された起動条件の Condition プロパティが空白のまま残されたとき発生します。この警告は、インポートまたは変換処理中に InstallShield が起動条件を構成できなかったことをアラートします。 この問題を解決するには、Visual Studio で条件を指定してから、Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートします。これ以外の方法としては、Visual Studio プロジェクトを変換またはインポートした後に InstallShield の [要件] ビューで起動条件を追加または構成できます。
-9054	ファイル検索 '%1' のファイル名が指定されていません。ファイル検索は変換されませんでした。	この警告は、Visual Studio の起動条件エディターで指定されたファイル検索の FileName プロパティが空白のまま残されたとき発生します。この警告は、インポートまたは変換処理中に InstallShield がファイル検索を構成できなかったことをアラートします。 この問題を解決するには、Visual Studio でファイル検索のファイル名を指定してから、Visual Studio プロジェクトを InstallShield プロジェクトに変換またはインポートします。これ以外の方法としては、Visual Studio プロジェクトを変換またはインポートした後に InstallShield の [要件] ビューでファイル検索処理を追加または構成できます。
-9055	機能 '%1' は、InstallShield プロジェクトに既に存在します。機能は変換されませんでした。	***エディション:*** *この警告は InstallShield Premier および InstallShield で発生する場合があります。* *この警告は Express Edition には適用しません。Express Edition では、すべてのアプリケーション データが Visual Studio プロジェクトから InstallShield プロジェクトの [すべてインストール] 機能に追加されます。* この警告は、Visual Studio プロジェクトに特定の機能が含まれていて、同じ機能が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9056	ディレクトリ '%1' は、InstallShield プロジェクトに既に存在します。フォルダー '%2' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のフォルダーが含まれていて、同じフォルダーが既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[ファイル] ビューをチェックして、警告メッセージに示されたフォルダーが InstallShield プロジェクトから不足していないかどうか確認してください。
-9057	ファイル キー '%1' は、InstallShield プロジェクトに既に存在します。ファイル '%2' は変換されませんでした。	この警告は、特定のファイル キーと関連付けられているファイルを含む Visual Studio プロジェクトを、同じファイル キーを持つファイルを既に含む InstallShield プロジェクトにインポートしたときに発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[ファイル] ビューをチェックして、警告メッセージに示されたファイルが InstallShield プロジェクトから不足していないかどうか確認してください。 ***エディション・*** *InstallShield Premier および InstallShield には [ダイレクト エディター] ビューがあり、プロジェクトに含まれる各ファイルに関連付けられたファイル キーが表示されます。ファイル キーは、.msi テーブルの* File *テーブルのプライマリ キーです。*File *テーブルは、重複するファイル キーを持つことができません。* *InstallShield Premier および InstallShieldでは、[コンポーネント] ビューの [ファイル] 領域でファイル キー情報も参照できます。*

テーブル 3-6 ・ Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9058	ショートカット キー '%1' は、InstallShield プロジェクトに既に存在します。ショートカット '%2' は変換されませんでした。	この警告は、特定のショートカット キーと関連付けられているショートカットを含む Visual Studio プロジェクトを、同じショートカット キーを持つショートカットを既に含む InstallShield プロジェクトにインポートしたときに発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[ショートカット/フォルダー] ビューをチェックして、警告メッセージに示されたショートカットが InstallShield プロジェクトから不足していないかどうか確認してください。 ***エディション・*** *InstallShield Premier および InstallShield には [ダイレクト エディター] ビューがあり、プロジェクトに含まれる各ショートカットに関連付けられたショートカット キーが表示されます。ショートカット キーは、.msi テーブルの Shortcut テーブルのプライマリ キーです。Shortcut テーブルは、重複するショートカット キーを持つことができません。*
-9059	拡張子 '%1' は、InstallShield プロジェクトに既に存在します。ファイルの種類 '%2' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定の拡張子が含まれていて、同じ拡張子が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[ファイル拡張子] ビューをチェックして、警告メッセージに示されたファイル拡張子が InstallShield プロジェクトから不足していないかどうか確認してください。
-9060	レジストリ キー '%1' は、InstallShield プロジェクトに既に存在します。レジストリ エントリ 'キー: %2 値: %3 データ: %4' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のレジストリ エントリが含まれていて、同じレジストリ エントリが既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[レジストリ] ビューをチェックして、警告メッセージに示されたレジストリ エントリが InstallShield プロジェクトから不足していないかどうか確認してください。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9061	カスタム アクション '%1' は、InstallShield プロジェクトに既に存在します。カスタム アクション %2 は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のカスタム アクションが含まれていて、同じ名前を持つカスタム アクションが既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[カスタム アクション] ビューをチェックして、警告メッセージに示されたカスタム アクションが InstallShield プロジェクトから不足していないかどうか確認してください。
-9062	起動条件 '%1' は、InstallShield プロジェクトに既に存在します。起動条件 %2 は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定の起動条件が含まれていて、同じ名前を持つ起動条件が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[要件] ビューをチェックして、警告メッセージに示された起動条件が InstallShield プロジェクトから不足していないかどうか確認してください。
-9063	RegLocator '%1' は、InstallShield プロジェクトに既に存在します。レジストリ検索 '%2' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のレジストリ検索が含まれていて、同じ名前を持つレジストリ検索が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[要件] ビューをチェックして、警告メッセージに示されたレジストリ検索が InstallShield プロジェクトから不足していないかどうか確認してください。
-9064	DrLocator '%1' は、InstallShield プロジェクトに既に存在します。ファイル検索 '%2' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のファイルまたはフォルダー検索が含まれていて、同じ名前を持つファイルまたはフォルダー検索が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[要件] ビューをチェックして、警告メッセージに示されたファイルまたはフォルダー検索が InstallShield プロジェクトから不足していないかどうか確認してください。

テーブル 3-6 ・ Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9065	CompLocator '%1' は、InstallShield プロジェクトに既に存在します。Windows Installer 検索 '%2' は変換されませんでした。	この警告は、Visual Studio プロジェクトに特定のファイルまたはフォルダー検索が含まれていて、同じ名前を持つファイルまたはフォルダー検索が既に含まれている InstallShield プロジェクトにその Visual Studio プロジェクトをインポートした場合に発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[要件] ビューをチェックして、警告メッセージに示されたファイルまたはフォルダー検索が InstallShield プロジェクトから不足していないかどうか確認してください。
-9069	64 ビットの場所を解決する Windows Installer プロパティは、ご使用中の InstallShield のエディションではサポートされていません。フォルダー '%1' は '%2' に変換されました。	***エディション:*** *64 ビットの場所を解決する Windows Installer プロパティのサポートは、以下の InstallShield のエディションで利用できます:* - *InstallShield Premier* - *InstallShield* この警告は、Visual Studio プロジェクト内の DefaultLocation プロパティが 64 ビットの場所 (**[ProgramFiles64Folder]**、**[CommonFiles64Folder]**、または **[System64Folder]**) を使用する場合に、Visual Studio デプロイメント プロジェクト インポート ウィザードで Visual Studio プロジェクトから INSTALLDIR 値をインポートすることを指定したときに発生します。この警告が発生した場合、InstallShield Express Edition は対応する 32 ビットの場所を使用します (例、**[ProgramFilesFolder]**)。InstallShield Premier および InstallShield は 64 ビットの場所を使用します。 インストールで 64 ビット サポートが必要な場合、InstallShield Premier または InstallShield へのアップグレードを考慮してください。これらのエディションでは、64 ビット パッケージの作成、64 ビットの場所へのインストール、およびその他の 64 ビット関連の設定を構成することができます。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9070	前提条件を変換中にエラーが発生しました。	***エディション・*** *InstallShield Premier Edition および InstallShield には、InstallShield 前提条件エディターが含まれています。InstallShield 前提条件エディターを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。これらの InstallShield のエディションの 1 つをご利用の場合、製品のインストール中に Microsoft 再配布可能ファイルのインストールを行う独自の InstallShield 前提条件を作成することができます。* このエラーは、前提条件を含む Visual Studio セットアップ プロジェクトをインポートするときに、InstallShield が Visual Studio 前提条件を対応する InstallShield 前提条件にマップしようとして問題が起こった場合に発生します。 このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9071	InstallShield には、'%1' に対応する InstallShield 前提条件がありません。前提条件は変換されませんでした。	***エディション・*** *InstallShield Premier Edition および InstallShield には、InstallShield 前提条件エディターが含まれています。InstallShield 前提条件エディターを利用して、カスタム InstallShield 前提条件を定義したり、既存の InstallShield 前提条件の設定を編集したりできます。これらの InstallShield のエディションの 1 つをご利用の場合、製品のインストール中に Microsoft 再配布可能ファイルのインストールを行う独自の InstallShield 前提条件を作成することができます。* この警告は、前提条件を含む Visual Studio セットアップ プロジェクトをインポートするときに、InstallShield で対応する InstallShield 前提条件がない場合に発生します。 この警告を解決するには、InstallShield Premier および InstallShield へのアップグレードを考慮してください。これらのエディションでは、独自の InstallShield 前提条件を作成して、InstallShield インストール プロジェクトに追加することができます。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9072	起動条件が、ご使用中の InstallShield のエディションでサポートされていません。起動条件 %1 は変換されませんでした。	***エディション**・次のディションでは、起動条件を作成することができます。* ・ *InstallShield Premier* ・ *InstallShield* *ターゲット システム上の起動条件のうち 1 つ以上が False 評価された場合、インストールは終了してエラーメッセージが表示されます。* この警告は、Visual Studio プロジェクトに起動条件が含まれていて、そのプロジェクトを InstallShield プロジェクトにインポートしようとした場合に発生します。 この警告を解決するには、InstallShield Premier または InstallShield へのアップグレードを考慮してください。
-9073	Windows Installer 検索は、ご使用中の InstallShield のエディションでサポートされていません。Windows Installer 検索 '%1' は変換されませんでした。	***エディション**・以下の InstallShield のエディションに含まれるシステム検索ウィザードを使って、Windows Installer 検索を作成できます。* ・ *InstallShield Premier* ・ *InstallShield* この警告は、Visual Studio プロジェクトにターゲット システム上で特定のコンポーネント ID の存在を確認する Windows Installer 検索が含まれている場合に、そのプロジェクトを InstallShield プロジェクトにインポートしようとすると発生します。 この警告を解決するには、この種類のターゲット システム検索をサポートする InstallShield Premier または InstallShield へのアップグレードを考慮してください。
-9074	プロジェクト出力を変換中にエラーが発生しました。	このエラーは、1 つ以上のプロジェクト出力を含む Visual Studio セットアップまたはマージ モジュール プロジェクトを変換するとき、InstallShield が プロジェクト出力を InstallShield プロジェクトに組み込むときに問題が発生した場合に起こります。 このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9075	プロジェクト出力 %1 を変換中にエラーが発生しました。種類: %2	このエラーは、1 つ以上のプロジェクト出力を含む Visual Studio セットアップまたはマージ モジュール プロジェクトを変換するとき、InstallShield が プロジェクト出力を InstallShield プロジェクトに組み込むときに問題が発生した場合に起こります。 このエラーを解決するには、テクニカル サポート にご連絡ください。。
-9076	プロジェクト出力の種類 %1 が無効です。プロジェクト出力 %2 は変換されませんでした。	このエラーは、1 つ以上のプロジェクト出力を含む Visual Studio セットアップまたはマージ モジュール プロジェクトを変換するとき、プロジェクト出力の 1 つが無効な場合に起こります。 このエラーを解決するためには、Visual Studio プロジェクトのプロジェクト出力を確認して、問題をすべて解決してください。その後、Visual Studio プロジェクトを InstallShield プロジェクトに変換します。
-9077	ファイル キー '%1' は、InstallShield プロジェクトに既に存在します。プロジェクト出力 '%2' は変換されませんでした。	この警告は、特定のファイル キーと関連付けられているファイル持つプロジェクト出力を含む Visual Studio プロジェクトを、同じファイル キーを持つファイルを既に含む InstallShield プロジェクトにインポートしたときに発生します。この状況は、同じ Visual Studio プロジェクトを InstallShield プロジェクトに複数回にわたってインポートする場合に起こります。 この警告が発生した場合、[ファイル] ビューをチェックして、警告メッセージに示されたファイルが InstallShield プロジェクトから不足していないかどうか確認してください。 ***エディション・*** *InstallShield Premier および InstallShield には [ダイレクト エディター] ビューがあり、プロジェクトに含まれる各ファイルに関連付けられたファイル キーが表示されます。ファイル キーは、.msi テーブルの File テーブルのプライマリ キーです。File テーブルは、重複するファイル キーを持つことができません。* *InstallShield Premier および InstallShield では、[コンポーネント] ビューの [ファイル] 領域でファイル キー情報も参照できます。*

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9078	InstallShield プロジェクトが Visual Studio ソリューションにありません。プロジェクト出力 '%1' は変換されませんでした。	このエラーは、InstallShield プロジェクトが InstallShield で開かれていて、Visual Studio 内部から開かれていない場合に、プロジェクト出力を含む Visual Studio プロジェクトを InstallShield プロジェクトにインポートしようとすると発生します。 このエラーを解決するには、InstallShield プロジェクトを Visual Studio 内部から開いて、そこに Visual Studio プロジェクトをインポートします。
-9079	現在の Visual Studio ソリューション内で Visual Studio プロジェクト '%1' が見つかりませんでした。プロジェクト出力 '%2' は変換されませんでした。	プロジェクト出力含む Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートする場合、その Visual Studio プロジェクトは、そのプロジェクトのすべての依存関係と同じソリューション内になくてはなりません。そうでない場合、Visual Studio プロジェクトをインポートしようとしたときに、このエラーが発生します。 このエラーを解決するには、インポートする Visual Studio セットアップまたはマージ モジュール プロジェクトを含んでいるのと同じ Visual Studio ソリューションに InstallShield プロジェクトがあることを確認してください。また、そのソリューションに Visual Studio セットアップまたはマージ モジュール プロジェクトの依存関係であるその他の Visual Studio プロジェクトのすべてが含まれていることも確認してください。確認が終わってから、Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートできます。
-9080	プロジェクト出力グループ '%1' が Visual Studio プロジェクト '%2' 中に見つかりませんでした。プロジェクト出力 '%3' は変換されませんでした。	このエラーは、プロジェクト出力を含む Visual Studio セットアップまたはマージ モジュール プロジェクトを変換する際に、InstallShield がそのグループのプロジェクト出力を InstallShield プロジェクトに組み込むときに問題が発生した場合に起こります。 このエラーを解決するには、テクニカル サポート にご連絡ください。。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9081	ファイル '%1' は、セットアップまたはマージ モジュールから除外するように指定さています。ファイルは変換されませんでした。	この警告は、Exclude プロパティが True に設定されているファイルを含む Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートする場合に発生します。 指定されたファイルを InstallShield プロジェクトから除外する場合は、この警告を無視してください。 指定されたファイルを InstallShield プロジェクトに含める場合は、InstallShield の [ファイルとフォルダー] ビューを使ってこれを追加できます。詳細については、「ファイルとフォルダーをプロジェクトに追加する」を参照してください。
-9082	プロジェクト出力 '%1' は、セットアップまたはマージ モジュールから除外するように指定さています。プロジェクト出力は変換されませんでした。	この警告は、プロジェクト出力にファイルを含む Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートする場合で、そのファイルの Exclude プロパティに True が選択されているときに発生します。 指定されたファイルを InstallShield プロジェクトから除外する場合は、この警告を無視してください。 指定されたファイルを InstallShield プロジェクトに含める場合は、InstallShield の [ファイルとフォルダー] ビューを使ってこれを追加できます。詳細については、「Visual Studio ソリューションにリファレンスを追加する」を参照してください。
-9083	Visual Studio のデフォルト起動条件は、InstallShield でサポートされていません。起動条件 %1 は変換されませんでした。	この警告は、デフォルト起動条件を含む Visual Studio セットアップまたはマージ モジュール プロジェクトを InstallShield プロジェクトにインポートしたときに発生します。 指定された起動条件を InstallShield プロジェクトから除外する場合は、この警告を無視してください。 Visual Studio が元のプロジェクトに追加したデフォルト起動条件をターゲット システム上でチェックする起動条件を InstallShield プロジェクトに追加するには、InstallShield の [要件] ビューを使います。詳細については、「製品のソフトウェアの要件を指定する」を参照してください。

テーブル 3-6 ・ Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9084	64 ビットの場所を解決する Windows Installer プロパティは、ご使用中の InstallShield のエディションではサポートされていません。'%2' という名前のファイル検索の Folder プロパティ '%1' が '%3' に変換されました	***エディション・** 64 ビットの場所を解決する Windows Installer プロパティのサポートは、以下の InstallShield のエディションで利用できます:* - *InstallShield Premier* - *InstallShield* この警告は、Visual Studio の起動条件エディタで指定されたファイル検索の Folder プロパティに 64 ビット フォルダの場所が含まれている場合に発生します。この警告が発生した場合、InstallShield Express Edition は対応する 32 ビットの場所を使用します (例、**[ProgramFilesFolder]**)。InstallShield Premier および InstallShield は 64 ビットの場所を使用します。 インストールで 64 ビット サポートが必要な場合、InstallShield Premier または InstallShield へのアップグレードを考慮してください。これらのエディションでは、64 ビット パッケージの作成、64 ビットの場所へのインストール、およびその他の 64 ビット関連の設定を構成することができます。
-9085	'%2' という名前のカスタム アクションの Condition '%1' は、InstallShield のこのエディションではサポートされていません。Condition プロパティは変換されませんでした。	***エディション・** InstallShield の以下のエディションでは、カスタム アクションの詳細条件の作成がサポートされています。* - *InstallShield Premier* - *InstallShield* この警告は、Visual Studio プロジェクトを InstallShield プロジェクトにインポートしようとしたときに、Visual Studio プロジェクトに使用中の InstallShield のエディションではサポートされていない種類の条件を持つカスタム アクションが含まれている場合に発生します。 この警告を解決するためには、InstallShield の [カスタム アクション] ビューを使って Visual Studio プロジェクトを変換またはインポートした後にカスタム アクションに条件を追加します。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9086	'%2' と名づけられたファイル検索の MinDate '%1' が無効です。MinDate プロパティは変換されませんでした。	この警告は、MinDate プロパティに無効な値を持つファイル検索を含む Visual Studio セットアップをインポートまたは変換しようとしたときに発生します。 この警告を解決するには、以下の手順に従います: 1. InstallShield の **[要件]** ビューを開きます。 2. ファイル検索を右クリックしてから、**[起動条件の変更]** をクリックします。**システム検索ウィザード**が開きます。 3. このウィザードの 2 番目のパネルで、**[詳細]** ボタンをクリックします。**[ファイルの詳細]** ダイアログ ボックスが開きます。 4. 適切な詳細を入力します。
-9087	'%2' と名づけられたファイル検索の MaxDate '%1' が無効です。MaxDate プロパティは変換されませんでした。	この警告は、MaxDate e プロパティに無効な値を持つファイル検索を含む Visual Studio セットアップをインポートまたは変換しようとしたときに発生します。 この警告を解決するには、以下の手順に従います: 1. InstallShield の **[要件]** ビューを開きます。 2. ファイル検索を右クリックしてから、**[起動条件の変更]** をクリックします。**システム検索ウィザード**が開きます。 3. このウィザードの 2 番目のパネルで、**[詳細]** ボタンをクリックします。**[ファイルの詳細]** ダイアログ ボックスが開きます。 4. 適切な詳細を入力します。
-9088	InstallShield では Visual Studio Web セットアップ プロジェクトを InstallShield プロジェクトに変換することはできません。	このエラーは、Visual Studio Web セットアップ プロジェクトを InstallShield にインポートまたは変換しようとしたときに発生します。InstallShield は、Visual Studio Web セットアップ プロジェクトの変換をサポートしません。この種類のエラーが発生した場合、InstallShield で新しいプロジェクトを作成し、[IIS 構成] ビューを使ってターゲット システム上の IIS Web サイトを管理する方法を考慮してください。
-9089	InstallShield では Visual Studio CAB プロジェクトを InstallShield プロジェクトに変換することはできません。	このエラーは、Visual Studio CAB セットアップ プロジェクトを InstallShield にインポートまたは変換しようとしたときに発生します。InstallShield は、Visual Studio CAB セットアップ プロジェクトの変換をサポートしません。

テーブル 3-6・Visual Studio プロジェクトのインポート エラーと警告 (続き)

エラー/警告番号	説明	トラブルシューティングのヒント
-9090	Visual Studio プロジェクトの種類を識別できませんでした。このため、プロジェクトは変換されませんでした。	このエラーは、InstallShield がその種類を識別できない Visual Studio プロジェクトをインポートまたは変換しようとしたときに発生します。
-10000	ユーザーにより変換がキャンセルされました。	このエラーは、変換処理を途中でキャンセルしたときに発生します。

InstallShield カスタム アクション リファレンス

このセクションでは、様々な機能をサポートするために InstallShield プロジェクトに自動的に追加されるビルトイン InstallShield カスタム アクションそれぞれについて説明します。

_serial_verifyCA_isx

SERIALNUMVALRETRYLIMIT プロパティの値を 1 減らします。

_serial_verifyCA_isx_helper

SERIALNUMVALRETRYLIMIT プロパティの値を 1 減らします。

CheckForProductUpdates

FlexNet Connect を使用して、製品のアップデートを確認します。

カスタム アクションは Agent.exe という名前の実行可能ファイルを起動して、次の内容を渡します。

/au[ProductCode] /EndOfInstall

CheckForProductUpdatesOnReboot

FlexNet Connect を使用して、起動時に製品のアップデートを確認します。

カスタム アクションは Agent.exe という名前の実行可能ファイルを起動して、次の内容を渡します。

/au[ProductCode] /EndOfInstall /Reboot

DLLWrapCleanup

抽出されたデータをクリーンアップする標準 DLL ラッパー。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は dllwrap.dll で、そのエントリ ポイントは DLLWrapCleanup です。

DLLWrapStartup

呼び出しを説明するデータを抽出する標準 DLL ラッパー。

これは、.dll カスタム アクションです。.dll ファイルの名前は dllwrap.dll で、そのエントリ ポイントは DLLWrapStartup です。

ISComponentServiceCosting

ISComPlusApplication テーブルから情報を抽出し、COM+ アプリケーションの一時ファイルに保存します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は iscomsrv.dll で、そのエントリ ポイントは ISComponentServiceCosting です。

ISComponentServiceFinalize

インストールおよびアンインストール時に、COM+ アプリケーションをコミットします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は iscomsrv.dll で、そのエントリ ポイントは ISComponentServiceFinalize です。

ISComponentServiceInstall

インストール時に、COM+ アプリケーションをインストールします。

これは、.dll カスタム アクションです。.dll ファイルの名前は iscomsrv.dll で、そのエントリ ポイントは ISComponentServiceInstall です。

ISComponentServiceRollback

インストールまたはアンインストールが失敗したとき、COM+ アプリケーションをロールバックします。

これは、.dll カスタム アクションです。.dll ファイルの名前は iscomsrv.dll で、そのエントリ ポイントは ISComponentServiceRollback です。

ISComponentServiceUninstall

アンインストール時に、COM+ アプリケーションを削除します。

これは、.dll カスタム アクションです。.dll ファイルの名前は iscomsrv.dll で、そのエントリ ポイントは ISComponentServiceUninstall です。

ISIISCleanup

IIS インストールの一時ファイルとレジストリ エントリを削除します。

これは、.dll カスタム アクションです。.dll ファイルの名前は IISHelper.dll で、そのエントリ ポイントは ISIISCleanup です。

ISIISCosting

IIS インストールの一時ファイルにアクション一覧を作成します。他の IIS アクションの CustomActionData プロパティを設定します。

これは、.dll カスタム アクションです。.dll ファイルの名前は IISHelper.dll で、そのエントリ ポイントは ISIISCosting です。

ISIISInstall

IIS インストールに Web サイト、アプリケーション、仮想ディレクトリ、および他のアイテムを作成します。

これは、.dll カスタム アクションです。.dll ファイルの名前は IISHelper.dll で、そのエントリ ポイントは ISIISInstall です。

ISIISRollback

IIS のロールバック中に Web サイト、アプリケーション、仮想ディレクトリ、および他のアイテムを削除します。

これは、.dll カスタム アクションです。.dll ファイルの名前は IISHelper.dll で、そのエントリ ポイントは ISIISRollback です。

ISIISUninstall

IIS のアンインストール中に Web サイト、アプリケーション、仮想ディレクトリ、および他のアイテムを削除します。

これは、.dll カスタム アクションです。.dll ファイルの名前は IISHelper.dll で、そのエントリ ポイントは ISIISUninstall です。

ISInstallPrerequisites

機能と関連付けられた前提条件のインストールを起動します。

これは、.dll カスタム アクションです。.dll ファイルの名前は PrqLaunch.dll で、そのエントリ ポイントは InstallPrerequisites です。

