an xml standard for farm assessment data exchange

version 2.4
august 2007

describes a standard for the exchange of farm assessment and assurance data over the Internet through the use of XML, SOAP and web services.

prepared by:

sasha rebmann
creative director

(0131) 652 3767
sasha@everysite.co.uk

everysite / tlr ltd.

everysite is a service offered by tlr ltd.
company registration: 2623795
vat no: 600411996

1. introduction

The availability of information on assessment and assurance processes is growing in importance throughout the food chain as consumers, retailers and food producers become more concerned with the manner in which food is produced. This requires gathering and combining data from many sources, including farmers, growers, producers and assessors and delivering this information to decision makers.

IT plays a critical role in many stages this process: farmers use automated record-keeping systems to store information on the production process; assurance schemes keep large databases recording their members' assurance status and history and various traceability schemes keep track of produce all the way from the farm to the consumer.

Despite the differing scopes for these systems there is a large overlap in the types of data recorded, and often this data needs to be shared between systems. This is usually complicated by systems having different internal formats for the same records and frequently holding differing subsets of the same data.

This document describes a common format for the exchange of data between systems. By using a single standard for data exchange, which all IT solution providers can easily implement, more opportunities for sharing common data open up between all players.

1.1 the CASI system

TLR has already developed the Certification, Assessment and Standards Interface (CASI) system, which combines assurance data from various schemes and certification bodies and delivers it to producers, buyers and customers through various mechanisms. As we already use the CASI system to exchange data with certification and standards bodies, we feel the data formats used by CASI are a good starting point for a common standard. Currently, CASI deals with the following data types:

Schemes	An overall standard that assessments are made against
Scopes	A specific version of a standard. Usually new scopes are added annually for each new version of an assessment standard
Sections	A group of related questions within a standard
Assessment Points	A single question to be assessed, can be answered yes/no, by multiple-choice from a set of answers or by a list of fielded data
Answers	One of a set of possible answers to a question

Enterprises	An area of business that a scope or certificate can cover
Certifiers	An independent organisation that can assess and issue certificates against scopes
Memberships	An organisation that is to be assessed against a particular scope
Sites	Address details for a single property associated with a particular membership
People	Contact details for an employee associated with a particular site
Assessments	An evaluation of the application of a standard to a particular member
Responses	The given answer to a question for a particular assessment
Certificates	Issued for a membership and a particular scope of a scheme after completion of an assessment

These thirteen data objects capture all of the data necessary to complete an assessment, record associated membership and certification details. The relationship between these objects is illustrated in the diagram below:

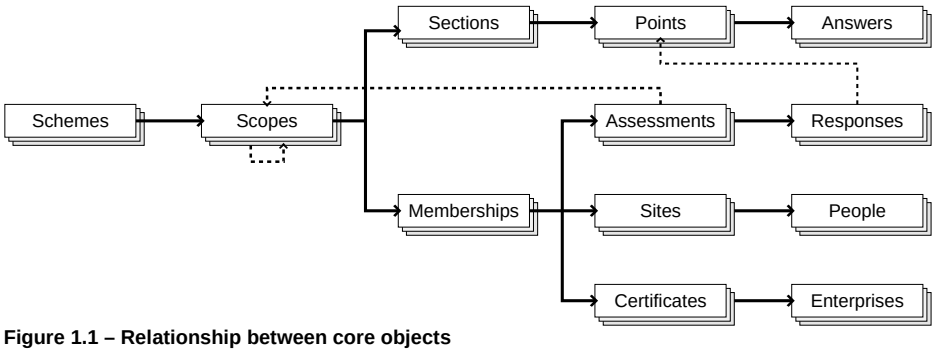


Figure 1.1 – Relationship between core objects

The Scheme object is the top level of the hierarchy. Below this are a set of scopes, which are particular versions of a scheme. All standards-related objects; Sections, Points, Answers and Enterprises belong to a Scope, which is a particular version of a scheme's standard. Sites, Contacts, Assessments, Responses and Certificates all belong to an instance of a Membership, which itself again belongs to a particular scope.

The dotted lines indicate important non-hierarchical relationships between objects. Additionally, response objects are linked to the answer chosen for the assessment point recorded and, like Assessments, Certificates are linked to a specific scope under which they are valid. Also, note that scopes can be linked to each other through the additional scopes collection.

1.2 using open standards

The data exchange specification is based around common open standards, using XML (extensible mark-up language) to describe data and SOAP (simple object access protocol), which itself is based on XML, and web services to provide the interface to any system requiring or providing data.

At the transmission level, data is sent between systems across the Internet using HTTP, the same protocol that is used by the World Wide Web.

Our implementation of the data exchange interface will be implemented using the Microsoft .NET framework, running on Windows 2000 server. .NET provides comprehensive class libraries for manipulating XML data, SOAP requests and creating web services, reducing the development work required to implement the final system.

1.3 the rest of this document

Section 2 outlines the XML schema definition for transmission of CASI data. Each of the core data types has an equivalent XML element defined by the schema, and the attributes, child elements and data types are described in detail with example XML fragments.

The section concludes by describing the format of the data transmission wrapper and data request objects.

Section 3 describes the SOAP protocol and methods for pushing data updates and requesting data from a system.

Section 4 details the method to be used for generating the globally unique identifiers required for all core data types.

Finally, appendix A contains the complete XML schema definition in XSD format and appendix B describes the changes made in recent versions of the specification.

2. xml schema definition

The common standard consists of three distinct parts. Firstly, an XML definition for each of the thirteen core CASI objects, allowing the complete assessment process to be represented through XML. On top of this is a common XML “wrapper” to facilitate the exchange of small chunks of data. Finally, a mechanism is needed to remotely request data updates from other systems.

This section describes the XML schema definition of the data exchange standard, covering the core data objects first, followed by the details of the transfer and request object formats.

