



# EDI and Fixed-Width Configuration Guide

Version 4.8.2

# Copyright

© 2018 Celigo Inc and/or its subsidiaries or affiliates. All rights reserved. This material and all Celigo Inc products are copyrighted and all rights are reserved by Celigo Inc. Celigo Inc reserves the right, at any time and without notice, to change this material or any of the functions, features, and specifications of any of the software described herein. Celigo Inc assumes no responsibility for any errors that may appear therein. The references in this material to specific platforms supported are subject to change.

# Revision Summary

Date (MM-DD-YYYY)	Revision History/Version	Revision Class	Comments	Author
1/21/2019		Major	Added Generator and Parser functions	Saumya
8/29/2018	4.8.2	Minor	Added _counter function	Rameez
2/14/2018	4.1.1	Major	Support for Fixed-Width	Rameez
1/11/2018	3.12.1	Major	Includes only EDI to JSON information	Rameez
1/23/2018	3.12.1	Major	Included JSON to EDI and other critical information	Rameez

# Table of Contents

<b>1. Preface</b>	<b>6</b>
1.1. Purpose	6
1.2. Audience	6
1.3. Typographical Conventions	6
1.4. Contacting Customer Support	6
<b>2. Introduction</b>	<b>7</b>
2.1. Sample Formats	8
2.1.1. Amazon 850 EDI File Format	8
2.1.2. Voya ASN Fixed-Width Format	9
<b>3. Generic Definition File Information</b>	<b>9</b>
3.1. File Structure	9
3.1.1. Header	10
3.1.2. Rules	10
3.1.2.1. Elements	11
3.1.2.2. Children	11
3.1.2.3. Container	12
3.1.2.4. Close Rule	13
<b>4. Functions</b>	<b>13</b>
4.1. _counter	13
<b>5. ISA (EDI To JSON)   Parse</b>	<b>14</b>
5.1. GS	17
5.1.1. ST	18
5.1.1.1. BEG	19
5.1.1.1.1. REF	21
5.1.1.1.2. FOB	22
5.1.1.1.3. CSH	23
5.1.1.1.4. DTM	23
5.1.1.1.5. PKG	24
5.1.1.1.6. N9	25
5.1.1.1.6.1. MSG	26
5.1.1.1.7. N1	27
5.1.1.1.8. PO1	28
<b>6. ISA (JSON to EDI)   Generate</b>	<b>29</b>

6.1. GS	32
6.1.1. ST	33
6.1.1.1. BEG	34
6.1.1.1.1. REF	36
6.1.1.1.2. FOB	37
6.1.1.1.3. CSH	38
6.1.1.1.4. DTM	39
6.1.1.1.5. PKG	40
6.1.1.1.6. N9	41
6.1.1.1.6.1. MSG	42
6.1.1.1.7. N1	42
6.1.1.1.8. PO1	43
<b>7. Fixed-Width to JSON   Generate</b>	<b>44</b>
7.1. 6.1. 000	44
7.2. 6.2. 100	47
7.3. 6.3. 110	49
7.4. 6.4. 200	51
7.5. 6.5. 300	56
<b>8. JSON to Fixed-Width   Parse</b>	<b>59</b>
8.1. 7.1. 000	59
8.2. 7.2. 100	62
8.3. 7.3. 110	63
8.4. 7.4. 200	66
8.5. 7.5. 300	72
<b>9. FAQs</b>	<b>74</b>

# 1. Preface

## 1.1. Purpose

The purpose of this guide is to educate the IO user on how integrator.io uses definition files to convert an EDI/Fixed-Width text file to JSON format and vice versa. This guide uses Amazon 850 EDI Specification (EDI) and Vaya ASN (Fixed-Width) for illustration purposes.

## 1.2. Audience

This guide is intended for expert level developers who want to build integrations themselves.

## 1.3. Typographical Conventions

Convention	Description
<code>{{[element]}}</code>	represents placeholder
<code>{{{[element]}}</code>	When special characters are present in the element
<code>objectproperties:</code>	represent JSON object properties
<code>//text</code>	represents comments that offer more explanation on the JSON object properties

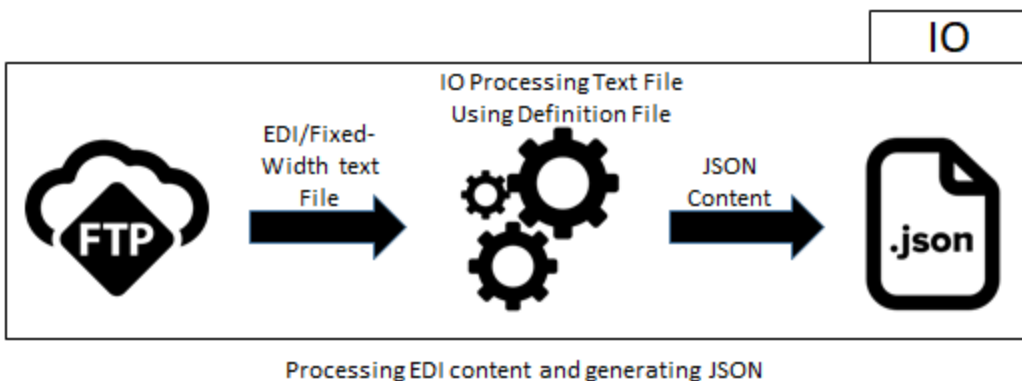
## 1.4. Contacting Customer Support

Please visit our Website for more details and for contact information:  
<https://www.celigo.com/support/>

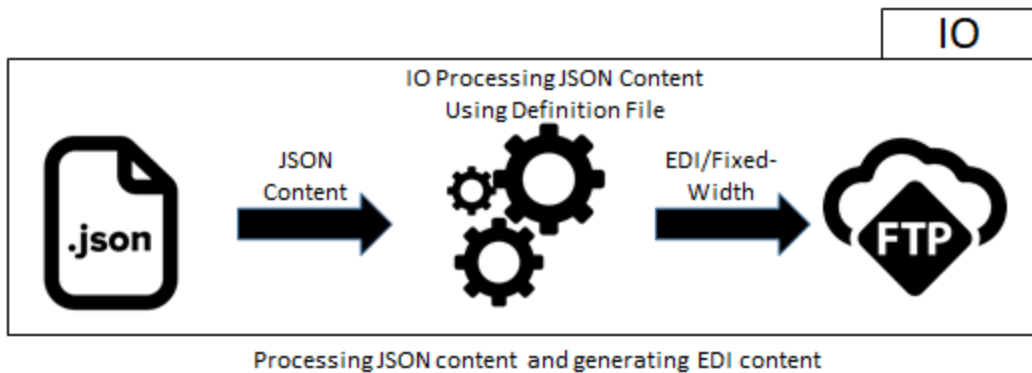
## 2. Introduction

integrator.io with the help of its definition files lets you convert an EDI/Fixed-Width format to the JSON format. This allows you to transfer your business document details from a source application, such as an FTP host to the destination application, such as NetSuite and vice versa.

For example, you can fetch a purchase order EDI file from an FTP host and send the data to NetSuite.



Or, you can fetch the purchase order details from NetSuite



to generate an EDI/Fixed-Width file that can be placed on an FTP host.

This document helps you understand how IO converts EDI/Fixed-Width content to JSON using a definition file. A very generic explanation of the definition file is present under [Generic Definition File Information](#).

### Examples of how IO converts:

- Amazon EDI 850 (Purchase Order in EDI format) to the JSON format and vice versa are present under [EDI To JSON](#) and [JSON to EDI](#). For more information on the sample file used, refer to [Amazon 850 EDI File Format](#).

**Note:** The subsections under [EDI To JSON](#) and [JSON to EDI](#) are arranged in a way to represent the parent-child relationship. The Table of Contents helps you in understanding the hierarchy.

- Voya ASN (Advance Shipment Notice in Fixed-Width format) to the JSON format and vice versa are present under [Fixed-Width to JSON](#) and [JSON to Fixed-Width](#). For more information on the sample file used, refer to [Voya ASN Fixed-Width Format](#).

## 2.1. Sample Formats

This section contains the EDI and the Fixed-Width file information that is being used as examples in this guide.

### 2.1.1. Amazon 850 EDI File Format

The following code-snippet illustrates the Amazon 850 Purchase Order EDI text file.

```
ISA*00* 00* *ZZ*AMAZON *ZZ*VENDOR *140731*0832*U*00400*000000112*0*P*>~
GS*PO*AMAZON*VENDOR*20140801*1400*367*X*004010~
ST*850*0001~
BEG*00*NE*G7984251* *20070730~
REF*CR*BTS~
REF*PD*BTS~
FOB*BP* * *01*CFR*OV*YANTIAN, CHINA 121~
CSH*N~
DTM*063*20140731~
DTM*064*20140725~
PKG*F*CS* * *40HC~
N9*L1*SPECIAL INSTRUCTIONS~
MSG*PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR
STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS FC DESTINATION.PAYMENT TERMS ARE PER
CONTAINER.~
N1*ST* *15*1553992~
PO1*1*1*EA*4.12*PE*EN*9780812508420~
CTT*5*60~
SE*16*0001~
GE*1*112~
IEA*1*000000112~
```

The EDI file will be available in the above-mentioned format. However, in the following image we have tried to create the actual hierarchy for Amazon 850 Purchase Order so that it will be easier for you to understand the processing done in the integrator.io definition file when it encounters each of these EDI lines.



```

ISA*00* *00* *ZZ*AMAZON *ZZ*VENDOR *140731*0832*U*00400*000000112*0*P*>~
GS*PO*AMAZON*VENDOR*20140801*1400*367*X*004010~
ST*850*0001~
  BEG*00*NE*G7984251* *20070730~
  REF*CR*BTS~
  REF*PD*BTS~
  FOB*BP* * *01*CFR*OV*YANTIAN, CHINA 121~
  CSH*N~
  DTM*063*20140731~
  DTM*064*20140725~
  PKG*F*CS* * *40HC~
  N9*L1*SPECIAL INSTRUCTIONS~
  MSG*PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE FLOOR-LOADED,
  N1*ST* *15*1553992~
  PO1*1*1*EA*4.12*PE*EN*9780812508420~
  CTT*5*60~
  SE*16*0001~
GE*1*112~
IEA*1*000000112~

```

## 2.1.2. Voya ASN Fixed-Width Format

The following code-snippet illustrates the Voya ASN (Advance Shipment Notice) text file in the Fixed-Width format.

```

000A20/07/2010      US010
0          0          PE-Recv Test2007H
110A          1PE-Recv Test2007H          0          0          0
100A          1PE-Recv Test2007H          20/07/2010
US0010000000235
200A          1          1          PE-Recv Test2007H          0          0.00
0          0002.0000EA
0.0000      US0010000000235      1      1          0          0
0.00000000
0      0          EA
300A          1          1          0PE-Recv Test2007H          0
0.00      0          0002.0000          US0010000000235      1      0

```

**Note:** You can copy and paste the above-mentioned code in Notepad and disable the Word Wrap to view the exact structure.

## 3. Generic Definition File Information

This section contains information on the various properties used in the integrator.io's definition files.

### 3.1. File Structure

There are two main properties in an integrator.io definition file. They are [Header](#) and [Rules](#).

```
{
  <Header>
  <Rules>
}
```

### 3.1.1. Header

Defines the header of a definition file.

The following code-snippet describes how the Header of a definition file looks like.

```
"name": "EDI 850 Purchase Order", // The EDI file name.
"description": "EDI Template-850 (Soylent Amazon Vendor Central Integration)", // The description of
what the EDI file does.
"version": 1,
"format": "delimited", // The format of the EDI file. Valid values are 'delimited' and 'fixed'.
"delimited": {
  "rowSuffix": "~", // The suffix that is present at the end of each row.
  "rowDelimiter": "\n", // The row delimiter that is present in the EDI file.
  "colDelimiter": "*" // The column delimiter that is present in the EDI file.
},
...
// The following is how the header might look when the format is fixed.
{
  "name": "fixed-width(Voya)",
  "description": "fixed-width(Voya)",
  "version": 1,
  "format": "fixed",
  "fixed": {
    "rowSuffix": "",
    "rowDelimiter": "\n",
    "paddingChar": " "
  },
  "rules": []
}
```

### 3.1.2. Rules

Defines the rules for EDI to JSON/Fixed-Width to JSON transformation and vice versa.

