

September 2005

## **EECS Registration Databases Specification of Standard for Data Exchange**

This document describes the set of documents that specify interfaces to be used for the transfer of information between EECS Registration Databases

<b>Document Reference</b>	AIB-PRO-FS03
<b>Version no.</b>	1f0
<b>Date of Issue</b>	22 September 2005
<b>Reason for Issue</b>	Release
<b>Author</b>	Systems Working Group

## I. DOCUMENT CONTROL

### A Authorities

Version	Date	Authors	Change Reference
0d1	29/4/2005	<b>M Sandford</b>	Initial version
0d2	14/6/2005	<b>M Sandford</b>	Prepared for General Meeting
1f0	22/9/2005	<b>M Sandford</b>	Release

### B Change History

Issue 0d1 is the first draft of a new Fact Sheet under EECS Principles and Rules of Operation. The Fact Sheet is based on documentation produced for the Renewable Energy Certification System that preceded EECS.

Issue 0d2 incorporates minor comments from the Systems Working Group in preparation for submission to the General Meeting.

Release 1.0 was approved by the Copenhagen General Meeting of the AIB on 22 September 2005.

### C Changes Forecast

None expected.

### D Related Documents

AIB Principles and Rules of Operation (PRO)

AIB-PRO-FS03-A: Rules for construction of Identifiers

AIB-PRO-FS03-B: Data Transfer File Formats

AIB-PRO-FS03-C: Data Transport Protocol

AIB-PRO-FS03-D: Interface Testing

### E Intellectual Property Rights and Copyright

This document is the copyright of, and all rights are reserved on behalf of, the Association of Issuing Bodies.

## II. CONTENTS TABLE

<b>I.</b>	<b>Document Control.....</b>	<b>2</b>
A	Authorities.....	2
B	Change History.....	2
C	Changes Forecast.....	2
D	Related Documents.....	2
E	Intellectual Property Rights and Copyright.....	2
<b>II.</b>	<b>Contents Table.....</b>	<b>3</b>
<b>1</b>	<b>Introduction.....</b>	<b>4</b>
1.1	Purpose and Scope.....	4
1.2	Approach.....	4
1.2.1	The Data Exchange Process and Requirements.....	4
1.2.2	Identifier Standards.....	5
1.2.3	Message Definition.....	5
1.2.4	Data Transfer Protocols.....	5
1.2.5	Interface Testing.....	5
<b>2</b>	<b>Data Exchange Processes.....</b>	<b>5</b>
2.1	Basic Data Transmission Protocol.....	7
2.1.1	Responsibilities of Sender.....	7
2.1.2	Responsibilities of Recipient.....	7
2.2	Export and Import of Certificates.....	8
2.2.1	Responsibilities of the Seller.....	8
2.2.2	Responsibilities of the Exporting Issuing Body.....	8
2.2.3	Responsibilities of the Importing Issuing Body.....	9
<b>3</b>	<b>Requirements.....</b>	<b>9</b>
3.1	Functional requirements.....	9
3.2	Process requirements.....	10
3.3	Implementation requirements.....	10
<b>4</b>	<b>Security Risk Assessment.....</b>	<b>10</b>
<b>5</b>	<b>Glossary.....</b>	<b>12</b>

# 1 INTRODUCTION

## 1.1 Purpose and Scope

This document is the head document of a set that addresses the technical requirements for an EECS Registration Database. Specifically, these are requirements relating to the transfer of information from or to an EECS Registration Database. They therefore relate primarily to file formats and transport protocols. The transfer of certificate information from an account in one EECS Registration Database to an account in another EECS Registration Database also imposes requirements on common identifiers. The document set also addresses testing of interfaces.

This set of Fact Sheets deal with electronic interfaces only.

## 1.2 Approach

The approach to the interface definition process adopted in these documents is to partition the specification into units that may, if required, be changed without affecting other units. This head document identifies the other documents and provides a description of the business process in order to provide a context for the other details.

### 1.2.1 The Data Exchange Process and Requirements

A Business Process can be represented by a 'transaction' - a message or sequence of messages that fulfil a business function, for example 'submit report request' leads to 'report sent' or 'error message - not available'. Each of these messages can be defined as a logical 'flow' to meet the requirement. The flow can be classified by its characteristics at the business level:

- Originating Party.
- Receiving Party.
- Initiating event (e.g. user request, another flow, timer expires).
- Frequency in unit time.
- Data content at the business level.
- Mechanism: Electronic Data File Transfer or Manual.
- Volume: frequency \* mean message size.
- Validation rules.

Flows are given unique identifiers. The same flow can be sent by more than one originator and to more than one party and as a result of different initiating events. These origin/destination/initiation cases are called here different 'instances' of the same flow.

Sections 2 and 3 cover these issues. Section 4 provides some background on security.

### 1.2.2 Identifier Standards

The data that is exchanged between EECS Registration Databases includes information relating to accounts and production devices. It is important that these entities are uniquely identified, and that the identification should remain unique even after a series of transactions. The scope of these identifiers must therefore include at least the whole EECS community, irrespective of the particular scheme or schemes that any particular Registration Database supports.

These matters are addressed in AIB-PRO-FS03-A: Rules for construction of Identifiers

### 1.2.3 Message Definition

This defines what the data flow contains in terms of fields, their attributes and how the fields are grouped within the flow. At the same time, the rules for which fields and groups are optional or mandatory and whether and how often groups can be repeated are specified.

This logical message definition encompasses all the data visible at the user level and is closely aligned to the database design, since the flows are used to populate the database and/or are derived from their contents. The physical file format defines the data representation and control information. Similarly to the logical definition, a naming convention and layout standards are set out so that the information can be exchanged and validated in a consistent and unambiguous form.

These matters are addressed in AIB-PRO-FS03-B: Data Transfer File Formats.

### 1.2.4 Data Transfer Protocols

The data transfer mechanism for electronic data interchange is considered to be separate from the format of the data file. The mechanism provides for secure and reliable exchange of data that is appropriate for the maintenance of a clear audit trail of certificate transfer and the avoidance of double counting.

These matters are addressed in AIB-PRO-FS03-C: Data Transport Protocol

### 1.2.5 Interface Testing

In order to ensure that each EECS Registration Database is able to transfer information in a form that complies with the requirements identified above it is necessary to test each instance of such a database. The specification addresses the basic test process and the tests that are to be performed and does not cover the reporting of tests for the purposes of assessment or the process for qualifying and EECS Registration Database for the purposes of certificate storage.

The tests are addressed in AIB-PRO-FS03-D: Interface Testing

## 2 DATA EXCHANGE PROCESSES

The following diagram represents the basic data exchange model. The model is based on data flows and authorisation routes.

The process described does not show all details of, for example, out of band error handling. It is intended to provide a basis for describing business requirements.

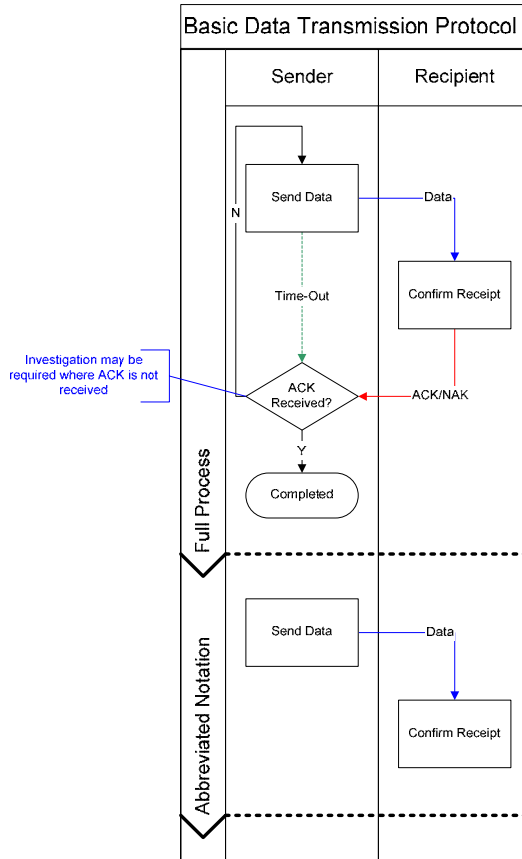


Figure 1 Basic Data Transmission Protocol

## 2.1 Basic Data Transmission Protocol

### 2.1.1 Responsibilities of Sender

The Sender is responsible for the accurate creation of messages, the application of appropriate security measures, the transmission of messages and the monitoring of responses from the Recipient(s). The Sender:

- Creates a transfer message.
- Checks the syntax of the message before sending.
- Applies appropriate message security measures.
- Sends the message to the Recipient.
- Checks if an acknowledgement is received from the Recipient within the appropriate timescale.
- In case the acknowledgement has not been received within the appropriate timescale:
  - resends the unacknowledged file;
  - checks if an acknowledgement is received from the recipient within the appropriate timescale; and
  - contacts the Recipient by email / fax / phone if again no response is received.
- In case an acknowledgement is received:
  - positive: logs the message as received and takes appropriate action;
  - negative on message: checks for software or message errors, repairs errors and resends data as a new file using a new Message ID; or
  - negative on content: checks content source and rectifies issue as appropriate. Resends data as a new file using a new Message ID if appropriate.