2.1 core data object xml

Each of the thirteen core objects has an equivalent definition in XML, with three features:

Optimised for data transmission
As it is often necessary to break up assessment data into small chunks for transmission and exchange, XML objects will not be nested. For example, although there exists a set of individual responses to assessment points (questions) for each assessment, the assessment XML data only contains references to the responses, rather than the response objects themselves. This allows responses and assessments to be transmitted separately and incrementally, even though they are related. It is expected that systems will implement a separate database system for storing assessment data locally.

Unique References
Every XML object should have a reference number unique across all systems. This allows records to be matched up between systems despite differing internal referencing, and should help prevent unnecessary duplication of data.

Timestamps
Every data object has two timestamps that record when the object was first created and when it was last updated. These facilitate the incremental synchronisation data incrementally by only requesting objects that have been added or modified since the last update.

Common Attributes

Although the child elements vary between all core objects, every object has four common attributes containing metadata about the object. These are detailed in the table below:

Attribute	Type	Description
id	xs:string	ID string for this object, unique across all systems participating in data exchange
createdate	xs:dateTime	Timestamp when object created, given as GMT
lastmodified	xs:dateTime	Timestamp when last updated, given as GMT
action	casi:actionEnumeration	Update action, defined by the actionEnumeration type. This is described further in section 2.2

The *id* string for an object must be unique, and not shared by another object regardless of the object type or origin of the data. This ensures referential integrity is maintained across all systems, despite not necessarily having a complete set of CASI data.

The *createdate* and *lastupdated* timestamps are used when determining what objects to send to a client, and the *lastupdated* timestamp should be maintained internally by systems and updated whenever any data is changed.

Data Fields

Every core object has a set of child elements which contain the data for the object. Fields are expected to occur in the order specified, except optional fields which can be excluded if no appropriate data is available.

Data fields can be one of the following types, which define the attributes and type of data contained by the field:

Type	Description
xs:string	Element contains string data, as defined by the XSD specification
xs:short	Element contains a short integer, as defined by the XSD specification
xs:boolean	Element contains a boolean value, as defined by the XSD specification
xs:dateTime	Element contains a date and time, as defined by the XSD specification
casi:referenceType	Element is a reference to another core object, whose id is given in the *ref* attribute of the element

Reference Collections

Where an object owns a set of child objects, for example, a section that contains a number of assessment points, the child objects are referenced through a collection element. A collection element contains one or more references, which are elements of a *casi:referenceType* type, each having a *ref* attribute containing the unique ID string of the object being referenced.

The rest of this section describes the data fields and reference collections for each of the thirteen core data objects.

2.1.1 the scheme object

The *scheme* object is the highest-level object in the CASI hierarchy, with all other objects either directly or indirectly belonging to a particular scheme.

Scheme Fields

Element	Type	Description
name	xs:string	Name of the scheme
description *(optional)*	xs:string	Description of the scheme
email *(optional)*	xs:string	Contact email address for the scheme
url *(optional)*	xs:string	URL of scheme website
currentscope	casi:referenceType	Reference to the current active scope for this scheme. A scheme can have data for multiple scopes, but only scope can be in active use

A scheme has a name, an optional description, email address and url, and a reference to the current active scope.

Scheme Collections

Element	Type referenced	Description
scopes	scope	All scopes that belong to this scheme
certifiers	certifier	All certification bodies that can carry out assessments and issue certificates for this scheme

Every scheme has a set of scopes, which define particular instances of the scheme which members can be assessed under and a set of certifiers who are permitted to perform assessments and issue certificates.

Example

```
<scheme id="sch001" createdate="2003-09-01T12:00:00"
    lastmodified="2003-09-01T15:30:00" action="update">
  <name>Sample Scheme</name>
  <email>info@samplescheme.com</email>
  <currentscope ref="sco002"/>
  <scopes>
    <scope ref="sco001"/>
    <scope ref="sco002"/>
  </scopes>
  <certifiers>
    <certifier ref="cer001"/>
  </certifiers>
</scheme>
```

This defines a scheme object, *sch001* with the name “Sample Scheme”, two scopes *sco1* and *sco2*, with the current scope being *sco002,* and a certifier *cer001*. The optional description is not given.

2.1.2 the scope object

A *scope* is a particular instance of a scheme's standard. A new scope is created whenever a new version of the standard is issued, usually annually. Old scopes are kept to ensure the integrity of historical assessments and certificates.

Scope Fields

Element	Type	Description
scheme	casi:referenceType	Scheme this scope belongs to
name	xs:string	Name of the scope
description *(optional)*	xs:string	Description of the scope
type	xs:string	Type of the scope

A scope has a name and an optional description, and a scope type. The type field describes the type of object the scope defines the data requirements for, with the standard type being an assessment scope.

Scope Collections

Element	Type referenced	Description
sections	section	All assessment standard sections that belong to this scope
enterprises	enterprise	All enterprises that can be assessed under this scope
memberships	membership	All memberships that belong to this scope
additionalscopes *(optional)*	scope	Any additional scopes that are to be used in conjunction with this scope to define the scheme's data requirements

The contents of the standard to be assessed is split into a number of sections, with the sections that belong to a particular scope referenced through the sections collection. The enterprises collection lists the enterprises that this scope covers, chosen from a global set of enterprises maintained by the CASI system. The memberships collection contains the set of members which are assessed under this particular scope.

Finally, there can be a collection of additional scopes related to the scope. Additional scopes usually define the data capture requirements for data relating to specific sites, people or membership information required by the scheme in additional to the assessment standard.

