ScotEID web services API 1.7.9

Last updated: **23/02/2023**

Table of Contents

Schema changes

Schema version	Description
1.1	Added operations for managing incomplete lots: DeleteIncompleteLot SplitIncompleteLotAt MergeIncompleteLots
1.2	Added new operations: CreateCompleteLot FindIncompleteLotDates GetIncompleteLots GetCompleteLots Added animal metadata to TagReading element (UK number, ETAS status and last destination CPH)
1.2.1	Added support for visual recording of groups of animals by flock number only and VisuallyRecorded flag on lots
1.2.2	Added operation for updating incomplete lots
1.2.3	Added operation for updating incomplete lots by reference
1.2.5	Added ability to manually record non reads when using CreateIncompleteLot and similiar
1.3	Added operations for farm management software: GetAnimalsOnHolding (Get|Find)LotsByDepartureLocation (Get|Find)LotsByDestinationLocation
1.4	Added additional fields for recording pig movements, and addition of UpdateCompleteLotByReference operation
1.4.1	Added species code to (Find|Get)LotsBy(Destination|Departure)Location requests
1.4.2	Added CancelCompleteLotByReference operation
1.5	Added BVD requests
1.5.1	Added KillNumber field for abattoir reads
1.5.2	Added BVD test submission for labs
1.5.3	Added OnwardDepartureDate for market moves
1.5.4	Added documentation for SPECC

Schema version	Description
1.5.5	Minor updates to SPECC
1.5.6	Added documentation for SPECC auction market webservice
1.5.7	Added data from main SPECC webservice to SPECC auction market webservice
1.6	Addition of BES webservices
1.7	Addition of ScotMoves webservices
1.7.4	GetCattleRequiringBVDTesting request added. BVD category added to GetAnimalFAStatus
1.7.5	Added SMGetCattleOnHoldings request
1.7.6	Added DateOnMainCPH to SMAnimal_Structure
1.7.7	Added cattle details to the SMGetCattleOnHoldings request
1.7.8	Added ControlledHousing and UngulateSeparation elements to pig FCI Documentation has been consolidated and restructured
1.7.9	Added export eligibility checker GetHoldingEUExportEligibilityRequest

Overview

This documentation is intended to provide an overview of the ScotEID API for developers wishing to integrate with it. It assumes the reader has experience of working with SOAP web services. For a full technical specification, please refer to the WSDL files given in the next section.

Endpoints

In order to aid developer integration and testing, two separate endpoints have been configured. The first is for testing and is a fully functional test environment whose database may be reset periodically.

Before access will be given to the production API, developers must successfully test their application against the test API.

Test site:
https://api.test.scoteid.com/api

Production site:
https://api.scoteid.com/api

A WSDL is available for each of these sites at https://api.test.scoteid.com/api?wsdl and https://api.scoteid.com/api?wsdl

Security & authentication

The ScotEID API uses HTTP basic authentication. To request a username and password for the test site, please contact support@scoteid.com.

Web service operation

Basic request information

All ScotEID API requests must contain the following elements:

XML element	Description
ApplicationName	The name of your application, e.g. “ScotEID Desktop”
ApplicationVersion	The version of your application, e.g. “1.1.4”
ApplicationKey	Vendor specific application key supplied by ScotEID
SchemaVersion	The version of the schema your application is built upon
Timestamp	The time at which your request was generated

Basic response information

All **successful** ScotEID API responses will contain at least a Timestamp element.

Global faults

Any ScotEID API request can return these SOAP Faults:

SOAP Fault	Description
MalformedRequestFault	Returned if your request is missing any of the basic request fields above
UnsupportedSchemaVersionFault	Returned if you are using an unsupported schema version
SecurityFault	Returned when attempting to carry out an operation which is not permitted

All of the faults documented here will contain a human readable ErrorDescription element giving specific details.

In addition to these faults (which can be returned for any operation), there are operation specific faults. Explanations of these faults can be found at the end of this document.

Sheep, Goat & Pig requests

SOAP fault explanations

LotValidationFault

The LotValidationFault will be returned whenever a required piece of lot information is missing or malformed, usually when creating, updating or completing a lot. The fault detail will contain 1 or more Error elements, as described below:

XML Element	Description
Property	The name of the property which contains an error
Message	A description of a problem with the property specified, e.g. “cannot be blank”

Example

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Client</faultcode>
         <faultstring>The lot could not be validated</faultstring>
         <detail>
            <LotValidationFault>
               <ErrorDescription>The lot contains errors</ErrorDescription>
               <Errors>
                  <Error>
                     <Property>tag_readings</Property>
                     <Message>cannot be empty</Message>
                  </Error>
               </Errors>
            </LotValidationFault>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

LotNotFoundFault

This fault will be returned when a lot could not be found for the given date and lot number; it contains no additional information regarding the error.

LotAlreadyExistsFault

This fault will be returned when attempting to create or update an incomplete lot to use a lot number and date which are already in use; it contains no additional information regarding the error.

Animal Metadata

This is a read-only element containing additional information which may be returned within any tag reading.

XML Element	Description
AnimalNumber	If this is a UK WYSIWYG tag, this element contains the formatted animal number
ETASVerified	True if the tag in question is known to have been issued. If this field is false, it **DOES NOT** mean that the tag was not issued, but rather its status has not yet been verified.
LastReadLocation	The last place this animal was read.
LastDestinationLocation	The last known destination of this animal.
LastReadDate	Date of the most recent movement.
LastMovementType	The movement type of the last recorded movement.
LastMovementDate	The date of the last recorded movement.

CreateIncompleteLot (Sheep only)

This operation allows tag readings to be uploaded to the ScotEID database in real time, without having to specify full details of the movement until later. Additional information on the data requirements for lot creation can be found in **21. Lot data requirements**.

Input

The minimum requirements for an incomplete lot are LotNumber, LotDate, MovementType and either TagReadings or FlockTags.

Response

A successful request will return a CreateIncompleteLotResponse containing a copy of the lot uploaded, which will have a LotReference. This reference can be saved and used to refer to the lot again in the future. Any unsuccessful request will return one of the SOAP faults documented below.

Faults

LotValidationFault
LotAlreadyExistsFault

Example request (with TagReadings and FlockTags)

Note that the request below contains a mixture of TagReadings (**21.4**) and FlockTags (**21.5**) - a lot created using either CreateIncompleteLot or its counterpart for creating completed lots can contain either tag readings, flock tags, a mixture of both, or none (for example if no tags in a lot were readable). In this instance 10 animals with flock tags, 8 EID tag readings and 2 manual tag readings, equalling the HeadCount of 20. Note that manual tag readings are entirely optionally, but being able to record “non reads” may be of potential use to some users.

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:CreateIncompleteLotRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:11+01:00</Timestamp>
      <Lot>
        <LotNumber>0004</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
        <TagReadings>
          <TagReading Timestamp="2011-09-21T12:55:31+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00100" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:55:41+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00101" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:55:51+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00102" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:56:01+01:00" Type="manual" />
          <TagReading Timestamp="2011-09-21T12:56:11+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00104" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:56:21+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00105" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:56:31+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00106" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:56:41+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00107" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:56:51+01:00" Type="manual" />
          <TagReading Timestamp="2011-09-21T12:57:01+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00108" Type="eid" />
        </TagReadings>
        <FlockTags>
          <FlockTag TagCount="10" FlockNumber="543210" />
        </FlockTags>
      </Lot>
    </api:CreateIncompleteLotRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:CreateIncompleteLotResponse>
      <Timestamp>2011-09-21T12:57:11+01:00</Timestamp>
      <Lot>
        <LotReference>269367</LotReference>
        <LotNumber>0004</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
      </Lot>
    </ns1:CreateIncompleteLotResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FindIncompleteLots (Sheep only)

Locates incomplete lots for the given date.

Input

LotDate element for which to find incomplete lots.

Output

A successful response to the FindIncompleteLots request will consist of a Lots element with multiple Lot elements.

Note: The Lots returned from FindIncompleteLots contain the basic information only, not TagReadings. If you require this data you must make a subsequent call to **GetIncompleteLot**.

Requesting incomplete lots for a date on which non exist will return an empty Lots element.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:FindIncompleteLotsRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:11+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
    </api:FindIncompleteLotsRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:FindIncompleteLotsResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lots>
        <Lot>
          <LotReference>269368</LotReference>
          <LotNumber>0001</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
        </Lot>
        <Lot>
          <LotReference>269369</LotReference>
          <LotNumber>0002</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
        </Lot>
        <Lot>
```

```
          <LotReference>269370</LotReference>
          <LotNumber>0003</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
        </Lot>
        <Lot>
          <LotReference>269367</LotReference>
          <LotNumber>0004</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>20</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
        </Lot>
      </Lots>
    </ns1:FindIncompleteLotsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FindIncompleteLotDates (Sheep only)

The FindIncompleteLotDates operation returns a list of distinct dates for which there are outstanding incomplete lots which need to be completed.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:FindIncompleteLotDatesRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
    </api:FindIncompleteLotDatesRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:FindIncompleteLotDatesResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-19</LotDate>
      <LotDate>2011-09-20</LotDate>
      <LotDate>2011-09-21</LotDate>
    </ns1:FindIncompleteLotDatesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetIncompleteLot (Sheep only)

The GetIncompleteLot operation allows retrieval of a previously created incomplete lot by lot number and date. As of schema version 1.2, each TagReading element returned may also contain animal metadata (see **20.Animal Metadata**).

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:GetIncompleteLotRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotNumber>0003</LotNumber>
      <LotDate>2011-09-21</LotDate>
    </api:GetIncompleteLotRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:GetIncompleteLotResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
```

```
      <Lot>
        <LotReference>269370</LotReference>
        <LotNumber>0003</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <ReadCount>2</ReadCount>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
        <TagReadings>
          <TagReading Hex="8200CE8CA5C8AFA8" Type="eid" Timestamp="2011-09-21T12:51:11+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000360</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AF9E" Type="eid" Timestamp="2011-09-21T12:51:21+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000350</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
        </TagReadings>
      </Lot>
    </ns1:GetIncompleteLotResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

LotNotFoundFault

GetIncompleteLots (Sheep only)

Retrieve all incomplete lots for a specified date. As of schema version 1.2, each TagReading element returned may also contain animal metadata (see **20.Animal Metadata**).

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:GetIncompleteLotsRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
    </api:GetIncompleteLotsRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns1:GetIncompleteLotsResponse xsi:type="ns1:FindIncompleteLotsResponse_Structure">
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lots>
        <Lot>
          <LotReference>269368</LotReference>
          <LotNumber>0001</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
          <TagReadings>
            <TagReading Hex="8200CE8CA5C8AFD0" Type="eid" Timestamp="2011-09-21T12:50:31+01:00"/>
            <TagReading Hex="8200CE8CA5C8AFC6" Type="eid" Timestamp="2011-09-21T12:50:41+01:00"/>
          </TagReadings>
        </Lot>
        <Lot>
          <LotReference>269369</LotReference>
          <LotNumber>0002</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
          <TagReadings>
            <TagReading Hex="8200CE8CA5C8AFBC" Type="eid" Timestamp="2011-09-21T12:50:51+01:00"/>
            <TagReading Hex="8200CE8CA5C8AFB2" Type="eid" Timestamp="2011-09-21T12:51:01+01:00"/>
          </TagReadings>
        </Lot>
        <Lot>
          <LotReference>269370</LotReference>
          <LotNumber>0003</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <ReadCount>2</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
          <TagReadings>
            <TagReading Hex="8200CE8CA5C8AFA8" Type="eid" Timestamp="2011-09-21T12:51:11+01:00">
              <Animal>
                <ETASVerified>false</ETASVerified>
                <AnimalNumber>UK054321000360</AnimalNumber>
```

```
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AF9E" Type="eid" Timestamp="2011-09-21T12:51:21+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000350</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
  </TagReadings>
</Lot>
<Lot>
  <LotReference>269367</LotReference>
  <LotNumber>0004</LotNumber>
  <LotDate>2011-09-21</LotDate>
  <MovementType>4</MovementType>
  <ReadCount>20</ReadCount>
  <DepartureLocation>11/222/3333</DepartureLocation>
  <ReadLocation>33/333/3333</ReadLocation>
  <DestinationLocation>44/555/6666</DestinationLocation>
  <TagReadings>
    <TagReading Hex="8200CE8CA5C8AEA4" Type="eid" Timestamp="2011-09-21T12:55:31+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000100</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEA5" Type="eid" Timestamp="2011-09-21T12:55:41+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000101</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEA6" Type="eid" Timestamp="2011-09-21T12:55:51+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000102</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Type="manual" Timestamp="2011-09-21T12:56:01+01:00"/>
    <TagReading Hex="8200CE8CA5C8AEA8" Type="eid" Timestamp="2011-09-21T12:56:11+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000104</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEA9" Type="eid" Timestamp="2011-09-21T12:56:21+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000105</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEAA" Type="eid" Timestamp="2011-09-21T12:56:31+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
```

```
            <AnimalNumber>UK054321000106</AnimalNumber>
            <LastDestinationLocation>44/555/6666</LastDestinationLocation>
            <LastMovementType>4</LastMovementType>
            <LastMovementDate>2011-09-21+01:00</LastMovementDate>
          </Animal>
        </TagReading>
        <TagReading Hex="8200CE8CA5C8AEAB" Type="eid" Timestamp="2011-09-21T12:56:41+01:00">
          <Animal>
            <ETASVerified>false</ETASVerified>
            <AnimalNumber>UK054321000107</AnimalNumber>
            <LastDestinationLocation>44/555/6666</LastDestinationLocation>
            <LastMovementType>4</LastMovementType>
            <LastMovementDate>2011-09-21+01:00</LastMovementDate>
          </Animal>
        </TagReading>
        <TagReading Type="manual" Timestamp="2011-09-21T12:56:51+01:00"/>
        <TagReading Hex="8200CE8CA5C8AEAC" Type="eid" Timestamp="2011-09-21T12:57:01+01:00">
          <Animal>
            <ETASVerified>false</ETASVerified>
            <AnimalNumber>UK054321000108</AnimalNumber>
            <LastDestinationLocation>44/555/6666</LastDestinationLocation>
            <LastMovementType>4</LastMovementType>
            <LastMovementDate>2011-09-21+01:00</LastMovementDate>
          </Animal>
        </TagReading>
      </TagReadings>
      <FlockTags>
        <FlockTag FlockNumber="543210" TagCount="10"/>
      </FlockTags>
    </Lot>
  </Lots>
 </ns1:GetIncompleteLotsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

CreateCompleteLot (Pigs, Sheep)

Creating a lot using this operation will perform a full validation of the lot, bypassing the need to create an incomplete lot and complete it separately.