### 2.1.2 Responsibilities of Recipient

The Recipient is responsible for the monitoring of all configured data submission ports and the correct handling and processing of data received. The Recipient:

- Checks for incoming messages in a manner that ensures a timely response.
- Extracts and validates message source from message.
- Verifies the correctness of message contents within the appropriate timescale.
- In case the message is correct:
  - sends a positive acknowledgement to the Sender.
- In case the message is not correct:
  - sends a negative acknowledgement to the Sender which is either:
    - negative on the message (e.g. failure on check sum or format); or
    - negative on the content (e.g. buyer Trader Account ID does not exist).

## 2.2 Export and Import of Certificates

Data exchange between EECS Registration Databases takes place based on the following procedures. The interface is designed for operation within an automated environment but may be implemented manually should circumstances dictate. Compliance with this specification does not depend upon a fully automated solution being used.

Some actions are not relevant to the operation of this interface and are included only for completeness of process.

### 2.2.1 Responsibilities of the Seller

The Seller is responsible for the correct submission of a trade notification to the Issuing Body operating the Seller's account. The Seller:

- Specifies a transfer order which contains:
  - the number of certificates;
  - which certificates; and
  - the buyer of the certificates (by his Trader Account ID number).
- Is responsible for the correct content of the transfer order.

### 2.2.2 Responsibilities of the Exporting Issuing Body

The Exporting Issuing Body is responsible for the correct handling of the order and submission of these details to the Importing Issuing Body. The Exporting Issuing Body:

- Validates the details submitted by the Seller.
- Detects which IB will import the certificates.
- Creates a transfer message conforming to the specification in AIB-PRO-FS03-B: Data Transfer File Formats
- Sends the message to the Importing Issuing Body using the secure transport mechanism defined in AIB-PRO-FS03-C: Data Transport Protocol.
- Sets the status of the certificates to "exported".
- Waits for acknowledgement message
- If a positive acknowledgement is received, record that the export has been completed.
- If a negative acknowledgement is received:
  - negative on content: contacts Seller and rectifies issue as appropriate. Resends data as a new file using a new Message ID as appropriate.
  - negative on message: review message generation process and rectify issues as appropriate. Resends data as a new file using a new Message ID as appropriate.

### 2.2.3 Responsibilities of the Importing Issuing Body

The Importing Issuing Body is responsible for the correct handling of the order, processing of these details and Acknowledgement to the Exporting Issuing Body. The Importing Issuing Body:

- In case the message is correct:
  - sends a positive acknowledgement (using required security measures) to Exporting Issuing Body (see AIB-PRO-FS03-B: Data Transfer File Formats); and
  - stores certificates on the account of the buyer.
- In case the message is not correct:
  - sends a negative acknowledgement (using required security measures) to exporting Issuing Body (see AIB-PRO-FS03-B: Data Transfer File Formats); which is either:
    - negative on the message (e.g. failure on check sum or format); or
    - negative on the content (e.g. buyer ID does not exist).

## 3 REQUIREMENTS

### 3.1 Functional requirements

These requirements identify what the mechanism is to do.

**Transfer:** The message must be transported from sender to recipient.

**Transparency:** Any failure of delivery must be discovered.

**Attributable:** The message must be clearly identifiable as having come from the intended sender.

**Accurate:** The message must arrive with a high confidence that it has not been altered in transit.

**Private:** The message must arrive with a high confidence that it will not be understood by any reasonably equipped third party.

The business process shows an acknowledgement activity. This is part of the solution to the **transparency** requirement and should be considered a mandatory solution. The requirements **transport** and **attributable** apply to the acknowledgement.

The security related requirements (**accurate** and **private**) have been written as functional requirements even though they contain potentially quantitative concepts ('high confidence', and 'reasonably equipped'). This has been done because these concepts are expensive to measure and the available solutions do not support the kind of variation which would be needed to properly address a quantitative analysis. In effect, solutions will be chosen on the basis of an investigation of risk.

### 3.2 Process requirements

These are quantitative requirements.

**CMO-CMO-time:** transfer time between initiation of transfer by sending CMO to receipt of message by receiving CMO: expected 10 minutes: maximum defined as 'less than ACK-time'.

**ACK-time:** total time from receipt of message by recipient to receipt of acknowledgement by sender: minimum not applicable: expected 20 minutes: maximum 3 business days (24 business hours).

### 3.3 Implementation requirements

These requirements address issues concerning the processes of development of the transport mechanism.

**Immediate:** The transport mechanism was originally required to support trading activity and the full RECS trial. It was therefore required to be relatively straightforward to implement.

**Delivery:** the transport mechanism must be in use by all CMOs prior to live operation.

**Cost:** development and implementation costs are to be kept to a minimum.

## 4 SECURITY RISK ASSESSMENT

The security related requirement, **private**, states that the message must arrive with a high confidence that any reasonably equipped third party will not understand it. This statement is imprecise and must be interpreted by reference to the relevant risks.

It is understood that, whatever key length is chosen, it will be possible to read a message given enough time and resource. Reported breaks of messages encrypted with 1024 bit keys in times of the order of a few days appear to be based on large resources. In the context of CMO transfers we might assume that eavesdroppers will not have the incentive to apply large resources to reading this data. However, in the context of an active certificate market, many messages may use the same key, so it only takes one break to open up many messages. While this risk may be reduced by frequent changes of key it is worth looking at the problem in a wider context.

The following table identifies some of the risks in the process and gives a broad indication of how significant they might be.

Risk	Discussion	Significance
Private key exposed by the user	It is possible for a user process to accidentally expose a private key. Reasonable internal procedures and high security awareness must be in place to reduce this possibility.	Medium
Message intercepted in transit	Before an encrypted message can be broken, it must be intercepted. This can happen in a number of ways: by being intercepted in flight; by being found in the message store of an	Low

	intermediate mail agent; by being sent to the wrong recipient. Of these, the last is the most likely if automation is not used. Otherwise, it would require deliberate action by someone to find a message.	
Message decrypted to extract private key	Someone with 'reasonable equipment' is going to require perhaps weeks of dedicated computation time to decrypt a message with a good 1024 bit key. Unfortunately, it is possible to generate poor keys and the proposal to use a common CA helps to ensure that only good keys are generated. The data in the decrypted message will loose value over time. This effect reduces the incentive to break the message. By applying a suitably rapid key change cycle the incentive can be further reduced.	low

It is a truism that the most significant security issues are internal to the organisation. The analysis above suggests that the weakest aspect of the proposed protocol is the issuing of the keys. This is an acceptable position for the purposes of the opening stages of the market.

## 5 GLOSSARY

Public Key Cryptography	<p>A set of encryption algorithms which support the use of two keys, one public and one private. One key, the public key, can be issued widely and used by any sender to encrypt a message. The message can only be decrypted by the matching private key.</p> <p>The process is symmetrical in that the private key can be used to encrypt messages which only the public key can decrypt. This form is used to support a Digital Signature.</p>
Digital Signature	<p>A process of encrypting a message or document with a private key so that the recipient can verify that the message was sent by the owner of the private key and that the message has not been changed. The message may be sent in plain text with the digital signature attached as an extra data block.</p>
Digital Certificate	<p>A signed copy of someone's public key and associated identification information. The key is normally signed by a Certificate Authority who warrants that the public key does indeed belong to the person identified in the certificate.</p> <p>A self-signed Digital Certificate is one which has been digitally signed by the person identified in the certificate.</p>
Certificate Authority (CA)	<p>An organisation who produces Digital Certificates. The CA must be trusted by all parties involved.</p> <p>The CA will sign Digital Certificates using its own private key, and those who use the certificates must have a trusted copy of the CA's own public key. In some cases this trusted copy will be provided by another CA, creating a hierarchy of trust. This hierarchy normally terminates with a self-signed Digital Certificate.</p>
PKI (Public Key Infrastructure)	<p>The set of processes and systems used to manage a set of public keys. The term may apply to a single organisation, to a closed group of organisations, or to an open market. The term does not cover any specific technology or process.</p>

September 2005

**EECS Identifiers  
Specification of Standard for Data Exchange**

This document describes the interfaces to be used for the transfer of information between Central Monitoring Offices participating in RECS

<b>Document Reference</b>	AIB-PRO-FS03-A
<b>Version no.</b>	1f0
<b>Date of Issue</b>	22 September 2005
<b>Reason for Issue</b>	Release
<b>Authors</b>	Systems Working Group

## I. DOCUMENT CONTROL

### A Authorities

Version	Date	Authors	Change Reference
0d1	29/4/2005	M Sandford	Initial version
0d2	14/6/2005	M Sandford	Prepared for General Meeting
1f0	22/9/2005	M Sandford	Release

### B Change History

Issue 0d1 is the first draft of a new Fact Sheet under EECS Principles and Rules of Operation. The Fact Sheet is based on documentation produced for the Renewable Energy Certification System that preceded EECS.

Issue 0d2 incorporates minor comments from the Systems Working Group in preparation for submission to the General Meeting.

Release 1f0 was approved by the Copenhagen General Meeting of the AIB on 22 September 2005.

### C Changes Forecast

None expected.

### D Related Documents

AIB Principles and Rules of Operation (PRO)

AIB-PRO-FS03: EECS Registration Databases

International Standard ISO/IEC 646.

General EAN·UCC Specifications.