Example

```
<scope id="sco002" createdate="2003-09-01T12:00:00"
    lastmodified="2003-09-01T12:00:00" action="update">
  <scheme ref=”sch001”/>
  <name>Sample Scope 2003-2004</name>
  <description>Sample scope, valid until August 2004</description>
  <type>nonconformances</type>
  <sections>
    <section ref="sec001"/>
    <section ref="sec002"/>
  </sections>
  <enterprises>
    <enterprise ref="ent001"/>
 </enterprises>
 <memberships>
    <membership ref="mem001"/>
  </memberships>
</scope>
```

This defines a scope for recording non-conformances, *sco002*, named “Sample Scope 2003-2004” which has two sections, *sec001* and *sec002*, covers a single enterprise: *ent001* and has one member *mem001*.

2.1.3 the section object

Each *section* object holds a set of assessment points, which are the individual requirements of the standard.

Section Fields

Element	Type	Description
scope	casi:referenceType	Scope this section belongs to
name	xs:string	Name of the section
description *(optional)*	xs:string	Description of the section
details *(optional)*	xs:string	Text giving full description of the contents of this section
reference	xs:string	Reference code for this section, as given in the standard
index	xs:short	A positive, non-zero integer giving the ordinal position of this section within the standard, used to order the sections for display

A section has a name, optional description and detail text, reference and index. The index field is used to order the sections when presenting the standard to the user.

Section Collections

Element	Type referenced	Description
points	point	All assessment points that belong to this section

A section has only a single collection, consisting of references to the individual assessment points within the section.

Example

```
<section id="sec001" createdate="2003-09-01T14:00:00"
    lastmodified="2003-09-01T14:00:00" action="update">
  <scope ref=”sco002”/>
  <name>Initial Section</name>
  <reference>1.1</reference>
  <index>1</index>
  <points>
    <point ref="poi001"/>
    <point ref="poi002"/>
    <point ref="poi003"/>
  </points>
</section>
```

This describes a section, *sec001*, named “Initial Section” with a standard reference “1.1” and an index of 1. The section contains three points: *poi001*, *poi002* and *poi003*.

2.1.4 the point object

The *point* object describes a single assessment point. An assessment point consists of a question with supporting information and guidelines, along with a set of references to possible answers.

Point Fields

Element	Type	Description
section	casi:referenceType	Section this assessment point belongs to
name	xs:string	Short name of the question – this is often a reference number
details	xs:string	The text of the actual question posed by the assessment point
evidence *(optional)*	xs:string	Summary of the evidence required to answer this question
guidance *(optional)*	xs:string	Explanatory text giving guidance on how to answer this question
criteria *(optional)*	xs:string	Explanatory text giving the criteria that should be used to ensure compliance of the chosen answer to this point with the standard
correctiveaction *(optional)*	xs:string	Explanatory text giving general corrective action information for this point
assessorguidance *(optional)*	xs:string	Explanatory text giving guidance for assessors
assessorinstructions *(optional)*	xs:string	Explanatory text giving instructions for assessors
resources *(optional)*	xs:string	Text giving information on further resources related to this assessment point
reference *(optional)*	xs:string	Reference code or number used to refer to this point in the published standard
score	xs:short	Integer weighting indicating the importance of this question
index	xs:short	A positive, non-zero integer giving the ordinal position of this assessment point within the section, used to order the points for display

An assessment point has a short name, further details giving the full question and a series of fields giving guidance and supporting information for answering the question.

Point Collections

Element	Type referenced	Description
links	link	Internet links applicable to this assessment point
answers	answer	All possible answer choices for this assessment point

A question has a collection of answer references, which list the possible answers to this question. All questions are multiple-choice, but each individual answer choice can require any amount of additional data to be captured.

Questions also can contain a collection of linked URLs, held in the links collection. Each link has a name, url and a description of the context in which the document relates to the assessment point.

Link Fields

Element	Type	Description
name	xs:string	Text name of the link
url	xs:string	Full URL of the link
context *(optional)*	xs:string	Description of the context in which the link relates to the associated assessment point

Example

```
<point id="poi001" createdate="2003-09-01T14:10:00"
    lastmodified="2003-09-01T14:10:00" action="overwrite">
  <section ref=”sec001”/>
  <name>W10</name>
  <details>On-farm transport of concentrated pesticides</details>
  <guidance>If possible take routes that avoid crossing or close
    proximity with watercourses even if these are
    longer.</guidance>
  <correctiveaction>A significant number of pollution incidents
    leading to death of aquatic life are caused each year by
    accidents during the transport of
    pesticides.</correctiveaction>
  <score>1</score>
  <index>10</index>
  <links>
    <link id=”lnk01”>
      <name>The Green Code: Part 1 Training &amp;
        Certification</name>
      <url>http://www.adlib.ac.uk/contentAD.aspx?doc=3684&
        page=3698</url>
      <context>The Green code provides general guidance.</context>
    </link>
  </links>
  <answers>
    <answer ref="ans001"/>
    <answer ref="ans002"/>
    <answer ref="ans003"/>
    <answer ref="ans004"/>
  </answers>
</point>
```

This defines a point, *p001*, for a question “W10” with guidance and corrective action notes, a score of 1 and an index position of 10. The point has four answer choices, *ans001* to *ans004*.

2.1.5 the answer object

The *answer* object defines a single choice of answer to an assessment point. Every answer can require additional data to be recorded for the assessment, and this is defined through the *additionaldata* element, which contains a set of field definitions.

Answer Fields

Element	Type	Description
point	casi:referenceType	Point this answer belongs to
name	xs:string	Short name for the answer – often a reference number
details	xs:string	Full text of the answer
evidence *(optional)*	xs:string	Text giving details of the evidence required in order to select this answer
guidance *(optional)*	xs:string	Explanatory text giving guidance on this answer
criteria *(optional)*	xs:string	Explanatory text giving the criteria that should be used to ensure compliance of an answer with the standard
correctiveaction *(optional)*	xs:string	Explanatory text giving the corrective action required if this answer is chosen
reference *(optional)*	xs:string	Reference code or number used to refer to this answer in the published standard
score	xs:short	Integer weighting giving the importance of this answer
nonconformance	xs:boolean	Boolean value indicating if this answer is a critical non-conformance to the standard
index	xs:short	A positive, non-zero integer giving the ordinal position of this assessment point within the section, used to order the answers for display

An answer has a short name, further details giving the full answer and a series of fields giving guidance and supporting information for the answer.