Input

A lot including tag readings and all required movement information.

Output

The response will contain a full copy of the lot you have just created, including tag readings.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:CreateCompleteLotRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lot>
        <LotNumber>0005</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <HeadCount>2</HeadCount>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
```

```
        <TagReadings>
          <TagReading Timestamp="2011-09-21T12:57:10+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00600" Type="eid" />
          <TagReading Timestamp="2011-09-21T12:57:11+01:00"
                      ISO24631="1 0 04 00 0 826 0543210 00601" Type="eid" />
        </TagReadings>
      </Lot>
    </api:CreateCompleteLotRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:CreateCompleteLotResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lot>
        <LotReference>201518</LotReference>
        <LotNumber>0005</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <HeadCount>2</HeadCount>
        <ReadCount>0</ReadCount>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
      </Lot>
    </ns1:CreateCompleteLotResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Faults

LotNotFoundFault
LotValidationFault
LotAlreadyExistsFault

FindCompleteLots (Pigs, Sheep)

The functionality of this operation is identical to FindIncompleteLots, for complete lots.

GetCompleteLots (Pigs, Sheep)

The functionality of this operation is identical to GetIncompleteLots, for complete lots.

GetCompleteLot (Pigs, Sheep)

The functionality of this operation is identical to GetCompleteLot, for complete lots.

GetCompleteLotByReference (Pigs, Sheep)

This operation is similar to **GetCompleteLot** but rather than specifying a LotDate and LotNumber you must specify the LotReference obtained through one of the operations.

CancelCompleteLotByReference (Pigs, Sheep)

This operation allows a completed lot to be cancelled (i.e. deleted) by passing the LotReference which was generated when the lot was created.

UpdateCompleteLotByReference (Pigs, Sheep)

This operation allows a completed lot to be updated by passing the LotReference which was generated when the lot was created, and any new information. See **22.2** for an example.

CompleteLot (Sheep only)

Calling this operation will perform a full validation of the lot, and if successful change its status from incomplete to complete.

Input

If the lot created via a previous call to CreateIncompleteLot consisted of a full set of data, the lot may be completed by supplying only the lot date and lot number. Alternatively, you may supply a Lot element containing as little or many lot attributes as are required to make it a valid complete lot.

Note: TagReadings cannot be updated with the **CompleteLot** operation, so any Lot element containing a TagReadings element will result in a LotValidationFault.

Output

The response will contain a full copy of the lot you have just completed, including tag readings.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:CompleteLotRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
      <LotNumber>0003</LotNumber>
      <Lot>
        <HeadCount>2</HeadCount>
      </Lot>
    </api:CompleteLotRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:CompleteLotResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lot>
        <LotReference>201519</LotReference>
        <LotNumber>0003</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <HeadCount>2</HeadCount>
        <ReadCount>2</ReadCount>
```

```
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
        <TagReadings>
          <TagReading Hex="8200CE8CA5C8AFA8" Type="eid" Timestamp="2011-09-21T12:51:11+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000360</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AF9E" Type="eid" Timestamp="2011-09-21T12:51:21+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000350</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
        </TagReadings>
      </Lot>
    </ns1:CompleteLotResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Faults

LotNotFoundFault
LotValidationFault
LotAlreadyExistsFault

DeleteIncompleteLot (Sheep only)

Incomplete lots may be removed from the ScotEID database via webservices using this operation.

Input

LotDate and LotNumber elements should be supplied. Only one incomplete lot may be deleted at a time.

Output

If successful a timestamped response will be returned.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:DeleteIncompleteLotRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
      <LotNumber>0002</LotNumber>
    </api:DeleteIncompleteLotRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:DeleteIncompleteLotResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
    </ns1:DeleteIncompleteLotResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Faults

LotNotFoundFault

SplitIncompleteLotAt (Sheep only)

This operation facilitates splitting a lot which was not correctly split during reading by indicating the time at which the lot should have been split.

Input

LotDate, LotNumber and SplitAt

Output

If the lot specified is successfully split, both resulting lots will be returned. A lot number will be automatically generated for the new lot by appending a forward slash and a letter not already in use to the original lot number. For example, when splitting lot 0001 for the first time, the new lot will be numbered 0001/A. Splitting lot 0001 again will result in 0001/B being created. Splitting lot 0001/B will create 0001/C.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:SplitIncompleteLotAtRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
      <LotNumber>0004</LotNumber>
      <SplitAt>Wed Sep 21 11:56:21 UTC 2011</SplitAt>
    </api:SplitIncompleteLotAtRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:SplitIncompleteLotAtResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lots>
        <Lot>
          <LotReference>269371</LotReference>
          <LotNumber>0004</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <HeadCount>20</HeadCount>
          <ReadCount>10</ReadCount>
          <DepartureLocation>11/222/3333</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>44/555/6666</DestinationLocation>
          <TagReadings>
            <TagReading Hex="8200CE8CA5C8AEA4" Type="eid" Timestamp="2011-09-21T12:55:31+01:00">
              <Animal>
                <ETASVerified>false</ETASVerified>
                <AnimalNumber>UK054321000100</AnimalNumber>
                <LastDestinationLocation>44/555/6666</LastDestinationLocation>
                <LastMovementType>4</LastMovementType>
                <LastMovementDate>2011-09-21+01:00</LastMovementDate>
              </Animal>
            </TagReading>
            <TagReading Hex="8200CE8CA5C8AEA5" Type="eid" Timestamp="2011-09-21T12:55:41+01:00">
              <Animal>
                <ETASVerified>false</ETASVerified>
                <AnimalNumber>UK054321000101</AnimalNumber>
```

```
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEA6" Type="eid" Timestamp="2011-09-21T12:55:51+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000102</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Type="manual" Timestamp="2011-09-21T12:56:01+01:00"/>
    <TagReading Hex="8200CE8CA5C8AEA8" Type="eid" Timestamp="2011-09-21T12:56:11+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000104</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
  </TagReadings>
</Lot>
<Lot>
  <LotReference>269372</LotReference>
  <LotNumber>0004/A</LotNumber>
  <LotDate>2011-09-21</LotDate>
  <MovementType>4</MovementType>
  <HeadCount>20</HeadCount>
  <ReadCount>10</ReadCount>
  <DepartureLocation>11/222/3333</DepartureLocation>
  <ReadLocation>33/333/3333</ReadLocation>
  <DestinationLocation>44/555/6666</DestinationLocation>
  <TagReadings>
    <TagReading Hex="8200CE8CA5C8AEA9" Type="eid" Timestamp="2011-09-21T12:56:21+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000105</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEAA" Type="eid" Timestamp="2011-09-21T12:56:31+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000106</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Hex="8200CE8CA5C8AEAB" Type="eid" Timestamp="2011-09-21T12:56:41+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000107</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
    <TagReading Type="manual" Timestamp="2011-09-21T12:56:51+01:00"/>
    <TagReading Hex="8200CE8CA5C8AEAC" Type="eid" Timestamp="2011-09-21T12:57:01+01:00">
      <Animal>
        <ETASVerified>false</ETASVerified>
        <AnimalNumber>UK054321000108</AnimalNumber>
        <LastDestinationLocation>44/555/6666</LastDestinationLocation>
        <LastMovementType>4</LastMovementType>
        <LastMovementDate>2011-09-21+01:00</LastMovementDate>
      </Animal>
    </TagReading>
  </TagReadings>
```

```
        </Lot>
      </Lots>
    </ns1:SplitIncompleteLotAtResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Faults

LotNotFoundFault

MergeIncompleteLots (Sheep only)

This operation allows two incomplete lots on the same date to be merged.

Input

A LotDate element followed by two LotNumber elements is expected.

Output

If single resulting lot will be returned. Note that this lot will retain all details from the first lot number specified, plus the additional tags from the second lot.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:MergeIncompleteLotsRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <LotDate>2011-09-21</LotDate>
      <LotNumber>0004</LotNumber>
      <LotNumber>0004/A</LotNumber>
    </api:MergeIncompleteLotsRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:MergeIncompleteLotsResponse>
      <Timestamp>2011-09-21T12:57:12+01:00</Timestamp>
      <Lot>
        <LotReference>269371</LotReference>
        <LotNumber>0004</LotNumber>
        <LotDate>2011-09-21</LotDate>
        <MovementType>4</MovementType>
        <HeadCount>20</HeadCount>
        <ReadCount>5</ReadCount>
        <DepartureLocation>11/222/3333</DepartureLocation>
        <ReadLocation>33/333/3333</ReadLocation>
        <DestinationLocation>44/555/6666</DestinationLocation>
        <TagReadings>
          <TagReading Hex="8200CE8CA5C8AEA4" Type="eid" Timestamp="2011-09-21T12:55:31+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000100</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AEA5" Type="eid" Timestamp="2011-09-21T12:55:41+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000101</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
```

```
          <TagReading Hex="8200CE8CA5C8AEA6" Type="eid" Timestamp="2011-09-21T12:55:51+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000102</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Type="manual" Timestamp="2011-09-21T12:56:01+01:00"/>
          <TagReading Hex="8200CE8CA5C8AEA8" Type="eid" Timestamp="2011-09-21T12:56:11+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000104</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AEA9" Type="eid" Timestamp="2011-09-21T12:56:21+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000105</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AEAA" Type="eid" Timestamp="2011-09-21T12:56:31+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000106</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Hex="8200CE8CA5C8AEAB" Type="eid" Timestamp="2011-09-21T12:56:41+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000107</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
          <TagReading Type="manual" Timestamp="2011-09-21T12:56:51+01:00"/>
          <TagReading Hex="8200CE8CA5C8AEAC" Type="eid" Timestamp="2011-09-21T12:57:01+01:00">
            <Animal>
              <ETASVerified>false</ETASVerified>
              <AnimalNumber>UK054321000108</AnimalNumber>
              <LastDestinationLocation>44/555/6666</LastDestinationLocation>
              <LastMovementType>4</LastMovementType>
              <LastMovementDate>2011-09-21+01:00</LastMovementDate>
            </Animal>
          </TagReading>
        </TagReadings>
      </Lot>
    </ns1:MergeIncompleteLotsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Faults

LotNotFoundFault
UnmergeableLotsFault

GetAnimalsOnHolding (Sheep only)

This operation allows a CCP or partner farm to retrieve a list of animals currently recorded as being on one of their registered holding numbers.

Input

The only input required is one of the users registered holding numbers.

Output

A list of animals as shown in the example output below.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:GetAnimalsOnHoldingRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T15:53:10+01:00</Timestamp>
      <HoldingNumber>70/001/0001</HoldingNumber>
    </api:GetAnimalsOnHoldingRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:GetAnimalsOnHoldingResponse>
      <Timestamp>2011-09-21T15:53:10+01:00</Timestamp>
      <Animals>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000100</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000101</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000102</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000104</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
```

```
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000105</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000106</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000107</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
        <Animal>
          <ETASVerified>false</ETASVerified>
          <AnimalNumber>UK054321000108</AnimalNumber>
          <LastReadLocation>33/333/3333</LastReadLocation>
          <LastDestinationLocation>70/001/0001</LastDestinationLocation>
          <LastMovementType>4</LastMovementType>
          <LastMovementDate>2011-09-21+01:00</LastMovementDate>
        </Animal>
      </Animals>
    </ns1:GetAnimalsOnHoldingResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

SecurityFault

(Find/Get)LostByDestinationLocation (Sheep only)

FindLotsByDestinationLocation and **GetLotsByDestinationLocation** allow partner farms to retrieve logs recorded on to one of their registered holdings by CCPs for a specified period of time. Note that the Find and Get methods both take the same arguments, but the response to the Get method includes all tag readings. As an alternative, **FindLotsByDestinationLocation** can be used on its own and individual lots retrieved using the lot reference with **GetLotByReference**.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:FindLotsByDestinationLocationRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T20:19:23+01:00</Timestamp>
      <DestinationLocation>70/001/0001</DestinationLocation>
      <FromDate>2011-09-14</FromDate>
      <ToDate>2011-09-21</ToDate>
    </api:FindLotsByDestinationLocationRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:FindLotsByDestinationLocationResponse>
      <Timestamp>2011-09-21T20:19:23+01:00</Timestamp>
      <Lots>
        <Lot>
          <LotReference>1</LotReference>
          <LotNumber>0006</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <HeadCount>20</HeadCount>
          <ReadCount>0</ReadCount>
          <DepartureLocation>70/001/0002</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
          <DestinationLocation>70/001/0001</DestinationLocation>
        </Lot>
      </Lots>
    </ns1:FindLotsByDestinationLocationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

SecurityFault

(Find/Get)LotsByDepartureLocation (Sheep only)

These operations are the equivalents of **(Find/Get)LotsByDestinationLocation** for retrieving lots recorded off a registered holding.