ISJITCompileActionAtInstall

インストール時に、.NET アセンブリをプリコンパイルします。

このカスタム アクションは、ngen.exe という名前の Microsoft 実行可能ファイルを起動します。

ISJITCompileActionAtUnInstall

アンインストール時にプリコンパイル済み .NET アセンブリを削除します。

このカスタム アクションは、ngen.exe という名前の Microsoft 実行可能ファイルを起動します。

ISLockPermissionsCost

ISLockPermissionsInstall アクションの CustomActionData プロパティを設定します。

これは、.dll カスタム アクションです。.dll ファイルの名前は ISLockPermissions.dll で、そのエントリ ポイントは ISLockPermissionsCostAction です。

ISLockPermissionsInstall

製品がインストールされるときのアクセス許可を設定します。

これは、.dll カスタム アクションです。.dll ファイルの名前は ISLockPermissions.dll で、そのエントリ ポイントは ISLockPermissionsInstallAction です。

ISNetApiInstall

.ini ファイルからユーザーおよびグループを作成します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetApiInstall です。

ISNetApiRollback

ユーザーとグループの変更をロールバックします。

これは、.dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetApiRollback です。

ISNetCreateIniForOneUser

ユーザーとグループ用の一時ファイルに操作を抽出します。

これは、ISNetAPI.dll という名前の .dll カスタム アクションです。

ISNetDeleteIniFile

ユーザーとグループ用の一時ファイルをクリーンアップします。

これは、ISNetAPI.dll という名前の .dll カスタム アクションです。

ISNetGetGroups

グループをコンボ ボックスに加えます。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetGetGroups です。

ISNetGetServers

サーバーの一覧をコンボ ボックスに加えます。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetGetServers です。

ISNetGetUsers

ユーザーの一覧をコンボ ボックスに加えます。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetGetUsers です。

ISNetSetLogonName

LogonInformation ダイアログで入力された "ユーザー"、"グループ"、および "サーバー" プロパティを格納します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetSetLogonName です。

ISNetValidateLogonName

LogonInformation ダイアログで、有効な組み合わせのユーザー名、サーバー、パスワードが入力されたことを検証してください。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetValidateLogonName です。

ISNetValidateNewUserInformation

LogonInformation ダイアログで入力された "ユーザー"、"グループ"、および "サーバー" プロパティを格納します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISNetAPI.dll で、そのエントリ ポイントは ISNetValidateNewUserInformation です。

ISPrint

ダイアログ上の ScrollableText コントロールの内容を印刷します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は SetAllUsers.dll で、そのエントリ ポイントは PrintScrollableText です。

ISQuickPatchFinalize

QuickPatch の共有参照カウントをクリーンアップします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchFinalize です。

ISQuickPatchFixShortcut

QuickPatch のショートカットを再インストールします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchFixShortcut です。

ISQuickPatchHelper

QuickPatch のクリーンな機能の状態を適用します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchHelper です。

ISQuickPatchInit

QuickPatch のコンポーネントと機能の状態をクリーンアップします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchInit です。

ISQuickPatchInit9X

QuickPatch のコンポーネントと機能の状態をクリーンアップします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchInit9X です。

ISQuickPatchInit9X2

QuickPatch のコンポーネントと機能の状態をクリーンアップします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は QuickPatchHelper.dll で、そのエントリポイントは ISQuickPatchInit9X2 です。

ISRunSetupTypeAddLocalEvent

[セットアップの種類] ダイアログの [次へ] ボタンに関連付けられている AddLocal イベントを実行します。このアクションは、インストールが [セットアップの種類] ダイアログを表示しない場合に呼び出す必要があります。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は ISXExpHlp.dll で、そのエントリ ポイントは RunSetupTypeAddLocalEvent です。

ISSelfRegisterCosting

自己登録用の一時ファイルに操作を抽出します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は isregsvr.dll で、そのエントリ ポイントは ISSelfRegisterCosting です。

ISSelfRegisterFiles

自己登録ファイルを登録します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は isregsvr.dll で、そのエントリ ポイントは ISSelfRegisterFiles です。

ISSelfRegisterFinalize

自己登録用の一時ファイルをクリーンアップします。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は isregsvr.dll で、そのエントリ ポイントは ISSelfRegisterFinalize です。

ISSetAllUsers

アップグレードまたは初回インストール要件の ALLUSERS を設定します

これは、.dll カスタム アクションです。.dll ファイルの名前は SetAllUsers.dll で、そのエントリ ポイントは SetAllUsers です。

ISSetTARGETDIR

TARGETDIR を [INSTALLDIR] に設定します。

これは、.dll カスタム アクションです。.dll ファイルの名前は SetAllUsers.dll で、そのエントリ ポイントは SetTARGETDIR です。

ISSetupFilesCleanup

サポート ファイルの一時ディレクトリをクリーンアップします。

これは、.dll カスタム アクションです。.dll ファイルの名前は SFHelper.dll で、そのエントリ ポイントは SFCleanupEx です。

ISSetupFilesExtract

一時ディレクトリに、サポート ファイルを抽出します。

これは、.dll カスタム アクションです。.dll ファイルの名前は SFHelper.dll で、そのエントリ ポイントは SFStartupEx です。

ISUnSelfRegisterFiles

自己登録ファイルの登録を解除します。

これは、.dll カスタム アクションです。.dll ファイルの名前は issqlsrv.dll で、そのエントリ ポイントは ISUnSelfRegisterFiles です。

LaunchProgramFileFromSetupCompleteSuccess

インストールの終わりで実行可能ファイルを起動します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は SerialNumCAHelper.dll で、そのエントリ ポイントは LaunchProgram です。

LaunchReadmeFileFromSetupCompleteSuccess

インストールの終わりで Readme ファイルを起動します。

これは、Windows Installer .dll カスタム アクションです。.dll ファイルの名前は SerialNumCAHelper.dll で、そのエントリ ポイントは LaunchReadMe です。

setAllUsersProfile2K

ALLUSERSPROFILE ディレクトリ識別子を初期化します。

SetARPINSTALLLOCATION

[プログラムの追加と削除] の "Readme" プロパティで使われるディレクトリ識別子を解決します。このカスタム アクションが必要な理由は、ARPREADME が Windows Installer プロパティであるため、自動的にフォーマットされないためです。

setUserProfileNT

USERPROFILE ディレクトリ識別子を初期化します。

ShowMsiLog

エンドユーザーが SetupCompleteSuccess、SetupCompleteError、または SetupInterrupted ダイアログで **"Windows Installer のログを表示"** チェック ボックスを選択して [完了] をクリックしたとき、メモ帳で Windows Installer ログ ファイルを表示します。これは、Windows Installer 4.0 以降でのみ使用できます。

コマンドライン ツール

グラフィック ユーザー インターフェイスの他に、InstallShield はビルド時 (たとえば、バッチ処理の一部、またはインストールの動作をカスタマイズするための実行時) に利用できるいくつかのコマンドライン アプリケーションも提供します。このセクションでは、それらのアプリケーションのコマンドライン パラメーターが説明されています。

- IsCmdBld.exe
- MsiExec.exe
- Setup.exe

IsCmdBld.exe

Express プロジェクトでは、IsCmdBld.exe を使って、コマンドラインからリリースをビルドすることができます。

構文

次に、ISCmdBld.exe を実行して**オセロベータ (Othello Beta)** のリリースをビルドする際のステートメントの例を示します。

ISCmdBld.exe –p "C:¥InstallShield 2021 Projects¥My Othello Project¥Othello.ise" –r "SingleImage" –c COMP

上記の例の –pで始まる 最初のパラメーターは、ビルドする .ise ファイルへのパスです。次に、–r Othello Beta はメディアの種類です。パラメーター –c COMP は、パッケージを 1 つのファイルに圧縮するかどうかを指定します。

コマンドライン ビルドがエラーを発生せずに完了した場合、InstallShield によって環境変数 ERRORLEVEL が 0 に設定されます。コマンドライン ビルドの最中にエラーが発生した場合、ERRORLEVEL は 1 に設定されます。ERRORLEVEL がその他の値に設定された場合、一般的に、無効なパラメーターが ISCmdBld.exe に渡されたことを示し、そのエラー原因が ISCmdBld.exe を実行中のコマンド プロンプト ウィンドウに表示されます。

コマンドライン パラメーター

ISCmdBld.exe は、次のコマンドライン パラメーターをサポートします。

テーブル 3-1 ・コマンドライン ビルドのパラメーター

パラメーター	説明
–b <ビルドの場所>	出力先フォルダーとファイルの保存先フォルダーへの完全なパスです。UNC パスを使用できます。ビルド済みインストールのファイルは、指定した場所の Disk Images¥Disk1 サブフォルダーに格納されます。 長いファイル名は引用符で囲んでください。 このパラメーターはオプションです。

テーブル 3-1・コマンドライン ビルドのパラメーター (続き)

パラメーター	説明
-c <圧縮オプション>	このパラメーターを使用して、リリースを単一ファイルに圧縮するか、複数のファイルに非圧縮のまま残すか指定することができます。このパラメーターの有効な引数は COMP および UNCOMP です。リリースを 1 つのファイルに圧縮するよう指定するには、引数 COMP を使用します。リリースを圧縮しない場合は、引数 UNCOMP を使用します。 このパラメーターはオプションです。リリース名が既に存在する場合は、IDE での指定内容に基づいて構成が行われます。新規リリースでこの構成を使用しない場合、新規パッケージは圧縮されません。
-e <Y/N> *セットアップ パッケージ*	このパラメーターを使って、インストール プロジェクトと同時に Setup.exe も作成するかどうかを指定することができます。Setup.exe をビルドするには *Y* を、インストール パッケージのみを作成するには *N* を選択します。 このパラメーターはオプションです。
-fv <バージョン番号>	Setup.exe のデフォルトファイル バージョンを独自のバージョン番号でオーバーライドするには、-fv 2.5.0.0. などの適切なバージョン番号を入力します。このパラメーターはオプションです。 ***メモ:*** *ファイル バージョンは、常に 4 つのフィールドが含まれます。ファイル バージョンに 4 フィールドよりも少ないフィールドを指定すると、残りのフィールドには 0 が挿入されます。たとえば、ファイル バージョンとして 1.1 を指定すると、Setup.exe のバージョン リソースで使用されるファイル バージョンは 1.1.0.0 となります。*
-i <.ini ファイル パス>	すべてのパラメーターをコマンドラインで渡す代わりに、すべてのパラメーターを初期化 (.ini) ファイルに含めておき、その .ini ファイルをコマンドラインから呼び出すことができます。詳細については、「.ini ファイルでコマンドライン ビルド パラメーターを渡す」を参照してください。 絶対および相対パスを使用できます。長いファイル名は引用符で囲んでください。 このパラメーターはオプションです。
-o <マージ モジュール検索パス>	このパラメーターは、プロジェクトで参照させるマージ モジュール (.msm) ファイルを含むフォルダーをコンマ区切りで指定します(複数指定可)。 InstallShield では、マージ モジュールを含むフォルダーを指定するその他の方法も提供されています。詳細については、「マージ モジュールを含むディレクトリを指定する」を参照してください。

テーブル 3-1・コマンドライン ビルドのパラメーター (続き)

パラメーター	説明
-p <プロジェクトの場所>	プロジェクト (.ise) ファイルへのパスを渡します。このパスは、完全修飾パス、相対パス、またはただのファイル名にできます。UNC パスも使用できます。プロジェクトのファイル名のみを渡した場合、ファイルは現在の作業ディレクトリを基準にして検索されます。例: IsCmdBld.exe -p "C:¥InstallShield 2021 Projects¥MyProject1¥MyProject1.ise" これは唯一の必須パラメーターです。 長いファイル名は引用符で囲んでください。
-prqpath <InstallShield 前提条件の検索パス>	このパラメーターは、プロジェクトで参照される InstallShield 前提条件ファイル (.prq) ファイルを含むフォルダーをコンマで区切って指定します(複数指定可)。 InstallShield では、InstallShield 前提条件ファイル ファイルを含むフォルダーを指定するその他の方法も提供されています。詳細については、「InstallShield 前提条件を含むディレクトリを指定する」を参照してください。
-r <リリースの種類>	リリースタイプ を指定する場合にこのパラメーターを使用します (例、SingleImage、WebDeployment)。有効なオプションは次のとおりです: ・ カスタム ・ CD_ROM ・ DVD-10 ・ DVD-18 ・ DVD-5 ・ DVD-9 ・ SingleImage ・ WebDeployment このパラメーターを指定せずにコマンドラインからビルドすると、コマンドラインのビルドは IDE で前回ビルドしたリリースの種類と同じ種類を使用します。IDE でリリースをビルドしたことがない場合、コマンドライン ビルドは SingleImage リリース タイプを使用します。 このパラメーターはオプションです。
-s	このパラメーターを使用すると、サイレント モードでリリースをビルドできます。サイレント ビルドは、ビルドの実行時にエラーや警告メッセージが表示されないようにする場合に便利です。 このパラメーターはオプションです。
-se または -SE	このパラメーターを使って、エラー メッセージ以外の情報を表示しないでビルドを実行することができます。このパラメーターを使って、ビルド エラー (エラー、致命的なエラー、および検証エラー) のみを表示してビルドを行います。 このパラメーターはオプションです。

テーブル 3-1・コマンドライン ビルドのパラメーター (続き)

パラメーター	説明
-u	このパラメーターを使って、リリースをビルドではなく、アップグレードすることができます。このパラメーターを使って、InstallShield の以前で作成したインストール プロジェクトをアップグレードできます。 このパラメーターはオプションです。
-w	このパラメーターはビルドプロセス中に発生する警告をエラーとして処理します。警告が発生するたびに、エラーカウントが 1 ずつ増加します。 このパラメーターはオプションです。
-x	エラーが発生したらビルドを中止する場合は、-x パラメータを使用します。警告の遭遇時にビルドを中止する場合は、このパラメーターを -w パラメータと組み合わせて使用します。 このパラメーターはオプションです。
-y <製品バージョン>	このパラメーターを使用して、コマンドラインから製品のバージョンを指定することができます。これは、製品バージョンのビルド バージョン (3 番目のフィールド) を増加するときに、特に便利です。たとえば、製品バージョンを 1.0.5 に設定するには、次のように入力します。 ISCmdBld.exe -y "1.0.5" 有効な製品バージョン番号については、「製品バージョンを指定する」を参照してください。 このパラメーターはオプションです。

MsiExec.exe

MsiExec.exe は Windows Installer の実行可能プログラムで、インストール パッケージの解析とターゲット システムへの製品のインストールに使用されます。リリースのビルド完了後に、Windows Installer パッケージ (.msi) をコマンドラインからインストールできます。現在、Setup.exe コマンドライン以外に Windows Installer パラメーターを渡すサポートはありません。Windows Installer がターゲット システムにインストールされていることを確認するため、InstallShield はデフォルトでアプリケーションの Setup.exe を作成します。Setup.exe は、ターゲット システムに検出されなければ Windows Installer をインストールし、古いバージョンがあればこれをアップグレードします。

製品のリリースをビルドした後で、コマンドラインからリリースをインストールできます。

MSIExec.exe の完全リファレンスについては、Windows Installer ヘルプ ライブラリのCommand-Line Options を参照してください。MsiExec.exe コマンドライン パラメーターを Setup.exe に渡す方法については、「Setup.exe」を参照してください。

Setup.exe

コンパイルされた .msi ファイルと同様に、Setup.exe では限られた数のコマンドライン パラメーターを利用できます。このパラメーターを使用することで、インストールを実行する言語を指定し、Setup.exe をサイレントで実行するか決定できます。含まれている .msi ファイルに Setup.exe を通してパラメーターを渡すことも可能です。

Setup.exe が受け取ることができるコマンドライン オプションとその説明は以下のとおりです。

- /v : コマンドラインのパラメーターを .msi パッケージに渡します。
- /s : サイレント
- /p : パスワード モード
- /a : 管理インストール
- /j : アドバタイズ モード
- /x : アンインストール モード
- /f : 修復モード
- /ua : Windows 9x MSI エンジンをインストール
- /uw : Windows NT MSI エンジンをインストール
- /b : インストールをローカルにキャッシュ
- /debuglog : デバッグのログ ファイルを生成する
- /w : 待機

Setup.exe 内の .msi ファイルにパラメーターを渡す

Setup.exe をインストールの一部として含めた場合、場合によって Setup.exe 内に保存されている .msi パッケージにコマンドライン パラメーターを渡す必要があります。.msi ファイルに引数を渡すには、/v オプションを使用します。このオプションを指定すると、Msiexec.exe に渡すことができるサポートされたパラメーターをすべて一覧表示できます。たとえば、インストールの詳細ログを作成する場合は、次のように入力します:

Setup.exe /v″/l*v ¥″c:¥My Log Files¥test.log¥″″

この方法でパラメーターを渡す場合に従う特定の書式の規則がいくつかあります。まず、既存の引用中にある引用符の前には円記号 (¥) を置く必要があります。たとえば、上記のコマンドラインには、以下のものがあります: v″/l*v ¥″c:¥My Log Files¥test.log¥″。ログ ファイルへのパスは長いパスなので、引用符を使用する必要があります。ただし、完全な引数には引用符が必要なため、すべての内部引用符の前に円記号を使用しなかった場合、コマンドラインステートメントは失敗します。

上記の例のように、コマンドラインオプション (/v) と渡している引数の間にスペースを入れてはいけないという書式上の規則があります。

また、/v オプションを使用して複数のパラメーターを渡す場合、それらを次のように空白で区切る必要があります:

Setup.exe /v″/l*v ¥″c:¥My Log Files¥test.log¥″ /qn″

このコマンドはログ ファイルを作成し、インストールをサイレントで実行します。

また代わりに、次の例のように、コマンドラインで /v オプションを複数回 (引数ごとに 1 つずつ) 使用することもできます:

Setup.exe /v″/l*v ¥″c:¥My Log Files¥test.log¥″″ /v″/qn″

Setup.exe をサイレントで実行

Setup.exe ファイルの起動時に進捗状況バーを表示しない場合、/s コマンドライン パラメーターを使用します。たとえば、次のコマンドライン ステートメントを入力すると、Setup.exe は起動しますが、ユーザー インターフェイスは表示されません:

Setup.exe /s

.msi セットアップもサイレントで実行する場合は、次のように /v パラメーターを使用して Setup.exe を通して /qn コマンドライン パラメーターを渡す必要があります:

Setup.exe /s /v/qn

メモ: *インストールがパスワード保護されている場合は、/p パラメーターも渡す必要があります。*

コマンドラインからパスワードを指定する

/p パラメーターを使用してパスワード保護のセットアップのパスワードを指定できます。パスワード保護をサイレントモードで実行する場合、コマンドラインパスワードを指定しないとインストールは失敗します。

パスワード保護のセットアップをサイレントで実行する場合、次のステートメントをコマンドラインで入力してください。

Setup.exe /s /v/qn /p"*password*"

コマンドラインからインストール モードを指定する

/a (管理)、/j (アドバタイズ)、/x (アンインストール)、または /f (修復) パラメーターを使用して、インストールを実行するモードを定義することができます。各オプションについては、次に説明します。

管理

インストールを管理モードで実行すると、インストール イメージをネットワークにインストールでき、そのディレクトリにアクセスできるユーザーは、管理インストールを実行した管理者としての権限で全員ローカル マシンにインストールをインストールできるようになります。このオプションに追加パラメーターは必要ありません。

アドバタイズ

アドバタイズは、インストーラーから要求があったときに機能がインストールされ、インストール中にすぐにはインストールされない「ジャストインタイム」タイプのインストールです。/j package オプションを使用して MsiExec.exe を起動すると、機能はエンド ユーザーのシステムでアドバタイズされますが即座にインストールされません。ほとんどの場合、ユーザーは [カスタム セットアップ] ダイアログの機能をアドバタイズするオプションがあります。このパラメーターに対する適切な構文については、Windows Installer ヘルプ ライブラリの Command-Line Options を参照してください。

アンインストール

メンテナンス ダイアログを表示せずにコマンドラインからインストールをアンインストールする場合、次のように /x オプションを使用する必要があります:

Setup.exe /v/x

追加パラメーターは必要ありません。

修復

修復モードでインストールを起動すると、すべてのポータブル実行可能ファイル (.exe、.com、.ocx、.tlb) とヘルプファイル (.chm と .hlp) が壊れずに存在しているかが確認されます。インストールは、これらのファイルの 1 つが存在しない、または壊れていると検出した場合、ファイルの修復を試みます。コマンドラインからインストールを修復モードで起動するには、/f オプションを使用します。このパラメーターの適切な構文についての情報は、「MsiExec.exe」を参照してください。

コマンドラインを使用して Windows Installer エンジンの場所を指定する

コマンドラインから、Web 配布用インストールにダウンロードする Windows Installer エンジンの場所を指定できます。この指定を行うには、次の構文を使用します:

Windows 9x MSI エンジン

Setup.exe /ua"http://www.installshield.com/msiengine20/InstmsiA.exe"

Windows NT MSI エンジン

Setup.exe /uw"http://www.installshield.com/msiengine20/InstmsiW.exe"

キャッシュする場所を指定する

アドバタイズやアプリケーション修復など、Windows Installer の高度機能を使用するために、インストールのコピーをターゲットマシンに置くことができます。これを行うには、完全修飾パスと共に次の構文を使用します。

Setup.exe /b"C:¥Storage¥MyCachedPrograms¥"

メモ: *インストールは、指定された場所のサブディレクトリにキャッシュされます。このサブディレクトリにはインストールのパッケージ コード GUID の名前が付けられます。その場所に展開する際に問題がある場合、インストールは別の場所を要求します。*

このオプションは、SingleImage、カスタムおよび WebDeployment ビルドでのみ使用できます。

デバッグ用のログ ファイルを生成する

/debuglog パラメーターを使って、Setup.exe のログ ファイルを生成できます。

長い名前を持つ InstallShield.log を Setup.exe ファイルと同じディレクトリに生成するには、コマンドライン パラメーターのみを渡します。Setup.exe ファイルが読み取り専用の場所にあるとき、この処理は実行できません。例:

Setup.exe /debuglog

ログ ファイルの名前と場所を指定するには、パスと名前を次の要領で渡します。

Setup.exe /debuglog"C:¥PathToLog¥setupexe.log"

Setup.exe の待機を要求する

/w コマンドライン オプションは、Setup.exe が終了する前に MsiExec.exe が完了するまで Setup.exe を待機させる場合に使用します。さらに、/w コマンドは MsiExec.exe で生成されたすべてのリターン コードを返します。

バッチ ファイルで /w オプションを使用している場合、Setup.exe コマンドライン引数全体を **start /WAIT** で優先させることをお勧めします。正しい形式の使用例は次の通りです。

start /WAIT setup.exe /w

エンドユーザー ダイアログ

このセクションは、InstallShield で利用できるエンド ユーザー ダイアログの参照情報として利用できます。ほとんどのダイアログには、[ダイアログ] ビューからアクセスできます。

すべてのエンドユーザー ダイアログのためのグローバル ダイアログ設定

グローバル ダイアログ設定を使うと、プロジェクトで使用されているすべてのダイアログに共通のダイアログ設定に変更を加えることができます。

[ダイアログ] ビューにある [ダイアログ] エクスプローラーをクリックすると、次のグローバル ダイアログ設定が表示されます。

テーブル 3-1 ・ グローバル ダイアログの設定

設定	説明
グローバル ダイアログ イメージ	グローバル ダイアログ イメージとして使用する画像のパスとファイル名を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。たとえば InstallWelcome のような、完全なダイアログ イメージを持つプロジェクトの各ダイアログは、このファイルをイメージとして表示します。このイメージはダイアログスペースー杯に収まるもので、499 (幅) x 312 (高さ) ピクセルである必要があります。 この設定への入力値は、完全ダイアログ ビットマップ イメージを含む各ダイアログの "ビットマップ イメージ" 設定のデフォルト値として使用されます。各ダイアログに選択されたイメージをオーバーライドするには、[ダイアログ] ビューでダイアログを選択してから、"ビットマップ イメージ" 設定のパスを変更します。 ***メモ:*** *[ダイアログ] ビューにリストされていないダイアログには、"グローバル ダイアログ イメージ" 設定で指定したイメージが使用されます。1 つの例は、インストールが中断されたときに表示されるダイアログです。このため、インストールの視覚的プレゼンテーションを変更する場合、グローバル ダイアログ イメージを指定する必要があります。* ***注意:*** *特定のダイアログに関して "ビットマップ イメージ" 設定の値を変更した後に、この "グローバル ダイアログ イメージ" 設定の値を変更すると、"ビットマップ イメージ" 設定の値は "グローバル ダイアログ イメージ" 設定の値で上書きされます。*

テーブル 3-1・グローバル ダイアログの設定 (続き)

