



ADEPTIA

---

443 North Clark St, Suite 350  
Chicago, IL 60654  
Phone: (312) 229-1727

# MTOM Support Guide

August 28, 2012



## Table of Contents

Introduction .....	3
Applies To.....	3
Enable MTOM in Web Service Consumer.....	3
Transform the SOAP attachment in Base64Binary encoding .....	6
Example Process Flow.....	9



## Introduction

Transmitting attachments such as Images, X-Rays, Design Documents and Business Contracts using SOAP messages has now become a common practice.

MTOM (Message Transmission Optimization Mechanism) is a way for optimizing the electronic transmission of attachments. It provides an elegant mechanism of efficiently transmitting binary data, such as images, PDF files and MS Word documents.

The MTOM message format allows bit stream compression of binary data. Data that would otherwise have to be encoded in the SOAP message is instead transmitted as raw binary data in a separate MIME part. A large chunk of binary data takes up less space than its encoded representation, so MTOM can reduce transmission time, although it can increase processor usage. Elements to be transmitted in this way are defined as base64Binary in the WSDL (XML Schema).

MTOM only optimizes element content that is in the canonical lexical representation of the base64Binary data type. Since there is no standard way to indicate whether data is in the canonical lexical representation, the mechanism for applying MTOM is implementation dependent.

This document provides high level guidelines of how to use MTOM properties for Web Service Consumer in Adeptia Suite to achieve the transmission of binary data through SOAP messages. Following are the high level steps for using MTOM:

1. [Enable MTOM option in WS Consumer](#)
2. [Transform the SOAP attachment in Base64Binary encoding](#)
3. [Example Process Flow](#)

## Applies To

Support for MTOM feature is added in Adeptia Suite version 5.3. This document applies to Adeptia Suite version 5.3 and later.

## Enable MTOM in Web Service Consumer

1. Go to Design ->Services -> Web Services -> Consumer
2. Create a new Web Service Consumer activity using URI.



**[+] Standard properties**

Specify web service definition location.

Name *	<input type="text" value="TestConsumer"/>
Description *	<input type="text" value="testing purpose"/>
Character Set Encoding *	<input type="text" value="ISO-8859-1"/>
Consumer Type *	<input checked="" type="radio"/> SOAP <input type="radio"/> REST
Web Service Definition Location*	<input type="radio"/> UDDI <input checked="" type="radio"/> URI
URI Location	<input checked="" type="radio"/> HTTP <input type="radio"/> Local/LAN
WSDL URL (HTTP URL)*	<input type="text" value="http://localhost:8080/wsx/services/S"/>
WSDL File Path(Local/Lan)*	<input type="text"/> <input type="button" value="Browse Wsdl"/>
Secure	<input type="checkbox"/>
UserId	<input type="text"/>
Password	<input type="text"/>

3. Attach the corresponding WSDL or fill WSDL URL and click next (Refer above figure).

**Web Service Consumer**

Select operation name for web service consumer activity TestConsumer.

Name *	<input type="text" value="TestConsumer"/>
Description *	<input type="text" value="testing purpose"/>
Service Name *	<input type="text" value="Service1_MTOM_Provider"/>
Ports *	<input type="text" value="Service1Soap"/>
Port Type *	<input type="text" value="Service1Soap"/>
Operation(s) *	<input type="text" value="Hello"/> <input type="text" value="RoundTrip"/>
Style Name *	<input type="text" value="document"/>

4. Select service name, ports, port type and operations etc and click next.

**WebServices > Consumer > TestConsumer**

**[-] Standard properties**

Name *	<input type="text" value="TestConsumer"/>
Description *	<input type="text" value="testing purpose"/>
Service Name *	<input type="text" value="Service1_MTOM_Provider"/>
Character Set Encoding *	<input type="text" value="ISO-8859-1"/>
Ports *	<input type="text" value="Service1Soap"/>
Port Type *	<input type="text" value="Service1Soap"/>
Operation *	<input type="text" value="RoundTrip"/>
Style *	<input type="text" value="document"/>
SOAP Version	<input type="text" value="SOAP 1.1"/>
Security Policy	<input type="text" value="None"/>
Maintain Session	<input type="checkbox"/>
Format Type	<input type="text" value="None"/>
Override SoapAction	<input type="checkbox"/>
<b>Enable MTOM</b>	<input checked="" type="checkbox"/>

