|  |
| --- |
| **Content Modifier** |
| **(Training Material) Version: 1.0** |

# Table of Contents

[Table of Contents 2](#_Toc1400985)

[**Document Control** 2](#_Toc1400986)

[**1.** **Use of content Modifier** 3](#_Toc1400987)

[**2.** **Understanding of Content Modifier** 4](#_Toc1400988)

[**2.1** **Message Header** 4](#_Toc1400989)

[**2.2** **Exchange Property:** 4](#_Toc1400990)

[**2.3** **Message Body:** 5](#_Toc1400991)

[**3.** **Action: Create** 6](#_Toc1400992)

[**3.1 Constant Type Header:** 6](#_Toc1400993)

[**3.2 Header Type Header:** 6](#_Toc1400994)

[**3.3 XPath Type Header:** 6](#_Toc1400995)

[**3.4 Expression Type Header:** 6](#_Toc1400996)

[**3.5 Property Type Header:** 7](#_Toc1400997)

[**3.6 Number-Range Type Header:** 7](#_Toc1400998)

[**3.7 Global Type Header:** 9](#_Toc1400999)

[**4.** **Action: Delete** 9](#_Toc1401000)

# **Document Control**

**Revision History:**

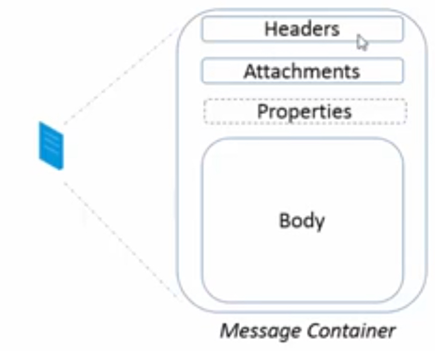
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Reviewer** | **Revision Notes** |
| 1.0 | 18-Feb-2019 | Netrey Powdwal Sayantani Deb | Binod Kumar Mondal | First release of document |
|  |  |  |  |  |
|  |  |  |  |  |

# **Use of content Modifier**

You use the content modifier step to modify the content of incoming message by providing additional information in the header or body of the message.

The Content Modifier allows you to modify a message by changing the content of the data containers that are involved in message processing (message header, message body, or message exchange).

Depending on which container you want to modify, select one of the tabs ***Message Header***, ***Message Body***, or ***Exchange Property***.



**Message Body**: It consists of any data/content which we need to send to the receiver. It can be in form of xml, json, etc. whichever format the sender and receiver agree on.

**Attachments:** They are the additional messages attached to the main message like in MAIL, ARIBA adapter’s responses.

**Message Header:** They are name-value pairs associated with the message. Each sender has its own set of headers in response, when called. Also, we can set additional or remove headers from a message.

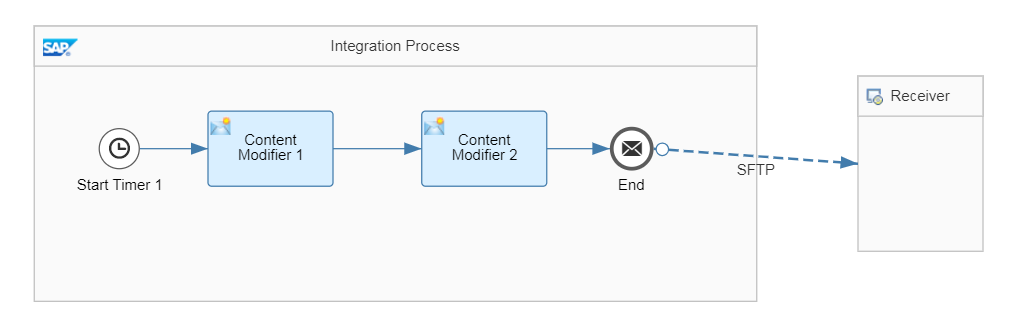
**Exchange Property:** They are also headers which can store name-value pairs.