Answer Collections

Element	Type referenced	Description
additionaldata	fieldgroup fieldtable	Contains a set of fieldgroup and fieldtable elements

The *additionaldata* collection contains a series of elements that describe the schema for capturing any additional data associated with this answer.

The collection contains a series of *fieldgroup* or *fieldtable* elements. A *fieldgroup* element represents a group of fields that describe the additional data to be recorded, and contains a series of individual *fielddef* elements.

A *fieldtable* describes a set of groups of field elements that should be presented in tabulated form. It contains a collection of *fieldgroup* elements, one group for each row of the table. Each group within the table must contain fields in the same order and of the same type and name, although the field ids must be unique.

FieldGroup Attributes

Attribute	Type	Description
id	xs:string	Unique id used to refer to this field group

The *id* attribute should contain a unique id string to refer to this group.

FieldGroup Fields

Element	Type	Description
name	xs:string	Name for the group. Often used as a row header in a field table.

A *fieldgroup* element contains a single field – the name of the group.

FieldTable Collections

Element	Type referenced	Description
groups	casi:fieldgroup	A collection of field groups, one for each row of this table

FieldGroup Collections

Element	Type referenced	Description
fields	casi:fielddef	A collection of field definitions for this field group

Each *fielddef* element contains the field name, data type and a collection of options.

FieldDef Attributes

Attribute	Type	Description
id	xs:string	Unique id used to refer to this field
type	casi:fieldtypeEnumeration	Type of data to be captured

The *type* attribute can be one of the following values:

Attribute	Description
text	Field is a short text field
longtext	Field is a long, multiple-line text field
number	Field contains a number
quantity	Field contains a positive integer
currency	Field contains a monetary value
datetime	Field contains a date and time
yesno	Field is a yes/no answer
selectlist	Field is a drop-down select list with a set of options
multipleselect	Field is a multiple-select list with a set of options
radiolist	Field is a collection of radio buttons
checkboxlist	Field is a collection of checkboxes
label	Field is an non-editable text label

FieldDef Fields

Element	Type	Description
name	xs:string	Name of this field
details *(optional)*	xs:string	Description of this field, giving guidance on what information to provide
defaultvalue *(optional)*	xs:string	Default value for the field, or label text for label field type

FieldDef Collections

Attribute	Type referenced	Description
options	option	Collection of options to choose from for a select list, multiple select, checkbox list or radio list field.

The options collection contains a set of *option* elements, each having an *id* attribute containing a unique id for the option, a *value* attribute containing the value that should be recorded if the option is chosen and the option name.

Example

```
<answer id="ans001" createdate="2003-09-01T14:15:00"
    lastmodified="2003-09-01T14:15:00" action="update">
  <point ref=”poi001”/>
  <name>Yes</name>
  <details>Yes, I dispose sheep-dip to land</details>
  <score>3</score>
  <nonconformance>false</nonconformance>
  <index>1</index>
  <additionaldata>
    <fieldgroup id="licensegroup">
      <name>License Numbers</name>
      <fielddef id="licensenumber" type="text">
        <name>License Number</name>
        <description>Enter your sheep dip disposal
          license number</description>
      </fielddef>
      <fielddef id="licensetype" type="selectlist">
        <name>License Type</name>
        <options>
          <option id="licprovisional" value="provisional">
            Provisional License
          </option>
          <option id="licfull" value="full">
            Full License
          </option>
        </options>
      </fielddef>
    </fieldgroup>
    <fieldtable>
      <fieldgroup id="qtyfresh">
        <name>Fresh Dip</name>
        <fielddef id="qtyfreshstored" type="number">
          <name>Quantity stored (litres)</name>
        </fielddef>
        <fielddef id=”qtyfreshdisposed" type="number">
          <name>Quantity disposed (litres)</name>
```

```
        </fielddef>
      </fieldgroup>
      <fieldgroup id="qtyused">
        <name>Used Dip</name>
        <fielddef id="qtyfreshstored" type="number">
          <name>Quantity stored (litres)</name>
        </fielddef>
        <fielddef id="qtyfreshdisposed" type="number">
          <name>Quantity disposed (litres)</name>
        </fielddef>
      </fieldgroup>
    </fieldtable>
  </additionaldata>
</answer>
```

This describes an answer to a point, *ans001*, named “yes” with two sets of additional data requirements. The first group defines fields to record a license number and type along with the answer choice in the response. The second group is a table consisting of two rows, recording two quantities each.

2.1.6 the enterprise object

The *enterprise* object defines an area of business for which a certificate can be issued to cover. The CASI system defines a global list of enterprises, from which schemes can select a subset that their standard covers. Every certificate issued by a certifier also lists the enterprises for which it is valid.

Enterprise Fields

Element	Type	Description
name	xs:string	Name of the enterprise
description *(optional)*	xs:string	Further description of the enterprise

An enterprise has only a name and optional description.

An enterprise does not reference any other objects and therefore has no collections.

Example

```
<enterprise id="ent001" createdate="2003-09-01T14:20:00"
    lastmodified="2003-09-01T14:20:00" action="update">
  <name>Top fruit</name>
</enterprise>
```

2.1.7 the certifier object

The *certifier* object represents an independent organisation that can perform assessments and issue certificates for a scheme.

Certifier Fields

Element	Type	Description
name	xs:string	Name of the certification body
description *(optional)*	xs:string	Further description of the certifier
email *(optional)*	xs:string	Contact email address for the certifier
url *(optional)*	xs:string	URL of certifier website

A certifier has only a name and optional description.

The certifiers object does not reference any other objects and therefore has no collections.

