MediaGateway 2 Data Mapping Examples

Bold Technologies
421 Windchime Place
Colorado Springs, CO 80919
719.593.2829

Table of Contents

Data Mapping

This document details the MediaGateway 2 process for Data Mapping.

Data Mapping consists of taking a signal that has arrived into Manitou through the UniversalConnector, breaking it apart according to a user-defined pattern, and then reassigning the fields to those associated with a Manitou alarm signal.

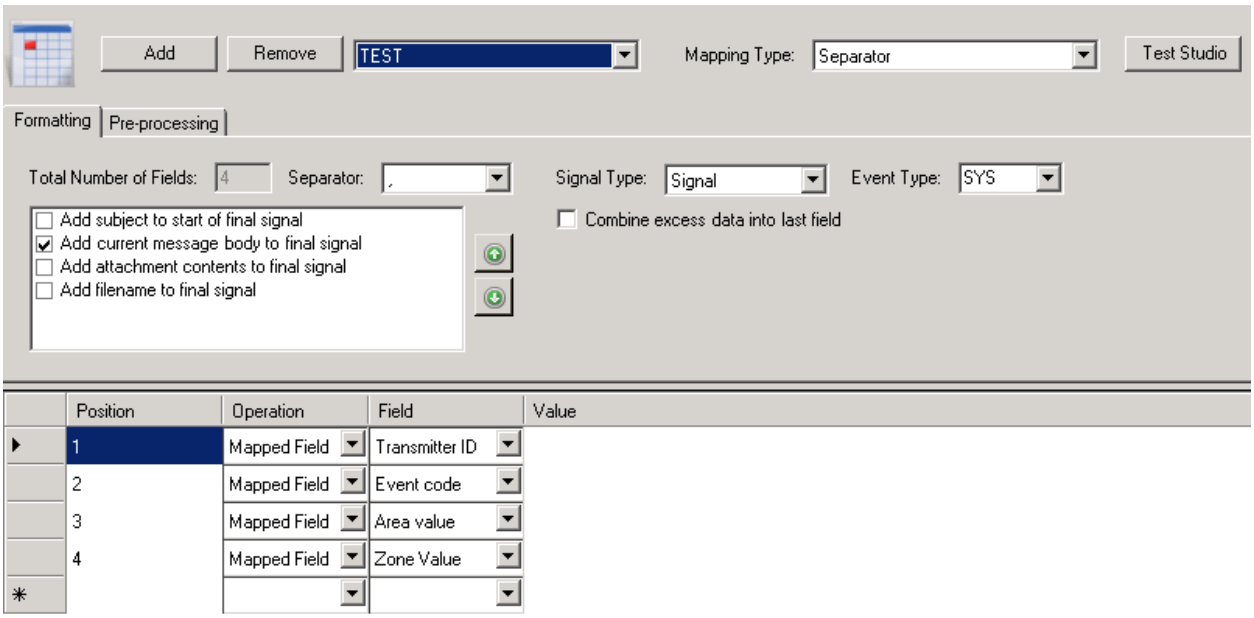
Mapping Types

The first decision you need to make when creating a Data Map for a signal is which Mapping Type you want to use. The following four Mapping Types are available:

1. **Separator -** this Mapping Type uses a single character to separate alarm data pieces contained in a signal. An example of a signal that uses the Separator Mapping Type is represented by the following signal:

   12345,BA,1,9

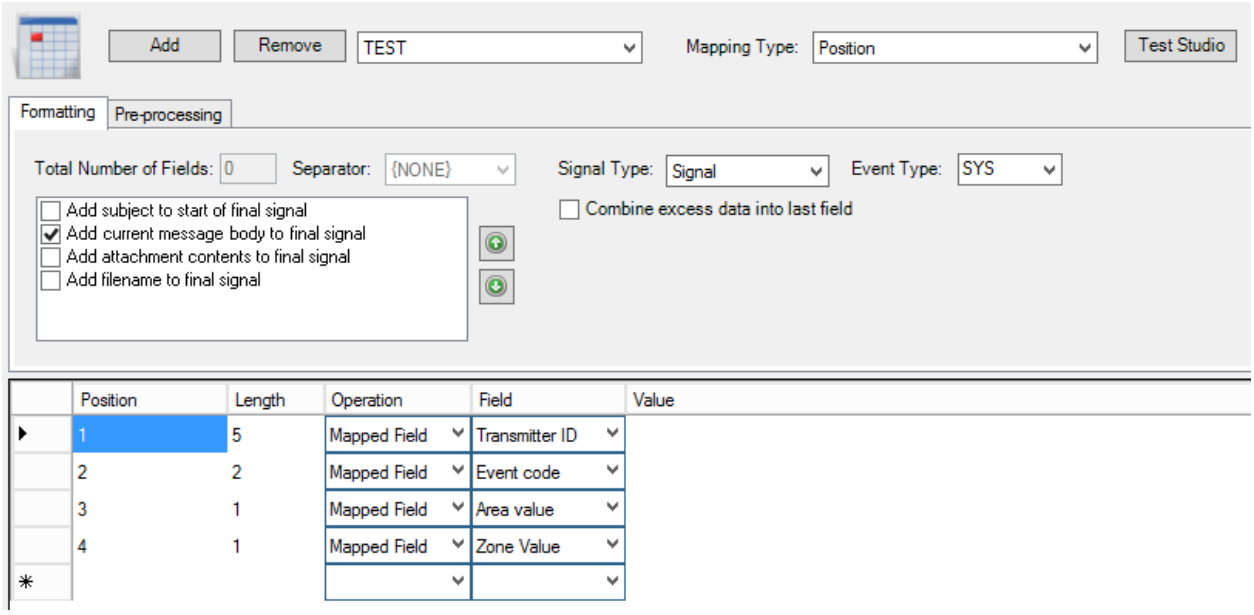
   The example signal contains a Transmitter ID, an Event Code, an area, and a zone all separated by a comma. The comma in the example signal is acting as the Separator.



2. **Position -** this Mapping Type uses the position and length of each piece of alarm data to break down a signal. An example of a Position Mapping Type signal is represented by the following:

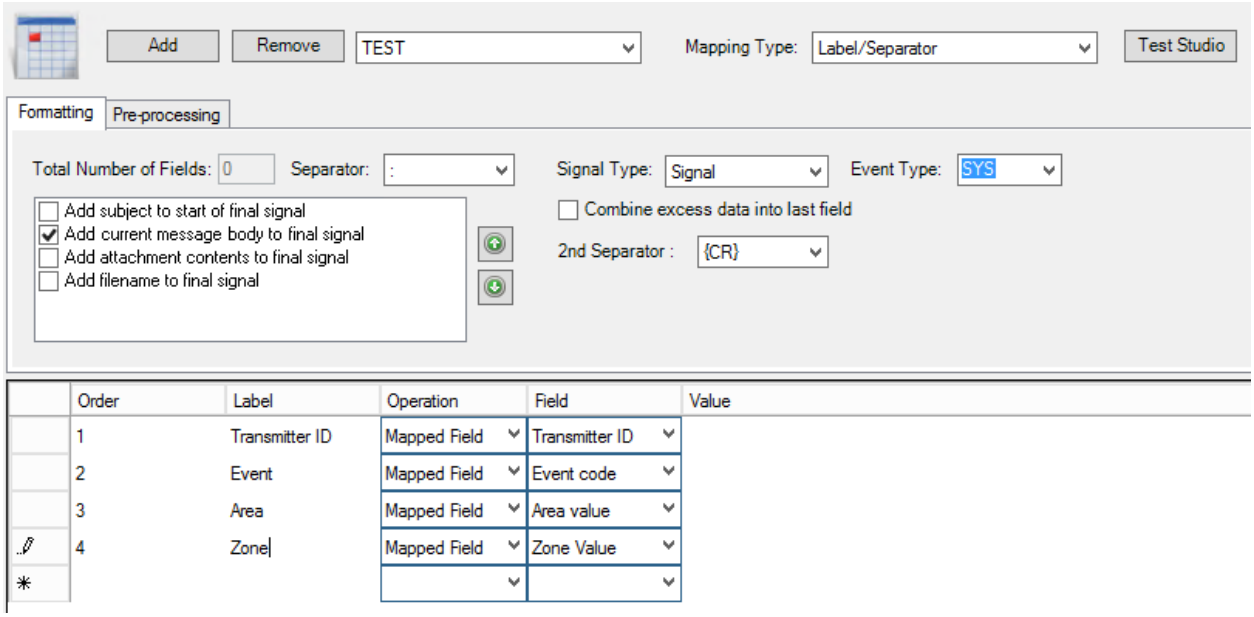
   12345BA19

   To break it apart, Manitou must know the following information: the Transmitter ID starts at character 1 and is 5 characters long, the Event Code starts at character 6 and is 2 characters long, the area starts at character 8 and is 1 character long, and the zone starts at character 9 and is one character long. The Data Map created for the example signal only works for other Position Mapping Type signals with the exact same fields and formatting. If the fields or their character lengths in the signal you receive vary at all, you must create another Data Map.



3. **Label/Separator** – this Mapping Type includes a field label usually followed by a Separator and a data value. The signal example below also contains a second Separator (a carriage return) between each alarm data piece. The following is an example of a Label/Separator type signal:

Transmitter ID: 12345
Event: BA
Area: 1
Zone: 9

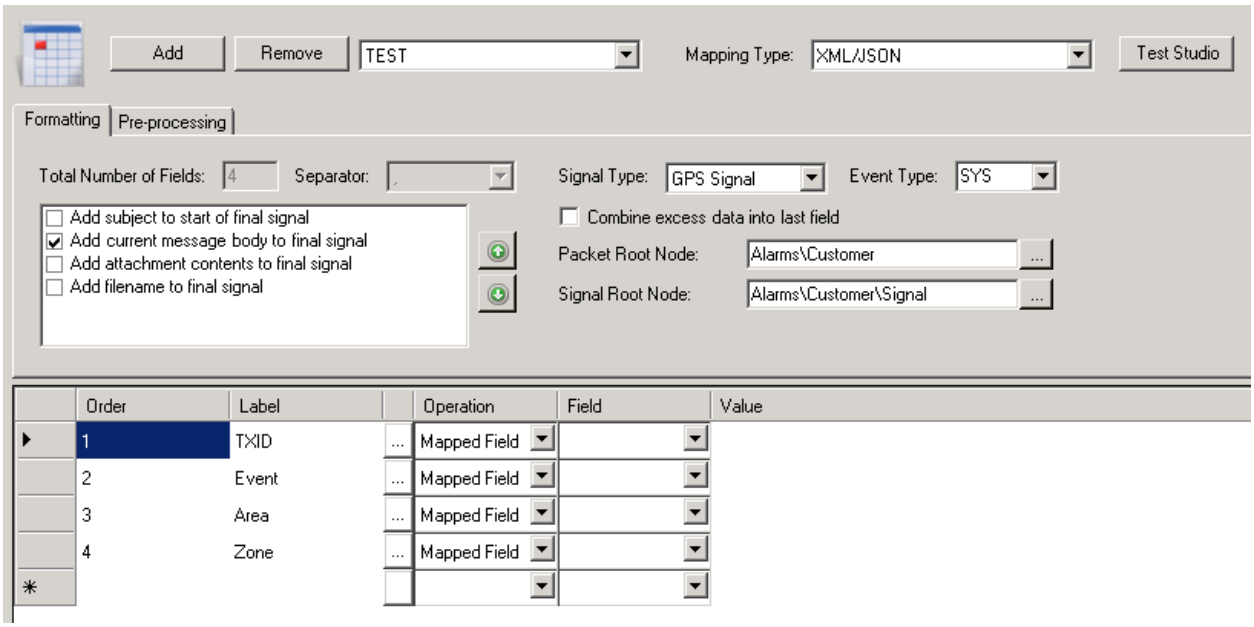


4. **XML/JSON -** this Mapping Type uses the standard XML (Extensible Markup Language) format. The XML/JSON Mapping Type is represented by the following signal example:

```
<?xml version="1.0"?>
<Alarms>
  <Customer TxId="12345" >
    <Signal Event="BA">
      <Area>1</Area>
<Zone>9</Zone>
    </Signal>
  </Customer>
 </Alarms>
```

From the XML/JSON signal, we can determine the following alarm data details:

TXID=12345
Event=BA
Area>1<
Zone>9<



Signal Types

Use the “Signal Type:” field to indicate whether the signal coming into Manitou is derived from a regular signal, GPS, or Telemetry. GPS signals differ from regular signals in that regular signals do not allow the user to designate GPS location information. Telemetry includes unit and value fields that are exclusive to it. The Telemetry signal type was developed for the purpose of reporting on measurement-based devices.

Event Types

The “Event Type:” field specifies to which Manitou Event Map format the signal belongs. Manitou uses the following three Event Type formats:

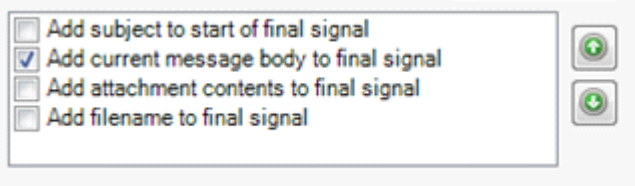
- SYS: Manitou standard system codes
- SIA: SIA standard codes
- CID: Contact ID standard codes

You can also enter your own code. In order for your code to be correctly decoded, however, you must enter it into the Manitou Supervisor Workstation Transmitter Protocol Types form and define it in Event Maps. Refer to the Supervisor Workstation User Guide for instructions.

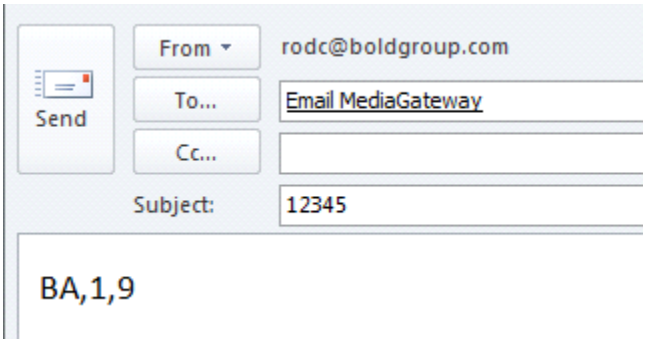
Signal Parts

Signals can have multiple parts. For example, an email can have a body, a subject, and attachments. Each part of that email might contain alarm data that can be decoded in the Data Mapping process.

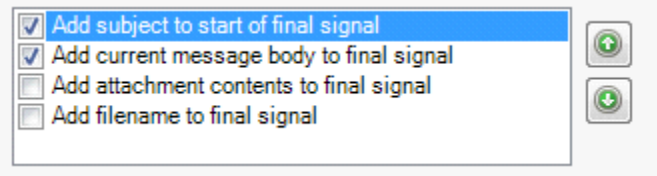
The signal parts box (displayed below) specifies which alarm data pieces should be processed, and in what order. So, if an email that is using the Separator Mapping Type is sent into Manitou, you must define which email parts are relevant as shown in the following screenshot:



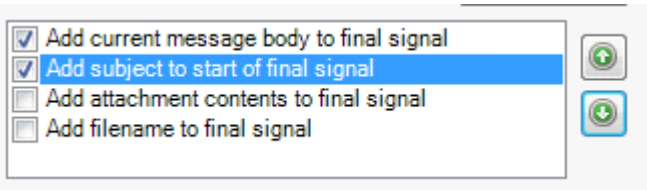
Note: you must select at least one of the checkboxes shown in the screenshot above.