Example request

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <api:FindLotsByDepartureLocationRequest xmlns:api="http://api.scoteid.com/api/">
      <ApplicationName>ScotEID Ruby SOAP Client</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <SchemaVersion>1.3</SchemaVersion>
      <Timestamp>2011-09-21T20:13:04+01:00</Timestamp>
      <DepartureLocation>70/001/0001</DepartureLocation>
      <FromDate>2011-09-14</FromDate>
      <ToDate>2011-09-21</ToDate>
    </api:FindLotsByDepartureLocationRequest>
  </env:Body>
</env:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
  <SOAP-ENV:Body>
    <ns1:FindLotsByDepartureLocationResponse>
      <Timestamp>2011-09-21T20:19:23+01:00</Timestamp>
      <Lots>
        <Lot>
          <LotReference>1</LotReference>
          <LotNumber>0006</LotNumber>
          <LotDate>2011-09-21</LotDate>
          <MovementType>4</MovementType>
          <HeadCount>20</HeadCount>
          <ReadCount>0</ReadCount>
          <DepartureLocation>70/001/0002</DepartureLocation>
          <ReadLocation>33/333/3333</ReadLocation>
```

```
          <DestinationLocation>70/001/0001</DestinationLocation>
        </Lot>
      </Lots>
    </ns1:FindLotsByDepartureLocationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

SecurityFault

Lot Element

** Pigs only - required when not using a single slap mark for a lot
*** Pigs only - required when a updating a lot to indicate that the consignment has been received.

When dealing with pig movements, the consignment is considered to have been received once **ArrivalDate** is specified - while this field is blank the presence of the additional data will not be validated.

XML Element	Type	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
LotReference	String (32 chars max)	**NB:** This is not an input requirement, but may be captured from responses as a reference to lots which your software has created. Maximum of 34 characters	N/A	N/A	N/A
ExternalReference	String (32 chars max)	**Optional** reference to this lot in an external system.	No	No	No
LotNumber	String (10 chars max)	Unique lot identifier for this date; maximum of 10 characters	Yes	Yes	Yes
LotDate	Date	Date of movement	Yes	Yes	Yes
MovementType	Integer - see WSDL for enumeration	Integer movement type. **See 21.7 for possible movement types.**	Yes	Yes	Yes
Species	Integer - see WSDL for enumeration	Integer species code - see WSDL for enumeration values	No - defaults to 4	No - defaults to 4	Yes - must be 3
LotDescription	Integer - see WSDL for enumeration	Description of the type of animals within this lot	No	No	No
BatchMark	String (8 chars max)	Batch identifier for pig lots, e.g. slap mark.	No	No	Yes - unless using PigIdentities

XML Element	Type	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
ArrivalDate	Date	**For pigs, once the arrival date is supplied the consignment will be considered received and additional information will also be required (e.g. DOACount, see those marked ***)**	No	No	Yes ***
DepartureDate	Date		No	No	No
OnwardDepartureDate	Date	For mart moves only - should be given as the date the animals left the market for their destination	No	Yes	No
ReadLocation	String (11 chars matching CC/PPP/HHHH)	Holding number where read was taken. **See 21.6 for more information.**	No	Yes	Yes
DepartureLocation	String (11 chars matching CC/PPP/HHHH)	Holding number where lot moved from. **See 21.6 for more information.**	No	Yes	Yes
DestinationLocation	String (11 chars matching CC/PPP/HHHH)	Holding number where lot is moving to. **See 21.6 for more information.**	No	Yes	Yes ***
HeadCount	Integer	Actual number of animals within this lot (NOT number of EID reads)	No	Yes	Yes
DOACount	Integer	Number of animals DOA	No	No	Yes ***
AllEID	Boolean	Optional flag specifying whether or not all sheep within a lot are electronically tagged	No	No	No

XML Element	Type	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
VisuallyRecorded	Boolean	Optional flag specifying whether or not this data was recorded visually rather than using an EID reader - especially useful if submitting FlockTags	No	No	No
DepartureKeeper	Complex type	Element containing departure keeper information. **See 21.1 for more information.**	No	No	Yes
DestinationKeeper	Complex type	Element containing departure keeper information. **See 21.1 for more information.**	No	No	Yes ***
TransportInformation	Complex type		No	No	No
BuyerInvoiceNumber	Integer		No	No	No
SellerPaymentNumber	Integer		No	No	No
FCI	Complex type		No	No	No
TagReadings	Array	Element containing multiple TagReading elements. **See 21.4 for more information.**	Yes (or FlockTags)	Yes (or FlockTags)	N/A
FlockTags	Array	Element containing multiple FlockTag elements (visually recorded groups under the same flock number) **See 21.5 for more information.**	No	No	N/A
PigIdentities	Array	Element containing multiple PigIdentity elements. **See 21.3 for more information.**	N/A	N/A	Yes **

Keeper Element

XML Element	Type	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
FlockNumber	Integer		No	No	No
SlapMark	String (max 8 chars)		No	No	No
Name	String (max 60 chars)		No	No	Yes*
Address1	String (max 60 chars)		No	Yes	No
Address2	String (max 60 chars)		No	Yes	No
Address3	String (max 60 chars)		No	No	Yes*
Address4	String (max 60 chars)		No	No	Yes*
Postcode	String (max 9 chars)		No	No	Yes*
Tel	String (max 16 chars)		No	No	Yes*
Email	String (max 100 chars)		No	No	No
AssuranceNumber	Integer		No	No	No

Only required for consigning keeper.

TransportInformation Element

XML Element	Type	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
HaulierName	String (max 60 chars)		No	No	No
DriverName	String (max 60 chars)		No	No	No
RegistrationNumber	String (max 14 chars)		No	No	No
PermitNumber	String (max 10 chars)		No	No	No
ExpectedDuration	Integer	Approximate expected transport duration in hours	No	No	No
DepartureTime	Time		No	No	No
LoadingDateTime	DateTime		No	No	No
UnloadingDateTime	DateTime		No	No	No

FCI Element

XML Element	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
ConsignmentMedicines	String	No	No	No
IndividualMedicines	String	No	No	No
WithdrawalPeriodsMet	Integer - see WSDL for enumeration	No	No	No
InjuredPigs	Integer	No	No	No
TailBitePigs	Integer	No	No	No
PoorDoerPigs	Integer	No	No	No
HerniaPigs	Integer	No	No	No

XML Element	Notes	Required: incomplete sheep lot	Required: complete sheep lot	Required: pig lot
ZNCPScore	Integer (between 0 and 100)	No	No	No

PigIdentity Element

One attribute named “Tag” - free text (14 characters max)

TagReading Element

Each TagReading element requires the Timestamp attribute and a Type attribute of either “eid” or “manual”.

If the TagReading Type is “eid” then either a **Hex** attribute containing the 16 character hex representation of a 64 bit tag, or an **ISO24631** attribute containing the ISO24631 decimal representation of a tag. See example request **5.4.4** for examples of both formats being used.

If the TagReading Type is “manual” then only a Timestamp need be supplied.

FlockTag Element

The flock tag element represents a group of 1 or more animals for which only the flock number is known. For example, this may be used for visual recording of animals which have been tagged with an electronic slaughter tag - where only the flock number is visible. The FlockTag element requires **FlockNumber** and **TagCount** attributes.

Locations

All locations should be given as a CPH number formatted CC/PPP/HHHH. Sub-locations are not supported and should be removed before submission.

Movement Types

Movement Type	Description
3	ON/OFF (Farm move)
4	MART (CCP use only)
5	DEATH (Not abattoir)
6	TAGGED
8	ABATTOIR (CCP use only)
10	MANAGEMENT
12	ONLY TESTING

Pig Movement Examples

CreateCompleteLot

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.scoteid.com/api/">
   <soapenv:Header/>
   <soapenv:Body>
      <api:CreateCompleteLotRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.3.5</ApplicationVersion>
         <SchemaVersion>1.4</SchemaVersion>
         <Timestamp>2011-10-26</Timestamp>
         <Lot>
            <!--Optional:-->
            <ExternalReference>61AC91EE21B46C016BF0E2C6812CE06</ExternalReference>
            <LotNumber>0004</LotNumber>
            <LotDate>2011-10-23</LotDate>
            <MovementType>3</MovementType>
            <Species>3</Species>
            <!--Required unless using PigIdentities -->
            <BatchMark>SL6699</BatchMark>
            <!--Optional:-->
            <LotDescription>2</LotDescription>
            <!--Optional:-->
            <ArrivalDate>2011-10-25</ArrivalDate>
            <!--Optional:-->
            <DepartureDate>2011-10-25</DepartureDate>
            <HeadCount>10</HeadCount>
          <DOACount>1</DOACount>
            <DepartureLocation>66/999/6666</DepartureLocation>
            <ReadLocation>66/999/6666</ReadLocation>
            <DestinationLocation>33/444/5555</DestinationLocation>
            <!--Required:-->
            <DepartureKeeper>
               <SlapMark>SL6699</SlapMark>
               <Name>Early Morning Farm</Name>
               <Address1>Address 1</Address1>
               <Address3>Address 3</Address3>
               <Address4>Address 4</Address4>
               <Postcode>HG4 5DB</Postcode>
            </DepartureKeeper>
            <!--Optional:-->
            <TransportInformation>
               <HaulierName>Haulier name</HaulierName>
               <DriverName>Driver name</DriverName>
               <RegistrationNumber>AB12 34C</RegistrationNumber>
               <PermitNumber>Permit number</PermitNumber>
               <ExpectedDuration>2</ExpectedDuration>
               <DepartureTime>10:00</DepartureTime>
               <LoadingDateTime>2011-10-25T09:30</LoadingDateTime>
               <UnloadingDateTime>2011-10-25T11:30</UnloadingDateTime>
            </TransportInformation>
            <FCI>
               <WithdrawalPeriodsMet>0</WithdrawalPeriodsMet>
            </FCI>
            <PigIdentities>
               <PigIdentity Tag="123456"/>
               <PigIdentity Tag="234567"/>
            </PigIdentities>
         </Lot>
```

```
      </api:CreateCompleteLotRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
   <SOAP-ENV:Body>
      <ns1:CreateCompleteLotResponse>
         <Timestamp>2011-10-27T10:45:13+01:00</Timestamp>
         <Lot>
            <LotReference>12</LotReference>
            <ExternalReference>61AC91EE21B46C016BF0E2C6812CE06</ExternalReference>
            <LotNumber>0004</LotNumber>
            <LotDate>2011-10-23</LotDate>
            <MovementType>3</MovementType>
            <Species>3</Species>
            <BatchMark>SL6699</BatchMark>
            <LotDescription>2</LotDescription>
            <ArrivalDate>2011-10-25+01:00</ArrivalDate>
            <DepartureDate>2011-10-25+01:00</DepartureDate>
            <HeadCount>10</HeadCount>
            <ReadCount>0</ReadCount>
            <DOACount>1</DOACount>
            <DepartureLocation>66/999/6666</DepartureLocation>
            <ReadLocation>66/999/6666</ReadLocation>
            <DestinationLocation>33/444/5555</DestinationLocation>
            <DepartureKeeper>
               <SlapMark>SL6699</SlapMark>
               <Name>Early Morning Farm</Name>
               <Address1>Address 1</Address1>
               <Address3>Address 3</Address3>
               <Address4>Address 4</Address4>
               <Postcode>HG4 5DB</Postcode>
            </DepartureKeeper>
            <TransportInformation>
               <HaulierName>Haulier name</HaulierName>
               <DriverName>Driver name</DriverName>
               <RegistrationNumber>AB12 34C</RegistrationNumber>
               <ExpectedDuration>2</ExpectedDuration>
               <DepartureTime>10:00</DepartureTime>
               <LoadingDateTime>2011-10-25T10:30:00+01:00</LoadingDateTime>
               <UnloadingDateTime>2011-10-25T12:30:00+01:00</UnloadingDateTime>
            </TransportInformation>
            <FCI>
               <WithdrawalPeriodsMet>0</WithdrawalPeriodsMet>
            </FCI>
            <PigIdentities>
               <PigIdentity Tag="123456"/>
               <PigIdentity Tag="234567"/>
            </PigIdentities>
         </Lot>
      </ns1:CreateCompleteLotResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

UpdateCompleteLotByReference

An existing lot created using the above operation can be updated subsequently to replace existing data or add additional data. In the example below the DestinationKeeper is updated and the resulting lot is returned.

