

# InstallShield 10.5

2004 年 11 月 4 日

## はじめに

InstallShield 10.5 は、すべてのプラットフォーム、オペレーティングシステムおよびデバイスのために開発された最も包括的なオールインワン インストール オーサリング ソリューションです。InstallShield 10.5 の 1 つしかない極めて使いやすいインターフェイスを利用して、開発者の方々は簡単に Windows Installer (MSI)、InstallScript およびクロス プラットフォームのためのインストールを作成し、さらにそれをデータベース サーバー、Web サービスおよびモバイルデバイスにまで拡張することができます。ユーザーは、状況に適したインストール テクノロジーを自由に選択することができます。InstallShield 10.5 は InstallShield DevStudio、InstallShield X、および、InstallShield MultiPlatform からの直接 アップグレードで、どのユーザーグループにも明確で使い慣れたプロジェクトタイプを提供します。

Windows インストール開発者の方々のために、InstallShield 10.5 では、その新しい特徴として、製品の試用版を素早く市場に届けることができる新しい Trialware 機能が追加されました。セットアップ作成者は、カスタムアクションを作成する手間をかけずに、ターゲットマシンで XML ファイルを簡単に編集するインストールを作成することができます。

Windows IDE は、セットアップ ダイアログおよび他のコンポーネントをプロジェクト間で共有するためのローカルおよびネットワーク リポジトリをサポートしています。Universal 開発者の方々は、アセンブリを使用してモジュラー式アプリケーションを共同で作成し、ソフトウェアの最重要製品データ (VPD) を顧客のシステム上で管理し、自社アップデート配布のためのソフトウェア製品を構築することができます。

また、InstallShield 10.5 では、様々な最先端テクノロジーのサポートもその大きな特徴になっています。InstallShield 10.5 では Visual Studio .NET IDE 内部で実行することが可能だけでなく、.NET アプリケーションも完全サポートしています。.NET Compact Framework をターゲットにしたアプリケーションを含む、様々な モバイルデバイス アプリケーション用インストールの作成が可能であり、Web サービス、SQL Server および 64-ビット アプリケーションもサポートしています。

[トップに戻る](#)

## システム要件

### Windows の場合:

#### プロセッサ

Pentium III クラスの PC (500 Mhz 以上を推奨)

#### RAM



256 MB の RAM (512 MB 推奨)

ハードディスク

500 MB ディスク空き容量 (再配布可能ファイルコードを含む)

ディスプレイ

1024 x 768 (XGA) 以上の解像度

オペレーティング システム

Windows 2000、Windows XP、または、Windows Server 2003

ブラウザ

Microsoft Internet Explorer 5.01 (IE 5.5 以降を推奨)

権限

システム上での管理者権限

マウス

Microsoft IntelliMouse またはその他の互換性のあるポインティング デバイス

**Linux、UNIX、Mac、OS/400 の場合:**

それぞれの対応プラットフォーム/OS 上でサポートされている特定の JVM については、Universal ヘルプ ライブラリの “開発およびエンドユーザー環境要件” をご覧ください。

プロセッサ

400 MHz のプロセッサ

RAM

128 MB の RAM

ハードディスク

150 MB のハードディスク空き容量

サポートされているプラットフォーム

AIX (Premier Edition)、Generic UNIX、HP-UX、Linux、Mac OS X、OS/400 (Premier Edition)、および Solaris。Universal IDE に必要な JRE/JDK バージョン 1.4.2 以降と共に

サポートされています。Universal Installer ランタイム用には JRE バージョン 1.2.2 以降がサポートされています。Java 1.5 のサポートは、インターフェイスおよびランタイムの両方に追加されました。

[トップに戻る](#)

## 参考情報

InstallShield での開発時におけるオペレーティングシステム要件が変更されました。Windows 95、Windows 98、Windows ME、および Windows NT4 用に作成されたインストーラーはまだサポートされていますが、開発には、Windows 2000、Windows XP、または Windows Server 2003 オペレーティングシステムが搭載されたコンピュータを使用してください。

Windows Installer 3.0 には、Windows 2000 SP3 以降の最低要件があります。この新しいバージョンを、セットアップの要件をそのみに制約することなく活用するために、新しい“バージョン 3.0 または バージョン 2.0 (どちらかシステムに最適なもの)” オプションが現在提供されている MSI エンジンに追加されました。今回のリリースでは、このオプションは両方の MSI エンジンを含みます。ターゲット オペレーティングシステムが Windows 2000 SP3 のとき、MSI 3.0 がインストールされます。MSI 3.0 のシステム要件が満たされない場合、MSI 2.0 がインストールされます。

以前のバージョンの InstallShield MultiPlatform および InstallShield X Universal でビルドされたプロジェクトは、InstallShield 10.5 Universal でアップデートまたはパッチすることはできません。オリジナルプロジェクトは InstallShield 10.5 Universal インターフェイスで開いてから保存し、それから InstallShield 10.5 Universal プロジェクトとして保存してください。保存された Universal 10.5 プロジェクトは、新しいインストールとして配布することができます。