In the email above, the Transmitter ID displays in the subject line, and the Event Code, area, and zone for the alarm display in the body. To process the signal, it is necessary to put it all together into the correct order. So, you must select the subject and body options, and then use the green arrows to move the lines up or down into the correct order.



This creates the following signal: 12345,BA,1,9

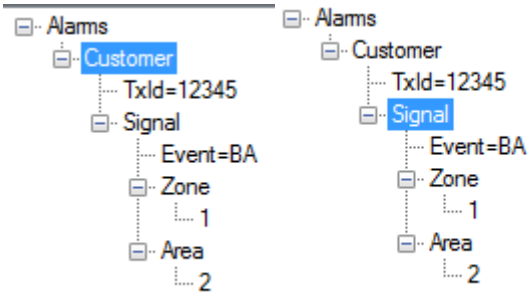


This modifies the existing signal as follows: BA,1,9,12345

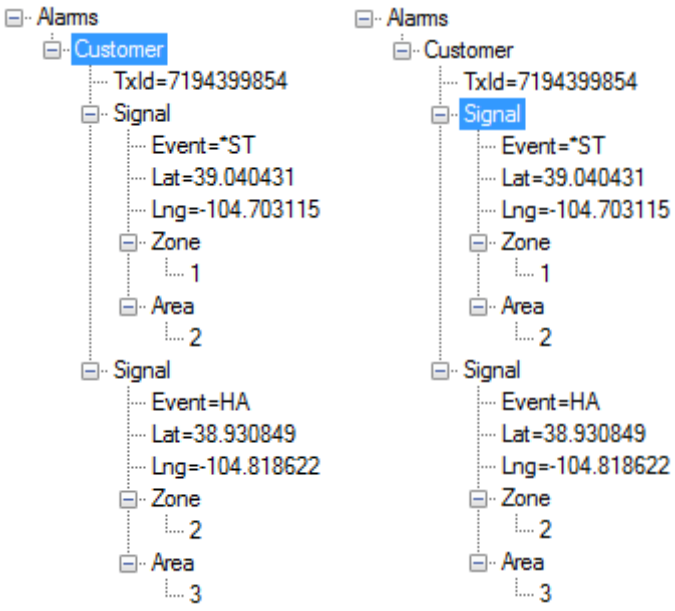
Use the "Combine excess data into last field" option (shown above) to lump all remaining data into the last field. This can be useful if the system is sending erroneous information (for e.g., signatures or unwanted data) at the end of the signal. This allows you to collect all the excess data into the last field, and ignore it.

If you are using the XML/JSON Mapping Type, the "XML Packet Root Node:" and "XML Signal Root Node:" fields enable you to specify the parts of the signal required for processing. The Packet Root Node identifies the start of the packet of information. Within this packet, there might be multiple signals. The Signal Root Node identifies where those signals are.

In the following example, the Packet Root Node and the Signal Root Node display as highlighted.

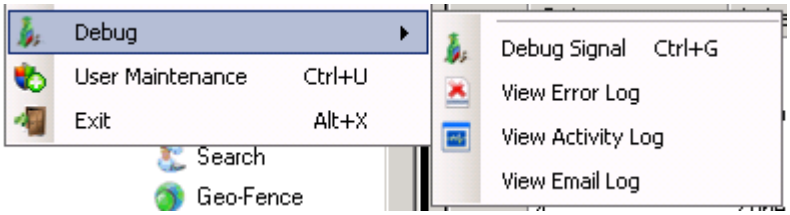


Depicted in the screenshot below is a more complex GPS signal with multiple parts. Again, the Packet Root Node and the Signal Root Node display as highlighted.

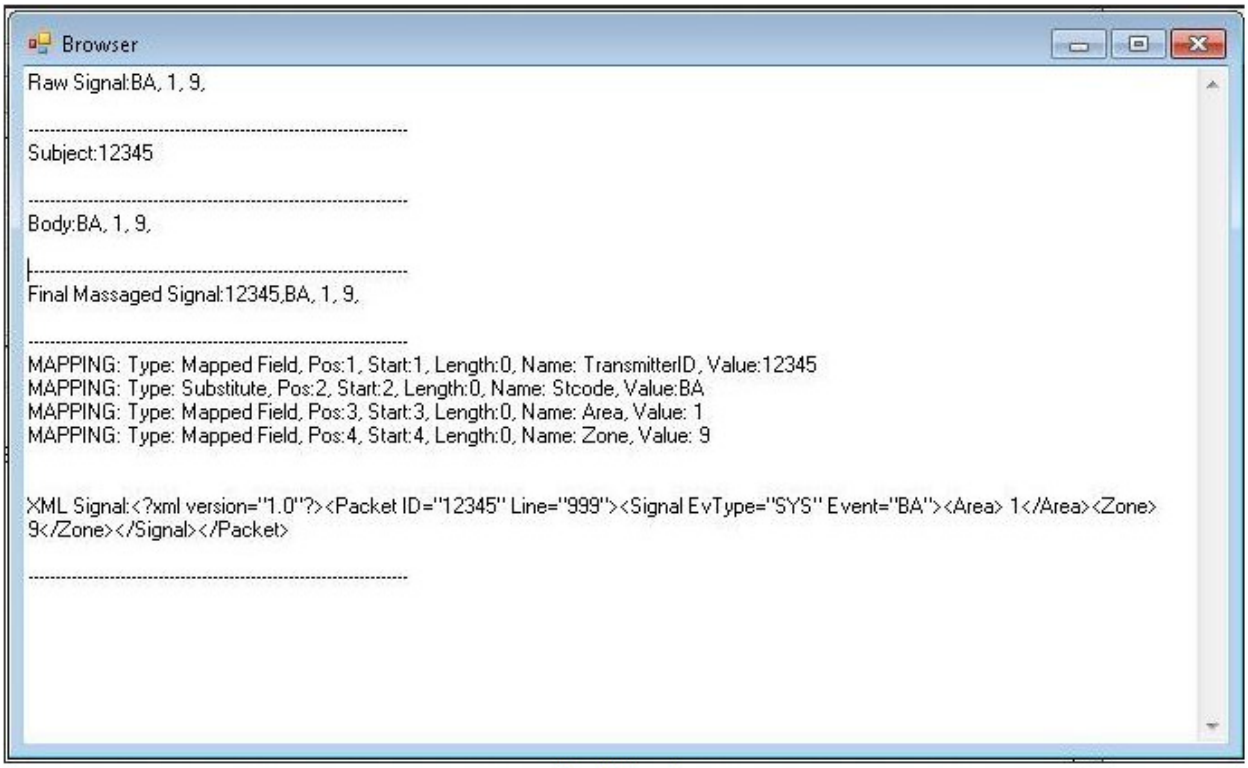


Debugging

Debugging is the mechanism used to trace what happens to a signal when it is processed. It can be very helpful in pinpointing errors in the Data Mapping setup. To display the last signal received into Manitou, press CTRL-G on your keyboard, or choose the following option from the File menu:



This results in your browser window displaying the signal as shown in the following screenshot. It displays the raw signal and the subject first. Then, it shows how they are combined together. Next, it shows how each field has been defined in the MediaGateway 2 for the Data Mapping process, and the value of the entry for each field. Finally, it shows the XML signal sent into Manitou. These pieces of information enable you to determine how the signal is transformed as it is processed through the UniversalConnector.



Signal Debugging

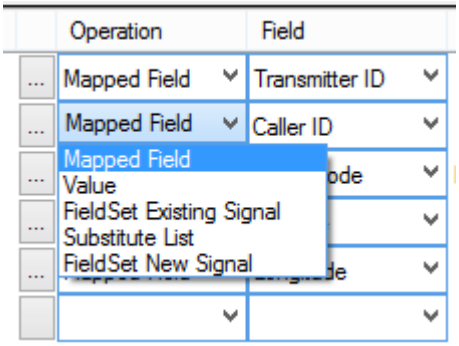
Mapping Fields

As previously discussed, field mapping varies depending on the Mapping Type you are using. In the example below, we are again mapping a simple comma separated signal.

	Position	Operation	Field	Value
▶	1	Mapped Field	Transmitter ID	
	2	Mapped Field	Event code	
	3	Mapped Field	Area value	
	4	Mapped Field	Zone Value	
*				

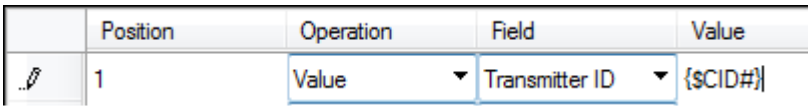
Fields in the “Position” column are numeric, and specify the order in which the fields should be processed.

You have the following five options for fields in the “Operation” column as shown in the following screenshot:

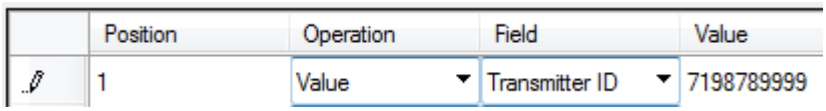


Mapped Field: select this option to accept the data contained in the signal.

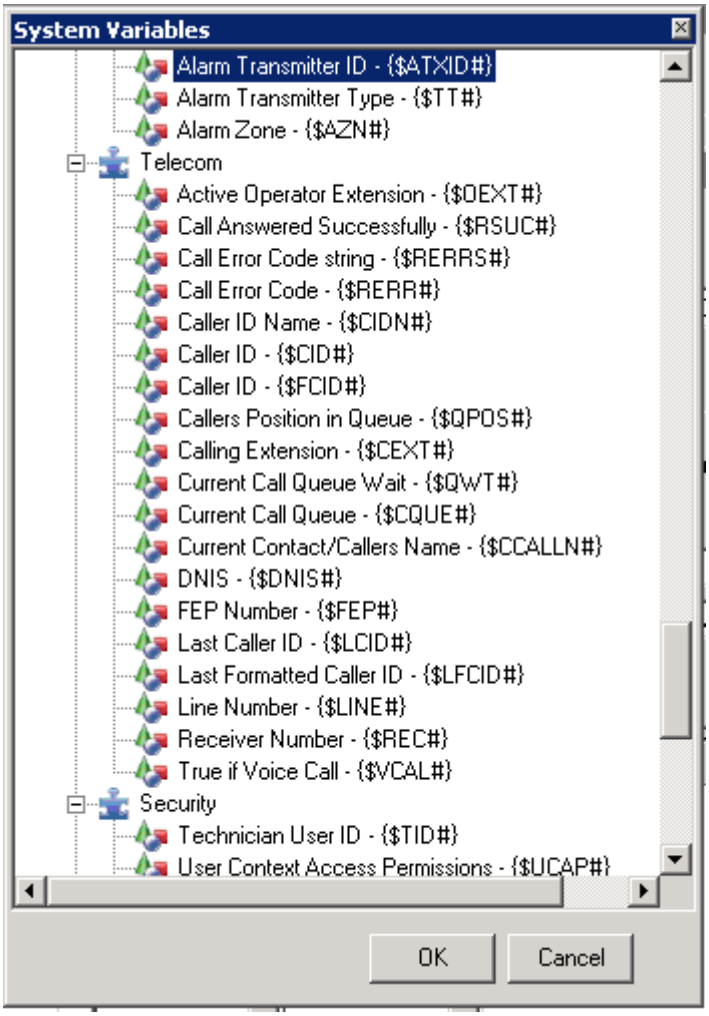
Value: select this option to enter a value for the data or to use a system variable. Then, enter your value into the "Value" column as shown in the screenshot below. For example, if the device sending the signal is a cellular device and you want to use the caller ID of the device as the Transmitter ID, you can do this using the following value operation:



You can also hard code a value for testing. See the example screenshot below:



{$CID#} is the caller ID for the device. You do not have to remember these variables. Left-click the values field so that the cursor displays as flashing. Then, right-click to display the “System Variables” window. Select the variable you want.



The value is also useful for manipulating fields. If you have a signal that sent an Event Code in two parts, and you wanted to combine them into one. You could do it as follows:

Signal - 12345,F,A,1,9

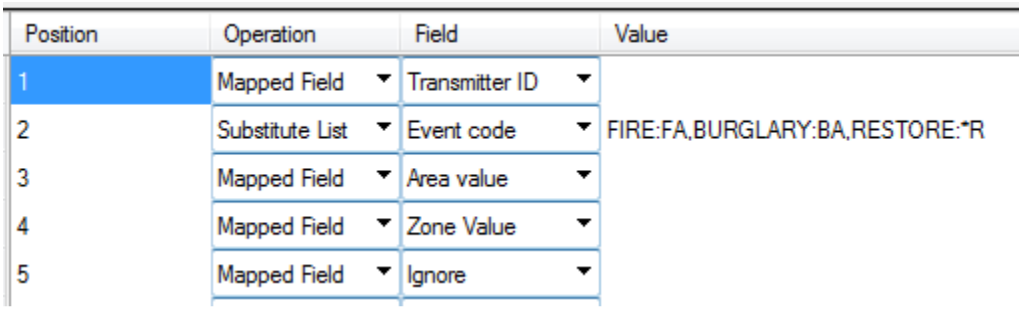
Position	Operation	Field	Value
1	Mapped Field	Transmitter ID	
2	Mapped Field	$Temp1	
3	Mapped Field	$Temp2	
4	Value	Event code	$Temp1$Temp2
5	Mapped Field	Area value	
6	Mapped Field	Zone Value	

Here, we are mapping the two parts of the Event Code into a temporary variable. Then using the value, we assign them to the Event Code. By placing them straight after each other in the value field, they are concatenated together. So, “F” and “A” becomes “FA”.

FieldSet Existing Signal: select this option to select the fieldset you want to use.

Substitute List: select this option to change the characters necessary to define the values. Take our current example, but assume that the Event Code was more verbose. It might send Event Codes of FIRE, BURGLARY, RESTORE etc. The Substitute List allows us to modify the Events into a shorter form.