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.scoteid.com/api/">
   <soapenv:Header/>
   <soapenv:Body>
      <api:UpdateCompleteLotByReferenceRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.3.5</ApplicationVersion>
         <SchemaVersion>1.4</SchemaVersion>
         <Timestamp>2011-10-26</Timestamp>
         <LotReference>12</LotReference>
         <Lot>
            <DestinationKeeper>
               <Name>Porko Meats Ltd</Name>
               <Postcode>MK6 1AX</Postcode>
            </DestinationKeeper>
         </Lot>
      </api:UpdateCompleteLotByReferenceRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
   <SOAP-ENV:Body>
      <ns1:UpdateCompleteLotResponse>
         <Timestamp>2011-10-27T11:10:30+01:00</Timestamp>
         <Lot>
            <LotReference>12</LotReference>
            <ExternalReference>61AC91EE21B46C016BF0E2C6812CE06</ExternalReference>
            <LotNumber>0004</LotNumber>
            <LotDate>2011-10-23</LotDate>
            <MovementType>3</MovementType>
            <Species>3</Species>
            <BatchMark>SL6699</BatchMark>
            <LotDescription>2</LotDescription>
            <ArrivalDate>2011-10-25+01:00</ArrivalDate>
            <DepartureDate>2011-10-25+01:00</DepartureDate>
            <HeadCount>10</HeadCount>
            <ReadCount>0</ReadCount>
            <DOACount>1</DOACount>
            <VisuallyRead>false</VisuallyRead>
            <DepartureLocation>66/999/6666</DepartureLocation>
            <ReadLocation>66/999/6666</ReadLocation>
            <DestinationLocation>33/444/5555</DestinationLocation>
            <DepartureKeeper>
               <SlapMark>SL6699</SlapMark>
               <Name>Early Morning Farm</Name>
               <Address1>Address 1</Address1>
               <Address3>Address 3</Address3>
               <Address4>Address 4</Address4>
               <Postcode>HG4 5DB</Postcode>
            </DepartureKeeper>
            <DestinationKeeper>
               <Name>Porko Meats Ltd</Name>
               <Postcode>MK6 1AX</Postcode>
            </DestinationKeeper>
            <TransportInformation>
               <HaulierName>Haulier name</HaulierName>
               <DriverName>Driver name</DriverName>
               <RegistrationNumber>AB12 34C</RegistrationNumber>
               <ExpectedDuration>2</ExpectedDuration>
               <DepartureTime>10:00:00</DepartureTime>
               <LoadingDateTime>2011-10-25T10:30:00+01:00</LoadingDateTime>
```

```
            <UnloadingDateTime>2011-10-25T12:30:00+01:00</UnloadingDateTime>
         </TransportInformation>
         <FCI>
            <WithdrawalPeriodsMet>0</WithdrawalPeriodsMet>
         </FCI>
         <PigIdentities>
            <PigIdentity Tag="123456"/>
            <PigIdentity Tag="234567"/>
         </PigIdentities>
      </Lot>
   </ns1:UpdateCompleteLotResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetCompleteLotByReference

A previously completed lot can be fetched at any time using the LotReference returned by the initial creation.

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.scoteid.com/api/">
   <soapenv:Header/>
   <soapenv:Body>
      <api:GetCompleteLotByReferenceRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.3.4</ApplicationVersion>
         <SchemaVersion>1.4</SchemaVersion>
              <Timestamp>2011-08-20T11:16:05.718+01:00</Timestamp>
         <LotReference>12</LotReference>
      </api:GetCompleteLotByReferenceRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
   <SOAP-ENV:Body>
      <ns1:GetCompleteLotByReferenceResponse>
         <Timestamp>2011-10-27T10:57:44+01:00</Timestamp>
         <Lot>
            <LotReference>12</LotReference>
            <ExternalReference>61AC91EE21B46C016BF0E2C6812CE06</ExternalReference>
            <LotNumber>0004</LotNumber>
            <LotDate>2011-10-23</LotDate>
            <MovementType>3</MovementType>
            <Species>3</Species>
            <BatchMark>SL6699</BatchMark>
            <LotDescription>2</LotDescription>
            <ArrivalDate>2011-10-25+01:00</ArrivalDate>
            <DepartureDate>2011-10-25+01:00</DepartureDate>
            <HeadCount>10</HeadCount>
            <ReadCount>0</ReadCount>
            <DOACount>1</DOACount>
            <VisuallyRead>false</VisuallyRead>
            <DepartureLocation>66/999/6666</DepartureLocation>
            <ReadLocation>66/999/6666</ReadLocation>
            <DestinationLocation>33/444/5555</DestinationLocation>
            <DepartureKeeper>
               <SlapMark>SL6699</SlapMark>
               <Name>Early Morning Farm</Name>
```

```
            <Address1>Address 1</Address1>
            <Address3>Address 3</Address3>
            <Address4>Address 4</Address4>
            <Postcode>HG4 5DB</Postcode>
         </DepartureKeeper>
         <DestinationKeeper>
            <Name>Porko Meats Ltd</Name>
            <Postcode>MK6 1AX</Postcode>
         </DestinationKeeper>
         <TransportInformation>
            <HaulierName>Haulier name</HaulierName>
            <DriverName>Driver name</DriverName>
            <RegistrationNumber>AB12 34C</RegistrationNumber>
            <ExpectedDuration>2</ExpectedDuration>
            <DepartureTime>10:00:00</DepartureTime>
            <LoadingDateTime>2011-10-25T10:30:00+01:00</LoadingDateTime>
            <UnloadingDateTime>2011-10-25T12:30:00+01:00</UnloadingDateTime>
         </TransportInformation>
         <FCI>
            <WithdrawalPeriodsMet>0</WithdrawalPeriodsMet>
         </FCI>
         <PigIdentities>
            <PigIdentity Tag="123456"/>
            <PigIdentity Tag="234567"/>
         </PigIdentities>
      </Lot>
   </ns1:GetCompleteLotByReferenceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Scotch Potential Eligibility Cattle Checker (SPECC)

GetAnimalFAStatus

This webservice request forms part of the Scotch Potential Eligibility Cattle Checker. It takes a valid cattle eartag number as a parameter and returns a short string indicating whether the animal is Scotch Potentially Eligible (SPE), farm assured (FA) for a given number of days, or neither.
The service will also return additional fields (QMSStatusDescription, QMSStatusReason, QMSStatusShortDescription) which may be helpful when determining why an animal is not Scotch Eligible.
The three possible SPE outputs are shown below:

SPE	Scotch Potential Eligible
FA{n}	Denotes number of cumulative and successive days on QMS assured holdings, e.g. FA38, FA158, FA968.
NA	Any animals not falling into either of the two catagories above

When Schema Version 1.7.4+ is used the response will also include the BVD status of the animal.

QMSStatus enumeration

At the time of writing the values returned in the QMSStatus field are as follows (please refer to the WSDL for the most up to date information)

QMSStatus	QMSStatusShortDescription	QMSStatusDescription	QMSStatusReason
0		Animal not on database	N/A
1	SPE	Scotch Potential Eligible (SPE)	NA
2-9		**RESERVED**	**RESERVED**
10	NSE	Not Scotch Eligible (NSE)	Incomplete movement record
11-49		**RESERVED**	**RESERVED**
50	NSE	Not Scotch Eligible (NSE)	Current holding not QMS assured
51	NSE	Not Scotch Eligible (NSE)	Incomplete movement record
52-95	NSE	Not Scotch Eligible (NSE)	**RESERVED**
96	NSE	Not Scotch Eligible (NSE)	Birth holding not QMS assured
97	NSE	Not Scotch Eligible (NSE)	Not whole life QMS assured
98	NSE	Not Scotch Eligible (NSE)	Over 4 y/o or has calved
99	NSE	Not Scotch Eligible (NSE)	Not born in Scotland

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:api="http://api.scoteid.com/api/">
   <soapenv:Header/>
   <soapenv:Body>
      <api:GetAnimalFAStatusRequest>
       <ApplicationName>ScotEID Desktop</ApplicationName>
       <ApplicationVersion>1.3</ApplicationVersion>
       <ApplicationKey>YOURKEYHERE</ApplicationKey>
       <SchemaVersion>1.5.5</SchemaVersion>
       <Timestamp>2016-02-09T20:49:00</Timestamp>
       <AnimalID>UK580941700119</AnimalID>
      </api:GetAnimalFAStatusRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
   <SOAP-ENV:Body>
      <ns1:GetAnimalFAStatusResponse>
         <Timestamp>2016-04-25T11:08:19+01:00</Timestamp>
         <FAStatus>FA163</FAStatus>
         <QMSStatus>97</QMSStatus>
         <QMSStatusDescription>Not Scotch Eligible (NSE)</QMSStatusDescription>
         <QMSStatusReason>Not whole life QMS assured</QMSStatusReason>
         <QMSStatusShortDescription>NSE</QMSStatusShortDescription>
         <BSENRStatus>IR</BSENRStatus>
         <BVDCategory>2</BVDCategory>
      </ns1:GetAnimalFAStatusResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP faults

In addition to the standard global soap faults this request may also throw **InvalidAnimalIDFault** if supplied with a badly formed or invalid animal eartag number.

Cattle: Basic Web Service information

Background

The ScotEID cattle webservices were originally developed for the recording of within business movements as a replacement for the linked holding arrangement operated by CTS. This system was called ScotMoves and was introduced in 2017. This system allowed keepers who move cattle within their business to do so without having to report the movement to the Cattle Tracing System (CTS) whilst remaining compliant with Regulation (EC) 1760/2000.

In the first year of operation there were as many within business moves (½ million) recorded on ScotMoves as the total number of on farm movements recorded on CTS

The Business Rules for recording within business moves can be found here: http://www.gov.scot/Resource/0050/00506060.pdf.

In October 2021 the ScotMoves system was expanded to cover the registration of births, deaths and change of keepership movements of cattle. This system was called ScotMoves+.

All of the GB administrations are also planning to transition away from BCMS in the near future.

To support legacy systems and to ensure that there is no loss of traceability for animals moving across the border it seems likely that CTS/BCMS will remain in place until each administration has developed its own replacement system. During this period CTS/BCMS will continue to receive Scottish movement data from ScotEID but the web services and website is blocked for Scottish keepers.

Validations

The validations in place for each web service are described in the section for each web service. To maintain compatibility with CTS for the purposes of UK traceability the validations will largely follow the existing CTS validations although there are some additional validations to ensure data integrity.

Landless keepers (7000 codes)

Keepers who do not have a permanent piece of land are registered using a landless keeper CPH, also known as a 7000 code. These codes have the same format as a normal CPH number but the holding part of the CPH (county/parish/holding) code is in the 7000 range.

Because these 7000 codes do not relate to a physical location they cannot be used to register the initial location of animals born or brought into the keepership. They also cannot be used to record movements within a business. Landless keepers are required to nominate a default holding for animals which are brought into the business. This default will be used to set the initial location of brought in animals, unless a physical location has been provided in the web service.

For animals born on a 7000 code the initial location of the animal will be set to the location of the dam unless a physical location has been provided in the webservice.

Create & allocate requests

Generally each create or allocate request should have a unique Row attribute specified which is used to reference the appropriate result in the response.

Each result consists of a **Status** (success or error), a **MovementReference** (if successful and depending on the request), **Row** to indicate which movement the result corresponds to and potentially a collection of **Errors**.

Errors

An example error response is shown below. Error messages will usually return the submission row where the error was encountered, a severity level, a description of the problem encountered and a code for the error. Where possible the error codes will follow those currently used by CTS.

In some circumstances an error will be generated when some additional input from the keeper is required. The ScotEID helpdesk will be able to help in these cases.

Each error has a severity level, either “fatal” or “warning”. A “fatal” error means that the submission has been unsuccessful and has not been saved. A “warning” is an advisory that the data submitted may be incorrect or require further evaluation.

```
   <ResultStatus="error"Row="1">
      <Errors>
         <Error Field="DamID" Severity="fatal" Message="Not found on main holding"
Code="CTWS199" />
      </Errors>
   </Result>
   <ResultStatus="success"Row="2"/>
```

Cattle: Within business movements

Recording within business moves

SMCreateCattleMovements allows multiple movements between a keeper's main holding and additional holdings. The additional holdings must be registered on ScotEID. Additional holdings must be renewed annually.

The web service will return a unique movement reference number for every successfully recorded movement.

Example Request

```
<soapenv:Header/>
 <soapenv:Body>
    <api:SMCreateCattleMovementsRequest>
       <ApplicationName>ScotEID Desktop</ApplicationName>
       <ApplicationVersion>1.0</ApplicationVersion>
       <ApplicationKey><!-- your key here --></ApplicationKey>
       <SchemaVersion>1.7</SchemaVersion>
       <Timestamp>2016-12-2113:42:00</Timestamp>
       <Movements>
          <!--Zero or more repetitions:-->
          <Movement Row="1"
                    AnimalID="UK1234560001234"
                    DepartureLocation="12/123/1234"
                    DestinationLocation="23/234/2345"
                    MoveDate="2019-01-01"
                    UserReference="optional user ref" />
```

```
            <Movement Row="2"
                      AnimalID="UK1234560001236"
                      DepartureLocation="23/234/2345"
                      DestinationLocation="12/123/1234"
                      MoveDate="2019-02-01"
                      UserReference="Return home" />
            <Movement Row="3"
                      AnimalID="UK1234560001237"
                      DepartureLocation="12/123/1234"
                      DestinationLocation="23/234/2345"
                      MoveDate="2019-03-01"/>
         </Movements>
      </api:SMCreateCattleMovementsRequest>
   </soapenv:Body>
```

Example Response

```
  <SOAP-ENV:Body>
      <ns1:SMCreateCattleMovementsResponse>
         <Timestamp>2016-12-22T12:39:58Z</Timestamp>
         <Results>
            <Result Status="success" MovementReference="123" Row="1"/>
            <Result Status="success" MovementReference="124" Row="2">
               <Errors>
                  <Error Field="move_date"
Severity="warning"
Message="description of warning here" />
               </Errors>
            </Result>
            <Result Status="error" Row="3">
               <Errors>
                  <Error Field="move_date"
Severity="fatal"
Message="description of fatal error here"/>
               </Errors>
            </Result>
         </Results>
      </ns1:SMCreateCattleMovementsResponse>
   </SOAP-ENV:Body>
```

Validations

Field	**Validations**
AnimalID	Checksum validation Length <= 14

	Length > 6 Tag has been issued Animal is currently recorded on departure holding Duplicate movement check Animal movement history is complete Animal is recorded on users main CPH
DepartureLocation DestinationLocation	Is registered to user Holding is not a landless keeper (7000 code) Holding is not a mart abattoir or Vi unit
MoveDate	>= 2017-01-01 < + 20 DAYS

Cancel within business moves

This can be used to correct an incorrect upload by cancelling the incorrect movement and recreating.