### E Intellectual Property Rights and Copyright

This document is the copyright of, and all rights are reserved on behalf of, the Association of Issuing Bodies.

## F Contents Table

<b>I.</b>	<b>Document Control</b> .....	<b>2</b>
A	Authorities.....	2
B	Change History .....	2
C	Changes Forecast .....	2
D	Related Documents.....	2
E	Intellectual Property Rights and Copyright.....	2
F	Contents Table .....	3
<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	Purpose .....	4
<b>2</b>	<b>Coding Structures</b> .....	<b>4</b>
2.1	Coding of CMOs.....	4
2.2	Coding of Certificates .....	4
2.3	Coding of Production Devices.....	5
2.4	Coding of Trader Account IDs .....	6
2.5	Coding of Issuing Bodies, Technologies and Earmarks.....	7
<b>3</b>	<b>Abbreviations</b> .....	<b>7</b>

## 1 INTRODUCTION

### 1.1 Purpose

The scope of this Interface Specification document is the definition of allowed identifiers for key entities used in the transfer of data between EECS Registration Databases.

## 2 CODING STRUCTURES

In order to ensure uniqueness of all data identifiers a methodology of coding has been implemented. The coding structure is based on the EAN.UCC numbering structure.

Alternative codes are supported by the data file structure so that, in principle, a trading account could be represented by some other suitably unique code. However, the use of alternative codes is not necessarily supported by all registries. Accordingly, all EECS Registration Databases must support at least the set of codes specified here.

### 2.1 Coding of CMOs

Each CMO must maintain at least one EAN prefix to be used in accordance with the EAN.UCC numbering structure. The CMO Prefix forms an essential part of the coding for Production Devices and Certificates. A Company Prefix is a numeric identifier of between 6 and 10 digits in length.

The CMO Company Prefix is used as the CMO ID. Where a CMO maintains more than one prefix, one prefix may be chosen as the CMO ID.

*Example CMO Company Prefixes are:*

*51234567 (8 digit Company Prefix)*

*598765432 (9 digit company prefix)*

### 2.2 Coding of Certificates

Certificates will be coded in accordance with Global Individual Asset Identifier (GIAI) (AI 8004), an element of the EAN.UCC numbering structure. The certificate number is always exactly 30 digits long.

Format of the Element String				
	Global Individual Asset Identifier			
	EAN.UCC Company Prefix for the CMO		Individual Asset Reference assigned by the CMO	
	$N_1 \dots$	$N_i$	$N_{i+1} \dots$	variable length

$i$  represents the length of the Company Prefix for the CMO.

The GIAI uses the EAN.UCC Company Prefix of the CMO assigning the Asset Reference. The structure and numbering of the Individual Asset Reference is determined by the

relevant CMO. CMOs may adopt any numbering methodology appropriate to the coding structure, although it is recommended that sequential Individual Asset Reference numbers be assigned.

Although the EAN.UCC specification for GIAI allows the Individual Asset Reference to contain all characters contained in Table 1 of the International Standard ISO/IEC 646, for the purposes of Certificate coding only numeric characters are permitted.

*Example GIAI-based Certificate Number:*

**51234567000000000000000000001234** (8 digit Company Prefix with 22 digit Individual Asset Reference)

### 2.3 Coding of Production Devices

Production Devices will be coded in accordance with Global Service Relation Number (GSRN) (AI 8018), an element of the EAN.UCC numbering structure.

Format of the Element String																	
	Global Service Relation Number																
	EAN.UCC Company Prefix For the CMO										Service Reference						Check Digit
	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>	N <sub>11</sub>	N <sub>12</sub>	N <sub>13</sub>	N <sub>14</sub>	N <sub>15</sub>	N <sub>16</sub>	N <sub>17</sub>

The GSRN uses the EAN.UCC Company Prefix of the CMO assigning the Service Reference. The Service Reference is assigned by the CMO and relates to an individual Production Device. The structure and content of the Service Reference number is at the discretion of the CMO.

The Check Digit is calculated as shown below. Its verification, which must be carried out in the application software, ensures that the number is correctly composed.

Check Digit Calculation																	
	Global Service Relation Number																
	EAN.UCC Company Prefix For the CMO										Service Reference						Check Digit
	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>	N <sub>11</sub>	N <sub>12</sub>	N <sub>13</sub>	N <sub>14</sub>	N <sub>15</sub>	N <sub>16</sub>	N <sub>17</sub>
	Multiply value of each position by																
	x3	x1	x3	x1	X3	x1	x3	x1	x3	x1	x3	x1	x3	x1	x3	x1	x3
	Accumulated results = 'sum'																
	Check digit = (nearest multiple of 10 ≥ 'sum') – 'sum'																

Example Check Digit Calculation																	
Start number	Global Service Relation Number																
	EAN.UCC Company Prefix For the CMO										Service Reference						Check Digit
	3	7	6	1	0	4	2	5	0	0	2	1	2	3	4	5	6
	Multiply value of each position by																
	x3	x1	x3	x1	X3	x1	x3	x1	x3	x1	x3	x1	x3	x1	x3	x1	x3
Interim	9	7	18	1	0	4	6	5	0	0	6	1	6	3	12	5	18
	Accumulated results = 'sum'																
	101																

Final number	Check digit = (nearest multiple of 10 ≥ 'sum') – 'sum'																110	
	3	7	6	1	0	4	2	5	0	0	2	1	2	3	4	5	6	-101
																		=9
																		9

Example GSRN-based Production Device Numbers are:

**512345670000012347** (8 digit Company Prefix with 9 digit Service Reference and single Check Digit)

**598765432000001235** (9 digit Company Prefix with 8 digit Service Reference and single Check Digit)

## 2.4 Coding of Trader Account IDs

Each trader shall be assigned a unique account reference by their host IB. The account reference shall be composed as follows:

- IB\_ID (2 numeric digits)
- X (single 'X' character)
- 6 character alphanumeric ID (0-9 and A-Z only)
- check character (see below)

An example Trader Account ID is 10XRWENETJ.

A check character is a character added to the end of the Trader Account ID that validates the authenticity of the code. A simple algorithm is applied to the other digits or letters of the code which yields the check character.

The last character of each of the Trader Account ID represents the check character that is calculated from the other characters using the following algorithm. An example of a Trader Account ID is 10XRWENETJ.

Calculation of the check character:

The first 9 characters of the code are individualised as follows

1	0	X	R	W	E	N	E	T
---	---	---	---	---	---	---	---	---

Where alphabetic characters are present, they are replaced by a numeric value with the value 10 for the letter « A » ; 11 for the letter « B » ; 12 for the letter « C », etc. and 35 for the letter « Z », as follows :

1	0	33	27	32	14	23	14	29
---	---	----	----	----	----	----	----	----

Then, the positions are again weighted, beginning with the greatest value to the left and ending with a one at the far right.

1	0	33	27	32	14	23	14	29
10	9	8	7	6	5	4	3	2

Each digit is multiplied by its position weight

10	0	264	189	192	70	92	42	58
----	---	-----	-----	-----	----	----	----	----

The products are then summed to give a total value: 917

A modulo 36 (which corresponds to the total number of characters available) is applied to the value 917 with the formula  $(36 - \text{MOD}([\text{value}],36))$ . This produces a numeric value in the range 1 to 36.

In the above example, the result is 19 which, since it is superior to 9 has to be converted to a letter using a similar mechanism as in Step 2. Number 0 is not an allowed output. Where the check character code is 36 this is represented as the character [.

Thus the code for the above example is: 10XRWENETJ. With an account base of 11XYWZNET the check character would be [ and the full account code would be 11XYWZNET[.

## 2.5 Coding of Issuing Bodies, Technologies and Earmarks

Permissible codes for Issuing Bodies, Technologies and Earmarks are published in AIB-PRO-FS06. Please consult the latest version of this list for details.

## 3 ABBREVIATIONS

AIB	Association of Issuing Bodies
CMO	Central Monitoring Office
EAN	International Article Numbering Association
GIAI	Global Individual Asset Identifier
GSRN	Global Service Relation Number
IB	Issuing Body
RECS	Renewable Energy Certificate System
UCC	Uniform Code Council
UTC	Universal Coordinated Time

September 2005

**EECS Transfer Interface File  
Specification**  
**Specification of Standard for Exchange of  
Certificate Data**

This document describes the interfaces to be used for the transfer of certificate information between registries.

<b>Document Reference</b>	AIB-PRO-FS03-B
<b>Version no.</b>	1f0
<b>Date of Issue</b>	22 September 2005
<b>Reason for Issue</b>	Release
<b>Authors</b>	Systems Working Group

## I. DOCUMENT CONTROL

### A Authorities

Version	Date	Authors	Change Reference
0d1	29/4/2005	<b>M Sandford</b>	Initial version
0d2	14/6/2005	<b>M Sandford</b>	Prepared for General Meeting
1f0	22/9/2005	<b>M Sandford</b>	Release

### B Change History

Issue 0d1 is the first draft of a new Fact Sheet under EECS Principles and Rules of Operation. The Fact Sheet is based on documentation produced for the Renewable Energy Certification System that preceded EECS.

Issue 0d2 incorporates minor comments from the Systems Working Group in preparation for submission of the General Meeting.

Release 1.0 was approved by the Copenhagen General Meeting of the AIB on 22 September 2005.

