

初回インストール後に MSI プロパティ値を利用する方法

この文書は Acrecco Software の次の文書を元に記載しています。

<http://www.acresco.com/webdocuments/PDF/MsiPropUninst.pdf>

検証したバージョン: InstallShield 2009 Premier Edition

概要

Windows Installer ではインストール時に設定した MSI プロパティの値をアンインストールやメンテナンス時に使用することができません。この記事では MSI プロパティ値を初回インストール後も使用可能にするためのいくつかのアプローチ方法について記載します。

プロパティの基本

Windows Installer のプロパティの初期値は MSI データベースの Property または Directory テーブルに保持されているものと(例: ProductName, ProductVersion, INSTALLDIR また、ARP から始まるプログラムの追加と削除パネル に使用されるプロパティ 等) セットアップの初期化時に Windows Installer エンジンにより初期化されるもの(例: VersionNT, Version9X, ProgramFilesFolder, AdminUser, Privileged 等)があります。

実行時、プロパティは以下の多くの異なる方法により、追加・変更を行なうことが可能です。

- msixec.exe のコマンドライン (例 `msixec /i product.msi ROPERTYNAME="Value"`)
- ユーザーインターフェイスのコントロールおよびコントロールイベント (例 USERNAME, COMPANYNAME, and ALLUSERS)
- Msi SetProperty や Session.Property などを使用したカスタムアクション

実行時、Windows Installer はパブリックプロパティ(すべて大文字が使用されている)のみを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引き渡します。そして、インストールの終了時にはプロパティの値は原則失われます。

インストール後に自動保存されるプロパティ

ある一部のプロパティの値はインストール時自動的にターゲットシステムに書き込みが行われ、

Windows Installer API や オートメーションを使用することで、インストール後も利用可能です。

例えば初回インストール時に設定される USERNAME, COMPANYNAME, ProductID(使用している場合)の値は、標準アクション RegisterUser によりターゲットシステムのレジストリに記録されます。メンテナンスモード時、MsiGetUserInfo API (InstallScript または DLL ファイルカスタムアクションを使用) を呼び出すことで元々設定されたこの値を取得することが可能です。

加えて、ProductName, ProductVersion, Manufacturer, および ARP がつくプロパティの大半に関しても、インストール時にターゲットシステムへ書き込みが行われ、メンテナンスモード時 MsiGetProductInfo API または ProductInfo オートメーションプロパティ により使用可能です。

予め定義される ARPINSTALLLOCATION プロパティ(プロダクトのインストールディレクトリを示す)はインストール中にターゲットシステムに記録され、MsiGetProductInfo か ProductInfo を使用することでインストール後も利用可能です。

しかしながら、このプロパティは規定では値が入力されていません。メンテナンス時もこの値を取得するために、INSTALLDIR の値をレジストリに記録するには、**ARPINSTALLLOCATION** へ **[INSTALLDIR]** を設定する プロパティのセット カスタムアクションを作成して、実行シーケンスの **CostFinalize** の後ろに 実行条件 **Not Installed** として配置します。(※InstallShield Developer 8.01 以降では、上記のカスタムアクションが自動的に含められます。) 設定後、以下のような VBScript を使用することでメンテナンス時も INSTALLDIR の値を取得することが可能です。

```
str = Installer.ProductInfo(Property("ProductCode"), "InstallLocation")
```

InstallScript では以下の類似した呼出が使用できます。(変数 nBuffer と svINSTALLDIR は宣言されていると仮定します。)

```
nBuffer = MAX_PATH + 1;
MsiGetProductInfo("{ProductCode}", INSTALLPROPERTY_INSTALLLOCATION,
svINSTALLDIR, nBuffer);
```

明示的なプロパティ値の保存・読出

上述の説明を行った以外のプロパティに関しては、インストール中に明示的にターゲットシステムへプロパティ値を保存して、メンテナンス時に読み出す必要があります。カスタムアクションがメンテナンスモード時のみ実行され、初回インストール時に実行されなくするには、“**Installed**” 条件を使用することができます。一般的な方法としては任意のプロパティ値をレジストリに保存方法があります。

レジストリテーブルではデータタイプに“Formatted”を使用しているため、IDE のレジストリビューにて [PROPERTYNAME] のような表記を行なうことができ、この場合プロパティ PROPERTYNAME の値が指定されたレジストリキーに記録されます。(Per-machine レジストリデータに関しては、アプリケーション情報キー HKLM¥Software¥[Manufacturer]¥[ProductName]¥[ProductVersion]に値を保存することが推奨されています。)

メンテナンス時、AppSearch および RegLocator table によりレジストリから値を読み取ることが可能です。また、MSI オートメーションインターフェイスの *RegistryValue* メソッドや InstallScript の *RegDBGetKeyValueEx* 関数を使用しても同様に取得を行なうことが可能です。

レジストリ値を読み取るために AppSearch table を使用するには、「システム検索」ビューを開いて、システム検索テーブルを右クリックして、「追加」を選択します。システム検索ウィザードでは、「レジストリ エントリ」を選択して、記録を行ったルートキー、サブキー、値名を指定して、最後に取得されたデータを保存するプロパティを指定します。

InstallScript を使う場合、以下のコードによりレジストリから値を取得することが可能です。

(使用されている変数はすべて定義済みであると仮定します。)

```
// set the root key
RegDBSetDefaultRoot(HKEY_LOCAL_MACHINE);

// read the desired value data
RegDBGetKeyValueEx(
    "SOFTWARE\\SampleCo\\SampleApp\\1.0.0", // [in] subkey
    "SavedValue",                          // [in] value name

    nvType,                                // [out] value type
    svSavedValue,                          // [out] value, in string form
    nvSize);                               // [out] value size
```

上述の RegDBGetKeyValueEx を呼び出した後、出力先の変数 svSavedValue にはレジストリより取得した値が格納されます。取得した値を MSI プロパティに割り当てる場合は Msi SetProperty 関数を使用可能です。