Example Request

```
  <soapenv:Header/>
   <soapenv:Body>
      <api:SMCancelCattleMovementRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2016-12-2113:42:00</Timestamp>
         <MovementReference>1345</MovementReference>
      </api:SMCancelCattleMovementRequest>
   </soapenv:Body>
```

Example Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://api.scoteid.com/api/">
   <SOAP-ENV:Body>
      <ns1:SMCancelCattleMovementResponse>
         <Timestamp>2016-12-22T12:23:41Z</Timestamp>
         <Result Status="success"/>
      </ns1:SMCancelCattleMovementResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Validations

Field	Validations
MovementReference	MovementReference exists Movement was recorded by user

Allocate bought in cattle

Cattle which are recorded as moving to a keeper#s main CPH will automatically be allocated to the default additional holding that the keeper has chosen. If the keeper has not chosen a default additional holding then the animal(s) must be allocated to an initial holding on ScotMoves. This can be the main holding or one of the additional holdings that have been registered on ScotMoves.

This is to cater for the scenario where an animal(s) have been recorded moving to a keeper's main CPH but the animal actually moved directly to one of the keeper's additional holdings. If the animal physically moves to the keepers main holding first and then to an additional holding then the animal should be allocated to the main holding and a movement to the additional holding should then be recorded using **SMCreateCattleMovements**.

Please note that ScotEID.com website users are able to select a default holding for new cattle to be allocated to.

Multiple allocations can be submitted in a single request. Each allocation should have a unique Row attribute specified which is used to reference the appropriate result in the response.

Results have a status (success or error) and optionally a collection of errors with severity warning or fatal. Errors with the status warning are advisory only.

Example Request

```
<soapenv:Header/>
 <soapenv:Body>
    <api:SMAllocateCattleRequest>
       <ApplicationName>ScotEID Desktop</ApplicationName>
       <ApplicationVersion>1.0</ApplicationVersion>
       <ApplicationKey><!-- your key --></ApplicationKey>
       <SchemaVersion>1.7</SchemaVersion>
       <Timestamp>2016-12-22T12:16:27Z</Timestamp>
       <Allocations>
          <Allocation Row="1"
                      InitialCPH="65/123/1234"
                      AnimalID="UK543210123456"/>
          <Allocation Row="2"
                      InitialCPH="65/123/1234"
                      AnimalID="UK543210123557"/>
       </Allocations>
    </api:SMAllocateCattleRequest>
 </soapenv:Body>
```

Example Response

```
<SOAP-ENV:Body>
    <ns1:SMAllocateCattleResponse>
       <Timestamp>2016-12-22T12:16:38Z</Timestamp>
       <Results>
          <Result Status="error" Row="1">
             <Errors>
                <Error Field="AnimalID" Severity="fatal" Message="Not found on main
```

```
holding"/>
                </Errors>
             </Result>
             <Result Status="success" Row="2"/>
          </Results>
       </ns1:SMAllocateCattleResponse>
    </SOAP-ENV:Body>
```

Validations

Field	Validations
AnimalID	Checksum validation Length <= 14 Length > 6 Tag has been issued Animal movement history is not broken Animal is recorded on users main CPH
InitialCPH	Is registered to user Holding is not a landless keeper (7000 code)

Get Cattle Holdings

Retrieve a list of currently active additional holdings registered on ScotMoves along the main holding for each. Each holding will have CPH, MainCPH, StartDate and EndDate attributes.

Example Request

```
  <soapenv:Body>
      <api:SMGetCattleHoldingsRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2016-12-2113:42:00</Timestamp>
      </api:SMGetCattleHoldingsRequest>
   </soapenv:Body>
```

Example Response

```
  <SOAP-ENV:Body>
      <ns1:SMGetCattleHoldingsResponse>
         <Timestamp>2016-12-22T12:22:35Z</Timestamp>
```

```
        <Holdings>
           <Holding CPH="65/123/1234"
 MainCPH="65/123/1234"
 StartDate="2017-01-01Z"
 EndDate="2017-12-31Z"/>
           <Holding CPH="65/123/1235"
 MainCPH="65/123/1234"
 StartDate="2017-01-01Z"
 EndDate="2017-12-31Z"/>
        </Holdings>
     </ns1:SMGetCattleHoldingsResponse>
  </SOAP-ENV:Body>
```

Get cattle on all holdings

This request retrieves a list of cattle that are registered against the user's main CPH. Each animal will show the **InitialCPH** within ScotMoves that the animal has been allocated to and the currentCPH field shows the location the animal is currently recorded on.

When using schema version 1.7.7 or greater the basic animal data will also be returned in the request.

Example Request

```
  <soapenv:Header/>
   <soapenv:Body>
      <api:SMGetCattleOnHoldingsRequest>
         <ApplicationName>ScotEID Desktop</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7.7</SchemaVersion>
         <Timestamp>2016-12-22T11:06:24Z</Timestamp>
         <MainCPH>12/1234/1234</MainCPH>
      </api:SMGetCattleOnHoldingsRequest>
   </soapenv:Body>
```

Example Response

```
  <SOAP-ENV:Body>
      <ns1:SMGetCattleOnHoldingsResponse>
         <Timestamp>2016-12-22T12:17:55Z</Timestamp>
         <Animals>
            <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
CurrentCPH="65/123/1234" Sex="Female" DOB="2020-08-11" BirthCPH="65/123/1234"
BreedCode="AAX" DamID="UK543210000001" GeneticDamID="UK543210000001"
SireID="UK543210000002"/>
            <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
```

```
CurrentCPH="65/123/1234" Sex="Female" DOB="2020-09-11" BirthCPH="65/123/1234"
BreedCode="AAX" DamID="UK543210000003" GeneticDamID="UK543210000003"
SireID="UK543210000002"/>
         </Animals>
      </ns1:SMGetCattleOnHoldingsResponse>
   </SOAP-ENV:Body>
```

Cattle: Birth, Death & Movements out of keepership

Registering births

Overview

The webservice SMECreateCattleBirthRegistrationsRequest registers a calf on ScotEID.

BirthCPH is optional. BirthCPH should be the physical location on which the animal was born, please be aware that this may not be the same as the keeper's main CPH. If not provided, BirthCPH will be assumed to be the holding on which the dam is currently recorded on ScotEID. For keepers with a landless keeper code (7000 code) the 7000 code cannot be used as the birth location because it does not relate to a physical location.

All of the data fields are required except Sire ID. However, we would strongly encourage the Sire ID to be recorded where known. Sire ID is not subject to cross compliance and there are potentially significant benefits to the keeper and the industry in recording this information, as identified in the Beef 2020 report.

Example Request

The example belows shows two births being registered at the same time.

```
   <soapenv:Header/>
   <soapenv:Body>
      <api:SMECreateCattleBirthRegistrationsRequest>
         <ApplicationName>ScotEID desktop application</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
         <CattleBirthRegistrations>
            <!--Zero or more repetitions:-->
            <CattleBirthRegistration Row="1" AnimalID="UK543210012345"
MainCPH="79/435/0157" Sex="Male" DOB=”2020-03-02" AnimalAlias="Calf1"
BirthCPH="79/435/0157" BreedCode="AA" DamID="UK543210054322"
GeneticDamID="UK543210054322" SireID="UK590000123456"/>
            <CattleBirthRegistration Row="2" AnimalID="UK543210612345"
MainCPH="79/435/0157" Sex="Female" DOB="2020-03-01" AnimalAlias="Calf2"
BirthCPH="79/435/0157" BreedCode="AA" DamID="UK543210054321"
```

```
GeneticDamID="UK543210054321" SireID="UK590000123456"/>
         </CattleBirthRegistrations>
      </api:SMECreateCattleBirthRegistrationsRequest>
   </soapenv:Body>
```

Example Response

The Dam for the first calf has not been found as being registered to the main holding provided in the request so the registration of the first calf has failed. The second calf has been successfully registered.

```
<Results>
   <Result Status="error" Row="1">
      <Errors>
         <Error Field="DamID" Severity="fatal" Message="Not found on main holding"
Code="CTWS199" />
      </Errors>
   </Result>
   <ResultStatus="success" MovementReference="103584" Row="2"/>
</Results>
```

Validations

Field	**Validations**
AnimalID	Checksum validation Length <= 14 Length > 6 Tag has been issued Animal doesn't already exist
DOB	>= -27 DAYS
BreedCode	Breed code valid
Sex	Male or Female
DamID	ID validity check Is alive at DOB Present on main holding Sex <4 calves born in last 240 days Last calving > 240 days Age >15 months < 20 years
GeneticDamID	ID validity check Sex

SireID	If supplied: ID validity check Sex Age > 6 months
MainCPH	Is registered or valid for user
BirthCPH	Is registered or valid for user. Holding is not a landless keeper (7000 code)

Recording Movements

Both the departure and the destination main CPH are required for movements in order to satisfy the legal requirements for an online holdings register. For a change of business movement both keepers will be required to report the movement (as is presently the case).

Recording a within business move using this web service

For recording within business movements we would recommend using the SMCreateCattleMovements service. However, if you wish to use this web service for recording within business movements then DepartureMainCPH and DestinationMainCPH should be given as the keeper's main holding. This means the movement will be treated the same as the current ScotMoves movements web service and it will be treated as a within business move.

Allocating a ScotMoves initial location

DestinationLocation can be used to record the location the animal is being physically moved to. If the destination keeper has a ScotMoves account this will automatically allocate the animal to that location within their account, so the destination keeper no longer needs to make the allocation. It is envisioned that it will mainly be marts which wish to record this information on their customers behalf.

Movements to/from an Mart/Abattoir

For a movement to/from a Scottish mart or abattoir there is no requirement on the consigning/receiving keeper to record the movement. The mart/abattoir will report both ends of the movement on the keepers behalf.

Movements recorded by a Mart/Abattoir

For a movement of an animal from farm to abattoir/mart only one movement needs to be recorded. As opposed to BCMS which would require a seperate on and off movements to be recorded.

If the departure and arrival date are different then the departure date should be recorded against the optional DepartureDate attribute and the arrival date should be recorded against MoveDate. By default departure date and arrival date will be assumed to be the same.

Mart should supply the lot number in the UserReference field if possible.

Movements recorded by an agent

For movements recorded by an agent on behalf of a keeper either "on" or "off" should be recorded against the MoveDirection attribute. This will let ScotEID know whether to update the departure or destination holding.

Imports/Exports

For imports/exports the CPH of the port of entry/export can be used if known. Alternatively, the appendix contains a list of country codes which can be used.

Example Request

In the example below two moves have been reported.

In the first move an animal has been sold to the keeper registered to main CPH 79/435/0199. If the animal moved to one of the destination keeper's additional holdings then the keeper will be responsible for allocating the animal to the correct CPH.

In the second example the DestinationLocation of the animal has been recorded so the destination keeper will not need to allocate the animal.

```
   <soapenv:Header/>
   <soapenv:Body>
      <api:SMECreateCattleMovementsRequest>
         <ApplicationName>ScotEID desktop application</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
         <CattleMovements>
            <!--Zero or more repetitions:-->
            <CattleMovement Row="1" AnimalID="UK543210123456"
DepartureMainCPH="79/435/0157" DestinationMainCPH="79/435/0153" MoveDate="2020-03-01"
UserReference="Sold to Stuart"/>
            <CattleMovement Row="2" AnimalID="UK543210123457"
DepartureMainCPH="79/435/0157" DestinationMainCPH="89/702/0001"
DestinationLocation="79/435/0199" DepartureDate="2020-03-01" MoveDate="2020-03-02"
UserReference="Lot 1">
               <!--Optional: EID data-->
               <AnimalEID ISO24631="1 0 04 00 0 826054321023457" Hex="" TID=""
Timestamp=""/>
         </CattleMovements>
      </api:SMECreateCattleMovementsRequest>
   </soapenv:Body>
```

Example Response

```
  <SOAP-ENV:Body>
     <ns1:SMCreateCattleMovementsResponse>
        <Timestamp>2016-12-22T12:39:58Z</Timestamp>
        <Results>
           <Result Status="success" MovementReference="123" Row="1"/>
           <Result Status="success" MovementReference="124" Row="2">
              <Errors>
                 <Error Field="move_date"
```

```
Severity="warning"
Message="description of warning here" />
                </Errors>
             </Result>
          </Results>
       </ns1:SMCreateCattleMovementsResponse>
    </SOAP-ENV:Body>
```

Validations

AnimalID	Checksum validation Length <= 14 Length > 6 Tag has been issued Duplicate movement check
DepartureMainCPH DestinationMainCPH	Either is registered or valid for user
MoveDate	Is not in the future. Animal alive on movement date Animal registered on holding at movement date
DepartureDate	Is prior to MoveDate Is less than a week before MoveDate

Recording Deaths

Overview

This web service is used to record an on farm death or an abattoir death. The carcase information can be recorded by abattoirs. Abattoirs should record the kill number in the UserReference field.

UserRerence - Abattoirs should record the kill number here. Optional for other users.

Alongside the EID data it would be useful to get the Transponder ID for each transponder if possible[1].

The Beef 2020 report identified key benefits to the industry from recording weight & grade information centrally.