設定	説明
グローバル ダイアログ バナー	グローバル ダイアログ バナーとして使用する画像のパスとファイル名を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。「使用許諾契約書」など、ダイアログの上部にバナー イメージを表示するプロジェクト内の各ダイアログでは、このファイルをバナーとして表示します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。 この設定への入力値は、バナー イメージを含む各ダイアログの "バナー イメージ" 設定のデフォルト値として使用されます。各ダイアログに選択されたイメージをオーバーライドするには、[ダイアログ] ビューでダイアログを選択してから、"バナー イメージ" 設定のパスを変更します。 ***注意:*** *特定のダイアログに関して "バナー イメージ" 設定の値を変更した後に、この "グローバル ダイアログ バナー" 設定の値を変更すると、"バナー イメージ" 設定の値は "グローバル ダイアログ バナー" 設定の値で上書きされます。*
グローバル ダイアログ テーマ	ダイアログ テーマは、エンドユーザー ダイアログに統一感のとれた個性的な印象を与えることができる、あらかじめ定義されている 1 セットのイメージです。プロジェクトに含まれるダイアログに使用するテーマを選択します。詳細については、「ダイアログのテーマ」を参照してください。

テーブル 3-1・グローバル ダイアログの設定 (続き)

設定	説明
All Users オプションの表示	エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供するかどうかを指定します。選択可能なオプションは以下のとおりです: ・ いいえ—エンド ユーザーが製品をインストールする方法を指定できるオプションを表示しません。 ・ はい (Windows 7 以降のみ)—ターゲット システムに Windows 7 または Windows Server 2008 R2 がインストールされている場合、[インストールの準備] ダイアログにボタンが追加されます。これらのボタンを使って、エンド ユーザーは製品をインストールする方法を指定できます。昇格された権限が必要な場合、[すべてのユーザー] ボタンにシールド アイコンが含まれます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 2 に、MSIINSTALLPERUSER プロパティが 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 1 に設定され、MSIINSTALLPERUSER プロパティは設定されません。 ・ はい (すべてのシステム)—ターゲット システムに Windows 7 または Windows Server 2008 R2 がインストールされている場合、[インストールの準備] ダイアログにボタンが追加されます。これらのボタンを使って、エンド ユーザーは製品をインストールする方法を指定できます。昇格された権限が必要な場合、[すべてのユーザー] ボタンにシールド アイコンが含まれます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 2 に、MSIINSTALLPERUSER プロパティが 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合、ALLUSERS プロパティが 1 に設定され、MSIINSTALLPERUSER プロパティは設定されません。 ターゲット システムに Windows Vista 以前、または Windows Server 2008 以前がインストールされている場合、InstallShield は [ユーザー情報] ダイアログにラジオボタンを追加します。これらのラジオ ボタンを使って、エンド ユーザーはプロジェクトをインストールする方法を指定できます。エンド ユーザーが [すべてのユーザー] ラジオ ボタンを選択した場合で、エンド ユーザーが昇格された権限を持つ場合、ALLUSERS プロパティは 1 に設定されます。エンド ユーザーが [ユーザーごと] ボタンを選択した場合で、エンド ユーザーが昇格された権限を持つ場合、ALLUSERS プロパティは空白文字列 (“”) に設定されます。 デフォルト値は [いいえ] です。より詳しい情報は、「ユーザーごとのインストールとマシンごとのインストールの違い」をご覧ください。

[スプラッシュ ビットマップ] ダイアログ

[スプラッシュ ビットマップ] ダイアログは、エンド ユーザーがインストールを起動したときに最初に表示されるダイアログです。通常このダイアログには、会社や製品の名前やロゴが入ったイメージ (.bmp または .jpg ファイル) が含まれます。

このダイアログはオプションで、デフォルトでは選択されていません。

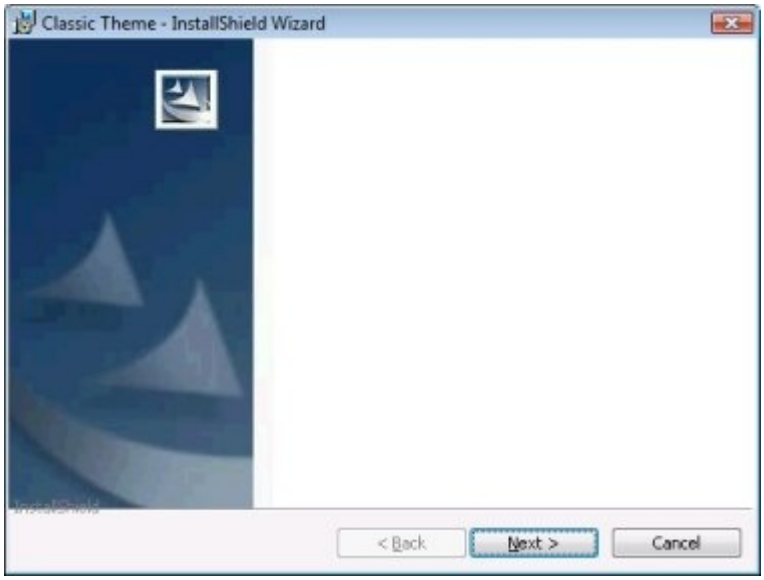


図 3-1: Classic テーマの [スプラッシュ ビットマップ] ダイアログ

図 3-2: InstallShield Blue テーマの [スプラッシュ ビットマップ] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [スプラッシュ ビットマップ] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[スプラッシュ ビットマップ] ダイアログには以下の設定があります。

テーブル 3-2 · [スプラッシュ ビットマップ] ダイアログの設定

設定	説明
スプラッシュ ビットマップ	スプラッシュ画面ダイアログに使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、465 (幅) x 281 (高さ) ピクセルである必要があります。
Sunken	スプラッシュ イメージがくぼんで見えるようにダイアログに表示する場合は、[はい] を選択します。ダイアログのイメージを平坦に表示する場合は、[いいえ] を選択します。

[インストール - ようこそ] ダイアログ

[スプラッシュ ビットマップ] ダイアログを表示するように選択したかどうかで、[インストール - ようこそ] ダイアログがインストールの起動時に表示される最初のダイアログになるかどうかが決定します。このダイアログには、2 つの目的があります。1 つ目は、ユーザーにインストールが実行中であることを知らせます。[一般情報] ビューの "製品名" 設定に入力する名前をようこそメッセージと一緒に表示することによって、セットアップの起動をユーザーに知らせます。[ようこそ] ダイアログの 2 つ目の目的は、インストールの著作権情報を表示することです。

[ようこそ] ダイアログはセットアップ作成の必須項目ですが、ダイアログの外観と表示する内容はカスタマイズできます。

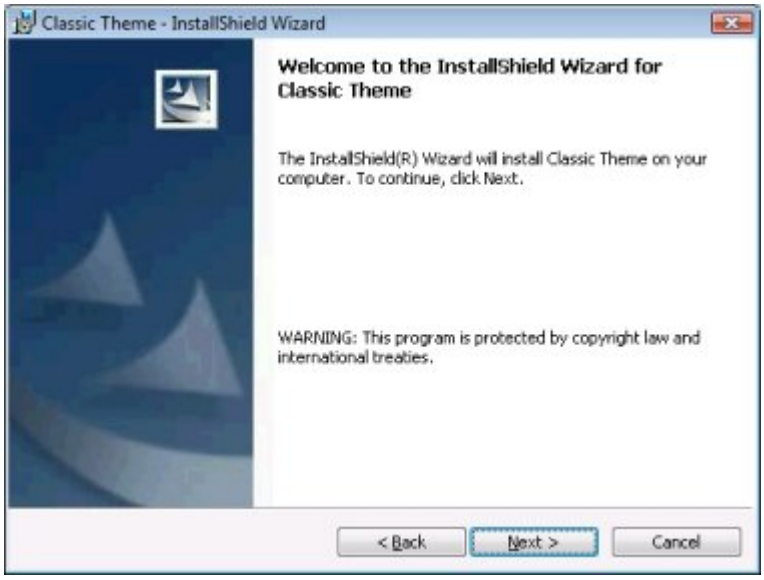


図 3-3: Classic テーマの [インストール - ようこそ] ダイアログ

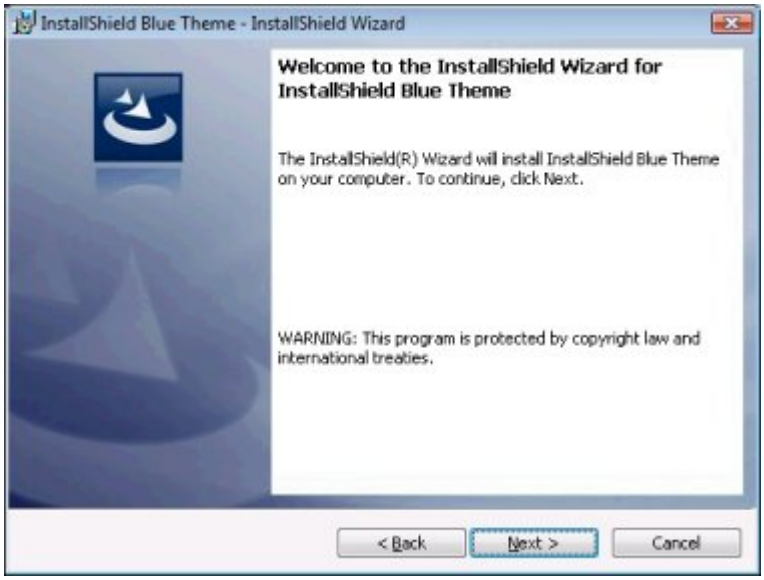


図 3-4: InstallShield Blue テーマの [インストール - ようこそ] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [インストールへようこそ] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[インストールへようこそ] ダイアログには以下の設定があります。

テーブル 3-3 · [インストールへようこそ] ダイアログの設定

設定	説明
ビットマップ イメージ	ダイアログの背景に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 312 (高さ) ピクセルである必要があります。
著作権の表示	製品の著作権情報を含めるかどうかを指定します。 [はい] を選択する場合は、"著作権テキスト" 設定に適切なテキストを入力します。
著作権テキスト	製品の著作権情報を入力します。"著作権の表示" 設定を [はい] に設定しない限り、この情報は表示されません。

[使用許諾契約書] ダイアログ

[使用許諾契約] ダイアログには、エンドユーザー使用許諾契約書 (EULA) が表示されます。このダイアログが表示されると、ユーザーはインストールを継続する前に使用許諾契約に承諾しなくてはなりません。このダイアログは必須ではありませんが、デフォルトでは選択されています。

図 3-5: Classic テーマの [使用許諾契約書] ダイアログ

図 3-6: InstallShield Blue テーマの [使用許諾契約書] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [使用許諾契約] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[使用許諾契約] ダイアログには以下の設定があります。

テーブル 3-4 · [使用許諾契約書] ダイアログ

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
ライセンス ファイル	使用許諾契約書を含むリッチ テキスト ファイル (.rtf) のパスを入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。 ***メモ·*** *ファイルは RTF 形式でなければなりません。プレーンテキストファイルは受け付けません。*

Readme ダイアログ

Readme ダイアログには、アプリケーションの Readme ファイルが表示されます。このダイアログはオプションで、デフォルトでは選択されていません。

図 3-7: Classic テーマの Readme ダイアログ

図 3-8: InstallShield Blue テーマの Readme ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は Readme ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。Readme ダイアログには以下の設定があります。

テーブル 3-5 ・ Readme ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
Readme ファイル	readme 情報を含むリッチ テキスト ファイル (.rtf) のパスを入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。 ***メモ・*** *ファイルは RTF 形式でなければなりません。プレーンテキストファイルは受け付けません。*

[ユーザー情報] ダイアログ

[ユーザー情報] ダイアログでは、エンドユーザーからユーザー名、会社名などの情報を集めることができます。さらに、ソフトウェアの不正使用を避けるために、シリアル番号を検証する DLL にリンクすることもできます。このダイアログは必須ではありませんが、デフォルトでは選択されています。

[ユーザー情報] ダイアログでは、エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供することもできます。詳細については、「ユーザーごとのインストールとマシンごとのインストールの違い」を参照してください。

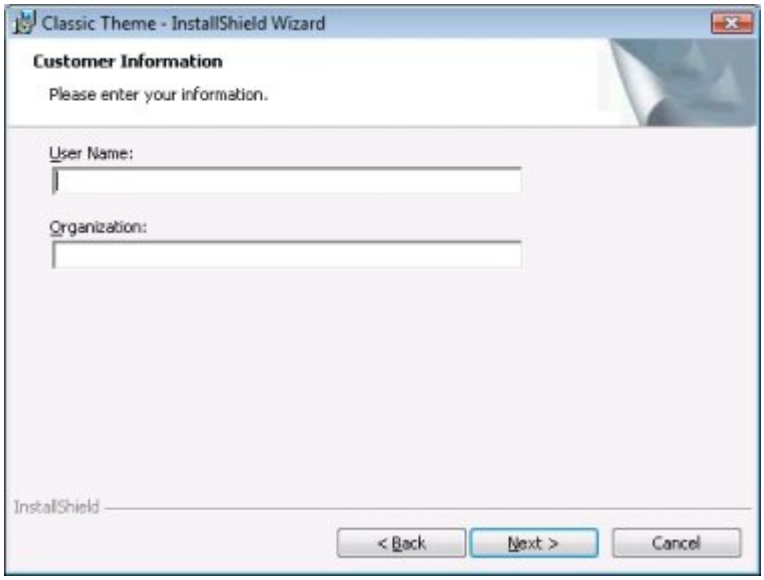


図 3-9: Classic テーマの [ユーザー情報] ダイアログ

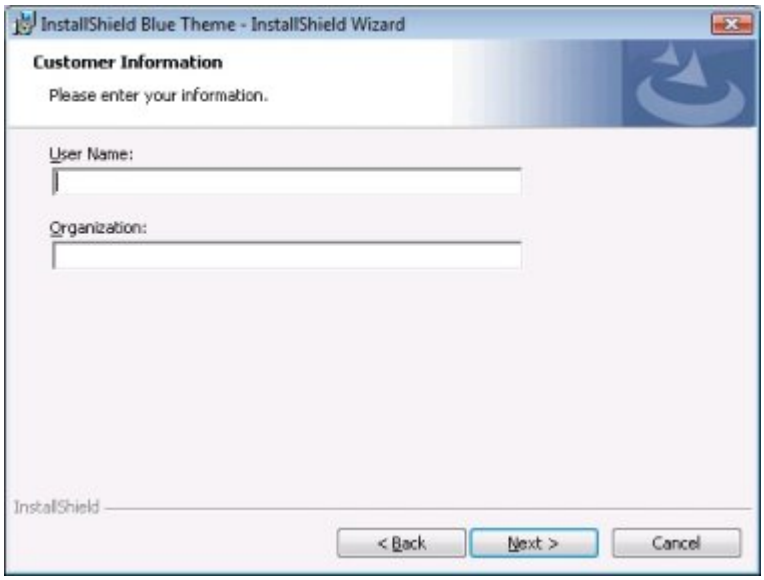


図 3-10: InstallShield Blue テーマの [ユーザー情報] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は、エクスプローラーから [ユーザー情報] ダイアログを選択すると、[ダイアログ] エクスプローラーの右に表示されます。[ユーザー情報] ダイアログには以下の設定があります。

テーブル 3-6 ・ [ユーザー情報] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
シリアル番号の表示	ダイアログに "シリアル番号" フィールドを表示するかどうかを指定します。エンド ユーザーがシリアル番号を入力するとき、その番号を "シリアル番号検証 DLL" 設定で指定する DLL で検証することができます。詳細については、「シリアル番号を検証するカスタム アクションを使用する」を参照してください。 [いいえ] を選択すると、シリアル番号の設定は無効になります。
シリアル番号テンプレート	製品のシリアル番号の形式を指定します: ・ 疑問符 (?) を英数文字にあたる部分に入力します。 ・ 数記号 (#) を数字にあたる部分に入力します。 ・ ダッシュ (-) を使って、文字列をグループに分けします。ダッシュ (-) は、1 つの文字グループがそこで終わり、次のグループがそこから始まるということを示すシリアル番号内のブレーク ポイントを意味します。 たとえばシリアル番号フォーマットが ###-???? の場合、[ユーザー情報] ダイアログの "シリアル番号" フィールドには 2 つのボックスが表示されます。エンド ユーザーは、最初のボックスに 3 つの数字、2 番目のボックスに 4 つの英数字しか入力することができません。
シリアル番号検証 DLL	シリアル番号の検証関数を含む DLL へのパスを入力するか、または省略記号ボタン (...) をクリックしてファイルを参照します。関数のフォーマット方法については、「シリアル番号を検証するカスタム アクションを使用する」をご覧ください。
検証関数	シリアル番号を検証する DLL 関数の名前を入力します。たとえば、DLL に **ValidateSN()** と呼ばれる関数が含まれる場合、以下のように入力します。 **ValidateSN** これには特別なフォーマットは必要ありません。ただし、この関数は、"シリアル番号の検証 DLL" 設定で指定する DLL に存在している必要があります。
正常時戻り値	正常時に戻される、検証の関数の値を指定します。 この戻り値のデータ型は、長整数型でなくてはなりません。ゼロはアクションが失敗したことを示すため、ここにはゼロ以外の任意の数字を入力します。

テーブル 3-6 · [ユーザー情報] ダイアログの設定 (続き)

設定	説明
再試行回数制限	インストールを終了する前に受け入れる、失敗した戻り値の最大試行回数を指定します。この設定に入力する数が少ない方が、第三者にシリアル番号を読まれる危険性が低くなります。

[インストール先のフォルダー] ダイアログ

[インストール先のフォルダー] ダイアログには、インストール パッケージのインストール先を表示し、そのインストール先を変更できる機能があります。このダイアログはオプションで、デフォルトでは選択されていません。

図 3-11: Classic テーマの [インストール先フォルダー] ダイアログ

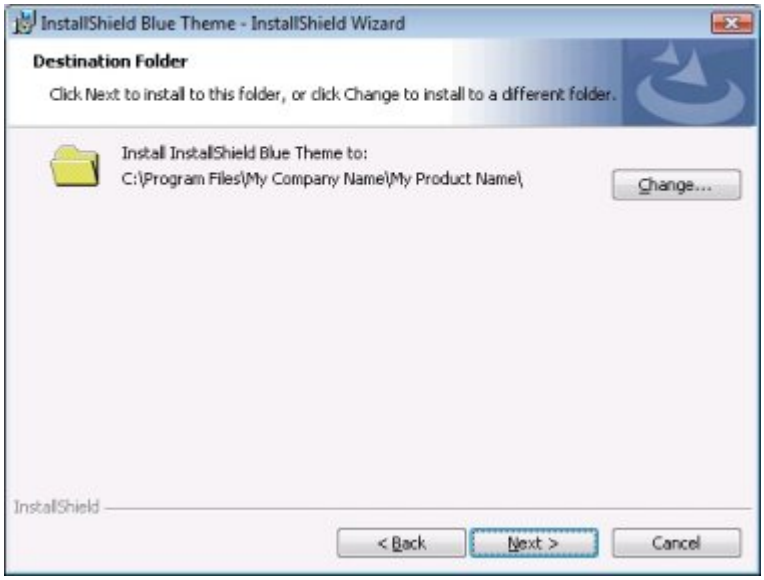


図 3-12: InstallShield Blue テーマの [インストール先フォルダー] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [インストール先のフォルダー] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[インストール先のフォルダー] ダイアログには以下の設定があります。

テーブル 3-7 ・ [インストール先のフォルダー] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
インストール先変更の表示	このダイアログに、エンド ユーザーが製品の新しいインストール先フォルダーを定義できる [変更] ボタンを含むかどうかを指定します。このボタンは、[セットアップの種類] ダイアログと [カスタム セットアップ] ダイアログにも表示できます。

[データベース フォルダー] ダイアログ

[データベース フォルダー] ダイアログには、インストール時にインストールされるデータベース ファイルのインストール先が表示されます。たとえば、ネットワーク ユーザーのためのインストールを作成して、全ユーザーのために共通のデータベースをインストールする場合、ローカルの各マシンにデータベースをインストールする代わりにネットワークにインストールします。データベース ファイルのデフォルト インストール先フォルダーを設定するには、[一般情報] ビューの DATABASEDIR 設定を使用します。

このダイアログはオプションで、デフォルトでは選択されていません。

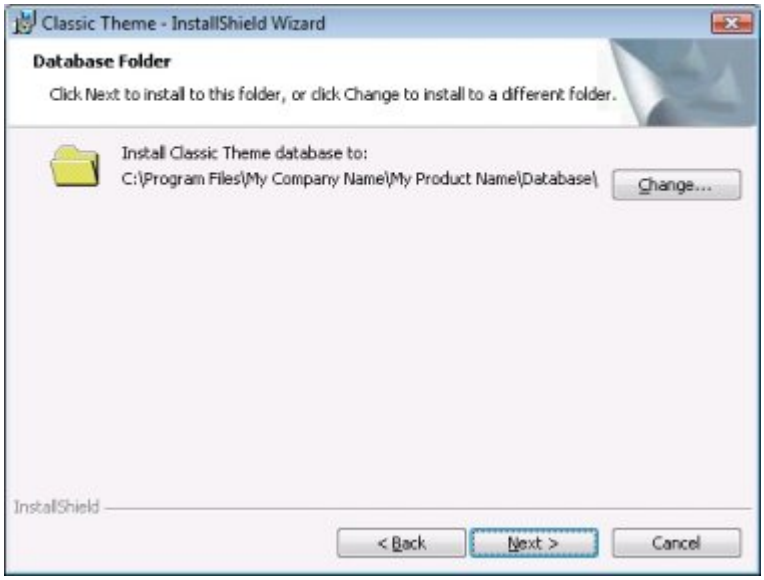


図 3-13: Classic テーマの [データベース フォルダー] ダイアログ

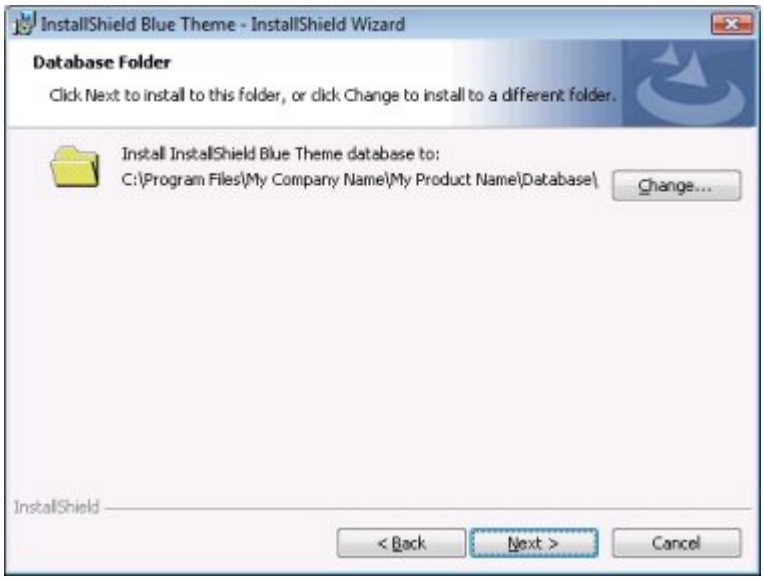


図 3-14: InstallShield Blue テーマの [データベース フォルダー] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [データベース フォルダー] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[データベース フォルダー] ダイアログには 2 つの設定があります。

テーブル 3-8 ・ [データベース フォルダー] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
インストール先変更の表示	このダイアログに、エンド ユーザーが製品の新しいデータベースのインストール先フォルダーを定義できる [変更] ボタンを含むかどうかを指定します。このボタンは、[セットアップの種類] ダイアログと [カスタム セットアップ] ダイアログにも表示できます。

[セットアップの種類] ダイアログ

[セットアップの種類] ダイアログには、[セットアップの種類] ビューで定義したさまざまなセットアップの構成が表示されます。ここでエンドユーザーは、任意のセットアップの種類を選択して、独自のニーズに合わせてインストールをカスタマイズできます。製品のインストール先を表示し、そのインストール先を変更できる機能もあります。

[セットアップの種類] ビューで複数のセットアップ タイプを選択している場合、[セットアップの種類] ダイアログが自動的にインストールに含まれます。選択されているセットアップの種類が 1 つのみの場合、このダイアログは表示されません。