今回のリリースでコラボレーション アーキテクチャを実現するために、全面的なアーカイブ Universal 構造のリファクタリングが行われました。以下の JAR ファイルは、“¥distributables¥platform¥ext” ディレクトリに移動されました。

- aixppk.jar
- genericunixppk.jar
- hpuxppk.jar
- linuxppk.jar
- macosxppk.jar
- solarisppk.jar
- windowsppk.jar

Universal ダイナミックスイートは、InstallShield 10.5 Universal で新しいアセンブリ機能が活用できるように大幅に強化されました。

- スイートで使用する Uninstaller は 今回より 1 つになりました。この Uninstaller は、スイート内で選択した製品をアンインストールするのに利用されます。スイート

アンインストールがスイート インストールと平行して配置されていることにより、シムトリーなエンドユーザーエクスペリエンスが提供されます。

- 製品の参照は、今回よりダイナミックスイートでは使われていません。スイート製品のビルドが完了すると、インストールが必要なアセンブリが、ランタイムの直前にスイートの "disk1/assemblies" ディレクトリに追加されます。これにより、誰もがスイートと共に配布する製品を選択できるようになるので、同じスイートプロジェクトを繰り返し利用して、無数の組み合わせの製品をインストールすることが可能になります。
- 製品とアセンブリの親スイートにダイアログを挿入する機能が大幅に強化されました。製品はこの新しいアセンブリ ダイアログ シークエンスを利用して、スイートにダイアログを投入します。

次の ウィザード リファレンス ビーンは、この新しいメカニズムで使用できません。

- ウィザード ビーン リファレンス
- マルチウィザード ビーン リファレンス
- スイート ウィザード ビーン リファレンス

ダイナミックスイートのデザイン変更に伴い、以前のバージョンの MultiPlatform または InstallShield X Universal でビルドされたダイナミック スイート プロジェクトもまた InstallShield 10.5 Universal ダイナミックスイートとして配布するために、下に説明に従ってデザインを変更する必要があります。

1. 参照された各製品のプロジェクトをアセンブリとしてリビルドします。詳細は、Universal ヘルplibラリの "アセンブリでの作業" をご覧ください。
2. ダイナミック スイートをビルドします。
3. スイートと共に配布する各ビルド済みアセンブリのディレクトリ構造全体を、スイートの "disk1/assemblies" ディレクトリに追加します。

すべての Universal 製品ビーン(製品、機能およびコンポーネント)の "必須" プロパティにある外部依存ファイルは使用不可になりました。この関数は、参照アセンブリによって置き換われました。詳細は、Universal ヘルplibラリの "既存のアセンブリをプロジェクトに追加する" を参照してください。

[トップに戻る](#)

## 新機能

### Windows ベースの開発者にとって新しい機能

#### MSI 3.0 サポート

InstallShield 10.5 は、新しい Microsoft Windows Installer (MSI) 3.0 と共にビルドされたすべてのプロジェクトをサポートしています。MSI 3.0 によりパッチの開発が単純化され、アンインストール可能なパッチの構成やパッチシークエンス機能の使用が可能になったほか、パッチパッケージにパッチメタデータを設定できるようになりました。

## [XML ファイルの変更] ビュー

ターゲットマシンで XML ファイルを簡単に編集することができるインストールを作成することによって開発時間を大幅に短縮することができます。カスタムアクションをまったく作成することなく、製品の設定や標準構成ファイルを保存する XML ファイルを変更することができます。

## MySQL サポート

InstallShield 10.5 は、[SQL スクリプト] ビューを拡張して、MySQL データベース サーバーをサポートします。MySQL データベース サーバーのすべての面を構成して (MySQL スクリプトをインストールの一部として実行します)、MySQL データベース サーバーに関連付けられているバージョン要件を追加、削除および編集することができます。

## ユーザー & グループ

インストール中に Windows ユーザーの追加および管理を行うことができます。インストール時に、既存の Windows ユーザーアカウントを選択または新規のアカウントを作成できます。

## Compact エンジン

シンプルかつ、即配布が可能なインストールを作成できます。少ない経費 (商用インストーラとしては業界最少!) でダウンロード時の負担も最小です。

## 常に上書きする

インストール時にバージョンや更新日に関わらず常に上書きされるようファイルをマークすることにより、セットアップ作成のスピードアップを図ることができます。これは、MSI プロジェクト用の InstallShield 10.5 の新機能です。以前は、InstallScript™ には、以前にも提供されていました。

## 強化されたヘルプ ライブラリ

Windows ヘルプ ライブラリは、参照ベースのシステムからタスクベースのシステムに再構成され、問題解決のための最新ソリューションを即座に見つけることができます。また、アプリケーション アップデートおよびパッチについての情報も拡張されました。

## ローカル リポジトリ

ローカルマシンに共通の要素のリポジトリをビルドして多数の新しいセットアップを素早く作成したり、プロジェクト全体を通して整合性を図ることができます。共通要素は、複数のプロジェクトで共有、再利用することができ、同じ作業を何度も繰り返す無駄を省くことができます。ネットワーク リポジトリは、Premier Edition でも利用することができます。