**Difference between Message Header and Exchange Property:**

The answer is ***Persistence***. Exchange Properties persist only till the life of a message transaction whereas Message Headers move along with the message to the receiver.

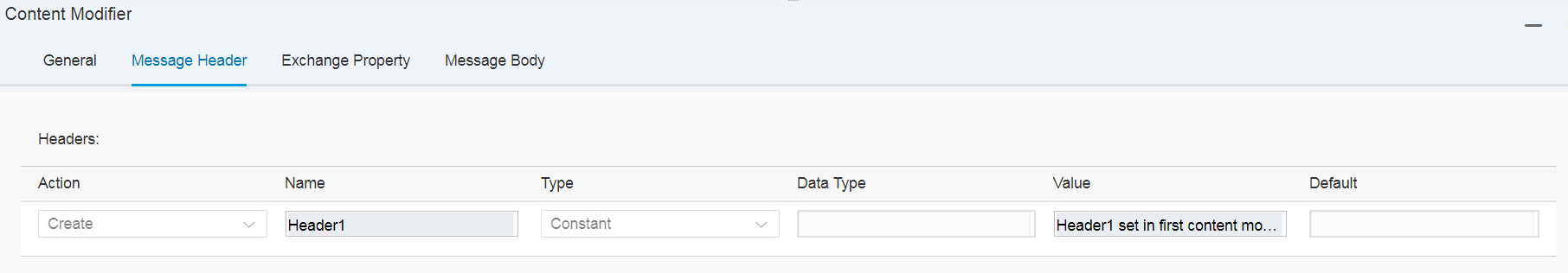
# **Understanding of Content Modifier**

Let us understand content modifier with help of example:

Consider an IFLOW as below:

**Content Modifier 1:**

## **Message Header**

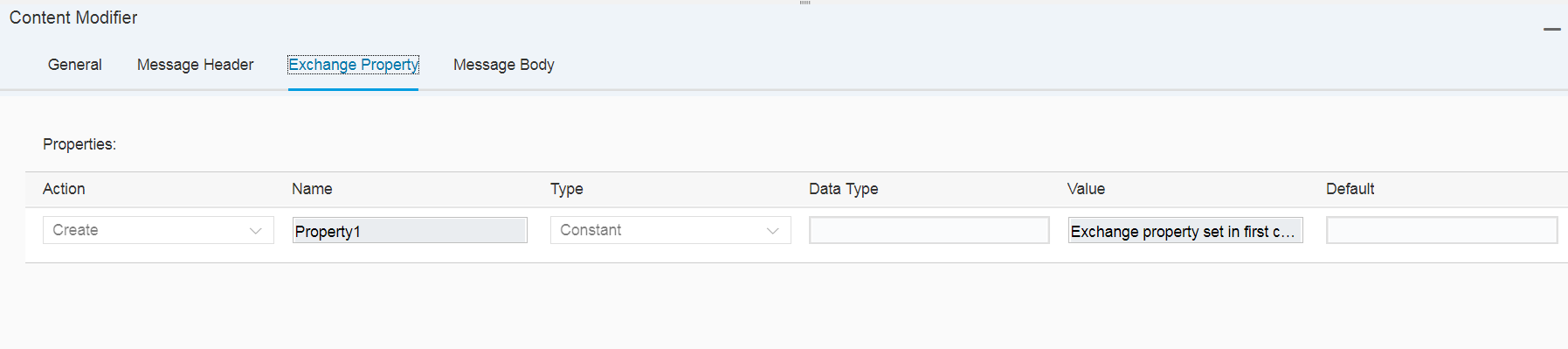


Name: ***Header1***

Type: ***Constant***

Value: ***Header1 set in first content modifier***

## **Exchange Property:**

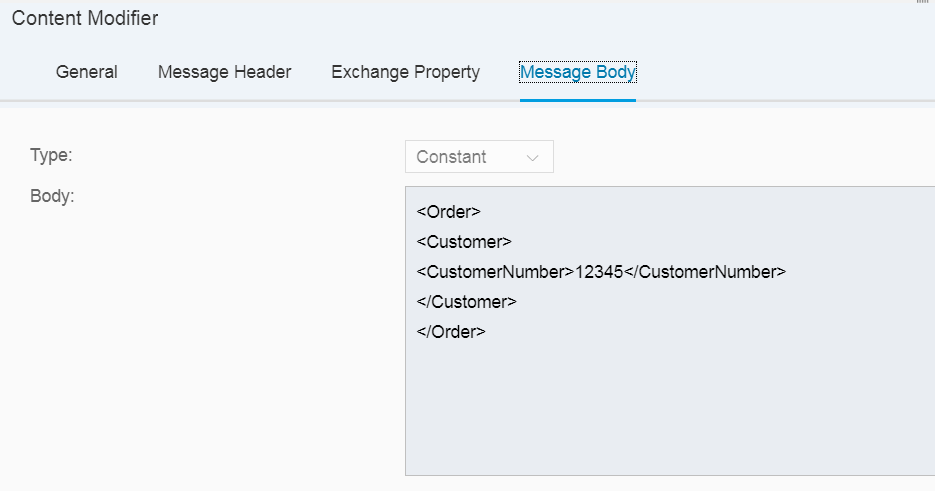


Name: ***Property1***

Type: ***Constant***

Value: ***Exchange property set in first content modifier***

## **Message Body:**



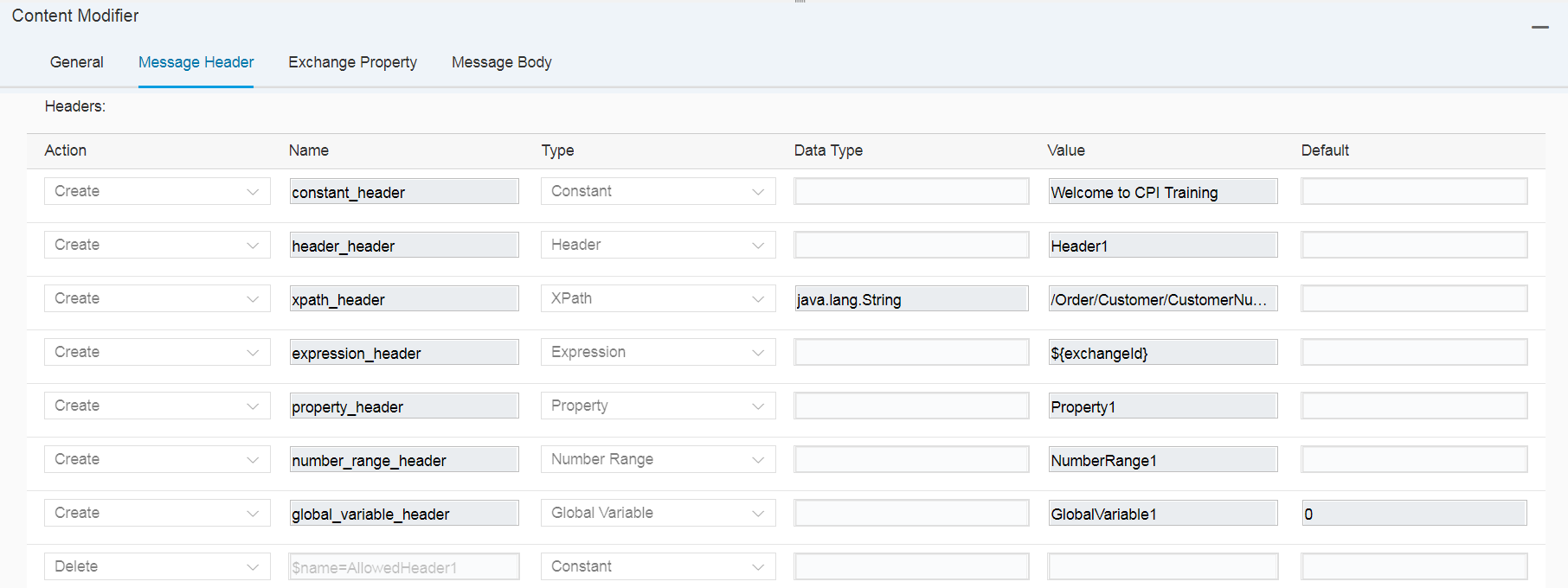
Type: ***Constant***

Body:

***<Order>  
<Customer>  
<CustomerNumber>12345</CustomerNumber>  
</Customer>  
</Order>***

**Content Modifier 2:**

**Message Header:**



**Types of Headers:**

# **Action: Create**

## **3.1 Constant Type Header:**

You can write a constant value to the header by maintaining the value of the constant in the ‘value’ tab.



Name: ***constant\_header***

Value: ***Welcome to CPI Training***

## **3.2 Header Type Header:**

This is used when you have an existing header which you need to add in the header of another message.

For e.g.: The header set in first content modifier (***Header1***) is utilized in the header of the message in the second content modifier as a Header type Header and value as the name of the header of the first content modifier.



Name: ***header\_header***

Value: ***Header1***

## **3.3 XPath Type Header:**

When you need to retrieve data from the incoming message using XML Path Language (XPath).

For example, we have an XML message in the Message Body Tab of the first content modifier. Suppose we need to extract the data from the field ***CustomerNumber*** in the header level of the message, we can use the XPath of the xml message and create the XPath Type header in the second modifier so that we can extract the value from the particular field in the xml to the header of the incoming message. Here:

***/Order/Customer/CustomerNumber*** (from content modifier 1 message body xml)



Name: ***xpath\_header***

Value: ***/Order/Customer/CustomerNumber***

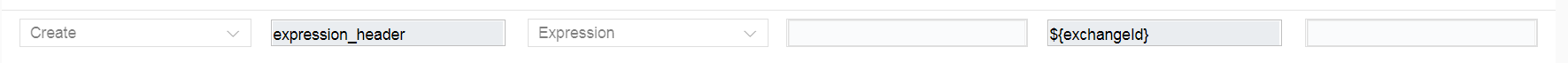
## **3.4 Expression Type Header:**

Allows you to enter a Camel simple expression

For example, you can use the expression ***${exchangeId}*** to add the exchange ID (a unique identifier of the message exchange) in the incoming message dynamically.

There are several other syntaxes used like for dynamic ***filename, message body*** (To get the body of the in message), etc.

Please refer: <http://camel.apache.org/simple.html>



Name: ***expression\_header***

Value: ***${exchangeId}***

## **3.5 Property Type Header:**

We use this type of header when we need to utilize an existing exchange property in the header level of another message. For e.g. - We have exchange property (***Property1***) in the first content modifier. The same objects created in the message header of content modifier can be mentioned in the exchange property of that very content modifier but the objects in the exchange property do not move along with the message to the target system. So, to use that in the header level of the message we can create a property type header in the message header of the message.



Name: ***property\_ header***

Value: ***Property1***

## **3.6 Number-Range Type Header:**

In operations tab, we can create/manage our number ranges. We need this type of header when we want to dynamically assign numbering of messages (e.g. - IDOCs) at the header level.