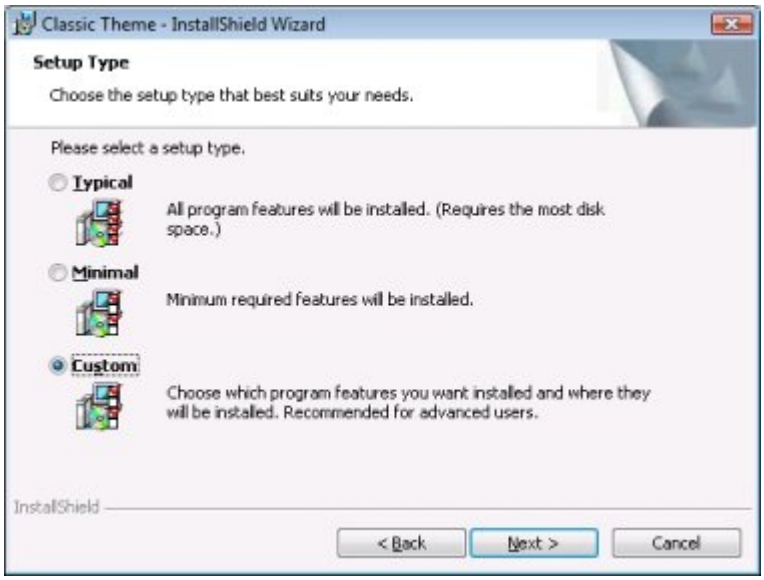


図 3-15: Classic テーマの [セットアップの種類] ダイアログ

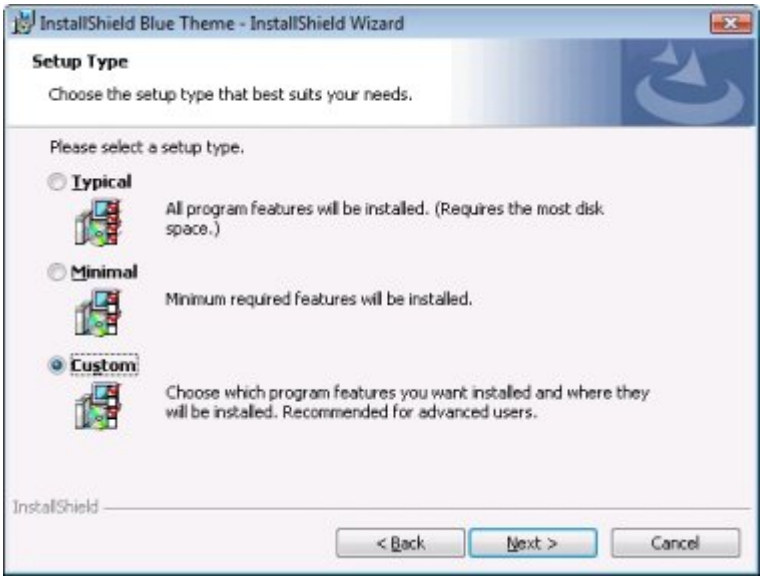


図 3-16: InstallShield Blue テーマの [セットアップの種類] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は、ダイアログ エクスプローラーで [セットアップの種類] ダイアログを選択すると、エクスプローラーの右に表示されます。[セットアップの種類] ダイアログには、次の設定があります。

テーブル 3-9 ・ [セットアップの種類] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。

[カスタム セットアップ] ダイアログ

[カスタム セットアップ] ダイアログは、ユーザーが [カスタム セットアップ] セットアップ タイプを選択した場合のみ表示されます。したがって、このダイアログをセットアップに含むためには、[セットアップの種類] ビューでカスタム セットアップを許可する必要があります。このダイアログでは、エンドユーザーがインストールする機能を選択できます。セットアップのインストール先を表示し、そのインストール先を変更できる機能もあります。

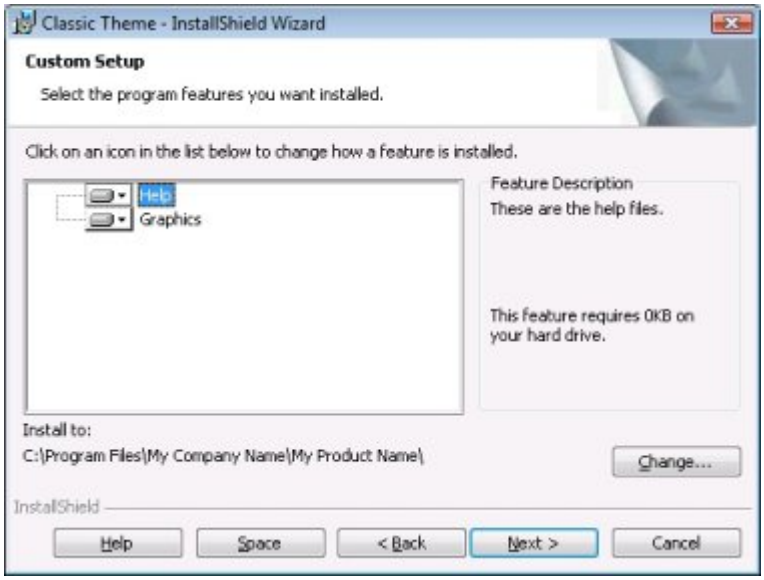


図 3-17: Classic テーマの [カスタム セットアップ] ダイアログ

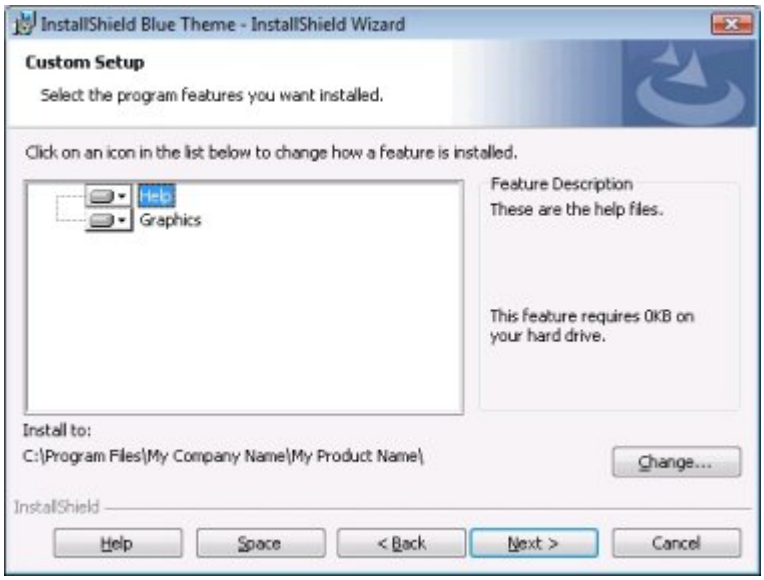


図 3-18: InstallShield Blue テーマの [カスタム セットアップ] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [カスタム セットアップ] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[カスタム セットアップ] ダイアログには以下の設定があります。

テーブル 3-10 • [カスタム セットアップ] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
インストール先変更の表示	このダイアログに、エンド ユーザーが製品の新しいインストール先フォルダーを定義できる [変更] ボタンを含むかどうかを指定します。このボタンは [インストール先のフォルダー] ダイアログでも表示できます。

[インストールの準備完了] ダイアログ

[インストールの準備完了] ダイアログは、ファイル転送の開始前に表示される最後のダイアログです。つまり、エンド ユーザーが、InstallShield Wizard がシステムを変更する前に、インストール オプションを変更またはキャンセルできる最後の機会です。既に設定済みの情報の概要を、このダイアログに表示することもできます。このダイアログはオプションですが、デフォルトでは選択されています。

インストールが Windows Vista 以降のシステム上で実行されているとき、インストールが昇格された権限でまだ実行されていない場合、[インストールの準備完了] ダイアログの [インストール] ボタンに [ユーザー アカウント制御 (UAC)] シールド アイコンが表示されます。

[インストールの準備] ダイアログでは、エンド ユーザーが製品をすべてのユーザー、または現在のユーザーのみにインストールするかを選択できるオプションを提供することもできます。詳細については、「ユーザーごとのインストールとマシンごとのインストールの違い」を参照してください。

インストールが Windows Vista 以降のシステム上で実行されているとき、インストールが昇格された権限でまだ実行されていない場合、[インストールの準備完了] ダイアログの [インストール] ボタンに [ユーザー アカウント制御 (UAC)] シールド アイコンが表示されます。

ヒント・ *InstallShield は昇格された権限で実行されます。このため、Windows Vista 以降のシステムでインストールを InstallShield 内から起動したとき、インストールは昇格された権限で実行され、[ユーザー アカウント制御 (UAC)] シールド アイコンは [インストール] ボタンに表示されません。*

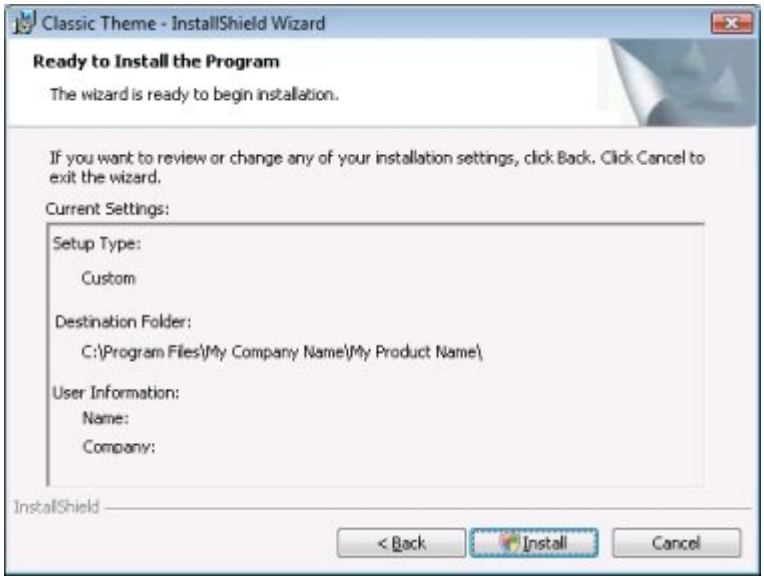


図 3-19: Classic テーマの [インストールの準備完了] ダイアログ

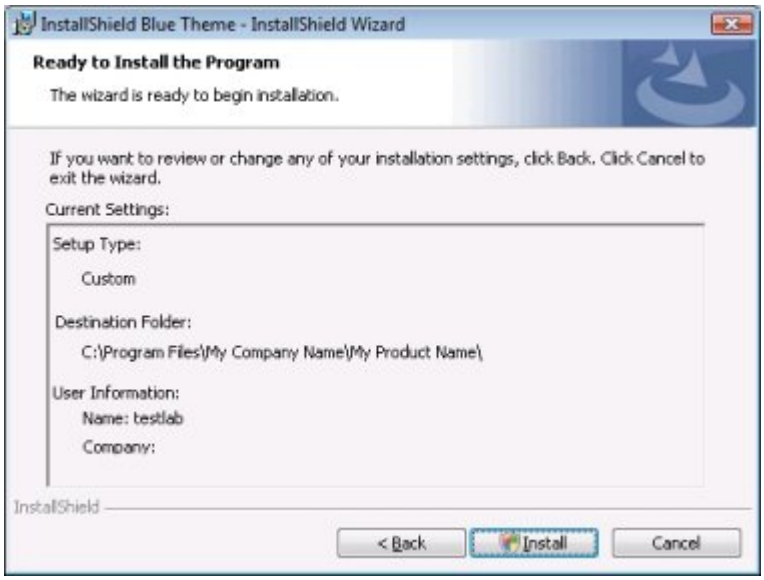


図 3-20: InstallShield Blue テーマの [インストールの準備完了] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [インストールの準備] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[インストールの準備] ダイアログには、以下の設定があります。

テーブル 3-11 ・ [インストールの準備] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
現在の設定の表示	このダイアログにエンド ユーザーが指定したインストール設定の概要を含めるかどうかを指定します。エンド ユーザーは、この情報を [インストールの準備完了] ダイアログから編集することはできませんが、必要であれば戻って設定を変更することができます。

[インストールの準備] ダイアログ

[セットアップ進行状況] ダイアログを使用すると、ファイル転送などのインストール中に実行されているすべてのアクションを表示できます。このダイアログは必須ではありません。

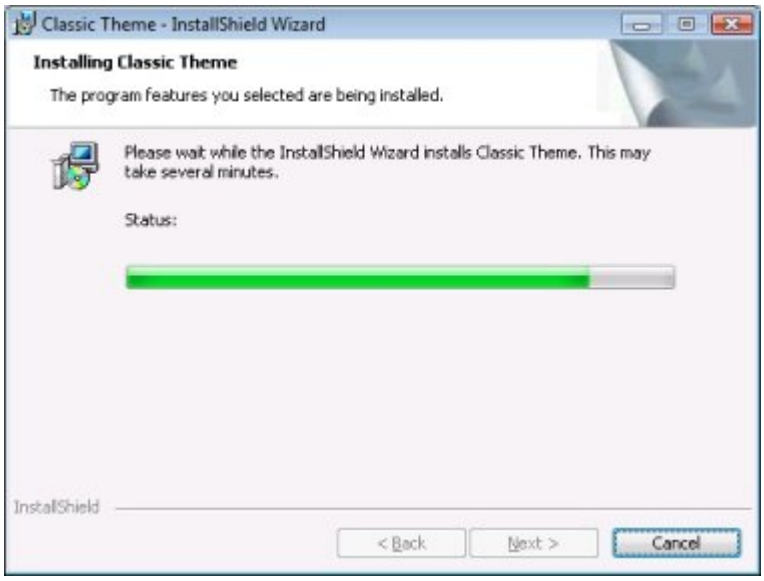


図 3-21: Classic テーマの [セットアップ進行状況] ダイアログ

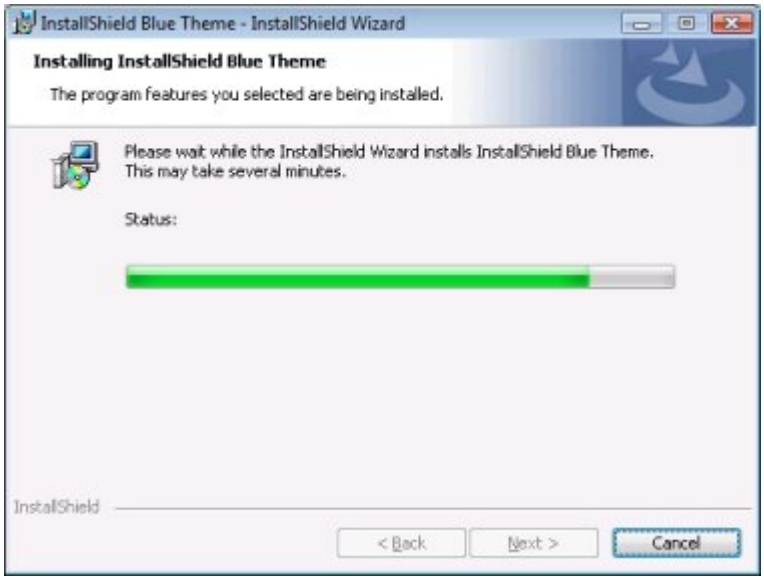


図 3-22: InstallShield Blue テーマの [セットアップ進行状況] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [セットアップ進行状況] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[セットアップの進行状況] ダイアログには、次の設定があります。

テーブル 3-12 • [セットアップ進行状況] ダイアログの設定

設定	説明
バナー ビットマップ	ダイアログのバナー (ダイアログの上部) に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。イメージは、499 (幅) x 58 (高さ) ピクセルである必要があります。
進行状況バーの表示	このダイアログに進行状況バーを含めるかどうかを指定します。進行状況バーは、インストール処理がどこまで完了したか、またどれだけ残っているかをエンドユーザーに視覚的に表示します。

[セットアップの完了] ダイアログ

インストールが正常に完了すると、[セットアップの完了] ダイアログが表示されます。このダイアログでは、エンドユーザーに、インストールが完了したことが通知されます。また、エンドユーザーはオプションで、製品の最新アップデートの確認、製品の起動、または Readme ファイルの表示を行うことができます。これは、必須ダイアログです。

[一般情報] ビューでログ記録を有効にすると、Windows Installer 4.0 が製品のインストール中にログ ファイルを作成して、ログ ファイルのパスを MsiLogFileLocation 設定に挿入します。また、[セットアップの完了] ダイアログに **[Windows Installer ログの表示]** チェック ボックスが追加されました。エンドユーザーがこのチェック ボックスを

選択してから [終了] をクリックすると、テキスト ファイル ビューアーまたはエディターでログ ファイルが開きます。詳細については、「Windows Installer インストールをログ記録するかどうかを指定する」を参照してください。

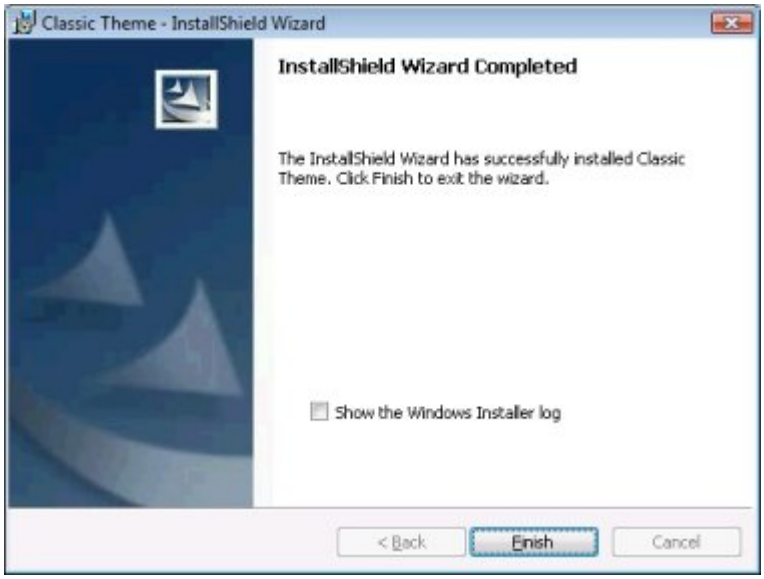


図 3-23: Classic テーマの [セットアップの完了] ダイアログ

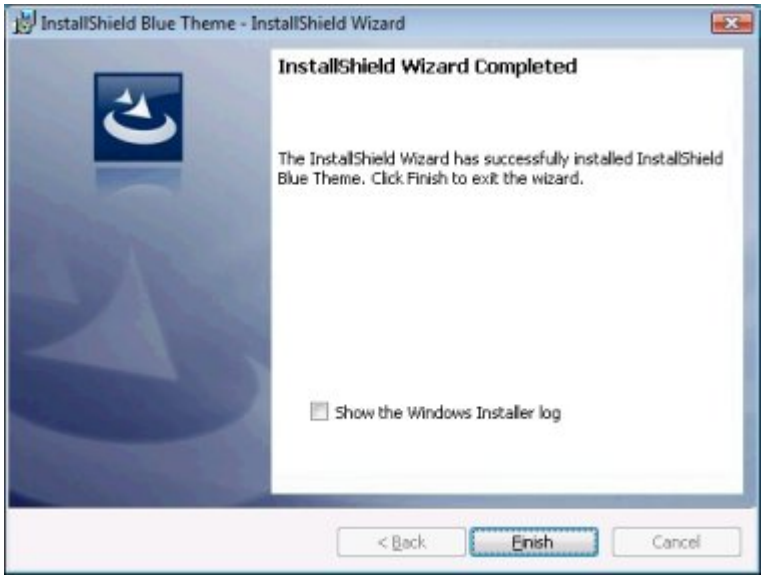


図 3-24: InstallShield Blue テーマの [セットアップの完了] ダイアログ

ダイアログのカスタマイズ

設定を編集して、このダイアログをカスタマイズできます。これらの設定は [セットアップの完了] ダイアログを選択すると、ダイアログ エクスプローラーの右に表示されます。[セットアップの完了] ダイアログには以下の設定があります。

テーブル 3-13 • [セットアップの完了] ダイアログの設定

設定	説明
ビットマップ イメージ	ダイアログの背景に使用するイメージ ファイル (.bmp または .jpg) のパスと名前を入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。このイメージはダイアログスペース一杯に収まるもので、499 (幅) x 312 (高さ) ピクセルである必要があります。
FlexNet Connect インターフェイスを使用する	インストールで自動アップデート通知を有効にして、[セットアップの完了] ダイアログに [**はい、セットアップが完了したあと、プログラムのアップデート (推奨) を確認します**] チェック ボックスを追加する場合、この設定に [はい] を選択します。エンドユーザーがこのチェック ボックスを選択してから、[完了] ボタンをクリックしてインストールを終了すると、FlexNet Connect が起動されます。詳細については、「FlexNet Connect を利用してエンドユーザーにアップグレードの通知をする」を参照してください。 このプ設定に [はい] を選択した場合、ダイアログに含まれるその他の設定は無効です。 [セットアップの完了] ダイアログに [**はい、プログラムのアップデートを確認します**] チェック ボックスを含めない場合、この設定には [いいえ] を選択します。
起動プログラムの表示	ダイアログに [**プログラムを起動する**] チェック ボックスを含めるかどうかを指定します。エンド ユーザーがこのチェック ボックスを選択してから、[完了] ボタンをクリックしてインストールを終了すると、製品が起動されます。 [はい] を選択する場合、"プログラム ファイル" 設定から起動する実行可能ファイルを選択しなくてはなりません。
プログラム ファイル	エンド ユーザーが [**プログラムを起動する**] チェック ボックスを選択した場合に起動する実行可能ファイルを選択します。"起動プログラムの表示" 設定には、[はい] を選択しなくてはなりません。
コマンドライン パラメーター	"プログラム ファイル" 設定で指定した実行可能ファイルに渡すコマンドライン パラメーターを入力します。
Readme の表示	インストールが完了した後、Readme ファイルを起動するオプションをユーザーに提供するかどうかを指定します。[はい] を選択すると、InstallShield が [セットアップの完了] ダイアログにチェック ボックスを追加します。このチェック ボックスが選択されると、Readme 設定で指定した Readme ファイルが起動されます。
Readme ファイル	**[Readme ファイルを表示する]** チェック ボックスに関連付けるファイルのパスを入力するか、省略記号ボタン (...) をクリックしてファイルを参照します。

MsiRMFilesInUse ダイアログ

インストール終了後のシステム再起動は、エンドユーザーにとって不都合なものです。Windows ロゴ プログラムの要件の 1 つに、エンドユーザーがインストール完了後自動的にアプリケーションを閉じて再起動を行うことができるオプションを含まなくてはならないという項目があります。

[使用中のファイル (再起動マネージャー)] ダイアログは、Express インストールにデフォルトで含まれています。インストール中に更新が必要なファイル (複数可) が他のアプリケーションによって使用中の場合、Windows Vista 以降のシステム上では [使用中のファイル (再起動マネージャー)] ダイアログが表示されます。ダイアログには、エンドユーザーが選択できる次の 2 つのオプションが含まれます。

- エンドユーザーは選択で、インストールの完了後、自動的にファイルを使用中のアプリケーションを閉じて、再起動することができる。
- エンドユーザーは、アプリケーションを閉じない選択ができる。インストールの終わりで再起動が必要。

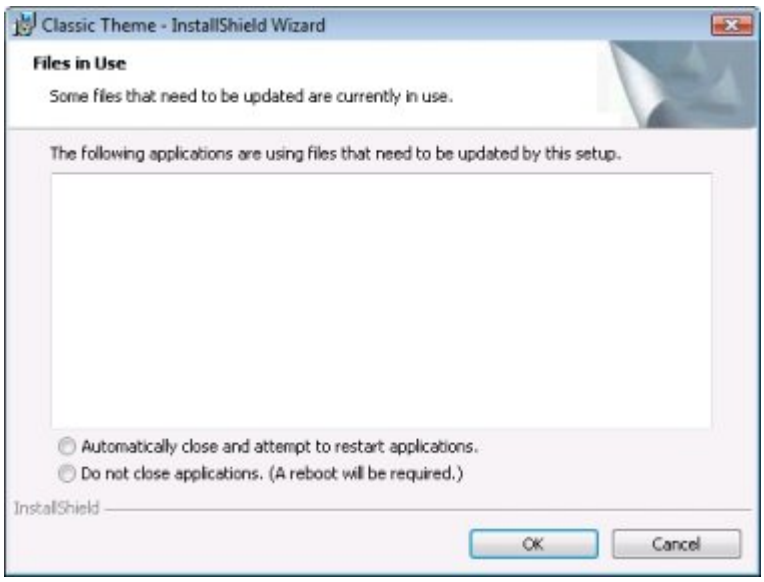


図 3-25: Classic テーマの [使用中のファイル (再起動マネージャー)] ダイアログ

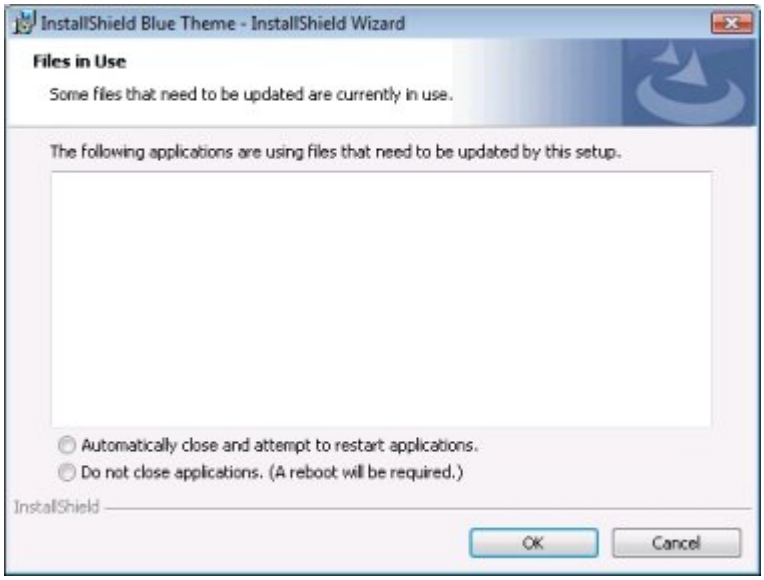


図 3-26: InstallShield Blue テーマの [使用中のファイル (再起動マネージャー)] ダイアログ

このダイアログの詳細は、「Windows Vista 以降のシステムの再起動を最小限にする」を参照してください。

4

よく寄せられる質問 (FAQ)