The following code-snippet describes the various options available in Rules to describe a row from the EDI/Fixed-Width file.

```
"Rules": [ // The rules that define the transformation of an EDI/Fixed-Width or JSON.
  {
    "maxOccurrence": 1, // The number of times that an EDI ID/Fixed-Width element should appear.
    "skipRowSuffix": true, // Set this to true if your text file has a rowSuffix.
    "required": true, // Enter true if the EDI ID/Fixed-Width element is mandatory. Otherwise, false.
    "elements": [], // The elements of an EDI ID/Fixed-Width element.
    "children": [], // Children of an EDI ID/Fixed-Width element.
    "closeRule": {} // The close rule of an EDI ID/Fixed-Width element.
  }
]
...
// The following is how rules might look for Fixed-Width files.
```

```

"rules": [
  {
    "name": "ASN", // The name of the container being created in JSON.
    "container": true,
    "required": true,
    "maxOccurrence": 50,
    "children": []
  }
]

```

### 3.1.2.1. Elements

Defines the elements of an EDI ID/Fixed-Width elements.

```

"elements": [
  {
    "name": "Element name 1",
    "value": "Ref value 1"
  },
  {
    "name": "Element name 2",
    "value": "Ref value 2"
  },
  ...
]
...
//The following is how elements might look for Fixed-Width files.

"elements": [
  {
    "name": "110", // Fixed-Width element name.
    "value": "110", // Fixed-Width element value.
    "length": 3, // The length of whitespaces.
    "direction": "right" //The direction of whitespaces. Valid values are left or right
  }
]

```

### 3.1.2.2. Children

Defines the children of an EDI ID/Fixed-Width elements.

```

"children": [
  {
    "maxOccurrence": 1,
    "skipRowSuffix": true,
    "Required": true,
    "container": true,
    "elements": [],
    "children": [],
    "closeRule": {}
  }
]

```

### 3.1.2.3. Container

Converts the EDI ID/Fixed-Width elements to a JSON array. For example, in the following example N1 is converted to a JSON array using `container`.

#### Input (EDI)

```
N1*ST* *15*1553992
```

#### Process (Definition File)

```
{
  "name": "N1",
  "maxOccurrence": 2,
  "required": true,
  "container": true,
  "children": [
    {
      "description": "Name",
      "required": true,
      "elements": [
        {
          "name": "N1",
          "value": "N1"
        },
        {
          "name": "Entity Identifier Code",
          "value": "N101"
        },
        {
          "name": "Name",
          "value": "N102"
        },
        {
          "name": "Identification Code Qualifier",
          "value": "N103"
        },
        {
          "name": "Identification Code",
          "value": "N104"
        }
      ]
    }
  ]
}
```

#### Output (JSON)

```
{
  N1:
  [ { 'Entity Identifier Code': 'ST',
      'Name': '',
      'Identification Code Qualifier': '15',
      'Identification Code': '1553992' } ],
}
```

### 3.1.2.4. Close Rule

Defines the closing rule of an EDI ID/Fixed-Width element if any.

```
"closeRule": {
  "maxOccurrence": 1,
  "required": true,
  "skipRowSuffix": true,
  "elements": [ // The elements will depend on the EDI specification. For example, refer
to close rule of ISA.
    {
      "name": "GE",
      "value": "GE"
    },
    {
      "name": "Number of Transaction Sets Included",
      "value": "GE01"
    },
    {
      "name": "Group Control Number (GE02)",
      "value": "GE02"
    }
  ]
}
```

## 4. Functions

This section describes functions that IO supports.

### 4.1. Generator

#### 4.1.1. `_counter`

`_counter` is the new EDI function introduced. It helps you to count the number of EDI segments generated for the same segment identifier. You just have to set the `counter` to `true` in the definition rules of an EDI segment. For example, you might want to count the number of TD1 segments generated based on the JSON data shown below.

#### Sample JSON

```
TD1: [ // EDI segment
{
  product: "ProductName1"
  qty: "10"
},
{
  product: "ProductName2"
  qty: "20"
},
{
  product: "ProductName3"
  qty: "40"
}
```

```
]
```

### Definition File:

```
"elements": [
{
"name": "TD1"
"value": "TD1"
}
{
"name": "<Field Name>",
"counter": true, //set the counter to true if you want to count the number of EDI segments generated for TD1.
"Value": "{{{_counter. [<Field Name>] }}"
},
...//elements of TD1
]
```

### Example

#### Example 1:

```
"elements": [
{
"name": "TD1"
"value": "TD1"
}
{
"name": "Dummy",
"counter": true,
"Value": "{{{_counter. [Dummy] }}" // dynamic value in integer
}
...//elements of TD1
]
```

#### Output:

```
TD1*1*ProductName1*10*1~// The element after TD1 is the counter.
TD1*2*ProductName2*20*1~
TD1*3*ProductName3*40*1~
```

#### Example 2:

```
"elements": [
{
"name": "TD1"
"value": "TD1"
}
{
"name": "dummy",
"counter": true,
"Value": "{{{_counter. [dummy] }}" //field not present in JSON and yet you implement _counter.
}
...//elements of TD1
]
```

#### Output:

```
TD1*1*ProductName1*10*1~// The element after TD1 is the counter.
TD1*2*ProductName2*20*1~
TD1*3*ProductName3*40*1~
```

#### 4.1.2. iterationCount:

iterationCount is the EDI function used in the closing rules. It helps you to count the number of rows generated for the respective rule. You just have to add this element in the closing rule of the rule for which you want to keep the count intact in the closing rule. You might want to count the number of N1 segments generated based on the JSON data shown below.

##### Definition rule:

```
"children": [{
  "name": "BEG",
  "description": "Begin Purchase Order Transaction",
  "relativeDataPath": "items",
  "maxOccurrence": 5,
  "required": true,
  "elements": [
    {
      "name": "segment",
      "value": "BEG"
    },
    {
      "name": "id",
      "value": "{{id}}"
    }
  ],
  "children": [
    {
      "name": "N9",
      "description": "Reference Identification",
      "required": true,
      "elements": [
        {
          "name": "segment",
          "value": "N9"
        }
      ]
    }
  ],
  {
    "name": "DTM",
    "description": "Date/Time",
    "required": true,
    "elements": [
      {
        "name": "segment",
        "value": "DTM"
      }
    ]
  },
  {
    "name": "MSG",
    "description": "Message Text",
    "relativeDataPath": "notes",
    "maxOccurrence": 5,
    "required": true,
    "elements": [
      {
```

```

        "name": "segment",
        "value": "MSG"
    },
    {
        "name": "message",
        "value": "{{description}}"
    }
]
},
{
    "name": "N1",
    "description": "Organization Name",
    "required": true,
    "elements": [
        {
            "name": "segment",
            "value": "N1"
        }
    ]
},
{
    "name": "N3",
    "description": "Address Information",
    "required": true,
    "elements": [
        {
            "name": "segment",
            "value": "N3"
        }
    ]
}
],
"closeRule": {
    "name": "CTT",
    "description": "Transaction Totals",
    "required": true,
    "elements": [
        {
            "name": "segment",
            "value": "CTT"
        },
        {
            "name": "count",
            "value": "{{_iterationCount}}"
        }
    ]
}
}
}]

```

### Example

```

[
  {
    "items": [
      {
        "id": 1,
        "notes": [
          {
            "description": "This is the item1.1",
            "description": "This is the item1.2"
          }
        ]
      }
    ],
    {

```



```

        {
          "id": 2,
          "notes": [{
            "description": "This is the item2.1"
          }],
        }
      ]]
    ]]
  ]]
}
]]

```

## Output

```

BEG*1
N9
DTM
MSG*This is the item1.1
MSG*This is the item1.2
N1
N3
CTT*8 // 8 represent the lines from BEG till CTT
BEG*2
N9
DTM
MSG*This is the item2.1
N1
N3
CTT*7 // 7 represent the lines from BEG till CTT

```

### 4.1.3. tally

tally is the EDI function used to get the tally of an element repeated to be represented in the respective closing rule. For example, if we have quantity being put in the list items and we need total quantity to be represented in the closing row of those lines items, we can have tally set to true in that element and refer that element name in the closing rule.

#### Definition rule:

```

"children": [
  {
    "name": "Group header",
    "maxOccurrence": 50,
    "required": true,
    "elements": [
      {
        "name": "Record Id",
        "value": "02"
      },
      {
        "name": "Record count",
        "value": "{{_counter.[Record count]}}",
        "counter": true
      },
      {
        "name": "Ultimate Receiver Id",
        "value": "hardcoded?"
      }
    ]
  }
]

```

```

    },
    {
      "name": "Originator Id",
      "value": "121000248"
    },
    {
      "name": "Group status",
      "value": "1"
    },
    {
      "name": "As-of date",
      "value": "{{dateFormat \"YMMDD\" asOfDate}}"
    }
  ],
  "children": [
    {
      "name": "Account header",
      "relativeDataPath": "accounts",
      "maxOccurrence": 50,
      "required": true,
      "elements": [
        {
          "name": "Record Id",
          "value": "03"
        },
        {
          "name": "Account count",
          "value": "{{_counter.[Account count]}}",
          "counter": true
        },
        {
          "name": "Account number",
          "value": "{{number}}"
        },
        {
          "name": "Amount",
          "value": "{{amount}}",
          "tally": true,
          "scope": "rule"
        }
      ],
      "children": [],
      "closeRule": {
        "name": "Account trailer",
        "required": true,
        "elements": [
          {
            "name": "Record Id",
            "value": "49"
          },
          {
            "name": "Account control number",
            "value": "{{_tally.Amount}}",
            "tally": true
          }
        ]
      }
    }
  ],
  "closeRule": {

```

```

        "name": "Group trailer",
        "required": true,
        "elements": [
            {
                "name": "Record Id",
                "value": "98"
            },
            {
                "name": "Group control number",
                "value": "{{_tally.[Account control number]}}"
            }
        ]
    }
}
]

```

#### Example:

```

"accounts": [
    {
        "number": 123,
        "currency": "USD",
        "typeCode": "100",
        "amount": 101,
        "itemCount": 3
    },
    {
        "number": 456,
        "currency": "CDN",
        "typeCode": "400",
        "amount": 51,
        "itemCount": 2
    }
]

```

#### Output:

```

02,1,hardcoded?,121000248,1,200121
03,1,123,USD,101
49,101 // 101 is the tally of amount present in 03 row
03,2,456,CDN,51
49,51 // 51 is the tally of amount present in 03 row
98,152 // 152 is tally of amounts present in both rows starting with 49

```

## 4.1.4 Length of any array

We can use `.length` keyword to get the length of an array structure that we are iterating through (we should be either in the closing rule of that respective array iteration rule or in the parental ruleset to get the length of that array.)

## Definition rule:

```
"children": [
  {
    "name": "Account header",
    "relativeDataPath": "accounts",
    "maxOccurrence": 50,
    "required": true,
    "elements": [
      {
        "name": "Record Id",
        "value": "03"
      },
      {
        "name": "Account number",
        "value": "{{number}}"
      },
      {
        "name": "Currency",
        "value": "{{currency}}"
      }
    ],
    "children": [],
    "closeRule": {
      "name": "Account trailer",
      "required": true,
      "elements": [
        {
          "name": "Record Id",
          "value": "49"
        },
        {
          "name": "Record count",
          "value": "{{{{accounts.length}}}"
        }
      ]
    }
  }
]
```

## Example

```
"accounts": [
  {
    "number": 123,
    "currency": "USD",
    "typeCode": "100",
    "amount": 101,
    "itemCount": 3
  },
  {
    "number": 456,
    "currency": "CDN",
    "typeCode": "400",
    "amount": 51,
    "itemCount": 2
  }
]
```

## Output

```
03,123,USD
03,456,CDN
49,2 // 2 is the length of accounts array
```

## 4.2. Parser

### 4.2.1. ignore

This is the flag that can be set at element level. If we set the flag, the respective element will not be reflected on the parsed json output.

#### Definition

```
{
  "required": true,
  "skipRowSuffix": true,
  "elements": [{
    "name": "BH",
    "value": "BH",
    "length": 2
  },
  {
    "name": "name",
    "value": "{{name}}",
    "length": 2,
    "ignore": true
  },
  {
    "name": "id",
    "value": "{{id}}",
    "length": 7
  }
  ]
}
```

#### Example

```
BH500B051820180105
```

#### Output

```
{
  "id": "0B05182"
}
```

Although we had a second element as "name" with length set to 2 which will result in "50" from the given content, since element has ignored the flag that entry is being ignored in the json.

## 4.2.2. ignoreIdentifier

This is the flag we can set on a rule of a container if we cannot rely on the starting element as the identifier to recognise its repetition to gather them into an array structure.