## Trialware 機能 (Premier Edition 限定機能)

InstallShield 10.5 の Premier Edition では、製品のソースコードを触ることなく、ごく簡単な構成変更で製品の完全試用バージョンの製品を作成することができます。Trialware の機能を利用すると、期限が切れるまで所定の日数だけ稼動するように設定することで、評価版を迅速に市場に送り出し、顧客および見込み客に製品を“試用”してもらうことができます。

## クロスプラットフォーム開発者にとっての新機能

### コラボレーション設計とアセンブリ

コラボレーションによって、規模の大きいプロジェクトをより小さい作業単位に分割することができます。“アセンブリ”として知られるこれらの作業単位は、再利用可能なモジュラー式コンポーネントで、プラグインするだけで複数のインストールに組み入れることができます。これらの 2 つの機能によって、アプリケーションのビルド、配布およびメンテナンスの仕方が大幅に変わります。

### 共通ファイル

すべてのプラットフォームで共通ファイルを自動的に管理できます。同じファイルが複数のコンポーネントまたはアセンブリ間で共有されているとき、そのファイルを同じインストール場所に複数回に渡ってインストールするように指定することができます。

### 新しいコマンドライン オプション

コマンドラインで、完全なランタイム環境用にアセンブリをマニュアルで提供することができます。新しい検索パス コマンドライン オプションは、追加のパスを指定して、アセンブリ データ ファイルを検索します。検索されたアセンブリ データ ファイルは、参照されたアセンブリがランタイムで見つかるようにクラスパスへ追加されます。

### システム変数

必要のないデータの再エントリを避けるためインストール間でシステム変数を共有することで、エンドユーザーのランタイム エクスペリエンスを改善することができます。

### シークレット変数

変数が“secret”に設定されると、その値は、VPD に書き込まれるとき暗号化されます。読み込まれたときは、複合化され、変数がランタイムに解決されるときプレーンテキストバージョンの値が取得されたものとして表示されます。

### UNIX シンボリック リンク

UNIX シンボリック リンクは、インストールの一部として簡単に指定および作成することができます。UNIX シンボリック リンクの製品アクションは、ターゲット UNIX システムで、プロジェクトに含められたファイルへのシンボリックリンクを作成します。

### ランタイム ヘルプ (Premier Edition 限定機能)

インストールランタイムへ “ヘルプ” ボタンを挿入できます。Universal プロジェクト ダイアログエディタのコントロールとして実装されるこのボタンは、ランタイムにクリックされたとき、所定の HTML ファイルをユーザーに表示します。ヘルプは、[ヘルプ] ボタンが配置されているパネルに合わせて表示されるように、コンテキスト センシティブにすることができます。

## 強化された機能コントロール

InstallShield 10.5 は、チェックボックスの表示によって示される部分的に選択された機能をサポートします。

## 強化されたロギング

InstallShield 10.5 Universal プロジェクトは、今回より、インストールアクションのログをより細かく記録し、その情報を Audit Trail (監査証跡) に保存します。

## ロケールとプラットフォームのプロパティ

ランタイムに素早くフィルタリングできるように、ロケールとプラットフォームのプロパティをコンポーネントにアサインすることができます。

## 新しい製品アクション

InstallShield 10.5 は、開発者が Windows と UNIX のターゲットマシンおよびドメインでグループとユーザーを作成または削除でき、Windows マシンでその特権を管理することができる Universal プロジェクトのための新しい 4 つの製品アクションを追加しました。

## 質問ダイアログ

エンドユーザーのフィードバックに基づいて必要な追加ランタイム関数を判断できます。開発者は、ユーザーに特定情報を入力するようプロンプトするいくつかの質問要素を設定することができます。

## 強化された実行プロセスアクション

強化された実行プロセスアクションは、プロセスの終了コードおよびアウトプットを変数に保存する能力を含みます。異なる所定のユーザー認証情報を使用してプロセスを実行できる能力も、同時に追加されました。

## 強化された最重要製品データ (VPD) レジストリ

最重要製品データ (VPD) レジストリは強化されて、より高い安全性のために今回よりデータベースに格納されています。今回より複数の VPD を持つことが可能になったターゲットマシンで、ユーザーはユニークで永続的な VPD の場所を簡単に指定できるようになりました。

## 強化されたダイナミック スイート

ダイナミックスイート プロジェクトの全面的なデザイン変更により、スイートインストールがより効率的に行えるようになりました。スイートには、今回よりランタイム バインディングが活用され、同じスイートを利用して無制限の組み合わせでアセンブリを配布することができます。アセンブリは、ターゲットマシンに即座にインストールされ、そのマシン上で複数の製品インストールによって利用されることが可能になります。

[トップに戻る](#)

