⌂ WiX managed SDK WixToolset.Dtf.WindowsInstaller namespace ProductInstallation Class

ProductInstallation Class

Represents a unique instance of a product that is either advertised, installed or unknown.

Methods

Method	Description
CollectUserInfo()	Obtains and stores the user information and product ID from an installation wizard.
GetComponentState(component)	Gets the installed state for a product component.
GetFeatureState(feature)	Gets the installed state for a product feature.
GetProducts(productCode, userSid, context)	Enumerates product installations based on certain criteria.
GetRelatedProducts(upgradeCode)	Gets the set of all products with a specified upgrade code. This method lists the currently installed and advertised products that have the specified UpgradeCode property in their Property table.

Properties

Property	Description
AdvertisedLanguage	Product language.
AdvertisedPackageCode	Identifier of the package that a product is installed from.
AdvertisedPackageName	Name of the installation package for the advertised product.
AdvertisedPerMachine	True if the product is advertised per-machine; false if it is per-user or not advertised.
AdvertisedProductIcon	Primary icon for the package.
AdvertisedProductName	Human readable product name.
AdvertisedTransforms	Transforms.
AdvertisedVersion	Version of the advertised product.
AllProducts	Enumerates all product installations on the system.
Features	Gets the set of published features for the product.
HelpLink	The support link.

Property	Description
HelpTelephone	The support telephone.
InstallDate	Date and time the product was installed.
InstallLocation	The installation location.
InstallSource	The installation source.
IsAdvertised	Gets a value indicating whether this product is advertised on the current system.
IsElevated	Checks whether the product is installed with elevated privileges. An application is called a "managed application" if elevated (system) privileges are used to install the application.
IsInstalled	Gets a value indicating whether this product is installed on the current system.
Item	Gets information about an installation of a product.
LocalPackage	The local cached package.
PrivilegedPatchingAuthorized	True if the advertised product can be serviced by non-administrators without elevation.
ProductCode	Gets the ProductCode (GUID) of the product.
ProductId	The product identifier.

Property	Description
ProductName	The installed product name.
ProductVersion	The product version.
Publisher	The publisher.
RegCompany	The company that is registered to use the product.
RegOwner	The owner who is registered to use the product.
SourceList	Gets the source list of this product installation.
UrlInfoAbout	URL about information.
UrlUpdateInfo	The URL update information.

`WixToolset.Dtf.WindowsInstaller.dll` version `4.0.2+644ed0118bae1aa885bb6cc906dc6c0b904de4d9`

CollectUserInfo() Method

Obtains and stores the user information and product ID from an installation wizard.

Declaration

```
public void CollectUserInfo()
```

This method is typically called by an application during the first run of the application. The application first gets the «see P:WixToolset.Dtf.WindowsInstaller.ProductInstallation.ProductId» or «see P:WixToolset.Dtf.WindowsInstaller.ProductInstallation.RegOwner» . If those properties are missing, the application calls CollectUserInfo. CollectUserInfo opens the product's installation package and invokes a wizard sequence that collects user information. Upon completion of the sequence, user information is registered. Since this API requires an authored user interface, the user interface level should be set to full by calling «see M:WixToolset.Dtf.WindowsInstaller.Installer.SetInternalUI(WixToolset.Dtf.WindowsInstaller.InstallUIOptions)» as «see F:WixToolset.Dtf.WindowsInstaller.InstallUIOptions.Full» . The CollectUserInfo method invokes a FirstRun dialog from the product installation database. Win32 MSI API: MsiCollectUserInfo

GetComponentState(component) Method

Gets the installed state for a product component.

Declaration

```
public InstallState GetComponentState(
  string component
)
```

Parameters

Parameter	Type	Description
component	string	The component being queried; GUID of the component as found in the ComponentId column of the Component table.

Return value

`InstallState` Installation state of the component for the product instance: either «see F:WixToolset.Dtf.WindowsInstaller.InstallState.Local» or «see F:WixToolset.Dtf.WindowsInstaller.InstallState.Source» .

Remarks

Win32 MSI API: MsiQueryComponentState

GetFeatureState(feature) Method

Gets the installed state for a product feature.

Declaration

```
public InstallState GetFeatureState(
  string feature
)
```

Parameters

Parameter	Type	Description
feature	string	The feature being queried; identifier from the Feature table

Return value

`InstallState` Installation state of the feature for the product instance: either «see F:WixToolset.Dtf.WindowsInstaller.InstallState.Local» , «see F:WixToolset.Dtf.WindowsInstaller.InstallState.Source» , or «see F:WixToolset.Dtf.WindowsInstaller.InstallState.Advertised» .

Remarks

Win32 MSI APIs: MsiQueryFeatureState , MsiQueryFeatureStateEx

GetProducts(productCode, userSid, context) Method

Enumerates product installations based on certain criteria.

Declaration

```
public static IEnumerable<WixToolset.Dtf.WindowsInstaller.ProductInstallation> GetProducts(
  string productCode,
  string userSid,
  UserContexts context
)
```

Parameters

Parameter	Type	Description
productCode	string	ProductCode (GUID) of the product instances to be enumerated. Only instances of products within the scope of the context specified by the and parameters will be enumerated. This parameter may be set to null to enumerate all products in the specified context.
userSid	string	Specifies a security identifier (SID) that restricts the context of enumeration. A SID value other than s-1-1-0 is considered a user SID and restricts enumeration to the current user or any user in the system. The special SID string s-1-1-0 (Everyone) specifies enumeration across all users in the system. This parameter can be set to null to restrict the enumeration scope to the current user. When is set to the machine context only, must be null.
context	UserContexts	Specifies the user context.