Example

```
<certifier id="ent001" createdate="2003-09-01T14:20:00"
    lastmodified="2003-09-01T14:20:00" action="update">
  <name>TLR Ltd</name>
  <email>info@tlr.ltd.uk</email>
</certifier >
```

2.1.8 the membership object

The *membership* object contains information about an individual membership of a scheme. A membership can cover one or more physical *sites*, and have multiple *people* listed as contacts for each site. Every membership has a record of all *assessments* carried out by certifiers and any *certificates* that have been issued.

A membership contains the name under which the membership is held, the start date and optional end date of the membership. The object also contains references to the primary site, primary contact and current certifier for the membership.

A membership can also contain an optional value for the previous membership number and an optional reference to a previous certifier. These are usually provided if the member has changed certifiers at some point.

Finally, a status field contains a string giving the memberships status chosen from a set range of values.

Membership Fields

Element	Type	Description
scope	casi:referenceType	Scope this membership belongs to
number	xs:string	Membership number assigned by scheme
previousnumber *(optional)*	xs:string	Previous membership number for this membership
startdate *(optional)*	xs:dateTime	Date which this member joined the scheme
enddate *(optional)*	xs:dateTime	Date which this member left the scheme, not provided if a current membership
primarysite	casi:referenceType	Reference to a site object which is the main location for this membership
primarycontact	casi:referenceType	Reference to a person object who is the main contact for this membership
certifier	casi:referenceType	Reference to a certifier object who is the current certifier for this membership
previouscertifier *(optional)*	casi:referenceType	Reference to a certifier object who was the previous certifier for this membership
status	xs:string	Current status of this membership

The *status* field can take one of the following values:

applied	reopened	grace
registered	transferredin	closed
provisional	transferredout	
current	expired	

For further explanation of how to use and understand theses status codes, please see the document entitled “Data submission requirements for certification bodies”.

Membership Collections

Element	Type referenced	Description
sites	site	All sites that this membership covers
assessments	assessment	All assessments that have been carried out for this membership
certificates	certificate	All certificates that have been issued for this membership
history	event	All historical events recorded for this membership

A membership contains collections of references to the set of *sites* the membership covers, any *assessments* that have been carried out for the member, any *certificates* that have been issued for the member. and an optional set of *event* elements containing historical information for the membership.

Each *event* element has a unique *id* attribute, and records changes made to the membership along with the event date and any associated data.

Event Elements

Element	Type	Description
eventtype	xs:string	Type of change made to the membership
eventdate	xs:dateTime	Date these changes were made to the membership
description	xs:string	Description of the changes made to the membership
data *(optional)*	xs:string	Data associated with the event

The *eventtype* field can take one of the following values:

applied	reopened	madegrace
registered	transferredin	closed
madeprovisional	transferredout	renewed
madecurrent	expired	reversal

For further explanation of how to use and understand these event codes, please see the document entitled “Data submission requirements for certification bodies”.

Example

```
<membership id="mem001" createdate="2003-09-01T15:00:00"
    lastmodified="2003-09-01T15:00:00" action="update">
  <scope ref=”sco002”/>
  <number>1234</number>
  <startdate>2003-09-01T15:00:00</startdate>
  <primarysite ref="sit001"/>
  <primarycontact ref=”per001”/>
  <certifier ref="cer001"/>
  <status>current</status>
  <assessments>
    <assessment ref="ass001"/>
    <assessment ref="ass002"/>
  </assessments>
  <certificates>
    <certificate ref="ctf001"/>
  </certificates>
</membership>
```

This defines a membership, *mem001*, with a membership number of “1234” starting on September 1st 2003. The primary site for the membership is *sit001* and the current certifier for this member is *cer001*. The member has had two assessments, *ass001* and *ass002* and one certificate issued: *cer001*.

2.1.9 the site object

The *site* object holds address and contact information about a physical location associated with a membership. Each site can have one or more *people* listed as points of contact for that site.

Site Fields

Element	Type	Description
company	xs:string	Name of the company or site
description *(optional)*	xs:string	Further description of the site
address	xs:string	Full address of the site, excluding postcode
postcode	xs:string	Site postcode
country	xs:string	Country for the address of the site
telephone	xs:string	General telephone number for the site
fax *(optional)*	xs:string	Fax number for the site
email	xs:string	General contact e-mail address for the site
url *(optional)*	xs:string	URL for a website associated with the site
holdingnumber *(optional)*	xs:string	CPH number for the site
herdmark *(optional)*	xs:string	DEFRA herd mark for the site
primarycontact *(optional)*	casi:referenceType	Reference to a person object who is the primary contact for the site

The site object holds contact information for the site, along with a reference to the person object that is the primary contact for the site, if the site has any contacts defined.

Site Collections

Element	Type referenced	Description
contacts	person	All contacts for this site

The site object contains references to all people who are listed as contacts for the site.

Example

```
<site id="sit001" createdate="2003-09-01T15:00:00"
    lastmodified="2003-09-01T15:00:00" action="update">
  <company>everysite</company>
  <address>PO Box 23660, Edinburgh</address>
  <postcode>EH7 4XX</postcode>
  <country/>
  <telephone>0131 652 3767</telephone>
  <email>info@everysite.co.uk</email>
  <primarycontact ref="con001"/>
  <contacts>
    <person ref="con001"/>
    <person ref="con002"/>
```

```
  </contacts>
</site>
```

This describes a single site for a membership, *sit001*, with general contact details and two individual contacts, *con001* and *con002*, where *con1* is the primary contact for the site.

2.1.10 the person object

The *person* object holds contact details for an individual person associated with a site and membership.

Person Fields

Element	Type	Description
title	xs:string	Title that should be used to address the person
initials *(optional)*	xs:string	Full initials (excluding surname) of the person
firstname *(optional)*	xs:string	First name of the person
surname	xs:string	Surname of the person
address	xs:string	Contact address, excluding postcode
postcode	xs:string	Postcode for the contact address of the person
country	xs:string	Country for the contact address of the person
telephone	xs:string	Telephone contact number
fax *(optional)*	xs:string	Fax number
mobile *(optional)*	xs:string	Mobile phone number
email	xs:string	Personal e-mail address

A person object holds contact information for an individual person.