## バグ修正

### 全てのプロジェクトタイプ

#### 1-VBCOE

Microsoft MSDE 2000 マージモジュール、および InstallShield MSDE 2000 マージモジュール オブジェクトが、[再配布可能ファイル (オブジェクト)] ビューから削除されました。Microsoft は、インストールされた MSDE 2000 インスタンスは Desktop Engine SP3a ファイルを使用してアップグレードすることができないため、これらのマージモジュールを使用することを推奨していません。Desktop Engine (MSDE 2000) SP3a では、マージモジュールを使用する既存のアプリケーションをサポートするマージモジュールが提供されています。

#### 1-L85YI/1-BM2DY

InstallShield オブジェクトインストーラは、今回より、オブジェクトウィザードで “追加のデザインタイムファイルを含める” チェックボックスが選択されて、配布されるオブジェクトがウィザードを含まないときも、適切にビルドされるようになりました。以前は、ビルドエラーが発生し、メディアは正常にビルドされませんでした。

#### 1-L7URO

InstallShield オブジェクトインストーラウィザードのオブジェクトの選択ボックスは、今回より、現在のバージョンで作成されたオブジェクトのみを表示します。以前は、その時のオブジェクトが、作成されたときのバージョンに関係なく、すべて表示され、オブジェクトインストーラでサポートされていないオブジェクトが選択されてしまうという結果を招く可能性がありました。

#### 1-ELB08

“バッチビルド” オプションは、今回より、VSIP モードで無効化されました。これまで、このオプションは提供されていましたが、リリースビューのコンテキストメニューで使用できませんでした。

#### 1-QTQIH

ダイレクトエディタのスクロールバーを右クリックしたときに未設定例外が発生していました。この問題は修正されました。

## 1-6X92S

より詳しい記述がファイル競合ビルドエラーに追加されました。

## 1-IAH8X

作成中のセットアップが読み込み専用ファイルを含むときに、[シャロービルド] リリースオプションがビルドを数回試みたときに失敗していた問題は、解決されました。

## 1-RS6Y1

SQL スクリプトでアポストロフィを含むコメントがあるときに、SQL スクリプトエラーが発生していた問題は解決されました。

## 1-RPB0D

[SQL ログイン] ダイアログで参照ボタンをクリックして SQL サーバーが見つからなかったときも、セットアップは終了しなくなりました。

## 1-RVYZ9

データベース インポート ウィザードで生成された SQL スクリプトを実行したときに、timestamp datatype がテーブルにあるときも、SQL スクリプトエラーは発生しなくなりました。

## 1-RWLCT

データベース インポート ウィザードで生成された SQL スクリプトを実行したときに、ユーザー定義データタイプの列がテーブルにあるときも、SQL スクリプトエラーは発生しなくなりました。

## 1-S0XOL

SQL スクリプトで、スクリプトの終わりに空白の行が無いとき SQL スクリプトエラーが発生していました。この問題は修正されました。

## 1-T7YAL

データベース インポート ウィザードの“スクリプトするオブジェクト” パネルで“レコード”オプションが選択されても、ウィザードは失敗しなくなりました。

## 1-Q17IG

SQL ランタイムは、データベースに Unicode 固有のキャラクタを適切にインストールするようになりました。

## 1-VFA99

SQL ランタイムが、'GO' ステートメントで実行する SQL ステートメントに遭遇しなかったときに、残りの SQL スクリプトをスキップして、次のスクリプトに移動していた問題は解決されました。

## 1-SXCI7

SQL スクリプトは、大規模のデータベースを修復するとき、30 秒ごとにタイムアウトになりました。InstallShield 10.5 では、タイムアウトの間隔は、接続の詳細タブの[コマンドタイムアウト] 編集ボックスを利用して変更することができます。

## 1-Q1FJ5

ナレッジベースの項目番号 Q104762 からのような、大きなシングル行のペーストが原因によるスクリプトエディタのクラッシュは発生しなくなりました。

## Windows Installer プロジェクトタイプ

### 1-TD20H

InstallShield IDE でトランスフォームを作成してから MSI パッケージに適用したあと、MSI がダイレクト MSI モードではなくダイレクト MST モードで開かれていた問題は解決されました。

### 1-PYVK1

\*.reg ファイルをマージモジュールプロジェクトにインポートしても、未設定例外は発生しなくなりました。

### 1-QENT6

Crystal Reports にマージモジュール依存ファイルを作成しようとしたときにエラーが発生していた問題は解決されました。

### 1-NEK0W

[サポートファイル] ビューにファイルを含む MSI プロジェクトに MSI デバッガを実行したとき、“不明なエラー” が発生しなくなりました。

### 1-NHINP

msp をダイレクト MSP モードで編集する場合、ダイレクト MSP を保存して閉じた後で MSP を開くことが可能になりました。

### 1-RT4WR

バイト「5C」で終わる日本語のディレクトリ名が原因でビルド中に IDE が途中終了する問題は解決されました。

## 1-CDGL2

ビルドタイムのスキャンで検索された、2 つ以上の異なる機能で他のファイルによって依存されている依存ファイルは、今回より、それらの機能すべてに添付されます。

## 1-Q5WN5

QuickPatch プロジェクトは、Visual Studio IDE. から再びビルドすることが可能になりました。

## 1-T8S01

古いプロジェクトが 32767 以上のファイルを含む大きいサイズのパッケージをサポートするときも、プロジェクトの移行は失敗しなくなりました。

## 1-Q03FL

COM+ ランタイムは、今回より既存の COM+ アプリケーションをアップデートするとき、それを閉じます。サーバーアプリケーションが、そのインスタンスが作成されている場合に、COM+ コンポーネントがプーリングとキャッシングを行っているときまだメモリーに残っていた問題は解決されました。