Example Request

The example request below shows a submission containing an on farm death and an abattoir death. The abattoir death includes the weight/grade/weight of the carcase.

```
      <api:SMECreateCattleDeathsRequest>
         <ApplicationName>ScotEID desktop application</ApplicationName>
```

[1] Transponder IDentifier. The TID is a read-only number written to the tag's microchip by the chip manufacturer to authenticate the tag.

```
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
         <CattleDeaths>
            <!--Zero or more repetitions:-->
            <CattleDeath Row="1" AnimalID="UK543210123457" MainCPH="79/435/0157"
DateOfDeath="2020-03-01" UserReference="On farm death">
<!--Optional: EID tag data-->
               <AnimalEID ISO24631="" Hex="8100CEBC1AE5D40426C61E18" TID="FFF23432345"
Timestamp="2019-09-21T12:51:11+01:00"/>
            </CattleDeath>

            <!-- Example below applicable to abattoirs includes EID and carcase
information -->
            <CattleDeath Row="2" AnimalID="UK543210123456" MainCPH="79/435/0157"
DateOfDeath="2020-03-01" UserReference="Abattoir death">
               <!--Optional: carcase information from abattoir-->
              <CarcaseClassification Category="B" Grade="R4L" Deadweight="350"
Liveweight=""/>
               <!--Optional: EID tag data-->
               <AnimalEID ISO24631="1 0 04 00 0 826054321023456" Hex="" TID=""
Timestamp="2019-09-21T12:51:11+01:00"/>
            </CattleDeath>
         </CattleDeaths>
      </api:SMECreateCattleDeathsRequest>
   </soapenv:Body>
```

Example Response

```
 <SOAP-ENV:Body>
      <ns1:CreateCattleDeathsResponse>
         <Timestamp>2016-12-22T12:39:58Z</Timestamp>
         <Results>
            <Result Status="success" MovementReference="UK543210123457" Row="1"/>
            <Result Status="success" MovementReference="UK543210123456" Row="2"/>
         </Results>
      </ns1:CreateCattleDeathsResponse>
   </SOAP-ENV:Body>
```

Validations

Field	**Validations**
AnimalID	Checksum validation Length <= 14

	Length > 6 Tag has been issued Duplicate movement check Animal is alive
MainCPH	Animal present on Main CPH Is registered or valid for user
DateOfDeath	>= -7 days

Animal Details

Will return the details for a registered animal. The animal must have been on the main CPH registered to the user's account.

Movements will be returned as three different structures: CTS movements, ScotEID change of keepership movements & ScotMoves movements. It may be necessary to add additional structures to cope with movements recorded by other administrations as they take over from BCMS.

Only within business (ScotMoves) relating to the user's main CPH will be shown. Within business moves recorded by other businesses will not be shown.

The row element indicates the correct order of ScotMoves movements.

If ReturnCalvings is set to true then the calving history of the animal is also returned. Querying the calving history is slow so it would be best to only request this data when needed.

Example Request

```
<soapenv:Header/>
<soapenv:Body>
   <api:SMEGetCattleDetailsRequest>
      <ApplicationName>ScotEID desktop application</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <ApplicationKey><!-- your key here --></ApplicationKey>
      <SchemaVersion>1.7</SchemaVersion>
      <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
      <AnimalID>UK543210123456</AnimalID>
      <ReturnCalvings>true</ReturnCalvings>
   </api:SMEGetCattleDetailsRequest>
</soapenv:Body>
```

Example Response

```
  <ns1:GetCattleDetailsResponse>
      <Timestamp>2020-04-01T09:08:28+01:00</Timestamp>
      <BasicCattleData>
         <AnimalID>UK510000000000</AnimalID>
```

```
            <AnimalEID ISO24631="1 0 04 00 0 8260510000000000" Hex=""
TID="ABCDEF0123456789"/>
            <Sex>Female</Sex>
            <DOB>2014-02-26Z</DOB>
            <MainCPH>79/435/0157</MainCPH>
            <LocationCPH>79/435/0157</LocationCPH>
            <BirthMainCPH>79/435/0157</BirthMainCPH>
            <BirthLocation>79/435/0157</BirthLocation>
            <BreedCode>LIMX</BreedCode>
            <DateOfDeath>2017-03-14Z</DateOfDeath>
            <DamID>UKAA0000000031</GeneticDamID>
            <GeneticDamID>UKAA0000000031</GeneticDamID>
            <SireID>AA1234</SireID>
            <PassportVersion>1</PassportVersion>
            <BVDCategory>1</BVDCategory>
         </BasicCattleData>
         <CTSMoves>
            <CTSMove  Row="1"
                      AnimalID="UK1234560001234"
                      CPH="03/234/2345"
                      MoveDate="2016-12-01"
                      MovementType="3" />
            <CTSMove  Row="2"
                      AnimalID="UK1234560001236"
                      CPH="65/123/1234"
                      MoveDate="2016-12-01"
                      MovementType="2" />
         </CTSMoves>
         <KeeperMoves>
            <ScotKeeperMove Row="1"
                            AnimalID="UK1234560001234"
                            DepartureMainCPH="65/123/1234"
                            DestinationMainCPH="88/234/2345"
                            MoveDate="2017-12-01"
                            UserReference="optional user ref" />
            <ScotKeeperMove Row="2"
                            AnimalID="UK1234560001236"
                            DepartureMainCPH="88/234/2345"
                            DestinationMainCPH="98/234/2345"
                            MoveDate="2018-12-01"/>
         </KeeperMoves>
         <ScotMoves>
            <Movement Row="1"
                      AnimalID="UK1234560001234"
                      DepartureLocation="98/234/2345"
                      DestinationLocation="98/234/1234"
```

```
                    MoveDate="2019-01-01"
                    UserReference="1st SM" />
         <Movement Row="2"
                    AnimalID="UK1234560001236"
                    DepartureLocation="98/234/1234"
                    DestinationLocation="98/234/9876"
                    MoveDate="2019-02-01"/>
         <Movement Row="3"
                    AnimalID="UK1234560001237"
                    DepartureLocation="98/234/9876"
                    DestinationLocation="98/234/2345"
                    MoveDate="2019-03-01"/>
      </ScotMoves>
      <Calvings>
         <Calving>
            <CalfID>UK500000000001</CalfID>
            <Date>2015-11-03Z</Date>
            <CPH>68/137/0001</CPH>
         </Calving>
         <Calving>
            <CalfID>UK500000000002</CalfID>
            <Date>2016-12-27Z</Date>
            <CPH>68/137/0001</CPH>
         </Calving>
      </Calvings>
   </ns1:GetCattleDetailsResponse>
</SOAP-ENV:Body>
```

Holding Details

This request will allow users to check if a Scottish holding has been registered as a main CPH for cattle. Some basic location address details will be returned. If the CPH is invalid an InvalidCPHFault will be thrown.

For English/Welsh holdings we will be unable to determine whether the holding has been registered as a main CPH for cattle.

Example Request

```
<soapenv:Header/>
<soapenv:Body>
   <api:GetHoldingDetailsRequest>
      <ApplicationName>ScotEID desktop application</ApplicationName>
      <ApplicationVersion>1.0</ApplicationVersion>
      <ApplicationKey><!-- your key here --></ApplicationKey>
      <SchemaVersion>1.7</SchemaVersion>
      <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
      <CPH>79/435/0157</CPH>
   </api:GetHoldingDetailsRequest>
```

```
   </soapenv:Body>
```

Example Response

```
<SOAP-ENV:Body>
   <ns1:GetHoldingDetailsResponse>
      <Timestamp>2016-12-22T12:39:58Z</Timestamp>
      <Country>Scotland</Country>
      <County>Inverness</County>
      <Parish>Kiltarlity</Parish>
      <Address>Kinerras</Address>
      <RegisteredCattleMainCPH>True</RegisteredCattleMainCPH>
     <RegisteredCattleAdditionalHolding>False</RegisteredCattleAdditionalHolding>
      <RegisteredSheepCPH>True</RegisteredSheepCPH>
      <RegisteredPigCPH>False</RegisteredPigCPH>
   </ns1:GetholdingDetailsResponse>
</SOAP-ENV:Body>
```

Unregistered Calf death

Data on unregistered calf deaths has been identified as important for driving industry improvements. This information will be anonymised, will not come under cross compliance, and will only be shared for research to benefit the industry.

If a calf dies before registration then the death should be reported using this web service. For calves which die after registration the cattle death web service should be used.

In order to avoid duplication, each unregistered dead calf should be given a unique CalfNo within the calving (e.g. triplets would be numbered 1,2,3).

If the calf has been tagged but is still unregistered please record the tag ID. If the calf has not been tagged please enumerate each dead calf to help avoid duplicates.

Example Request

```
  <soapenv:Header/>
   <soapenv:Body>
      <api:CreateUnregisteredCalfDeathsRequest>
        <ApplicationName>ScotEID desktop application</ApplicationName>
         <ApplicationVersion>1.0</ApplicationVersion>
         <ApplicationKey><!-- your key here --></ApplicationKey>
         <SchemaVersion>1.7</SchemaVersion>
         <Timestamp>2020-03-01T12:57:11+01:00</Timestamp>
         <!--Zero or more repetitions:-->
         <UnregisteredCalfDeaths>
            <!--Zero or more repetitions:-->
            <UnregisteredCalfDeath Row="1" TagID="" CalfNo="1" MainCPH="79/435/0157"
DOB="2020-03-01" DamID="UK500000123456" GeneticDamID="UK500000123456"
DateOfDeath="2020-03-01" SireID="UK590000123456" UserReference="Slipped calf"/>
```

```
            <UnregisteredCalfDeath Row="2" TagID="" CalfNo="2" MainCPH="79/435/0157"
DOB="2020-03-01" DamID="UK500000123456" GeneticDamID="UK500000123456"
DateOfDeath="2020-03-06" SireID="UK590000123456"/>
            <UnregisteredCalfDeath Row="3" TagID="UK500000054321" CalfNo="3"
MainCPH="79/435/0157" DOB="2020-03-01" DamID="UK500000123456"
GeneticDamID="UK500000123456" DateOfDeath="2020-03-07" SireID="UK590000123456"/>
         </UnregisteredCalfDeaths>
      </api:CreateUnregisteredCalfDeathsRequest>
   </soapenv:Body>
```

Example Response

```
  <SOAP-ENV:Body>
      <ns1:CreateUnregisteredCalfDeathsResponse>
         <Timestamp>2016-12-22T12:39:58Z</Timestamp>
         <Results>
            <Result Status="success" Row="1"/>
            <Result Status="success" Row="2"/>
            <Result Status="success" Row="3"/>
         </Results>
      </ns1:CreateUnregisteredCalfDeathsResponse>
   </SOAP-ENV:Body>
```

Tags issued to holding

The tags recorded as issued to the holding will be returned for the date range supplied. If a tag ID or partial ID is supplied then only those tags matching the partial ID will be returned. There may be limits placed on the period over the input date range.

Example Request

```
  <soapenv:Header/>
   <soapenv:Body>
      <api:SMEGetIssuedCattleTagsRequest>
         <ApplicationName>test</ApplicationName>
         <ApplicationVersion>1</ApplicationVersion>
         <ApplicationKey></ApplicationKey>
         <SchemaVersion>1.5</SchemaVersion>
         <Timestamp>2015-10-10T12:54:23.1234</Timestamp>
         <MainCPH>79/435/0157</MainCPH>
         <TagID>UK50000</TagID>
         <StartDate>2011-04-18</StartDate>
         <StopDate>2015-04-18</StopDate>
      </api:SMEGetIssuedCattleTagsRequest>
   </soapenv:Body>
```

Example Response

```
   <SOAP-ENV:Body>
      <ns1:SMEGetIssuedCattleTagsResponse>
         <Timestamp>2020-04-01T10:02:20+01:00</Timestamp>
         <TagGroupings>
            <TagGrouping Status="REPLACEMENT" TagType="P" TagCode="004_1"
EmanIdentifier="4" OrderDate="2012-06-26+01:00" MainCPH="794350157">
               <TagID>UK500000000001</TagID>
            </TagGrouping>
            <TagGrouping Status="NEW" TagType="B" TagCode="004_B" EmanIdentifier="4"
OrderDate="2013-03-15Z" MainCPH="794350157">
               <TagID>UK500000000002</TagID>
               <TagID>UK500000000003</TagID>
            </TagGrouping>
         </TagGroupings>
      </ns1:SMEGetIssuedCattleTagsResponse>
   </SOAP-ENV:Body>
```

BVD Requests

The BVD status requests documented below return BVDCategory, an enumerated type defined in the WSDL. The current categories are listed below but further information is available here:

http://www.scoteid.com/scoteid/bvd_tests/help

BVD Category	Description
1	Category 1: BVD Certified Negative
2	Category 2: BVD Screened Negative
3	Category 3: BVD Not Negative
4	Category 4: Suspected BVD PI

GetCPHBVDCategory

Look up the current BVD status of a holding.

Example request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://
api.scoteid.com/api/">
       <soapenv:Header/>
       <soapenv:Body>
              <api:GetCPHBVDCategoryRequest>
                     <ApplicationName>ScotEID Desktop</ApplicationName>
                     <ApplicationVersion>1.0</ApplicationVersion>
                     <SchemaVersion>1.5</SchemaVersion>
                     <Timestamp>2013-07-02T12:15:00</Timestamp> <CPH>11/222/3333</CPH>
              </api:GetCPHBVDCategoryRequest>
       </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://
api.scoteid.com/api/">
<SOAP-ENV:Body>
<ns1:GetCPHBVDCategoryResponse>
         <Timestamp>2013-07-08T15:25:58+01:00</Timestamp>
         <BVDCategory>3</BVDCategory>
</ns1:GetCPHBVDCategoryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

In addition to the standard global soap faults this request may also throw **InvalidCPHFault** and **InvalidAnimalIDFault**.