**[+] WS-A addressing properties**  
**[+] Advanced properties**  
 \* Mandatory fields.

5. Check the Enable MTOM checkbox to enable MTOM (As highlighted).
6. Click on Save.



## Transform the SOAP attachment in Base64Binary encoding

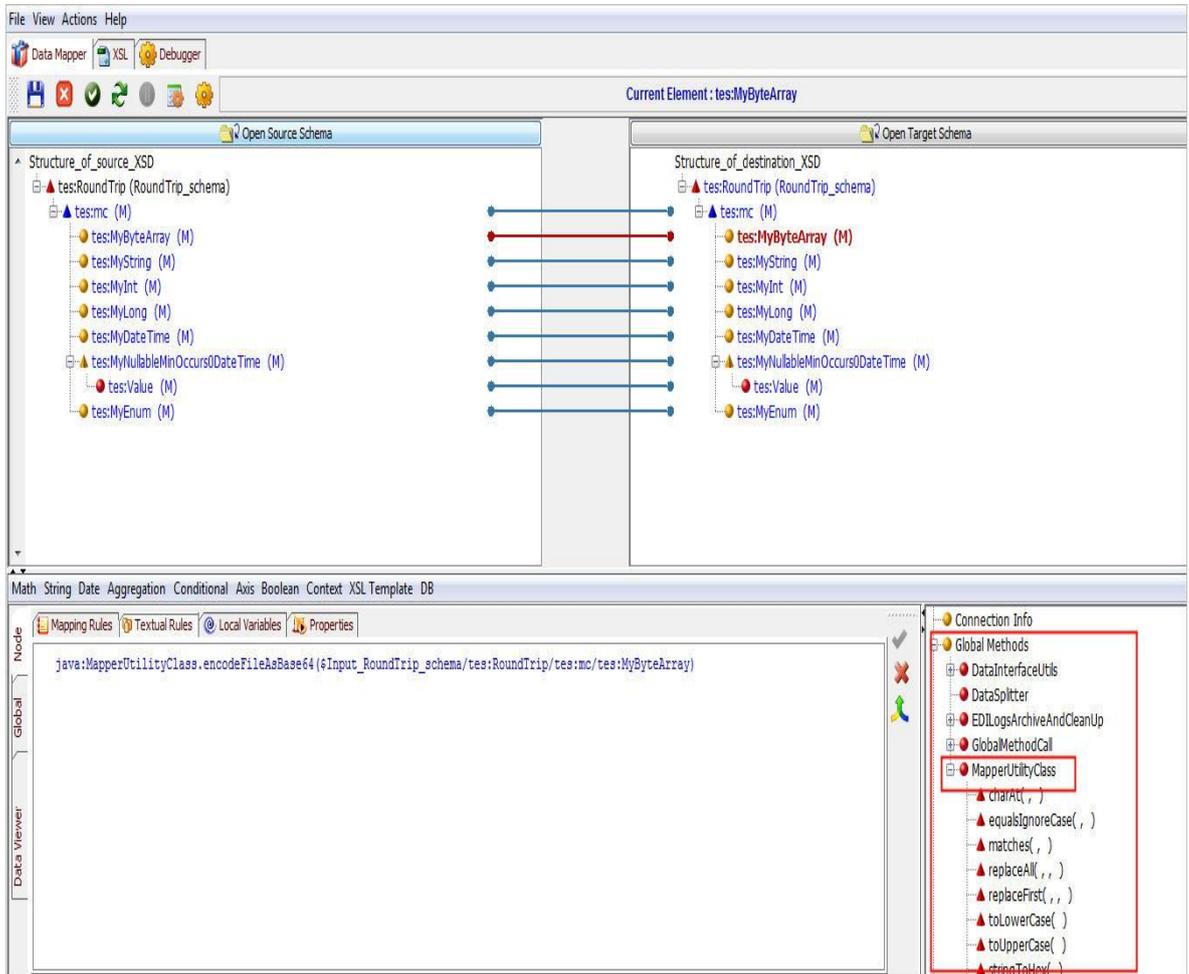
The SOAP Message should contain the attachments in Base64Binary format.