### Definition

```
{
  "name": "Generic fixed-width",
  "description": "Sample fixed width definition to test parser features",
  "version": 1,
  "format": "fixed",
  "fixed": {
    "rowSuffix": "~",
    "rowDelimiter": "\n",
    "paddingChar": ""
  },
  "rules": [{
    "name": "file Header",
    "required": true,
    "elements": [{
      "name": "HeaderRow",
      "value": "HMAPIDCOMP",
      "length": 10,
      "ignore": true
    }],
    "children": [{
      "name": "data",
      "description": "This is a container",
      "container": "true",
      "required": true,
      "maxOccurrence": 5,
      "children": [{
        "required": true,
        "skipRowSuffix": true,
        "elements": [{
          "name": "BH",
          "value": "BH",
          "length": 2
        }],
        {
          "name": "name",
          "value": "{{name}}",
          "length": 2,
          "ignore": true
        },
        {
          "name": "id",
          "value": "{{id}}",
          "length": 7
        }
      ]
    }],
    "children": [{
      "name": "items",
      "required": true,
      "maxOccurrence": 100,
      "container": true,

```

```
        "children": [{
            "required": true,
            "skipRowSuffix": true,
            "ignoreIdentifier": true,
            "elements": [{
                "name": "Id",
                "value": "02",
                "length": 3
            },
            {
                "name": "id",
                "value": "{{id}}",
                "length": 4
            },
            {
                "name": "descr",
                "value": "{{qty}}",
                "length": 4
            }
        ]
    }
    ]},
    "closeRule": {
        "required": true,
        "skipRowSuffix": true,
        "elements": [{
            "name": "BT",
            "value": "BT",
            "length": 2
        },
        {
            "name": "qty",
            "value": "{{qty}}",
            "length": 2
        }
    ]
    }
    ],
    "closeRule": {
        "required": true,
        "elements": [{
            "name": "T",
            "value": "T",
            "length": 1
        }
    ]
    }
    ]
}
```

## Example

```
HMAPIDCOMP120228962120180105010520180005
BH500B051820180105
001500B0518
002500B0518
003500B051
010500B0518
011500B0518
012500B0518
BT060B05180150864900
```

In this kind of fixed-width file we cannot rely on starting tags to establish our container rule, for that purpose we have introduced this flag.

## Output

```
{ "data": [
  {
    "id": "0B05182",
    "items": [
      {
        "Id": "001",
        "id": "500B",
        "descr": "0518"
      },
      {
        "Id": "002",
        "id": "500B",
        "descr": "0518",
      },
      {
        "Id": "003",
        "id": "500B",
        "descr": "0518"
      },
      {
        "Id": "010",
        "id": "500B",
        "descr": "0518"
      },
      {
        "Id": "011",
        "id": "500B",
        "descr": "0518"
      },
      {
        "Id": "012",
        "id": "500B",
        "descr": "0518"
      }
    ],
    "qty": "06" }
]
```



## 5. ISA (EDI To JSON) | Parse

This section describes how `ISA` an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

### Input (EDI)

```
ISA*00* *00* *ZZ*AMAZON *ZZ*VENDOR *140731*0832*U*00400*000000112*0*P*>~  
IEA*1*000000112~
```

### Process (Definition File)

```
{  
  <Generic File Information comes here>  
  
  "rules": [  
    {  
      "maxOccurrence": 1, // The number of times the ID can appear in an EDI file. For example, ISA must appear only  
once in an EDI file.  
      "required": true, //  
      "elements": [ // This array represents each value that appears after column delimiter (**) that is present in the input  
text of the EDI file.  
        {  
          "name": "ISA",  
          "value": "ISA"  
        },  
  
        {  
          "name": "Authorization Information Qualifier",  
          "value": "ISA01"  
        },  
  
        {  
          "name": "Authorization Information",  
          "value": "ISA02"  
        },  
  
        {  
          "name": "Security Information Qualifier",  
          "value": "ISA03"  
        },  
  
        {  
          "name": "Security Information",  
          "value": "ISA04"  
        },  
  
        {  
          "name": "Interchange ID Qualifier",  
          "value": "ISA05"  
        },  
  
        {  
          "name": "Interchange Sender ID",  
          "value": "ISA06"  
        },  
  
        {  
          "name": "Interchange ID Qualifier footer",  
          "value": "ISA07"  
        },  
  
        {  
          "name": "Interchange Receiver ID",  
          "value": "ISA08"  
        },  
  
        {  
          "name": "Interchange Date",
```

```

        "value": "ISA09"
    },
    {
        "name": "Interchange Time",
        "value": "ISA10"
    },
    {
        "name": "ISA11",
        "value": "ISA11"
    },
    {
        "name": "Interchange Control Version Number",
        "value": "ISA12"
    },
    {
        "name": "Interchange Control Number",
        "value": "ISA13"
    },
    {
        "name": "Acknowledgment Requested",
        "value": "ISA14"
    },
    {
        "name": "Usage Indicator",
        "value": "ISA15"
    },
    {
        "name": "Component Element Separator",
        "value": "ISA16"
    }
},
"children": [GS], // All the below subsections are children for ISA. The subsections are arranged in an way to
represent the parent-child relationship. This stands true for the "children":[] array in all the below subsections.

"closeRule": { // Closing rule for ISA as per Amazon 850 EDI specification.
    "required": true,
    "maxOccurrence": 1,
    "elements": [
        {
            "name": "IEA",
            "value": "IEA"
        },
        {
            "name": "Number of Included Functional Groups",
            "value": "IEA01"
        },
        {
            "name": "Interchange Control Number footer",
            "value": "IEA02"
        }
    ]
}
]
}
}

```

## Output (JSON)

```

{
  'Authorization Information Qualifier': '00',
  'Authorization Information': '',
  'Security Information Qualifier': '00',
  'Security Information': '',
  'Interchange ID Qualifier': 'ZZ',
  'Interchange Sender ID': 'AMAZON',
  'Interchange ID Qualifier footer': 'ZZ',
  'Interchange Receiver ID': 'VENDOR',
  'Interchange Date': '140731',
  'Interchange Time': '0832',
  'ISAll': 'U',
  'Interchange Control Version Number': '00400',
  'Interchange Control Number': '000000112',
  'Acknowledgment Requested': '0',
  'Usage Indicator': 'P',
  'Component Element Separator': '>',
  'Number of Included Functional Groups': '1',
  'Interchange Control Number footer': '000000112'
}

```

## 5.1. GS

This section describes how GS an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

### Input (EDI)

```

GS*PO*AMAZON*VENDOR*20140801*1400*367*X*004010~
GE*1*112~

```

### Process (Definition File)

```

{
  "required": true,
  "maxOccurrence": 1,
  "elements": [
    {
      "name": "GS",
      "value": "GS"
    },
    {
      "name": "Functional Identifier Code",
      "value": "GS01"
    },
    {
      "name": "Application Sender's Code",
      "value": "GS02"
    },
    {
      "name": "Application Receiver's Code",
      "value": "GS03"
    },
    {
      "name": "Date",
      "value": "GS04"
    },
    {

```

```

        "name": "Time",
        "value": "GS05"
    },
    {
        "name": "Group Control Number",
        "value": "GS06"
    },
    {
        "name": "Responsible Agency Code",
        "value": "GS07"
    },
    {
        "name": "Version / Release / Industry Identifier Code",
        "value": "GS08"
    }
],
"children": [ST],
"closeRule": {
    "maxOccurrence": 1,
    "required": true,
    "elements": [
        {
            "name": "GE",
            "value": "GE"
        },
        {
            "name": "Number of Transaction Sets Included",
            "value": "GE01"
        },
        {
            "name": "Group Control Number footer",
            "value": "GE02"
        }
    ]
}
}

```

## Output (JSON)

```

{
  'Functional Identifier Code': 'PO',
  'Application Sender\'s Code': 'AMAZON',
  'Application Receiver\'s Code': 'VENDOR',
  Date: '20140801',
  Time: '1400',
  'Group Control Number': '367',
  'Responsible Agency Code': 'X',
  'Version / Release / Industry Identifier Code': '004010',
  'Number of Transaction Sets Included': '1',
  'Group Control Number footer': '112'
}

```

### 5.1.1. ST

This section describes how `ST` an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```
ST*850*0001~  
SE*16*0001~
```

#### Process (Definition File)

```
{  
  "maxOccurrence": 1,  
  "required": true,  
  "elements": [  
    {  
      "name": "ST",  
      "value": "ST"  
    },  
    {  
      "name": "Transaction Set Identifier Code",  
      "value": "ST01"  
    },  
    {  
      "name": "Transaction Set Control Number",  
      "value": "ST02"  
    }  
  ],  
  "children": [BEG],  
  "closeRule": {  
    "maxOccurrence": 1,  
    "required": true,  
    "elements": [  
      {  
        "name": "SE",  
        "value": "SE"  
      },  
      {  
        "name": "Number of Included Segments",  
        "value": "SE01"  
      },  
      {  
        "name": "Transaction Set Control Number footer",  
        "value": "SE02"  
      }  
    ]  
  }  
}
```

#### Output (JSON)

```
{  
  'Transaction Set Identifier Code': '850',  
  'Transaction Set Control Number': '0001',  
  'Number of Included Segments': '16',  
  'Transaction Set Control Number footer': '0001'  
}
```

### 5.1.1.1. BEG

This section describes how BEG an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```
BEG*00*NE*G7984251* *20070730~  
CTT*5*60~
```

#### Process (Definition File)

```
{  
  "maxOccurrence": 1,  
  "required": true,  
  "elements": [  
    {  
      "name": "BEG",  
      "value": "BEG"  
    },  
    {  
      "name": "Transaction Set Purpose Code",  
      "value": "BEG01"  
    },  
    {  
      "name": "Purchase Order Type Code",  
      "value": "BEG02"  
    },  
    {  
      "name": "Purchase Order Number",  
      "value": "BEG03"  
    },  
    {  
      "name": "Optional 2",  
      "value": "BEG04"  
    },  
    {  
      "name": "Date footer",  
      "value": "BEG05"  
    }  
  ],  
  "children": [REF, FOB, CSH, DTM, PKG, N9, N1, PO1],  
  "closeRule": {  
    "maxOccurrence": 1,  
    "required": true,  
    "elements": [  
      {  
        "name": "CTT",  
        "value": "CTT"  
      },  
      {  
        "name": "Number of Line Items",  
        "value": "CTT01"  
      },  
      {  
        "name": "Hash Total",  
        "value": "CTT02"  
      }  
    ]  
  }  
}
```

```
}  
}
```

## Output (JSON)

```
{  
  'Transaction Set Purpose Code': '00',  
  'Purchase Order Type Code': 'NE',  
  'Purchase Order Number': 'G7984251',  
  'Optional 2': '',  
  'Date footer': '20070730',  
  'Number of Line Items': '5',  
  'Hash Total': '60'  
}
```

### 5.1.1.1.1. REF

This section describes how REF an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```
REF*CR*BTS~  
REF*PD*BTS~
```

#### Process (Definition File)

```
{  
  "name": "REF",  
  "maxOccurrence": 2,  
  "required": true,  
  "container": true,  
  "children": [  
    {  
      "required": true,  
      "elements": [  
        {  
          "name": "REF",  
          "value": "REF"  
        },  
        {  
          "name": "Reference Identification Qualifier",  
          "value": "REF01"  
        },  
        {  
          "name": "Reference Identification",  
          "value": "REF02"  
        }  
      ]  
    }  
  ]  
}
```

#### Output (JSON)

```
{
```

```

REF:
  [ { 'Reference Identification Qualifier': 'CR',
      'Reference Identification': 'BTS' },
    { 'Reference Identification Qualifier': 'PD',
      'Reference Identification': 'BTS' } ],
}

```

#### 5.1.1.1.2. FOB

This section describes how FOB an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```

FOB*BP* * *01*CFR*OV*YANTIAN, CHINA 121~

```

#### Process (Definition File)

```

{
  "name": "FOB",
  "maxOccurrence": 2,
  "required": true,
  "container": true,
  "children": [
    {
      "required": false,
      "elements": [
        {
          "name": "FOB",
          "value": "FOB"
        },
        {
          "name": "Shipment Method of Payment",
          "value": "FOB01"
        },
        {
          "name": "Optional 3",
          "value": "FOB02"
        },
        {
          "name": "Optional 4",
          "value": "FOB03"
        },
        {
          "name": "Transportation Terms Qualifier Code",
          "value": "FOB04"
        },
        {
          "name": "Transportation Terms Code",
          "value": "FOB05"
        },
        {
          "name": "Location Qualifier",
          "value": "FOB06"
        },
        {
          "name": "Description",
          "value": "FOB07"
        }
      ]
    }
  ]
}

```



```
    ]
  }
}
```

## Output (JSON)

```
{
  FOB:
  [ { 'Shipment Method of Payment': 'BP',
      'Optional 3': '',
      'Optional 4': '',
      'Transportation Terms Qualifier Code': '01',
      'Transportation Terms Code': 'CFR',
      'Location Qualifier': 'OV',
      Description: 'YANTIAN, CHINA 121' } ],
}
```

### 5.1.1.1.3. CSH

This section describes how CSH an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

## Input (EDI)

```
CSH*N~
```

## Process (Definition File)

```
{
  "name": "CSH",
  "maxOccurrence": 5,
  "required": true,
  "container": true,
  "children": [
    {
      "required": true,
      "elements": [
        {
          "name": "CSH",
          "value": "CSH"
        },
        {
          "name": "Sales Requirement Code",
          "value": "CSH01"
        }
      ]
    }
  ]
}
```

## Output (JSON)

```
{
```

```
CSH: [ { 'Sales Requirement Code': 'N' } ],
}
```

#### 5.1.1.1.4. DTM

This section describes how `DTM` an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```
DTM*063*20140731~
DTM*064*20140725~
```

#### Process (Definition File)

```
{
  "name": "DTM",
  "maxOccurrence": 10,
  "required": true,
  "container": true,
  "children": [
    {
      "required": false,
      "elements": [
        {
          "name": "DTM",
          "value": "DTM"
        },
        {
          "name": "Date/Time Qualifier",
          "value": "DTM01"
        },
        {
          "name": "Date",
          "value": "DTM02"
        }
      ]
    }
  ]
}
```

#### Output (JSON)

