

1.1.1 Data Specs (Low to High) Filter Call-out:

Rule # 1: Line 1 of the file will conform to the following format:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Rule # 1: Line 2 of the file will conform to the following format:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Rule # 1: Line 3 of the file will conform to the following format

```
<soap:Header>
```

Rule # 1: Line 4 of the file will conform to the following format

```
<soap:FVSCMTransferHeader>
```

Rule # 1: Line 5 of the file will conform to the following format

```
<soap:Source>
```

Rule # 2: Line 6 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Address>1</soap:Address>
```

Rule # 3: Line 7 of the file will conform to the following format, until the '>', after that it should only accept a positive integer value up to the '>'. After that the line will conform to the following format.

```
<soap:Port>1</soap:Port>
```

Rule # 1: Line 8 of the file will conform to the following format

```
</soap:Source>
```

Rule # 1: Line 9 of the file will conform to the following format

```
<soap:Destination>
```

Rule # 2: Line 10 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Address>2</soap:Address>
```

Rule # 3: Line 11 of the file will conform to the following format, until the '>', after that it should only accept a positive integer value up to the '>'. After that the line will conform to the following format.

```
<soap:Port>2</soap:Port>
```

Rule # 1: Line 12 of the file will conform to the following format

```
</soap:Destination>
```

Rule # 2: Line 13 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Type>1</soap:Type>
```

Rule # 2: Line 14 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:MessageID>3984</soap:MessageID>
```

Rule # 1: Line 15 of the file will conform to the following format

```
</soap:FVSCMTransferHeader>
```

Rule # 1: Line 16 of the file will conform to the following format

```
</soap:Header>
```

Rule # 1: Line 17 of the file will conform to the following format

```
<soap:Body>
```

Rule # 4: Line 18 of the file will contain the payload value. The payload will either be NULL, or an XML document value (consisting of ascii characters), or a binary value. If the payload value is not NULL, then it will contain no less than 3 lines (meaning at least 2 carriage returns) up to an unknown amount of lines.

```
<A>aaa</A>
```

Rule # 1: Line (unknown) of the file will conform to the following format

```
</soap:Body>
```

Rule # 1: Line (unknown) of the file will conform to the following format

```
</soap:Envelope>
```

1.1.2 Data Specs (High to Low) Filter Call-out:

Rule # 1: Line 1 of the file will conform to the following format:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Rule # 1: Line 2 of the file will conform to the following format:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Rule # 1: Line 3 of the file will conform to the following format

```
<soap:Header>
```

Rule # 1: Line 4 of the file will conform to the following format

```
<soap:FVSCMTransferHeader>
```

Rule # 1: Line 5 of the file will conform to the following format

```
<soap:Source>
```

Rule # 2: Line 6 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Address>1</soap:Address>
```

Rule # 3: Line 7 of the file will conform to the following format, until the '>', after that it should only accept a positive integer value up to the '>'. After that the line will conform to the following format.

```
<soap:Port>1</soap:Port>
```

Rule # 1: Line 8 of the file will conform to the following format

```
</soap:Source>
```

Rule # 1: Line 9 of the file will conform to the following format

```
<soap:Destination>
```

Rule # 2: Line 10 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Address>2</soap:Address>
```

Rule # 3: Line 11 of the file will conform to the following format, until the '>', after that it should only accept a positive integer value up to the '>'. After that the line will conform to the following format.

```
<soap:Port>2</soap:Port>
```

Rule # 1: Line 12 of the file will conform to the following format

```
</soap:Destination>
```

Rule # 2: Line 13 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

```
<soap:Type>1</soap:Type>
```

Rule # 2: Line 14 of the file will conform to the following format, until the '>', after that it should only accept a string value up to the '>'. After that the line will conform to the following format.

<soap:MessageID>3984</soap:MessageID>

Rule # 1: Line 15 of the file will conform to the following format

</soap:FVSCMTransferHeader>

Rule # 1: Line 16 of the file will conform to the following format

</soap:Header>

Rule # 1: Line 17 of the file will conform to the following format

<soap:Body>

Rule # 5: Line 18 of the file may contain the payload value. The payload will either be NULL, or an XML document value (consisting of ascii characters). If the payload value is not NULL, then it will contain no less than 3 lines (meaning at least 2 carriage returns) up to an unknown amount of lines.