The person object does not reference any other objects and therefore has no collections.

```
<person id="per001" createdate="2003-09-01T15:00:00"
    lastmodified="2003-09-01T15:00:00" action="update">
  <title>Mr</title>
  <firstname>Sasha</firstname>
  <surname>Rebmann</surname>
  <address>PO Box 23660, Edinburgh</address>
  <postcode>EH7 4XX</postcode>
  <country/>
  <telephone>0131 652 3767</telephone>
  <email>sasha@everysite.co.uk</email>
</person>
```

2.1.11 the assessment object

The *assessment* object holds information about a single assessment visit by a certifier to a scheme member.

Assessment Fields

Element	Type	Description
membership	casi:referenceType	Membership this assessment belongs to
description *(optional)*	xs:string	Further description of this assessment
status	xs:string	Text giving current status of this assessment
assessmentdate	xs:dateTime	Date assessment was carried out
qcdate	xs:dateTime	Date assessment was added to the system
statusdate	xs:dateTime	Date assessment status was last updated
scope	casi:referenceType	Reference to the scope object under which this assessment was carried out
assessedobject *(optional)*	casi:typedReferenceType	Reference to the object this assessment relates to
free1 *(optional)*	xs:string	Optional scheme-specific data field
free2 *(optional)*	xs:string	Optional scheme-specific data field
free3 *(optional)*	xs:string	Optional scheme-specific data field

An assessment has a description, status and records of the assessment, quality control and status update dates. Each assessment also contains a reference to the scope object under which the assessment was performed, and an optional reference to the particular object the assessment relates to. This is used for assessments under a scope that records data about a specific membership, site or person rather than a standard assessment.

Finally, three optional data fields can contain scheme-specific information relating to the assessment. These are used, for example, to record the vet code for a QVVR assessment under the Assured British Pigs scheme.

For a self-audit assessment, the free1 field is used to store a reference to the question currently being asked. This allows audit interface implementers to return users to the point they left off in an audit.

Assessment Collections

Element	Type referenced	Description
responses	response	All responses recorded for this assessment

An assessment contains all the recorded responses to the assessment points contained in the standard.

Example

```
<assessment id="ass001" createdate="2003-09-01T15:30:00"
    lastmodified="2003-09-01T15:30:00" action="update">
  <membership ref=”mem001”/>
  <status>Initial Visit</status>
  <assessmentdate>2003-08-20T12:00:00</assessmentdate>
  <qcdate>2003-09-01T15:30:00</qcdate>
  <statusdate>2003-09-01T15:30:00</statusdate>
  <scope ref="sco001"/>
  <responses>
    <response ref="res001"/>
    <response ref="res002"/>
  </responses>
</assessment>
```

This assessment, *ass001* describes an initial visit to a member, with two responses recorded, *res001* and *res002* for assessment points under the scope *sco001*.

2.1.12 the response object

The *response* object holds information about an answer given to a single assessment point during an assessment.

Response Fields

Element	Type	Description
assessment	casi:referenceType	Assessment this response belongs to
state	xs:string	Text giving the state of this response
score	xs:short	Integer score of the chosen answer for this response
point	casi:referenceType	Reference to the point object this response is for
answer *(optional)*	casi:referenceType	Reference to the answer object chosen for this response

A response records the current state and score for the chosen answer and references to the assessment point the response is made to and the answer chosen, if any.

The state can be one of the following values:

State	Description
unanswered	No answer has been recorded for this response
answered	An answer chosen by the assessor has been recorded for this response
presupplied	An answer has been recorded for this response as the result of previously supplied information
notrequired	It has been indicated that no answer is required for the assessment point this response is for
disabled	The interface displaying the assessment point for this response should not allow the user to alter the answer recorded by the response
hidden	The interface displaying the assessment point for this response should not allow the user to view or alter the answer recorded by the response

Response Collections

Element	Type referenced	Description
fields	field	All additional data recorded for this response

The response object also records any additional data that is required to be collected by the chosen answer. The responses are held in the *fields* collection, which holds a series of *field* elements.

Each *field* element has a *ref* attribute containing the id of the *fielddef* element that this element contains data for, and the recorded data itself.

Example

```
<response id="res002" createdate="2003-09-01T15:30:00"
    lastmodified="2003-09-01T15:30:00" action="update">
  <assessment ref=″ass001″/>
  <state>answered</state>
  <score>1</score>
  <point ref="poi005"/>
  <answer ref="ans001"/>
  <fields>
    <field ref="licensenumber">0045784</field>
  </fields>
</response>
```

This response, *res001*, describes the answer to assessment point *poi005*. The answer selected was *ans001*, and the response contains a single item of additional data, referencing the *licensenumber* field in the original point.

2.1.13 the certificate object

A *certificate* is issued or updated by a certifier in response to an assessment. A member may have multiple certificates under different scopes, but only the certificate under the most current scope gives the live certification status of a member.

Certificate Fields

Element	Type	Description
membership	casi:referenceType	Membership this certificate belongs to
status	xs:string	Current status of the certificate
startdate *(optional)*	xs:dateTime	Date this certificate is valid from
enddate *(optional)*	xs:dateTime	Date this certificate is valid to
scope	casi:referenceType	Reference to the scope object that this certificate was issued under

A certificate contains the assurance status of the membership, along with the start and end dates of the validity of the certificate. A certificate object also contains a reference to the scope object under which the certificate was issued.

Certificate Collections

Element	Type referenced	Description
enterprises	enterprise	All enterprises this certificate is valid for
history	event	All historical events recorded for this certificate

A certificate object contains a collection of references to the enterprises the certificate is valid for, along with an optional set of *event* elements containing historical information for the certificate.

