



WiX managed SDK

WixToolset.Dtf.WindowsInstaller namespace

PatchInstallation Class

PatchInstallation Class

The Patch object represents a unique instance of a patch that has been registered or applied.

Methods

| Method | Description |
|---|---|
| <code>GetPatches(patchCode, targetProductCode, userSid, context, states)</code> | Enumerates patch installations based on certain criteria. |

Properties

| Property | Description |
|--------------------------|--|
| <code>AllPatches</code> | Enumerates all patch installations on the system. |
| <code>DisplayName</code> | Get the registered display name for the patch. |
| <code>InstallDate</code> | Gets the date and time when the patch is applied to the product. |

| Property | Description |
|---------------|---|
| IsInstalled | Gets a value indicating whether this patch is currently installed. |
| IsObsoleted | Gets a value indicating whether this patch is marked as obsolete. |
| IsSuperseded | Gets a value indicating whether this patch is present but has been superseded by a more recent installed patch. |
| Item | Gets information about a specific patch installation. |
| LocalPackage | Gets the cached patch file that the product uses. |
| MoreInfoUrl | Gets the registered support information URL for the patch. |
| PatchCode | Gets the patch code (GUID) of the patch. |
| ProductCode | Gets the ProductCode (GUID) of the product. |
| State | Gets the installation state of this instance of the patch. |
| Transforms | Gets the set of patch transforms that the last patch installation applied to the product. |
| Uninstallable | True patch is marked as possible to uninstall from the product. |

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

GetPatches(patchCode, targetProductCode, userSid, context, states) Method

Enumerates patch installations based on certain criteria.

Declaration

```
public static IEnumerable<WixToolset.Dtf.WindowsInstaller.PatchInstallation> GetPatches(  
    string patchCode,  
    string targetProductCode,  
    string userSid,  
    UserContexts context,  
    PatchStates states  
)
```

Parameters

| Parameter | Type | Description |
|-------------------|--------|--|
| patchCode | string | PatchCode (GUID) of the patch to be enumerated. Only instances of patches within the scope of the context specified by the and parameters will be enumerated. This parameter may be set to null to enumerate all patches in the specified context. |
| targetProductCode | string | ProductCode (GUID) product whose patches are to be enumerated. If non-null, patch enumeration is restricted to instances of this product within the specified context. If null, |

| Parameter | Type | Description |
|-----------|--------------|---|
| | | the patches for all products under the specified context are enumerated. |
| 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 must be null. |
| context | UserContexts | Specifies the user context. |
| states | PatchStates | The of patches to return. |

Remarks

Win32 MSI APIs: [MsiEnumPatchesEx](#)

AllPatches Property

Enumerates all patch installations on the system.

Declaration

```
public static IEnumerable<WixToolset.Dtf.WindowsInstaller.PatchInstallation> AllPatches { get; set; }
```

Remarks

Win32 MSI API: [MsiEnumPatches](#)

DisplayName Property

Get the registered display name for the patch.

Declaration

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

InstallDate Property

Gets the date and time when the patch is applied to the product.

Declaration

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

IsInstalled Property

Gets a value indicating whether this patch is currently installed.

Declaration

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

IsObsoleted Property

Gets a value indicating whether this patch is marked as obsolete.

Declaration

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

IsSuperseded Property

Gets a value indicating whether this patch is present but has been superseded by a more recent installed patch.

Declaration

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

Item Property

Gets information about a specific patch installation.

Declaration

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

Parameters

| Parameter | Type | Description |
|--------------|--------|---|
| propertyName | string | The property being retrieved; see remarks for valid properties. |

Remarks

Win32 MSI APIs: [MsiGetPatchInfo](#) , [MsiGetPatchInfoEx](#)

Exceptions

| Exception | Description |
|-----------|-------------|
|-----------|-------------|

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

LocalPackage Property

Gets the cached patch file that the product uses.

Declaration

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

MoreInfoUrl Property

Gets the registered support information URL for the patch.

Declaration

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


PatchCode Property

Gets the patch code (GUID) of the patch.

Declaration

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

ProductCode Property

Gets the ProductCode (GUID) of the product.

Declaration

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

State Property

Gets the installation state of this instance of the patch.

Declaration

```
public PatchStates State { get; set; }
```

| Exception | Description |
|--|---|
| T:System.ArgumentException | An unknown patch was requested |
| T:WixToolset.Dtf.WindowsInstaller.InstallerException | The installer configuration data is corrupt |

Transforms Property

Gets the set of patch transforms that the last patch installation applied to the product.

Declaration

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

Remarks

This value may not be available for per-user, non-managed applications if the user is not logged on.

Uninstallable Property

True patch is marked as possible to uninstall from the product.

Declaration

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

Remarks

Even if this property is true, the installer can still block the uninstallation if this patch is required by another patch that cannot be uninstalled.