### C Changes Forecast

None expected.

### D Related Documents

AIB Principles and Rules of Operation (PRO)

CMO Interface Transport Mechanism 0d2, October 2002

CMO Interface Specification 1d1, October 2002

### E Intellectual Property Rights and Copyright

This document is the copyright of, and all rights are reserved on behalf of, the Association of Issuing Bodies.

## II. CONTENTS TABLE

<b>I.</b>	<b>Document Control.....</b>	<b>2</b>
A	Authorities.....	2
B	Change History.....	2
C	Changes Forecast.....	2
D	Related Documents.....	2
E	Intellectual Property Rights and Copyright.....	2
<b>II.</b>	<b>Contents Table.....</b>	<b>3</b>
<b>1</b>	<b>Introduction.....</b>	<b>4</b>
1.1	Purpose and Scope.....	4
<b>2</b>	<b>Overview of File Structure.....</b>	<b>4</b>
2.1	Introduction.....	4
2.2	Preamble.....	4
2.3	Header.....	5
2.4	Certificate Transfer File: Body.....	5
2.5	Acknowledgement File: Body.....	6
<b>3</b>	<b>Logical Message Definition.....</b>	<b>7</b>
3.1	Message Schema – Original RECS.....	7
3.2	Message Schema - EECS.....	11
<b>4</b>	<b>Physical Message Definition.....</b>	<b>15</b>
4.1	Optional and Mandatory Elements.....	15
4.2	Data Field Definitions – Header.....	16
4.2.1	Message Transmission Time.....	16
4.2.2	Transfer Message ID.....	16
4.2.3	Acknowledgement Message ID.....	16
4.2.4	From.....	16
4.2.5	To.....	17
4.2.6	Context.....	17
4.3	Data Field Definitions – Certificate Transfer File Body.....	17
4.3.1	Association.....	17
4.3.2	Original Holder.....	17
4.3.3	New Holder.....	18
4.3.4	Number of Coupons.....	18
4.3.5	Start Coupon.....	18
4.3.6	End Coupon.....	18
4.3.7	Certificate Coupon size and units.....	19
4.3.8	Issuing Body ID.....	19
4.3.9	Production Device ID.....	19
4.3.10	Certificate Issue Time.....	19
4.3.11	Technology Code.....	20
4.3.12	Earmark Flag.....	20
4.3.13	Capacity.....	20
4.3.14	Generation period.....	20
4.4	Data Field Definitions – Acknowledgement File Body.....	21
4.4.1	Transfer Message ID.....	21
4.4.2	Content Validation Flag.....	21
4.4.3	Message Validation Flag.....	21
<b>5</b>	<b>Abbreviations.....</b>	<b>21</b>

## 1 INTRODUCTION

### 1.1 Purpose and Scope

This document describes the file structures for:

- transferring certificates between registries
- acknowledging the receipt of such transfers.

This interface specification document addresses data transfer, including acknowledgement of transfer, between registries, specifically relating to certificates for EECS and RECS. The definitions in this paper include certificates designed to support Guarantees of Origin (GO), as defined under European Commission Directive 2001/77/EC.

The acceptability of certificates in certain markets is not a matter for this document. The data file is designed to allow certificates issued and processed under different rules to be distinguished from each other, even though the underlying data elements may be the same across a number of systems.

File record specifications are defined for each data record relating to transfers of certificates between registries. Message content and management process are defined.

## 2 OVERVIEW OF FILE STRUCTURE

### 2.1 Introduction

The transfer data file is designed to handle certificates from a number of different regimes. It is also intended to be backwards compatible with the original RECS data file.

Backwards compatibility is achieved by making all new elements optional in the XML schema. Thus, a file conforming to the original RECS schema is still acceptable. However, registries designed to process files defined in this document must take specific action to generate default values for elements that do not appear in the original schema. These elements, and their suggested defaults, are identified in section 4.

### 2.2 Preamble

The XML preamble describes the encoding and data schema that apply to the file. It takes the form:

```
<?xml version="1.0" encoding="UTF-8"?>
<r:Env xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://system.fingrid.fi/xml/cmo
http://system.fingrid.fi/xml/schemas/exportenv65.xsd"
xmlns:r="http://system.fingrid.fi/xml/cmo">
```

The hosting of the schema does not form part of this specification.

Two schemas are supported:

'exportenv65.xsd' is described in this standard and addresses the requirements for EECS.

'exportenv34.xsd' is described in the previous version of this standard (CMO Interface Specification 1d1), and addresses the original RECS requirements.

The references to fingrid in the namespace definition and schema location are historical. They are to be replaced at some time when an appropriate location has been identified. Registries are expected to validate messages against a local copy of the schema.

## 2.3 Header

Both the transfer and the acknowledgement files have a header section. This is designed to identify the file and to assist registries in routing the file to the appropriate process in their system. The header structures are similar for both files.

As an example, this file fragment shows a transfer between the registry using the GLN code number 6420616413223 to the registry using the SMTP address "recs@cmouk.co.uk". The context is "transfer", indicating a certificate transfer.

```
<r:header mst="2001-07-30T15:07:00+03:00">  
  <r:ID>042001073000001</r:ID>  
  <r:from cS="GLN">6420616413223</r:from>  
  <r:to cS="SMTP">recs@cmouk.co.uk</r:to>  
  <r:context>transfer</r:context>  
</r:header>
```

The example shows the use of 'cS' attributes on the r:from and r:to fields to identify different interpretations of the fields. The default is to have no such attribute, in which case the registry EAN number should be used.

## 2.4 Certificate Transfer File: Body

The body section of the certificate transfer file contains the data on the certificates to be transferred, and the identities of the old and new holders of the certificates.

The file structure is designed to handle certificates associated with any of the certificate systems within the scope of this document. The 'r:association' element defines which system the certificates in this file are associated with. This element may be repeated, allowing a set of certificates to be associated with more than one certificate system.

It is the responsibility of the sending registry to ensure that the data contained in the file is consistent with all the certificate schemes that it is associated with.

It is the responsibility of the receiving registry to ensure that the data contained in the file is used in accordance with the rules appropriate to the particular certificate system.

The element descriptions in section 4 contain details of which systems each element relates to.

This fragment shows a transfer of EECS certificates.

```
<r:Body>  
  <r:s cS="recs">04X00VAPOU</r:s>
```

```
<r:b cS="recs">02XSHELL0V</r:b>
<r:nroc>10</r:nroc>
<r:c>
  <r:association>eecs</r:association>
  <r:sc cS="recs">642061641313321101</r:sc>
  <r:ec cS="recs">642061641313321110</r:ec>
  <r:Csize unit="MWh">1</r:Csize>
  <r:IB>04</r:IB>
  <r:PD cS="recs">6420616413130000012</r:PD>
  <r:genperiod startd="2001-05-01" endd="2001-05-31"/>
  <r:ist>2001-06-15T11:23:00+03:00</r:ist>
  <r:tec cS="recs">01</r:tec>
  <r:earm cS="recs">1</r:earm>
</r:c>
</r:Body>
```

The original holder and the new holder are identified in elements 'r:s' and 'r:b' respectively. The 'cS' attribute shows that the identifiers are both RECS identifiers.

The actual certificates are described in the 'r:c' block. This refers to a contiguous set of certificates with serial numbers between 642061641313321101 and 642061641313321110 inclusive. If the transfer involves non-contiguous sets of certificates then further 'r:c' blocks can be included as required.

A single transfer file can only have one body element. This implies that:

All the certificates are to be transferred from the same original holder.

All the certificates are to be transferred to the same new holder.

The association element is placed inside the c element. This allows certificates of different associations or combinations of associations to be accommodated in a single data file.

## 2.5 Acknowledgement File: Body

The body of the acknowledgement file contains an indication of the acceptability of the related transfer file.

As an example, this fragment shows an ACK acknowledgement file where both the message and content aspects of the transfer file identified as "102002101800001" have been determined to be acceptable.

```
<r:Body>
  <r:ID>102002101800001</r:ID>
  <r:message>true</r:message>
  <r:content>true</r:content>
</r:Body>
```

### 3 LOGICAL MESSAGE DEFINITION

The following schema definitions are available on request.

#### 3.1 Message Schema – Original RECS

Interface Files for certificate transfer follow the schema described below.

```
<xs:element name="Body">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:s"/>
      <xs:element ref="r:b"/>
      <xs:element ref="r:nroc"/>
      <xs:element ref="r:c" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Csize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="unit" type="r:unittype" use="optional"
          default="MWh"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Env">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:header"/>
      <xs:element ref="r:Body"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="IB" type="xs:token"/>
<xs:element name="ID" type="xs:token"/>
<xs:element name="PD">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:long">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="b">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="c">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:association" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="r:sc" />
      <xs:element ref="r:ec" />
      <xs:element ref="r:Csize" minOccurs="0" />
      <xs:element ref="r:IB" />
      <xs:element ref="r:PD" />
      <xs:element ref="r:genperiod" minOccurs="0" />
      <xs:element ref="r:ist" />
      <xs:element ref="r:tec" />
      <xs:element ref="r:earm" />
      <xs:element ref="r:icap" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="association">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="recs" />
      <xs:enumeration value="go" />
      <xs:enumeration value="eecs" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="context">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="transfer" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="earm">
  <xs:complexType>
```

```
<xs:simpleContent>
  <xs:extension base="xs:byte">
    <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
      default="recs"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="ec">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:nonNegativeInteger">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="genperiod">
  <xs:complexType>
    <xs:attribute name="startd" type="xs:date" use="required"/>
    <xs:attribute name="endd" type="xs:date" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="header">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:ID"/>
      <xs:element ref="r:from"/>
      <xs:element ref="r:to"/>
      <xs:element ref="r:context" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="mst" type="xs:dateTime" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="icap" type="xs:positiveInteger"/>
<xs:element name="ist" type="xs:dateTime"/>
<xs:element name="nroc" type="xs:short"/>
<xs:element name="s">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

```
</xs:element>
<xs:element name="sc">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:nonNegativeInteger">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="tec">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="from">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="to">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="unittype">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="Wh"/>
    <xs:enumeration value="kWh"/>
    <xs:enumeration value="MWh"/>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:restriction>  
</xs:simpleType>
```

### 3.2 Message Schema - EECS

Interface Files for certificate transfer follow the schema described below.

