**How to: Create a Windows Communication Foundation Contract with a Class**

The preferred way of creating a Windows Communication Foundation (WCF) contract is by using an interface. For more information, see [How to: Define a Windows Communication Foundation Service Contract](http://msdn.microsoft.com/en-us/library/ms731835.aspx). An alternative, outlined here, is to create a class and then apply the [ServiceContractAttribute](http://msdn.microsoft.com/en-us/library/system.servicemodel.servicecontractattribute.aspx) class to the class directly and the [OperationContractAttribute](http://msdn.microsoft.com/en-us/library/system.servicemodel.operationcontractattribute.aspx) class to each of the methods in the class that are part of the contract.

For more information about service contracts, see [Designing Service Contracts](http://msdn.microsoft.com/en-us/library/ms733070.aspx).

**Creating a Windows Communication Foundation contract with a class**

1. Create a new class using Visual Basic, C#, or any other common language runtime language.
2. Apply the **ServiceContractAttribute** class to the class.
3. Create methods in the class.
4. Apply the **OperationContractAttribute** class to each method that must be exposed as part of the public WCF contract.

**Example**

The following code example shows a class that defines a service contract.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl05_code');" \o "Copy Code)

<ServiceContract()> \_

Public Class CalculatorService

<OperationContract()> \_

Public Function Add(ByVal n1 As Double, ByVal n2 As Double) As Double

Return n1 + n2

End Function

<OperationContract()> \_

Public Function Subtract(ByVal n1 As Double, ByVal n2 As Double) As Double

Return n1 - n2

End Function

<OperationContract()> \_

Public Function Multiply(ByVal n1 As Double, ByVal n2 As Double) As Double

Return n1 \* n2

End Function

<OperationContract()> \_

Public Function Divide(ByVal n1 As Double, ByVal n2 As Double) As Double

Return n1 / n2

End Function

End Class

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl06_code');" \o "Copy Code)

[ServiceContract]

public class CalculatorService

{

[OperationContract]

public double Add(double n1, double n2)

{

return n1 + n2;

}

[OperationContract]

public double Subtract(double n1, double n2)

{

return n1 - n2;

}

[OperationContract]

public double Multiply(double n1, double n2)

{

return n1 \* n2;

}

[OperationContract]

public double Divide(double n1, double n2)

{

return n1 / n2;

}

}

The methods that have the **OperationContractAttribute** class applied use a request-reply message pattern by default. For more information about this message pattern, see [How to: Create a Request-Reply Contract](http://msdn.microsoft.com/en-us/library/ms732381.aspx). You can also create and use other message patterns by setting properties of the attribute. For more examples, see [How to: Create a One-Way Contract](http://msdn.microsoft.com/en-us/library/ms733035.aspx) and [How to: Create a Duplex Contract](http://msdn.microsoft.com/en-us/library/ms731184.aspx).