

ScotMoves+ web service guide

Scottish Cattle Traceability System

ScotEID

v1.1.2

07/09/2020

Revision history

Version	Description of changes	Date
1.0	Initial version	11/05/2020
1.1	Added test environment details	17/06/2020
1.1.1	Further information on web services	18/06/2020
1.1.2	Testing section update to reflect updates to testing server WSDL update weights updated from int to float	31/08/2020

Revision history	2
Overview	5
Background	5
Transition from CTS	5
How the new system will operate	5
Future developments	5
ScotMoves	6
What is ScotMoves?	6
Further information on ScotMoves	6
Web Services Overview	6
Web services available	6
Validations	7
Create & allocate requests	7
Errors	7
Currently available web services	8
Record within business moves	8
Example Request	8
Example Response	9
Validations	9
Cancel within business moves	10
Example Request	10
Example Response	10
Validations	11

Allocate brought-in cattle	11
Example Request	11
Example Response	12
Validations	12
Get Cattle Holdings	13
Example Request	13
Example Response	13
Get cattle on holding	14
Example Request	14
Example Response	14
Get cattle on all holdings	14
Example Request	15
Example Response	15
ScotMoves+ web services	15
Registering births	15
Overview	15
Example Request	16
Example Response	16
Validations	17
Recording Movements	17
Recording a within business move using this web service	18
Allocating a ScotMoves initial location	18
Movements to/from mart/abattoir	18
Imports/Exports	18
Example Request	18
Example Response	19
Validations	19
Recording Deaths	20
Overview	20
Example Request	20
Example Response	21
Validations	21
Animal Details	22
Example Request	22
Example Response	22
Holding Details	24
Example Request	25
Example Response	25

Unregistered Calf death	25
Example Request	26
Example Response	26
Tags issued to holding	27
Example Request	27
Example Response	27
Testing	29
Data Dictionary	30
Descriptions of data fields	30
Glossary of acronyms	31
Appendices	32
Frequently used CPH numbers	32
Cattle Abattoirs	32
Cattle Marts	32
Cattle Shows	33
CTS movement types	36
Country codes	37

Overview

This documentation is intended as a companion piece to the ScotEID web service API documentation. It provides a greater level of practical detail about the web services for the ScotMoves+ System which will assume many of the cattle reporting responsibilities, currently managed by CTS.

Background

Scottish Government plan to transition the recording of cattle movements away from BCMS. As part of the transition much of the functionality currently provided by BCMS will move to ScotEID. This document describes the web services which will be developed to replace those currently provided by BCMS

Transition from CTS

All of the GB administrations are planning to transition away from BCMS in the near future. Given current circumstances the timelines are difficult to predict.

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 for a period of time after each administration has developed its own replacement system. During this period CTS/BCMS may continue to receive Scottish movement data from ScotEID but the web services and website is likely to be blocked for Scottish keepers.

Because CTS is likely to continue to be needed for a period of time after the transition it is important that the systems are compatible. This means that many of the data requirements for cattle movements are unlikely to change significantly. For instance the main holding number will continue to be used to identify a keeper.

How the new system will operate

ScotMoves will be expanded to cover the registration of births, deaths and movements of cattle. Because reporting requirements remain the same the system will operate in a very similar manner to the current CTS system in order to maintain compatibility. The existing ScotMoves web services for within business movements will remain the same.

Future developments

Scottish Government will be consulting on the introduction of EID for cattle, removal of passports, pre-notification and other possible changes to the way movements are recorded.

This may lead to changes which will require updates to the proposed web services. A date for the consultation and the legislative changes have not yet been set. However, it is unlikely that there will be any significant changes until a significant period after births, deaths and movement recording have moved to ScotEID.

ScotMoves

What is ScotMoves?

ScotMoves was introduced in 2017 as the Scottish replacement for the linked holding arrangement operated by CTS. It allows 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. Births, deaths and movements involving a change of keepership are still reported to CTS.

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.

Scottish Government's intention is that the existing ScotMoves system be expanded to facilitate the recording of births, deaths and change of keepership movements.

Further information on ScotMoves

More information about ScotMoves can be found in the link below.

