Enumerating Microsoft DirectInput Devices

Article11/06/2009

In order to capture device input using Microsoft DirectInput, you have to know from which device you want to capture, and if it is currently attached to the user's computer. You can do this by enumerating DeviceInstance objects using several different static methods of the Manager class.

- Enumerating All Available DirectInput Devices
- Enumerating a Particular Type of Device
- Enumerating a Certain Class of Device
- Enumerating All Force Feedback Devices

Enumerating All Available DirectInput Devices

You should enumerate when you are looking for a particular kind of device, want to offer the user a choice of devices, or need to work with two or more devices.

Enumeration serves three purposes:

- Reports what DirectInput devices are available.
- Supplies a globally unique identifier (GUID) for each DirectInput device.
- Enables you to create a DeviceInstance object for each DirectInput device as it is enumerated so that you can check the type of device it is and its capabilities.

You can enumerate all the available DirectInput devices on the user's computer by using the Devices property of the Manager class. The following C# sample code is used to populate a TreeView control with all available DirectInput devices.

```
[C#]
private void PopulateAllDevices(TreeView tvDevices)
{
    //Add "All Devices" node to TreeView
    TreeNode allNode = new TreeNode("All Devices");
    tvDevices.Nodes.Add(allNode);

    //Populate All devices
    foreach(DeviceInstance di in Manager.Devices)
    {
```

```
        //Get Device name
        TreeNode nameNode = new TreeNode(di.InstanceName);

        //Is device attached?
        TreeNode attachedNode = new TreeNode(
            "Attached = " +
            Manager.GetDeviceAttached(di.ProductGuid));

        //Get device Guid
        TreeNode guidNode = new TreeNode(
            "Guid = " + di.InstanceGuid);

        //Add nodes
        nameNode.Nodes.Add(attachedNode);
        nameNode.Nodes.Add(guidNode);
        allNode.Nodes.Add(nameNode);
    }

}
```

Enumerating a Particular Type of Device

To enumerate a particular type of device, such as keyboards, you can use the GetDevices method of the Manager class. GetDevices is a static method allows you to retrieve a collection of DeviceInstance objects of a certain type, like keyboards. It has a parameter that gives you the ability to pass it a flag or a combination of flags in order to filter the collection. In this way, you can retrieve a list containing only attached keyboards.

The following C# sample code is used to populate a TreeView control with all attached keyboards.

```
[C#]
private void PopulateKeyboards(TreeView tvDevices)
{
    //Add "Keyboard Devices" node to TreeView
    TreeNode keyboardNode = new TreeNode("Keyboard Devices");
    tvInputDevices.Nodes.Add(keyboardNode);

    //Populate Attached Keyboards
    foreach(DeviceInstance di in

Manager.GetDevices(DeviceType.Keyboard,EnumDevicesFlags.AttachedOnly))
    {

        //Get device name
        TreeNode nameNode = new TreeNode(di.InstanceName);
        TreeNode guidNode = new TreeNode(
            "Guid = " + di.InstanceGuid);
```

```
        //Add nodes
        nameNode.Nodes.Add(guidNode);
        keyboardNode.Nodes.Add(nameNode);
    }

}
```

Enumerating a Certain Class of Device

When enumerating pointing devices on the user's computer, you likely will not want to limit yourself to just a mouse pointer. Some laptops use track-pads or other types of pointing devices.

The C# sample below illustrates how to use the DeviceClass enumeration with the GetDevices method to populate a TreeView control with all attached devices that fall into the screen pointer class, such as mice or track-pads.

```
[C#]
private void PopulatePointers(TreeView tvDevices)
{
    //Add "Pointer Devices" node to TreeView
    TreeNode pointerNode = new TreeNode("Pointer Devices");
    tvInputDevices.Nodes.Add(pointerNode);

    //Populate Attached Mouse/Pointing Devices
    foreach(DeviceInstance di in

Manager.GetDevices(DeviceClass.Pointer,EnumDevicesFlags.AttachedOnly))
    {

        //Get device name
        TreeNode nameNode = new TreeNode(di.InstanceName);
        nameNode.Tag = di;
        TreeNode guidNode = new TreeNode(
            "Guid = " + di.InstanceGuid);

        //Add nodes
        nameNode.Nodes.Add(guidNode);
        pointerNode.Nodes.Add(nameNode);
    }

}
```

Note: This code can also be used to enumerate all types of joysticks by using the **DeviceClass.GameControl** value in the GetDevices method.

Enumerating All Force Feedback Devices

Sometimes you may want to just enumerate devices that have force feedback capabilities. Because many different types of devices use force feedback, you may need to look for any type of device that supports it.

The following C# sample code illustrates how to enumerate all devices that support force feedback, and populate a TreeView control with those devices.

```
[C#]
private void PopulateForceFeedback(TreeView tvDevices)
{
    //Add "ForceFeedback Devices" node to TreeView
    TreeNode forcefeedbackNode = new TreeNode("ForceFeedback Devices");
    tvInputDevices.Nodes.Add(forcefeedbackNode);

    //Populate ForceFeedback Devices
    foreach(DeviceInstance di in
        Manager.GetDevices(
DeviceClass.All,EnumDevicesFlags.ForceFeedback))
    {
        //Get device name
        TreeNode nameNode = new TreeNode(di.InstanceName);
        nameNode.Tag = di;
        TreeNode guidNode = new TreeNode("Guid = " + di.InstanceGuid);

        //Add nodes
        nameNode.Nodes.Add(guidNode);
        forcefeedbackNode.Nodes.Add(nameNode);
    }
}
```