```
<xs:element name="Body">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="r:s"/>  
      <xs:element ref="r:b"/>  
      <xs:element ref="r:nroc"/>  
      <xs:element ref="r:c" maxOccurs="unbounded"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
<xs:element name="Csize">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:integer">  
        <xs:attribute name="unit" type="r:unittype" use="optional"  
          default="MWh"/>  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>  
<xs:element name="Env">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element ref="r:header"/>  
      <xs:element ref="r:Body"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
<xs:element name="IB" type="xs:token"/>  
<xs:element name="ID" type="xs:token"/>  
<xs:element name="PD">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:long">  
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"  
          default="recs"/>  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

```
<xs:element name="b">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="c">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:association" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:element ref="r:sc"/>
      <xs:element ref="r:ec"/>
      <xs:element ref="r:Csize" minOccurs="0"/>
      <xs:element ref="r:IB"/>
      <xs:element ref="r:PD"/>
      <xs:element ref="r:genperiod" minOccurs="0"/>
      <xs:element ref="r:ist"/>
      <xs:element ref="r:tec"/>
      <xs:element ref="r:earm"/>
      <xs:element ref="r:icap" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="association">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="recs"/>
      <xs:enumeration value="go"/>
      <xs:enumeration value="eecs"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="context">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="transfer"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="earm">
  <xs:complexType>
```

```
<xs:simpleContent>
  <xs:extension base="xs:byte">
    <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
      default="recs"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="ec">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:nonNegativeInteger">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="genperiod">
  <xs:complexType>
    <xs:attribute name="startd" type="xs:date" use="required"/>
    <xs:attribute name="endd" type="xs:date" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="header">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="r:ID"/>
      <xs:element ref="r:from"/>
      <xs:element ref="r:to"/>
      <xs:element ref="r:context" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="mst" type="xs:dateTime" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="icap" type="xs:positiveInteger"/>
<xs:element name="ist" type="xs:dateTime"/>
<xs:element name="nroc" type="xs:short"/>
<xs:element name="s">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

```
</xs:element>
<xs:element name="sc">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:nonNegativeInteger">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="tec">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="from">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="to">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute name="cS" type="xs:NMTOKEN" use="optional"
          default="recs"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="unittype">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="Wh"/>
    <xs:enumeration value="kWh"/>
    <xs:enumeration value="MWh"/>
  </xs:restriction>
</xs:simpleType>

```

