

در WCF سه نوع Message Encoder وجود دارد:

Text(Xml) Message Encoder (به صورت پیش فرض در تمام Http-Base Binding ها از این Encoder استفاده می‌شود)

Binary Message Encoder (به صورت پیش فرض در تمام Net\* Binding ها از این encoder استفاده می‌شود که برای سرویس‌های وب مناسب نیست)

MTOM Message Encoder (در حالت استفاده از Http-Base Binding ها و انتقال اطلاعات به صورت باینری از این گزینه استفاده می‌شود که به صورت پیش فرض غیر فعال است)  
 Encoding یا رمزگذاری در WCF به این معنی است که داده‌های مورد نظر برای انتقال، به یکی از فرمت‌های Text-Xml ، MTOM یا Binary سریالایز شوند.

وضعیتی را در نظر بگیرید که در یک پروژه مبتنی بر WCF قصد دارید حجمی زیاد از داده به فرمت باینری (نظیر فایل ها) را بین سرور و کلاینت رد و بدل کنید. به صورت معمول بسیاری از برنامه نویسان، یک کلاس به همراه DataContractAttribute ایجاد می‌کنند که در آن خاصیتی به صورت آرایه ای از نوع بایت که DataMemberAttribute را نیز دارد برای انتقال محتویات فایل استفاده می‌شود. اما باید یک نکته را مد نظر داشت و آن این است که به صورت پیش فرض فرمت انتقال داده‌ها در WCF به صورت Text/Xml است و برای انتقال داده‌ها نیز از فرمت [Base 64](#) استفاده خواهد شد. مشکل اصلی این جاست که در حالت Text/Xml Encoding برای انتقال داده‌های باینری، برای هر سه بایت، چهار کاراکتر استفاده می‌شود در نتیجه، این باعث افزایش حجم داده تا 33 درصد خواهد شد که کارایی سیستم را تحت تاثیر قرار می‌دهد.

اما خبر خوب این است که استاندارد وجود دارد به نام **MTOM** یا همان **M essage T ransmission O ptimization M echanism**، برای این که بتوان محتوای باینری را بدون افزایش حجم داده انتقال داد. برای پیاده سازی این روش باید موارد زیر را در نظر داشته باشید:

«متد یا همان OperationContract که وظیفه آن ارسال یا دریافت داده‌ها با فرمت MTOM است فقط کلاس هایی را انتقال دهد که دارای MessageContractAttribute هستند. نباید از DataContractAttribute استفاده نمایید.  
 «خاصیتی که نوع آن آرایه ای از بایت‌ها است نباید دارای DataMemberAttribute باشد؛ بلکه به جای آن باید از MessageBodyMember استفاده نمایید.

«به جای [Byte] می‌توان از نوع Stream نیز استفاده کرد(الزامی نیست).  
 «مقدار خاصیت MessageEncoding در Binding استفاده شده باید MTOM تعیین شود.

### پیاده سازی یک مثال

ابتدا کلاس مورد نظر را به صورت زیر تهیه می‌کنیم:

```
[MessageContract]
public class MyFile
{
    [MessageHeader]
    public String Filename { get; set; }

    [MessageBodyMember]
    public Byte[] Contents { get; set; }
}
```

## چند تذکر

به جای DataContract از MessageContract استفاده می‌شود؛  
تمام خاصیت‌هایی که نوع آن‌ها غیر از Byte[] است باید دارای MessageHeader باشند؛  
خاصیتی که برای انتقال محتوای باینری تهیه شده است، باید از MessageBodyMember استفاده نماید؛  
مجاز به تعریف فیلد یا فیلدهایی که نوع آن‌ها Primitive Type است نمی‌باشید.

تنظیمات مربوط به Binding نیز به صورت خواهد بود:

```
<bindings>
  <wsHttpBinding>
    <binding name="WsHttpMtomBinding" messageEncoding="Mtom" />
  </wsHttpBinding>
</bindings>
```

## اما یک نکته...

هدف از استفاده از MTOM برای افزایش کارایی انتقال داده‌های باینری در حجم زیاد است. در زیر نتایج مقایسه بررسی انتقال اطلاعات به دو صورت MTOM و Text برای حجم داده‌های متفاوت را مشاهده می‌کنید:

```
Text encoding with a 100 byte payload: 433
MTOM encoding with a 100 byte payload: 912

Text encoding with a 1000 byte payload: 1633
MTOM encoding with a 1000 byte payload: 2080

Text encoding with a 10000 byte payload: 13633
MTOM encoding with a 10000 byte payload: 11080

Text encoding with a 100000 byte payload: 133633
MTOM encoding with a 100000 byte payload: 101080

Text encoding with a 1000000 byte payload: 1333633
MTOM encoding with a 1000000 byte payload: 1001080
```

با دقت در نتایج بالا مشخص می‌شود که این روش در حجم داده‌های پایین (مثل 100 بایت یا 1000 بایت) عملکرد مورد انتظار را نخواهد داشت. پس این نکته را نیز در هنگام پیاده‌سازی به این روش مد نظر داشته باشید.