<A>aaa

Rule # 1: Line (unknown) of the file will conform to the following format

</soap:Body>

Rule # 1: Line (unknown) of the file will conform to the following format

</soap:Envelope>

Appendix 1 – FVS-CM XML Message schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Schema for the SOAP/1.1 envelope -->
<xs:schema
targetNamespace="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://schemas.xmlsoap.org
/soap/envelope/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Envelope, header and body -->
  <xs:element name="Envelope" type="tns:Envelope" />
  <xs:complexType name="Envelope">
    <xs:sequence>
      <xs:element ref="tns:Header" minOccurs="0" />
      <xs:element ref="tns:Body" />
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Header" type="tns:Header" />
  <xs:complexType name="Header">
    <xs:sequence>
      <xs:element name="FVSCMTransferHeader">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Source">
              <xs:complexType>
                <xs:sequence>
                  <xs:element
name="Address" type="xs:string" />
                  <xs:element
name="Port" type="xs:positiveInteger" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Destination">
              <xs:complexType>
                <xs:sequence>
                  <xs:element
name="Address" type="xs:string" />
                  <xs:element
name="Port" type="xs:positiveInteger" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Type" type="xs:string" />
            <xs:element name="MessageID" type="xs:string"
/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Body" type="tns:Body" />
  <xs:complexType name="Body">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:anyAttribute namespace="##any" processContents="lax">

```

```

        <xs:annotation>
            <xs:documentation>
                Prose in the spec does not specify that attributes are allowed on the Body element
            </xs:documentation>
        </xs:annotation>
    </xs:anyAttribute>
</xs:complexType>
<!-- Global Attributes. The following attributes are intended to be usable via qualified attribute
names on any complex type referencing them. -->
<xs:attribute name="mustUnderstand">
    <xs:simpleType>
        <xs:restriction base="xs:boolean">
            <xs:pattern value="0|1" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="actor" type="xs:anyURI" />
<xs:simpleType name="encodingStyle">
    <xs:annotation>
        <xs:documentation>
            'encodingStyle' indicates any canonicalization conventions followed in the contents of the
            containing element. For example, the value 'http://schemas.xmlsoap.org/soap/encoding/' indicates the
            pattern described in SOAP specification
        </xs:documentation>
    </xs:annotation>
    <xs:list itemType="xs:anyURI" />
</xs:simpleType>
<xs:attribute name="encodingStyle" type="tns:encodingStyle" />
<xs:attributeGroup name="encodingStyle">
    <xs:attribute ref="tns:encodingStyle" />
</xs:attributeGroup>
<xs:element name="Fault" type="tns:Fault" />
<xs:complexType name="Fault" final="extension">
    <xs:annotation>
        <xs:documentation>
            Fault reporting structure
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="faultcode" type="xs:QName" />
        <xs:element name="faultstring" type="xs:string" />
        <xs:element name="faultactor" type="xs:anyURI" minOccurs="0" />
        <xs:element name="detail" type="tns:detail" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="detail">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax" />
</xs:complexType>
</xs:schema>

```

Appendix 2 – Sample FVS-CM XML Message Document

Sample 1 – Low to High and High to Low, XML Message document Example

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
USX_message.xsd">
  <soap:Header>
    <soap:FVSCMTransferHeader>
      <soap:Source>
        <soap:Address>1</soap:Address>
        <soap:Port>1</soap:Port>
      </soap:Source>
      <soap:Destination>
        <soap:Address>2</soap:Address>
        <soap:Port>2</soap:Port>
      </soap:Destination>
      <soap:Type>1</soap:Type>
      <soap:MessageID>3984</soap:MessageID>
    </soap:FVSCMTransferHeader>
  </soap:Header>
  <soap:Body>
    <A>aaa</A>
  </soap:Body>
</soap:Envelope>
```