So, we might want FIRE -> FA, BURGLARY -> BA, RESTORE -> *R**.**

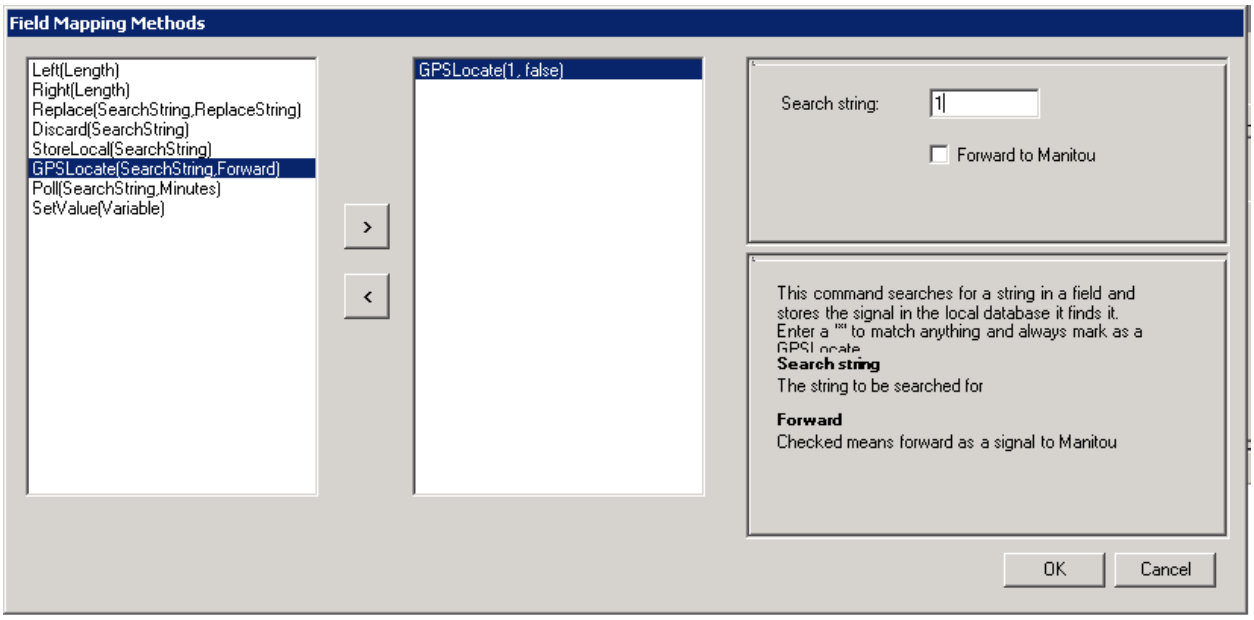


To achieve this, we create a Substitute List as shown above. Use a comma to separate the items, and a colon to link the substitutions.

FieldSet New Signal: select this option to create multiple signals each with its own fieldset.

Field Mapping Methods

Double-click in the "Value" cell to display the "Field Mapping Methods" dialog displayed in the following screenshot:



Left(Length) – this method takes the character length of a field and makes it the current value. So, if a field contains the value “FIRE” or “BURGLARY”, the Left(1) method modifies the method to an “F” or “B”.

Right(Length) – this method works like the Left(Length) method, but takes from the right side of the field.

Replace(SearchString, ReplaceString) – this method replaces one string with another. "Replace(FIRE,FA)" replaces the string “FIRE” with “FA”. It reports partial matches, so “FIREALARM” does not get changed to “FAALARM”.

Note: if you are performing a “Replace” action with “GPSLocate”, you must enter the original value instead of the replacement value into the GPSLocate value fields. For example: if want to replace 4 with GS4, you must enter the following text into the Replace value field: “Replace(4,GS4)”. This input directs the MediaGateway 2 to search for the number “4” and to replace every instance it finds with “GS4”. Now, if you also want to include a GPSLocate value for the same Label, you must enter “Replace(4,GS4);GPSLocate(4,false)”, rather than Replace(4,GS4);GPSLocate(GS4,false)”.

Discard(SearchString) – this method results in the signal being discarded if the field matches the search string.

StoreLocal(SearchString) – this method results in the signal being stored in the MG_SIGNALS table if the field matches the search string.

GPSLocate(SearchString, Forward) – this method results in the signal being stored in the MG_SIGNALS table if the field matches the search string as a GPSLocate signal. GPS signal processing (for e.g., checking against Geo-fences) occurs. If "Forward to Manitou" is checked, the signal forwards to Manitou.

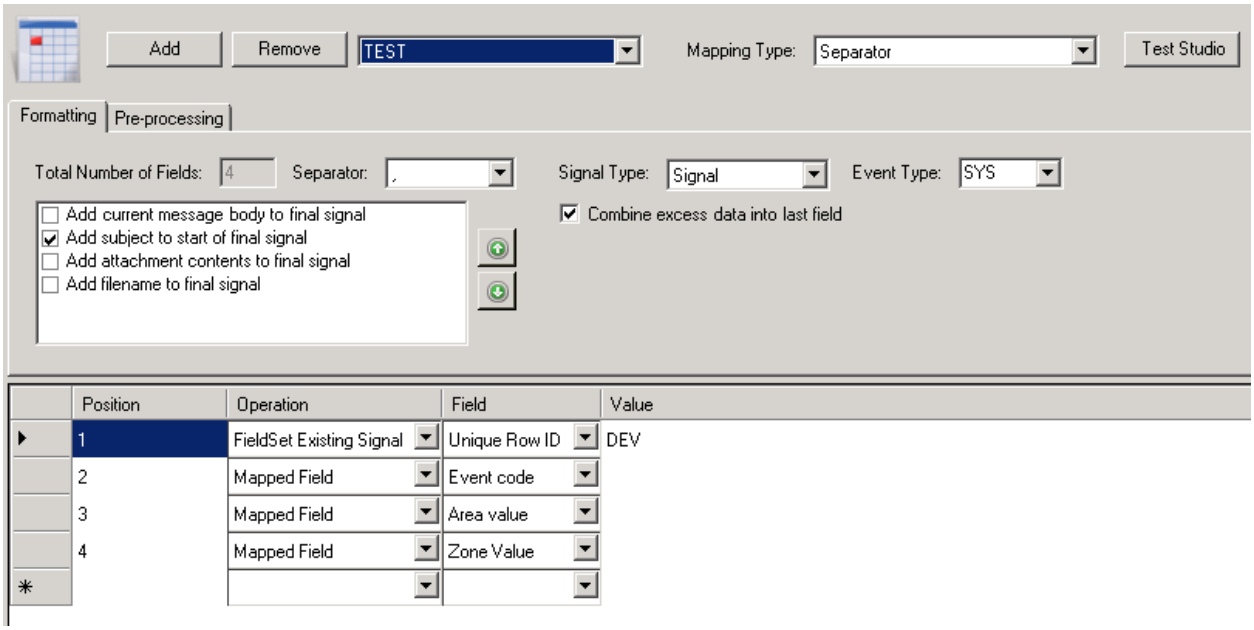
Poll(SearchString, Minutes) - if the field matches the search string, this method results in the signal being counted as a polling signal for the device sending it. If the device does not send

another polling signal within the number of minutes you define, an alarm triggers in Manitou for the device. A device must send at least one polling signal in order to register. If the device fails to poll, only one alarm will be sent to Manitou. The device will have to signal again to restart the process.

SetValue(Variable) – this method assigns the value in the field to the named variable.

The Formatting Tab

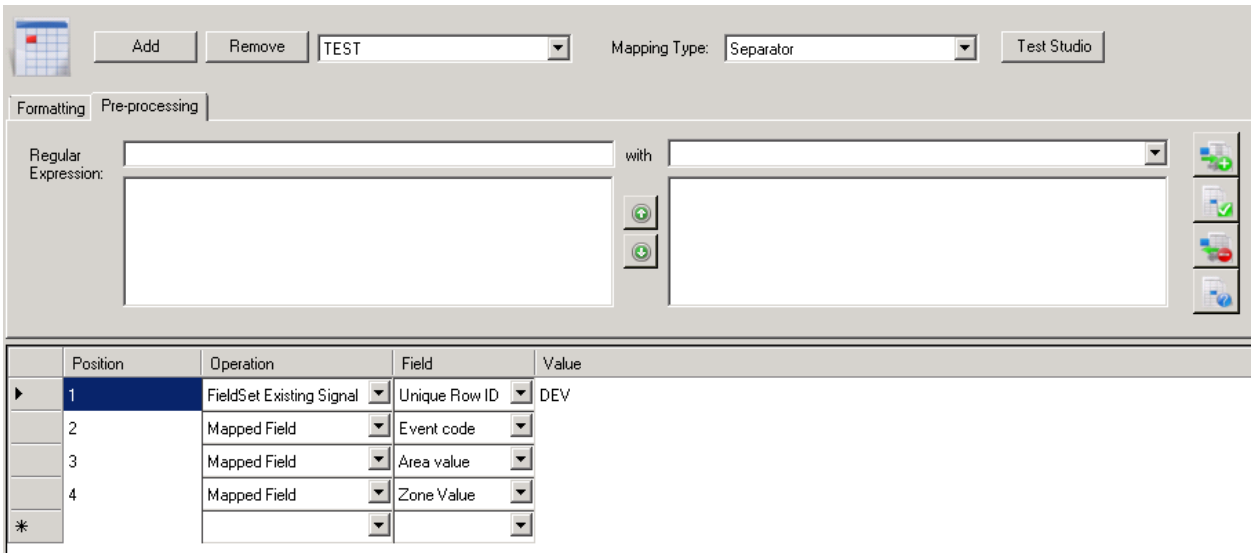
Up to this point, we have dealt exclusively with the Formatting Tab on the MediaGateway 2 Data Mapping form displayed in the following screenshot:



The Formatting Tab allows you to create, import, and format a Data Map.

The Pre-Processing Tab

The Data Mapping form also includes the “Pre-processing” Tab shown in the following screenshot:



Use the Pre-processing Tab to define alternative forms in which a signal could arrive. For example, if you designated a comma as the Separator for a signal on the Formatting Tab, but you know that the signal might also arrive using a semi-colon, you can prepare Manitou for the arrival of the alternate signal by using the Pre-processing Tab.

As another example, let's say you have a signal you want to process in a comma separated format, but each piece of data is attached to an information label, such as:

TXID:12345,Event:Burglary,Area:2,Zone:5

Performing a comma separation results in the following four fields:

TXID:12345
Event:Burglary
Area:2
Zone:5

As you can see from the example, using only comma separation failed to adequately isolate the alarm data. The superfluous labels are still connected to the necessary alarm data pieces. To work around this situation, use the Pre-processing Tab to change the colons to comma separated items. This procedure results in eight fields, instead of the original four. You can now use Data Mapping to ignore the labels. The resulting signal displays as follows:

TXID,12345,Event,Burglary,Area,2,Zone,5

Refer to the Email Pre-Processing Example in this document for more information on pre-processing.

Examples of Data Mapping

Simple Email Example

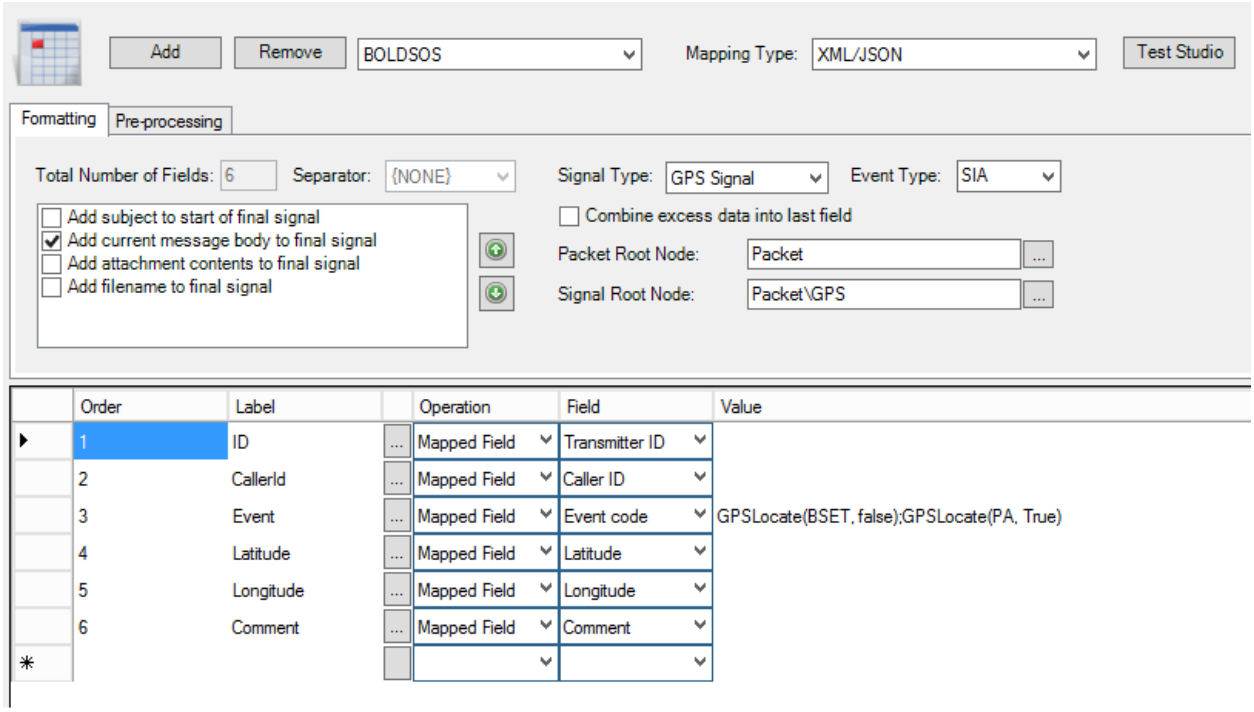
Data Mapping is usually required for email signals. This is because they often arrive with alarm data in the subject, body, and attachment. Even when alarm data is contained in a single area of the email, however, the signal still likely requires parsing. Data Mapping allows you to designate which parts of the email signal correspond to which Manitou fields.

The following screenshot shows an email with a relatively simple format:

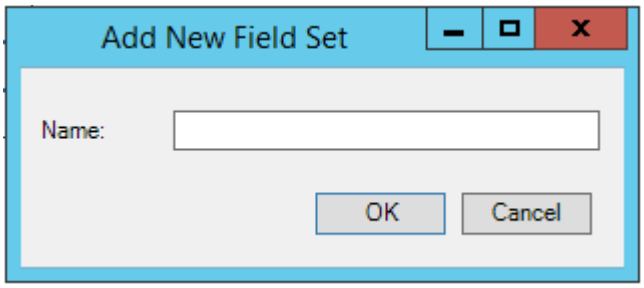
There is no alarm data in the subject of this email, and there is no attachment. All the usable alarm data is contained in the email body only. Each piece of data is separated by a hyphen. Data Mapping is still necessary, however, because Manitou does not know the fields to which each data piece belongs.

