The following standard LINQ query operators belong to **Conversion Operators**category  
ToList  
ToArray  
ToDictionary  
ToLookup  
Cast  
OfType  
AsEnumerable   
AsQueryable   
  
   
  
We discussed the following operators in [Part 15](http://csharp-video-tutorials.blogspot.com/2014/07/part-15-conversion-operators-in-linq.html)  
ToList  
ToArray  
ToDictionary  
ToLookup  
  
**In this video we will discuss**  
**1.** Cast and OfType operators  
**2.** Difference between Cast and OfType operators  
**3.** When to use one over the other  
  
**Cast operator**attempts to convert all of the items within an existing collection to another type and return them in a new collection. If an item fails conversion an exception will be thrown. This method uses deferred execution.  
  
**Example :**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

namespace Demo

{

    class Program

    {

        public static void Main()

        {

            ArrayList list = new ArrayList();

            list.Add(1);

            list.Add(2);

            list.Add(3);

            // The following item causes an exception

            // list.Add("ABC");

            IEnumerable<int> result = list.Cast<int>();

            foreach (int i in result)

            {

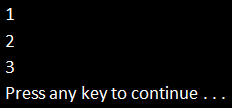
                Console.WriteLine(i);

            }

        }

    }

}

**Output :**   
   
  
**OfType operator**will return only elements of the specified type. The other type elements are simply ignored and excluded from the result set.  
  
**Example :** In the example below, items **"4"**and **"ABC"**will be ignored from the result set. No exception will be thrown.

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

namespace Demo

{

    class Program

    {

        public static void Main()

        {

            ArrayList list = new ArrayList();

            list.Add(1);

            list.Add(2);

            list.Add(3);

            list.Add("4");

            list.Add("ABC");

            IEnumerable<int> result = list.OfType<int>();

            foreach (int i in result)

            {

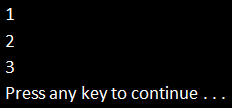
                Console.WriteLine(i);

            }

        }

    }

}

**Output** :   
   
  
**What is the difference between Cast and OfType operators**  
OfType operator returns only the elements of the specified type and the rest of the items in the collection will be ignored and excluded from the result.   
  
Cast operator will try to cast all the elements in the collection into the specified type. If some of the items fail conversion, InvalidCastException will be thrown.  
  
**When to use Cast over OfType and vice versa?**  
We would generally use Cast when the following 2 conditions are met  
**1.** We want to cast all the items in the collection &  
**2.** We know for sure the collection contains only elements of the specified type  
  
If we want to filter the elements and return only the ones of the specified type, then we would use OfType.