このセクションには、トピックごとに一般的な質問、およびその回答へのリンクが掲載されています。

- Windows Installer ランタイム エラーについての情報を探す
- アプリケーションのインストール先フォルダーを指定する方法
- ターゲット マシンに Windows Installer サービスがあることを確認する方法
- InstallShield で行なった変更をインストールの実行時に有効にする方法
- 使用許諾契約書の表示方法
- ショートカットの作成方法
- ファイルの種類をアプリケーションに関連付ける方法
- インストールから外部アプリケーションを起動する方法
- ターゲット マシンのオペレーティング システムに基づいてインストールを条件付でインストールする方法
- アプリケーションをアップグレードする方法
- Windows Installer が上書きするファイルを判別する方法
- インストールに自己展開型実行可能ファイルを作成する方法
- ビルド中に製品バージョンを変更する方法
- Web ベースのインストールを作成する方法
- インストールが 1 枚のフロッピーディスクに収まらない理由
- インストールのサイズを縮小する方法
- ターゲット オペレーティング システムに基づいてファイルをインストールするようにインストールを作成する方法
- COM 情報抽出と自己登録の違い
- InstallShield Premier Edition または Professional Edition にアップグレードするときに必要な手順

その他の質問に対する答えの多くは、ナレッジベースに用意されています。

用語集

テーブル 1・InstallShield Express 用語集

用語	定義
.bin ファイル	Unix プラットフォームで実行可能なバイナリ ファイル。アプリケーションファイルは通常 .bin 拡張子を持ちます。
.bmp ファイル	ビットの配列として表示されるイメージのビットマップファイル。ビットマップ グラフィックでは、イメージは集合体としてパターンを形成するピクセルと呼ばれる小さな四角の集まりでスクリーン上に表示されます。イメージ内の各ピクセルは、1 つまたは複数のビットに対応します。
.cab ファイル	「キャビネット ファイル」を参照してください。
.command ファイル	Mac OS X プラットフォーム上で実行可能なコマンドファイル。アプリケーションファイルは通常 .command 拡張子を持ちます。
.cub ファイル	ファイル内部整合性評価プログラム (ICE) のカスタム アクションを格納し、アクセスを提供する検証モジュール。「カスタム アクション」および「内部整合性評価プログラム」も参照してください。
.exe ファイル	Windows プラットフォーム上で実行可能なバイナリファイル。アプリケーションファイルは通常 .exe 拡張子を持ちます。

テーブル 1・InstallShield Express 用語集

用語	定義
.gif ファイル	Web 上でサポートされているグラフィック交換形式ファイル。このファイルの種類は、ビットマップ イメージを表示するのに利用されるビットマップ グラフィック ファイル形式です。これらのファイルはデータが損なわれないロスレス圧縮メソッドを使います。このファイルタイプは 256 色までサポートします。そのほか透明色もサポートし、背景に透明を設定することで下にあるレイヤーの色が見えるようにすることが可能です。このファイル形式は、線画のような限られた色のみを使ったイメージの場合 .jpg 形式よりも適しています。
.idt ファイル	エクスポートされた Windows Installer データベース テーブル。
.ism ファイル	InstallShield がプロジェクト情報を保存するために利用する作業ファイル。リリースをビルドする際、InstallShield は .ism ファイルを使用して配布用 .msi ファイルを作成します。
.jar ファイル	インストールに必要なクラス、イメージ、およびサウンド ファイルを含み、単一ファイルに zip されている Java アーカイブ ファイルです。
.jpg ファイル	JPEG として知られる。標準化した委員会の元の名称をとった Joint Photographic Experts Group の略語。このファイルの種類は 256 色までサポートします。そのほか透明色もサポートし、背景に透明を設定することで下にあるレイヤーの色が見えるようにすることが可能です。ファイルは非可逆圧縮を利用します。この種類の圧縮はカラーおよびグレースケールの濃淡イメージを圧縮し、余分な情報または不必要な情報を省略するためにデータが失われます。これらのイメージは 1600 万色をサポートし、写真や複雑なグラフィックに適しています。
.mif ファイル	ハードウェアまたはソフトウェア コンポーネントについて記述するために使われる管理情報フォーマット ファイル。
.msi ファイル	完成した状態の Windows Installer インストール パッケージ。インストール リソースファイルを含み、アプリケーションのデータファイルすべてを圧縮して格納することができます。.msi ファイルはエンドユーザーに配布されるファイルで、Windows Installer サービスとインタラクトしてアプリケーションをマシンにインストールします。
.msm ファイル	マージ モジュールのデータベースファイル。このファイル タイプはモジュールのインストール プロパティおよびインストール ロジックすべてを含みます。「マージ モジュール」も参照してください。
.msp ファイル	Windows Installer パッチ ファイル。このファイルの種類は概要情報ストリーム、トランスフォーム サブストレージ、およびキャビネット ファイルで構成されます。.msp ファイルは、ターゲットとするインストールパッケージのデータベースにパッチ情報を追加するデータベーストランスフォームが 1 つ以上含まれます。インストーラーはこのパッチ情報を利用してキャビネット ファイルに格納されているパッチファイルを適用します。

テーブル 1・InstallShield Express 用語集

用語	定義
.mst ファイル	トランスフォーム パッケージ。このファイルの種類は、2 つの .msi データベース間の差分を含む簡素化された Windows Installer データベースです。トランスフォームを使うと、管理者がインストールパッケージを配布する際、変更した設定をデータベースに適用することができます。「トランスフォーム」も参照してください。
.NET	情報、人間、システム、そしてデバイスをつなげるために Microsoft が作成したオペレーティング システムプラットフォーム。.NET 環境を利用すると、開発者は共通言語ランタイムを通して好みの言語を使ってアプリケーションおよび Web サービスをビルド、作成、そして配布することができます。
.NET Compact Framework	モバイル デバイスなど、リソースに制限のあるデバイス上で実行できるようデザインされた .NET Framework の縮小版。.NET Compact Framework は現在、Pocket PC と Windows CE .NET デバイスでのみ使用できます。他のデバイスのサポートは今後のリリースで実現していく予定です。
.NET Framework	アプリケーションおよび Web サービスのような .NET 技術を利用したサービスをビルド、配布、および実行するために Microsoft が作成したプログラムの基盤。.NET Framework は 3 つの主要な部分で構成されます。一般言語ランタイム、統一されたクラス ライブラリが階層に分かれたセット、および ASP.NET と呼ばれる Active Server Pages のコンポーネント化されたバージョン。
.ocx ファイル	OLE (Object Linking and Embedding) カスタム コントロール ファイル。このファイルタイプは、ウィンドウサイズの変更などを可能にする機能を実行するためにアプリケーションによって呼び出される、クロスプラットフォームの COM ファイルです。
.p12 ファイル	PKCS (Public-Key Cryptography Standard) 番号 12 に準拠する証明書ファイル。これはユーザーのプライベートキー、証明、機密情報、またはその他の情報を安全に保管したり移動したりするためのポータブルフォーマットを指定します。
.pcp ファイル	Windows Installer パッチ作成プロパティ ファイル。
.prq ファイル	メインのアプリケーションがインストールされる前にターゲットマシン上に既にインストール済みでなくてはならない、ベースアプリケーションまたはコンポーネントについての情報を含む前提条件ファイル。インストール プロジェクトにセットアップ前提条件を含めると、開発者は複数のインストーラー ファイルを単一の実行可能ファイルに連鎖させることができます。
.reg ファイル	「レジストリ ファイル」を参照してください。
.scm ファイル	ScreenCam ムービー ファイル。
.wmf ファイル	Windows メタファイル形式のファイル。この種類のグラフィック ファイルは Microsoft Windows ベースのアプリケーション間でグラフィック情報を交換するのに使用されます。

テーブル 1・InstallShield Express 用語集

用語	定義
ActiveX	オブジェクト指向プログラミング技術およびそのツールのセット。ActiveX を利用すると、開発者は他のアプリケーションやオペレーティング システムが呼び出すようにアプリケーションを作成することができます。ActiveX 技術はスタティックページではなく、アプリケーションのような外観を持ち作動するインタラクティブ Web ページの作成を可能にします。
AlwaysInstallElevated	HKEY_LOCAL_MACHINE¥Software¥Polices¥Microsoft¥Windows¥Installer および HKEY_CURRENT_USER¥Software¥Polices¥Microsoft¥Windows¥Installer レジストリ キーの下で Windows プラットフォーム用に構成可能なユーザー ポリシー。Microsoft¥Windows¥Installer and HKEY_CURRENT_USER¥Software¥Polices¥Microsoft¥Windows¥Installer.ハイレベルなシステム権限でパッケージをインストールするには、これら両方のレジストリキーの下にある AlwaysInstallElevated の値を 1 に設定します。
American Standard Code for Information Interchange (ASCII)	各キャラクタに 0 から 127 が割り当てられている、文字、数字、句読点および他のキャラクタを表すコード。このコードによって、異なる種類のコンピュータとコンピュータ アプリケーション間でデータを交換できます。
API	「アプリケーション プログラミング インターフェイス」を参照してください。
ASCII	「American Standard Code for Information Interchange (ASCII)」を参照してください。
ATL	「アクティブ テンプレート ライブラリ」を参照してください。
Binary テーブル	ビットマップやアイコンなどのバイナリ データを収納するテーブル。カスタム アクションのデータも含みます。
CBT	「コンピューター ベース トレーニング」を参照してください。
CD	「コンパクト ディスク」を参照してください。
CD ブラウザー	CD がドライブに挿入された時に起動されるグラフィックユーザー インターフェイス。CD に含まれる 1 つまたは複数のアプリケーションを起動するのに利用されます。
CD-ROM	「コンパクト ディスク 読み取り専用メモリ」を参照してください。
COM	「コンポーネント オブジェクト モデル」を参照してください。
COM サーバー	コンポーネントオブジェクトモデル仕様に従って他のアプリケーションにオブジェクトを公開する実行可能ファイル。COM サーバーにより公開されるオブジェクトの種類には、ActiveX コントロール、ActiveX ドキュメント、Automation オブジェクト、MTS コンポーネントがあります。

テーブル 1・InstallShield Express 用語集

用語	定義
COM ファイル	製品間で共有できる再利用可能なコードが含まれるバイナリ ソフトウェア コンポーネントです。COM ファイルの代表的な例は、Word 文書に埋めこまれた Microsoft Excel スプレッドシートです。この場合 Excel が COM サーバーとして作動し、Word はそのクライアントとして作動します。
COM+ ファイル	拡張 COM ファイル。インストールに関して、COM+ は配布されたコンポーネントのビルド用にさらなる拡張サポートを追加します。たとえば、appID キーおよび値をレジストリに作成し、これによって様々な COM コンポーネントをリモートで実行または権限つきで実行するよう構成することが可能です。
DBMS	「データベース管理システム」を参照してください。
DCOM	「分散コンポーネント オブジェクト モデル」を参照してください。
DCOM ファイル	ネットワーク全体で通信できるソフトウェア コンポーネント。DCOM (以前はネットワーク OLE と呼ばれる) によって、1 つのアプリケーションのコンポーネントを複数のネットワーク コンピューターに渡って配布できるようになります。
DLL	「ダイナミック リンク ライブラリ」を参照してください。
EXE メディア タイプ	単一の自己展開型ファイルに含まれるすべてのインストールまたはアンインストールファイル。
GUI	Graphical user interface (グラフィカル ユーザー インターフェイス)。
GUID	「Globally Unique Identifier (GUID)」を参照してください。
HTML	「hypertext markup language (ハイパーテキスト マークアップ言語)」を参照してください。
IAT	「インポート アドレス テーブル」を参照してください。
ICE	「Internal Consistency Evaluator (内部整合性評価プログラム)」を参照してください。
IDE	「インタラクティブ開発環境」 または 「統合開発環境」を参照してください。
IDriver.exe	InstallShield スクリプト ランタイム エンジン。IDriver.exe は、InstallShield で作成したインストールの一部を実行するためにコンピューター上で必要です。IDriver.exe は、デフォルトで、C:¥Program Files¥Common Files¥InstallShield¥Driver¥7¥Intel 32 と C:¥Program Files¥Common Files¥InstallShield¥Driver¥8¥Intel 32 の両方またはいずれかの共通ファイルの場所にあります。一部のインストール エラー メッセージは、IDriver.exe を参照する場合があります。
IE	「インターネット エクスプローラー」を参照してください。
ikernel.exe	InstallShield エンジン。「エンジン」も参照してください。

テーブル 1・InstallShield Express 用語集

用語	定義
INSTALLDIR	インストールのルート インストール先ディレクトリを指定するシステム変数。
INSTALLLEVEL	デフォルトで機能のインストールは [ON] に選択されている初期レベル。
InstallShield スクリプト ランタイム	「IDriver.exe」を参照してください。
ISM	「.ism ファイル」を参照してください。
ISScript.msi	InstallScript エンジンインストーラー。ISScript.msi はインストールを実行するのに必要なファイルをインストールします。InstallScript エンジンは、InstallShield スクリプト ランタイムとしても知られています。InstallScript はインストールを作成するのに利用されるプログラム言語です。
JAR	Java アーカイブ ファイルの頭文字。「.jar ファイル」も参照してください。
Java アーカイブ	「.jar ファイル」を参照してください。
Java 仮想マシン (JVM)	バイトコードにコンパイルされた Java プログラムを解釈する擬似コンピューター。Java プログラムは通常 .class ファイルに保存されています。
Java ネイティブ インターフェイス (JNI)	共有ライブラリのネイティブコードを呼び出すための Java コードを有効にする Java の仕様。JNI ライブラリのネイティブコードは、JNI を通して Java コードを呼び出すこともできます。
JNI	「Java ネイティブ インターフェイス」を参照してください。
JPEG	「.jpg ファイル」を参照してください。
JVM	「Java 仮想マシン」を参照してください。
KB	「ナレッジ ベース」を参照してください。
LogDB	「ログ データベース」を参照してください。
MD	「メディア ディスクリプタ」を参照してください。
MFC	「Microsoft Foundation Classes」を参照してください。
Microsoft Developer Network (MSDN)	Microsoft の製品およびテクノロジを使用してアプリケーションを作成している開発者を支援するためにデザインされたオンラインおよびオフライン サービス。
Microsoft Foundation Classes (MFC)	開発者が Win32 アプリケーションを書く際に使用することができる大規模な C++ クラスライブラリ。MFC を使用するアプリケーションを実行するには、MFC エンジンがターゲットマシンにインストールされている必要があります。

テーブル 1・InstallShield Express 用語集

用語	定義
Microsoft ディスク オペレーティング システム (MS-DOS)	Microsoft が開発した一番最初のオペレーティング システム。MS-DOS は、Windows 95、98 および ME の基礎となるオペレーティング システムです。Windows NT、2000 および XP システムは存在する DOS アプリケーションをサポートしています。
Microsoft ビルド エンジン (MSBuild)	SQL Server と完全互換性があるデスクトップおよび共有ソリューションのビルドのためのデータ エンジン。SQL Server 7.0 への簡単な移行パスを提供します。MSDE で構築されたソリューションは、コードを 1 行も変えずに SQL サーバ 7.0 に完全に移行することができます。
MIDI	「musical instrument digital interface」を参照してください。
MIME	「multipurpose internet mail extensions」を参照してください。
MM	「マージ モジュール」を参照してください。
MS-DOS	「Microsoft ディスク オペレーティング システム」を参照してください。
MS-DOS プロンプト	MS-DOS(Microsoft Disk Operating System)プロンプトは、MS-DOS が新しいコマンドを受け付けられる状態にあることを示す [コマンドプロンプト] ウィンドウ内のビジュアル インジケーターです。デフォルトの MS-DOS プロンプトは、通常、C:> プラス点滅カーソルです。
MSDE	「Microsoft Data Engine」を参照してください。
MSDE オブジェクト テンプレート	MSDE マージ モジュールのベース。MSDE マージ モジュールを Windows Installer プロジェクトまたは InstallScript プロジェクトに追加すると、InstallShield アプリケーションは、このマージ モジュールのベースファイル(または、スタブ)と共に起動します。アプリケーションが、ベースファイルに設定を適用できるように、このマージ モジュールを構成する必要があります。
MSDE 名前付きインスタンス	MSDE (Microsoft Data Engine) インストールの名前付きインスタンス。「名前付きインスタンス」も参照してください。
MSDN	「Microsoft Developer Network」を参照してください。
MSI	「.msi ファイル」を参照してください。
MSM	「.msm ファイル」を参照してください。
MSP	「.msp ファイル」を参照してください。
MST	「.mst ファイル」を参照してください。
multipurpose Internet mail extensions (MIME)	Web クライアントに送るファイルを識別する Web サーバーが使用する規格。MIME 規格は、送るときのファイルの種類およびそれをオリジナルの形式に戻す方法を指定する方法です。

テーブル 1・InstallShield Express 用語集

用語	定義
musical instrument digital interface (MIDI)	音楽をデジタルシンセサイザーで録音/再生するために設計されたプロトコル。実際のサウンド (音声または音楽) をデジタル録音する .wav ファイルとは異なり、MIDI ファイルは単純に楽器、旋律、および、それをどのように再生するかを定義するだけです。エンドユーザーのシステム上にあるシンセサイザ (一般に MIDI 可能なサウンドボードの一部) は MIDI 指示を読み込んで、音楽を再生します。MIDI ファイルには実際の音楽ではなく命令のみが含まれているので、サイズは同じ長さのデジタル オーディオ ファイルに比較して何倍も小さくなっています。また、MIDI ファイルは再生においても、より少ないプロセッサパワーが必要です。
NT サービス	サービスとして実行するために Windows NT にインストールされたアプリケーション。つまり、インストールされた後は自動的に実行され、ユーザーには表示されず、サービスを開始するためにユーザーがログインする必要もありません。NT はレジストリで定義、および記述されているサービスを管理します。
One Really Cool Application (ORCA)	Windows Installer ファイルを編集するための外部ツール。
One-Click Install	One-Click Install インストールを使用すると、エンドユーザーは最小限の労力でアプリケーションをダウンロードし、すぐに使用することができます。ユーザーによるインプットは最小限で、インストールが自動的にダウンロードされ、圧縮解除され、そして実行されます。エンドユーザーにダウンロード場所を問い合わせることもなく、インストールを手動で起動する必要もありません。
ORCA	「One Really Cool Application」を参照してください。
OS	「オペレーティング システム」を参照してください。
PKCS	「public-key cryptography standards」を参照してください。
public-key cryptography standards (PKCS)	RSA Security Inc.によって開発され、ここで定義されている仕様は、パブリックキー暗号化に関する既存の規格団体が扱っていない部分を標準化するために開発されました。
QuickPatch	アプリケーションのファイルおよびインストールを特定バージョンにアップデートするのに必要なデータベースの一部のみを含む、特殊な種類のパッチ パッケージ。QuickPatch プロジェクトを使用して、InstallShield で QuickPatch を作成することができます。
Readme ファイル	重要な情報を含むアプリケーションの配布に含まれているテキストファイル。この情報は通常、インストール、アンインストール、または機能に関する、他のドキュメントに含まれていない可能性がある情報です。
red green blue (RGB)	混ぜて他の色を作成することができる光の三原色。色付きのイメージは通常、RGB 三重項のシーケンス、または、別々の赤、緑、青のオーバーレイとして格納されますが、これらに限りません。これらの三色は、カラー ブラウン管 (CRT) の三本の ガン と人間の眼の色レセプターに相当します。

テーブル 1・InstallShield Express 用語集

用語	定義
Regedt32.exe、Regedit.exe	Windows Registry Editor の 2 つのバージョン。レジストリ エディターを使用して、レジストリ内のエントリの編集をすることができます。Regedt32 には、より多くのレジストリ編集機能が用意されています。インストール中にエラーが発生した場合、インストールを完了するためにレジストリの編集が必要な場合があります。Windows XP コンピューターでは、Regedt32.exe と Regedit.exe が既に融合されているので、新しいオペレーティング システム上では、これらの 2 つのアプリケーションは同じ機能を実行します。
RGB	「red green blue」を参照してください。
RPC スタブ	「リモート プロシージャ コール スタブ」を参照してください。
RTF	「リッチ テキスト フォーマット」を参照してください。
SCM	「サービス コントロール マネージャー」を参照してください。
SDK	「Software Development Kit (ソフトウエア開発キット)」を参照してください。
Section 508 (米国リハビリテーション法 508 条)	リハビリテーション法を改訂して、連邦政府各機関に対して障害を持つ人々もその電子情報技術にアクセスできるように要求した米国の法律。508 条は、情報技術におけるバリアの除去、障害を持つ人々に新たな機会の提供、およびこれらの目的を達成することを支援する技術の開発を推進することを目的に制定されました。この法律は、電子情報技術を使用、開発、購入、メンテナンスをするすべての連邦政府各機関に適用します。508 条の下で連邦政府各機関は、障害を持つ被雇用者および一般の人々に対して他の人々と同様の情報へのアクセシビリティを提供することが義務付けられています。
Setup.inx	コンパイルされたスクリプト ファイル。インストーラー エンジンが実行するオブジェクト コード。
SP	「サービス パック」を参照してください。
SQL	「Structured Query Language (構造化クエリ言語)」を参照してください。
SQL ステートメント	キーワードで始まり実行されるアクションを完全に記述する SQL の完全フレーズ。例、SELECT * FROM Orders。
TARGETDIR	Windows Installer ベースのインストールでは、TARGETDIR プロパティはインストールのルート インストール先ディレクトリを指定します。InstallScript インストールでは、TARGETDIR システム変数はデフォルトでインストールのルート インストール先ディレクトリを指定します。
UI	「ユーザー インターフェイス」を参照してください。
uniform/universal resource locator (URL)	Web サイト/ページの場所を示します。

テーブル 1・InstallShield Express 用語集

用語	定義
unInstallShield	エンド ユーザーの PC から製品をアンインストールするセットアップ ウィザードに含まれているコンポーネント。
Update Manager (Windows または Java)	エンドユーザーがアップデートのスケジュールを設定する際、またはエージェントによって実行されるアップデートのクエリをサイレンスにするために使用することができるオプションの Win32 または Java ベースのクライアント アプリケーション。
URL	「uniform/universal resource locator (URL)」を参照してください。
VP	「バリューパック」を参照してください。
Web installation (Web インストール)	エンドユーザーが Web サイトにアクセスして、そこから実行するインストール。
Web サイト	World Wide Web 上の場所
WinDir	Win16 on Win32 (WOW) 用の実行可能ファイルへのパスを格納します。
Windows API	Windows アプリケーション プログラム インターフェイス (API) は、Windows プラットフォーム用に書かれたアプリケーションが利用するビルドブロックを提供します。各 API はコンピューター オペレーティング システム、またはアプリケーション プログラムが指定する特定のメソッドです。アプリケーションを作成するプログラマーは、Windows API を利用してオペレーティング システム、または別のアプリケーションを要求することができます。各 API を適切に実行するための要件は異なります。
Windows Installer	(1) インストールおよび構成サービス。Microsoft Windows Installer は、データ ドリブン モデルのエンジンで、すべてのインストールデータおよび命令が 1 つの完結したパッケージになっています。データ ドリブン インストール モデルでは、インストール テーブルのプライマリセットは、すべてのアプリケーション リソース(ファイル、レジストリキー等)がそれがサポートするコンポーネントまたは機能に明確に関連しているところに作成されます。ユーザーがインストールするオブジェクト、インストールする場所を選択すると、その手続き命令は Windows Installer によって処理されます。 (2) .msi パッケージのサービス、プロパティ、およびテーブルを意味します。
Windows Installer サービス	アプリケーションインストールの構成、およびアプリケーションのアンインストールを一元的に管理するオペレーティング システム コンポーネント。
Windows NT サービス	「Windows サービス」を参照してください。
Windows インストール CD	Windows オペレーティング システムを含むコンパクトディスク。コンピューター製作会社によってコンピューターにオペレーティング システムがインストールされた場合、通常マニュアルと共にインストール CD が付属されています。自分で Windows オペレーティング システムをインストールした場合、そのときに使った CD がインストール CD です。

