



Canadian Food
Inspection Agency

Agence canadienne
d'inspection des aliments

Canada



CFIA

AIRS Validation Service (AVS)

Technical Manual

2009-11-02

Table of Contents

1.0 Overview.....	3
2.0 Service Descriptors	3
3.0 Service Call.....	3
3.1 Restrictions	4
3.2 Request Format	4
3.3 Validation.....	7
3.4 Response Format.....	7
3.5 SOAP Errors	8

1.0 Overview

The AIRS Validation Service is a web service designed to validate multiple AIRS commodities in a single request. Information is sent to the service in an XML format, where it is validated and a response message is generated.

2.0 Service Descriptors

The service WSDL description can be obtained by accessing the URL:

<http://airs-sari.inspection.gc.ca/avs/bvs.svc?wsdl>

The WSDL provides the service description in an XML format which can be used by programmers to create an interface to the service.

The schemas which define the XML format used for requests and response are available at:

Validation request schema: http://airs-sari.inspection.gc.ca/avs/schema/vr_in_1.0.xsd

Validation response schema: http://airs-sari.inspection.gc.ca/avs/schema/vr_out_1.0.xsd

3.0 Service Call

The service has a single callable method: “ValidateRequirements”. This method takes 4 parameters, all are required.

xmlContent	A byte array containing the XML content to be validated. The content must be valid according to the published schema <i>vr_in_1.0.xsd</i> .
schemaVersion	A string indicating the version of the schema being used. Presently there is only one version which is accepted: “1.0”
key	A string containing a broker-specific identifier assigned by CFIA. This key must be valid for the importer/broker to access the system.
lang	An integer indicating the language in which the service should provide error messages: 1=English, 2=French

3.1 Restrictions

- The format of the XML is restricted by the schema. The schema includes string-length restrictions and element-ordering.
- The size of the entire message cannot exceed 500k.
- The number of commodities which can be validated in a single request cannot exceed 400.
- Large files transferred over a slow connection may result in a timeout error.

Note: Every commodity in the request counts towards the 400 maximum, so it is recommended to pre-process the request to remove duplicates.

3.2 Request Format

The request XML must be valid according to the published schema *vr_in_1.0.xsd*, otherwise the system will return a parsing error. The request schema's root element is *ValidateTransaction*. Within this element there can be 1 to 999 groups of commodities. Each group can contain 1 to 999 commodities, though the service is capped at a specified maximum number of commodities across all groups (see *Restrictions*). A commodity element must have an *HSNumber* at minimum. An example follows.

Element descriptions:

ValidateTransaction	Root node, required.
CommodityGroup	Used to define groups of commodities. This element exists to allow systems to group sets of commodities for validation. An ID attribute (string) is required so that the response can show an error count per group. At least one group is required.
Commodity	This element defines an AIRS commodity. It requires an ID attribute (string) so that if any errors are present the response will associate the error with the provided ID. At least one commodity is required per group.
HSNumber	The "HS number" or "HS code". The AIRS web application provides this as a 6-character code, though the field accepts up to 10 characters to allow for CBSA tariff codes (only the first 6 characters are read). The <i>HSNumber</i> must be provided for each commodity.

RequirementId	The requirement number determined by the AIRS web application. This field may not always be necessary.
RequirementVersion	The requirement's version determined by the AIRS web application. This field may not always be necessary.
AirsCode	The OGD extension or "AIRS code". This code is always 6 characters long (e.g. "940000"). This field may not always be necessary.
DestinationProvince	The 2-character code which identifies the destination province (e.g. "ON", "BC"). This field may not always be necessary.
OriginCountry	The 2-character code for the commodity's country of origin (e.g. "US", "NZ"). This field may not always be necessary.
OriginState	The 2-character code for the commodity's state of origin; this field is for US commodities only (e.g. "NY", "TN"). This field may not always be necessary.
Enduse	The 2-character code for the end use (e.g. "01", "14"). This field may not always be necessary.
Miscellaneous	The 2 or 3-character code for the miscellaneous field (e.g. "01", "34"). This field may not always be necessary.
Registration	This element is repeatable and is used to identify the registrations which belong to the requirement for this commodity. The element has no data; the code for the registration is identified through the registrationId attribute.