<https://www.scoteid.com/scoteid/wbm/ScotMoves/guidance>

Web Services Overview

The existing ScotMoves web service will remain in place. Births, deaths and movements outside of keepership are being added as part of ScotMoves+

Web services available

ScotMoves production web service:

- Within business moves
 - SMCreateCattleMovements
 - SMCancelCattleMovement
- Allocate brought-in cattle
 - SMAAllocateCattle
- Get holdings
 - SMGetCattleHoldings
- Get cattle on holding

- SMGetCattleOnHolding
- Get cattle on holdings
 - SMGetCattleOnHoldings

ScotMoves+ draft web services:

- Calf registration
 - SMECreateCattleBirthRegistrationsRequest
- Cattle death
 - SMECreateCattleDeathsRequest
- Cattle movement (change of keepership)
 - SMECreateCattleMovementsRequest
- Animal details
 - SMEGetCattleDetailsRequest
- Holding Details
 - GetHoldingDetailsRequest
- Tags issued to holding
 - SMEGetIssuedCattleTagsRequest
- Unregistered calf death
 - SMECreateUnregisteredCalfDeathsRequest
- Carcase disposal*
 - SMECarcaseDisposalStructure (just structure at moment)

Validations

The validations in place for the existing ScotMoves web services 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.

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 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"/>
```

Currently available web services

Record 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>2016-12-21T13:42:00</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 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>2016-12-21 13:42:00</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 brought-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:SMAAllocateCattleRequest>
</soapenv:Body>

```

Example Response

```

<SOAP-ENV:Body>
  <ns1:SMAAllocateCattleResponse>
    <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:SMAAllocateCattleResponse>
</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

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>2016-12-21T13:42:00</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 holding

This request retrieves a list of cattle that are on the specified holding according to ScotMoves. Note that fetching a list of cattle on your main holding will only return a list of cattle on that holding on ScotMoves which will often be a subset of the cattle on that holding according to CTS. 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.

Example Request

```
<soapenv:Header/>
<soapenv:Body>
  <api:SMGetCattleOnHoldingRequest>
    <ApplicationName>ScotEID Desktop</ApplicationName>
    <ApplicationVersion>1.8</ApplicationVersion>
    <ApplicationKey><!-- your key here --></ApplicationKey>
    <SchemaVersion>1.7</SchemaVersion>
    <Timestamp>2016-12-22T11:06:24Z</Timestamp>
    <HoldingNumber>12/1234/1234</HoldingNumber>
  </api:SMGetCattleOnHoldingRequest>
</soapenv:Body>
```

Example Response

```
<SOAP-ENV:Body>
  <ns1:SMGetCattleOnHoldingResponse>
    <Timestamp>2016-12-22T12:17:55Z</Timestamp>
    <Animals>
      <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
Sex="Male" DOB="2013-03-19" BirthCPH="08/161/0025" BreedCode="BSH"
GeneticDamID="UK121060400049" SireID="UK529999700001"/>
      <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
Sex="Female" DOB="2016-03-27" BirthCPH="73/281/0024" BreedCode="SMX"
GeneticDamID="UK500122400934" SireID="UK529999700001"/>
    </Animals>
  </ns1:SMGetCattleOnHoldingResponse>
</SOAP-ENV:Body>
```

Get cattle on all holdings

This request retrieves a list of cattle that are registered against the user's main CPH. The field **CurrentCPH** describes which registered holding the animal is currently recorded on.

Example Request

```
<soapenv:Header/>
<soapenv:Body>
  <api:SMGetCattleOnHoldingRequest>
    <ApplicationName>ScotEID Desktop</ApplicationName>
    <ApplicationVersion>1.0</ApplicationVersion>
    <ApplicationKey><!-- your key here --></ApplicationKey>
    <SchemaVersion>1.7</SchemaVersion>
    <Timestamp>2016-12-22T11:06:24Z</Timestamp>
    <HoldingNumber>12/1234/1234</HoldingNumber>
  </api:SMGetCattleOnHoldingRequest>
</soapenv:Body>
```

Example Response

```
<SOAP-ENV:Body>
  <ns1:SMGetCattleOnHoldingResponse>
    <Timestamp>2016-12-22T12:17:55Z</Timestamp>
    <Animals>
      <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
CurrentCPH="65/123/1234" />
      <Animal AnimalID="UK54321012345" InitialCPH="65/123/1234"
CurrentCPH="65/123/1234"/>
    </Animals>
  </ns1:SMGetCattleOnHoldingResponse>
</SOAP-ENV:Body>
```

ScotMoves+ web services

Registering births

Overview

The webservice `SMECreateCattleBirthRegistrationsRequest` registers a calf on ScotEID. Birth CPH will be assumed to be the holding on which the dam is currently recorded on ScotEID.

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="UK543210012346"
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>
  <ResultStatus="error"Row="1">
    <Errors>
      <Error Field="DamID" Severity="fatal" Message="Not found on main
holding" Code="CTWS199" />
    </Errors>
  </Result>
```



```
<ResultStatus="success"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

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 SMCreatCattleMovements 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 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.

Marts & Abattoirs should supply the lot/kill number in the UserReference field.

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" MoveDate="2020-03-01" UserReference="Lot
1">
        <!--Optional: EID data-->
        <AnimaleID ISO24631="1 0 04 00 0 826054321023457" Hex=""
TID="" Timestamp=""/>
      <CattleMovement/>
    </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

Recording Deaths

Overview

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

UserReference - 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 possible¹.

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 carcass.