## 1-TF7BT

COM+ アプリケーション プロキシのインストールは、.NET サービス コンポーネントにも可能になりました。

## 1-E1QZ2

ダイナミック ファイル リンクで、[ファイルを除外する] ボックスで 255 以上の文字を入力したときに、アプリケーションが "ファイルをすべて含める" ラジオボタンをサイレントで選択していた問題は解決されました。

## 1-C163C

プロジェクトに 12,000 以上のファイルがあるとき、"製品と共にインストールされた ".exe を起動するカスタムアクションを追加しようとしたときインターフェイスが閉じてしまうという問題は解決されました。

## 1-CQI7V

.NET アセンブリのレジストリで作成された Codebase のエントリは、今回より、Regasm ツールによって作成されたレジストリエントリと同じものが入ります。これにより、.NET アセンブリを呼び出したときに発生する COM クライアントの問題が解決されます。

## 1-OOEVD

[完全コントロール] オプションは、“高度な許可” から戻されたあと、[アクセス許可] ダイアログで自動的にチェック解除されることはなくなりました。

## 1-RVBZH

モバイル デバイス インストールのコンポーネントの名前を変更しても、ビルド中にクラッシュが発生することはなくなりました。

## 1-EZER1

MSI コマンドラインの引数は、今回より、[パッチデザイン] ビューの [詳細] タブを通して指定できるようになりました。この変更により、ユーザーは簡単に REINSTALL と REINSTALLMODE のプロパティを設定できるようになりました。

## 1-LRHK1

[変更] オプションは、今回より、ユーザーがそれを選択したとき、アクセス許可があるフォルダに設定されます。

## 1-UYBTH

セットアップ前提条件ダイアログのテキストは、Windows 2000 の日本語環境でも適切に表示されるようになりました。

## 1-V99N9

NT プラットフォーム用 MSDE 2000 オブジェクトは、今回より、セットアップが再起動を必要としたとき、MSDE サービスを起動します。

## 1-LQDID

プロジェクトアシスタントの [インストールのビルド] パネルにある [ビルド後に配布] オプションが、ビルドの後、ファイルの転送に失敗していた問題は解決されました。

## 1-6L4CZ

[セットアップ エラー] にデフォルトで設定されている垂直方向のセンタリングが、親ダイアログで正しく中心に来るように補正されました。

## 1-OMC3U

FeatureGetCost が今回からサポートされ、InstallScript MSI セットアップで呼び出すことができます。

## 1-GOSWA

パッチ セットアップは、今回より、オリジナルインストール中に選択された機能をインスト

ールしません。以前は、初期セットアップ中に選択されたかどうかに関わらず、すべての機能がパッチの実行中に選択されていました。

## 1-NLJ6H

指定された [アプリケーションの追加と削除] アイコンは、64ビットの Windows で適切に設定されます。以前 64ビット Windows では、デフォルトのアイコンは、InstallScript MSI セットアップによってインストールされたアプリケーションに使用されていました。

## 1-UWDLP

すべてのユーザーがログアウトした Windows 2003 で実行される無人インストールは、強化されたセキュリティが原因で失敗することはなくなりました。

## 1-9BVIH

[.NET インストーラ クラス] プロパティが .NET アセンブリに [はい] に設定されているとき、マージモジュール プロジェクトがビルド時にクラッシュしていた問題は解決されました。

## InstallScript プロジェクトタイプ

### 1-B951H

右クリックで [ファイルの追加] を選択し、.url ファイルをファイルグループに追加したとき、"[url] 上記のファイル名は無効です" という警告メッセージが表示されていました。この問題は修正されました。

### 1-MFF1J

別のオブジェクトを含むオブジェクトプロジェクトを登録したときにクラッシュする問題は解決されました。

### 1-T4D7I

追加または削除される特定のテクノロジ (IIS など) に基づいてインターフェイスによって追加または削除される追加のスク립トライブラリ (.obj) は、今回より適切に削除されました。これまでは、特定のテクノロジのサポートが削除された場合、そのライブラリが削除されないということが、あるケースにおいて起きていました。これにより、同じライブラリが数回に渡ってリストされたり、実際必要としていないセットアップに含められてしまうということが起きていました。

### 1-POC6O

[一般情報] ビューの下にある [アプリケーションの追加と削除] プロパティシートの [アイコンの表示] プロパティで指定されたパスは、ランタイムで適切に使用されるようになります。