To achieve this use Adeptia Data Mapping which can transform the attachment in *Base64Binary* encoded *format*.

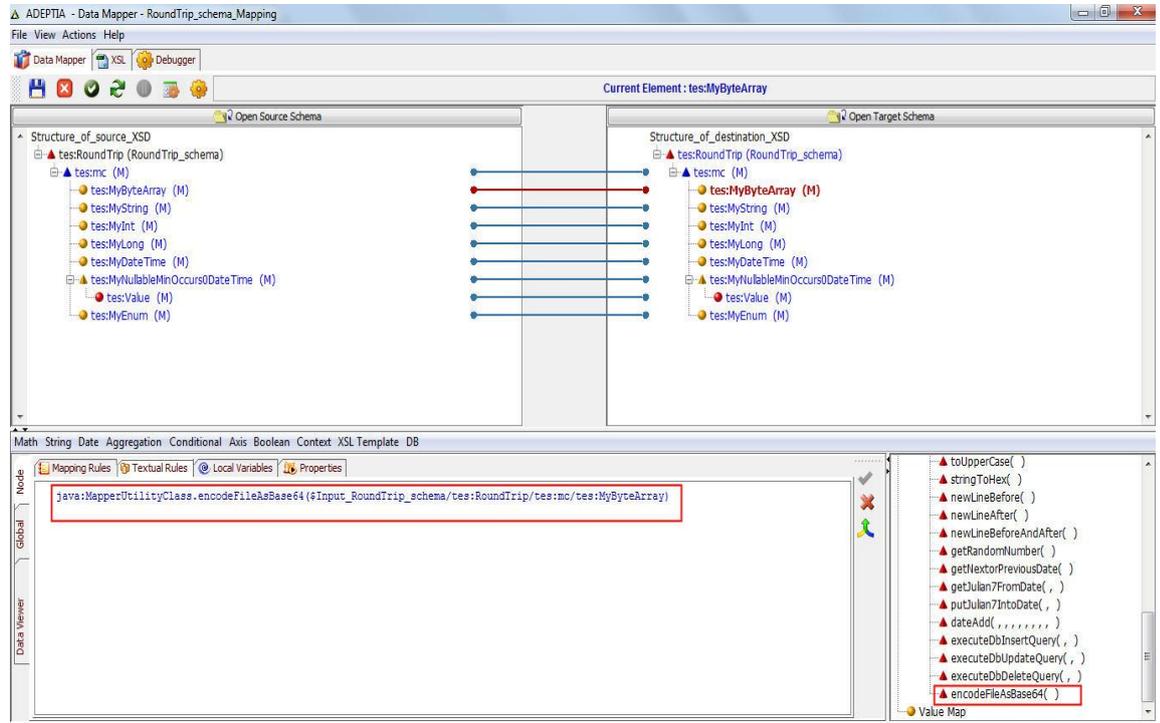
Pass the SOAP Message request to Data Mapper and transform the SOAP attachment to Base64 in Data Mapper. Please follow below steps to achieve this.