```
{
  DTM:
  [ { 'Date/Time Qualifier': '063', Date: '20140731' },
    { 'Date/Time Qualifier': '064', Date: '20140725' } ],
}
```

#### 5.1.1.1.5. PKG

This section describes how `PKG` an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

## Input (EDI)

```
PKG*F*CS* * *40HC~
```

## Process (Definition File)

```
{
  "name": "PKG",
  "maxOccurrence": 2,
  "required": true,
  "container": true,
  "children": [
    {
      "description": "Marking, Packaging, Loading Details",
      "required": false,
      "elements": [
        {
          "name": "PKG",
          "value": "PKG"
        },
        {
          "name": "Item Description Type",
          "value": "PKG01"
        },
        {
          "name": "Packaging Characteristic Code",
          "value": "PKG02"
        },
        {
          "name": "Optional 5",
          "value": "PKG03"
        },
        {
          "name": "Optional 6",
          "value": "PKG04"
        },
        {
          "name": "Description",
          "value": "PKG05"
        }
      ]
    }
  ]
}
```

## Output (JSON)

```
{
  PKG:
  [ { 'Item Description Type': 'F',
      'Packaging Characteristic Code': 'CS',
      'Optional 5': '',
      'Optional 6': '',
      Description: '40HC' } ],
}
```

#### 5.1.1.1.6. N9

This section describes how N9 an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

##### Input (EDI)

```
N9*L1*SPECIAL INSTRUCTIONS~
```

##### Process (Definition File)

```
{
  "name": "N9",
  "maxOccurrence": 2,
  "container": true,
  "required": true,
  "children": [
    {
      "name": "N9",
      "required": true,
      "skipRowSuffix": true,
      "elements": [
        {
          "name": "N9",
          "value": "N9"
        },
        {
          "name": "Reference Identification Qualifier",
          "value": "N901"
        },
        {
          "name": "Reference Identification",
          "value": "N902"
        }
      ]
    }
  ],
  {MSG}
}
```

##### Output (JSON)

```
{
  N9:
  [ { 'Reference Identification Qualifier': 'L1',
    'Reference Identification': 'SPECIAL INSTRUCTIONS',
    MSG: [ { <message here> } ] } ],
}
```

#### 5.1.1.1.6.1. MSG

This section describes how MSG an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

##### Input (EDI)

```
MSG*PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS FC DESTINATION.PAYMENT TERMS ARE PER
```

```
CONTAINER.~
```

### Process ( Definition File)

```
{
  "name": "MSG",
  "maxOccurrence": 2,
  "container": true,
  "required": true,
  "children": [
    {
      "required": false,
      "elements": [
        {
          "name": "MSG",
          "value": "MSG"
        },
        {
          "name": "Free-Form Message Text",
          "value": "MSG01"
        }
      ]
    }
  ]
}
```

### Output (JSON)

```
MSG: [ { { 'Free-Form Message Text': 'PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE
FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS
FC DESTINATION.PAYMENT TERMS ARE PER CONTAINER.' } } ] ,
```

#### 5.1.1.1.7. N1

This section describes how N1 an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

### Input (EDI)

```
N1*ST* *15*1553992~
```

### Process (Definition File)

```
{
  "name": "N1",
  "maxOccurrence": 2,
  "required": true,
  "container": true,
  "children": [
    {
      "description": "Name",
      "required": true,
      "elements": [
        {
          "name": "N1",
          "value": "N1"
        }
      ]
    }
  ]
}
```

```

        {
            "name": "Entity Identifier Code",
            "value": "N101"
        },
        {
            "name": "Optional 5",
            "value": "N102"
        },
        {
            "name": "Identification Code Qualifier",
            "value": "N103"
        },
        {
            "name": "Identification Code",
            "value": "N104"
        }
    ]
}

```

### Output (JSON)

```

{
  N1: [ { 'Entity Identifier Code': 'ST',
          'Optional 5': '',
          'Identification Code Qualifier': '15',
          'Identification Code': '1553992' } ],
}

```

#### 5.1.1.1.8. PO1

This section describes how PO1 an EDI ID in the EDI format is converted to JSON with the help of the IO definition file.

#### Input (EDI)

```

PO1*1*1*EA*4.12*PE*EN*9780812508420~

```

#### Process (Definition File)

```

{
  "name": "PO1",
  "maxOccurrence": 2,
  "required": true,
  "container": true, //Helps you create a JSON array in the output.
  "children": [
    {
      "description": "Baseline Item Data",
      "required": true,
      "elements": [
        {
          "name": "PO1",
          "value": "PO1"
        },
        {

```

```

        "name": "Assigned Identification",
        "value": "PO101"
    },
    {
        "name": "Quantity Ordered",
        "value": "PO102"
    },
    {
        "name": "Unit or Basis for Measurement Code",
        "value": "PO103"
    },
    {
        "name": "Unit Price",
        "value": "PO104"
    },
    {
        "name": "Basis of Unit Price Code",
        "value": "PO105"
    },
    {
        "name": "Product/Service ID Qualifier",
        "value": "PO106"
    },
    {
        "name": "Product/Service ID",
        "value": "PO107"
    }
    ]
}

```

## Output (JSON)

```

{
  PO1:
  [ { 'Assigned Identification': '1',
      'Quantity Ordered': '1',
      'Unit or Basis for Measurement Code': 'EA',
      'Unit Price': '4.12',
      'Basis of Unit Price Code': 'PE',
      'Product/Service ID Qualifier': 'EN',
      'Product/Service ID': '9780812508420' } ],
}

```

## 6. ISA (JSON to EDI) | Generate

This section describes how `ISA` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

### Input (JSON)

```

{
  "Authorization Information Qualifier": "00",
  "Authorization Information": "",
  "Security Information Qualifier": "00",
}

```

```

"Security Information": "",
"Interchange ID Qualifier(ISA05)": "ZZ",
"Interchange Sender ID": "AMAZON",
"Interchange ID Qualifier(ISA07)": "ZZ",
"Interchange Receiver ID": "VENDOR",
"Interchange Date": "140731",
"Interchange Time": "0832",
"ISA11": "U",
"Interchange Control Version Number": "00400",
"Interchange Control Number": "000000112",
"Acknowledgment Requested": "0",
"Usage Indicator": "P",
"Component Element Separator": ">",
"Number of Included Functional Groups": "1",
"Interchange Control Number(IEA02)": "000000112"
}

```

## Process (Definition File)

```

{
  <General File Header comes here>

  "rules": [
    {
      "name": "ISA",
      "description": "Interchange Control Header",
      "maxOccurrence": 2,
      "required": true,
      "elements": [
        {
          "name": "name",
          "value": "ISA"
        },
        {
          "name": "Authorization Information Qualifier",
          "value": "{{[Authorization Information Qualifier]}}"
        },
        {
          "name": "Authorization Information",
          "value": "{{[Authorization Information]}}"
        },
        {
          "name": "Security Information Qualifier",
          "value": "{{[Security Information Qualifier]}}"
        },
        {
          "name": "Security Information",
          "value": "{{[Security Information]}}"
        },
        {
          "name": "Interchange ID Qualifier(ISA05)",
          "value": "{{[Interchange ID Qualifier(ISA05)]}}"
        },
        {
          "name": "Interchange Sender ID",
          "value": "{{[Interchange Sender ID]}}"
        },
        {
          "name": "Interchange ID Qualifier(ISA07)",
          "value": "{{[Interchange ID Qualifier(ISA07)]}}"
        }
      ]
    }
  ]
}

```



```

    },
    {
      "name": "Interchange Receiver ID",
      "value": "{{[Interchange Receiver ID]}}"
    },
    {
      "name": "Interchange Date",
      "value": "{{[Interchange Date]}}"
    },
    {
      "name": "Interchange Time",
      "value": "{{[Interchange Time]}}"
    },
    {
      "name": "ISA11",
      "value": "{{[ISA11]}}"
    },
    {
      "name": "Interchange Control Version Number",
      "value": "{{[Interchange Control Version Number]}}"
    },
    {
      "name": "Interchange Control Number",
      "value": "{{[Interchange Control Number]}}"
    },
    {
      "name": "Acknowledgment Requested",
      "value": "{{[Acknowledgment Requested]}}"
    },
    {
      "name": "Usage Indicator",
      "value": "{{[Usage Indicator]}}"
    },
    {
      "name": "Component Element Separator",
      "value": "{{[Component Element Separator]}}"
    }
  ],
  "children": [GS],
  "closeRule": {
    "name": "IEA",
    "description": "Interchange Control Trailer",
    "required": true,
    "maxOccurrence": 1,
    "elements": [
      {
        "name": "name",
        "value": "IEA"
      },
      {
        "name": "Number of Included Functional Groups",
        "value": "{{[Number of Included Functional Groups]}}"
      },
      {
        "name": "Interchange Control Number(IEA02)",
        "value": "{{[Interchange Control Number(IEA02)]}}"
      }
    ]
  }
}
]
}
}

```

## Output (EDI)

```
ISA*00**00**ZZ*AMAZON*ZZ*VENDOR*140731*0832*U*00400*000000112*0*P*>~  
IEA*1*000000112~
```

## 6.1. GS

This section describes how GS an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

### Input (JSON)

```
{  
  "Functional Identifier Code": "PO",  
  "Application Sender's Code": "AMAZON",  
  "Application Receiver's Code": "VENDOR",  
  "Date": "20140801",  
  "Time": "1400",  
  "Group Control Number": "367",  
  "Responsible Agency Code": "X",  
  "Version / Release / Industry Identifier Code": "004010",  
  "Number of Transaction Sets Included": "1",  
  "Group Control Number(GE02)": "112"  
}
```

### Process (Definition File)

```
{  
  "name": "GS",  
  "description": "Functional Group Header",  
  "required": true,  
  "maxOccurrence": 1,  
  "elements": [  
    {  
      "name": "name",  
      "value": "GS"  
    },  
    {  
      "name": "Functional Identifier Code",  
      "value": "{{[Functional Identifier Code]}}"  
    },  
    {  
      "name": "Application Sender's Code",  
      "value": "{{[Application Sender's Code]}}"  
    },  
    {  
      "name": "Application Receiver's Code",  
      "value": "{{[Application Receiver's Code]}}"  
    },  
    {  
      "name": "Date",  
      "value": "{{Date}}"  
    },  
    {  
      "name": "Time",  
      "value": "{{Time}}"  
    }  
  ]  
}
```

```

    },
    {
      "name": "Group Control Number",
      "value": "{{[Group Control Number]}}"
    },
    {
      "name": "Responsible Agency Code",
      "value": "{{[Responsible Agency Code]}}"
    },
    {
      "name": "Version / Release / Industry Identifier Code",
      "value": "{{[Version / Release / Industry Identifier Code]}}"
    }
  ],
  "children": [ST],
  "closeRule": {
    "name": "GE",
    "description": "Functional Group Trailer",
    "maxOccurrence": 1,
    "required": true,
    "elements": [
      {
        "name": "name",
        "value": "GE"
      },
      {
        "name": "Number of Transaction Sets Included",
        "value": "{{[Number of Transaction Sets Included]}}"
      },
      {
        "name": "Group Control Number(GE02)",
        "value": "{{[Group Control Number(GE02)]}}"
      }
    ]
  }
}

```

## Output (EDI)

```

GS*PO*AMAZON*VENDOR*20140801*1400*367*X*004010~
GE*1*112~

```

### 6.1.1. ST

This section describes how `ST` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```

{
  "Transaction Set Identifier Code": "850",
  "Transaction Set Control Number": "0001",
  "Number of Included Segments": "16",
  "Transaction Set Control Number(SE02)": "0001"
}

```

## Process (Definition File)

```
{
  "name": "ST",
  "description": "Transaction Set Header",
  "maxOccurrence": 1,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "ST"
    },
    {
      "name": "Transaction Set Identifier Code",
      "value": "{{[Transaction Set Identifier Code]}}"
    },
    {
      "name": "Transaction Set Control Number",
      "value": "{{[Transaction Set Control Number]}}"
    }
  ],
  "children": [BEG],
  "closeRule": {
    "name": "SE",
    "description": "Transaction Set Trailer",
    "maxOccurrence": 1,
    "required": true,
    "elements": [
      {
        "name": "name",
        "value": "SE"
      },
      {
        "name": "Number of Included Segments",
        "value": "{{[Number of Included Segments]}}"
      },
      {
        "name": "Transaction Set Control Number(SE02)",
        "value": "{{[Transaction Set Control Number(SE02)]}}"
      }
    ]
  }
}
```

## Output (EDI)

```
ST*850*0001~
SE*16*0001~
```

### 6.1.1.1. BEG

This section describes how BEG an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```
{
  "Transaction Set Purpose Code": "00",
  "Purchase Order Type Code": "NE",
```