した。これまでは、指定された値が使用されなかったため、表示アイコンプロパティを使用するためには、スクリプトで設定しなければなりませんでした。

#### 1-V9C9P

レジストリセットでキーを右クリックして \*.reg ファイルをインポートしても、未設定例外は発生しなくなりました。

#### 1-MUJ19

機能のダイアログは、サイズの大きい機能 (2GB 以上) を適切に表示するようになりました。これまで、これらの機能は、機能のダイアログで適切に表示されませんでした。スクリプトで作成された機能は、現在も、各機能のサイズを 2GB に制限しているご注意ください。

#### 1-RWI9P

InstallShield 10.5 より以前のバージョンで作成されたセットアップは、InstallShield 10.5 で作成された差分メディアでアップデートしたあと [アプリケーションの追加と削除] から実行すると適切に動作します。これまで、以前のバージョンでビルドされたセットアップは、現在のバージョンでビルドされた差分メディアを使ってアップデートした場合、セットアップを [アプリケーションの追加と削除] から実行したとき初期化中に初期化エラーが出されていました。

#### 1-S37N2

セットアップ内で複数言語のオブジェクトを実行するためにランタイムで正しい言語の文字列テーブルがロードされます。これまでは、正しい言語の文字列テーブルがオブジェクトによって使用されないというケースがありました。

#### 1-SYJ0P/1-SYJ1B

すべての Update Service 関数は、isrt.obl から 新しい isus.obl へ移動されました。Update Service を含むインストールは、今回から、InstallShield インターフェイスで \*.obl ファイルのリストに isus.obl を自動的に含めます。さらに、プロジェクトで Updated Service サポートが有効化されていると、InstallShield の以前のバージョンから変換されたプロジェクトは isus.obl を自動的に含めます。

#### 1-T4D7Z

UpdateServiceCreateEntryPoints 関数は、今回より、いかなる場合もロギングを許可しません。以前、この関数は、呼び出されると、すべての場合において、結果を戻す前にアンインストールのロギングを許可していました。この関数は InstallShield 10.5 では使用されません。

#### 1-O1GHD

複数の機能によって共有されているコンポーネントにあるファイルは、適切にアンインストールされるようにアンインストールのログファイルに正しくリストされます。以前は、コンポーネントが複数の機能で共有されていて、そのコンポーネントを共有する最初の機能が選択されなかった場合、共有コンポーネントからのファイルは適切にログされず、アンインストール中にも適切に削除されませんでした。

## 1-TOL7P

Is 関数は、nIsFlag パラメータの REBOOTED フラグを今回からサポートします。このフラグは、セットアップが再起動の後に実行されるかどうかを判断するのに利用できます。

## 1-UL522

今回より、セットアップで、ファイルのハンドルがファイル転送中に、指定されたまま開放されないということがなくなりました。これまでセットアップは、ファイル転送中にファイルのハンドルを指定すると、それらをセットアップが完了するまで開放しませんでした。このため、数千のファイルハンドルがサイズの大きいセットアップで指定されてしまい、システムリソースの限界を超えて、セットアップが失敗に終わることがありました。

## 1-VI5K6

複数ディスクのセットアップによって、セットアップ中にオペレーションシステムが“正しいディスクを挿入してください”というプロンプトを出すことがなくなりました。以前、セットアップの実行中またはその終わりで、オペレーティングシステムが disk1 の挿入を求めるプロンプトを表示したり、現在のディスクを削除するとファイル転送エラーが起きることがありました。

オペレーティングシステムにディスクの挿入を求めるプロンプトを出させないようにするには、セットアップ実行中 setup.exe プロセスを常に使用可能な固定の場所から実行してください (temp dir)。このようにすることで、setup.exe は自動的にそれ自身を temp dir にコピーし、コピー (クローン) された setup.exe を起動してセットアップを実行することができます。

コピー/クローンの動作は以下に定義されています。

- '-deleter' が指定されていると、セットアップはいかなる場合もコピーまたはクローンされません。
- Setup.exe は、セットアップが自己展開型 exe で、かつ、'/runfromtemp' が指定されているか、または、セットアップが削除可能なドライブから実行されている、または、セットアップが実行されているドライブタイプが判明できず、セットアップが複数ディスクの場合、それ自身を temp dir にコピーします。

オリジナル setup.exe のプロセスが継続して実行されるかどうかは、以下によって決定されます。

- セットアップが削除可能なドライブから実行されている (かつ、自己展開型 exe では

ない)場合、オリジナルプロセスは待機することなく即座に閉じます。

- '/clone\_nowait' が指定されている場合も、オリジナルプロセスは待機しません。

そうではない場合、オリジナルセットアップはクローンされたセットアップが完了するのを待機します。オリジナル セットアッププロセスが待機しない場合、起動されたセットアッププロセスが完了するのを待機する LAAW および他の機能が、セットアッププロセスが即座に完了することを判明します。これは、初回 setup.exe がセットアップを temp dir から起動したあと即座に終了するためです。これは、OS がセットアップ完了中に disk1 の挿入を求めるプロンプトを出すのを防ぐために必要です。