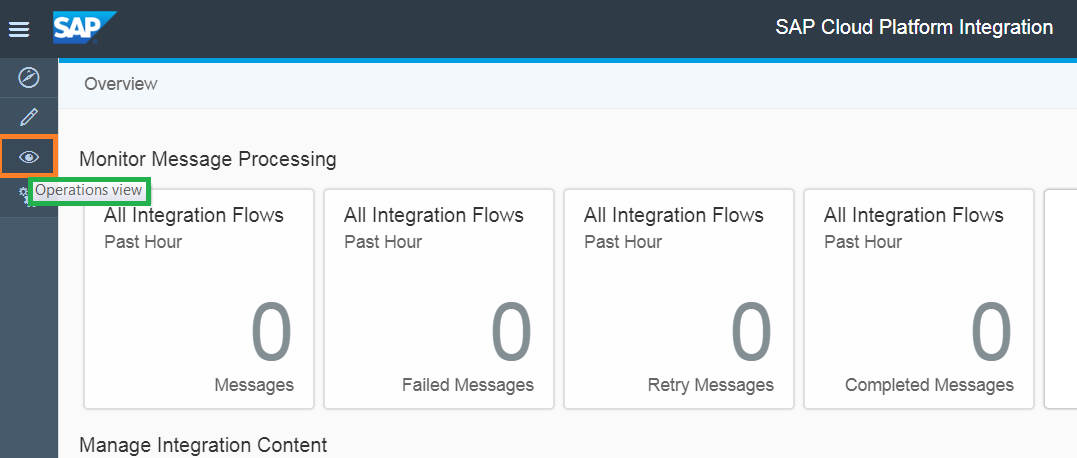


Name: ***number\_range\_header***

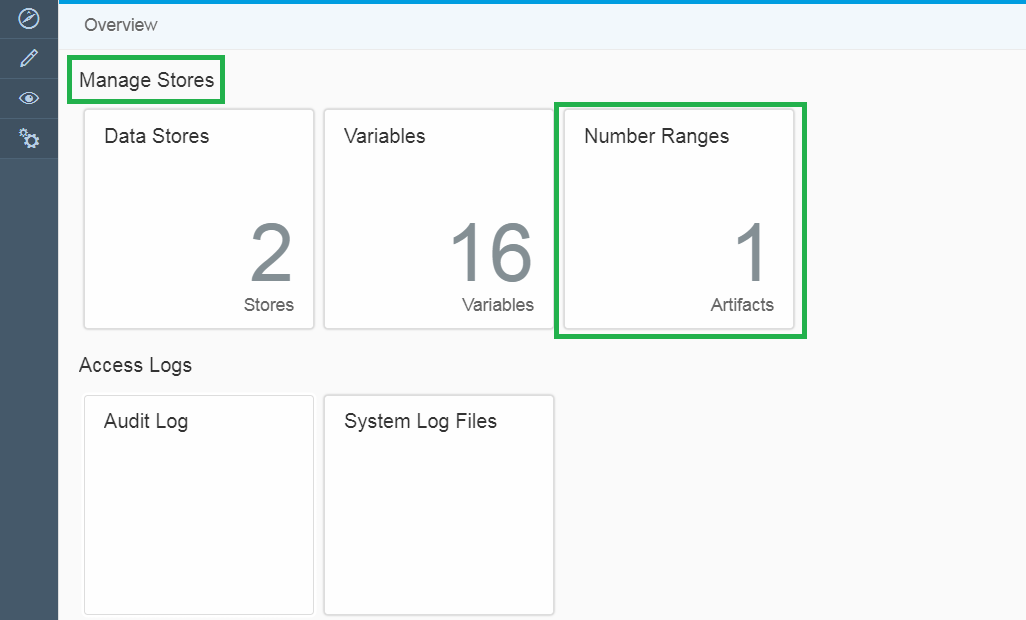
Value: ***NumberRange1***

**Creation of Number Range**

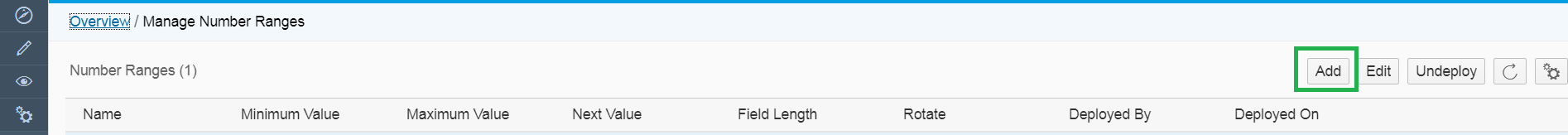
* Go to the Operations View



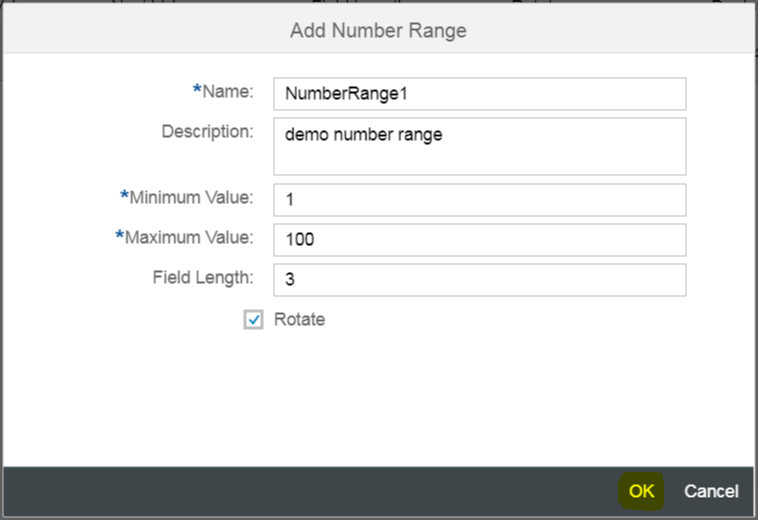
* Scroll down to get to the Number Ranges box.



* Click to ***‘Add’*** to add a number range.



* Add ***‘Name’***, ***‘Description’***, ***‘Minimum Value’***, ***‘Maximum Value’*** and ***‘Field Length’***. Click on ***‘OK’***.



Name= ***NumberRange1***

Description= ***demo number range*** (say)

Minimum Value= ***1*** (say)

Maximum Value= ***100*** (say)