```
<api:SMECreateCattleDeathsRequest>
  <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>
  <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"/>
```

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

```

    </CattleDeath>

    <!-- Example below applicable to abattoirs includes EID and
    carcass information -->
    <CattleDeath Row="2" AnimalID="UK543210123456"
MainCPH="79/435/0157" DateOfDeath="2020-03-01" UserReference="Abattoir
death">
        <!--Optional: carcass information from abattoir-->
        <CarcassClassification 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" Row="1"/>
      <Result Status="success" 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>
<DateOfDeath></DateOfDeath>
<Status></Status>
<StatusDesc></StatusDesc>
<PassportVersion>1</PassportVersion>
<FAStatus FAStatus="NA" QMSStatus="98"
QMSStatusDescription="Not Scotch Eligible (NSE)" QMSStatusReason="FAStatus"
QMSStatusShortDescription="Over 4 y/o or has calved"/>
  <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="UK50000054321"
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>UK50000000000001</TagID>
      </TagGrouping>
      <TagGrouping Status="NEW" TagType="B" TagCode="004_B"
EmanIdentifier="4" OrderDate="2013-03-15Z" MainCPH="794350157">
        <TagID>UK50000000000002</TagID>
        <TagID>UK50000000000003</TagID>
      </TagGrouping>
    </TagGroupings>
  </ns1:SMEGetIssuedCattleTagsResponse>
</SOAP-ENV:Body>
```


Testing

The WSDLURL for the test environment is:

<http://api.test.scoteid.com/api?wsdl>

The data on the test environment is only updated periodically and does not mirror production. In order to commence testing you will need an account on the test server. If you do not have an account please contact support@scoteid.com and provide some basic details of your intended use.

Data Dictionary

Descriptions of data fields

Name	Description	Type	Example(s)
Row	Enumeration of each item within the submission	Integer	1 2
AnimalID	Official ID used for the animal. Spaces are not considered part of the ID and will be removed.	CattleID_Type	UK543210123456
MainCPH	Main CPH for the keeper as registered on CTS and ScotMoves	CPH_Type	79/435/0157
Sex	Sex of the animal	Sex_Type	Male
DOB	Date of birth	Date (ISO 8601)	2019-11-01
AnimalAlias	Alias for the animal	String	"Brutus"
BirthCPH	The holding number the animal was physically born on. In most instance this will be the Main CPH	CPH_Type	79/435/0157
BreedCode			
DamID	Calf's birth dam. If a surrogate was used the surrogate's ID would be recorded here rather than the genetic dam	CattleID_Type	UK543210123456
GeneticDamID	ID of the genetic dam. In most instances this will be the same as the DamID	CattleID_Type	UK543210123456
SireID	Enter either the full ID, or the AI code, or "UNKNOWN"	SireID_Type	AA1234 UK543210123456 UNKNOWN
DepartureLocation	For within keepership movement this is the physical location the animal was moved from.	CPH_Type	79/435/0157

DestinationLocation	For within keepership movement this is the physical location the animal was moved to.	CPH_Type	79/435/0155
Movedate	The date on which the animal was moved	Date (ISO 8601)	2019-01-01
UserReference	Optional field for recording a user generated reference against a movement	CattleMovementUserReference_Type	Summer grazings
DepartureMainCPH	The main CPH registered with BCMS/ScotMoves for the keeper moving the animal	CPH_Type	79/435/0155
DestinationMainCPH	The main CPH registered with BCMS/ScotMoves for the keeper receiving the animal	CPH_Type	79/435/0155

Glossary of acronyms

EID	Electronic Identification. Unique RFID identification stored in the eartag
RFID	Radio Frequency Identification. A passive transponder which when activated transmits an identification.
CPH	County Parish Holding number. Used to uniquely identify a holding.
TID	Transponder Identifier. The TID is a read-only number written to the tag's microchip by the chip manufacturer to authenticate the tag.
CTS	Cattle Traceability System.
BCMS	British Cattle Movement Service

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	Ayrshire 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 & bradaldane 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)