テーブル 1・InstallShield Express 用語集

用語	定義
Windows サービス	それ自身の Windows セッションで長期にわたって作動する実行可能アプリケーション。例、イベント ログにメッセージを書き込むアプリケーション。
Windows システム フォルダ	Windows System フォルダーは、コンピューターが適切に稼動し続けるために必要なコア オペレーティング システムを含みます。インストール中に発生するエラーの原因としては、System フォルダーに含まれているファイルが破損しているか、または存在していないことが考えられます。
WYSIWYG	What You See Is What You Get (見たとおりのものが結果に反映されること) の略 。
XCopyFile	1 つ、または複数のファイルおよび/またはサブディレクトリを、必要に応じてターゲット マシンにサブディレクトリを作成して、ターゲット ディレクトリにコピーする InstallScript 関数。
XML	「拡張マークアップ言語」を参照してください。
XSL	「拡張スタイルシート言語」を参照してください。
アクション	インストールまたはアンインストール ウィザードの実行中に特定の時点で処理を行なうコマンド。インストール パッケージの開発者はビルトイン標準アクションを利用、および独自のカスタム アクションを作成することができます。アクションはエンドユーザーに対して処理状況を表示したり、処理をキャンセルできるようにすることが可能です。
アクセシビリティ	障害者を含めてすべての人々に対する有用性、または利用の可能性。
アクティブ ディレクトリ	管理者がネットワーク上に存在するオブジェクトを検索することができる Microsoft Windows 2000 がサポートする構造。アクティブ ディレクトリは、分散コンピューター環境用 Windows 2000 Server で使用されるディレクトリです。
アクティブ テンプレート ライブラリ (ATL)	たとえば MFC などのバイナリよりも小さい COM コントロールを書くための一連のクラス。
圧縮	1 つまたは複数のファイルを基に、オリジナルファイルの合計サイズよりも小さいサイズの新しいファイルを作成します。この新しいファイルからオリジナルファイルを複製することが可能です。
圧縮解除	「伸張」を参照してください。
アップグレード	アプリケーションの古いバージョンからより最新のバージョン(通常、アプリケーションの改善版)への移動。
アップグレードコード	既存ソフトウェアの新しいバージョンをインストールするために必要なコード。
アップグレード	既存ソフトウェアの新しいバージョンをインストールするための処理。

テーブル 1・InstallShield Express 用語集

用語	定義
アドバタイズ	アプリケーションのインストール処理中に機能がインストールされない「ジャスト イン タイム」インストール。その代わり、要求された時に直ちにインストールされます。機能のアドバタイズを有効にすると、アドバタイズを阻止するその他の要素がない限り、インストールが実行されているモードにかかわらず機能はアドバタイズされます。/jm オプションを使ってコマンドラインから MsiExec.exe を起動すると、機能はエンドユーザーのマシンにアドバタイズされます。/ju 関数を利用すると、機能は現在のエンドユーザーにアドバタイズされます。Custom Setup ダイアログで、エンドユーザーはすぐにインストールする機能とあとで使用する機能を制御できます。アドバタイズには assigning (割り当て) および publish (パブリッシュ) の 2 つの種類があります。「割り当て」、および「パブリッシュ」も参照してください。
[アドバタイズ] シーケンス	[アドバタイズ] シーケンスには、ユーザーが /jコマンドライン オプションを使ってインストールを起動したときに実行されるアクションのリストが含まれます。検証規則 ICE78 では、[アドバタイズ]-[ユーザー インターフェイス] シーケンスが空でなくてはなりません。
アドバタイズ ショートカット	エンドユーザーがショートカットを初めて起動するまでインストールされない、製品または機能へのショートカット。インストールの実行時、[エンドユーザーがショートカットを持つ製品または機能に対して 要求があった場合に、この機能をインストールします] オプションを選択した場合、ショートカットが作成されますが、コンポーネントのファイルはユーザーがショートカットを起動するまでインストールされません。ショートカットを初めて起動すると、Windows Installer サービスはコンポーネントのファイルと他のデータをインストールし、その後、ショートカットによってターゲット ファイルが起動されます。その後、ショートカットを使用すると常に、通常のショートカットと同じように動作します。
アドバタイズ	アプリケーションのインストール処理中に直ちにインストールされない機能をエンドユーザーに向けて表示する処理。その代わり、これらの機能は要求された時に直ちにインストールされます。「アドバタイズ」も参照してください。
アプリケーション パス	システムのパスにアプリケーションとその .dll ファイルの場所が指定されていない場合に、Windows がそれらの検索に使用するレジストリキー。app パスエントリは、コンポーネントの詳細設定で設定することができます。
アプリケーション プログラミング インターフェイス (API)	アプリケーションがコンピューターのオペレーティング システムとコミュニケーションをとるために利用する、またオペレーティング システムがアプリケーションに対してサービスを有効にするために利用する一連のルーチン。
アンインストール	インストールを取り消す処理です。アンインストールは、エンドユーザーが製品ファイルを削除し、インストール中にマシンに対して加えられた変更を元に戻すインストールのメンテナンス オプションです。
アンインストール ログ ファイル	インストール中に発生するすべてのアンインストール関連のイベントの記録です。ログファイルはインストールの始めに初期化されます。このログファイルが破損した場合、アンインストール中にエラーが発生する可能性があります。

テーブル 1・InstallShield Express 用語集

用語	定義
移行	1 つのオペレーティング環境から他の一般的により優れていると思われているオペレーティング環境に移るプロセス。たとえば、Windows NT Server から Windows 2000 Server へ変えることも、このプロセスによって新しい機能が生かされ、古い設定は今後変更が必要なくなり、現在のアプリケーションが継続して新しい環境で動作するようにいくつかの手順がとられるので、移行と考えることができます。また、Windows NT から UNIX ベースのオペレーティング システム (または、その逆)も移行と考えられます。また、1 つのデータベースを違う種類のデータベースに移行することも可能です。このためには、通常、データを古いデータベースからアウトプットして新しいデータベースにインプットできる共通のフォーマットにデータを変換する必要があります。また、データを 1 つの記憶装置から別の記憶装置に移すといった単純なプロセスも移行といいます。
依存関係	別のソフトウェア オブジェクトが必要とするソフトウェア オブジェクト。
一時ディレクトリ	オペレーティング システムまたはアプリケーションを使用中に、そのファイルを一時的に保管するハードドライブ上のフォルダー。アプリケーションが終了したとき、一時ファイルは削除されます。一時ディレクトリから手作業でファイルをクリーン、または削除する必要がある場合もあります。
インストーラー	コンピューター上でのインストールおよび構成サービス。
インストーラー データベース	アプリケーションのインストールに必要な情報すべてを含むデータベース。アプリケーションのグループをインストールするのに必要な情報を含む関係データベースを構成する、多くの相互に関係するテーブルで構成されます。
インストーラー プロパティ	インストール中に利用される変数。
インストーラー関数	アプリケーションがインストーラー サービスを取得するために呼び出すアプリケーション プログラム インターフェイス。
インストール	プログラムとその構成ファイル、機能、およびコンポーネントをソース メディアからターゲット システムへ転送すること。「インストール プロセス」も参照してください。
インストール オン デマンド	ファイルが存在しない機能でも、ユーザーやアプリケーションに対してその機能の提供を可能にする Windows Installer の機能性。
[インストール] シーケンス	実行時に、アプリケーション ファイルのインストール、アクションの実行、およびダイアログの表示が行われる順番(シーケンス)。[インストール] シーケンスは、たとえば インストールのランチャをダブルクリックするなどしてインストールを起動したときにデフォルトで実行されます。
インストール タイプ	エンドユーザーがインストールまたはアンインストールを選択することができる、予め定義された機能のグループ。

テーブル 1・InstallShield Express 用語集

用語	定義
インストール パッケージ	アプリケーションのインストールまたはアンインストールのために合わせて使われる 1 つまたは複数のファイル。
インストール パッケージ オーサリング ツール	ユーザーがインストールパッケージを作成するのに利用できるサードパーティ ツール。
インストール プロジェクト	インストール作成中、インストールを構成するソースファイル、ダイアログ、アクションおよび条件の集合体。
インストール プロセス	パッケージまたはアプリケーションのコンテンツをターゲット コンピューターにインストールする全プロセス。インストール処理は、取得段階、実行段階、およびインストールに失敗した場合、ロールバックの 3 つの段階で構成されています。
インストール レベル	機能のインストールレベルは、インストールのデフォルトでそれが選択されているかどうかを部分的に判断します。その値は Windows Installer プロパティ の INSTALLLEVEL と比較され、どの機能のインストールが選択されたかを判断します。機能のインストール レベル プロパティが INSTALLLEVEL の値と同等またはそれ以下の場合、その機能がインストールされます。
インターネット エクスプローラー (IE)	Microsoft によって開発され、Windows オペレーティング システムと共に配布されるブラウザー。
インターフェイス	独立システムが相互にコミュニケーションをとるポイント。アプリケーションのユーザー インターフェイスは一般的にツールバー、メニュー、ボタン、ウィンドウ、およびその他のアイテムで構成されます。
インタラクティブ開発環境 (IDE)	ソフトウェア プログラムのインターフェイス。「統合開発環境」とも呼ばれます。
インポート アドレス テーブル (IAT)	.dll ファイルまたは実行可能ファイルをインポートするのに使用されるテーブル。
ウィザード	(1) ステップ バイ ステップ方式でユーザーが不慣れなタスクを完了することができる対話型ガイド プログラム ユーティリティ。 (2) InstallShield Wizard はインストールまたはアンインストールの実行時に表示されるウィザードです。インストールまたはアンインストールの手順をエンドユーザーにフレキシブルに案内する、予め定義されたひとまとまりのダイアログのことを指します。
ウィザード インターフェイス	実行時にエンドユーザーがウィザードとインタラクトするために使用されます。
ウィザード ダイアログ	インストール/アンインストールの最中にエンドユーザーに表示されるダイアログ。情報ウィンドウによって、エンドユーザーは情報を読み込んだり指定したりしながらオペレーションとインタラクトできます。

テーブル 1・InstallShield Express 用語集

用語	定義
ウィンドウ	開かれたアプリケーションがスクリーン上に表示される領域。
エンジン	基本の関数を実行し、その他のプログラムの処理を全体的に調整するプログラム。エンジンは内部的に動作します。InstallShield は、インストールを駆動するために InstallScript エンジン (ISScript.msi) および Windows Installer エンジンを利用します。InstallShield はまた、ikernel と呼ばれる独自のエンジンも装備しています。
エンドユーザー	製品をインストールまたはアンインストールするユーザー。
エンドユーザー マシン	エンドが製品をインストールまたはアンインストールするマシン(コンピューター)。
オートノミック コンピューティング	人体の自律神経系にちなんで名づけられ、またそのパターンに従った自己管理コンピューティング モデル。オートノミック コンピューティング システムは、ユーザーがインプットすることなくコンピューター アプリケーションおよびシステムの機能をコントロールします。これは人体の自律神経系が身体を調節および保護するのと同じ原理です。
オブジェクト	(1) 個別の機能をインストールするために必要なロジックとファイルがすべて含まれたパッケージ。
オペレーティング システム (OS)	メモリの中のストレージ スペースを割り当てたり、入力または出力関数を制御したりすることで、コンピューターのオペレーションを制御し、プログラムの処理を管理するソフトウェア。
親/子インストール	「ネスト インストール」を参照してください。
外部ユーザー インターフェイス	インストールパッケージの作成者によって開発されたユーザー インターフェイス。インストーラーで利用可能な内部ユーザー インターフェイス機能は利用しません。
概要情報ストリーム	Windows Installer パッケージのために定義されアプリケーションのインストール時にインストーラーによって使用されるタイトル、作成者、パッケージ コード、テンプレート、概要およびスキーマ等のプロパティ。
拡張可能スタイルシート言語 (XSL)	XML 情報をどのように表示するかを記述するのに使われる言語。
拡張マークアップ言語 (XML)	(1) 実質的に Standard Generalized Markup Language (SGML) を簡素化したプログラム言語。開発者はこれを使用してカスタマイズされたタグを作成し、効率良くコンテンツを編成および配布することができます。 (2) MultiPlatform プロジェクトの出力フォーマットです。 (3) InstallShield で作成された Windows ベースの プロジェクトは、ソースコード管理ソフトウェアで保存できるように XML に変換することも可能です。

テーブル 1・InstallShield Express 用語集

用語	定義
カスタム アクション	Windows Installer 組み込まれた標準アクションに対して、インストール作成者によって作成されたアクション。アクションは、アプリケーションのインストール、アンインストール、またはメンテナンス中に実行される関数をカプセル化します。
簡易ユーザー インターフェイス (UI)	簡易 UIは、インストール時に作成されたモードレス ダイアログ、ビルトイン モーダル エラー メッセージ、および ディスク プロンプト メッセージを表示します。この UI レベルは、作成されたモーダル ダイアログの表示はしません。この UI レベルでは、インストーラーの内部ユーザー インターフェイス 機能が使われます。「モーダル」、「モードレス」および「内部ユーザー インターフェイス」も参照してください。
環境変数	ターゲット システム上の複数のプログラムでアクセスできる変数。
間接ビルド	ペイロードファイルのみにリファレンスを作成するフルビルド。つまりインストール時、ペイロードはビルド時にあった場所と同じ場所になくてはなりません。たとえば、ペイロードがネットワーク サーバーにある場合、インストールは同じネットワーク サーバー上でペイロードを探します。
完全アップデート	製品のアップデート バージョンで以前のバージョンを上書きするインストール。
完全ユーザー インターフェイス (UI)	インストーラーの内部ユーザー インターフェイス (UI) 性能レベルのひとつ。完全 UI レベルでビルドされるインストール パッケージは、内部 UI に含まれるモーダル ダイアログおよびモードレス ダイアログの両方を表示することができます。
管理	物事を管理する処理またはその動作。
管理インストール	管理インストールはデータファイルをユーザーが指定したディレクトリにコピー (および展開) しますが、ショートカットの作成や COM サーバーの登録、またはアンインストールログは作成しません。
[管理] シーケンス	ユーザーが /a コマンドライン オプションを使ってインストールを起動した場合に実行されるアクションのリスト。これは、ネットワーク管理者がネットワーク上に共通のインストールポイントを提供し、複数のユーザーに対して異なるインストール基準を設定する場合に便利です。各 [管理] シーケンスは、[ユーザー インターフェイス] シーケンスと [実行] シーケンスから構成されます。
管理者権限	コンピューター ユーザーに与えられる最も高いレベルの権限。ネットワーク環境では、システムのセキュリティを確保し、コンピュータのハードウェアおよびソフトウェアに対するダメージを防ぐために、異なるレベルのアクセス許可が必要です。管理者権限を持つユーザーはソフトウェアのインストール及びアンインストール、並びにコンピューターの構成を変更することが可能です。管理者権限は通常 Windows NT 4.0、2000、または XP マシンに付属していますが、Windows 95、98、または ME マシンには付属しません。
キー パス	コンポーネントの有無を検出するために Windows Installer が使用する各コンポーネントの一意のレジストリ値。コンポーネントにはキーファイルまたはキーパスのいずれかを割り当てることができますが、両方を割り当てることはできません。

テーブル 1・InstallShield Express 用語集

用語	定義
キーファイル	コンポーネントの有無を検出するために Windows Installer が使用する各コンポーネント専用に 1 つだけ存在するファイル。コンポーネントの詳細設定やショートカットを作成するには、キーファイルを指定する必要があります。
機能	製品の機能についての論理表現。たとえば、インストールにデータベース、メイン アプリケーション ファイル、およびヘルプ システム ファイルを含む場合があります。データベース、アプリケーション、およびヘルプシステムはすべて製品の機能です。機能はエンドユーザーに表示されますが、機能を構成する各ファイルは機能のコンポーネント内部にあります。「コンポーネント」も参照してください。
基本の UI	インストーラーの内部ユーザー インターフェイス (UI) 性能レベルのひとつ。一般的に、基本の UI レベルを使ってビルドされるインストールパッケージは処理状況メッセージおよびディスクプロンプト メッセージを表示するモードレスのビルトインダイアログを表示します。作成されたダイアログは表示しません。
キャッシュ	*cash* と同じ読み方で、頻繁にアクセスされるデータの一時的な格納場所。キャッシュの目的は、頻繁にアクセスされる情報を簡単にアクセスできる場所へ保存することで、コンピューターの効率を高めることです。キャッシュには、メモリ キャッシュとディスク キャッシュの 2 種類があります。メモリキャッシュはデータおよびメインメモリ内でデータが保存されているアドレスを保存します。ほとんどのアプリケーションは同じデータに繰り返しアクセスするため、メモリキャッシュは便利です。ディスクキャッシュはメインメ モリを使用します。ハード ディスクが最近要求した情報、またはハード ディスクに最近書き込まれた情報を保存するのに使用されます。一般に、インストールは通常ディスク キャッシュを使用します。データがメイン メモリから読み出される、または書き込まれるとき、そのコピーがメモリキャッシュに保存されます。データが呼び出されるとき、コンピューターはまずメモリキャッシュをチェックしてからディスクキャッシュ、最後にメインメモリをチェックします。
キャビネット ファイル	いくつかの圧縮ファイルを含む単一ファイル。アプリケーションのインストール中、圧縮ファイルは圧縮が解除され、コンピューターにコピーされます。キャビネット ファイルはソフトウェア配布中に容量と時間を節約できるという点で効率的です。キャビネット ファイルは通常ファイル拡張子 .cab を持ちます。ファイルが破損または不足している場合はインストールの完了を妨げることになります。オペレーティング システム ファイル、または InstallShield ファイルの不足または破損ファイルは、キャビネット ファイルから抽出して置換する必要があります。
キャプション	ウィンドウの表題にあたるテキスト。ウィンドウキャプションは、スクリーン上でアクティブウィンドウの位置を変更する際にユーザーがクリック アンド ドラッグする部分です。
ギャラリー	(4) 共有可能な使用できるリソースのコレクション。
行	特定の要素を記述する関連する列のセット。*レコード*とも呼ばれます。

テーブル 1・InstallShield Express 用語集

用語	定義
グローバル一意識別子 (GUID)	製品を識別するために InstallShield が生成する長い数値の文字列。InstallShield で文字列 GUID を入力するとき、常に {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX} という形式を使用します。
結果セット	SELECT ステートメント の実行によって作成される行のセット。
検証	検証処理。データが適切な形式であることを確認すること。
構造化クエリ言語 (SQL)	データベースを操作するときに使用される言語。
コスト分析	ファイル コスト分析は、現在のインストールで必要なディスク容量の合計を判断するために InstallShield が利用する処理です。
コマンド	(1) コンピューターに与えられる指示。 (2) 特定のタスクを実行するために オペレーティング システムがコマンドラインから、または [コマンドプロンプト] ウィンドウから実行する指示。インストール中にエラーが発生した場合、たとえばユーティリティを実行するため、コンピューターのディレクトリを検索するため、またはファイルを削除するためなどの目的でコマンドを実行する必要がある場合があります。
コマンド プロンプト ウィンドウ	[コマンドプロンプト] ウィンドウは MS-DOS へのインターフェイスです。[コマンドプロンプト] ウィンドウへアクセスするには、[スタート] メニューで [ファイル名を指定して実行] をクリックし、Windows 95、98、および ME では **command.com** を入力、Windows NT、2000、または XP では **cmd.exe** を入力します。インストール中にエラーが発生した場合、たとえばコマンドを入力してユーティリティを実行する、またはコンピューターのディレクトリを検索するなどの目的で [コマンドプロンプト] ウィンドウを開く必要があります。
コマンドライン	コマンドが入力される [コマンドプロンプト] ウィンドウの領域。コマンドラインは、実行可能ファイル (例、.exe、.bin) にコマンドを渡すのに利用されます。
コマンドライン オプション	コマンドラインオプションは、コマンドの実行方法を変更するコマンドへの引数です。「コマンド」も参照してください。
コミット実行	インストールがファイルの転送、COM サーバーの登録、およびショートカット並びにレジストリ エントリの作成を完了した時に発生する InstallFinalize アクションの完了と同時に Windows Installer アクションを実行します。
コンテキスト メニュー	コンテキストメニュー (右クリックメニューまたはポップアップメニューとも呼ばれる) は、デスクトップ、Windows エクスプローラー、またはアプリケーションにある項目を右クリックすると表示されるメニューです。
コンパクト ディスク (CD)	電子録音、保存、およびオーディオ、ビデオの再生、並びにその他のデジタル形式の情報に利用されるディスク。

テーブル 1・InstallShield Express 用語集

用語	定義
コンパクト ディスク 読み取り専用メモリ (CD-ROM)	読み取り専用のコンパクト ディスク。
コンピューター ベース トレーニング (CBT)	コンピューターを用いて行なわれる指導。人材を指導するために CBT ではよくグラフィック、サウンド、および動画が用いられます。
コンポーネント	インストール開発者の視点から見たアプリケーションの要素。コンポーネントはエンドユーザーには表示されません。エンドユーザーがインストール用の機能を選択すると、インストーラーによってその機能に関連したコンポーネントが判断され、それらのコンポーネントがインストールされます。アプリケーションのコンポーネントには実行可能バイナリファイル、データファイル、ショートカット、ヘルプ システムファイル、およびレジストリ エントリなどが含まれます。
コンポーネント オブジェクト モデル (COM)	コンポーネントベースのアプリケーションの作成を可能にする Microsoft が開発したソフトウェア アーキテクチャ。コンポーネントベースのアプリケーションは、その他のコンポーネントおよび他のアプリケーションが機能を利用できるようにし、プログラムに機能を追加します。
サービス	Windows Installer ベースまたは InstallScript ベースのプロジェクトでは、これはコンピューターがオンになっているとき常にバックグラウンドで実行されるプログラムです。サービスはソフトウェア インストール、プロセス モニターリング、ファイル転送、タスク スケジューリング、ネットワーク管理、その他多数のユーザー側の対話操作を必要としないタスクを実行します。Win32 では、サービスはサービス コントロール マネージャーによって管理されます。
サービス コントロール マネージャー (SCM)	サービスに使用するシステムのデータベースをメンテナンスし、これらのサービスを制御するインターフェイスを提供します。
サービス パック (SP)	既存の問題を修正し、製品強化を図るためのソフトウェア製品のアップデート。製品の次回のバージョンには前回までリリースされたすべてのサービスパックが取り込まれています。
再インストール	製品が既にマシンにインストールされている状態でインストールをもう一度実行すると、既存のファイル、ショートカット、およびレジストリ エントリが上書きされ、製品が再インストールされます。
再配布可能ファイル	インストールまたはアプリケーションと共に合法的に配布することができるマージ モジュール、オブジェクト、または他のファイル。
サイレント インストール	ユーザー インターフェイスまたはエンドユーザーの介入なしで実行されるインストール。

テーブル 1 • InstallShield Express 用語集

用語	定義
差分リリース	既存リリースの指定したセットに存在しないファイル (複数可) のみを含むリリース。差分リリースは、既存のリリースによってインストールした製品のバージョンをアップデートするのに使用します。
参照カウント	共有ファイルがインストールされるたびに増えるカウントの集計。HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥ に保存されます。Windows¥CurrentVersion¥SharedDLLs の下に保存されます。
シーケンス	インストール プロジェクトで連続して実行される 1 セットのアクションとダイアログです。
シーケンス テーブル	インストール プロセスの制御および実行順序を指定するアクションをリストするテーブル。
自己修復	「自動修復」を参照してください。
自己登録ファイル	Windows レジストリにそれ自身についての情報をエントリーして、アンインストール時にその情報を削除することができるファイル。他の種類のファイルも情報をレジストリにエントリすることなく使用することができます。自己登録ファイルをインストールするには、そのファイルを希望の場所にインストールし、それからコンピューターに登録します。
システム ポリシー	システムが準拠していなければならない規則。
システム権限	各ユーザーが利用できるシステム、プログラム、および関数。
実行可能 (.bin) ファイル	UNIX プラットフォームで実行可能なプログラムを含むファイル。
実行可能 (.exe) ファイル	Windows プラットフォームで実行可能なプログラムを含むファイル。
実行時	ターゲットマシンでアプリケーションをインストール/アンインストールするために、インストールが、インストーラーとインタラクトする時間。
実行スクリプト	Windows Installer インストール用の インストーラー アクション。実行スクリプトはインストールの取得段階で生成され、実行段階で実行されます。
実行段階	インストーラーのアクションが実行される段階。
自動修復	インストーラーがアプリケーションを元の状態に戻す自動復元。機能と関連付けられているファイルが不足、または破損している場合、アプリケーションの部分的、または完全な再インストールが必要な場合があります。
修飾コンポーネント	ポインターに類似する単一レベル インダイレクションのメソッド。修飾コンポーネントは、主として、並列機能を持つコンポーネントを複数のカテゴリーにグループ分けするのに使用されます。