```
</xs:restriction>
</xs:simpleType>
```

## 4 PHYSICAL MESSAGE DEFINITION

Data fields defined in the message schema are described in further detail in this section. Examples of application are presented in Section 2.

Where appropriate, details of field structure have been included.

### 4.1 Optional and Mandatory Elements

Certificate data may be associated with one or more than one certificate system. This table indicates which elements are required for each certificate system. The numbers refer to notes below the table.

Note that the RECS data elements have been expanded from the original definition to include the EECS data requirements. The use of the association element allows for appropriate business rules to be applied within each registry to support the relevant markets.

The set of certificate types supported may be extended without reference to this specification so long as the data element set remains as described here.

Element	Original RECS	RECS	GOO
context	1	6	6
association	2	6	6
s	3	6	6
b	3	6	6
nroc	3,4	6	6
sc	3	6	6
ec	3	6	6
Csize	2	8	7,8
IB	3	6	6
PD	3	6	6
ist	3	6	7
tec	3	6	6
earm	3	6	7
icap	3	6	7
genperiod	2, 5	6,9	6

1. The context element must be omitted to allow the file to be interpreted as an original RECS file.
2. The association, Csize and genperiod elements must be omitted if the context element is also omitted.
3. These elements all are part of the original RECS file. This format does not support attributes and therefore does not support any identifier encodings other than those originally defined for recs.
4. The original RECS file is only capable of transferring certificates, rather than coupons. All references to coupons in the following descriptions should be taken to mean coupons of size 1 in the context of the original RECS file.

5. If the receiving registry is capable of storing start and end dates the registry should store a null value for the start date and make the end date equal to the issue date.
6. These elements are mandatory.
7. These elements are treated as optional when receiving data files for non-EECS certificates.
8. The Csize element is optional. The default size is 1 MWh.
9. If the sending registry is transferring a certificate created under the original RECS standard, then the data file must contain a genperiod element where the start date is empty and the end date is set equal to the issuing date.

## 4.2 Data Field Definitions – Header

### 4.2.1 Message Transmission Time

Timestamp for message file.

The recipient may validate the format of this field but may not reject the message if the date is beyond some arbitrary limit in the past. It is the responsibility of the sender to monitor the total turn round time of the transaction to ensure that an ACK message is received within the required time.

Element Name	Mst
Type	DateTime
Format	UTC (Z). Use of referential time zones (e.g. +1:00) is not permitted.
Structure	YYYY-MM-DDTHH:MM:SSZ
Example	2002-10-15T12:24:00Z

### 4.2.2 Transfer Message ID

Message ID for transfer message.

Element Name	ID
Type	Long
Format	15 digit fixed length number
Structure	YYYYMMDD & sequential number (5 digits)
Example	2002101800001

### 4.2.3 Acknowledgement Message ID

Message ID for acknowledgement message.

Element Name	AID
Type	Long
Format	15 digit fixed length number
Structure	YYYYMMDD & sequential number (5 digits)
Example	2002101800001

### 4.2.4 From

Identifier for sending registry.

This field may be validated for agreement with the XML specification. The recipient may not reject the message based on the content.

Element Name	From
Type	token
Format	Depends on setting of cS attribute
Structure	

Example	urn:ean:642061641
Attribute	Cs
Type	String
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.2.5 To**

Identifier for receiving registry.

This field may be validated for agreement with the XML specification. The recipient may not reject the message based on the content.

Element Name	To
Type	token
Format	Depends on setting of cS attribute
Structure	
Example	urn:ean:642061641
Attribute	Cs
Type	String
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.2.6 Context**

Processing context to assist file routing.

Element Name	Context
Type	String
Format	One of: 'transfer', 'acknowledge'
Structure	
Example	

**4.3 Data Field Definitions – Certificate Transfer File Body**

**4.3.1 Association**

Certificate system the certificates are associated with.

Element Name	Association
Type	String
Format	One of: 'recs', 'goo', 'eecs'. This list may be extended from time to time.
Structure	
Example	

**4.3.2 Original Holder**

Account ID for party transferring certificates.

Element Name	s
Type	token
Format	Depends on setting of cS attribute

Structure	
Example	10XRWENETJ
Attribute	cS
Type	String
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.3 New Holder**

Account ID for party receiving certificates.

<b>Element Name</b>	<b>b</b>
Type	token
Format	Depends on setting of cS attribute
Structure	
Example	10XRWENETJ
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.4 Number of Coupons**

Number of coupons transferred in the message.

<b>Element Name</b>	<b>nroc</b>
Type	Short
Format	Number
Structure	N...[N]
Example	682

**4.3.5 Start Coupon**

The number of the first coupon in the block of coupons to be transferred.

<b>Element Name</b>	<b>sc</b>
Type	NonNegativeInteger
Format	Depends on setting of cS attribute
Structure	
Example	
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.6 End Coupon**

The number of the last coupon in the block of coupons to be transferred.

<b>Element Name</b>	<b>ec</b>
Type	nonNegativeInteger

Format	Depends on setting of cS attribute
Structure	
Example	
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.7 Certificate Coupon size and units**

The number of certificates represented by each coupon, and the unit of measure for each certificate.

Element Name	Csize
Type	nonNegativeInteger
Format	
Structure	
Example	
Attribute	unit
Type	string
Format	One of: 'Wh', 'kWh', 'MWh'
Default	'MWh'
Structure	
Example	

**4.3.8 Issuing Body ID**

The ID of the Issuing Body responsible for the issue of the certificates being transferred.

Element Name	IB
Type	Token
Format	NN
Structure	2 character numeric, leading zero if required
Example	07

**4.3.9 Production Device ID**

The ID of the Production Device for which the certificates being transferred were issued.

Element Name	PD
Type	Long
Format	Depends on setting of cS attribute
Structure	
Example	506003453000000275
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.10 Certificate Issue Time**

Timestamp for original certificate issuing.

<b>Element Name</b>	<b>ist</b>
Type	dateTime
Format	UTC (Z). Use of referential time zones (e.g. +1:00) is not permitted.
Structure	YYYY-MM-DDTHH:MM:SSZ
Example	2002-10-15T12:24:00Z

**4.3.11 Technology Code**

Technology code for production device.

<b>Element Name</b>	<b>tec</b>
Type	token
Format	Depends on setting of cS attribute
Structure	
Example	07
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.12 Earmark Flag**

Earmark flag applied to production device at time of certificate issue.

<b>Element Name</b>	<b>earm</b>
Type	Byte
Format	Depends on setting of cS attribute
Structure	
Example	1
Attribute	cS
Type	string
Format	One of: 'recs' or other encodings to be agreed from time to time.
Default	'recs'
Structure	
Example	

**4.3.13 Capacity**

Production device capacity in kW.

<b>Element Name</b>	<b>icap</b>
Type	positiveInteger
Format	N...[N]
Structure	Up to 7 character numeric
Example	785

**4.3.14 Generation period**

Period of actual generation. This element has no data associated with it. The period is defined by two mandatory attributes.

<b>Element Name</b>	<b>genperiod</b>
Type	Empty
Format	
Structure	

Example	
Attribute	startd
Type	Date
Format	UTC (Z). Use of referential time zones (e.g. +1:00) is not permitted.
Structure	YYYY-MM-DD
Example	2002-09-15
Attribute	endd
Type	Date
Format	UTC (Z). Use of referential time zones (e.g. +1:00) is not permitted.
Structure	YYYY-MM-DD
Example	2002-10-15

## 4.4 Data Field Definitions – Acknowledgement File Body

### 4.4.1 Transfer Message ID

Message ID for transfer notification message.

Element Name	ID
Type	Long
Format	15 digit fixed length number
Structure	YYYYMMDD & sequential number (5 digits)
Example	2002101800001

### 4.4.2 Content Validation Flag

Indicator flag identifying validation status of transfer message content.

Element Name	Content
Type	Boolean
Format	Text
Structure	true/false single word
Example	True

### 4.4.3 Message Validation Flag

Indicator flag identifying validation status of transfer message format.

Element Name	Message
Type	Boolean
Format	Text
Structure	true/false single word
Example	True

## 5 ABBREVIATIONS

AIB	Association of Issuing Bodies
CMO	Central Monitoring Office
EECS	European Energy Certificate Scheme
EAN	International Article Numbering Association

GIAI	Global Individual Asset Identifier
GO	Guarantee of Origin
GSRN	Global Service Relation Number
IB	Issuing Body
RECS	Renewable Energy Certificate System
UCC	Uniform Code Council
UTC	Universal Coordinated Time

September 2005

## **EECS Transfer Interface Transport Specification**

### **Specification of Standard for Transport of Data**

This document describes the transport of information  
between EECS Registration Databases

<b>Document Reference</b>	AIB-PRO-FS03-C
<b>Version no.</b>	1f0
<b>Date of Issue</b>	22 September 2005
<b>Reason for Issue</b>	Release
<b>Authors</b>	Systems Working Group

## I. DOCUMENT CONTROL

### A Authorities

Version	Date	Authors	Change Reference
0d1	29/4/2005	<b>M Sandford</b>	Initial version
0d2	14/6/2005	<b>M Sandford</b>	Prepared for General Meeting
1f0	22/9/2005	<b>M Sandford</b>	Release

### B Change History

Issue 0d1 is the first draft of a new Fact Sheet under EECS Principles and Rules of Operation. The Fact Sheet is based on documentation produced for the Renewable Energy Certification System that preceded EECS.

Issue 0d2 incorporates minor comments from the Systems Working Group in preparation for submission to the General Meeting.

Release 1.0 was approved by the Copenhagen General Meeting of the AIB on 22 September 2005.

### C Changes Forecast

None expected.

### D Related Documents

AIB Principles and Rules of Operation (PRO)

AIB-PRO-FS03: EECS Registration Databases

### E Intellectual Property Rights and Copyright

This document is the copyright of, and all rights are reserved on behalf of, the Association of Issuing Bodies.

## **II. CONTENTS TABLE**

<b>I.</b>	<b>Document Control.....</b>	<b>2</b>
<b>II.</b>	<b>Contents Table.....</b>	<b>3</b>
<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Purpose .....	4
1.2	Approach .....	4
<b>2</b>	<b>Protocol Specification.....</b>	<b>4</b>
<b>3</b>	<b>Management of Public/Private Keys .....</b>	<b>5</b>

## 1 INTRODUCTION

### 1.1 Purpose

This document details the message transport aspects of the Interface Specification for communication between EECS Registration Databases.

The scope of this Interface Specification (IS) document is the definition of all interfaces between EECS Registration Databases. Interfaces between individual database operators and other national or participant systems unique to a specific domain are not part of the IS and are therefore not included.

### 1.2 Approach

The document takes the requirements outlined in the head document, AIB-PRO-FS03, and specifies a protocol for transferring files and a method of providing public/private security keys.

## 2 PROTOCOL SPECIFICATION

**Signed:** all messages are digitally signed by the sender. The recipient validates the digital signature on receipt. The signature format is X509 as incorporated into the SSL (Secure Sockets Layer) protocol.

Use of a signature addresses the **attributable** and **accurate** requirements. X509 is supported by Microsoft and the openssl project and is available on suitably up to date versions of Windows and Unix. This form of the signature therefore supports the **immediate, delivery, and cost** requirements.

**Encrypted:** all messages are encrypted after signature. The encrypted message conforms to the S/MIME message structure.

Encryption addresses the **private** requirement. S/MIME is supported by Microsoft and the openssl project and is available on suitably up to date versions of Windows and Unix. This form of encryption therefore supports the **immediate, delivery, and cost** requirements.

**SMTP:** signed and encrypted messages are transported from sender to recipient using SMTP (Simple Mail Transfer Protocol) using the public Internet.

Use of SMTP addresses the **transfer** requirement. SMTP is a 'push' protocol which forms the basis of (almost) all external email systems and is therefore easily available to everyone. Using SMTP allows the sender or the receiver to process the message using common email clients. Recent versions of Microsoft Outlook also handle signature and encryption, allowing the whole CMO-CMO transport to be handled with package software. The automation of SMTP mail creation and receipt is supported by a number of programming environments and this protocol therefore addresses the **immediate, delivery, and cost** requirements.

