

# Using Force Feedback

Article 11/06/2009

Force Feedback allows you to send messages and properties to a user's Microsoft DirectInput device to relay feedback in the form of physical force. This is only applicable to DirectInput devices that support force feedback.

- Types of Force Feedback
- Creating and Configuring a Device for Force Feedback
- Loading a Force Feedback Effect File
- Assigning a Predefined Effect to a Device

## Types of Force Feedback

There are two methods to relay physical force to an input device. One method is to download a force feedback effect file into the device. A force feedback effect file contains a collection of effects that can be played at any time once they are downloaded into the device.

The other method is to assign a particular effect that is supported by the device. These effects have parameters that can be configured and assigned to axes on the device. The following is a list of effects that are included in Microsoft DirectX 9.0 for Managed Code.

- [ForceFeedbackGuid.ConstantForce](#)
- [ForceFeedbackGuid.CustomForce](#)
- [ForceFeedbackGuid.Damper](#)
- [ForceFeedbackGuid.Friction](#)
- [ForceFeedbackGuid.Inertia](#)
- [ForceFeedbackGuid.RampForce](#)
- [ForceFeedbackGuid.SawtoothDown](#)
- [ForceFeedbackGuid.SawtoothUp](#)
- [ForceFeedbackGuid.Sine](#)
- [ForceFeedbackGuid.Spring](#)
- [ForceFeedbackGuid.Square](#)
- [ForceFeedbackGuid.Triangle](#)

## Creating and Configuring a Device for Force Feedback

In order to use force feedback, you must first create a [Device](#) object for the force feedback compatible device. The following C# sample code creates a [Device](#) object from the first available force feedback-capable game control.

```
[C#]
//create force feedback capable joystick device.
foreach(DeviceInstance di in Manager.GetDevices(
    DeviceClass.GameControl,
    EnumDevicesFlags.AttachedOnly | EnumDevicesFlags.ForceFeedback))
{
    joystick = new Device(di.InstanceGuid);
    break;
}
if (joystick == null)
{
    //Throw exception if joystick not found.
    throw new Exception("No joystick that supports forced feedback
found.");
}
```

Once the [Device](#) object is created, it needs to be configured properly to work with force feedback. In order for force feedback to work properly, you must set the cooperative level to exclusive. You also need to create an array of axes to use with the force feedback.

The following C# sample code illustrates how to configure the [Device](#) object created in the previous code.

```
[C#]
//set cooperative level.
joystick.SetCooperativeLevel(
    this,
    CooperativeLevelFlags.Exclusive | CooperativeLevelFlags.Background);

//Set axis mode absolute.
joystick.Properties.AxisModeAbsolute = true;

//Acquire joystick for capturing.
joystick.Acquire();

//Configure axes
int[] axis = null;
foreach(DeviceObjectInstance doi in joystick.Objects)
{
    //Set axes ranges.
    if((doi.ObjectId & (int)DeviceObjectTypeFlags.Axis) != 0)
```

```

{
    joystick.Properties.SetRange(
        ParameterHow.ById,
        doi.ObjectId,
        new InputRange(-5000,5000));
}

int[] temp;

// Get info about first two FF axii on the device
if ((doi.Flags & (int)ObjectInstanceFlags.Actuator) != 0)
{
    if (axis != null)
    {
        temp = new int[axis.Length + 1];
        axis.CopyTo(temp,0);
        axis = temp;
    }
    else
    {
        axis = new int[1];
    }

    // Store the offset of each axis.
    axis[axis.Length - 1] = doi.Offset;
    if (axis.Length == 2)
    {
        break;
    }
}
}
}

```

## Loading a Force Feedback Effect File

The following C# sample code shows you how to download a force feedback effect file into the [Device](#) object created and configured in previous samples.

```

[C#]
//Load force feedback effect file
string path =
    @"C:\Program Files\Microsoft DirectX 9.0 SDK (December
2004)\Samples\Managed\DirectInput\ReadFFE\guna.ffe";

EffectList el = null;
el = joystick.GetEffects(path,FileEffectsFlags.ModifyIfNeeded);
foreach(FileEffect fe in el)
{
    EffectObject feo = new EffectObject(
        fe.EffectGuid,

```

```

        fe.EffectStruct,
        joystick);
    try
    {
        feo.Download();
    }
    catch(Exception ex)
    {
        throw new Exception("Could not download force feedback effect
file.", ex);
    }
    fxList.Add(feo);
}

```

Once the above force feedback effect file is downloaded into the [Device](#) object, it can be played by using the following C# sample code.

```

[C#]
//Play gun force feedback file effect
foreach(EffectObject eo in fxList)
{
    eo.Start(1,EffectStartFlags.NoDownload);
}
button0pressed = true;

```

## Assigning a Predefined Effect to a Device

If you want to relay a physical force such as a constant force to the axes of a [Device](#) object, you can do this by assigning a predefined effect to its axes. The following C# sample code first checks to see if the [Device](#) object supports the effect. It then creates the effects, sets some parameters for the effects, then applies them to the axes of the [Device](#) object.

```

[C#]
//See if joystick supports ConstantForce and set it.
EffectObject eo = null;
Effect e;

foreach (EffectInformation ei in joystick.GetEffects(EffectType.All))
{
    //If the joystick supports ConstantForce, then apply it.

    if (DInputHelper.GetTypeCode(ei.EffectType)

```

```

        == (int)EffectType.ConstantForce)
    {
        // Fill in some generic values for the effect.
        e = new Effect();
        e.SetDirection(new int[axis.Length]);
        e.SetAxes(new int[axis.Length]);
        e.ConditionStruct = new Condition[axis.Length];

        e.EffectType = EffectType.ConstantForce;
        e.Duration = (int)DI.Infinite;
        e.Gain = 10000;
        e.SamplePeriod = 0;
        e.TriggerButton =
(int)Microsoft.DirectX.DirectInput.Button.NoTrigger;
        e.TriggerRepeatInterval = (int)DI.Infinite;
        e.Flags = EffectFlags.ObjectOffsets | EffectFlags.Cartesian;
        e.SetAxes(axis);

        // Create the effect, using the passed in guid.
        eo = new EffectObject(ei.EffectGuid, e, joystick);
    }
}

if (eo == null)
{
    throw new Exception("ConstantForce is not supported.");
}

```