テーブル 1・InstallShield Express 用語集

用語	定義
修復モード	修復モードでは、紛失または破損されたファイル、不正確なレジストリ エントリおよび自己登録ファイル等のすべての種類の不具合がインストールによってチェックされます。不具合が見つかった場合、インストールは修正を試みます。「メンテナンス モード」も参照してください。
取得段階	インストール処理で、インストーラーがデータベースを参照して指示を得る段階。取得段階の後に実行段階が続きます。
ショートカット	アプリケーションをポイントするファイル。ショートカットをクリックするだけでアプリケーションを素早く開くことができます。ショートカットは、通常 [スタート] メニューまたはデスクトップに配置されます。「アドバタイズ ショートカット」も参照してください。
昇格された権限	標準の権限よりも高いレベルの権限で、通常は一時的なもの。権限とは、システム上で特定の動作を行うことができる権限のことです。
条件	プロパティの値を固定値と比較する、またはプロパティが定義されているかどうかを判断する論理のステートメント。条件はオブジェクト、機能、コンポーネント、アクション、またはその他の条件と関連付けられている項目がインストール、または実行される場合 true 評価しなくてはなりません。
条件付きインストール	特定条件が満たされた場合のみ、その項目をインストールする。たとえば、オペレーティング システム条件がアプリケーションのインストールに関連付けられている場合、ターゲットマシンが特定のオペレーティング システムを実行している場合のみアプリケーションをインストールします。
消費/コンシューム	インストールがマージ モジュールまたは InstallShield オブジェクトなどの再配布可能ファイル パッケージに対して行なう処理。
進行状況バー	実行ファイルの進行状況を視覚的に表示するバー。
伸張	スペース文字を削除してファイルのサイズを最小化する圧縮解除プロセス。この処理はファイルを元の状態に戻します。
スキーマ	データ ソース内における現在のテーブルおよびビューの構造の記述。スキーマは各テーブルが持っている列、各列のデータ タイプおよびテーブル間の関係について説明します。
スパイウェア	個人または組織から知られることなしにその情報を集めることを手助けするテクノロジ。スパイウェアは、インターネット上で、個人のコンピューター上に置かれ、ユーザーについての情報をひそかに集め、それを広告主または他の関連団体に伝えるプログラムです。スパイウェアはソフトウェア ウィルスまたは新しいプログラムをインストールしたときの結果としてコンピューターにロードすることが可能です。正確には、ユーザーの知る知らないに限らずインストールされるデータ収集プログラムについて、ユーザーがどのデータが収集されていて、それが誰と共有されているかを理解している場合はスパイウェアとは呼びません。

テーブル 1・InstallShield Express 用語集

用語	定義
スプラッシュ画面	製品のインストールの起動中にエンドユーザーに表示されるイメージ。表示するイメージ、表示時間、ローカライズされたイメージの有無を指定することができます。
スモール アップデート	インストール済みパッケージと最新バージョンのパッケージのバージョン番号が同じ場合にパッケージをアップグレードするパッチ。
セーフ モード	セーフモードは、Windows 95、98、ME および 2000 で提供されているトラブルシューティング モードです。セーフモードでコンピュータを起動すると、オペレーティング システム、マウス、キーボードおよびディスプレイドライバのみがロードされます。コンピューターが何をしても起動しないときでも、セーフモードを使用すると起動できる場合があります。また、セーフ モードでは、オペレーティング システムをトラブルシュートして適切に機能していない箇所を見つけることができます。
制限付きパブリック プロパティ	インストール作成者が設定または変更可能範囲を制限することができるグローバル変数。制限付きパブリック プロパティの操作は、通常、システム管理者のみが行うことができます。制限は、安全な環境を維持するために使用されます。
製品	実際にインストールまたはアンインストールするアプリケーション。
製品コード	製品を一意に識別する文字列。
セキュリティ ツール	トロイの木馬、スパイウェア、アドウェアおよびハッカー ツールなど破壊的なペストをコンピューター上で検出し除去するプライバシー ツール。ウイルス対策およびファイヤー ウォール ソフトウェアを補完し、エンド ユーザーの既存のセキュリティを回避し個人情報を侵害することができる非ウィルス系の悪質なソフトウェアからのプロテクションを強化します。
絶対パス	指定されたドライブのルート ディレクトリを起点にしてファイル検索をするのに必要な情報をすべて含む絶対パス。たとえば、C:¥Program Files¥InstallShield は、C ドライブにインストールされたときの InstallShield フォルダへの絶対パスです。
セットアップ	「インストール」を参照してください。
セットアップ プロジェクト	「インストール プロジェクト」を参照してください。
ソース リスト	インストーラーがインストール ファイルを検索する場所を指定するリスト。ソースリストのエントリは、ネットワークの場所、URL およびコンパクト ディスクが可能です。
相対パス	現在のドライブ上の現在のフォルダーを起点にして、ファイル検索に必要な情報をすべて含むパス (例、InstallShield¥Support)。フォルダーは、現在のディレクトリに存在する場合のみ、相対パスで検索することができます。

テーブル 1・InstallShield Express 用語集

用語	定義
即時実行	インストール段階に使われる用語。一般的に InstallScript プロジェクトでインストーラーがスクリプトをビルドするときに直ぐに実行されるカスタム アクションを指します。
ソフトウェ ア開発キット(SDK)	開発者が特定のオペレーティング システムをターゲットしたアプリケーションおよびライブラリを作成することができるようにデザインされたドキュメント、サンプル、コマンドライン コンパイラ、デバッグ ツール、ユーティリティ、およびツール。SDK は通常、オペレーティング システムの製造元により提供されます。
ダイアログ	インストール/アンインストールの最中にエンドユーザーに表示される情報ウィンドウ。情報ウィンドウによって、エンドユーザーは情報を読み込んだり指定したりしながらオペレーションとインタラクトできます。「ウィザード」も参照してください。
ダイアログ ボックス	アプリケーション インターフェイス内で表示され、ユーザーが情報を入力したりコマンドを指定したりすることができるウィンドウ。
ダイナミック リンク ライブラリ (DLL)	他のアプリケーションから呼び出される関数を含むコードベースの共有ファイル。
タスク マネージャー	コンピューターのパフォーマンス、およびコンピューター上で実行中のアプリケーション並びに処理についての情報を提供する Windows プラットフォームで利用可能なツール。場合によっては、インストールする前に、競合を避けるために [タスクマネージャー] を使って実行中のアプリケーションを終了する必要があります。
チェックサム	破損がないかをテストするためにデータに適用される計算値。チェックサムは、ファイル内のデータのバイトを系統立てて順に結合することにより得られます。転送やストレージ圧縮の後、チェックサムの計算が再び行われ、結果が以前の結果と比較されます。数値が一致しない場合は、保存または転送されたデータにエラーがある可能性があります。
遅延実行	インストール スクリプトの実行と同時にインストール アクションを実行する。
抽出	キャビネット ファイルなどの圧縮ファイルからファイルを除外、または圧縮解除すること。圧縮されたファイルは、1 つまたは複数の他のファイルを含む単一ファイルです。圧縮ファイルはサイズが縮小されているので、容量を節約できます。圧縮済みファイルを使用するためには、まず圧縮ファイルからそれを抽出する必要があります。ファイルを抽出するには、Extract.exe と呼ばれるコマンドライン ユーティリティを利用することができます。インストール中に破損または不足しているオペレーティング システム ファイルまたは InstallShield ファイルは、インストールを完了するために圧縮ファイルから抽出する必要がある場合もあります。
ツリー	木のような構造を持つ追加データ。
ツリーノード	データ構造は、「子」ノードを含まない場合と、1 つ上含む場合があります。ノードはツリーの「ルート」です。「子」を含まないツリーノードは、一般的に「リーフ」と呼ばれます。

テーブル 1・InstallShield Express 用語集

用語	定義
データベース	データベース管理システム (DBMS) の個別の部分からなるデータの集まり。
データベース ハンドル	データベース関数が呼び出されたときにデータベースを指定する構成要素。
データベース関数	データベースで作動する関数。
データベース管理システム (DBMS)	編成、保存、および多くのユーザー用のデータの読み出しをコントロールするアプリケーションのセット。
データベースのコミット	Windows Installer データベースについて蓄積された変更。変更はデータベースがコミットされるまで、つまり **MsiDatabaseCommit** が呼び出されるまで反映されません。
テーブル	データ行の集まり。
デジタル署名	アプリケーション内のコードがリリース後に改ざんまたは変更されていないことをエンドユーザーに対して保証するため、アプリケーションにデジタル署名を付けることができます。アプリケーションにデジタル署名を付けると、エンドユーザーが製品をダウンロードするときにデジタル証明書が表示されます。
展開	(1) 圧縮ファイルをオリジナルのサイズに復元すること。圧縮ファイルとは、たとえばキャビネット ファイルのような 1 つまたは複数の他のファイルを含む単一のファイル。圧縮ファイルはサイズが縮小されているので、容量を節約できます。 (2) 圧縮ファイルを復元するのに利用される DOS コマンド。インストール中に破損または不足しているオペレーティング システム ファイルまたは InstallShield ファイルは、インストールを完了するために圧縮ファイルから展開する必要がある場合もあります。
テンプレート	規定する、またはひな型として利用できるもの。
動画	動作を表現するために、1 秒間に特定数のフレームが交互に表示される複数のグラフィックイメージ。
同期実行	非同期実行の反対。非同期実行の場合、プロセス制御は全プロセスが完了するまでリリースされません。
統合開発環境 (IDE)	ソフトウェア プログラムのインターフェイス。「インタラクティブ開発環境」とも呼ばれます。
ドラッグ	アイテムを選択して別の場所へ移動させる動作。
トランスフォーム	トランスフォーム(.mst ファイル)は、2 つのインストール データベースの違いを表します。たとえば、ネットワーク管理者がある製品を会社の部署によって異なる構成で配布する場合があります。この場合、製品の各構成に対しトランスフォームを作成し、必要に応じて適切なトランスフォームを適用します。

テーブル 1・InstallShield Express 用語集

用語	定義
トランスフォーム エラー条件フラグ	トランスフォームのエラー条件を設定する際に使用するプロパティのセット。
トランスフォーム検証フラグ	トランスフォームが Windows Installer パッケージに適用可能であることを検証するのに利用されるプロパティのセット。
内部整合性評価プログラム (ICE)	エラーの可能性 (またはエラーそのもの) を警告するために Windows Installer データベース上で実行できるテスト (Orca、Msival2、InstallShield Premier または InstallShield を使用)。
内部ソース ファイル	インストール アーカイブ ファイルに含まれるファイル。
内部ユーザー インターフェイス	インストール中にエンドユーザーに表示される、グラフィック ユーザー インターフェイスの作成に利用できるインストーラーのビルトイン機能。
名前付きインスタンス	インスタンスとは、規定のサーバー上にある SQL Server の完成された独立インストールです。ひとつのデフォルトのインスタンスおよび名前付きインスタンスの任意の番号 (16 まで) を単一の サーバーにインストールすることができます。インスタンス間で共有される管理ツールおよびクライアント コネクティビティ コンポーネント以外、各インスタンスは原則的にスタンドアロンです。各インスタンスは独自のセキュリティを持ち、個別に開始、および停止させることができます。さらに同じサーバー上にあるその他のインスタンスに対して個別にサービスパックを当てることができます。
ナレッジ ベース (KB)	ソフトウェアのヘルプ情報の範囲外にあるテクニカル情報、手順、記事を集めたもの。ナレッジベースでは、ヒント、こつ、およびテクニックに加え、FAQ(よく聞かれる質問)、技術およびデザイン上の問題に関する記事を参照することができます。Revenera の Web サイト (http://www.installshield.com) では、ナレッジベースの内容が定期的に更新されています。
ネスト インストール	実行中のインストール (「親製品」と呼ぶ) から別のインストールパッケージ (「子製品」と呼ぶこともある) をインストールまたは削除するカスタム アクションの一種。
ネットワーク	ハードウェア、ソフトウェア、および情報を共有するため相互に接続された 2 台以上のコンピューター。インストール中、ソフトウェアを適切にインストールするため、ネットワーク上にあるインストールファイルをコンピューターのハード ドライブへ移動する必要がある場合があります。
ハード ドライブ	コンピューターの主要な記憶デバイス。ハードドライブには、磁気的にデータが読み書きされるディスクが含まれます。ハード という用語はハードドライブで利用されるアルミニウムやガラス製のディスクを、プラスチックでできたフロッピーディスクから区別するのに使われます。
ハイパーテキスト マークアップ言語 (HTML)	ハイパーリンクやテキスト形式のマークアップを含む Web ページを作成するための言語。

テーブル 1・InstallShield Express 用語集

用語	定義
配布メディア	ユーザーに配布される CD-ROM またはその他のメディア フォーマット。
パス	コンピューター オペレーティング システムで、パスは特定のファイルへのファイル システム内の経路。パス名はそのパスを詳述します。各オペレーティング システムは、パス名の指定に独自のフォーマットを持ちます。DOS、Windows、および OS/2 オペレーティング システムでは、driveletter:directorynamesubdirectorynamefilename.suffix フォーマットが使用されます。UNIX ベース システムは、/directory/subdirectory/filename フォーマットが使用されます。
パス変数	プロジェクトを移動したり、フォルダー構造を変更するたびに各ソース ファイルのパスを変更する必要がなくなるよう、中央ロケーションに 1 度だけ定義するだけで済むロケーションを示す変数。代わりにパス変数を使用して、l共通に使われるパスを1 度定義するだけで済みます。共通パスは、インストール プロジェクトの開発中に使用されます。これらのパスは、アプリケーションがインストールされるターゲットマシンには適用されません。代わりに、インストール プロジェクトに含める必要があるのソースファイルにリンクするために使用されます。プロジェクトをビルドする際、これらのリンクが評価され、ポイント先のファイルがインストール パッケージに組み込まれます。
パッケージ	インストールデータベースファイル、アプリケーションのファイル (インストールデータベースファイルとは別、またはインストールデータベースファイルに圧縮されたもの)、および 実行可能ファイル (上記すべてのファイルを圧縮して含むことが可能) を含む、インストールの実行に必要なすべてのファイル。
パッケージ コード	インストール パッケージの Globally Unique Identifier (GUID)。「globally unique identifier (GUID)」も参照してください。
パッチ	アプリケーションのファイルおよびインストールを特定バージョンにアップデートする、または以前のバージョンのバグを修正するのに必要なアプリケーションの一部のみを含む、特殊なタイプのインストール パッケージ。
パッチ ファイル	パッチに利用されるパッチパッケージ。パッチパッケージ (.msp) ファイルには、インストールされた 1 つ以上の製品バージョンをアップグレードするのに必要なトランスフォームおよび指示が含まれています。Windows Installer はパッチパッケージを利用してローカルインストールおよび管理インストールをパッチします。パッチパッケージは通常のインストールパッケージのようなデータベースを含みません。その代わり、ターゲットとするインストールパッケージのデータベースにパッチ情報を追加するデータベーストランスフォームが 1 つ以上含まれます。インストーラーはこの情報を利用して、パッチパッケージのキャビネット ファイル ストリームに格納されているパッチファイルを適用します。
パッチの適用	アプリケーション全体ではなく、変更された部分のみを置換するインストールのアップデート方法。これによって、エンドユーザーは製品全体ではなく規模の小さい製品のパッチをダウンロードすることができます。

テーブル 1・InstallShield Express 用語集

用語	定義
パネル	エンドユーザーが構成できる 1 つ以上の関連する設定を含むウィザード内にあるウィンドウ。「ウィザード」も参照してください。
パブリック プロパティ	エンドユーザーまたはシステム管理者によってその値が設定されるグローバル変数。パブリック プロパティは、インストール中、ユーザー インターフェイスとの対話、コマンドラインでのプロパティの設定、トランスフォームの適用、あるいは標準またはカスタム アクションの使用によって設定または変更することができます。プライベート プロパティの値とは異なり、パブリック プロパティの値は変更可能です。パブリック プロパティ名には、小文字は使えません。
パブリッシュ	アドバタイズ (「ジャストインタイム」インストール) の一種。インストール中にコンポーネントのユーザー インターフェイス要素は作成されませんが、コンポーネントは [コントロール パネル] の [プログラムの追加と削除] アプレットからインストールすることができます。また、インストールされたコンポーネントがインストーラーからパブリッシュされたコンポーネントを要求した際にもインストールできます。
バリューパック (VP)	インストールされたアプリケーションの機能をアップグレードするインストール。
左クリック	マウスの左ボタンを選択してクリックする動作。
ビットマップ	ビットの配列として表示されるイメージ。ビットマップ グラフィックでは、イメージは集合体としてパターンを形成するピクセルと呼ばれる小さな四角の集まりでスクリーン上に表示されます。イメージ内の各ピクセルは、1 つまたは複数のビットに対応します。
非同期実行	インストーラーがメインのインストールを実行中に独立してスレッドを実行し続けるカスタム アクションです。
非分割 CD/DVD メディア タイプ	ペイロードは CD/DVD アーカイブにレイアウトされますが、複数メディアへの分割はされません。
標準アクション	インストール開発ソフトウェア製品に組み込まれているアクション。InstallShield 製品は、カスタム アクションの作成および使用もサポートしています。「アクション」、「カスタム アクション」も参照してください。
ビルボード	インストール中に表示することが可能なマーケティング メッセージなどのイメージ。
ファイル	ファイルシステム内のデータ格納要素。
ファイル拡張子	ファイルに格納されているデータの種類を示す、最終ドットに続くファイル名の一部。
フォーマット済み	データが保存または表示のために分割、または配列された状態。

テーブル 1・InstallShield Express 用語集

用語	定義
フォールス ポジティブ	テストで true 評価を出すが実際はそうではないもの。多くの場合、false positive はウィルス対策ソフトウェアに存在します。たとえば、ウィルス対策ソフトウェアは InstallShield ファイルの中にウィルスを発見したと報告するが、実際はそうではない場合があります。ソフトウェアのウィルス定義をアップデートする必要があります。
復元性	必要に応じてコンポーネントを再インストールするアプリケーションの機能。コンポーネントが誤って削除されたり壊れたりした場合でも、Windows Installer の技術によりアプリケーションの自己修復が可能です。
プラットフォーム	インストール プログラムがターゲットにするオペレーティング システム。あるインストール テクノロジは、特定のプラットフォームへのインストールの作成に限られていることもあります。
プレビュー モード	ユーザー インターフェイスのデザイン、またはダイアログおよびビルボードの現在の概観を表示するモード。*プレビューモード*は Windows Installer で使われている用語です。
プロパティ	インストールまたはアンインストール中にインストーラーまたはアンインストーラーによって使用されるインストール プロジェクト内のオブジェクトに配置された値。
分散コンポーネント オブジェクト モデル (DCOM)	拡張型コンポーネント オブジェクト モデル (COM)。コンポーネントベースのアプリケーションの作成を可能にする Microsoft が開発したソフトウェア アーキテクチャ。
分離アプリケーション	元のアプリケーションが開発およびテストされたときに使用されたすべてのバージョンのコンポーネント (.dll ファイルなど) が常にロードされるように変更されるアプリケーション。
変換プロジェクト	(1) 別のタイプに変換されたファイル。 (2) InstallShield の MSI/MSM オープン ウィザードは、インストール プロジェクト (.msi ファイル) やマージ モジュール (.msm ファイル) を、InstallShield で変更とビルドが可能な InstallShield インストール プロジェクト (.ism ファイル) に変換するためのツールです。
編集フィールド オブジェクト	ユーザーによるフィールドへのテキスト入力を可能にする対話型オブジェクト。
変数	コンピューターに格納されている値。値は、印刷可能な文字、数値、およびテキストを使って作成することができます。値が変化しない定数とは違い、変数はいつでも値を変更することが可能です。
ホットフィックス、または、修正プログラム	アプリケーションの次のリリースまで配布を延期することができない、重要かつ規模の小さいバグ修正。ほとんどのホットフィックスは小さいパッチ形式で、ソフトウェア ベンダーの Web サイトからダウンロードすることができます。
ボリューム	1 つのメディア上にある、ブロック単位の空きスペースの総容量。

テーブル 1・InstallShield Express 用語集

用語	定義
マージ モジュール (MM)	ランタイム .dll ファイルや仮想マシンなど特定の機能のインストールに必要なすべてのロジックおよびファイルを含むパッケージ。マージ モジュールは一度ビルドすると、他のセットアップ プロジェクトに追加できます。
マイナー アップグレード	インストールに加えられる変更の為に製品コードを変更する必要がない製品のアップデート。
マクロ	(1) ひとまとまりの命令群からなる命令。
マネージド アプリケーション	アプリケーションのインストールにシステム権限が使われた場合、アプリケーションは「マネージド アプリケーション」と呼ばれます。
マルチメディア	サウンド、ビデオ、グラフィック、および動画など、情報を伝える為にまとめて利用される様々なメディアの集合。
メジャー アップグレード	ProductCode プロパティで変更が必要な製品の完全アップデート。同じマシンに 同じ製品の 2 つのバージョンをインストールする場合、新しい製品コードが必要になります。
メタファイル	標準 Windows メタファイル形式で保存されたイメージメタファイルは 1 セットのドローイング命令群です。Windows メタファイルは .wmf というファイル拡張子を持っています。
メディア ディスクリプタ (MD)	インストール可能な単位 (IU) の意味。通常ファイルとして存在するデータの物理的なソースに対して、デプロイメント ディスクリプタ (DD) で要素のマッピングを行います。
メンテナンス モード	既に製品がインストールされているシステム上で、2 回目 (およびそれ以降) にインストール プログラムを実行をすると、インストールはメンテナンス モードで行われます。このメンテナンスモードでユーザーは、最初のインストールで選択した機能の変更、インストール済みの機能の修復、およびプログラム全体の削除が行えます。
モーダル	アプリケーションが続行する前に、ダイアログを表示してユーザーからの返答を待ちます。その間、ユーザーは他のウィンドウまたはダイアログ ボックスを使用できません。「モードレス」も参照してください。
モードレス	ユーザーが他のウィンドウ、ダイアログ ボックスと対話できるようにします。「モーダル」も参照してください。
文字列テーブル	すべてのサポート言語に対する文字列 ID、値、コメントを保存するデータベース。これらの文字列は、実行時にエンドユーザーに表示されるダイアログおよびメッセージボックスで使われます。
ユーザー インターフェイス (UI)	ソフトウェアのユーザーへのアクセス。

テーブル 1・InstallShield Express 用語集

用語	定義
ユーザー プロファイル	ショートカット、お気に入り、アプリケーション、ディスプレイ、並びにハードウェアの設定など、各エンドユーザーの設定についての記録。ユーザープロファイルは、複数のユーザーがそれぞれの環境設定を保持しながら 1 台のコンピューターを共有することを可能にします。
ライブラリ	予備に保存されている類似オブジェクトのコレクション。たとえば、ソースコードまたはオブジェクトコード形式のプログラム、データファイル、スクリプトおよびテンプレート。プログラム ライブラリは、プログラマーがコードを書いているときに ″呼び出す″ ことができるプリコンパイル済み、再利用可能プログラム ルーチンのコレクションです。
リッチ テキスト フォーマット (RTF)	フォントやマージン等のフォーマット情報を表示する特殊コマンドを含む Microsoft のファイル フォーマット。RTF を使用すると、異なるワードプロセッサーおよびオペレーティング システム間でファイルを交換することができます。
リフレッシュ ビルド	このタイプのビルドはカスタムコードのみ再コンパイルします。また、全インストールを再ビルドすること無しに変更があった単一ファイルのみ再ビルドおよび追加/変更ができるので、単一ファイルの置換も可能です。このタイプのビルドは、InstallScript プロジェクトでのみ使用することができます。
リモート プロシージャ コール スタブ (RPC)	他のコンピューター上のサービスを要求するのに使用されるプログラムの中の小さいサイズのルーチン。RPC は、プロトコルまたは予め合意されたデータ送信のためのフォーマットで、これをプログラムで使用することにより、他のコンピューターにあるプログラムからサービスを要求することが可能になります。スタブはプログラムからのリクエスト受け入れて、それをリモートプロシージャへ転送します。プロシージャが完了すると、スタブは結果を受け取り、リクエストをしたプログラムへまた戻します。
リリース ノート	アプリケーションの配布と共に含まれるファイル。このファイルには、アプリケーションのインストールおよびアンインストールに関する重要な情報が含まれています。また、他の製品ドキュメントに含まれていない可能性がある情報も含まれることがあります。
リリース ビルド	リリースに適した完全再ビルド。
レジストリ	個人設定およびコンピューターにインストールされたソフトウェアおよびハードウェアを記録するために Windows オペレーティング システムによって使用される中央データベース。インストール時、インストールに関して選択された項目はレジストリに書き込まれます。
レジストリ ファイル	定義済み形式のテキストファイルで、レジストリにマージできるキーと値が含まれています。
列	縦に並んだ一連の情報。
ローカリゼーション	製品またはサービスを特定の言語および文化に翻訳適合するプロセス。