## 1-ERONX

プロジェクトアシスタントの [インストールのビルド] パネルにある [ビルド後に配布] オプションが、ビルドの後、ファイルの転送に失敗していた問題は解決されました。

## 1-OMAHN

スタティックにリンクされた TTF フォントファイルがある InstallScript セットアップは、正常に移行することができるようになりました。

## 1-MMUVL

最近の (2004) Thawte および Verisign キーは、今回より、OCI の署名にサポートされています。

## 1-RD3W1

複数の SQL ダイアログは、今回より、[次へ] および [戻る] ボタンを使用して普通の個々のダイアログのようにナビゲートできるようになりました。OnSQLServerInitialize に変更があった場合、その修正方法に関する詳細についてはナレッジベースの Q111022 をご覧ください。

## 1-E3BIH

setup.ini が予期せずに変更されたとき、IsCab.exe が途中終了してファイルを切り捨てるという問題は解決されました。

## 1-LIMNY

機能の必要条件に関する制約事項は、機能がランタイムの失敗を解決するためにインターフェイスで再編成される場合、維持されます。

## 1-Q5FIH

パス環境変数は、今回より、その名前をそれがプローブする環境変数として使用します。これにより、移行が適切に行われるようになります。

## 1-IBOJP

今回より、ダイアログが関連ファイルなしにアイコンコントロールまたはビットマップコントロールを含んでいても、6118ビルドエラーは発生しなくなりました。これはまた、プロジェクトでファイルをすべてのビットマップコントロールとアイコンコントロールにマニュアルで指定することでも解決することができます。この修正は、InstallScript MSI プロジェクトにも適用されます。

## 1-ODIQP

LaunchAppAndWait 関数は、今回より、呼び出されたとき結果を戻す前にカレントディレクトリを初期値にリセットします。これまで、カレントディレクトリはリセットされず、起動されるファイルのロケーションに変更されたままでした。この修正は、InstallScript MSI プロジェクトにも適用されます。

## 1-ODIR5

VarSaveとVarRestore 関数は、今回より、nType に VAR\_CURRENTDIR フラグをサポートします。このフラグが指定されると、カレントディレクトリは適切に保存または復元されます。この修正は、InstallScript MSI プロジェクトにも適用されます。

## 1-ODIRG

新しい関数、GetCurrentDirは、スクリプトからカレントディレクトリを取り出すために作成されました。これは、InstallScript MSI プロジェクトにも適用されます。

## 1-PC9AA/1-BM2DY

ServiceStartService は、結果を戻す前にサービスが実行状態になるのを待機します。また、ServiceStopService は以下の機能の変更共、継続して待機します。この変更は、今回より同じ機能性を利用する ServiceStartService にも適用します。これまで、この関数はいかなる場合でも 3 秒後に結果を戻していました。

今回より、この関数は、サービスが最低 dwWaitHint ミリ秒ごとに SERVICE\_STATUS 構造体の dwCheckPoint メンバをアップデートしている間無制限で待機します。また今回よりこの関数は、結果を戻す前にサービスの最終状態をチェックします。以前サービスが適切な時間内で dwCheckPoint をアップデートを終了しなかった場合、この関数が単に失敗 (Failure) を戻している場合があります。

新しい 2 つの SERVICE\_IS\_PARAMS 構造体のメンバが今回より利用可能になりました。

- number nStartServiceWaitCount; // サービスタイムアウトの開始 (秒)
- number nStartServiceWaitCount; // サービスタイムアウトの停止 (秒)

これらの値は ServiceInitParams によって INFINITE に設定されます。サービスが意図された状態に達しているかいないに関わらず、関数がある一定の間隔で結果を強制的に

戻すようにこれらの値を特定の値に変更することができます。これらの値の変更は、ほとんどの場合必要ありません。

ServiceRemoveService を含むすべての関数は、今回より、サービスハンドルを適切に閉じ、これにより、セットアップがまだ実行中にサービスを完全に削除することが可能になりました。この修正は、InstallScript MSI プロジェクトにも適用されます。

#### 1-MMMP5

DeleteFile 関数が、あるシステムでファイルを削除しようとする試みがなされたとき失敗 (failure) の代わりに例外を戻していた問題は解決されました。以前は稀なケースにおいて、DeleteFile 関数はセットアップの失敗の原因になる例外を戻すことがありました。この修正は、InstallScript MSI プロジェクトにも適用されます。

#### 1-TB6KP

次の変数が今回より使用できます。

- FOLDER\_COMMON\_APPDATA- Windows 2000、Windows XP および Windows Server 2003 では、CSIDL\_COMMON\_APPDATA を用いた SHGetSpecialFolderPath の呼び出しに対応するフォルダに設定されています。以前のプラットフォームでの設定値に関する詳細は FOLDER\_COMMON\_APPDATA についての文書をご覧ください。
- FOLDER\_LOCAL\_APPDATA- Windows 2000、Windows XP および Windows Server 2003 では、CSIDL\_LOCAL\_APPDATA を用いた SHGetSpecialFolderPath の呼び出しに対応するフォルダに設定されています。以前のプラットフォームでの設定値に関する詳細は FOLDER\_LOCAL\_APPDATA についての文書をご覧ください。