Perform the following steps to create a Data Map for the simple email example:

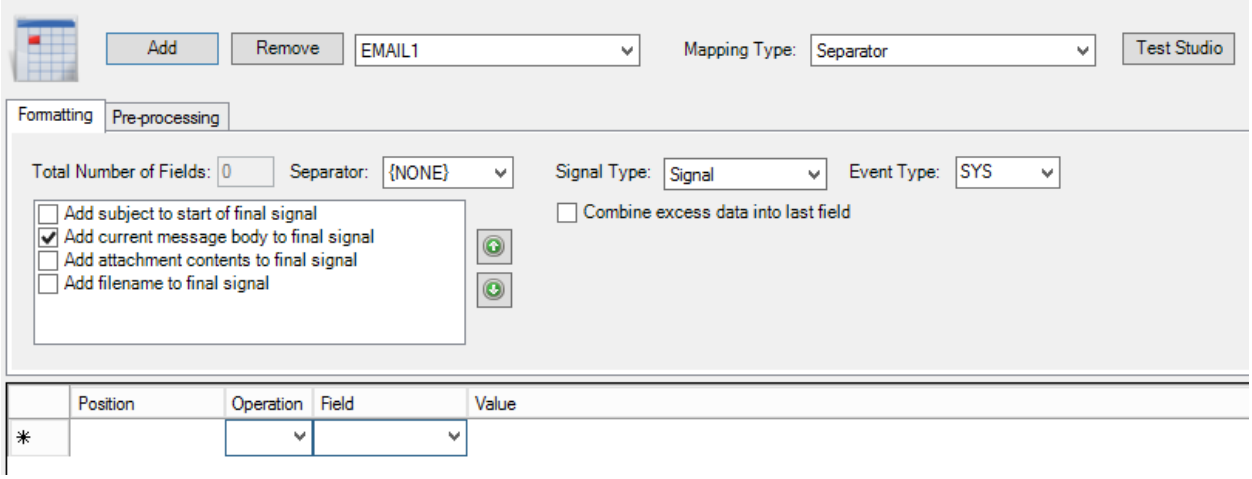
1. If you have not yet set up an Email Connector, do that first. Refer to the instructions in the MediaGateway 2 User's Guide, if necessary.
2. Open the Data Mapping section of the MediaGateway 2.
   Result: the Data Mapping form displays as shown in the following screenshot:



3. Click "**Add**".
   Result: the "Add New Field Set" dialog displays as shown in the following screenshot:

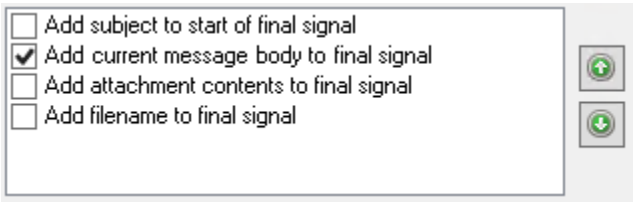


4. Enter a name, and then click "**OK**".
   Result: the "Add New Field Set" window closes and the system returns you to the Data Mapping form as shown in the following screenshot:



Note: because you may eventually create multiple Data Mapping field sets for email signals, we recommend using a name that allows you to differentiate your field set from others.

5. If not already selected, select "Separator" from the "Mapping Type:" dropdown menu. We are selecting this option because the alarm data pieces in the email is separated with a hyphen.
6. Select the hyphen symbol from the "Separator:" dropdown menu. We are selecting this option because the alarm data pieces in the email arrived separated with a hyphen.
7. If not already selected, select "Signal" from the "Signal Type:" dropdown menu. We are selecting this option because nothing in the email indicates the latitude and longitude coordinates necessary for a GPS signal, nor does the email indicate the unit and value fields associated with measurement-based devices.
8. If not already selected, select "SYS" from the "Event Type:" dropdown menu. We are selecting this option because the signal includes a BA. This is the Manitou system Event Code for a Burglary Alarm. For more information on Event Map options, refer to the Supervisor Workstation User's Manual, if necessary.
9. Because the email we received only has alarm data in the body, we need only designate to Manitou how to handle that one part. Our instructions to Manitou about how to process the different email parts of the signal displays like this:

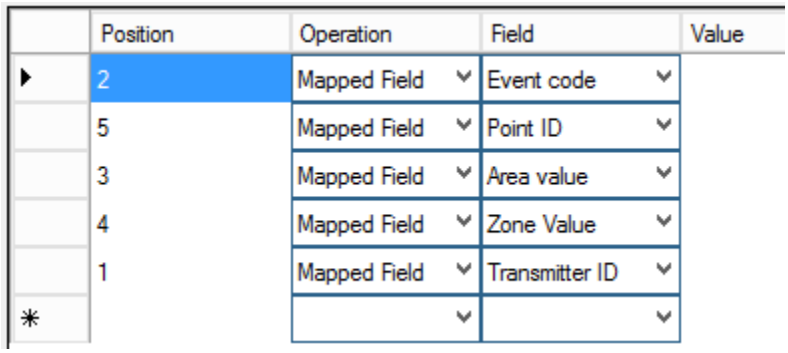


Next, we have to consider what each piece of alarm data means to Manitou and the order in which each piece will arrive. Here is our simple email example again:

1234-BA-1-1-Front Door

The alarm data includes a Transmitter ID, an Event Code, an area, a zone, and a Point ID. Therefore, we need to map those five fields. Also, we need to indicate the order in which the data fields will arrive.

10. To do that, enter the position hierarchy you want into the “Position” column as shown in the following screenshot:

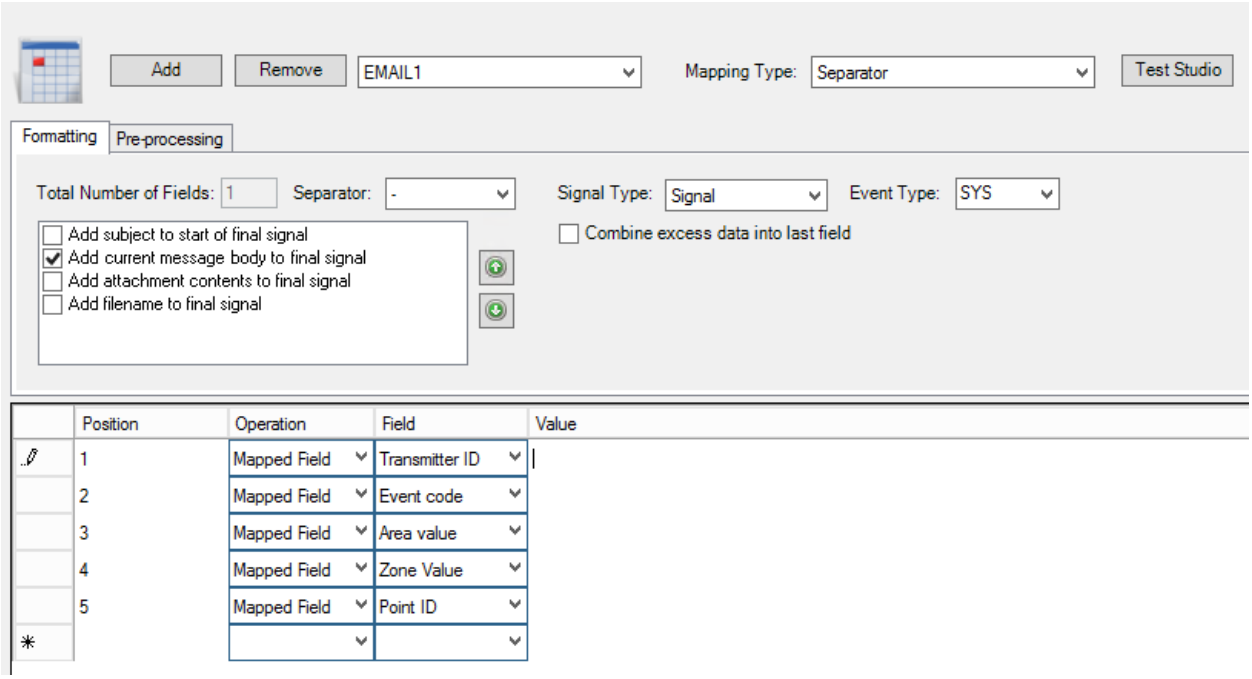


	Position	Operation	Field	Value
▶	2	Mapped Field	Event code	
	5	Mapped Field	Point ID	
	3	Mapped Field	Area value	
	4	Mapped Field	Zone Value	
	1	Mapped Field	Transmitter ID	
*				

11. Click “**Save**”.

These position settings indicate to Manitou that the Transmitter ID will arrive first, the Event Code will arrive second, etc.

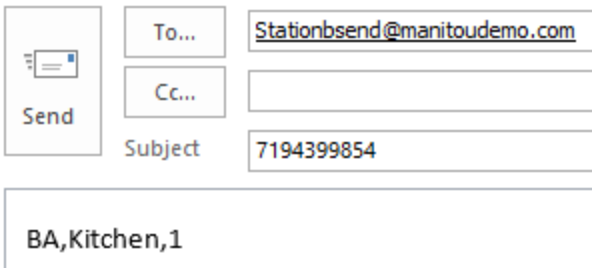
Our completed Data Map for our simple email example displays as shown in the following screenshot:



	Position	Operation	Field	Value
	1	Mapped Field	Transmitter ID	
	2	Mapped Field	Event code	
	3	Mapped Field	Area value	
	4	Mapped Field	Zone Value	
	5	Mapped Field	Point ID	
*				

Email Example with Signal Part in the Subject

The following example involves a signal received via the following email message:

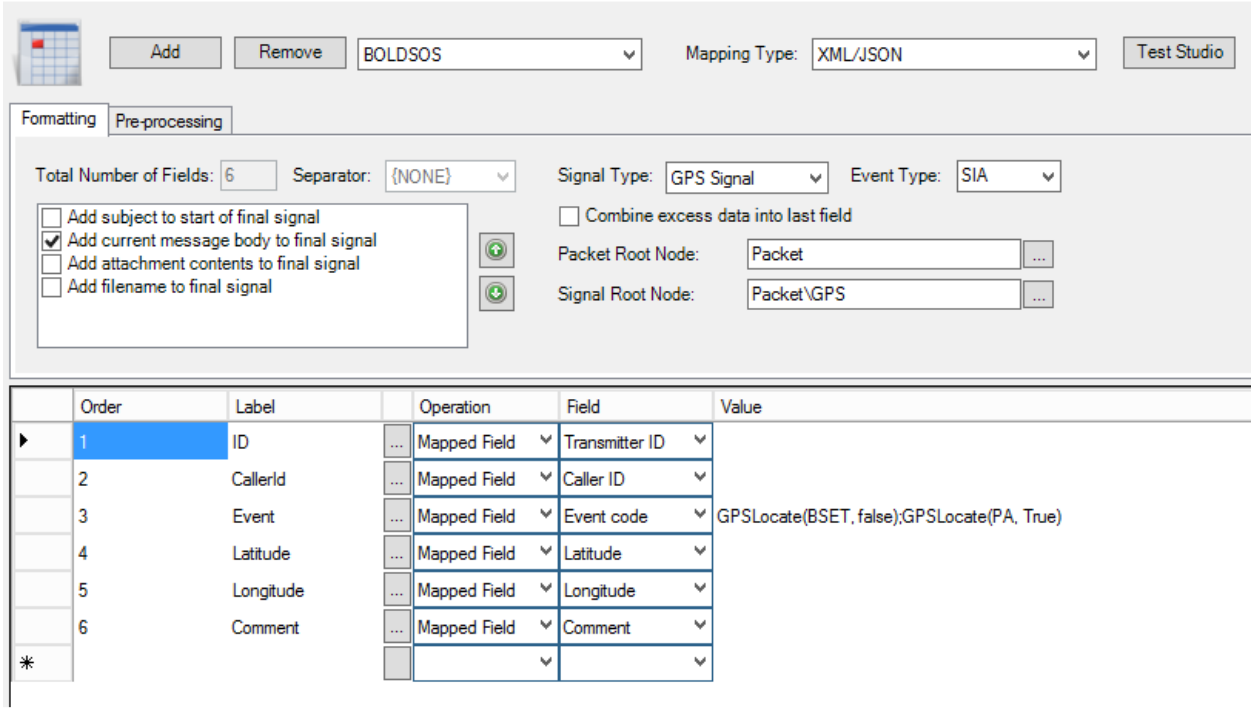


As you can see, this email includes alarm data in both the subject line as well as the body of the email.

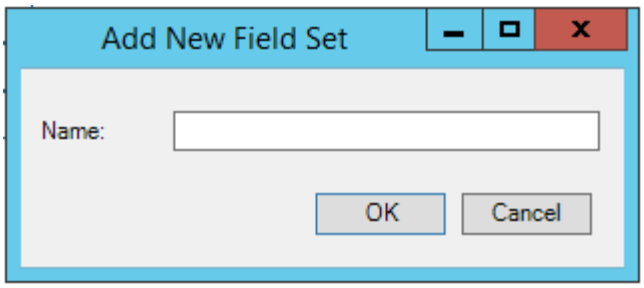
Subject: 7194399854
Body: BA,Kitchen,1

Perform the following steps to create the Data Map:

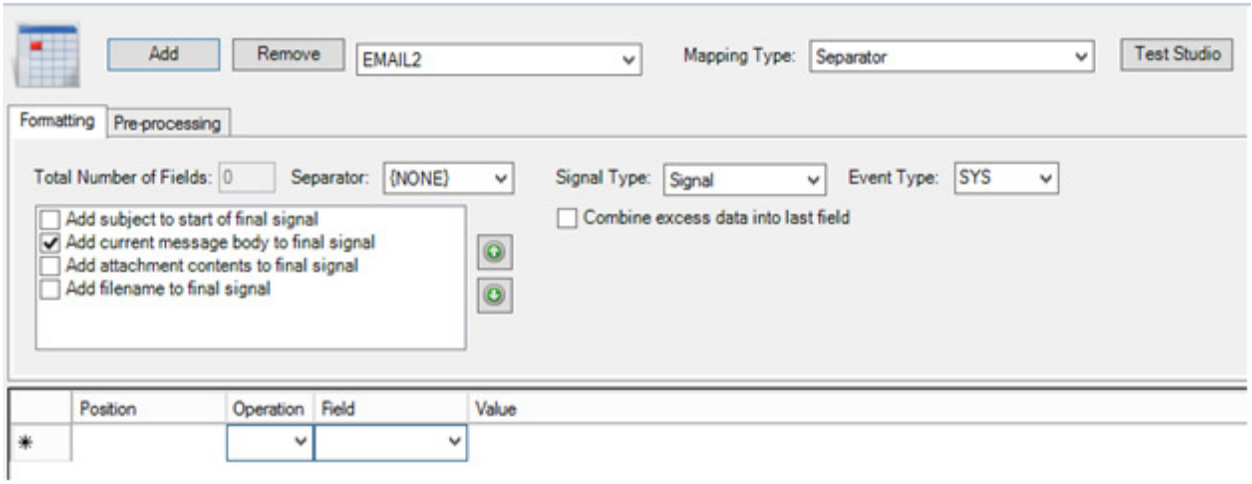
1. If you have not yet set up an Email Connector, do that first. Refer to the instructions in the MediaGateway 2 User's Guide, if necessary.
2. Open the Data Mapping section of the MediaGateway 2.
   Result: the Data Mapping form displays as shown in the following screenshot:



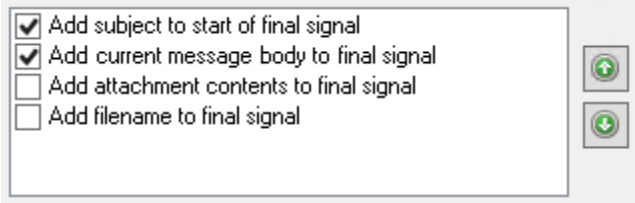
3. Click "**Add**".
   Result: the "Add New Field Set" dialog displays as shown in the following screenshot:



4. Enter a name, and click "**OK**".
   Result: the "Add New Field Set" window closes and the system returns you to the Data Mapping form as shown in the following screenshot:



   Note: because you may eventually create multiple Data Mapping field sets for email signals, we recommend using a name that allows you to differentiate your field set from others.
5. If not already selected, select "Separator" from the "Mapping Type:" dropdown menu. We are selecting this option because the alarm data in our email body is separated with a comma.
6. Select the comma symbol from the "Separator:" dropdown menu.
7. If not already selected, select "Signal" from the "Signal Type:" dropdown menu. We are selecting this option because nothing in the email indicates the latitude and longitude coordinates necessary for a GPS signal, nor does the email indicate the unit and value fields associated with measurement-based devices.
8. If not already selected, select "SYS" from the "Event Type:" dropdown menu. We are selecting this option because the signal includes a BA. This is the Manitou system Event Code for a Burglary Alarm. For more information on Event Map options, refer to the Supervisor Workstation User's Manual.
9. Because the email we received includes alarm data pieces in both the subject and the body, we need to designate to Manitou how to process both parts. Our instructions to Manitou about how to process the different email parts of the signal displays like this:

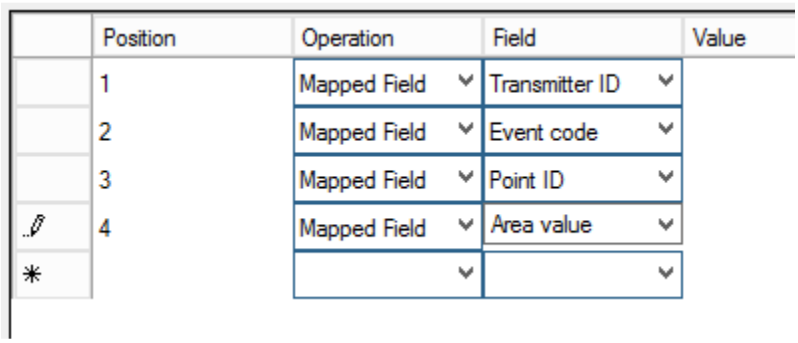


Next, we have to consider what each piece of the email alarm data means to Manitou and the order in which Manitou should parse it. Here is our email example again:

Subject: 7194399854
Body: BA,Kitchen,1

The alarm data includes a Transmitter ID, an Event Code, a Point ID, and an area. Therefore, we need to map those four fields. Also, we need to indicate the order in which the data fields will arrive.