Field Length= ***3*** (maximum up to two digits (99) beyond that it truncates the value)

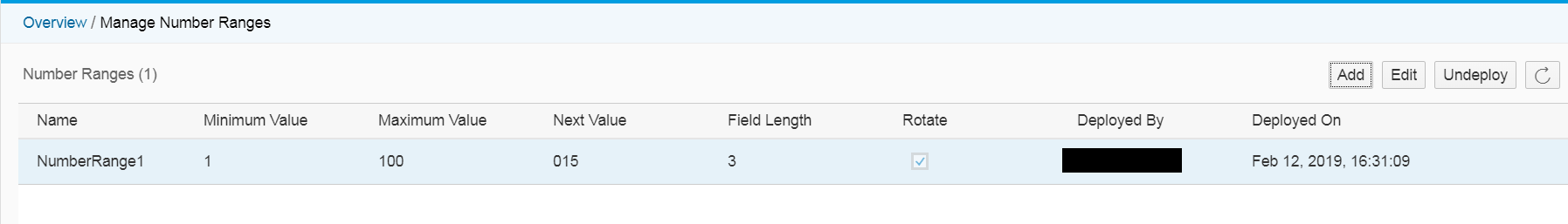
Rotate checkbox functionality:

Checking the rotate box ensures the counter will revert back to the minimum value once it touches the maximum value.

\*\*Note: If unchecked, the maximum value mentioned won’t have any impact on the ***‘Next Value’*** and keeps incrementing.

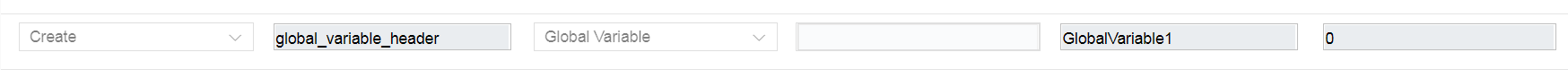
***‘Next Value’*** is nothing but a counter which gives you the count of the number of deployments or the number of times the IFLOW is triggered\*\*

* The Number Range is created.