これらの変数は InstallScript MSI プロジェクトでも使用できます。

#### 1-TB6L8

今回より、以下の REGDB\_KEYPATH 定数がスクリプトコードでの使用できます。

- REGDB\_KEYPATH\_SHARED DLLS-  
"Software¥Microsoft¥Windows¥CurrentVersion¥SharedDLLs¥" に設定されます
- REGDB\_KEYPATH\_RUN- "Software¥Microsoft¥Windows¥CurrentVersion¥Run¥"  
に設定されます
- REGDB\_KEYPATH\_RUNONCE-  
"Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce¥" に設定されます
- REGDB\_KEYPATH\_RUNONCEEX-  
"Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx¥" に設定されます

これらの定数は InstallScript MSI プロジェクトでも使用できます。

#### 1-QYCY5

SQLServerLogin ダイアログは、今回より、関数パラメータで指定された値を使います。これまで関数は、関数により定義されたデフォルトの値を表示していました。

## 1-UVCKD

RegDBSetKeyValueEx 関数は、今回より、埋め込まれたヌル文字を含むバイナリのレジストリ値を適切に設定します。これまでバイナリ値がヌル文字を含んでいた場合、その値は適切に書き込まれませんでした。バイナリ値を設定する際の埋め込まれたテキスト置換のサポートは、今回より対応されていません。バイナリ値を設定する際にテキスト置換を指定する場合、RegDBSetKeyValueEx を呼び出す前に TextSubSubstitute を呼び出して、テキスト置換の値を適切なバイナリデータに変換します。

## Universal Installer プロジェクトタイプ

### 1-EIJ0

多数のマウントポイントを持つ Linux マシンでインストーラがハングしていた問題は解決されました。

### 1-JN2PV

[条件ビルダ] ウィンドウで条件が追加されなかったときも、編集および削除ボタンが有効化されていました。この問題は解決されました。

### 1-JN2Q7

Ctrl キーを押した状態で選択した JVM をクリックすることで、選択した JVM エントリを [JVM の追加] ウィンドウから選択解除できるようになりました。

### 1-O8YPL

Mac OS X AppBundle の配布が、ビルドログで表示されるようになりました。

### 1-O9K33

アクションパネルは今回より適切に表示されるようになりました。そのタブキーを利用してこれらのパネルで [キャンセル] ボタンを選択することができるようになりました。

### 1-OI217

コンソールモードで、パスワードパネルが正しくアクセスできるようになりました。これまで、パスワードのプロンプトは Jaws ユーティリティを使用して読み込まれていませんでした。

### 1-OI220

アクセシビリティを使用したコンソールモードで、[使用許諾書] ダイアログ、[概要] ダイア

ログ、および [機能] ダイアログで Enter をクリックしたときに、ナビゲーション文字列 ' [次へ] は 1 を、[戻る] は 2 を、キャンセルするときは 3 を、または、再表示するときは 5 を押してください' が繰り返し表示されていました。この問題は解決されました。

## 1-Q4AUS

動画の GIF がイメージコントロールに追加されて背景プロパティが変更されたとき、それが消えてしまうという問題は解決されました。

## 1-QNVHA

バンドルされた Windows 用 JVM を使用していて [環境変数アップデート] ビーンがあるとき、プロジェクトのアンインストール中に NullPointerException が戻されていましたが、この問題は解決されました。

## 1-RTFHQ

[別名で保存] メニューオプションを使用して保存されたプロジェクトは、[イベント] ビューで追加されたメソッドを表示するようになりました。

## 1-S4VKX

ひとつのダイアログから別の多数のコントロールがあるダイアログに切り替えたとき、コントロールがフラッシュするという問題は解決されました。

## 1-SEH7P

5.x の機能パネルでエンドユーザーが他の機能で必要な機能を選択解除しようとしたとき、括弧表示 (必須という意味) されて無効化されていました。Universal ではこれはもう表示されません。この問題は解決されました。

## 1-T11X2

プロジェクトを 5.03 から移行する場合、サイレンとモードでオプションファイルと共に起動されると、インストーラは正常に開始します。インストーラは、例外エラー "次のエラーが原因でウィザードは続行できません。無効なコマンドラインオプション: V はサポートされていません (1001) (403)" を返しませんが、

## 1-T44VD

ネイティブ ランチャを実行中、Java の使用エラーは AIX マシン上で表示されなくなりました。次を含むランチャコマンドはログで表示されなくなりました: `-cp ....."/setup.jar3:.....`  
instead of: `-cp ....."/setup.jar".....`

## 1-TOQX9

プロパティボックスの状態を変更後、別のビューに移動してから元のビューに戻っても、プ

ロパティボックス(開いた状態または閉じた状態)が各ビューに記憶されるようになりました。

[トップに戻る](#)

## 既知の問題

既知の問題の全リストは、ナレッジベースの記事 [[Q110942](#)] を参照してください。

[トップに戻る](#)