Each *event* element has a unique *id* attribute, and records the changes made to the certificate along with the event date and any associated data.

Event Elements

Element	Type	Description
eventtype	xs:string	Type of change made to the certificate
eventdate	xs:dateTime	Date these changes were made to the certificate
description	xs:string	Description of the changes made to the certificate
data *(optional)*	xs:string	Data associated with the event

The *eventtype* field can take one of the following values:

statuschange	enterprisechange	enddatechange
scopechange	startdatechange	schemeevent

For further explanation of how to use and understand these event codes, please see the document entitled “Data submission requirements for certification bodies”.

Example

```
<certificate id="ctf001" createdate="2003-09-01T15:40:00"
    lastmodified="2003-09-01T15:45:00" action="update">
  <membership ref=”mem001”/>
  <status>FULL</status>
  <startdate>2003-09-01T15:40:00</startdate>
  <enddate>2004-09-01T15:40:00</enddate>
  <scope ref="sco001"/>
  <enterprises>
    <enterprise ref="ent003"/>
    <enterprise ref="ent004"/>
  </enterprises>
  <history>
    <event id="evt001">
      <eventtype>statusupdated</eventtype>
      <description>Status updated to "FULL" from
        "SUSPENDED"</description>
      <eventdate>2003-09-01T15:40:00</eventdate>
    </event>
  </history>
</certificate>
```

This describes a certificate, *ctf001*, currently valid, issued 1st September 2003 under the scope *sco001*. The certificate covers two enterprises, *ent003* and *ent004*. The certificate has one recorded historical event, where the status was updated from SUSPENDED to FULL.

2.2 data exchange wrapper

Now the format of all the core data objects has been defined it is relatively simple to bundle a set of objects together for transmission.

The schema defines a root element, called *wrapper*, to contain all objects included in a single transmission. The wrapper element can contain any of the thirteen core objects in any order and in any quantity, and has no other child elements or attributes.

When preparing data for transmission, the *action* attribute for each object must be set to one of the following values, which specifies how the recipient should handle the data:

Value	Description
update	Recipient should update or add this object to the local database only if the timestamp given is more recent that the current version
overwrite	Recipient should update or add this object to the local database ignoring the timestamp
delete	Delete this object from the local database (in this case, the XML description of the object does not need to contain any data)

Example

```
<wrapper>
  <scope id="sco002" createdate="2003-09-01T12:00:00"
      lastmodified="2003-09-01T12:00:00" action="update">
    <scheme ref=”sch001”/>
    <name>Sample Scope 2003-2004</name>
    <description>Sample scope, valid until August
      2004</description>
    <sections>
      <section ref="sec001"/>
      <section ref="sec002"/>
    </sections>
    <enterprises>
      <enterprise ref="ent001"/>
      <enterprise ref="ent003"/>
      <enterprise ref="ent004"/>
    </enterprises>
  </scope>
  <certifier id="ent001" createdate="2003-09-01T14:20:00"
      lastmodified="2003-09-01T14:20:00" action="update">
    <name>TLR Ltd</name>
  </certifier>
  <enterprise id="ent001" createdate="2003-09-01T14:20:00"
```

```
      lastmodified="2003-09-01T14:20:00" action="update">
    <name>Top fruit</name>
  </enterprise>
  <enterprise id=”ent002” createdate=”2003-09-01T14:20:00”
      lastmodified=”2003-09-01T14:45:00” action=”delete” />
</wrapper>
```

This data transmission wrapper contains four objects: a Scope, a Certifier and two Enterprise objects. The second enterprise object has been deleted, and therefore has no further definition.

2.3 data requests

It is often necessary for a system to request a new set of data through a *pull subscription*, rather than updated data being *pushed* from the source.

It is envisaged that most systems will implement a SOAP interface, allowing data to be exchanged via HTTP requests over the Internet. The schema defines a set of elements and types for forming a request for data.

A request for data for a specific object is made through an *requestObject* type element. There are thirteen of these elements defined, one for each of the core object types. Each request element has three attributes which precisely specify the data to be returned.

requestObject Type Attributes

Attribute	Type	Description
ref	xs:string	ID string for object requested
return	casi:returnEnumeration	A value from the returnEnumeration enumeration. Set to “this” to return only the object given by the ref attribute, or “all” to recursively return all objects referenced by the requested object.
objects	casi:objectList	A value matching the casi:objectList type. Used in conjunction with return=”all”, specifies a list of object types to return.

Individual requests for data are grouped together in a wrapper element, named *request*. This element has a single attribute, *updatedsince*, which is used to request an incremental update of only data that has been modified since the date given.

Request Attributes

Attribute	Type	Description
updatedsince	xs:dateTime	Set to request only data objects that have been modified after this date

The flexibility of the request format is best illustrated through a series of examples:

To request data for a single site object only:

```
<request>
```

```
  <site ref="00120030526190502457" type="this"/>
</request>
```

To request data for multiple single objects:

```
<request>
  <membership ref="00120030418201045085" type="this"/>
  <site ref="00120030418201106031" type="this"/>
  <site ref="00120030526190502457" type="this"/>
</request>
```

To request all objects referenced by a membership that have been updated since midnight on 20/06/2003:

```
<request updatedsince="2003-06-20T00:00:00">
  <membership ref="00120030418201045085" type="all"/>
</request>
```

To request the set of sections, questions and answers for a particular scope of a scheme:

```
<request>
  <scope ref="00420030505112330731" type="all"
    objects="section point answer"/>
</request>
```

2.4 authentication

To request data or push data updates, a client must authenticate with the server. Each person or system that interacts with CASI has a user account that gives them permissions to view data for specific schemes and certifiers. To authenticate, the username and password for an account with the appropriate permissions to access the required data must be included in the wrapper element.