テーブル 1・InstallShield Express 用語集

用語	定義
ロード順グループ	サービスの中には、起動前に他のサービスが既に実行されていなければならないものがあります。そのため、サービスをグループ化して指定した順序でロードするよう設定する必要があります。サービスのロード順グループは、HKEY_LOCAL_MACHINE¥System¥CurrentControlSet¥Control¥ServiceGroupOrder にリストされています。 Set¥Control¥ServiceGroupOrder にリストされています。サービスの ″起動タイプ″ プロパティにより、グループ内でいつロードされるかが決定されます。
ロールバック	インストールは、インストール プロセス中に行われた変更をすべて記録しており、エラーが発生してインストールが中止された場合でも、これらの変更は「ロールバック」され、マシンは元の状態に復元されます。
ログ データベース (LogDB)	インストール/アンインストール中にログを記録するように構成されたすべてのログを含むターゲットマシン上にあるデータベース。
ログ ファイル	ファイルには、コンピューター ワークステーションまたは Web サーバー上でソフトウェア プロセス中(インストール、ビルド、ダウンロード等) に発生したアクティビティの記録が含まれています。たとえば、Web サーバーは、すべてのサーバーへのリクエストをリストし、コピーされたファイル、転送されたバイト数、リクエストされたページ、イメージおよびファイルを要約して、ログファイルのメンテナンスを行います。ビルド ログ ファイルは、ビルドに含められたすべての機能、セットアップ タイプ、マージ モジュール、ダイナミック リンク、およびファイルをリストします。
ログの記録	インストール、ビルドまたはダウンロード等のソフトウェア プロセス のアクティビティをログファイルに記録するプロセス。
割り当て	アドバタイズの種類。管理者がアプリケーションをマシンに割り当てると、次回マシンを起動または再起動したときにインストール プログラムが自動的に実行します。管理者がアプリケーションをユーザーに割り当てると、インストール プログラムはユーザーのマシン上にある [スタート] メニューにショートカットを配置します。ユーザーがショートカットを選択する、または割り当てられたアプリケーションに関連付けられた文書を起動したとき、アプリケーションがインストールされます。

索引

記号

_IsSetupTypeMin 194
_serial_verifyCA_isx custom action 707
_serial_verifyCA_isx_helper custom action 707
.cab ファイル
    最大サイズの構成 123, 125
    制限 123, 125
.exe ファイル 274
    カスタム アクション 274
.ini file
    ビュー 544
.ini ファイル 544
    キーワードの作成 260
    セクションの追加 259
    ファイルへのポイント 258
.isproj 332
.msi
    パッケージ ファイルの名前を変更 591
.msi ファイル
    複数を同時に実行 218
.NET
    再配布可能ファイル 231
.NET Framework
    バージョン 2.0、64 ビット 231
    バージョン 3.0 231, 232
    バージョン 3.0、64 ビット 231
.NET アセンブリ 409
.swf 315
.vdproj 150
    InstallShield プロジェクト (.ise ファイル) に変換 150

数字

27500 680
27501 680
27508 681
27509 681
27510 681
27517 681
27555 682
64 ビット
    64 ビット フォルダーのインストール 195
    System32 フォルダー ソース ファイル 198
    オペレーティング システム、ターゲットにする 83
    レジストリの 64 ビット領域にインストールする 246

A

ALLUSERS 388

C

CheckForProductUpdates カスタム アクション 707
CheckForProductUpdatesOnReboot カスタム アクション 707
COM サーバー、登録 237
COM サーバーの登録 237
COM 抽出
    管理者権限あり/管理者権限なし 86
    レジストリ変更を除外 237
COM+ アプリケーションとコンポーネント 565
    インストールに追加 285
COM+ アプリケーションとコンポーネントの管理 565
CommonFiles64Folder 83

D

DATABASEDIR 500
DirectX 9 オブジェクト 233
DirectX オブジェクト ウィザード 470

Disk1 フォルダー 282
DLL カスタム アクション 271
DLLWrapCleanup カスタム アクション 707
DLLWrapStartup カスタム アクション 707

E

Express 2.x プロジェクトのアップグレード
　トラブルシューティング 677
Express プロジェクト 98

F

Filters.xml 235
　COM 抽出のレジストリ変更の除外を指定 237
　依存関係スキャナー除外を指定 235
Flash ファイル 315
　ビルボードとして追加 315
FlexNet Code Aware 統合 32
FlexNet Connect 521
　ビュー 521

G

GUID 102

H

HTTP エラー、構成 297

I

IIS 550
　ASP.NET バージョン 295
　INSTALLSHIELD_SSI_PROP 288
　SSIEnableCmdDirective 288
　SSL 証明書 293
　TCP ポート番号 291
　Web サイトのアンインストール 295
　Web サイトの作成 289
　アプリケーションの作成 289
　アプリケーションのマッピング 296
　アプリケーションをアンインストール 295
　エラーメッセージの構成 297
　概要 286
　仮想ディレクトリのアンインストール 295
　仮想ディレクトリの作成 289
　機能の関連付け 295
　サイト番号 291
　サポート対象バージョン 286
　実行時の要件 287
　タイムアウトのパラメーター 297
　ネスト仮想ディレクトリ 291
　ビュー 550
　ホスト ヘッダー 293
IISROOTFOLDER サポート 300
の参照 544
 544
INSTALLDIR 383
InstallShield 2019 Express Edition の強化機能 26
InstallShield 2019 の新しい機能 27
InstallShield 前提条件 216, 218
　InstallShield 前提条件を使ってアプリケーションをアンインストール 227
　InstallShield 前提条件を使ってインストールを実行 224
　インストール順 221
　概要 218
　機能から削除 221
　機能との関連付け 220
　個別の場所を設定 223
　再配布可能ファイル ギャラリーに追加 214
　前提条件を含むリリースを構成 222
　必要実行レベルを指定 338
　ビルド時の場所を設定 222
　プロジェクトから削除 218
　プロジェクトに追加 216
　ユーザー アカウント制御のプロンプト 171
　リリースの場所 344
ダイアログ ボックス 448
 448
INSTALLSHIELD_SSI_PROP プロパティ 288
Internet Explorer 前提条件 218
IsCmdBld.exe 329, 715
ISCmdBld.exe のコマンドライン パラメーター 715
ISComponentServiceCosting カスタム アクション 707
ISComponentServiceFinalize カスタム アクション 707
ISComponentServiceInstall カスタム アクション 708
ISComponentServiceRollback カスタム アクション 708
ISComponentServiceUninstall カスタム アクション 708
ISIISCleanup カスタム アクション 708
ISIISCosting カスタム アクション 708
ISIISInstall カスタム アクション 708
ISIISRollback カスタム アクション 708
ISIISUninstall カスタム アクション 708
ISInstallPrerequisites 709
ISJITCompileActionAtInstall カスタム アクション 709
ISJITCompileActionAtUnInstall カスタム アクション 709
ISLockPermissionsCost カスタム アクション 709
ISLockPermissionsInstall カスタム アクション 709
ISNetApiInstall カスタム アクション 709
ISNetApiRollback カスタム アクション 709
ISNetCreateIniForOneUser カスタム アクション 709
ISNetDeleteIniFile カスタム アクション 710
ISNetGetGroups カスタム アクション 710
ISNetGetServers カスタム アクション 710
ISNetGetUsers カスタム アクション 710
ISNetSetLogonName カスタム アクション 710

ISNetValidateLogonName カスタム アクション 710
ISNetValidateNewUserInformation カスタム アクション 710
ISO/IEC 19770-2 184
ISPreventDowngrade 141
　追加または削除 175
ISPrint カスタム アクション 710
ISQuickPatchFinalize カスタム アクション 711
ISQuickPatchFixShortcut カスタム アクション 711
ISQuickPatchHelper カスタム アクション 711
ISQuickPatchInit カスタム アクション 711
ISQuickPatchInit9X カスタム アクション 711
ISQuickPatchInit9X2 カスタム アクション 711
ISRunSetupTypeAddLocalEvent カスタム アクション 711
ISSelfRegisterCosting カスタム アクション 712
ISSelfRegisterFiles カスタム アクション 712
ISSelfRegisterFinalize カスタム アクション 712
ISSetAllUsers カスタム アクション 712
ISSetTARGETDIR カスタム アクション 712
ISSetupFilesCleanup カスタム アクション 712
ISSetupFilesExtract カスタム アクション 712
ISUnSelfRegisterFiles カスタム アクション 712
Itanium
　および .NET Framework 231

J

Java(TM) 2 Runtime Environment 前提条件 218
Jet 4.0 前提条件 218
JRE 前提条件 218
JScript カスタム アクション 275

L

LaunchProgramFileFromSetupCompleteSuccess カスタム アクション 713
LaunchReadmeFileFromSetupCompleteSuccesscustom アクション 713

M

MDAC 2.8 前提条件 218
Microsoft .NET Framework 231
　バージョン 2.0、64 ビット 231
　バージョン 3.0 232
　バージョン 3.0、64 ビット 231
MSBuild 332
MSDE 2000 前提条件 218
DLL カスタム アクション
　設定 581
MSI DLL カスタム アクション 269
　設定 579
MSI ログ記録の作成 506
MsiPatchOldAssemblyFile 375
MsiPatchOldAssemblyName 375
MsiRMFilesInUse ダイアログ 308

O

ODBC リソース 542
　インストールに含める 260
　機能に関連付ける 261
　属性 262
ODBC リソースのインストール
　インストール 542

P

Palm OS デバイスのインストール
　要件 81
ProductCode 383
ProductName 383
ProductVersion 383
Program Files (x86) と Program Files の違い 83
ProgramFiles64Folder 83

Q

QuickPatch
　InstallShield で作成 368
　QuickPatch 作成の簡素化 368
　ウィザード 469
　簡素化の制限事項 368
　ビルトイン InstallShield カスタムアクション 368
　プロジェクトの種類 98
QuickPatch パッケージの簡素化 368
QuickPatch プロジェクト 99
　カスタム アクション 625
　手順 367

R

Readme 500

S

setAllUsersProfile2K カスタム アクション 713
SetARPINSTALLLOCATION カスタム アクション 713
Settings.xml 122
Setup.Exe
　Setup.exe コマンドライン パラメーター 718
Setup.exe 322, 338
　InstallShield 前提条件を含める 218, 223
　アプリケーションと共に Windows Installer をインストール 230
　インストールに作成 338
　プロパティのカスタマイズ 323

Seup.exe
名前を変更 591
Setup.exe の著作権情報 323
SETUPEXEDIR 383
setUserProfileNT カスタム アクション 713
ShowMsiLog カスタム アクション 713
SourceDir 383
SSL 証明書 293
subweb
エラーメッセージの構成 297
Sysnative フォルダー 198
System64Folder 83
SysWow64 83

T

TARGETDIR 383
Team Explorer 410
Team Foundation Server の統合 410
TFS の統合 410

U

UAC とインストール 171
Update.exe
プロパティのカスタマイズ 373
Update.exe の著作権 373

V

VBScript カスタム アクション 275
Visual Basic プロジェクト タイプ 98
Visual Studio 405
Visual Studio .NET
InstallShield ツールバーまたはコマンドを追加 408
InstallShield との統合 405
InstallShield プロジェクトを開く 406
Web サービスまたはアプリケーションからプロジェクト出力を追加 410
Visual Studio Team Foundation Server
InstallShield と統合 410
voicewarmupx 449
VSSolutionFolder 407

W

Web サービス、配布 298
Web サイト 289
エラーメッセージの構成 297
作成 289
ファイルを追加 294
フォーム認証 299
Web サイト、アプリケーション、または仮想ディレクトリの 404 エラー メッセージ 297
Web サイト、アプリケーション、または仮想ディレクトリのカスタム エラー メッセージ 297
Web 配置 341
Windows Installer エンジン要件 81
Windows Installer 再配布可能ファイル
バージョン 4.5 231
Windows Mobile デバイスのインストール
要件 81
Windows サービス 263
インストールと開始 263
Windows ロゴ 175
Windows ロゴ プログラム 175
WMI (Windows Managment Instrumentation) 前提条件 218
WMI 前提条件 218
Wow6342Node 83

あ

アイコン
ショートカット 242
ダイアログ ボックス 465
465
アクション、カスタム 578
.exe の起動 274
プロパティの設定 274
DLL 関数の呼び出し 273
プロパティを設定 269, 271
JScript 275
MSI DLL 関数の呼び出し 270
VBScript 275
インストールまたはアンインストールでスケジュール 278
アセンブリ 375
グローバルアセンブリキャッシュのパッチ 375
圧縮インストール、作成 338
アップグレード 127, 517
Express 2.x プロジェクト 143
InstallShield の新しいバージョンへ 127
ターゲットマシンの製品 517
ダウングレードの防止 175
パッチとの違い 356
アプリケーション ライフサイクル 96
アプリケーションのライフサイクル 96
アンインストール 227
前提条件を持つアプリケーション 227
パッチ 366

い

依存関係 530
スキャン結果の確認 235
スタティック スキャン 234
ダイナミックスキャン 234

依存関係スキャナー
ファイルのフィルター 235
イメージ、カスタム、ダイアログ ボックス 435
インストール 94
定義 94
インストール プロジェクト 97, 98
作業を行う 98
作成 100
名前またはロケーションの変更 101
開く 100
古いバージョンから新しいバージョンへアップグレード 127
保存 101
標準的な要素 95
インストール先フォルダー 500
一般情報プロパティ 500
ハードコード化されたインストール先ディレクトリを指定する 199
レジストリキー値をディレクトリ識別子として使用する 180
インストールを実行
InstallShield 前提条件を使う 224
複数の .msi ファイルを同時に実行 218
インストールを連鎖させる 218
インターネット インフォメーション サービス 550
ASP .NET バージョン 295
INSTALLSHIELD_SSI_PROP 288
SSIEnableCmdDirective 288
SSL 証明書 293
TCP ポート番号 291
Web サイトのアンインストール 295
Web サイトの作成 289
アプリケーションのアンインストール 295
アプリケーションの作成 289
アプリケーションのマッピング 296
エラー メッセージの構成 297
概要 286
仮想ディレクトリのアンインストール 295
仮想ディレクトリの作成 289
機能の関連付け 295
サイト番号 291
サポート対象バージョン 286
実行時の要件 287
タイムアウトのパラメーター 297
ネスト仮想ディレクトリ 291
ビュー 550
ホスト ヘッダー 293
インターネット配布 339
プロキシ サーバー サポート 339

え

エラー 631
Setup.exe 実行時 682
ビルド 631
エラー、Web サイト、アプリケーション、または仮想ディレクトリの構成 297
エンド ユーザー ダイアログ
テーマ 301
エンドユーザー ダイアログを参照 571
エンドユーザーにアップデートの通知をする
FlexNet Connect 521

お

オートプレイ 347
オブジェクト 216
ギャラリーを管理 211
使用中のコンピューターへのダウンロード 213
プロジェクトから削除 218
プロジェクトに追加 216

か

会社名
Setup.exe のプロパティ ダイアログ ボックスに表示 323
Update.exe のプロパティ ダイアログ ボックスに表示 373
カスタム アクション 578
_serial_verifyCA_isx 707
_serial_verifyCA_isx_helper 707
.exe の起動 274
プロパティの設定 274
CheckForProductUpdates 707
CheckForProductUpdatesOnReboot 707
DLL 関数の呼び出し
プロパティを設定 269, 271
DLLWrapCleanup 707
DLLWrapStartup 707
InstallShield、説明 707
ISComponentServiceCosting 707
ISComponentServiceFinalize 707
ISComponentServiceInstall 708
ISComponentServiceRollback 708
ISComponentServiceUninstall 708
ISJITCompileActionAtInstall 709
ISJITCompileActionAtUnInstall 709
ISNetApiInstall 709
ISNetApiRollback 709
ISNetCreateIniForOneUser 709
ISNetDeleteIniFile 710
ISNetGetGroups 710
ISNetGetServers 710
ISNetGetUsers 710
ISNetSetLogonName 710
ISNetValidateLogonName 710
ISNetValidateNewUserInformation 710
ISPrint 710
ISQuickPatchFinalize 711

ISQuickPatchFixShortcut 711
ISQuickPatchHelper 711
ISQuickPatchInit 711
ISQuickPatchInit9X 711
ISQuickPatchInit9X2 711
ISRunSetupTypeAddLocalEvent 711
ISSelfRegisterCosting 712
ISSelfRegisterFiles 712
ISSelfRegisterFinalize 712
ISSetAllUsers 712
ISSetTARGETDIR 712
ISSetupFilesCleanup 712
ISSetupFilesExtract 712
ISUnSelfRegisterFiles 712
JScript 275
LaunchProgramFileFromSetupCompleteSuccess 713
LaunchReadmeFileFromSetupCompleteSuccess 713
MSI DLL 関数の呼び出し 273
QuickPatch に含める 625
setAllUsersProfile2K 713
SetARPINSTALLLOCATION 713
setUserProfileNT 713
ShowMsiLog 713
VBScript 275
インストールまたはアンインストールでスケジュール 278
738
カスタム セットアップ ダイアログ 738
オプション 308
ダイアログ ボックス
サンプル 738
仮想ディレクトリ 289
エラーメッセージの構成 297
作成 289
ファイルを追加 294
環境変数 547
設定 263
ビュー 547
プロパティ 548

き

起動
管理者権限あり/管理者権限なしで InstallShield を起動 86
起動する
InstallShield を 32 ビットと 64 ビット システムで起動する違い 87
機能 514
定義 94
必須とする 191
表示設定 190
リモート インストール 192

く

クイック ビルド 328
グローバリゼーション 154
概要 154
サポート言語 155
グローバル アセンブリ キャッシュ 375
アセンブリのパッチ 375

け

言語 ID 155
言語識別子 155

こ

コマンドライン 329
MsiExec.exe パラメーター 718
Setup.exe パラメーター 718
コマンドラインからインストールをビルド 329
コンテキスト ヘルプ 89
178
保存されている情報 178
コンポーネント
定義 94

さ

サービス 263
インストールと開始 263
再起動 280
再配布可能ファイル
.NET Framework および言語パック 231
InstallShield 209
InstallShield 前提条件 218
ギャラリーを管理 211
参照する 214
プロジェクトに追加 216
再配布可能ファイルをコンピューターへダウンロード 213
サイレント インストール 348
削除
InstallShield 前提条件を再配布可能ファイル ギャラリーから削除 214
再配布可能ファイル ギャラリーから削除 214
プロジェクトから InstallShield 前提条件を削除 218
プロジェクトからオブジェクトを削除 218
プロジェクトからマージ モジュールを削除 218
作成
Microsoft Visual Studio で InstallShield プロジェクト 406
QuickPatch プロジェクト 367
概要 367
既存の QuickPatch 367
圧縮、自己展開型インストール ファイル 338

セットアップ プロジェクト 100
サブ Web 289
作成 289
ファイルを追加 294
サポート ファイル 589

し

シーケンス
インストール前提条件 221
スモール アップデート 364
シールド アイコン 171
自己展開型インストール ファイル、作成 338
システム検索
ファイルの詳細 437
481
実行可能ファイル
配布する単一の実行可能ファイルを作成
InstallShield 前提条件を含める条件 218, 223
実行時にセットアップの種類にアクセスする 194
自動アップデート通知
FlexNet Connect 521
ウィンドウ 422
固定/取り外す 121
ショートカット
アイコン 242
ショートカット フォルダーとプログラム フォルダー 532
アンインストール 245
作成 242
アプリケーション ショートカット 242
ビュー 532
プロパティの設定 533
ホット キー 244
昇格された権限 171
シリアル番号の検証 279
ダイアログ ボックス 453
進行状況バー 312
ビルボードを使って表示、または使わずに表示 312

す

スプラッシュ画面 725
スマート デバイス インストール
プロジェクトの種類 98
スモール アップデート 364
パッチ シーケンス 364

せ

製品
機能とコンポーネントの階層構造 94
製品名
Setup.exe のプロパティ ダイアログ ボックスに表示 323
Update.exe のプロパティ ダイアログ ボックスに表示 373
セットアップの種類 517
実行時にアクセス 194
ビュー 517
編集 193
前提条件 216, 218
インストール順 221
概要 218
機能から削除 221
機能との関連付け 220
個別の前提条件の場所を設定 223
再配布可能ファイル ギャラリーに追加 214
前提条件を使ってアプリケーションをアンインストール 227
前提条件を使ってインストールを実行 224
前提条件を含むリリースを構成 222
必要実行レベルを指定 338
ビルド時の場所を設定 222
プロジェクトから削除 218
プロジェクトに追加 216
ユーザー アカウント制御のプロンプト 171
リリースの場所 344
再配布可能ファイル ギャラリーから削除 214

そ

ソフトウェア識別タグ 184

た

ターゲット システムの要件 81
ターゲット システム要件 81
ターゲット マシンの再起動
Windows Vista 以降における UAC プロンプトへの影響 171
ダイアログ 571
ダイアログ テーマ 301
ダイアログ ボックスのカスタム イメージ 435
ダイナミック ファイル リンク 205
制限事項 203
ディレクトリごとのメソッド 204
ベスト プラクティス 204
タイル構成
設定 539
タイルの構成
スタート画面上のデスクトップ アプリのタイルを構成する 245
ダウングレード、防止 175
タグ 184
製品を識別 184

つ

ツールバー
InstallShield ツールバーを Visual Studio ツールバーに追加する 408
追加 409
.NET アセンブリをプロジェクトに追加する 409
InstallShield 前提条件、マージ モジュール、オブジェクトをプロジェクトに追加 216
再配布可能ファイル ギャラリーに InstallShield 前提条件を追加 214
ファイルを IIS 仮想ディレクトリに追加 294

て

データソース名 (DSN) の参照 542
テーマ、ダイアログ 301
テキストとメッセージ 577
InstallShield で編集 310
デジタル署名
QuickPatch パッケージ 371
アプリケーション 341
タイムスタンプ サーバー 122
ファイル 341
デジタル署名のタイムスタンプ サーバー 122

は

パスワード保護
QuickPatch 372
パッケージ コード 591
パッチ 364
アップグレードとの違い 356
アンインストール 366
ウィザード 469
グローバル アセンブリ キャッシュのアセンブリ 375
スモール アップデートのシーケンス 364
非管理者によるパッチ 366
プロジェクトの種類 98
353
ビュー 353

ひ

非管理者によるパッチ 366
要件 366
"必要実行レベル" 設定 338
ビュー 499
アプリケーション データの指定 521
セットアップ外観のカスタマイズ 571
セットアップの編成 499
セットアップ要件およびアクションの定義 577
ターゲット システムの構成 531
ビューリリースの準備 590
リスト、InstallShield 内 499
開く
インストール プロジェクト 100
ビルボード 312
Flash とイメージ ファイルの設定 574
Flash ファイルの追加 315
イメージを追加 316
削除 318
サポート対象ファイル タイプ 312
実行時の動作 317
種類 312
順序の設定 317
使用する種類を指定する 315
スクリーン ショットのサンプル 312
設定の構成 316
追加および構成を行うビュー 572
目的 312
リリースをビルドおよび実行せずにプレビューする 316

ふ

ファイル 522
アプリケーション ファイルを含める 195
依存関係のスキャン 530
機能との関連付け 527
ターゲットマシンで上書き 206
定義済みインストール先フォルダー 524
動的にリンク 205
プロジェクトで検索 207
プロパティ 438
ファイル拡張子の関連付け
作成 257
詳細 546
ファイルとディレクトリのアクセス許可のロック 460
ファイル拡張子の関連付け 546
フォーム認証 299
フォルダー 207
プロジェクトで検索 207
445
ダイアログ ボックス 445
プロキシ サーバーの設定 339
コントロール パネルの 178
プログラム互換性アシスタント 62
プロジェクト アシスタント 103
プロジェクトでファイルとフォルダーを検索 207
プロジェクトの種類
Visual Studio プロジェクトを InstallShield プロジェクトに変換 150
98
プロジェクトを参照 97
プロパティ 383
Windows Installer のプロパティ 383
使用 382

へ

ベスト プラクティス
ダイナミック ファイル リンク 204
ヘルプ
コンテキスト ヘルプ 89
使用 89
変換
Visual Studio プロジェクトを InstallShield プロジェクトに変換 150

ほ

保存
新しい名前または場所のプロジェクト 101

ま

マージ モジュール 216
ギャラリーを管理 215
使用中のコンピューターへのダウンロード 213
ビルド時の場所を設定 228
プロジェクトから削除 218
プロジェクトに追加 216
マージ モジュールを参照 214
マイナー アップグレード 352
マップされたドライブ
プロジェクトで参照 86

め

メジャー アップグレード 351

ゆ

ユーザー アカウント制御 (UAC) とインストール 171
ユーザーごとのインストールとマシンごとのインストールの違い 264
HKEY_CURRENT_USER 256

よ

要件 577

り

リモートインストール 192
リリースのビルド 321
Windows Installer をインストール 230
クイック ビルドを実行 329
コマンドラインを使う 329
前提条件の場所を指定 344
単一の自己展開型インストール ファイル 338
ビルド エラーのトラブルシューティング 631
ビルドの開始 321
マルチディスク インストールにボリューム ラベルを設定する 346
ランタイム エラーのトラブルシューティング 682
リリースの準備 590

れ

レジストリ エントリ 248
.reg ファイルのインポート 476
REG_MULTI_SZ 253
値の作成 251
インストール/アンインストールの動作 255
キーの作成 248
ドラッグ アンド ドロップ 249
レジストリ キーの "キー全体のアンインストール" オプション 255
レジストリ キーのアンインストール動作を設定する 256
レジストリ キーのインストール/アンインストール動作 255
レジストリ キーの "存在しない場合はインストール、存在する場合アンインストール" オプション 255
レジストリキーの "インストールのみ" オプション 255
レジストリのアクセス許可のロック 463

ろ

ロックダウン環境
パッチ 366