```

"Purchase Order Number": "G7984251",
"BEG04": "",
"PO Date": "20070730",
"Number of Line Items": "5",
"Hash Total": "60"
}

```

## Process (Definition File)

```

{
  "name": "BEG",
  "description": "Beginning Segment for Purchase Order",
  "maxOccurrence": 1,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "BEG"
    },
    {
      "name": "Transaction Set Purpose Code",
      "value": "{{[Transaction Set Purpose Code]}}"
    },
    {
      "name": "Purchase Order Type Code",
      "value": "{{[Purchase Order Type Code]}}"
    },
    {
      "name": "Purchase Order Number",
      "value": "{{[Purchase Order Number]}}"
    },
    {
      "name": "BEG04",
      "value": "{{[BEG04]}}"
    },
    {
      "name": "PO Date",
      "value": "{{[PO Date]}}"
    }
  ],
  "children": [REF, FOB, CSH, DTM, PKG, N9, N1, PO1],
  "closeRule": {
    "name": "CTT",
    "description": "Transaction Totals",
    "maxOccurrence": 1,
    "required": true,
    "elements": [
      {
        "name": "name",
        "value": "CTT"
      },
      {
        "name": "Number of Line Items",
        "value": "{{[Number of Line Items]}}"
      },
      {
        "name": "Hash Total",
        "value": "{{[Hash Total]}}"
      }
    ]
  }
}

```

## Output (EDI)

```
BEG*00*NE*G7984251**20070730~  
CTT*5*60~
```

### 6.1.1.1.1. REF

This section describes how REF an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```
"REF": [  
  {  
    "Reference Identification Qualifier": "CR",  
    "Reference Identification": "BTS"  
  },  
  {  
    "Reference Identification Qualifier": "PD",  
    "Reference Identification": "BTS"  
  }  
]
```

## Process (Definition File)

```
{  
  "name": "BEG",  
  "description": "Beginning Segment for Purchase Order",  
  "maxOccurrence": 1,  
  "required": true,  
  "elements": [  
    {  
      "name": "name",  
      "value": "BEG"  
    },  
    {  
      "name": "Transaction Set Purpose Code",  
      "value": "{{[Transaction Set Purpose Code]}}"  
    },  
    {  
      "name": "Purchase Order Type Code",  
      "value": "{{[Purchase Order Type Code]}}"  
    },  
    {  
      "name": "Purchase Order Number",  
      "value": "{{[Purchase Order Number]}}"  
    },  
    {  
      "name": "BEG04",  
      "value": "[BEG04]"  
    },  
    {  
      "name": "PO Date",  
      "value": "[PO Date]"  
    }  
  ],  
  "children": [],  
  "closeRule": {
```

```

    "name": "CTT",
    "description": "Transaction Totals",
    "maxOccurrence": 1,
    "required": true,
    "elements": [
      {
        "name": "name",
        "value": "CTT"
      },
      {
        "name": "Number of Line Items",
        "value": "{{[Number of Line Items]}}"
      },
      {
        "name": "Hash Total",
        "value": "{{[Hash Total]}}"
      }
    ]
  }
}

```

### Output (EDI)

```

REF*CR*BTS~
REF*PD*BTS~

```

#### 6.1.1.1.2. FOB

This section describes how `FOB` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

### Input (JSON)

```

"FOB": [
  {
    "Shipment Method of Payment": "BP",
    "FOB02": "",
    "FOB03": "",
    "Transportation Terms Qualifier Code": "01",
    "Transportation Terms Code": "CFR",
    "Location Qualifier": "OV",
    "Description": "YANTIAN, CHINA"
  },
  {
    "Shipment Method of Payment": "BP",
    "FOB02": "",
    "Transportation Terms Qualifier Code": "01",
    "Transportation Terms Code": "CFR",
    "Location Qualifier": "OV",
    "Description": "YANTIAN, CHINA 121"
  }
]

```

### Process (Definition File)

```

{
  "name": "FOB",
  "description": "Free On Board Related Instructions",

```

```

    "relativeDataPath": "FOB",
    "maxOccurrence": 50,
    "required": true,
    "elements": [
      {
        "name": "name",
        "value": "FOB"
      },
      {
        "name": "Shipment Method of Payment",
        "value": "{{[Shipment Method of Payment]}}"
      },
      {
        "name": "FOB02",
        "value": "{{[FOB02]}}"
      },
      {
        "name": "FOB03",
        "value": "{{[FOB03]}}"
      },
      {
        "name": "Transportation Terms Qualifier Code",
        "value": "{{[Transportation Terms Qualifier Code]}}"
      },
      {
        "name": "Transportation Terms Code",
        "value": "{{[Transportation Terms Code]}}"
      },
      {
        "name": "Location Qualifier",
        "value": "{{[Location Qualifier]}}"
      },
      {
        "name": "Description",
        "value": "{{[Description]}}"
      }
    ]
  }
}

```

## Output (EDI)

```

FOB*BP***01*CFR*OV*YANTIAN, CHINA~
FOB*BP***01*CFR*OV*YANTIAN, CHINA 121~

```

### 6.1.1.1.3. CSH

This section describes how CSH an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```

"CSH": [
  {
    "Sales Requirement Code": "N"
  }
]

```



## Process (Definition File)

```
{
  "name": "CSH",
  "description": "Sales Requirements",
  "relativeDataPath": "CSH",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "CSH"
    },
    {
      "name": "Sales Requirement Code",
      "value": "{{[Sales Requirement Code]}}"
    }
  ]
}
```

## Output (EDI)

```
CSH*N~
```

### 6.1.1.1.4. DTM

This section describes how *DTM* an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```
"DTM": [
  {
    "Date/Time Qualifier": "063",
    "Date": "20140731"
  },
  {
    "Date/Time Qualifier": "064",
    "Date": "20140725"
  }
]
```

## Process (Definition File)

```
{
  "name": "DTM",
  "description": "Date/Time Reference",
  "relativeDataPath": "DTM",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "DTM"
    },
    {
      "name": "Date/Time Qualifier",
      "value": "{{[Date/Time Qualifier]}}"
    }
  ]
}
```

```

    },
    {
      "name": "Date",
      "value": "{{[Date]}}"
    }
  ]
}

```

## Output (EDI)

```

DTM*063*20140731~
DTM*064*20140725~

```

### 6.1.1.1.5. PKG

This section describes how `PKG` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```

"PKG": [
  {
    "Item Description Type": "F",
    "Packaging Characteristic Code": "CS",
    "Agency Qualifier Code": "",
    "Packaging Description Code": "",
    "Description": "40HC"
  }
]

```

## Process (Definition File)

```

{
  "name": "PKG",
  "description": "Marking, Packaging, Loading",
  "relativeDataPath": "PKG",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "PKG"
    },
    {
      "name": "Item Description Type",
      "value": "{{[Item Description Type]}}"
    },
    {
      "name": "Packaging Characteristic Code",
      "value": "{{[Packaging Characteristic Code]}}"
    },
    {
      "name": "Agency Qualifier Code",
      "value": "{{[Agency Qualifier Code]}}"
    },
    {
      "name": "Packaging Description Code",
      "value": "{{[Packaging Description Code]}}"
    }
  ]
}

```

```

    },
    {
      "name": "Description",
      "value": "{{[Description]}}"
    }
  ]
}

```

## Output (EDI)

```

PKG*F*CS***40HC~

```

### 6.1.1.1.6. N9

This section describes how N9 an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```

"N9": [
  {
    "Reference Identification Qualifier": "L1",
    "Reference Identification": "SPECIAL INSTRUCTIONS",
    "MSG": []
  }
]

```

## Process (Definition File)

```

{
  "name": "N9",
  "description": "Reference Identification",
  "relativeDataPath": "N9",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "N9"
    },
    {
      "name": "Reference Identification Qualifier",
      "value": "{{[Reference Identification Qualifier]}}"
    },
    {
      "name": "Reference Identification",
      "value": "{{[Reference Identification]}}"
    }
  ],
  "children": [MSG]
}

```

## Output (EDI)

```

N9*L1*SPECIAL INSTRUCTIONS~

```

#### 6.1.1.1.6.1. MSG

This section describes how `MSG` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

#### Input (JSON)

```
"MSG": [{
  "Free-Form Message Text": "PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE
FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS
FC DESTINATION.PAYMENT TERMS ARE PER CONTAINER."
}]
```

#### Process (Definition File)

```
{
  "name": "MSG",
  "description": "Message Text",
  "relativeDataPath": "MSG",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "MSG"
    },
    {
      "name": "Free-Form Message Text",
      "value": "[Free-Form Message Text]"
    }
  ]
}
```

#### Output (EDI)

```
MSG*PREFERENCE IS PALLET-LOAD, BUT IF CONTAINERS ARE FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR
STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS FC DESTINATION.PAYMENT TERMS ARE PER
CONTAINER.~
```

#### 6.1.1.1.7. N1

This section describes how `N1` an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

#### Input (JSON)

```
"N1": [
  {
    "Entity Identifier Code": "ST",
    "Name": "",
    "Identification Code Qualifier": "15",
    "Identification Code": "1553992"
  }
]
```

## Process (Definition File)

```
{
  "name": "N1",
  "description": "Name",
  "maxOccurrence": 50,
  "relativeDataPath": "N1",
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "N1"
    },
    {
      "name": "Entity Identifier Code",
      "value": "{{[Entity Identifier Code]}}"
    },
    {
      "name": "Name",
      "value": "{{[Name]}}"
    },
    {
      "name": "Identification Code Qualifier",
      "value": "{{[Identification Code Qualifier]}}"
    },
    {
      "name": "Identification Code",
      "value": "{{[Identification Code]}}"
    }
  ]
}
```

## Output (EDI)

```
N1*ST**15*1553992~
```

### 6.1.1.1.8. PO1

This section describes how PO1 an EDI ID in the JSON format is converted to the EDI format with the help of the IO definition file.

## Input (JSON)

```
"PO1": [
  {
    "Assigned Identification": "1",
    "Quantity Ordered": "1",
    "Unit or Basis for Measurement Code": "EA",
    "Unit Price": "4.12",
    "Basis of Unit Price Code": "PE",
    "Product/Service ID Qualifier": "EN",
    "Product/Service ID": "9780812508420"
  }
]
```

## Process (Definition File)

```
{
```

```

"name": "PO1",
"maxOccurrence": 50,
"relativeDataPath": "PO1",
"required": true,
"elements": [
  {
    "name": "name",
    "value": "PO1"
  },
  {
    "name": "Assigned Identification",
    "value": "{{[Assigned Identification]}}"
  },
  {
    "name": "Quantity Ordered",
    "value": "{{[Quantity Ordered]}}"
  },
  {
    "name": "Unit or Basis for Measurement Code",
    "value": "{{[Unit or Basis for Measurement Code]}}"
  },
  {
    "name": "Unit Price",
    "value": "{{[Unit Price]}}"
  },
  {
    "name": "Basis of Unit Price Code",
    "value": "{{[Basis of Unit Price Code]}}"
  },
  {
    "name": "Product/Service ID Qualifier",
    "value": "{{[Product/Service ID Qualifier]}}"
  },
  {
    "name": "Product/Service ID",
    "value": "{{[Product/Service ID]}}"
  }
]
}

```

### Output ( EDI)

```
PO1*1*1*EA*4.12*PE*EN*9780812508420~
```

## 7. Fixed-Width to JSON | Generate

This section describes how an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

### 7.1. 6.1. 000

This section describes how 000 an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

#### Input (Fixed-Width)

```
000A20/07/2010      US010
```

**Process (Definition File)**

```
"rules": [
{
"name": "ASN",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"required": true,
"elements": [
{
"name": "000",
"value": "000",
"length": 3
},
{
"name": "Audit Action(000)",
"length": 1
},
{
"name": "ASN Receipt Date",
"length": 10
},
{
"name": "ASN Processing Status(000)",
"length": 1
},
{
"name": "Order Management Business Unit",
"length": 5
},
{
"name": "Inventory Unit",
"length": 5
},
{
"name": "Carrier ID",
"length": 10
},
{
"name": "Customer Id",
"length": 15
},
{
"name": "Device ID",
"length": 10
},
{
"name": "Device Date Time",
"length": 31
},
{
"name": "EC Primary Event Code",
"length": 6
},
{
"name": "EC Secondary Event Code",
"length": 10
}
],
}
],
}
```

```

    {
      "name": "Message Number(000)",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Process Instance(000)",
      "length": 10,
      "direction": "right"
    },
    {
      "name": "Ship Via Code",
      "length": 10
    },
    {
      "name": "Shipment Number(000)",
      "length": 30
    },
    {
      "name": "ShipTo SetID(000)",
      "length": 5
    },
    {
      "name": "ShipTo Location(000)",
      "length": 10
    },
    {
      "name": "Supplier ID(000)",
      "length": 10,
      "direction": "right"
    },
    {
      "name": "Supplier SetID(000)",
      "length": 5
    },
    {
      "name": "Supplier Location",
      "length": 10
    },
    {
      "name": "Supplier UPN Ready",
      "length": 1
    },
    {
      "name": "Supplier ID Number(000)",
      "length": 13
    },
    {
      "name": "ShipTo GLN(000)",
      "length": 13
    }
  ]
}
]
]
]