Return value

`IEnumerable<WixToolset.Dtf.WindowsInstaller.ProductInstallation>` An enumeration of product objects for enumerated product instances.

Remarks

Win32 MSI API: MsiEnumProductsEx

GetRelatedProducts(upgradeCode) Method

Gets the set of all products with a specified upgrade code. This method lists the currently installed and advertised products that have the specified UpgradeCode property in their Property table.

Declaration

```
public static IEnumerable<WixToolset.Dtf.WindowsInstaller.ProductInstallation> GetRelatedProducts(
  string upgradeCode
)
```

Parameters

Parameter	Type	Description
upgradeCode	string	Upgrade code of related products

Return value

`IEnumerable<WixToolset.Dtf.WindowsInstaller.ProductInstallation>` Enumeration of product codes

Remarks

Win32 MSI API: MsiEnumRelatedProducts

AdvertisedLanguage Property

Product language.

Declaration

```
public string AdvertisedLanguage { get; set; }
```

AdvertisedPackageCode Property

Identifier of the package that a product is installed from.

Declaration

```
public string AdvertisedPackageCode { get; set; }
```

AdvertisedPackageName Property

Name of the installation package for the advertised product.

Declaration

```
public string AdvertisedPackageName { get; set; }
```

AdvertisedPerMachine Property

True if the product is advertised per-machine; false if it is per-user or not advertised.

Declaration

```
public bool AdvertisedPerMachine { get; set; }
```

AdvertisedProductIcon Property

Primary icon for the package.

Declaration

```
public string AdvertisedProductIcon { get; set; }
```

AdvertisedProductName Property

Human readable product name.

Declaration

```
public string AdvertisedProductName { get; set; }
```

AdvertisedTransforms Property

Transforms.

Declaration

```
public string AdvertisedTransforms { get; set; }
```

AdvertisedVersion Property

Version of the advertised product.

Declaration

```
public System.Version AdvertisedVersion { get; set; }
```

AllProducts Property

Enumerates all product installations on the system.

Declaration

```
public static IEnumerable<WixToolset.Dtf.WindowsInstaller.ProductInstallation> AllProducts { get; set; }
```

Remarks

Win32 MSI API: MsiEnumProducts ,

Features Property

Gets the set of published features for the product.

Declaration

```
public IEnumerable<WixToolset.Dtf.WindowsInstaller.FeatureInstallation> Features { get; set; }
```

Remarks

Because features are not ordered, any new feature has an arbitrary index, meaning this property can return features in any order.

Win32 MSI API: MsiEnumFeatures

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	The installer configuration data is corrupt

HelpLink Property

The support link.

Declaration

```
public string HelpLink { get; set; }
```

HelpTelephone Property

The support telephone.

Declaration

```
public string HelpTelephone { get; set; }
```

Date and time the product was installed.

Declaration

```
public System.DateTime InstallDate { get; set; }
```

InstallLocation Property

The installation location.

Declaration

```
public string InstallLocation { get; set; }
```

InstallSource Property

The installation source.

Declaration

```
public string InstallSource { get; set; }
```

Gets a value indicating whether this product is advertised on the current system.

Declaration

```
public bool IsAdvertised { get; set; }
```

IsElevated Property

Checks whether the product is installed with elevated privileges. An application is called a "managed application" if elevated (system) privileges are used to install the application.

Declaration

```
public bool IsElevated { get; set; }
```

Remarks

Note that this property does not take into account policies such as AlwaysInstallElevated, but verifies that the local system owns the product's registry data.

IsInstalled Property

Gets a value indicating whether this product is installed on the current system.

Declaration

```
public bool IsInstalled { get; set; }
```

Item Property

Gets information about an installation of a product.

Declaration

```
public string Item[
  string propertyName
] { get; set; }
```

Parameters

Parameter	Type	Description
propertyName	string	Name of the property being retrieved.

Remarks

Win32 MSI APIs: MsiGetProductInfo , MsiGetProductInfoEx

Exceptions

Exception	Description
T:System.ArgumentOutOfRangeException	An unknown product or property was requested
T:WixToolset.Dtf.WindowsInstaller.InstallerException	The installer configuration data is corrupt

LocalPackage Property

The local cached package.

Declaration

```
public string LocalPackage { get; set; }
```

PrivilegedPatchingAuthorized Property

True if the advertised product can be serviced by non-administrators without elevation.

Declaration

```
public bool PrivilegedPatchingAuthorized { get; set; }
```

ProductCode Property

Gets the ProductCode (GUID) of the product.

Declaration

```
public string ProductCode { get; set; }
```

ProductId Property

The product identifier.

Declaration

```
public string ProductId { get; set; }
```

Remarks

For more information, see ProductID

ProductName Property

The installed product name.

Declaration

```
public string ProductName { get; set; }
```

ProductVersion Property

The product version.

Declaration

```
public System.Version ProductVersion { get; set; }
```

Publisher Property

The publisher.

Declaration

```
public string Publisher { get; set; }
```

RegCompany Property

The company that is registered to use the product.

Declaration

```
public string RegCompany { get; set; }
```

RegOwner Property

The owner who is registered to use the product.

Declaration

```
public string RegOwner { get; set; }
```

SourceList Property

Gets the source list of this product installation.

Declaration

```
public SourceList SourceList { get; set; }
```

UrlInfoAbout Property

URL about information.

Declaration

```
public System.Uri UrlInfoAbout { get; set; }
```

UrlUpdateInfo Property

The URL update information.

Declaration

```
public System.Uri UrlUpdateInfo { get; set; }
```