## **3.7 Global Type Header:**

Allows you to define a global variable and to write its value at runtime to the header or property data container. It can be used across different integration flows of the same tenant.



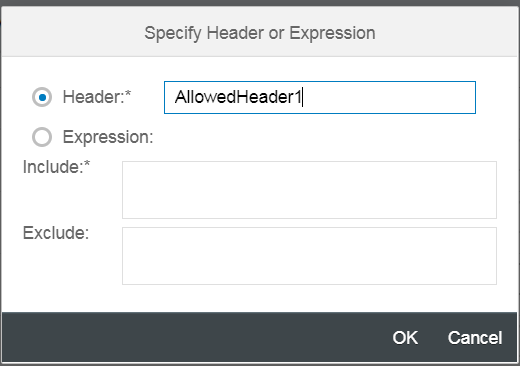
Name: ***global\_variable\_header***

Value: ***GlobalVariable1***

# **Action: Delete**

We use this when we want to omit the unwanted headers coming from the source which are not needed to be appended to the header of the message required by the Receiver.

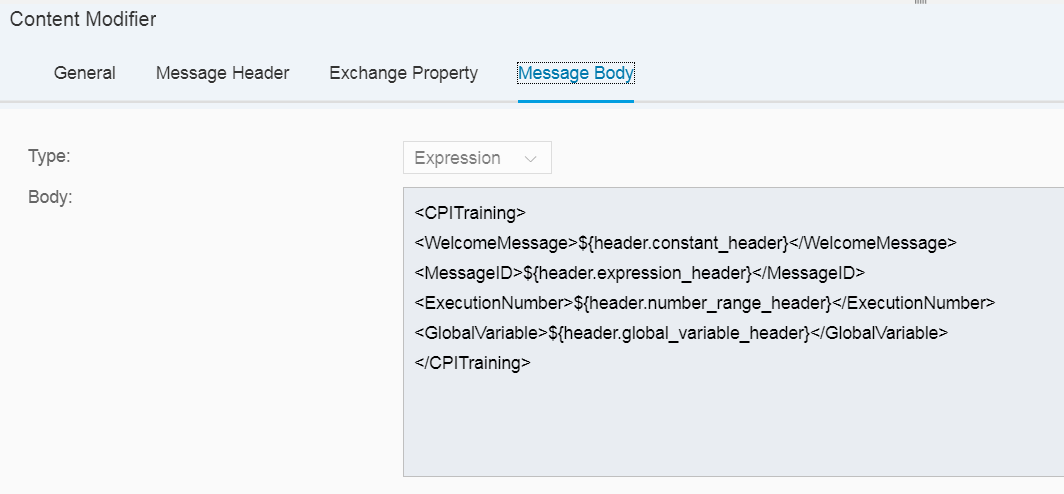




**Exchange Property:**

You can maintain same fields as mentioned in Message Header Property. But the difference is, the header constitutes the fields mentioned in Message Header.

**Message Body:**



Type: ***Expression***

Body:

***<CPITraining>  
<WelcomeMessage>${header.constant\_header}</WelcomeMessage>  
<MessageID>${header.expression\_header}</MessageID>  
<ExecutionNumber>${header.number\_range\_header}</ExecutionNumber>  
<GlobalVariable>${header.global\_variable\_header}</GlobalVariable>  
</CPITraining>***

**The Output would be like:**

***<CPITraining>  
<WelcomeMessage>***Header1 set in first content modifier***</WelcomeMessage>  
<MessageID>\*\*\*\*\*\*\*\*\*\*\*\*\*\*</MessageID> (dynamic exchangeID)  
<ExecutionNumber>***1***</ExecutionNumber> (if executed once)  
<GlobalVariable>***GlobalVariable1***</GlobalVariable>  
</CPITraining>***