```

## Output

```

{
  ASN:[{
    'Audit Action(000)': 'A',
    'ASN Receipt Date': '20/07/2010',

```



```

'ASN Processing Status(000)': '',
'Order Management Business Unit': '',
'Inventory Unit': 'US010',
'Carrier ID': '',
'Customer Id': '',
'Device ID': '',
'Device Date Time': '',
'EC Primary Event Code': '',
'EC Secondary Event Code': '',
'Message Number(000)': '0',
'Process Instance(000)': '0',
'Ship Via Code': '',
'Shipment Number(000)': 'PE-Recv Test2007H',
'ShipTo SetID(000)': '',
'ShipTo Location(000)': '',
'Supplier ID(000)': '',
'Supplier SetID(000)': '',
'Supplier Location': '',
'Supplier UPN Ready': undefined,
'Supplier ID Number(000)': undefined,
'ShipTo GLN(000)': undefined
}}
}

```

## 7.2. 6.2. 100

This section describes how 100 an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

### Input (Fixed-Width)

100A	1PE-Recv Test2007H	20/07/2010
US0010000000235		

### Process (Definition File)

```

"rules": [
{
"name": "ASN",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"name": "100",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"required": true,
"elements": [
{
"name": "100",
"value": "100",
"length": 3
},
{

```

```

        "name": "Audit Action(100)",
        "length": 1
    },
    {
        "name": "ASN Sequence number(100)",
        "length": 9,
        "direction": "right"
    },
    {
        "name": "Shipment Number(100)",
        "length": 30
    },
    {
        "name": "Order Number(100)",
        "length": 10
    },
    {
        "name": "Bill Of Landing(100)",
        "length": 30
    },
    {
        "name": "Item Shipping Date",
        "length": 10
    },
    {
        "name": "Pro Number(100)",
        "length": 40,
        "direction": "right"
    },
    {
        "name": "PO Business Unit(100)",
        "length": 5
    },
    {
        "name": "PO Number(100)",
        "length": 10
    }
}
]
}
]
]
]

```

## Output (JSON)

```

{ ASN:
[ { '100':
[ { 'Audit Action(100)': 'A',
    'ASN Sequence number(100)': '1',
    'Shipment Number(100)': 'PE-Recv Test2007H',
    'Order Number(100)': '',
    'Bill Of Landing(100)': '',
    'Item Shipping Date': '20/07/2010',
    'Pro Number(100)': 'US00100000',
    'PO Business Unit(100)': '00235',
    'PO Number(100)': undefined } ]
}]
}

```

### 7.3. 6.3. 110

This section describes how 110 an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

#### Input (Fixed-Width)

110A	1PE-Recv Test2007H	0	0	0
------	--------------------	---	---	---

#### Process (Definition File)

```
"rules": [
{
  "name": "ASN",
  "container": true,
  "required": true,
  "maxOccurrence": 50,
  "children": [
    {
      "name": "110",
      "container": true,
      "required": true,
      "maxOccurrence": 50,
      "children": [
        {
          "required": true,
          "elements": [
            {
              "name": "110",
              "value": "110",
              "length": 3
            },
            {
              "name": "Audit Action(110)",
              "length": 1
            },
            {
              "name": "ASN Sequence number(110)",
              "length": 9,
              "direction": "right"
            },
            {
              "name": "Shipment Number(110)",
              "length": 30
            },
            {
              "name": "Order Number(110)",
              "length": 10
            },
            {
              "name": "Order Line(110)",
              "length": 5,
              "direction": "right"
            },
            {
              "name": "Schedule Line Number",
              "length": 7,
              "direction": "right"
            },
            {
              "name": "Demand Line Number(110)",
```

```

        "length": 4,
        "direction": "right"
    },
    {
        "name": "Customer Address type",
        "length": 1
    },
    {
        "name": "ShipTo Name",
        "length": 40
    },
    {
        "name": "ShipTo Name2",
        "length": 40
    },
    {
        "name": "Address Line 1",
        "length": 35
    },
    {
        "name": "Address Line 2",
        "length": 35
    },
    {
        "name": "Address Line 3",
        "length": 35
    },
    {
        "name": "Address Line 4",
        "length": 35
    },
    {
        "name": "City",
        "length": 30
    },
    {
        "name": "In City Limit",
        "length": 1
    },
    {
        "name": "Country1",
        "length": 30
    },
    {
        "name": "State",
        "length": 6
    },
    {
        "name": "Country2",
        "length": 3
    },
    {
        "name": "Postal Code",
        "length": 12
    }
}
]
}
]
}
]
]

```

## Output

```

{ASN:
[ {
'110':
[ { 'Audit Action(110)': 'A',
'ASN Sequence number(110)': '1',
'Shipment Number(110)': 'PE-Recv Test2007H',
'Order Number(110)': '',
'Order Line(110)': '0',
'Schedule Line Number': '0',
'Demand Line Number(110)': '0',
'Customer Address type': '',
'ShipTo Name': '',
'ShipTo Name2': '',
'Address Line 1': '',
'Address Line 2': '',
'Address Line 3': '',
'Address Line 4': '',
City: '',
'In City Limit': '',
Country1: '',
State: '',
Country2: '',
'Postal Code': '' } ]
}]
}

```

## 7.4. 6.4. 200

This section describes how 200 an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

### Input (Fixed-Width)

```

200A      1      1      PE-Recv Test2007H      0 0.00
0
0.0000   US0010000000235  1 1      0 0
0.00000000
0 0      EA

```

### Process (Definition File)

```

"rules": [
{
"name": "ASN",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"name": "200",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"required": true,
"elements": [
{
"name": "200",

```

```

    "value": "200",
    "length": 3
  },
  {
    "name": "Audit Action(200)",
    "length": 1
  },
  {
    "name": "ASN Sequence number(200)",
    "length": 9,
    "direction": "right"
  },
  {
    "name": "ASN Schedule Sequence Number(200)",
    "length": 9,
    "direction": "right"
  },
  {
    "name": "Container ID",
    "length": 10
  },
  {
    "name": "Shipment Number(200)",
    "length": 30
  },
  {
    "name": "Order Number(200)",
    "length": 10
  },
  {
    "name": "Order Line(200)",
    "length": 5,
    "direction": "right"
  },
  {
    "name": "Sched Line NO(200)",
    "length": 7,
    "direction": "right"
  },
  {
    "name": "ASN Item ID(200)",
    "length": 18
  },
  {
    "name": "Demand Line Number(200)",
    "length": 4,
    "direction": "right"
  },
  {
    "name": "ASN Item Description",
    "length": 30
  },
  {
    "name": "Quantity Shipped(200)",
    "length": 17,
    "direction": "right"
  },
  {
    "name": "Shipment Unit of Measure(200)",
    "length": 3
  },
  {
    "name": "Customer Item Number",
    "length": 18
  },
  },

```

```

    {
      "name": "Parent Product ID",
      "length": 18
    },
    {
      "name": "Parent Product Description",
      "length": 30
    },
    {
      "name": "Quantity",
      "length": 10,
      "direction": "right"
    },
    {
      "name": "Quantity Code",
      "length": 3
    },
    {
      "name": "PO Business Unit(200)",
      "length": 5
    },
    {
      "name": "PO Number(200)",
      "length": 10
    },
    {
      "name": "Line Number(200)",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Schedule Number(200)",
      "length": 3,
      "direction": "right"
    },
    {
      "name": "Bill Of Landing(200)",
      "length": 30
    },
    {
      "name": "Process Instance(200)",
      "length": 10,
      "direction": "right"
    },
    {
      "name": "ASN Default Key",
      "length": 1
    },
    {
      "name": "ASN Processing Status(200)",
      "length": 1
    },
    {
      "name": "Message Number(200)",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Receiving Business Unit",
      "length": 5
    },
    {
      "name": "Conversion rate",
      "length": 17,
      "direction": "right"
    }
  
```

```

    },
    {
      "name": "Currency Code",
      "length": 3
    },
    {
      "name": "ERS Action",
      "length": 1
    },
    {
      "name": "Item ID",
      "length": 18
    },
    {
      "name": "More Information",
      "length": 100
    },
    {
      "name": "Item SetID",
      "length": 5
    },
    {
      "name": "Lot Control",
      "length": 1
    },
    {
      "name": "Receipt Number",
      "length": 10
    },
    {
      "name": "Receipt Line",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Receiver Shipping Sequence",
      "length": 3,
      "direction": "right"
    },
    {
      "name": "Serial Control",
      "length": 1
    },
    {
      "name": "ShipTo Location(200)",
      "length": 10
    },
    {
      "name": "ShipTo SetID(200)",
      "length": 5
    },
    {
      "name": "Standard Unit of Measure",
      "length": 3
    },
    {
      "name": "Unit of Measure",
      "length": 3
    },
    {
      "name": "Supplier ID(200)",
      "length": 10
    },
    {
      "name": "Supplier SetID(200)",

```



```

        "length": 5
    },
    {
        "name": "Revision Override",
        "length": 1
    },
    {
        "name": "Revision(200)",
        "length": 4
    },
    {
        "name": "Pro Number(200)",
        "length": 40
    },
    {
        "name": "Supplier Contract ID",
        "length": 10
    },
    {
        "name": "UPN ID",
        "length": 13
    },
    {
        "name": "Supplier ID Number(200)",
        "length": 13
    },
    {
        "name": "ShipTo GLN(200)",
        "length": 13
    }
}
]
}
]
}
]
}
]
]

```

## Output (JSON)

```

{ ASN:
[ { '200':
[ { 'Audit Action(200)': 'A',
    'ASN Sequence number(200)': '1',
    'ASN Schedule Sequence Number(200)': '1',
    'Container ID': '',
    'Shipment Number(200)': 'PE-Recv Test2007H',
    'Order Number(200)': '',
    'Order Line(200)': '0',
    'Sched Line NO(200)': '0.00',
    'ASN Item ID(200)': '',
    'Demand Line Number(200)': '0',
    'ASN Item Description': '',
    'Quantity Shipped(200)': '0002.0000',
    'Shipment Unit of Measure(200)': 'EA',
    'Customer Item Number': '',
    'Parent Product ID': '',
    'Parent Product Description': '',
    Quantity: '0.0000',
    'Quantity Code': '',
    'PO Business Unit(200)': 'US001',
    'PO Number(200)': '0000000235',
    'Line Number(200)': '1',

```

```

'Schedule Number(200)': '1',
'Bill Of Landing(200)': '',
'Process Instance(200)': '0',
'ASN Default Key': '',
'ASN Processing Status(200)': '',
'Message Number(200)': '0',
'Receiving Business Unit': '',
'Conversion rate': '0.00000000',
'Currency Code': '',
'ERS Action': '',
'Item ID': '',
'More Information': '',
'Item SetID': '',
'Lot Control': '',
'Receipt Number': '',
'Receipt Line': '',
'Receiver Shipping Sequence': '',
'Serial Control': '',
'ShipTo Location(200)': '',
'ShipTo SetID(200)': '',
'Standard Unit of Measure': '',
'Unit of Measure': '',
'Supplier ID(200)': '',
'Supplier SetID(200)': '',
'Revision Override': '',
'Revision(200)': '',
'Pro Number(200)': '',
'Supplier Contract ID': '',
'UPN ID': '',
'Supplier ID Number(200)': '',
'ShipTo GLN(200)': '' } ]
}}
}

```

## 7.5. 6.5. 300

This section describes how 300 an element in the Fixed-Width format is converted to the JSON format with the help of the IO definition file.