The *authentication* element defines the credentials to use when retrieving or updating assurance information in CASI. The packet contains two fields holding the username and password of the account to use:

This element must always be placed first inside the wrapper or request elements.

Authentication Fields

Element	Type	Description
username	xs:string	username issued for authentication purposes
password	xs:string	password issued for authentication purposes

Example

```
<authentication>
  <username>abc99</username>
  <password>password</password>
```

```
</authentication>
```

2.4 errors

If an invalid request is submitted, or an error occurs when generating a response, an error element is sent instead of the results. The element contains two string fields that describe the error:

Element	Type	Description
exception	xs:string	the type of error that occurred
message	xs:string	the error message

Example 1 – error packet

```
<error>
  <exception>AuthenticationException</exception>
  <message>Bad username or password.</message>
</error>
```

3. soap interface

The SOAP web service interface specification is still in development. However servers participating in data exchange will implement two web service methods:

3.1 the pushUpdate method

The *pushUpdate* method is used to push a collection of data objects to another server. The SOAP document object contains a single *Wrapper* element as defined in the data exchange schema.

3.2 the requestData method

The *requestData* method is used to request a set of data from a remote server. The SOAP document object contains a *Request* element detailing the data objects requested, with the server returning the collection of objects contained in a *Wrapper* element.

4. generating globally unique identifiers

All objects in the CASI system are tagged with a globally-unique identifier, or GUID. These are used to refer to a specific object, and are constant across all systems that use and process CASI data.

It is vital to ensure that all systems generate GUIDs in the same manner, and that there is no possibility of two separate systems generating the same identifier. The CASI system uses a 24-character string as a GUID. The string consists of two parts, known as the *base* and the *extension*, separated by a period (“.”).

4.1 the guid base

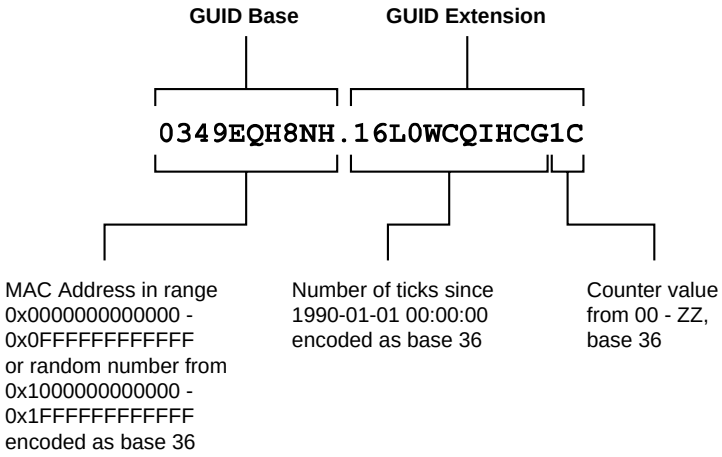
The GUID base is an unchanging value, unique to a particular system. The base consists of an encoded version of the systems MAC address. This is a unique hexadecimal number assigned to a computer’s network interface and encoded in the network hardware. A MAC address number can be anything in the range 0x0000000000000 – 0x0FFFFFFFFFFF and is usually quoted as six pairs of two-digit hexadecimal number, for example 00:45:BF:A8:44:C3. By converting this to base-36 representation (base 36 uses the numbers 0-9 and the letters A-Z to represent a single digit), this is reduced to a 10 character string.

If no MAC address is available for a system, a random number in the range 0x1000000000000 – 0x1FFFFFFFFFFF should be used instead, and again encoded as base 36.

4.2 the guid extension

The extension part of a GUID is unique to all identifiers generated by a single system, although it is possible for two separate systems to generate the same extension. Combined with the base, the resulting identifier will be unique across all systems.

The first eleven characters of the extension consist of a base-36 encoded count of the number of *ticks* since 1990-01-01 00:00:00. A tick is a system clock interval, equivalent to 100 nanoseconds. The last two characters contain an auto-incrementing value from 0 to 1295 (00 – ZZ as base 36). This counter should be maintained by the ID generator, and incremented after a new identity is created.



5. alternate implementations

TLR maintains a reference implementation of this data exchange specification, giving access to CASI data through SOAP web services, using the Microsoft .NET framework.

The SOAP specification allows for messages and data to be exchanged through methods other than HTTP. For example, it would be possible to use FTP for the transfer of less time-critical data, or to send updates via e-mail from client software where the computer may not be connected to the Internet regularly.

6. references

Extensible Markup Language (XML) 1.0 (Second Edition)
http://www.w3.org/TR/REC-xml-20001006/

XML Schema Part 0: Primer
http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

SOAP version 1.2 Part 0: Primer

http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

Hypertext Transfer Protocol – HTTP/1.1
ftp://ftp.isi.edu/in-notes/rfc2616.txt

Microsoft .NET Framework
http://www.microsoft.com/net/

appendix a. complete xml schema

The complete XML schema definition for farm assessment data exchange is available at http://www.tlr.ltd.uk/casi/xml/casi.2.4.xsd

appendix b. version history

Changes in version 2.4

- Country field added to Site object
- Country field added to Person object

Changes in version 2.3

- HerdMark field added to Site object

Changes in version 2.2

- Version numbering synchronised with “Data submission requirements for certification bodies”
- HoldingNumber field added to Site object
- Authentication element added to specification
- Error element added to specification

Changes in version 2.0

- Figure 1.1 updated to include relationship between scopes introduced by the additional scopes collection.
- Type field and AdditionalScopes collection added to scope object.
- Details field added to section object.
- Restructured additional data collection for an answer, adding field group and field table elements.
- Added label type to field type enumeration for a field definition
- Added value attribute to a field option element
- AssessedObject reference field added to Assessment object to record the object being assessed under a membership, site or person type scope
- Added mention of the use of the free1 field in an self-audit assessment
- State field added to response object.
- Answer reference in response object made optional to cope with unanswered, disabled and hidden response states.