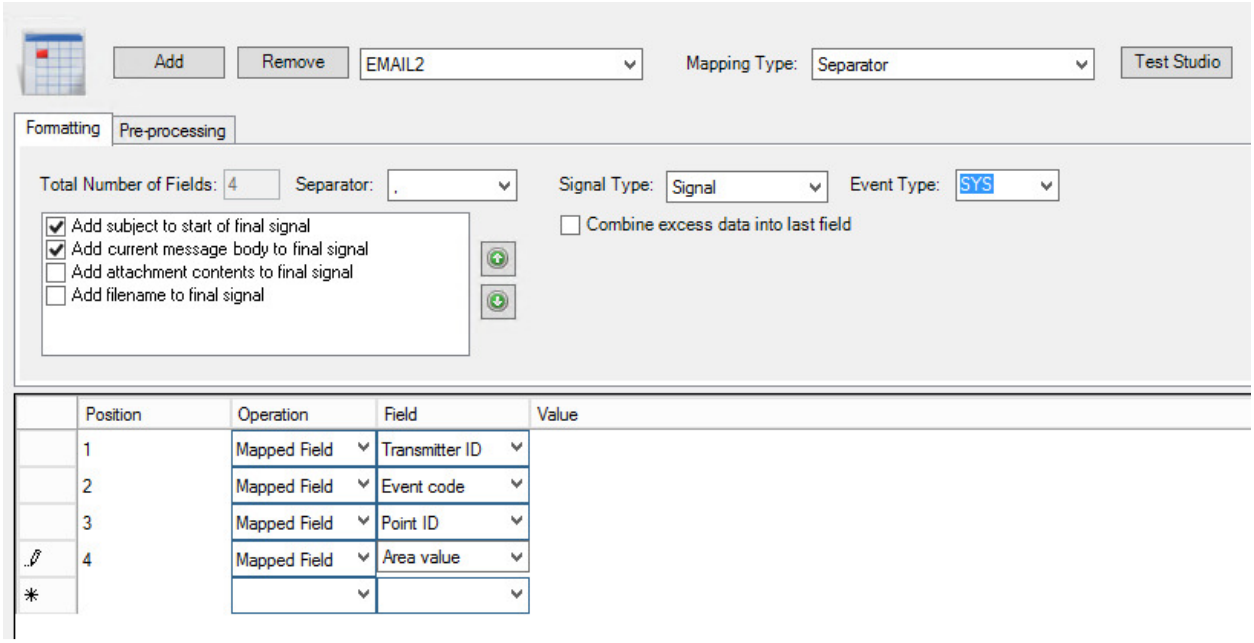
10. To do that, enter the position hierarchy you want into the "Position" column as shown in the following screenshot:



11. Click "**Save**".

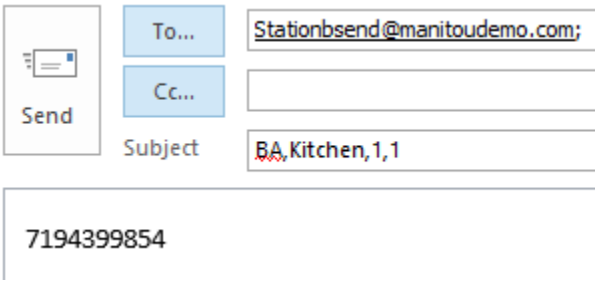
These position settings indicate to Manitou that the Transmitter ID will arrive first, the Event Code will arrive second, etc.

Our completed Data Map for this email example displays as shown in the following screenshot:



Complex Email Example

The following signal shows a more complex email Data Mapping example:



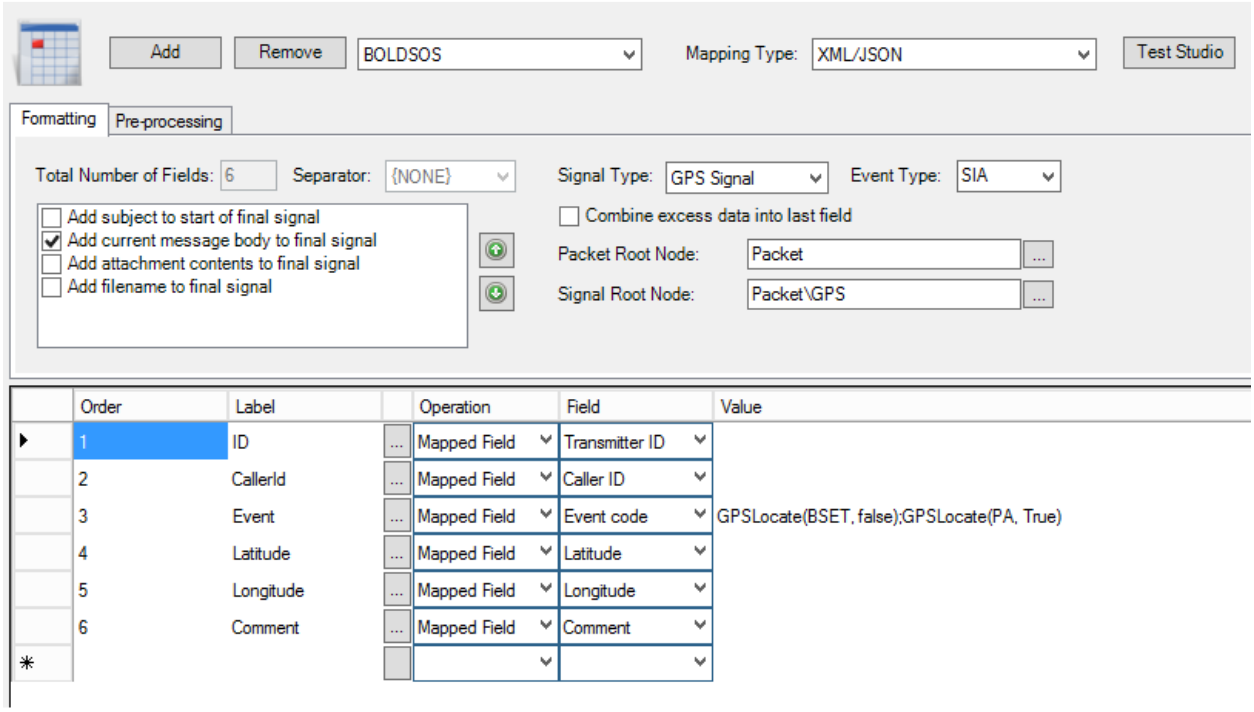
This email also includes .mpeg video footage as an attachment.

So, this email includes alarm data information in the subject, the body, and in the attachment, as follows:

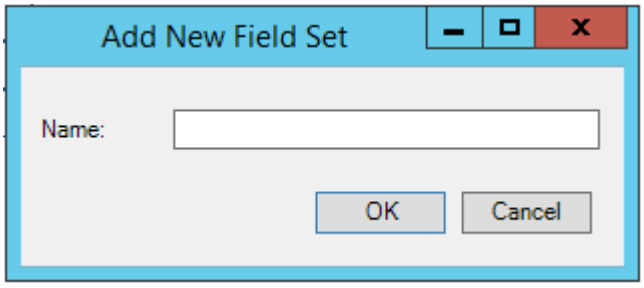
Subject: BA,Kitchen,1,1
Body: 7194399854
Attachment: alarmmotion.mpeg

Perform the following steps to create the Data Map for the complex email example:

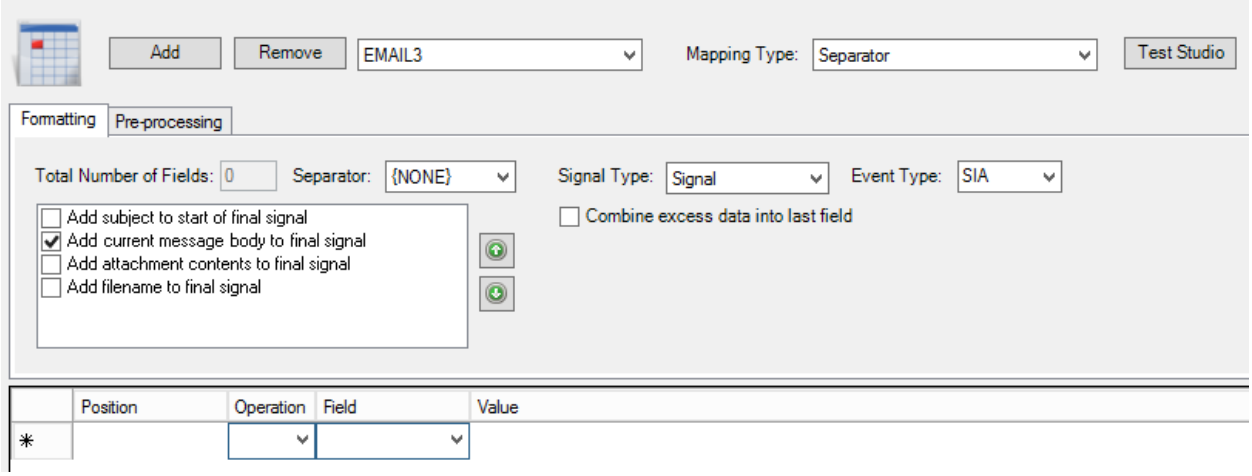
1. If you have not yet set up an Email Connector, do that first. Refer to the instructions in the MediaGateway 2 User Guide, if necessary.
2. Open the Data Mapping section of the MediaGateway 2.
   Result: the Data Mapping form displays as shown in the following screenshot:



3. Click "**Add**".
   Result: the "Add New Field Set" dialog displays as shown in the following screenshot:



4. Enter a name, and click "**OK**".
   Result: the "Add New Field Set" window closes and the system returns you to the Data Mapping form as shown in the following screenshot:

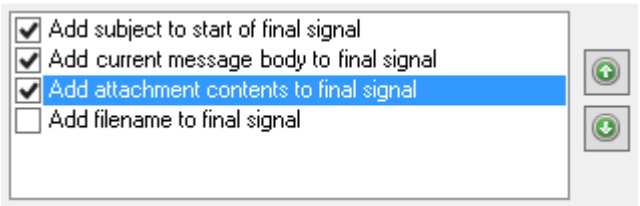


Note: because you may eventually create multiple Data Mapping field sets for email signals, we recommend using a name that allows you to differentiate your field set from others.

5. If not already selected, select "Separator" from the "Mapping Type:" dropdown menu. We are selecting this option because data in the email is separated with a comma.
6. Select the comma symbol from the "Separator:" dropdown menu.
7. If not already selected, select "Signal" from the "Signal Type:" dropdown menu. We are selecting this option because nothing in the email indicates the latitude and longitude coordinates necessary for a GPS signal, nor does the email indicate the unit and value fields associated with measurement-based devices.
8. If not already selected, select "SYS" from the "Event Type:" dropdown menu. We are selecting this option because the signal includes a BA. This is the Manitou system Event Code for a Burglary Alarm. For more information on Event Map options, refer to the Supervisor Workstation User's Manual.

   Because the email we received has alarm data in the subject, the body, and in an attachment, we need to designate to Manitou that it must handle all three parts.

9. Select the checkboxes for both "Add subject to start of final signal", and "Add attachment contents to final signal". We are selecting these options because we know that Manitou must process all three parts of the email. Once selected the checkboxes display as shown in the following screenshot:



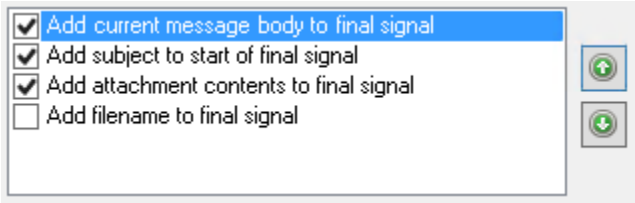
Next, we have to consider what each piece of the email alarm data means to Manitou and the order in which each data piece will be received. Here is our complex email example again:

Subject: BA,Kitchen,1,1
Body: 7194399854

Attachment: alarmmotion.mpeg

Because the Transmitter ID is contained in the body of the email, and because we want Manitou to process the Transmitter ID first, we need to change the processing order.