### Input (Fixed-Width)

300A	1	1	0PE-Recv Test2007H	0		
0.00	0	0002.0000		US0010000000235	1	0

### Process (Definition File)

```

"rules": [
{
"name": "ASN",
"container": true,
"required": true,
"maxOccurrence": 50,
"children": [
{
"name": "300",
"container": true,
"required": true,
"maxOccurrence": 50,

```

```

"children": [
{
  "required": true,
  "elements": [
    {
      "name": "300",
      "value": "300",
      "length": 3
    },
    {
      "name": "Audit Action(300)",
      "length": 1
    },
    {
      "name": "ASN Sequence number(300)",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "ASN Schedule Sequence Number(300)",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "ASN Lot/Serial Number",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "Shipment Number(300)",
      "length": 30
    },
    {
      "name": "Order Number(300)",
      "length": 10
    },
    {
      "name": "Order Line(300)",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "ASN Item ID(300)",
      "length": 18
    },
    {
      "name": "Sched Line NO(300)",
      "length": 7,
      "direction": "right"
    },
    {
      "name": "Demand Line No",
      "length": 4,
      "direction": "right"
    },
    {
      "name": "Quantity Shipped(300)",
      "length": 17,
      "direction": "right"
    },
    {
      "name": "Shipment Unit of Measure(300)",
      "length": 3
    },
    {

```

```

        "name": "Lot Id",
        "length": 15
    },
    {
        "name": "Serial ID(300)",
        "length": 20
    },
    {
        "name": "PO Business Unit(300)",
        "length": 5
    },
    {
        "name": "PO Number(300)",
        "length": 10
    },
    {
        "name": "Line Number(300)",
        "length": 5,
        "direction": "right"
    },
    {
        "name": "Schedule Number(300)",
        "length": 3,
        "direction": "right"
    },
    {
        "name": "Supplier Lot",
        "length": 15
    },
    {
        "name": "Revision(300)",
        "length": 4
    }
}
]
}
]
}
]

```

## Output (JSON)

```

{ ASN:
[ { '300':
[ { 'Audit Action(300)': 'A',
    'ASN Sequence number(300)': '1',
    'ASN Schedule Sequence Number(300)': '1',
    'ASN Lot/Serial Number': '0',
    'Shipment Number(300)': 'PE-Recv Test2007H',
    'Order Number(300)': '',
    'Order Line(300)': '0',
    'ASN Item ID(300)': '',
    'Sched Line NO(300)': '0.00',
    'Demand Line No': '0',
    'Quantity Shipped(300)': '0002.0000',
    'Shipment Unit of Measure(300)': '',
    'Lot Id': '',
    'Serial ID(300)': '',
    'PO Business Unit(300)': 'US001',
    'PO Number(300)': '0000000235',
    'Line Number(300)': '1',
    'Schedule Number(300)': '0',

```

```
'Supplier Lot': '',
  'Revision(300)': '' } ]
}]
}
```

## 8. JSON to Fixed-Width | Parse

This section describes how a fixed-width element in the JSON format is converted to the fixed-width format with the help of the IO definition file.

### 8.1. 7.1. 000

This section describes how 000 a Fixed-Width element in the JSON format is converted to the Fixed-Width format with the help of the IO definition file.

#### Input (JSON)

```
[
{
"Audit Action(000)": "A",
"ASN Receipt Date": "20/07/2010",
"ASN Processing Status(000)": "",
"Order Management Business Unit": "",
"Inventory Unit": "US010",
"Carrier ID": "",
"Customer Id": "",
"Device ID": "",
"Device Date Time": "",
"EC Primary Event Code": "",
"EC Secondary Event Code": "",
"Message Number(000)": "0",
"Process Instance(000)": "0",
"Ship Via Code": "",
"Shipment Number(000)": "PE-Recv Test2007H",
"ShipTo SetID(000)": "",
"ShipTo Location(000)": "",
"Supplier ID(000)": "",
"Supplier SetID(000)": "",
"Supplier Location": "",
"Supplier UPN Ready": "u",
"Supplier ID Number(000)": "undefined",
"ShipTo GLN(000)": "undefined"
}
]
```

#### Process (Definition File)

```
rules:[{
  "name": "ASN",
  "required": true,
  "maxOccurrence": 50,
  "elements": [
    {
      "name": "000",
```

```

    "value": "000",
    "length": 3
  },
  {
    "name": "Audit Action(000)",
    "value": "{{[Audit Action(000)]}}",
    "length": 1
  },
  {
    "name": "ASN Receipt Date",
    "value": "{{[ASN Receipt Date]}}",
    "length": 10
  },
  {
    "name": "ASN Processing Status(000)",
    "value": "{{[ASN Processing Status(000)]}}",
    "length": 1
  },
  {
    "name": "Order Management Business Unit",
    "value": "{{[Order Management Business Unit]}}",
    "length": 5
  },
  {
    "name": "Inventory Unit",
    "value": "{{[Inventory Unit]}}",
    "length": 5
  },
  {
    "name": "Carrier ID",
    "value": "{{[Carrier ID]}}",
    "length": 10
  },
  {
    "name": "Customer Id",
    "value": "{{[Customer Id]}}",
    "length": 15
  },
  {
    "name": "Device ID",
    "value": "{{[Device ID]}}",
    "length": 10
  },
  {
    "name": "Device Date Time",
    "value": "{{[Device Date Time]}}",
    "length": 31
  },
  {
    "name": "EC Primary Event Code",
    "value": "{{[EC Primary Event Code]}}",
    "length": 6
  },
  {
    "name": "EC Secondary Event Code",
    "value": "{{[EC Secondary Event Code]}}",
    "length": 10
  },
  {
    "name": "Message Number(000)",
    "value": "{{[Message Number(000)]}}",
    "length": 5,
    "direction": "right"
  },
  {

```

```

    "name": "Process Instance(000)",
    "value": "{{[[Process Instance(000)]}}",
    "length": 10,
    "direction": "right"
  },
  {
    "name": "Ship Via Code",
    "value": "{{[[Ship Via Code]]}",
    "length": 10
  },
  {
    "name": "Shipment Number(000)",
    "value": "{{[[Shipment Number(000)]}}",
    "length": 30
  },
  {
    "name": "ShipTo SetID(000)",
    "value": "{{[[ShipTo SetID(000)]}}",
    "length": 5
  },
  {
    "name": "ShipTo Location(000)",
    "value": "{{[[ShipTo Location(000)]}}",
    "length": 10
  },
  {
    "name": "Supplier ID(000)",
    "value": "{{[[Supplier ID(000)]}}",
    "length": 10,
    "direction": "right"
  },
  {
    "name": "Supplier SetID(000)",
    "value": "{{[[Supplier SetID(000)]}}",
    "length": 5
  },
  {
    "name": "Supplier Location",
    "value": "{{[[Supplier Location]]}",
    "length": 10
  },
  {
    "name": "Supplier UPN Ready",
    "value": "{{[[Supplier UPN Ready]]}",
    "length": 1
  },
  {
    "name": "Supplier ID Number(000)",
    "value": "{{[[Supplier ID Number(000)]}}",
    "length": 13
  },
  {
    "name": "ShipTo GLN(000)",
    "value": "{{[[ShipTo GLN(000)]}}",
    "length": 13
  }
]
}]

```

### Output (Fixed-Width)

```

000A20/07/2010      US010
0          0          PE-Recv Test2007H

```

## 8.2. 7.2. 100

This section describes how 100 a Fixed-Width element in the JSON format is converted to the Fixed-Width format with the help of the IO definition file.

### Input (JSON)

```
[
{
"100": [
{
"Audit Action(100)": "A",
"ASN Sequence number(100)": "1",
"Shipment Number(100)": "PE-Recv Test2007H",
"Order Number(100)": "",
"Bill Of Landing(100)": "",
"Item Shipping Date": "20/07/2010",
"Pro Number(100)": "US00100000",
"PO Business Unit(100)": "00235",
"PO Number(100)": ""
}
]
}
]
```

### Process (Definition File)

```
children":
[
{
"maxOccurrence": 50,
"relativeDataPath": "100",
"required": true,
"elements": [
{
"name": "100",
"value": "100",
"length": 3
},
{
"name": "Audit Action(100)",
"value": "{{[Audit Action(100)]}}",
"length": 1
},
{
"name": "ASN Sequence number(100)",
"value": "{{[ASN Sequence number(100)]}}",
"length": 9,
"direction": "right"
},
{
"name": "Shipment Number(100)",
"value": "{{[Shipment Number(100)]}}",
"length": 30
},
{
"name": "Order Number(100)",
"value": "{{[Order Number(100)]}}",
"length": 10
},
{
"name": "Bill Of Landing(100)",
```



```

        "value": "[{{[Bill Of Landing(100)]}}]",
        "length": 30
    },
    {
        "name": "Item Shipping Date",
        "value": "[{{[Item Shipping Date]}}]",
        "length": 10
    },
    {
        "name": "Pro Number(100)",
        "value": "[{{[Pro Number(100)]}}]",
        "length": 40,
        "direction": "right"
    },
    {
        "name": "PO Business Unit(100)",
        "value": "[{{[PO Business Unit(100)]}}]",
        "length": 5
    },
    {
        "name": "PO Number(100)",
        "value": "[{{[PO Number(100)]}}]",
        "length": 10
    }
]
}
]

```

### Output (Fixed-Width)

100A	1PE-Recv Test2007H	20/07/2010
US0010000000235		

## 8.3. 7.3. 110

This section describes how 110 a Fixed-Width element in the JSON format is converted to the Fixed-Width format with the help of the IO definition file.

### Input (JSON)

```

[
{
"110": [
{
"Audit Action(110)": "A",
"ASN Sequence number(110)": "1",
"Shipment Number(110)": "PE-Recv Test2007H",
"Order Number(110)": "",
"Order Line(110)": "0",
"Schedule Line Number": "0",
"Demand Line Number(110)": "0",
"Customer Address type": "",
"ShipTo Name": "",
"ShipTo Name2": "",
"Address Line 1": "",
"Address Line 2": "",
"Address Line 3": "",
"Address Line 4": "",
"City": "",

```

```

    "In City Limit": "",
    "Country1": "",
    "State": "",
    "Country2": "",
    "Postal Code": ""
  }
]
}
]

```

## Process (Definition File)

```

children":
[
{
  "maxOccurrence": 50,
  "relativeDataPath": "110",
  "required": true,
  "elements": [
    {
      "name": "110",
      "value": "110",
      "length": 3
    },
    {
      "name": "Audit Action(110)",
      "value": "{{[Audit Action(110)]}}",
      "length": 1
    },
    {
      "name": "ASN Sequence number(110)",
      "value": "{{[ASN Sequence number(110)]}}",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "Shipment Number(110)",
      "value": "{{[Shipment Number(110)]}}",
      "length": 30
    },
    {
      "name": "Order Number(110)",
      "value": "{{[Order Number(110)]}}",
      "length": 10
    },
    {
      "name": "Order Line(110)",
      "value": "{{[Order Line(110)]}}",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Schedule Line Number",
      "value": "{{[Schedule Line Number]}}",
      "length": 7,
      "direction": "right"
    },
    {
      "name": "Demand Line Number(110)",
      "value": "{{[Demand Line Number(110)]}}",
      "length": 4,
      "direction": "right"
    }
  ],
}
]

```

```

    {
      "name": "Customer Address type",
      "value": "{{[Customer Address type]}}",
      "length": 1
    },
    {
      "name": "ShipTo Name",
      "value": "{{[ShipTo Name]}}",
      "length": 40
    },
    {
      "name": "ShipTo Name2",
      "value": "{{[ShipTo Name2]}}",
      "length": 40
    },
    {
      "name": "Address Line 1",
      "value": "{{[Address Line 1]}}",
      "length": 35
    },
    {
      "name": "Address Line 2",
      "value": "{{[Address Line 2]}}",
      "length": 35
    },
    {
      "name": "Address Line 3",
      "value": "{{[Address Line 3]}}",
      "length": 35
    },
    {
      "name": "Address Line 4",
      "value": "{{[Address Line 4]}}",
      "length": 35
    },
    {
      "name": "City",
      "value": "{{[City]}}",
      "length": 30
    },
    {
      "name": "In City Limit",
      "value": "{{[In City Limit]}}",
      "length": 1
    },
    {
      "name": "Country1",
      "value": "{{[Country1]}}",
      "length": 30
    },
    {
      "name": "State",
      "value": "{{[State]}}",
      "length": 6
    },
    {
      "name": "Country2",
      "value": "{{[Country2]}}",
      "length": 3
    },
    {
      "name": "Postal Code",
      "value": "{{[Postal Code]}}",
      "length": 12
    }
  }

```

```
]
}
]
```

## Output (Fixed-Width)

```
110A          1PE-Recv Test2007H          0    0    0
```

## 8.4. 7.4. 200

This section describes how 200 a Fixed-Width element in the JSON format is converted to the Fixed-Width format with the help of the IO definition file.