GetAnimalBVDCategory

Look up the current BVD status of an animal. This method requires a valid Animal ID and it is highly advised, to ensure the correct category is given, to provide the current location cph or in the case of marts/shows the cph of the holding of departure.

Example request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://
api.scoteid.com/api/">
	<soapenv:Header/>
	<soapenv:Body>
		<api:GetAnimalBVDCategoryRequest>
			<ApplicationName>ScotEID Desktop</ApplicationName>
			<ApplicationVersion>1.0</ApplicationVersion>
			<SchemaVersion>1.5</SchemaVersion>
			<Timestamp>2013-07-02T12:15:00</Timestamp>
			<AnimalID>UK123456400011</AnimalID>
			<CPH>11/222/3333</CPH>
		</api:GetAnimalBVDCategoryRequest>
	</soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://
api.scoteid.com/api/">
	<SOAP-ENV:Body>
		<ns1:GetAnimalBVDCategoryResponse>
			<Timestamp>2013-07-08T15:49:22+01:00</Timestamp>
			<BVDCategory>3</BVDCategory>
		</ns1:GetAnimalBVDCategoryResponse>
	</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

In addition to the standard global soap faults this request may also throw **InvalidCPHFault** and **InvalidAnimalIDFault**.

GetCattleRequiringBVDTesting

Currently limited to laboratories participating in the Scottish BVD eradication programme. This service will return all cattle requiring testing for a given period.
By default all live cattle in Scotland that reuire testinv will be returned for the dates given. The search can be narrowed down by providing animal ID or part of the animal ID. So for instance a country or herd number could be searched for. To search for cattle from specific EU countries the two letter code can be used. Please refer to the EU document "Country codes and protocol order". Birth CPH or Current CPH can also be searched on by providing part or the entire CPH

Example request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://
api.scoteid.com/api/">
	<soapenv:Header/>
	<soapenv:Body>
		<api:GetAnimalBVDCategoryRequest>
```

```
                    <ApplicationName>Test</ApplicationName>
                    <ApplicationVersion>1.0</ApplicationVersion>
                    <SchemaVersion>1.7.4</SchemaVersion>
                    <Timestamp>2018-07-02T12:15:00</Timestamp>
                    <StartDate>2017-01-01</StartDate>
                    <StopDate>2017-01-01</StopDate>
                    <!--Optional:-->
                    <AnimalID>IE</AnimalID>
                    <!--Optional:-->
                    <BirthCPH>2017-01-01</BirthCPH>
                    <!--Optional:-->
                    <CurrentCPH>2017-01-01</CurrentCPH>
                </api:GetAnimalBVDCategoryRequest>
        </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://
api.scoteid.com/api/">
        <SOAP-ENV:Body>
                <ns1:GetCattleRequiringBVDTestingResponse>
                        <Timestamp>2018-07-08T1:49:22+01:00</Timestamp>
                        <Animal>
                                <AnimalID>IE123456789012</AnimalID>
                                <Sex>Male</Sex>
                                <DOB>2016-01-01</DOB>
                                <BreedCode>BSH</BreedCode>
                                <CurrentCPH>79/435/0157</CurrentCPH>
                        </Animal>
                </ns1:GetCattleRequiringBVDTestingResponse >
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Soap Faults

In addition to the standard global soap faults this request may also throw **InvalidCPHFault.**

Export eligibility checker (All livestock)

EU animal health regulation (EU 2016/429) requires that animals or products of animal origin must come from holdings which have been regularly inspected for signs of disease.

To satisfy this requirement abattoirs must confirm that the last farm holding (i.e. not markets) on which the animals resided had an assured status (QMS) or had a Veterinary Attestation in place.

Please note that, according to current guidance, an assured status for any species would demonstrate compliance with the requirement for a regular vet visit, and would mean that all other species on the holding are export eligible regardless of whether that species is part of an assurance scheme.

Export eligibility is determined based on the date the animals left the last farm. For abattoirs checking market lots the move date should be given as the date on which the animals were moved to the market. If no move date is given the current export eligibility status of the holding will be returned.

Example request

```
    <soapenv:Header/>
    <soapenv:Body>
        <api:GetHoldingEUExportEligibilityRequest>
        <ApplicationVersion>0.1</ApplicationVersion>
        <SchemaVersion>1.7.9</SchemaVersion>
```

```
        <Timestamp>2023-12-07T13:02:42</Timestamp>
        <CPH>79/442/0010</CPH>
        <Species>2</Species>
        <!--Optional:-->
        <MoveDate>2023-12-01</MoveDate>
        </api:GetHoldingEUExportEligibilityRequest>
    </soapenv:Body>
```

Example Responses

Holding with a vet attestation in place

```
<SOAP-ENV:Body>
 <ns1:GetHoldingEUExportEligibilityResponse>
    <Timestamp>2023-12-08T17:18:06Z</Timestamp>
    <ExportEligible>true</ExportEligible>
    <ExportEligibleExpiryDate>2024-11-30Z</ExportEligibleExpiryDate>
    <EligibleVetVisitDetails>
       <VetAttestationNumber>2233445-79/435/0157-1124</VetAttestationNumber>
       <VisitDate>2023-11-27Z</VisitDate>
       <ExpiryDate>2024-11-30Z</ExpiryDate>
       <MRCVSNumber>2233445</MRCVSNumber>
    </EligibleVetVisitDetails>
 </ns1:GetHoldingEUExportEligibilityResponse>
</SOAP-ENV:Body>
```

Holding which is part of an assurance scheme

```
<SOAP-ENV:Body>
  <ns1:GetHoldingEUExportEligibilityResponse>
    <Timestamp>2023-10-25T11:06:54Z</Timestamp>
    <ExportEligible>true</ExportEligible>
    <ExportEligibleExpiryDate>2023-10-24Z</ExportEligibleExpiryDate>
    <EligibleAssuranceDetails>
      <AssuranceScheme>QMS</AssuranceScheme>
    </EligibleAssuranceDetails>
  </ns1:GetHoldingEUExportEligibilityResponse>
</SOAP-ENV:Body>
```

Appendices

Frequently used CPH numbers

Cattle Abattoirs

CPH	Business
66/062/8004	Scotbeef (Inverurie)
66/083/8000	Woodhead Bros (Turriff)
68/149/8005	Islay Abattoir
68/166/8000	Mull Slaughterhouse Ltd.
69/176/8002	Heathfield Abattoir (Ayr)
69/216/8000	Highland Meats Ltd
75/319/8005	Lockerbie Abattoir
80/471/8000	McIntosh Donald Ltd (Portlethen)
83/546/8500	Wishaw Abattoir
83/548/8500	Shotts Abattoir
85/586/8500	Miller (Grantown on Spey)
89/705/8500	ABP (Perth)
89/715/8500	Scotbeef (Bridge of Allan)
89/715/8501	Dunblane abattoir (DS Slaughterhouse Ltd.)
90/725/8004	Sandyford Abattoir
91/740/8500	Dingwall Abattoir (Munro's)
91/755/8500	Stornoway Abattoir
94/875/8011	Shetland Abattoir

Cattle Marts

CPH	Business
08/067/8019	Borderway Mart
08/181/8000	Longtown Market
66/062/8001	Thainstone Mart (ANM)
66/077/8000	Huntly Mart
67/102/8001	Forfar Mart
68/149/8001	Bridgend Mart
68/168/8001	Tiree Mart
68/171/8001	Dalmally Mart
68/174/8000	Oban Mart
68/174/8001	Moleigh Mart
69/176/8004	Ayr Mart (CW Ltd)
73/282/8600	Quoybrae Mart (ANM)
75/300/8001	Dumfries Mart
75/319/8004	Lockerbie Market
79/455/8600	Fort William Mart (D&HM)
79/457/8600	Lochmaddy Mart (D&HM)

79/461/8600	Portree Mart (D&HM)
79/465/8601	Lochboisdale Mart
82/492/8004	Castle Douglas Mart (Wallets)
83/536/8300	Lanark Mart (L&S)
87/620/8007	Orkney Mart
89/718/8008	Stirling Mart (UA)
91/740/8600	Dingwall Mart(D&HM)
91/755/8600	Stornoway Mart
92/769/8005	Newcastleton Mart
92/796/8003	St Boswells Mart (Swans)
94/875/8001	Shetland Mart
95/808/8301	Stirling Mart (Caledonian Marts)
98/856/8003	Newton Stewart Mart (CW Ltd)

Cattle Shows

CPH	Show
66/005/8004	Echt Show
66/030/8004	New Deer Show
66/083/8020	Turriff Show
67/115/8000	Kirriemuir Show
68/138/8008	Sunart Agricultural Show
68/149/8000	Islay Show
68/152/8001	Kintyre Show
68/161/8000	Argyll Show
68/165/8000	Bunessan Show
68/166/8001	Salen Show
68/168/8003	Tiree Agricultural Society Show
68/169/8003	Lorn Show
68/170/8000	Appin Showground
68/171/8000	Dalmally Showground
69/176/8012	Ayr Show
69/176/8017	Ayrhsire County Rally
69/178/8003	Dundonald Show
69/182/8004	Coylton Show
69/184/8003	Dalrymple Show
69/186/8003	Catrine Cattle Show
69/186/8004	Lawrie Chairman
69/188/8005	New Cumnock Show
69/189/8003	Ochiltree Show
69/201/8004	Beith Show
69/202/8003	Dalry Show
69/205/8003	Parker Sec West Scotland Dairy Show
69/206/8006	Kilmaurs Show

69/209/8010	Stewarton Show
69/213/8003	Straiton Show
69/214/8005	Newmilns Show
70/240/8002	Keith Show
71/254/8003	Berwickshire County Show
72/274/8004	Arran Show Ground
72/277/8001	Bute Show Rothesay
73/286/8009	Latheron Show
74/290/8005	Stirling Show
75/300/8006	Dumfries & Lockerbie Show
75/326/8002	Moffat Show
76/348/8000	Drymen Show
77/360/8002	East Lothian Show
78/381/8000	Fife Show
78/394/8000	Central & West Fife Agricultural Show
79/454/8006	Lochaber Show
79/457/8005	Hosta Showground
79/463/8001	Skyeportree Show
79/465/8005	South Uist Show
80/466/8001	Fettercairn Show
80/482/8000	Banchory Show
82/491/8001	Stewartry Show
83/518/8000	Biggar Show
83/530/8001	Lesmahagow Show
83/532/8006	Carnwath Show
83/538/8008	Strathaven Agric. Exposition
83/544/8005	East Kilbride Show
84/582/8011	Agriscot
84/582/8012	Royal Highland Show
85/586/8000	Granton Show
86/609/8010	Nairn Show
87/620/8010	County Show
87/629/8010	Sanday Agricultural Show
87/630/8010	Shapinsey Show
87/631/8010	South Ronaldsay & Burray Show
88/647/8005	Peebles Show
89/662/8000	Alyth Show
89/671/8000	Strathardle & Blairgowrie Show
89/677/8000	Atholl &bradalbane Agricultural Show
89/705/8000	Perthshire Agr Soc
89/712/8000	Braco Show
89/716/8000	Killin Show
89/717/8001	Doune & Dunblane Show
90/723/8000	Neilston Show

90/729/8003	Houston Show
90/730/8001	Kilbarchan Show
90/733/8001	Largs Show
90/734/8001	Kilmacolm Show
91/755/8000	Point Show
91/762/8001	Black Isle Show
92/785/8001	Border Union Agri. Show
94/870/8010	Voe Show
94/872/8010	Cunningsburgh Show
94/886/8010	Unst Show, Shetland
94/887/8010	Walls & District Agricultural Show
94/891/8010	Yell Show
95/808/8001	Scottish Premier Meat Exhibition
95/822/8000	Gargunnock Show
98/859/8001	Wigtown Show

CTS movement types

Movement Type	Movement Number	Reason For Use
Birth	0	Birth Movement
Import	1	Automatically created by CTS, equal to a birth but for imported animals, can only be changed by amending Passport details.
Normal ON	2	Used when a normal ON movement is entered, either from SMC, SIS, CTS on-line, CTS Web, CTS Self Service line, Manual Entry.
Normal OFF	3	Used when a normal OFF movement is entered, either from SMC, SIS, CTS on-line, CTS Web, CTS Self Service Line Manual Entry.
Placement	10	For use on CORs only, used for creating an ON movement when the movement date is unknown.
Displacement	11	For use on CORs for creating an OFF movement when date of movement is unknown and the COR has not been re-issued to another keeper. For use on CPP1s for creating an OFF movement from birth/last known location when the current keeper is unknown and the passport has not been replaced or re-issued to another keeper. This removes an animal from Animals in Location History.
Infer ON	12	Auto-generated by CTS where missing ON movement to slaughter houses are detected.
Administrative ON OFF	42 43	The Passport Operations department uses this when a keeper's holding number changes. All of the animals are then 'Admin moved' from the old holding to the new holding. The use of Admin moves reflects the fact that the animals themselves have not actually moved.
Death	7	Used to register an animal's death on CTS or an 'admin' death.
Summary ON	22	Used when entering movements from the movement summary section of a passport.
Summary OFF	23	Used when entering movements from the movement summary section of a passport.
Determined ON	32	Entered to replace missing movement where the circumstances meet the rules
Determined OFF	33	Entered to replace missing movement where the circumstances meet the rules
Inferred ON	52	Automatically created by CTS when animals have a movement anomaly involving missing movements. Works only within specific criteria.
Inferred OFF	53	Automatically created by CTS when animals have a movement anomaly involving missing movements. Works only within specific criteria.