10. Use the arrow keys to modify the processing order as follows:



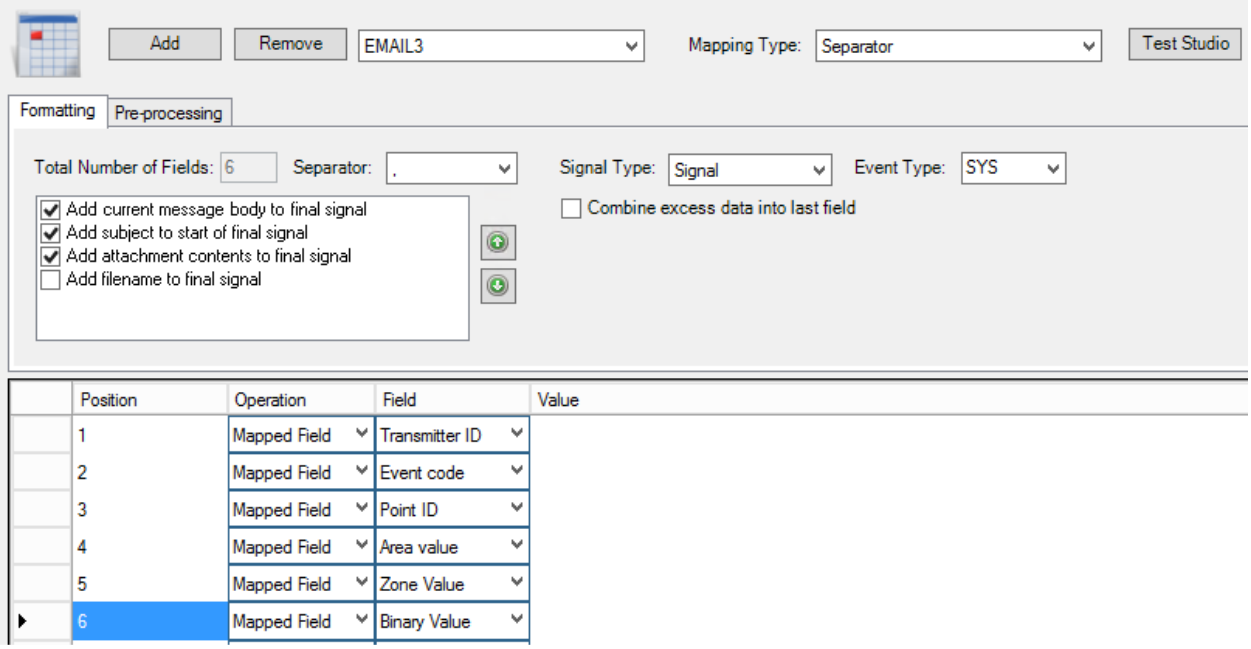
Now the order indicates to Manitou that information in the email body must be processed first.

The alarm data includes a Transmitter ID, an Event Code, a Point ID, an area, a zone, and video footage. Therefore, we need to map those six fields. Also, we need to indicate the order in which the data fields will arrive. Do that by entering the position hierarchy you want into the "Position" column as shown in the following screenshot:

	Position	Operation	Field	Value
	1	Mapped Field	Transmitter ID	
	2	Mapped Field	Event code	
	3	Mapped Field	Point ID	
	4	Mapped Field	Area value	
	5	Mapped Field	Zone Value	
▶	6	Mapped Field	Binary Value	

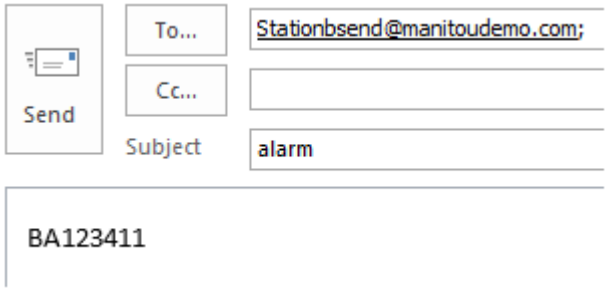
11. Click "**Save**".

Our completed Data Map for our simple email example displays as shown in the following screenshot:



Position Type Email Example

The email examples we have seen so far have used the “Separator” Mapping Type. The “Separator” type is effective to map signals that use punctuation to separate pieces of alarm data. Not all signals use Separators, however. Some signals lump the alarm data pieces together without using punctuation as you can see from the email displayed in the following screenshot:



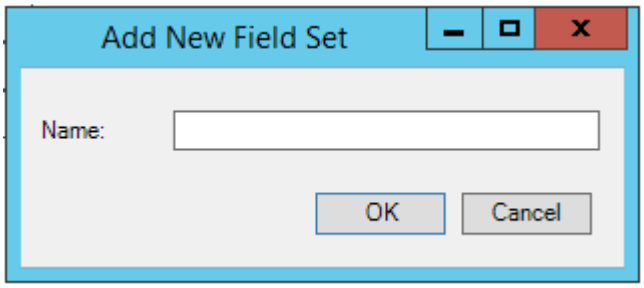
Only the body of the above email contains alarm data. The string in the email body contains the following text without punctuation Separators or spaces:

BA123411

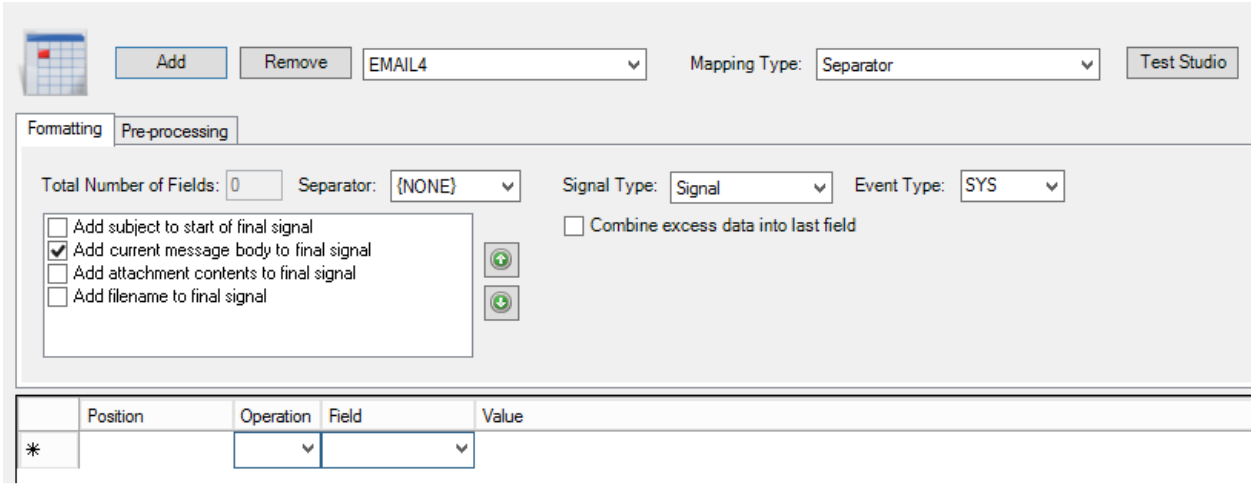
So, to decode the alarm data contained in the email, we need to know the position in which each data piece is situated and how many characters in length it is.

Perform the following steps to create a Position type Data Map:

1. Open the MediaGateway 2 Data Mapping form.
2. Click "**Add**".
   Result: the "Add New Field Set" dialog displays as shown in the following screenshot:



3. Enter a name, and click "**OK**".
   Result: the "Add New Field Set" window closes and the system returns you to the Data Mapping form as shown in the following screenshot:



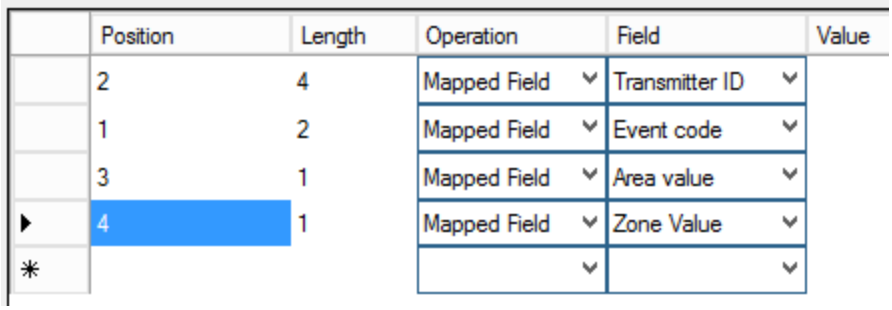
4. Select "Position" from the "Mapping Type:" dropdown menu.
5. If not already selected, choose "Signal" from the "Signal Type:" dropdown menu.
6. If not already selected, choose "SYS" from the "Event Type:" dropdown menu.

   The string in the email body contains the following text without punctuation Separators or spaces:

   BA123411

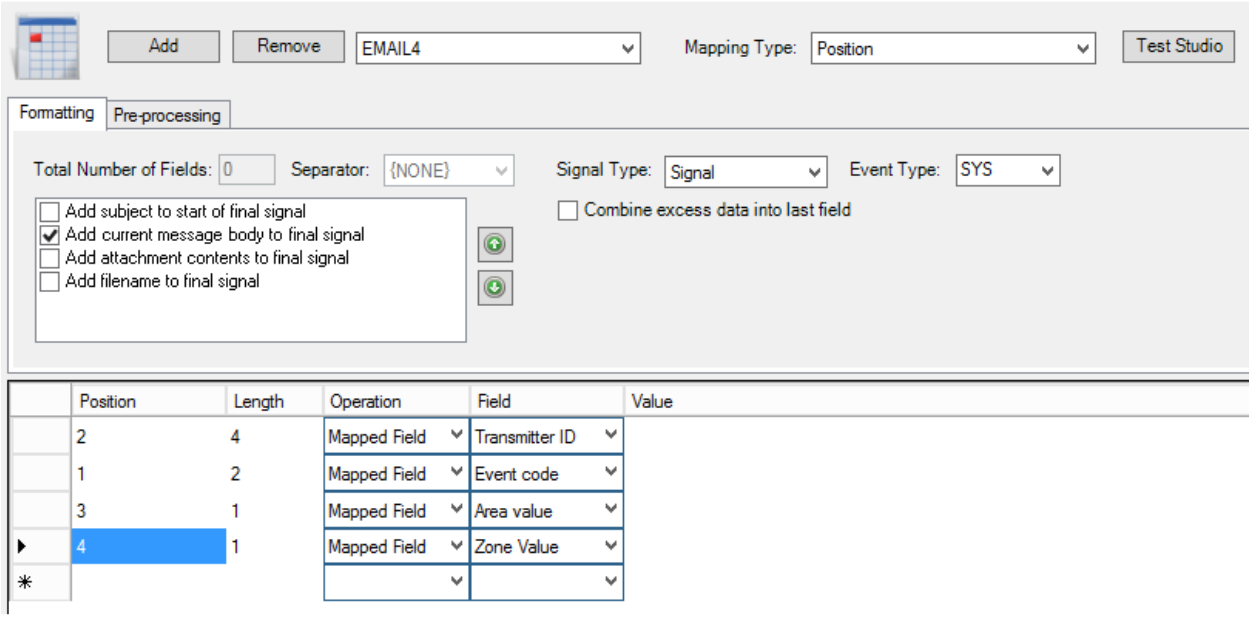
   The alarm data includes a Transmitter ID, an Event Code, an area, and a zone. Therefore, we need to map those four fields. Also, we need to indicate the order in which the data pieces will arrive.

7. To do that, enter the position hierarchy you want into the "Position" column as shown in the following screenshot:



8. Next, enter the number of characters for each data piece into the “Length” column.
9. Click “**Save**”.

Our completed Data Map for our Position type email example displays as shown in the following screenshot:



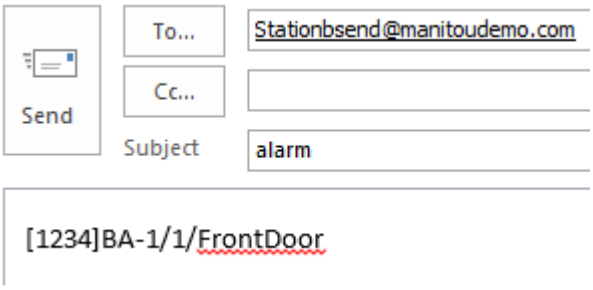
Email Pre-Processing Example

The pre-processing capability in the MediaGateway 2 allows you to address and resolve signal data deviations before Manitou ever receives the signal.

The MediaGateway Pre-processing Tab allows you to use Regular Expressions. Regular Expressions are patterns in text that can be used to parse alarm data. For example, let’s say the email signal you expect to receive almost always arrives as comma separated pieces of alarm data. However, the signal occasionally arrives as alarm data pieces separated by carriage returns. You can create a Data Map, and subject it to pre-processing to prepare Manitou for such a deviation.

Now, let’s try another example.

You received the following alarm signal via email:



Only the body of the email contains alarm data. The body contains the following string:

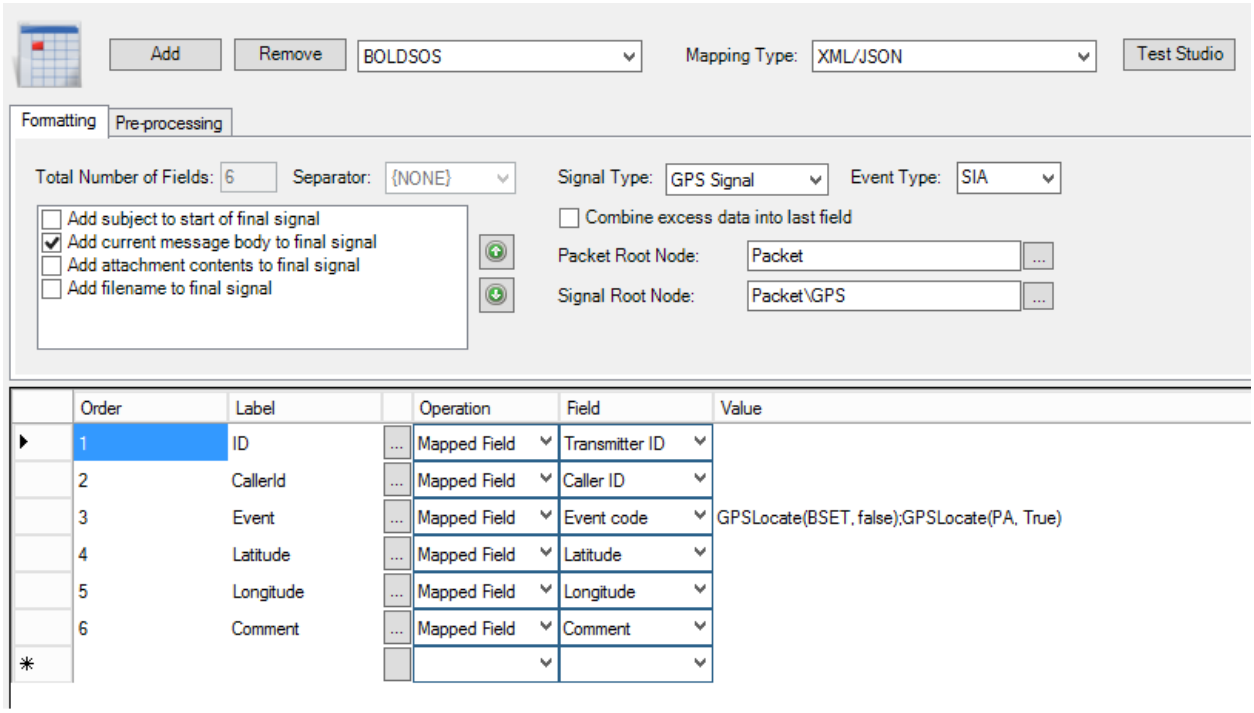
[1234]BA-1/1/FrontDoor

We want to use Regular Expressions in the MediaGateway 2 Pre-processing Tab to transform the example signal into the following string:

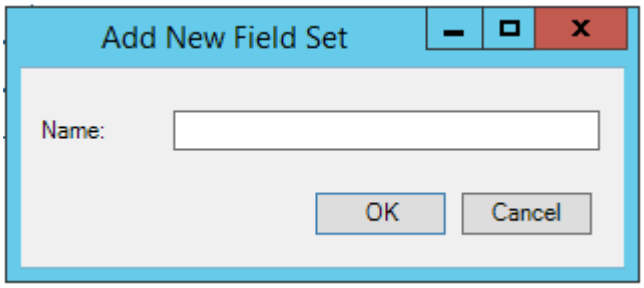
1234-BA-1-1-FrontDoor

Perform the following steps to create a Data Map for the signal, and to subject it to pre-processing:

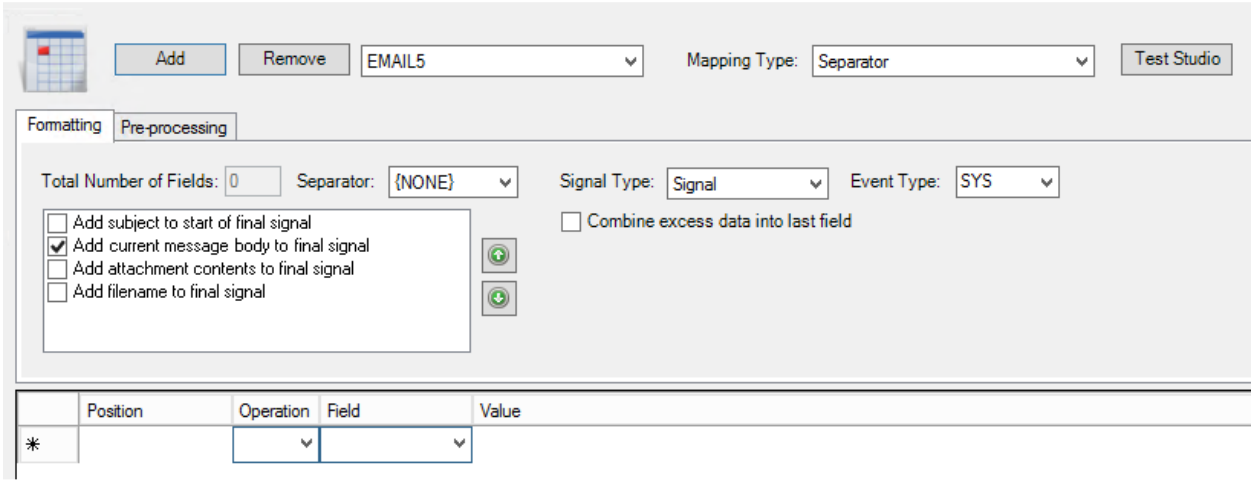
1. If you have not yet set up an Email Connector, do that first. Refer to the instructions in the MediaGateway 2 User's Guide, if necessary.
2. Open the Data Mapping section.
   Result: the Data Mapping form displays as shown in the following screenshot:



3. Click "**Add**".
   Result: the "Add New Field Set" dialog displays as shown in the following screenshot:



4. Enter a name, and click "**OK**".
   Result: the "Add New Field Set" window closes and the system returns you to the Data Mapping form as shown in the following screenshot:



   Note: because you may eventually create multiple Data Mapping field sets for email signals, we recommend using a name that allows you to differentiate your field set from others.
5. If not already selected, select "Separator" from the "Mapping Type:" dropdown menu. We are selecting this option because the signal includes multiple Separators without labels.
6. Select the hyphen symbol from the "Separator:" dropdown menu. Although there are several different punctuation Separators included in the signal, we are selecting the hyphen symbol because that will be the only Separator once pre-processing is complete.
7. If not already selected, select "Signal" from the "Signal Type:" dropdown menu. We are selecting this option because nothing in the email indicates the latitude and longitude coordinates necessary for a GPS signal, nor does the email indicate the unit and value fields associated with measurement-based devices.
8. If not already selected, select "SYS" from the "Event Type:" dropdown menu. We are selecting this option because the signal includes a BA. This is the Manitou system Event Code for a Burglary Alarm. For more information on Event Map options, refer to the Supervisor Workstation User's Manual.

   Let's look at our example signal again:

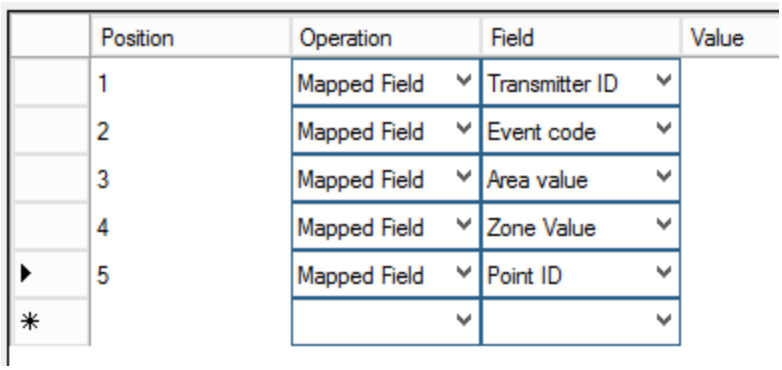
[1234]BA-1/1/FrontDoor

We have a total of five alarm data pieces for which we need to create a Data Map. Even though the signal does not include any labels, we know from our previous examples that this signal can be translated into the following five Manitou fields:

Transmitter ID: 1234
Event Code: BA
Area: 1
Zone: 1
Point ID: FrontDoor

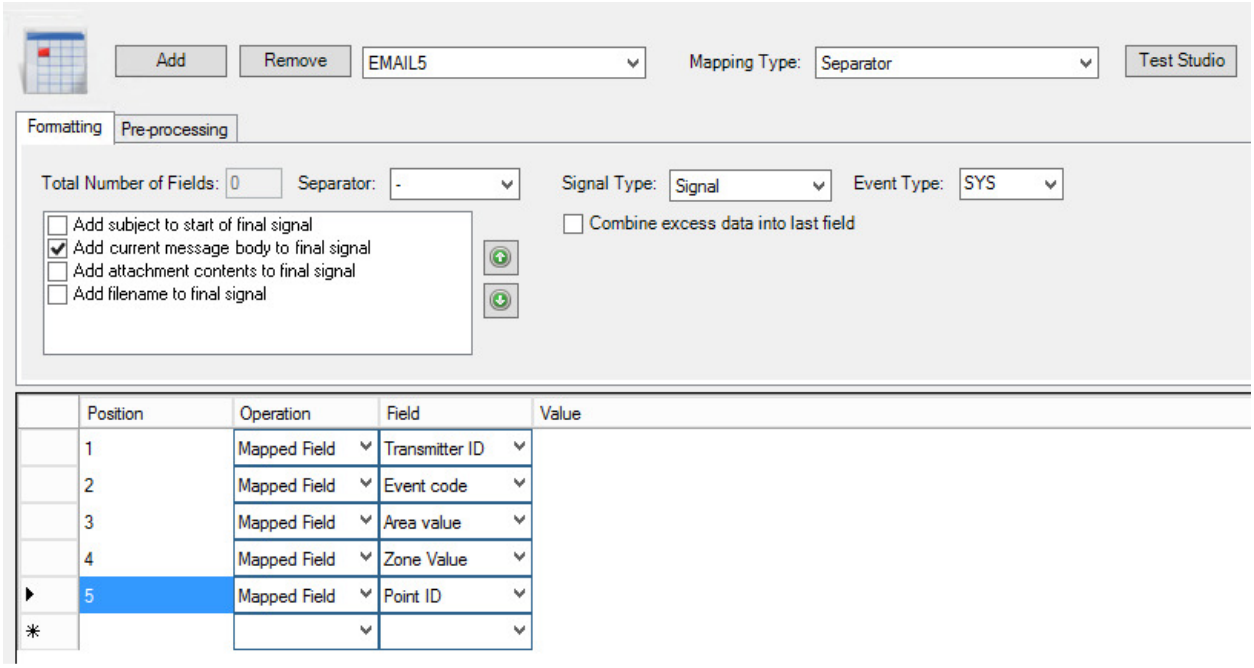
The signal includes irregular Separators that we must eventually translate using pre-processing. First though, we must create a Data Map for the signal.

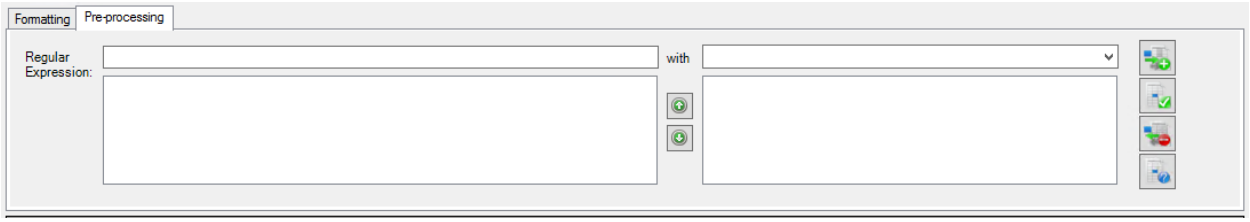
Because we have five fields, and each translates into a field in Manitou, your Data Map should match the following screenshot:



	Position	Operation	Field	Value
	1	Mapped Field	Transmitter ID	
	2	Mapped Field	Event code	
	3	Mapped Field	Area value	
	4	Mapped Field	Zone Value	
▶	5	Mapped Field	Point ID	
*				

Our completed Data Map for our pre-processing email example displays as shown in the following screenshot:

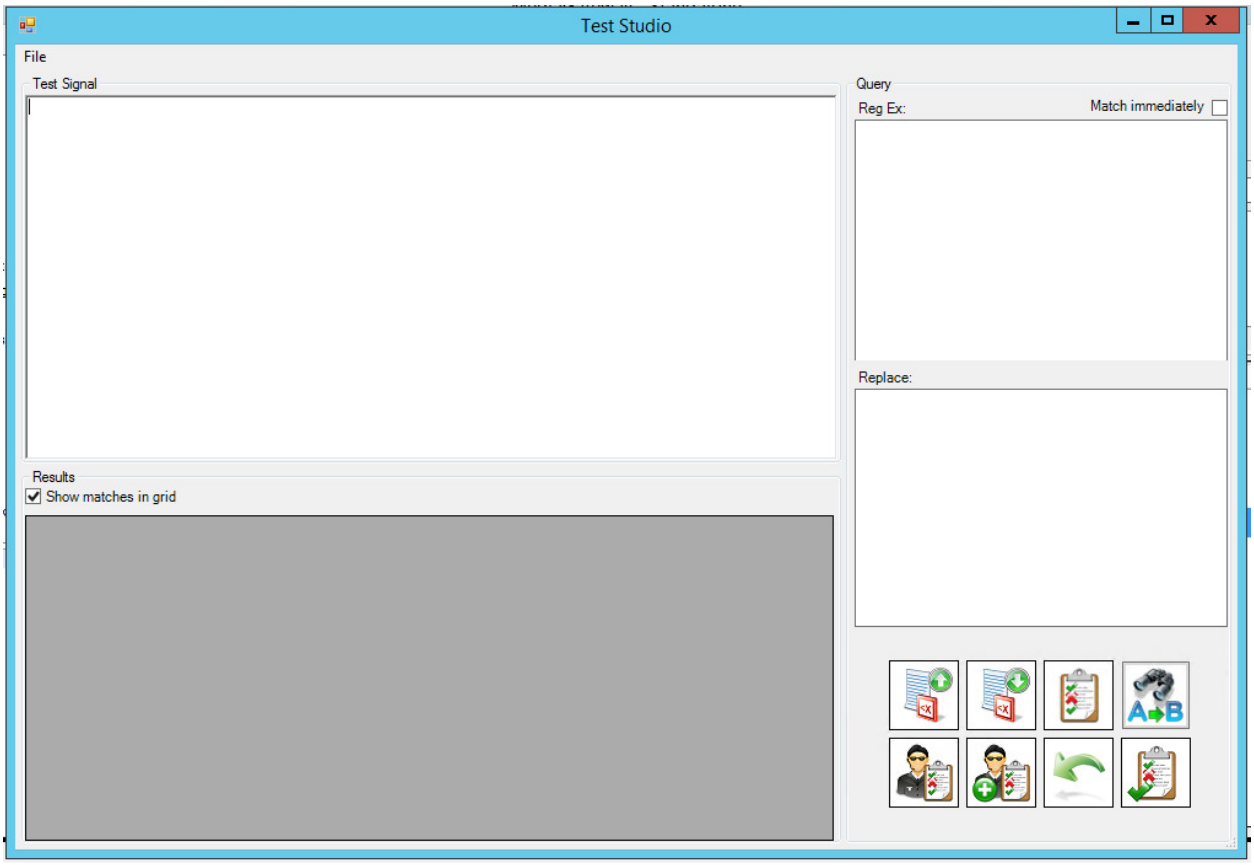




9. Click "**Save**".

   Now that our Data Map is complete, we must subject the signal to pre-processing.

10. Click "**Test Studio**".
    Result: the "Test Studio" window displays as shown in the following screenshot:



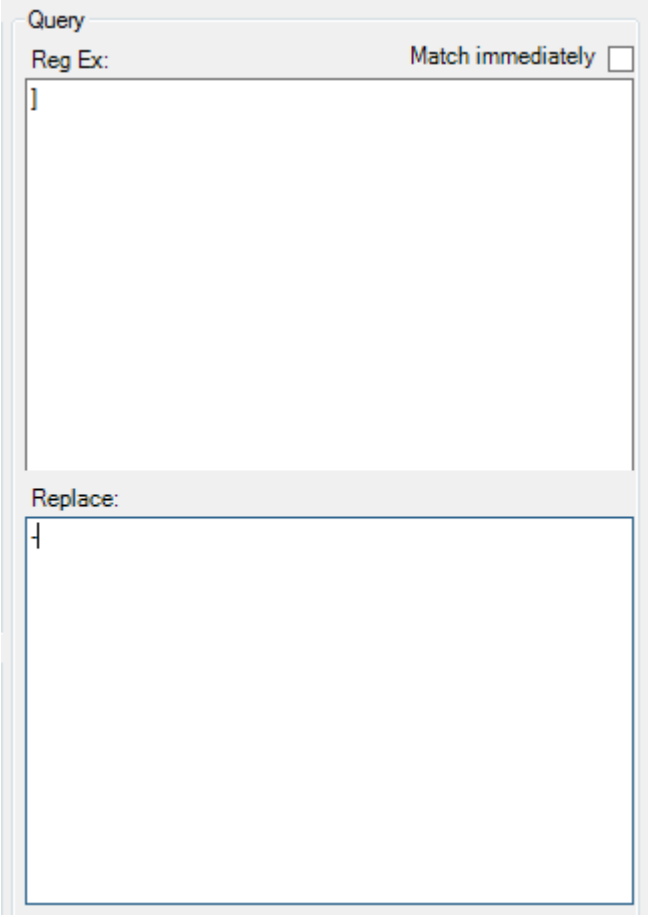
Note: the Test Studio allows you perform step-by-step Search and Replace actions on the Regular Expressions in your signal text.

11. Enter your signal text into the “Test Signal” field as shown in the following screenshot:

We want to use pre-processing to transform punctuation items acting as Separators into hyphens. The only exception to our mission relates to the first "[" bracket character in the signal. Because that character does not perform a Separator function and needs to be deleted, it requires a greater understanding of Regular Expressions than do the other pre-processing actions we have already performed.