### Input (JSON)

```
[
{
"200": [
{
"Audit Action(200)": "A",
"ASN Sequence number(200)": "1",
"ASN Schedule Sequence Number(200)": "1",
"Container ID": "",
"Shipment Number(200)": "PE-Recv Test2007H",
"Order Number(200)": "",
"Order Line(200)": "0",
"Sched Line NO(200)": "0.00",
"ASN Item ID(200)": "",
"Demand Line Number(200)": "0",
"ASN Item Description": "",
"Quantity Shipped(200)": "0002.0000",
"Shipment Unit of Measure(200)": "EA",
"Customer Item Number": "",
"Parent Product ID": "",
"Parent Product Description": "",
"Quantity": "0.0000",
"Quantity Code": "",
"PO Business Unit(200)": "US001",
"PO Number(200)": "0000000235",
"Line Number(200)": "1",
"Schedule Number(200)": "1",
"Bill Of Landing(200)": "",
"Process Instance(200)": "0",
"ASN Default Key": "",
"ASN Processing Status(200)": "",
"Message Number(200)": "0",
"Receiving Business Unit": "",
"Conversion rate": "0.00000000",
"Currency Code": "",
"ERS Action": "",
"Item ID": "",
"More Information": "",
"Item SetID": "",

```

```

    "Lot Control": "",
    "Receipt Number": "",
    "Receipt Line": "",
    "Receiver Shipping Sequence": "",
    "Serial Control": "",
    "ShipTo Location(200)": "",
    "ShipTo SetID(200)": "",
    "Standard Unit of Measure": "",
    "Unit of Measure": "",
    "Supplier ID(200)": "",
    "Supplier SetID(200)": "",
    "Revision Override": "",
    "Revision(200)": "",
    "Pro Number(200)": "",
    "Supplier Contract ID": "",
    "UPN ID": "",
    "Supplier ID Number(200)": "",
    "ShipTo GLN(200)": ""
  }
]
}
]

```

## Process (Definition File)

```

"children":
[
{
  "maxOccurrence": 50,
  "relativeDataPath": "200",
  "required": true,
  "elements": [
    {
      "name": "200",
      "value": "200",
      "length": 3
    },
    {
      "name": "Audit Action(200)",
      "value": "{{[Audit Action(200)]}}",
      "length": 1
    },
    {
      "name": "ASN Sequence number(200)",
      "value": "{{[ASN Sequence number(200)]}}",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "ASN Schedule Sequence Number(200)",
      "value": "{{[ASN Schedule Sequence Number(200)]}}",
      "length": 9,
      "direction": "right"
    },
    {
      "name": "Container ID",
      "value": "{{[Container ID]}}",
      "length": 10
    },
    {
      "name": "Shipment Number(200)",
      "value": "{{[Shipment Number(200)]}}",
      "length": 30
    }
  ]
}
]

```

```

},
{
  "name": "Order Number(200)",
  "value": "{{[Order Number(200)]}}",
  "length": 10
},
{
  "name": "Order Line(200)",
  "value": "{{[Order Line(200)]}}",
  "length": 5,
  "direction": "right"
},
{
  "name": "Sched Line NO(200)",
  "value": "{{toFixed [Sched Line NO(200)] \"2\"}}",
  "length": 7,
  "direction": "right"
},
{
  "name": "ASN Item ID(200)",
  "value": "{{[ASN Item ID(200)]}}",
  "length": 18
},
{
  "name": "Demand Line Number(200)",
  "value": "{{[Demand Line Number(200)]}}",
  "length": 4,
  "direction": "right"
},
{
  "name": "ASN Item Description",
  "value": "{{[ASN Item Description]}}",
  "length": 30
},
{
  "name": "Quantity Shipped(200)",
  "value": "{{toFixed [Quantity Shipped(200)] \"4\"}}",
  "length": 17,
  "direction": "right"
},
{
  "name": "Shipment Unit of Measure(200)",
  "value": "{{[Shipment Unit of Measure(200)]}}",
  "length": 3
},
{
  "name": "Customer Item Number",
  "value": "{{[Customer Item Number]}}",
  "length": 18
},
{
  "name": "Parent Product ID",
  "value": "{{[Parent Product ID]}}",
  "length": 18
},
{
  "name": "Parent Product Description",
  "value": "{{[Parent Product Description]}}",
  "length": 30
},
{
  "name": "Quantity",
  "value": "{{toFixed Quantity \"4\"}}",
  "length": 10,
  "direction": "right"
}

```

```

},
{
  "name": "Quantity Code",
  "value": "{{[Quantity Code]}}",
  "length": 3
},
{
  "name": "PO Business Unit(200)",
  "value": "{{[PO Business Unit(200)]}}",
  "length": 5
},
{
  "name": "PO Number(200)",
  "value": "{{[PO Number(200)]}}",
  "length": 10
},
{
  "name": "Line Number(200)",
  "value": "{{[Line Number(200)]}}",
  "length": 5,
  "direction": "right"
},
{
  "name": "Schedule Number(200)",
  "value": "{{[Schedule Number(200)]}}",
  "length": 3,
  "direction": "right"
},
{
  "name": "Bill Of Landing(200)",
  "value": "{{[Bill Of Landing(200)]}}",
  "length": 30
},
{
  "name": "Process Instance(200)",
  "value": "{{[Process Instance(200)]}}",
  "length": 10,
  "direction": "right"
},
{
  "name": "ASN Default Key",
  "value": "{{[ASN Default Key]}}",
  "length": 1
},
{
  "name": "ASN Processing Status(200)",
  "value": "{{[ASN Processing Status(200)]}}",
  "length": 1
},
{
  "name": "Message Number(200)",
  "value": "{{[Message Number(200)]}}",
  "length": 5,
  "direction": "right"
},
{
  "name": "Receiving Business Unit",
  "value": "{{[Receiving Business Unit]}}",
  "length": 5
},
{
  "name": "Conversion rate",
  "value": "{{toFixed [Conversion rate] \"8\"}}",
  "length": 17,
  "direction": "right"
}

```

```

},
{
  "name": "Currency Code",
  "value": "{{[Currency Code]}}",
  "length": 3
},
{
  "name": "ERS Action",
  "value": "{{[ERS Action]}}",
  "length": 1
},
{
  "name": "Item ID",
  "value": "{{[Item ID]}}",
  "length": 18
},
{
  "name": "More Information",
  "value": "{{[More Information]}}",
  "length": 100
},
{
  "name": "Item SetID",
  "value": "{{[Item SetID]}}",
  "length": 5
},
{
  "name": "Lot Control",
  "value": "{{[Lot Control]}}",
  "length": 1
},
{
  "name": "Receipt Number",
  "value": "{{[Receipt Number]}}",
  "length": 10
},
{
  "name": "Receipt Line",
  "value": "{{[Receipt Line]}}",
  "length": 5,
  "direction": "right"
},
{
  "name": "Receiver Shipping Sequence",
  "value": "{{[Receiver Shipping Sequence]}}",
  "length": 3,
  "direction": "right"
},
{
  "name": "Serial Control",
  "value": "{{[Serial Control]}}",
  "length": 1
},
{
  "name": "ShipTo Location(200)",
  "value": "{{[ShipTo Location(200)]}}",
  "length": 10
},
{
  "name": "ShipTo SetID(200)",
  "value": "{{[ShipTo SetID(200)]}}",
  "length": 5
},
{
  "name": "Standard Unit of Measure",

```



```

        "value": "[Standard Unit of Measure]",
        "length": 3
    },
    {
        "name": "Unit of Measure",
        "value": "[Unit of Measure]",
        "length": 3
    },
    {
        "name": "Supplier ID(200)",
        "value": "[Supplier ID(200)]",
        "length": 10
    },
    {
        "name": "Supplier SetID(200)",
        "value": "[Supplier SetID(200)]",
        "length": 5
    },
    {
        "name": "Revision Override",
        "value": "[Revision Override]",
        "length": 1
    },
    {
        "name": "Revision(200)",
        "value": "[Revision(200)]",
        "length": 4
    },
    {
        "name": "Pro Number(200)",
        "value": "[Pro Number(200)]",
        "length": 40
    },
    {
        "name": "Supplier Contract ID",
        "value": "[Supplier Contract ID]",
        "length": 10
    },
    {
        "name": "UPN ID",
        "value": "[UPN ID]",
        "length": 13
    },
    {
        "name": "Supplier ID Number(200)",
        "value": "[Supplier ID Number(200)]",
        "length": 13
    },
    {
        "name": "ShipTo GLN(200)",
        "value": "[ShipTo GLN(200)]",
        "length": 13
    }
]
}
]

```

### Output (Fixed-Width)

200A	1	1	PE-Recv Test2007H	0	0.00
0			0002.0000EA		
0.0000	US0010000000235	1 1		0	0
0.00000000					

## 8.5. 7.5. 300

This section describes how 300 a Fixed-Width element in the JSON format is converted to the Fixed-Width format with the help of the IO definition file.

### Input (JSON)

```
[
{
"300": [
{
"Audit Action(300)": "A",
"ASN Sequence number(300)": "1",
"ASN Schedule Sequence Number(300)": "1",
"ASN Lot/Serial Number": "0",
"Shipment Number(300)": "PE-Recv Test2007H",
"Order Number(300)": "",
"Order Line(300)": "0",
"ASN Item ID(300)": "",
"Sched Line NO(300)": "0.00",
"Demand Line No": "0",
"Quantity Shipped(300)": "0002.0000",
"Shipment Unit of Measure(300)": "",
"Lot Id": "",
"Serial ID(300)": "",
"PO Business Unit(300)": "US001",
"PO Number(300)": "0000000235",
"Line Number(300)": "1",
"Schedule Number(300)": "0",
"Supplier Lot": "",
"Revision(300)": ""
}
]
}
]
```

### Process (Definition File)

```
"children":
[
{
"maxOccurrence": 50,
"relativeDataPath": "300",
"required": true,
"elements": [
{
"name": "300",
"value": "300",
"length": 3
},
{
"name": "Audit Action(300)",
"value": "{{[Audit Action(300)]}}",
"length": 1
},
{
"name": "ASN Sequence number(300)",
```

```

    "value": "[{{[ASN Sequence number(300)]}}]",
    "length": 9,
    "direction": "right"
  },
  {
    "name": "ASN Schedule Sequence Number(300)",
    "value": "[{{[ASN Schedule Sequence Number(300)]}}]",
    "length": 9,
    "direction": "right"
  },
  {
    "name": "ASN Lot/Serial Number",
    "value": "[{{[ASN Lot/Serial Number]}}]",
    "length": 9,
    "direction": "right"
  },
  {
    "name": "Shipment Number(300)",
    "value": "[{{[Shipment Number(300)]}}]",
    "length": 30
  },
  {
    "name": "Order Number(300)",
    "value": "[{{[Order Number(300)]}}]",
    "length": 10
  },
  {
    "name": "Order Line(300)",
    "value": "[{{[Order Line(300)]}}]",
    "length": 5,
    "direction": "right"
  },
  {
    "name": "ASN Item ID(300)",
    "value": "[{{[ASN Item ID(300)]}}]",
    "length": 18
  },
  {
    "name": "Sched Line NO(300)",
    "value": "[{{toFixed [Sched Line NO(300)] \"2\"}}]",
    "length": 7,
    "direction": "right"
  },
  {
    "name": "Demand Line No",
    "value": "[{{[Demand Line No]}}]",
    "length": 4,
    "direction": "right"
  },
  {
    "name": "Quantity Shipped(300)",
    "value": "[{{toFixed [Quantity Shipped(300)] \"4\"}}]",
    "length": 17,
    "direction": "right"
  },
  {
    "name": "Shipment Unit of Measure(300)",
    "value": "[{{[Shipment Unit of Measure(300)]}}]",
    "length": 3
  },
  {
    "name": "Lot Id",
    "value": "[{{[Lot Id]}}]",
    "length": 15
  },
  },

```

```

    {
      "name": "Serial ID(300)",
      "value": "{{[Serial ID(300)]}}",
      "length": 20
    },
    {
      "name": "PO Business Unit(300)",
      "value": "{{[PO Business Unit(300)]}}",
      "length": 5
    },
    {
      "name": "PO Number(300)",
      "value": "{{[PO Number(300)]}}",
      "length": 10
    },
    {
      "name": "Line Number(300)",
      "value": "{{[Line Number(300)]}}",
      "length": 5,
      "direction": "right"
    },
    {
      "name": "Schedule Number(300)",
      "value": "{{[Schedule Number(300)]}}",
      "length": 3,
      "direction": "right"
    },
    {
      "name": "Supplier Lot",
      "value": "{{[Supplier Lot]}}",
      "length": 15
    },
    {
      "name": "Revision(300)",
      "value": "{{[Revision(300)]}}",
      "length": 4
    }
  ]
}
]

```

**Output (Fixed-Width)**

300A	1	1	0PE-Recv Test2007H	0		
0.00	0	0002.0000		US0010000000235	1	0

## 9. FAQs

1. While performing a JSON to EDI conversion, I am unable to convert special characters. For example, a "<" symbol appears as "&lt;". What is the workaround?

**Ans:** You are requested to use {{{ }} in the Definition File while converting the JSON element to EDI.

The following code-snippet illustrates how to use {{{ }} when there is a special character in the input data. In the following example, MSG contains a less than (<) symbol.

## Input (JSON)

```
"MSG": [{
    "Free-Form Message Text": "PREFERENCE IS < PALLET-LOAD, BUT IF CONTAINERS
ARE FLOOR-LOADED, THEN PLEASE DO CLAMP-LOAD OR STRAIGHT FLOOR-LOAD. DO NOT USE SLIP
SHEET FOR THIS FC DESTINATION.PAYMENT TERMS ARE PER CONTAINER."
}]
```

## Process (Definition File)

```
{
  "name": "MSG",
  "description": "Message Text",
  "relativeDataPath": "MSG",
  "maxOccurrence": 50,
  "required": true,
  "elements": [
    {
      "name": "name",
      "value": "MSG"
    },
    {
      "name": "Free-Form Message Text",
      "value": "{{[Free-Form Message Text]}}// {{ }} here is used so that the special
characters appear as is in the EDI format."
    }
  ]
}
```

## Output (EDI)

```
MSG*PREFERENCE IS < PALLET-LOAD, BUT IF CONTAINERS ARE FLOOR-LOADED, THEN PLEASE DO
CLAMP-LOAD OR STRAIGHT FLOOR-LOAD. DO NOT USE SLIP SHEET FOR THIS FC
DESTINATION.PAYMENT TERMS ARE PER CONTAINER.~
```