An example XML request. This request has just 2 groups and 3 commodities.

```
<?xml version="1.0"?>
<ValidateTransaction>
  <CommodityGroup commodityGroupId="A1">
    <Commodity commodityId="GGL023--0001">
      <HSNumber>020500</HSNumber>
      <RequirementId>21247</RequirementId>
      <RequirementVersion>14</RequirementVersion>
      <AirsCode>512700</AirsCode>
      <OriginCountry>US</OriginCountry>
      <OriginState>TN</OriginState>
      <Enduse>69</Enduse>
      <Miscellaneous>34</Miscellaneous>
      <Registration registrationId="14" />
      <Registration registrationId="28" />
      <Registration registrationId="40" />
      <Registration registrationId="41" />
      <Registration registrationId="98" />
      <Registration registrationId="99" />
      <Registration registrationId="102" />
      <Registration registrationId="104" />
      <Registration registrationId="105" />
      <Registration registrationId="111" />
      <Registration registrationId="120" />
      <Registration registrationId="121" />
      <Registration registrationId="132" />
      <Registration registrationId="141" />
      <Registration registrationId="318" />
    </Commodity>
  </CommodityGroup>
  <CommodityGroup commodityGroupId="A2">
    <Commodity commodityId="GHL030--0001">
      <HSNumber>960000</HSNumber>
    </Commodity>
    <Commodity commodityId="GHL030--0002">
      <HSNumber>920000</HSNumber>
    </Commodity>
  </CommodityGroup>
</ValidateTransaction>
```

In the above example, the first commodity has requirement #21247 with version 14. This is a regulated commodity which requires the selection of an *origin*, *enduse*, and *miscellaneous* code. The registrations IDs listed are those indicated by the AIRS web application “EDI” page (in this example, two registrations are missing). The commodities in the second group are unregulated commodities and therefore have no associated AIRS attributes or registrations.

Other elements which are not shown in this example are:

```
<DestinationProvince>BC</DestinationProvince>
```

3.3 Validation

Validation of the commodities will only occur if the XML format is valid. The length of time a request takes will vary depending on the number of commodities in the request. A large request (300+ commodities) may take a minute or more. Once validated, the result is returned as an XML document.

3.4 Response Format

If the above example XML message was passed to the AIRS Validation Service with a valid webkey, the response would be as follows. (NOTE: while the format will not change, AIRS requirements are subject to change over time and therefore this example may not return the same result).

```
<ValidateTransactionResult errorCount="1">
  <CommodityGroup commodityGroupId="A1" errorCount="1">
    <Commodity commodityId="GGL023--0001">
      <Error>MISSING OR INVALID REGISTRATION NUMBER</Error>
    </Commodity>
  </CommodityGroup>
  <CommodityGroup commodityGroupId="A2" errorCount="0" />
</ValidateTransactionResult>
```

There was 1 error in the entire request, which is indicated in the top level element `<ValidateTransactionResult errorCount="1">`. The number of errors per group are displayed in the header for each group. If a group contains no commodities with errors, there will be nothing inside the `CommodityGroup` element. If the group does have an error, the commodity in error will be listed with an error element containing a message which indicates the cause of the error. In this example, the first commodity did not have the correct set of registrations.

3.5 SOAP Errors

AIRS-related errors are returned within the response XML if the request was otherwise valid. An invalid request will result in SOAP fault being returned by the service. The DETAIL portion of the soap fault contains the following information:

ErrorCode	A code identifying the type of error: ERR001 = system error ERR002 = invalid xml ERR003 = empty xml content ERR004 = unsupported schema version ERR005 = invalid key ERR006 = request too large (too many commodities)
ErrorMessage	A general error message which describes the error type.
ErrorDetail	Specific details about the error.
RequestId	An identifier associated with the request.

The following is an example of the detail portion of the soap fault message:

```
<ServiceFaultContract xmlns=http://cfia-air inspection.gc.ca
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <ErrorCode>ERR002</ErrorCode>
  <ErrorMessage>The XML is invalid.</ErrorMessage>
  <ErrorDetail>Data at the root level is invalid. Line 1, position 1.</ErrorDetail>
  <RequestId>R#000005</RequestId>
</ServiceFaultContract>
```

In cases where the request XML was not valid according to the schema, parser errors are returned in the ErrorDetail field (NOTE: the detail portion of the error message is only available in English).

Explanation of Error codes:

- ERR001 (system error) An unknown error occurred. Try the request again later. If this problem persists, contact the system administrator.
- ERR002 (invalid xml) The xmlContent parameter was not valid according to the schema. Specific parser errors will be available in the ErrorDetail.
- ERR003 (empty xml content) The xmlContent parameter did not have any data.
- ERR004 (unsupported schema) The schemaVersion parameter is not valid.

version)

- ERR005 (invalid key) The key parameter is not a valid web key.
- ERR006 (request too large) The request contained more than the maximum allowed number of commodities. The ErrorDetail will indicate the maximum allowed.