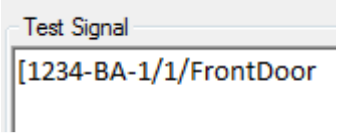
Let's perform the simple pre-processing actions first.

12. Enter "]" into "Reg Ex:" field, and "-" into the "Replace:" field as shown in the following screenshot:



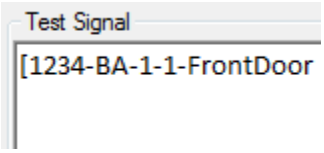
13. Click the "Search and Replace" icon shown in the following screenshot:

Result: the "]" bracket character in the signal text transforms into a hyphen as shown in the following screenshot:



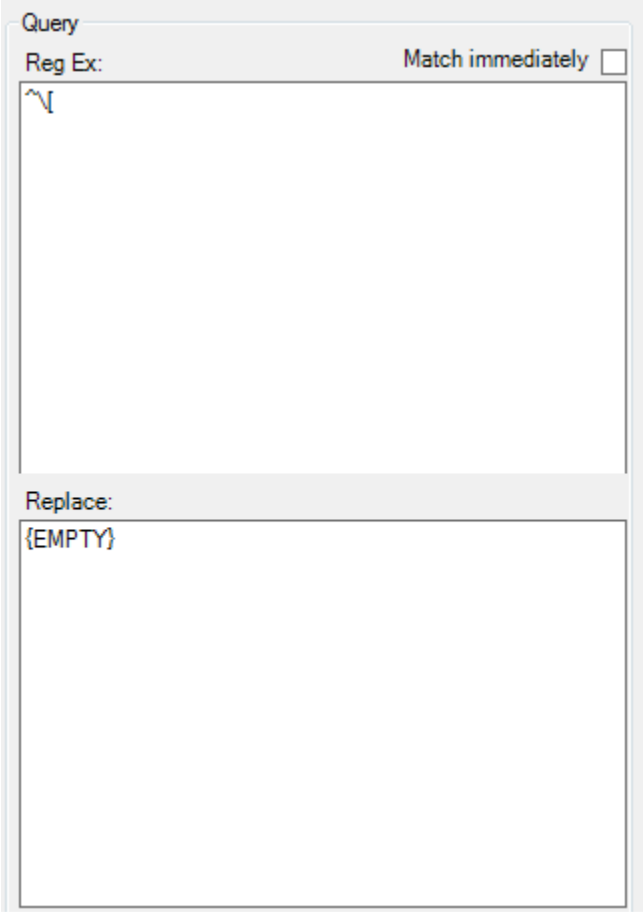
14. Enter “/” into the “Reg. Ex:” field, and the “-“ into the “Replace” field as shown in the following screenshot:

Result: the “/” in the signal text transforms into a hyphen as shown in the following screenshot:

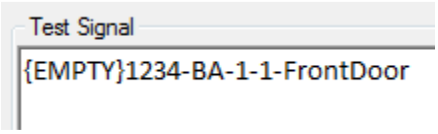


Now, the signal displays with all hyphen Separators like we want except for the “[“ bracket at the beginning of the string. We cannot merely perform a character substitution here because we want to delete the character and eliminate the blank space created by the deletion.

15. Enter the following text into the “Reg Ex:” field: “^”, “\”, and “[“ as shown in the following screenshot:

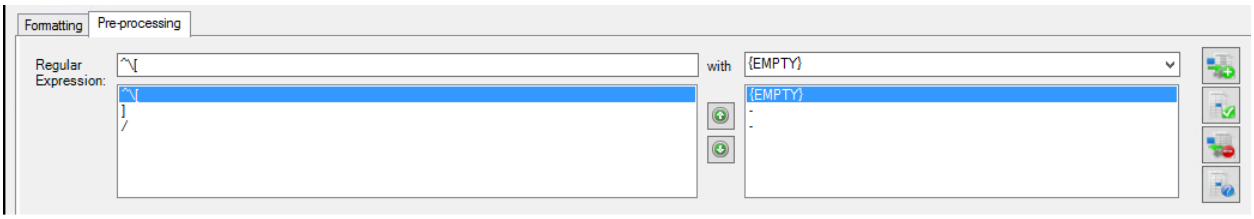


16. Enter "{EMPTY}" into the "Replace:" field as shown in the screenshot above.
    Result: the "[" character is deleted as shown in the following screenshot:



    Note: the "^" character indicates the beginning of a text string. The "\" character indicates the character following it is a literal character (i.e., exists in the string exactly as written, and is not a variable). The "{Empty}" entry means that the Regular Expression translates into nothing. The "[" character is deleted, and the empty space it would have created is also deleted. Refer to the following webpage for more information on Regular Expressions:
    http://regexlib.com/CheatSheet.aspx?AspxAutoDetectCookieSupport=1

17. Enter your Regular Expressions into the "Regular Expression:" portion on the "Pre-processing" Tab as shown in the following screenshot:



18. Click "**Save**".

XML/JSON for BoldSOS Example

When a user engages BoldSOS, the user's smartphone sends a Panic Alarm and GPS coordinates of the user's location into the MediaGateway 2.

The following text represents our example XML signal for BoldSOS:

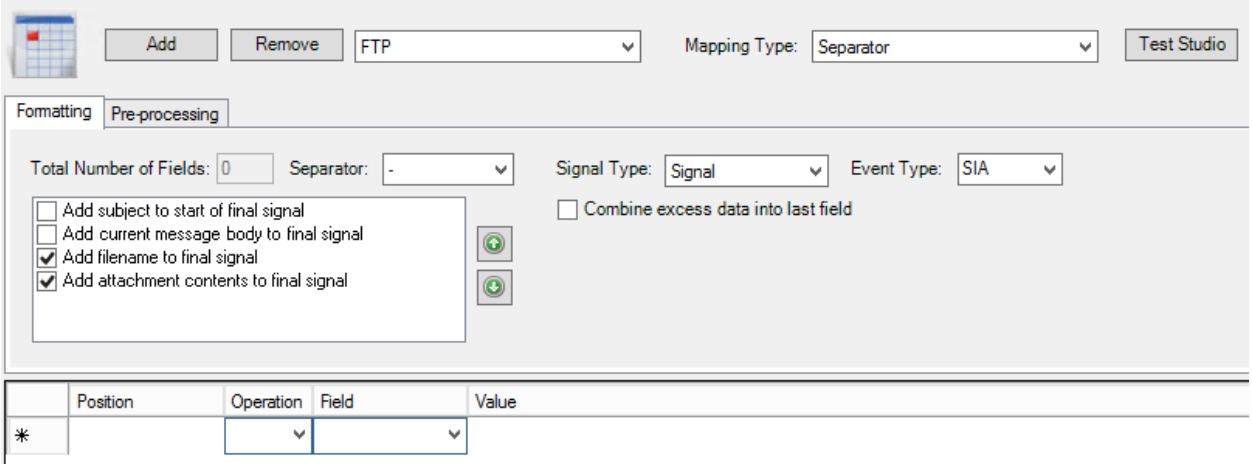
<?xml version="1.0"?><Packet ID="7195551234" CallerId="7195551234" Line="999"><GPS EvType="SIA" Event="PA"><Latitude>38.930473</Latitude><Longitude>-104.818306</Longitude><Comment>Using OTHER</Comment></GPS></Packet>

We know from the text that the signal contains the following eight pieces of alarm data:

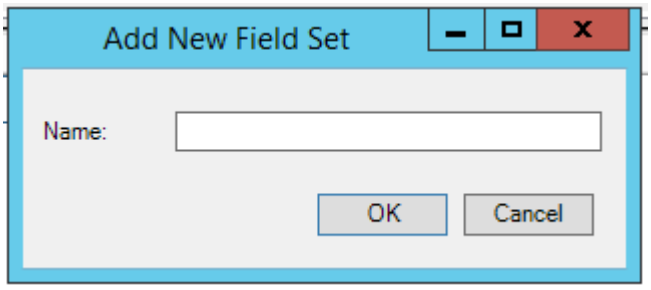
Packet ID: 7195551234
Caller ID: 7195551234
Line: 999
GPS EvType: SIA
Event Code: PA
Latitude: 38.930473
Longitude: 104.818306
Comment: Using OTHER

Perform the following steps to create a Data Map for the signal:

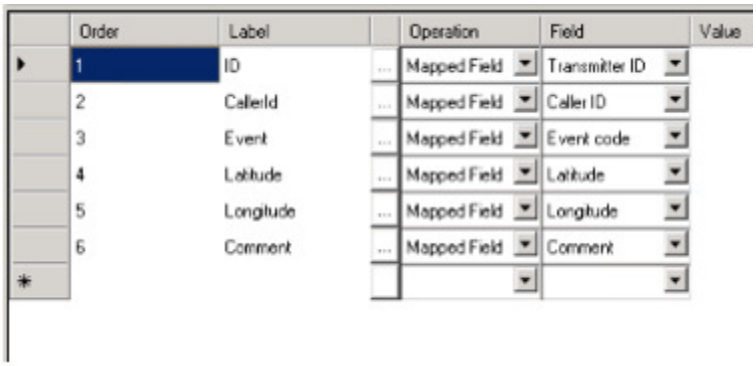
1. Open the MediaGateway 2.
2. If you have not already created the TCP Connector, do that first. Refer to the MediaGateway 2 User Guide for instructions on adding a Connector.
3. Open the Data Mapping section.
   Result: the "Data Mapping" form displays as shown in the following screenshot:



4. Click "**Add**".
   Result: the "Add New Field Set" widow displays as shown in the following screenshot:

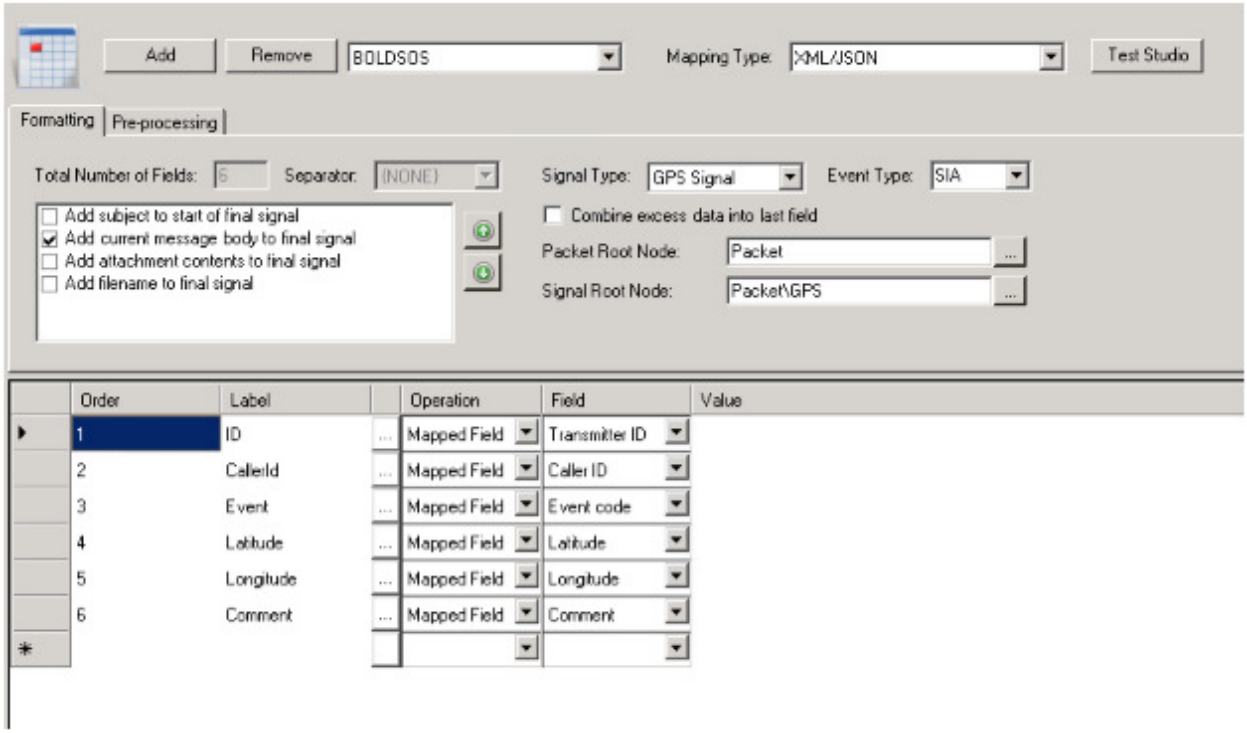


5. Enter "BoldSOS", and then click "**OK**".
   Result: the "Add New Field Set" window closes, and the Data Mapping form displays.
   Note: because you may eventually create multiple Data Mapping field sets for XML/JSON signals, we recommend using a name that allows you to differentiate your field set from others.
6. Select "XML/JSON" from the "Mapping Type:" dropdown menu.
7. Select "GPS Signal" from the "Signal Type:" dropdown menu. We are selecting this option because the signal includes latitude and longitude location information.
8. Select "SIA" from the "Event Type:" dropdown menu. We are selecting this option because the XML signal indicates that the EvType is SIA.
9. Enter information into your Data Map until it matches the following screenshot:



Note: there are eight pieces of alarm data in the signal, and only six of those fields have been mapped. We omitted the "Line" and the "GPS EvType" fields from our Data Map because it is not necessary for Manitou to receive that information. It is possible to simply leave the fields out of the Data Map because the XML/JSON Mapping Type only looks for items with the labels you designate in the Data Map.

Once complete, your Data Map should match the following screenshot:



Although your Data Map is now complete, your XML example signal requires additional data entry in the MediaGateway 2, as well as in the Manitou Supervisor and Operator Workstations.

10. Next, create a Line Driver in the MediaGateway 2 that matches the following screenshot:

Note: the FEP, Receiver, and Line information you enter into the "Properties" field may vary depending on your configuration. Refer to the MediaGateway 2 User Guide for instructions on creating a Line Driver.

11. Open the Manitou Supervisor Workstation and perform the following steps:
    a. Create a Receiver Line Prefix called "BoldSOS". Add a Receiver Line Map for your new prefix to the MediaGateway GPS Receiver. Refer to the Supervisor Workstation User Guide for instructions on creating a Receiver Line Prefix and adding a Receiver Line Map to a Receiver.

b. Create a new Audio Type called “InstantConnect”. Refer to the Supervisor Workstation User Guide for instructions on creating an Audio Type.
c. Click “**Save**”.

13. Open the Operator Workstation and perform the following steps:
    a. Create a new TX Type called “BoldSOS”. Select the “Audio Capable” checkbox, and then select “Instant Connect” from the dropdown menu. Refer to the Operator Workstation User Guide for instructions on creating a TX Type.
    b. Open the Manitou Customer for whom you want to add the BoldSOS Transmitter Type. When creating the system, select “GPS”. Do not use the default Event Monitoring option. Enter “7195551234” as the Transmitter ID. This matches the Transmitter ID in the XML signal. Refer to the Operator Workstation User Guide for instructions on creating a system on a Customer account.
    c. Click “**Save**”.