Country codes

Code	Country
99/000/0004	Afghanistan (AF)
99/000/0008	Albania, People's Socialist Republic of (AL)
99/000/0010	Antarctica (the territory South of 60 deg S) (AQ)
99/000/0012	Algeria, People's Democratic Republic of (DZ)
99/000/0016	American Samoa (AS)
99/000/0020	Andorra, Principality of (AD)
99/000/0024	Angola, Republic of (AO)
99/000/0028	Antigua and Barbuda (AG)
99/000/0031	Azerbaijan, Republic of (AZ)
99/000/0032	Argentina, Argentine Republic (AR)
99/000/0036	Australia, Commonwealth of (AU)
99/000/0040	Austria (AT)
99/000/0044	Bahamas, Commonwealth of the (BS)
99/000/0048	Bahrain, Kingdom of (BH)
99/000/0050	Bangladesh, People's Republic of (BD)
99/000/0051	Armenia (AM)
99/000/0052	Barbados (BB)
99/000/0056	Belgium (BE)
99/000/0060	Bermuda (BM)
99/000/0064	Bhutan, Kingdom of (BT)
99/000/0068	Bolivia, Republic of (BO)
99/000/0070	Bosnia and Herzegovina (BA)
99/000/0072	Botswana, Republic of (BW)
99/000/0074	Bouvet Island (Bouvetoya) (BV)
99/000/0076	Brazil, Federative Republic of (BR)
99/000/0084	Belize (BZ)
99/000/0086	British Indian Ocean Territory (Chagos Archipelago) (IO)
99/000/0090	Solomon Islands (was British Solomon Islands) (SB)
99/000/0092	British Virgin Islands (VG)
99/000/0096	Brunei Darussalam (BN)
99/000/0100	Bulgaria (BG)
99/000/0104	Myanmar (was Burma) (MM)
99/000/0108	Burundi, Republic of (BI)
99/000/0112	Belarus (BY)
99/000/0116	Cambodia, Kingdom of (was Khmer Republic/Kampuchea, Democrat
99/000/0120	Cameroon, United Republic of (CM)
99/000/0124	Canada (CA)
99/000/0132	Cape Verde, Republic of (CV)
99/000/0136	Cayman Islands (KY)
99/000/0140	Central African Republic (CF)

99/000/0144	Sri Lanka, Democratic Socialist Republic of (was Ceylon) (L
99/000/0148	Chad, Republic of (TD)
99/000/0152	Chile, Republic of (CL)
99/000/0156	China, People's Republic of (CN)
99/000/0158	Taiwan, Province of China (TW)
99/000/0162	Christmas Island (CX)
99/000/0166	Cocos (Keeling) Islands (CC)
99/000/0170	Colombia, Republic of (CO)
99/000/0174	Comoros, Union of the (KM)
99/000/0175	Mayotte (YT)
99/000/0178	Congo, People's Republic of (CG)
99/000/0180	Congo, Democratic Republic of (was Zaire) (CD)
99/000/0184	Cook Islands (CK)
99/000/0188	Costa Rica, Republic of (CR)
99/000/0191	Hrvatska (Croatia) (HR)
99/000/0192	Cuba, Republic of (CU)
99/000/0196	Cyprus (CY)
99/000/0203	Czech Republic (CZ)
99/000/0204	Benin (was Dahomey), People's Republic of (BJ)
99/000/0208	Denmark (DK)
99/000/0212	Dominica, Commonwealth of (DM)
99/000/0214	Dominican Republic (DO)
99/000/0218	Ecuador, Republic of (EC)
99/000/0222	El Salvador, Republic of (SV)
99/000/0226	Equatorial Guinea, Republic of (GQ)
99/000/0231	Ethiopia (ET)
99/000/0232	Eritrea (ER)
99/000/0233	Estonia (EE)
99/000/0234	Faeroe Islands (FO)
99/000/0238	Falkland Islands (Malvinas) (FK)
99/000/0239	South Georgia and the South Sandwich Islands (GS)
99/000/0242	Fiji, Republic of the Fiji Islands (FJ)
99/000/0246	Finland (FI)
99/000/0248	Aland Islands
99/000/0250	France, (FR)
99/000/0254	French Guiana (GF)
99/000/0258	French Polynesia (PF)
99/000/0260	French Southern Territories (TF)
99/000/0262	Djibouti, Republic of (was French Afars and Issas) (DJ)
99/000/0266	Gabon, Gabonese Republic (GA)
99/000/0268	Georgia (GE)
99/000/0270	Gambia, Republic of the (GM)
99/000/0275	Palestinian Territory, Occupied (PS)
99/000/0276	Germany (DE)

99/000/0288	Ghana, Republic of (GH)
99/000/0292	Gibraltar (GI)
99/000/0296	Kiribati, Republic of (was Gilbert Islands) (KI)
99/000/0300	Greece (GR)
99/000/0304	Greenland (GL)
99/000/0308	Grenada (GD)
99/000/0312	Guadaloupe (GP)
99/000/0316	Guam (GU)
99/000/0320	Guatemala, Republic of (GT)
99/000/0324	Guinea, Revolutionary People's Rep'c of (GN)
99/000/0328	Guyana, Republic of (GY)
99/000/0332	Haiti, Republic of (HT)
99/000/0334	Heard and McDonald Islands (HM)
99/000/0336	Holy See (Vatican City State) (VA)
99/000/0340	Honduras, Republic of (HN)
99/000/0344	Hong Kong, Special Administrative Region of China (HK)
99/000/0348	Hungary (HU)
99/000/0352	Iceland, Republic of (IS)
99/000/0356	India, Republic of (IN)
99/000/0360	Indonesia, Republic of (ID)
99/000/0364	Iran, Islamic Republic of (IR)
99/000/0368	Iraq, Republic of (IQ)
99/000/0372	Ireland (IE)
99/000/0376	Israel, State of (IL)
99/000/0380	Italy (IT)
99/000/0384	Cote D'Ivoire, Ivory Coast, Republic of the (CI)
99/000/0388	Jamaica (JM)
99/000/0392	Japan (JP)
99/000/0398	Kazakhstan, Republic of (KZ)
99/000/0400	Jordan, Hashemite Kingdom of (JO)
99/000/0404	Kenya, Republic of (KE)
99/000/0408	Korea, Democratic People's Republic of (KP)
99/000/0410	Korea, Republic of (KR)
99/000/0414	Kuwait, State of (KW)
99/000/0417	Kyrgyz Republic (KG)
99/000/0418	Lao People's Democratic Republic (LA)
99/000/0422	Lebanon, Lebanese Republic (LB)
99/000/0426	Lesotho, Kingdom of (LS)
99/000/0428	Latvia (LV)
99/000/0430	Liberia, Republic of (LR)
99/000/0434	Libyan Arab Jamahiriya (LY)
99/000/0438	Liechtenstein, Principality of (LI)
99/000/0440	Lithuania (LT)
99/000/0442	Luxembourg (LU)

99/000/0446	Macao, Special Administrative Region of China (MO)
99/000/0450	Madagascar, Republic of (MG)
99/000/0454	Malawi, Republic of (MW)
99/000/0458	Malaysia (MY)
99/000/0462	Maldives, Republic of (MV)
99/000/0466	Mali, Republic of (ML)
99/000/0470	Malta (MT)
99/000/0474	Martinique (MQ)
99/000/0478	Mauritania, Islamic Republic of (MR)
99/000/0480	Mauritius (MU)
99/000/0484	Mexico, United Mexican States (MX)
99/000/0492	Monaco, Principality of (MC)
99/000/0496	Mongolia, Mongolian People's Republic (MN)
99/000/0498	Moldova, Republic of (MD)
99/000/0499	Montenegro
99/000/0500	Montserrat (MS)
99/000/0504	Morocco, Kingdom of (MA)
99/000/0508	Mozambique, People's Republic of (MZ)
99/000/0512	Oman, Sultanate of (was Muscat and Oman) (OM)
99/000/0516	Namibia (NA)
99/000/0520	Nauru, Republic of (NR)
99/000/0524	Nepal, Kingdom of (NP)
99/000/0528	Netherlands (NL)
99/000/0530	Netherlands Antilles (AN)
99/000/0531	Curacao
99/000/0533	Aruba (AW)
99/000/0534	Sint Maarten (Dutch part)
99/000/0535	Bonaire, Sint Eustatius and Saba
99/000/0540	New Caledonia (NC)
99/000/0548	Vanuatu (was New Hebrides) (VU)
99/000/0554	New Zealand (NZ)
99/000/0558	Nicaragua, Republic of (NI)
99/000/0562	Niger, Republic of the (NE)
99/000/0566	Nigeria, Federal Republic of (NG)
99/000/0570	Niue, Republic of (NU)
99/000/0574	Norfolk Island (NF)
99/000/0578	Norway, Kingdom of (NO)
99/000/0580	Northern Mariana Islands (MP)
99/000/0581	United States Minor Outlying Islands (UM)
99/000/0583	Micronesia, Federated States of (FM)
99/000/0584	Marshall Islands (MH)
99/000/0585	Palau (PW)
99/000/0586	Pakistan, Islamic Republic of (PK)
99/000/0591	Panama, Republic of (PA)

99/000/0598	Papua New Guinea (PG)
99/000/0600	Paraguay, Republic of (PY)
99/000/0604	Peru, Republic of (PE)
99/000/0608	Philippines, Republic of the (PH)
99/000/0612	Pitcairn Island (PN)
99/000/0616	Poland (PL)
99/000/0620	Portugal (PT)
99/000/0624	Guinea-Bissau, Republic of (was Portuguese Guinea) (GW)
99/000/0626	Timor-Leste, Democratic Republic of (TL)
99/000/0630	Puerto Rico (PR)
99/000/0634	Qatar, State of (QA)
99/000/0638	Reunion (RE)
99/000/0642	Romania (RO)
99/000/0643	Russian Federation (RU)
99/000/0646	Rwanda, Rwandese Republic (RW)
99/000/0654	St. Helena (SH)
99/000/0659	St. Kitts and Nevis (KN)
99/000/0660	Anguilla (AI)
99/000/0662	St. Lucia (LC)
99/000/0663	Saint Martin (French part)
99/000/0666	St. Pierre and Miquelon (PM)
99/000/0670	St. Vincent and the Grenadines (VC)
99/000/0674	San Marino, Republic of (SM)
99/000/0678	Sao Tome and Principe, Democratic Republic of (ST)
99/000/0682	Saudi Arabia, Kingdom of (SA)
99/000/0686	Senegal, Republic of (SN)
99/000/0688	Serbia
99/000/0690	Seychelles, Republic of (SC)
99/000/0694	Sierra Leone, Republic of (SL)
99/000/0702	Singapore, Republic of (SG)
99/000/0703	Slovakia (Slovak Republic) (SK)
99/000/0704	Viet Nam, Socialist Republic of (was Democratic Republic of
99/000/0705	Slovenia (SI)
99/000/0706	Somalia, Somali Republic (SO)
99/000/0710	South Africa, Republic of (ZA)
99/000/0716	Zimbabwe (was Southern Rhodesia) (ZW)
99/000/0724	Spain (ES)
99/000/0728	South Sudan
99/000/0729	Sudan
99/000/0732	Western Sahara (was Spanish Sahara) (EH)
99/000/0736	Sudan, Democratic Republic of the (SD)
99/000/0740	Suriname, Republic of (SR)
99/000/0744	Svalbard & Jan Mayen Islands (SJ)
99/000/0748	Swaziland, Kingdom of (SZ)

99/000/0752	Sweden (SE)
99/000/0756	Switzerland, Swiss Confederation (CH)
99/000/0760	Syrian Arab Republic (SY)
99/000/0762	Tajikistan (TJ)
99/000/0764	Thailand, Kingdom of (TH)
99/000/0768	Togo, Togolese Republic (TG)
99/000/0772	Tokelau (Tokelau Islands) (TK)
99/000/0776	Tonga, Kingdom of (TO)
99/000/0780	Trinidad and Tobago, Republic of (TT)
99/000/0784	United Arab Emirates (was Trucial States) (AE)
99/000/0788	Tunisia, Republic of (TN)
99/000/0792	Turkey, Republic of (TR)
99/000/0795	Turkmenistan (TM)
99/000/0796	Turks and Caicos Islands (TC)
99/000/0798	Tuvalu (was part of Gilbert & Ellice Islands) (TV)
99/000/0800	Uganda, Republic of (UG)
99/000/0804	Ukraine (UA)
99/000/0807	Macedonia, the former Yugoslav Republic of (MK)
99/000/0818	Egypt, Arab Republic of (EG)
99/000/0831	Guernsey
99/000/0832	Jersey
99/000/0833	Isle of Man
99/000/0834	Tanzania, United Republic of (TZ)
99/000/0840	United States of America (US)
99/000/0850	US Virgin Islands (VI)
99/000/0854	Burkina Faso (was Upper Volta) (BF)
99/000/0858	Uruguay, Eastern Republic of (UY)
99/000/0860	Uzbekistan (UZ)
99/000/0862	Venezuela, Bolivarian Republic of (VE)
99/000/0876	Wallis and Futuna Islands (WF)
99/000/0882	Samoa, Independent State of (was Western Samoa) (WS)
99/000/0887	Yemen (YE)
99/000/0891	Serbia and Montenegro (CS)
99/000/0894	Zambia, Republic of (ZM)