SMTP is only provides a direct transfer between pairs of mail agents. Many organisations structure their internal mail system using one or more mail agents, each one of which might store incoming mail before forwarding it. The cycle times of these intermediate agents introduce delays. These delays can be reduced by appropriate configuration within each organisation if necessary and SMTP appears to address the **CMO-CMO-time** requirement.

**Acknowledgement:** on receipt of a message the recipient is required to validate the signature and confirm that the message conforms to the expected structure and that the data content is of the correct type and falls within expected ranges. The recipient must generate an ACK message if the message is acceptable and a NAK message otherwise. This acknowledgement is then signed and returned to the return address given on the incoming message. The original sender should look for a valid acknowledgement. If there is no response within the defined **ACK-time**, or if a NAK is received, the original sender should attempt to resolve the problem by direct contact with the original recipient.

The use of an acknowledgement response in this way addresses the **transparency** requirement. The need to validate the original message before transmitting the acknowledgement introduces a process within the receiving CMO which is outside the scope of this document. It is therefore not possible to say what impact the acknowledgement activity will have on the **ACK-time** requirement. It is left to each individual CMO to devise appropriate procedures to address this.

### 3 MANAGEMENT OF PUBLIC/PRIVATE KEYS

**CA:** A Certificate Authority is required to issue Digital Certificates on behalf of all CMOs. This body also generates the public/private key pair for each CMO. The AIB is responsible for providing this service.

The use of a CA simplifies the issue and management of keys. It therefore addresses the **attributable, private, immediate** and **delivery** requirements. The **cost** implications will vary from CMO to CMO depending on their level of expertise in handling Digital Certificates.

**Key protocol:** The CA generates a public/private key pair for each CMO. The public keys are signed and issued to all CMOs. The private key, signed by the CA, is sent to the relevant CMO.

Allowing a CMO to generate its own keys would add complexity to the key exchange protocols.

The CA can be required to ensure that a sufficiently random input is used to generate key pairs.

CMOs are responsible for the configuration and maintenance of firewalls to manage the allowed connections from all other CMOs.

Each CMO is responsible for monitoring the contents of its specified receipt container on its own server location, and initiating the processing operation once a file is detected.

It is the responsibility of individual CMOs to manage the files and security on their own server locations.

September 2005

EECS Transfer Interface Test  
Specification  
**Test Requirements for Data Exchange**

This document describes the test specification for the standard interface architecture to be used for the transfer of information between Central Monitoring Offices participating in EECS

<b>Document Reference</b>	AIB-PRO-FS03-D
<b>Version no.</b>	1f0
<b>Date of Issue</b>	22 September 2005
<b>Reason for Issue</b>	Release
<b>Authors</b>	Systems Working Group

## I. DOCUMENT CONTROL

### A Authorities

Version	Date	Authors	Change Reference
0d1	29/4/2005	M Sandford	Initial version
0d2	14/6/2005	M Sandford	Prepared for General Meeting
1f0	22/9/2005	M Sandford	Release

### B Change History

Issue 0d1 is the first draft of a new Fact Sheet under EECS Principles and Rules of Operation. The Fact Sheet is based on documentation produced for the Renewable Energy Certification System that preceded EECS.

Issue 0d2 incorporates minor comments from the Systems Working Group in preparation for submission to the General Meeting.

Release 1.0 was approved by the Copenhagen General Meeting of the AIB on 22 September 2005.

### C Changes Forecast

None expected.

### D Related Documents

AIB Principles and Rules of Operation (PRO)

AIB-PRO-FS03: EECS Registration Databases

### E Intellectual Property Rights and Copyright

This document is the copyright of, and all rights are reserved on behalf of, the Association of Issuing Bodies.

## II. CONTENTS TABLE

<b>I.</b>	<b>Document Control</b> .....	<b>2</b>
A	Authorities.....	2
B	Change History.....	2
C	Changes Forecast.....	2
D	Related Documents.....	2
E	Intellectual Property Rights and Copyright.....	2
<b>II.</b>	<b>Contents Table</b> .....	<b>3</b>
<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	Purpose.....	4
1.2	Approach.....	4
1.2.1	Scope of testing.....	4
<b>2</b>	<b>Party Roles</b> .....	<b>4</b>
2.1	Responsibilities of Exporting CMOs.....	5
2.1.1	Data Submission.....	5
2.2	Responsibilities of Importing CMOs.....	7
2.2.1	Data Receipt.....	7
2.3	Responsibilities of the AIB.....	7
<b>3</b>	<b>Interface Categories</b> .....	<b>8</b>
3.1	Electronic File Transfer.....	8
3.2	Manual Data Transfer.....	8
<b>4</b>	<b>Test System Configuration</b> .....	<b>8</b>
4.1	Test Scripts and Data.....	8
4.2	Test Environment.....	9
4.3	Systems and Procedures.....	9
<b>5</b>	<b>Test Conduct and Processes</b> .....	<b>9</b>
5.1	Conduct of Tests.....	9
5.1.1	Electronic Data Interchanges.....	10
5.2	Test Processes.....	10
5.2.1	Problem Reporting and Problem Management.....	10
5.2.2	Problem Escalation.....	10
5.2.3	Test Result Reporting.....	10
5.2.4	Witnessing and Evidence.....	11
<b>6</b>	<b>Test Details</b> .....	<b>11</b>
6.1	Interface Configuration.....	11
6.1.1	Test 1. CMO-CMO Interface Setup.....	11
6.1.2	Test 2. CMO-CMO Interface Validation.....	12
6.2	Interface Operation.....	12
6.2.1	Test 3. CMO-CMO Valid Data Volumes.....	12
6.2.2	Test 4. CMO-CMO Invalid Data Volumes.....	13

# 1 INTRODUCTION

## 1.1 Purpose

This document details the Interface Test Specification (ITS) for the common communication interfaces operated by Central Monitoring Offices (CMOs).

The scope of this ITS document is the definition of all tests to be completed when commissioning interfaces between two CMOs and also between CMOs and the AIB. Interfaces between individual CMOs and other national or participant systems unique to CMOs are not part of this ITS and are therefore not included.

## 1.2 Approach

The approach to interface testing is defined, along with requirements for test data and test environments. Individual tests are identified. Specific test scripts, detailed file specifications and CMO-specific procedures are not detailed in this document. Each CMO will be responsible for the production of these items and the submission of these documents to the AIB. Following completion of any tests each CMO shall provide a test report to the AIB.

The concept of Exporting and Importing CMOs facilitates testing. For an interface between two CMOs to be deemed to have passed testing the relevant CMOs must have successfully completed testing in both Exporting and Importing roles. The Exporting CMO is the CMO that initiates a file transfer. The Importing CMO is the CMO that receives and validates a file and creates the required acknowledgement.

Reference is made to EECS Transfer Interface File specification (AIB-PRO-FS03-B) and EECS Transfer Interface Transport Specification (AIB-PRO-FS03-C).

### 1.2.1 Scope of testing

A CMO must undertake tests if:

- The CMO is newly constituted and has never previously undertaken tests;
- The CMO has introduced a new, or significantly modified registry; or
- The CMO is intending to support a new certificate type or new file format.

A CMO must undertake a full set of the tests described in this document with every other CMO operating any certificate scheme supported by that CMO.

The following are excluded from CMO interface testing:

- tests of interfaces between national or participant systems unique to CMOs; and
- testing of the business functionality of the CMO provided services.

# 2 PARTY ROLES

Three parties are involved in interface testing, their responsibilities are summarised below.

Product / Activity	Exporting CMO	Importing CMO	AIB
Test Plan			✓
Test Definitions			✓
Test Schedule	✓	✓	
Test Procedures	✓	✓	
Test Scripts	✓	✓	
Test Data	✓	✓	
Test Execution	✓	✓	
Problem Reporting	✓	✓	✓
Problem Management	✓	✓	✓
Issue Resolution			✓
Prepare Test Reports	✓	✓	
Review Test Reports			✓
Authorise Interfaces			✓

## 2.1 Responsibilities of Exporting CMOs

Exporting CMOs will be responsible for:

- contacting the AIB and Importing CMO pre-testing to ensure all parties are aware of the tests to be performed;
- liaison with the appropriate parties in respect of any security requirements as defined by the ITM;
- specifying the tests between CMOs and the data to be provided to the AIB;
- planning and specifying the tests and expected results within their own systems;
- setting up test data for flows to the Importing CMO;
- executing the tests at the Exporting CMO end (sending flows output to and receiving acknowledgement flows from Importing CMOs);
- supporting queries from the Importing CMO and resolving issues on the content of the data exchanged and the progress of the test; and
- providing to the AIB an audit trail of each test and overall test results.

### 2.1.1 Data Submission

The Exporting CMO is responsible for the correct handling of the order and submission of these details to the Importing CMO. The Exporting CMO:

- creates a transfer message according to the IS.
- checks the syntax of the message before sending.
- sends the message according to the ITM to the Importing CMO.
- set the status of the certificates to "exported".
- checks if an acknowledgement is received from the Importing CMO within the permitted timescale specified in the ITM.
- in case the acknowledgement has not been received within permitted timescale:

- o resends the unacknowledged file;
  - o checks if an acknowledgement is received from the Importing CMO within permitted timescale; and
  - o contacts the importing CMO by email / fax / phone if again no response is received.
- in case an acknowledgement is received:
  - o positive: logs the message as received and takes no further action.
  - o negative on message: checks for software or message errors, repairs errors and resends data as a new file using a new Message ID if so required by the IS.
  - o negative on content: notifies party initiating file transfer and rectifies issue as appropriate.

## 2.2 Responsibilities of Importing CMOs

Importing CMOs will be responsible for:

- accepting all reasonable requests to carry out tests;
- liaison with the appropriate parties in respect of any security requirements as defined by the ITM;
- specifying the tests between CMOs;
- planning and specifying the tests and expected results within their own systems;
- executing the tests at the Importing CMO end (receiving flows output from and sending acknowledgement flows to Exporting CMOs);
- supporting queries from the Exporting CMO and resolving issues on the content of the data exchanged and the progress of the test; and
- providing to the AIB an audit trail of each test and overall test results.

### 2.2.1 Data Receipt

The Importing CMO is responsible for the correct handling of the order, processing of these details and Acknowledgement to the Exporting CMO. The Importing CMO:

- checks for incoming messages regularly in accordance with the ITM;
- verifies the correctness of messages received in accordance with the ITM;
- as required by the ITM, in case the message is correct:
- sends a positive acknowledgement to exporting CMO; and
- stores certificates on the account of the buyer.
- as required by the ITM, in case the message is not correct:
  - o sends a negative acknowledgement to exporting CMO; which is either:
  - o negative on the message (e.g. failure on format); or
  - o negative on the content (e.g. failure on data).

## 2.3 Responsibilities of the AIB

The AIB will be responsible for:

- preparation of the ITS (this document);
- organisation of any security requirements as defined by the ITM.
- ensuring that CMOs satisfactorily complete the tests with other CMOs;
- collation of the test results provided by Exporting and Importing CMOs to provide an overall test result;
- issue resolution, if required;

- overall management of the process; and
- authorisation of valid interfaces for live use following successful testing.

