WiX managed SDK WixToolset.Dtf.WindowsInstaller namespace SourceList Class

SourceList Class

A list of sources for an installed product or patch.

Methods

Method	Description
Add(item)	Adds a network or URL source to the source list of the installed product.
Clear()	Clears sources of all types: network, url, and media.
ClearNetworkSources()	Removes all network sources from the list. URL sources are not affected.
ClearUrlSources()	Removes all URL sources from the list. Network sources are not affected.
Contains(item)	Checks if the specified source exists in the list.
CopyTo(array, arrayIndex)	Copies the network and URL sources from this list into an array.
ForceResolution()	Forces the installer to search the source list for a valid source the next time a source is required. For

Method	Description
	example, when the installer performs an installation or reinstallation, or when it requires the path for a component that is set to run from source.
GetEnumerator()	Enumerates the network and URL sources in the source list of the patch or product installation.
Insert(item, index)	Adds or reorders a network or URL source for the product or patch.
Remove()	Removes a network or URL source.

Properties

Property	Description
Count	Gets the number of network and URL sources in the list.
DiskPrompt	Gets or sets the prompt template that is used when prompting the user for installation media.
IsReadOnly	Gets a boolean value indicating whether the list is read-only. A SourceList is never read-only.
Item	Gets or sets source list information properties of a product or patch installation.
LastUsedSource	Gets or sets the most recently used source location for the product.

Property	Description
LastUsedType	Gets the type of the last-used source.
MediaList	Gets the list of disks registered for the media source of the patch or product installation.
MediaPackagePath	Gets or sets the path relative to the root of the installation media.
PackageName	Gets or sets the name of the Windows Installer package or patch package on the source.

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

Add(item) Method

Adds a network or URL source to the source list of the installed product.

Declaration

```
public void Add(
  string item
)
```

Parameters

Parameter	Type	Description
item	string	Path to the source to be added. This parameter is expected to contain only the path without the filename.

Remarks

If this method is called with a new source, the installer adds the source to the end of the source list. If this method is called with a source already existing in the source list, it has no effect. Win32 MSI APIs: MsiSourceListAddSource , MsiSourceListAddSourceEx

See also

- M:WixToolset.Dtf.WindowsInstaller.SourceList.Insert(System.String,System.Int32)

Clear() Method

Clears sources of all types: network, url, and media.

Declaration

```
public void Clear()
```

Remarks

Win32 MSI API: MsiSourceListClearAll

ClearNetworkSources() Method

Removes all network sources from the list. URL sources are not affected.

Declaration

```
public void ClearNetworkSources()
```

Remarks

Win32 MSI API: MsiSourceListClearAllEx

ClearUrlSources() Method

Removes all URL sources from the list. Network sources are not affected.

Declaration

```
public void ClearUrlSources()
```

Remarks

Win32 MSI API: MsiSourceListClearAllEx

Contains(item) Method

Checks if the specified source exists in the list.

Declaration

```
public bool Contains(
  string item
)
```

Parameters

Parameter	Type	Description
item	string	case-insensitive source to look for

Return value

`bool` true if the source exists in the list, false otherwise

CopyTo(array, arrayIndex) Method

Copies the network and URL sources from this list into an array.

Declaration

```
public void CopyTo(
  System.String[] array,
  int arrayIndex
)
```

Parameters

Parameter	Type	Description
array	System.String[]	destination array to be filed
arrayIndex	int	offset into the destination array where copying begins

ForceResolution() Method

Forces the installer to search the source list for a valid source the next time a source is required. For example, when the installer performs an installation or reinstallation, or when it requires the path for a component that is set to run from source.

Declaration

```
public void ForceResolution()
```

Win32 MSI APIs: MsiSourceListForceResolution , MsiSourceListForceResolutionEx

GetEnumerator() Method

Enumerates the network and URL sources in the source list of the patch or product installation.

Declaration

```
public IEnumerator<System.String> GetEnumerator()
```

Remarks

Win32 MSI API: MsiSourceListEnumSources

Insert(item, index) Method

Adds or reorders a network or URL source for the product or patch.

Declaration

```
public void Insert(
    string item,
    int index
)
```

Parameters

Parameter	Type	Description
item	string	Path to the source to be added. This parameter is expected to contain only the path without the filename.
index	int	Specifies the priority order in which the source will be inserted

Remarks

If this method is called with a new source and *index* is set to 0, the installer adds the source to the end of the source list. If this method is called with a source already existing in the source list and *index* is set to 0, the installer retains the source's existing index. If the method is called with an existing source in the source list and *index* is set to a non-zero value, the source is removed from its current location in the list and inserted at the position specified by Index, before any source that already exists at that position. If the method is called with a new source and Index is set to a non-zero value, the source is inserted at the position specified by *index* , before any source that already exists at that position. The index value for all sources in the list after the index specified by Index are updated to ensure unique index values and the pre-existing order is guaranteed to remain unchanged. If *index* is greater than the number of sources in the list, the source is placed at the end of the list with an index value one larger than any existing source. Win32 MSI API: MsiSourceListAddSourceEx

Remove() Method

Removes a network or URL source.

Declaration

```
public bool Remove()
```

Remarks

Win32 MSI API: MsiSourceListClearSource

Count Property

Gets the number of network and URL sources in the list.

Declaration

```
public int Count { get; set; }
```

DiskPrompt Property

Gets or sets the prompt template that is used when prompting the user for installation media.

Declaration

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

IsReadOnly Property

Gets a boolean value indicating whether the list is read-only. A SourceList is never read-only.

Declaration

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

Item Property

Gets or sets source list information properties of a product or patch installation.

Declaration

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

Parameters

Parameter	Type	Description
property	string	The source list information property name.

Remarks

Win32 MSI API: MsiSourceListGetInfo

Exceptions

Exception	Description
T:System.ArgumentOutOfRangeException	An unknown product, patch, or property was requested

LastUsedSource Property

Gets or sets the most recently used source location for the product.

Declaration

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

LastUsedType Property

Gets the type of the last-used source.

Declaration

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

Remarks

- "n" = network location
- "u" = URL location
- "m" = media location
- (empty string) = no last used source

MediaList Property

Gets the list of disks registered for the media source of the patch or product installation.

Declaration

```
public SourceMediaList MediaList { get; set; }
```

MediaPackagePath Property

Gets or sets the path relative to the root of the installation media.

Declaration

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

PackageName Property

Gets or sets the name of the Windows Installer package or patch package on the source.

Declaration

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