### Steps

1. Create a XML schema using SOAP Message request XML. For detailed description of XML schema creation please refer to the Developer guide.
2. Go to Design -> Services-> Data Transform -> Data Mapping
3. Create new Data Mapping and load the XML schema created by the SOAP Message request XML on both source and target end in Data Mapper.
4. To transform the attachment field in Base64Binary encoded data, follow the steps as below:
  - a) Map the attachment field in the source to the respective Target attachment field.
    - Select the attachment field of the target schema .
    - Open the Textual rule tab.
    - Go to the Global Methods -> MapperUtilityClass on the parameter panel as highlighted in figure:



- Double click on encodeFileAsBase64 function to bring in the Textual rule Area. Please see the figure below.

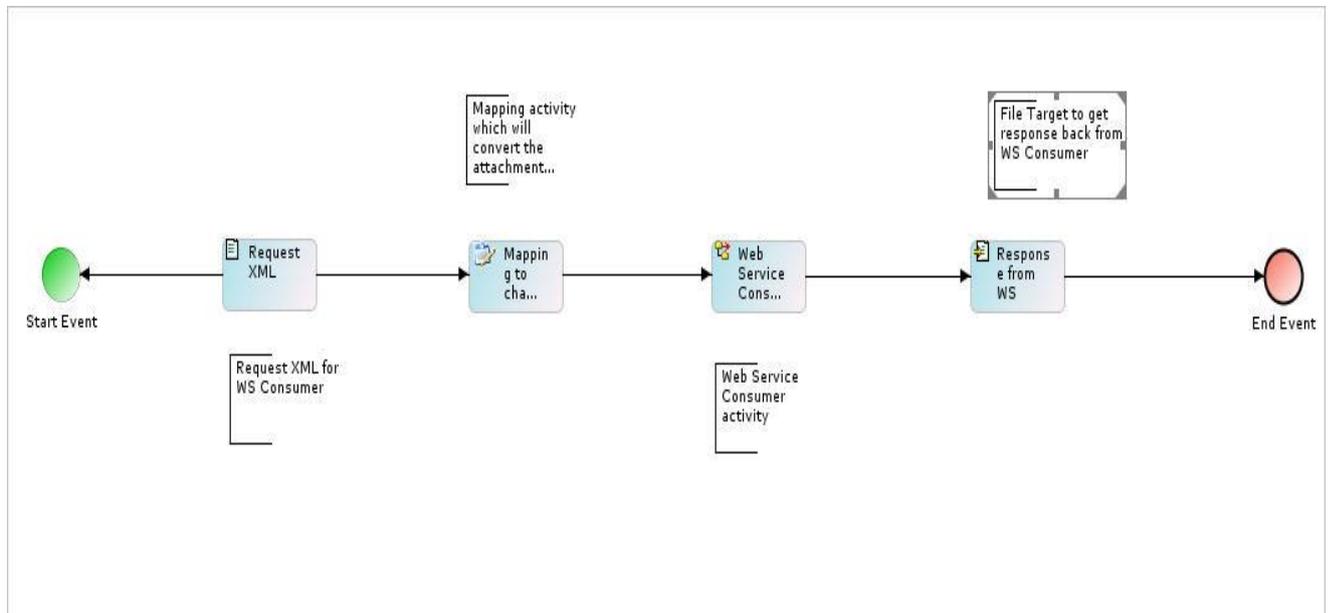


- Double click the source element of the source schema which contains the data of attachments. Please see the figure below.
  - Click the Apply Mapping (  ) button.
- b) Map the other elements as one to one and Save the Mapping.



## Example Process Flow

1. Create a File source activity which will contain the SOAP Message request XML. For details how to create file source activity please refer to the Developer Guide.
2. Create a File target activity which will contain the response from the Web Service Consumer. For details how to create File target activity please refer to the Developer Guide.
3. Go to Design -> Process Flow -> Process Flow
4. Create a new Process flow and add the activities as in below figure:



- Click on save and execute the flow.
- After successful execution of the flow, the file target activity will contain the response back from the WS consumer activity and it will contain the binary data in Base64 format.