### **3 INTERFACE CATEGORIES**

There are two types of interface between the CMOs that must be tested. The electronic file transfer system for operational data is supported by a manual interface for the exchange of configuration data such as encryption keys. Whilst not an 'operational' interface, effective communication of configuration details is essential to the correct operation of the CMO systems.

#### **3.1 Electronic File Transfer**

CMO services utilise transfers of files specified in the IS and transported in accordance with the ITM. These files are sent in both directions between the CMOs.

CMOs shall provide a test environment which will:

- send syntactically correct messages as specified to other CMOs. Note that these messages may not contain representative business data, indeed it is likely that the similar flows will be used in each test to each CMO. The Exporting CMO systems will then monitor the receipt of an acknowledgement from the Importing CMO as specified in the ITM.
- send syntactically incorrect messages to other CMOs. The Exporting CMO system will then monitor the receipt of a rejection message from the Importing CMO as specified in the ITM.
- receive messages from CMOs, validate their contents, send an acknowledgement or rejection as appropriate and report the results of the validation as specified in the ITM.

#### **3.2 Manual Data Transfer**

Configuration of interfaces will require a manual mechanism. The information may be delivered by mail, by telephone, by email or by fax from one person to another. This data may be sent in either direction between the CMOs.

CMOs are expected to use the configuration data that they are planning to use in live operation to receive and generate data.

### **4 TEST SYSTEM CONFIGURATION**

Each CMO is responsible for the preparation of test data and a test environment.

#### **4.1 Test Scripts and Data**

Test data will be required by all CMOs. Exporting CMOs should have planned and prepared test scripts describing how they are to generate the data to be sent to other CMOs and the expected results on receiving data from the Importing CMO. Additionally, Importing CMOs

shall have planned and prepared test scripts describing how they are to respond to the data received from Exporting CMOs and the expected results on receiving data.

CMOs are expected to use the same applications, procedures and local working instructions that will be used in live operation to receive and generate flows and to generate acknowledgements or rejections. However, it is up to individual CMOs to decide whether they can generate the required flows without the need to run their full business processes - the data must only be syntactically correct, not necessarily representative business data.

## 4.2 Test Environment

Each CMO must provide a test environment on their live communication system (this may be a number of temporary participant accounts used solely for testing purposes). The use of a mirror system with identical functionality is not appropriate as it would have to be configured to use identical communication settings to the live system, introducing a potential system conflict with other operational CMOs.

The AIB may authorise use the use of a mirror system on a test communication system if a CMO is unable to support testing on their live system. This environment must be using the same software applications as are to be used in production, and the database configured in a similar way. Use of a mirror system must be confirmed with the AIB prior to commencement of testing.

Initial contact, with the other CMO, will determine the preferred mechanism of manual data transfers.

## 4.3 Systems and Procedures

CMOs are expected to use the same procedures and systems as will be used in live operation to:

- receive input flows and generate acknowledgements or rejections; and
- generate output messages.

# 5 TEST CONDUCT AND PROCESSES

This section defines the conduct of tests that will be carried out for each CMO interface and the test processes to support testing.

## 5.1 Conduct of Tests

Each test will be the subject of a test script identifying the test cases to be undertaken. Each test will not be tested for continuity whether it is an output or input flow. A file will be validated against the IS and no checks will be made for dependency on a previous flow, with the exception of acknowledgement files. Every flow received by Importing CMOs will have an associated acknowledgement flow with the appropriate header information.

### 5.1.1 Electronic Data Interchanges

For electronic data interchanges, CMOs must pass all tests within the specified time slots i.e. both:

- provide correct responses (either acknowledgement or rejection) to flows the Exporting CMO sends; and
- provide valid data for all flows the Importing CMO receives.

If any tests cannot be completed in the specified time slot, the CMOs involved must investigate the reasons for delay.

For these tests the Exporting CMO will transmit to Recipient CMOs files containing data covering the test cases identified in the scripts. This will cover both valid and invalid data. The Importing CMO will validate the data and respond with an acknowledgement as appropriate.

The Exporting CMO will then confirm that the expected responses (acknowledgement with correct rejection reason if appropriate) are received from the Importing CMO. The Importing CMO and AIB test that their systems behave in the expected manner.

## 5.2 Test Processes

### 5.2.1 Problem Reporting and Problem Management

The CMO will log all test failures observed at the CMO end. It will be the responsibility of the CMO to track the problem resolution in their systems/processes and to ensure that the problem is fixed before the scheduled retest.

### 5.2.2 Problem Escalation

Any dispute arising from a CMO failing a test will be subject to the following escalation route:

- firstly the Exporting CMO with the Importing CMO will attempt to resolve the problem.
- if the dispute is not resolved, it will be escalated to the AIB Test Manager.
- if the dispute is still not resolved it will be referred to the full AIB.

### 5.2.3 Test Result Reporting

CMOs will be responsible for reporting the progress of the test to the AIB.

For each flow tested, as defined within the Interface Specification and specific test scripts, the CMO will report one the following test results.

- Pass
  - o For a flow output from the Exporting CMO, Exporting CMO received the expected reply (acknowledgement of valid data, rejection of invalid data).
  - o For a flow input to Importing CMO, that the data was processed as expected and the appropriate acknowledgement transmitted.

- Fail
  - o For a flow output from the Exporting CMO, no response was received or the response was not that expected.
  - o For a flow input to Importing CMO, that the data was not as expected or no response was created.

In this case a copy of the CMO Observation Report detailing the problem will be attached to the report.

An overall test result covering all the flows within the test will be reported:

- Pass
  - o Tests for all flows within the slot were successful.
- Fail
  - o If any tests fail, the CMO must rebook another slot and repeat all tests.

#### 5.2.4 Witnessing and Evidence

There is no requirement for on-site witnessing of CMO tests by the AIB. CMOs are expected to self-witness tests, and to provide the necessary results/reports/evidence directly to the AIB Test Manager.

## 6 TEST DETAILS

This section defines all tests that must be performed with associated success criteria.

### 6.1 Interface Configuration

The mechanism used to send and receive files is defined in the ITM.

CMOs are responsible for the configuration and maintenance of firewalls to manage the allowed connections from all other CMOs.

CMOs must be issued with appropriate configuration details for all parties to a set of tests. It is the responsibility of the transmitting CMO to ensure the successful completion of the transfer.

#### 6.1.1 Test 1. CMO-CMO Interface Setup

##### Test Elements

The two CMOs involved in the test shall exchange the requisite information required for the configuration used for the transmission of data to CMOs.

Each CMO shall:

- liaise with the other CMO to establish required configuration details as both sender and recipient;
- establish contact with the other CMO by use of the correctly configured interface;
- send a correctly configured single certificate transfer notification; and

- respond as appropriate with an acknowledgement to any test message received.

#### Success Criteria

Each CMO can:

- successfully send files to the other CMO;
- issue acknowledgements for correctly configured files that are received; and
- correctly not retransmit data if an acknowledgement is received.

### 6.1.2 Test 2. CMO-CMO Interface Validation

#### Test Elements

The two CMOs involved in the test shall exchange simple valid and invalid test files to test the procedural functions of the interface.

Each CMO shall:

- establish contact with the other CMO by use of the correctly configured interface;
- send a correctly configured single certificate transfer notification intended for a different CMO;
- send an incorrectly configured single certificate transfer notification where the error is contained within either header or footer information but not in the record section; and
- respond with an acknowledgement to the test messages received and disable acknowledgements to trial retransmission procedures.

#### Success Criteria

Each CMO can:

- successfully send files to the other CMO;
- identify and issue acknowledgements for incorrectly configured files that are received;
- correctly retransmit data if an acknowledgement is not received; and
- correctly not retransmit data if an acknowledgement is received.

## 6.2 Interface Operation

### 6.2.1 Test 3. CMO-CMO Valid Data Volumes

#### 6.2.1.1 Test Elements

The two CMOs involved in the test shall exchange complex valid test files to test the procedural functions of the interface.

Each CMO shall:

- establish contact with the other CMO by use of the correctly configured interface;
- send at least 10 correctly configured certificate transfer notifications in the EECS file format, using a variety of Technology and Earmark codings. All volume types specified below must form part of the test:

- o 10 certificates contiguous per file;
  - o 100 certificates contiguous per file;
  - o 1,000 certificates (non-contiguous, comprised of at least 20 blocks); and
  - o 10,000 certificates (non-contiguous, comprised of at least 50 blocks).
- send at least 10 correctly configured certificate transfer notifications in the old RECS file format, using a variety of Technology and Earmark codings and the volumes listed above.
  - respond as appropriate with an acknowledgement to any test message received.

#### **6.2.1.2 Success Criteria**

Each CMO can:

- successfully send files to the other CMO;
- issue acknowledgements for correctly configured files that are received;
- correctly not retransmit data if an acknowledgement is received; and
- correctly retransmit data if an acknowledgement is not received.

### **6.2.2 Test 4. CMO-CMO Invalid Data Volumes**

#### **6.2.2.1 Test Elements**

The two CMOs involved in the test shall exchange complex invalid test files to test the procedural functions of the interface.

Each CMO shall:

- establish contact with the other CMO by use of the correctly configured interface;
- send at least 10 incorrectly configured certificate transfer notifications in the EECS file format using a variety of Technology and Earmark codings. All volume types specified below must form part of the test:
  - o 10 certificates contiguous per file;
  - o 100 certificates contiguous per file;
  - o 1,000 certificates (non-contiguous, comprised of at least 20 blocks); and
  - o 10,000 certificates (non-contiguous, comprised of at least 50 blocks).
- send at least 10 incorrectly configured certificate transfer notifications in the old RECS file format, using a variety of Technology and Earmark codings and the volumes listed above.
- Error types must contain a variety of errors to include, but not be limited to, all of the following:
  - o invalid file structure or format;
  - o invalid recipient account;

- o incorrect dates;
  - o invalid production device; and
  - o invalid checksums.
- respond as appropriate with an acknowledgement to any test message received.

**6.2.2.2 Success Criteria**

Each CMO can:

- successfully send files to the other CMO;
- identify and issue acknowledgements for incorrectly configured files that are received;
- correctly retransmit data if an acknowledgement is not received; and
- correctly not retransmit data if an acknowledgement is received.