



Joint Council for  
Qualifications<sup>CIC</sup>



# JCQ<sup>CIC</sup> A2C Data Standards Specification

## Section 11

### Solutions Architecture

2022 Version

28<sup>th</sup> February 2022

# Table of Contents

---

1	Solutions Architecture .....	4
1.1	Overview .....	4
1.2	Context and Scope .....	4
1.3	Data Protection .....	4
2	Messages .....	5
2.1	Description .....	5
2.2	Message Schema.....	5
2.3	Message Header.....	5
3	Transactions.....	13
3.1	Description .....	13
3.2	Transaction Types.....	13
4	Message Sequencing, Consistency and Recovery .....	15
4.1	Context and Scope .....	15
4.2	Assumptions .....	17
4.3	Managing XSD upgrades.....	17
4.4	Roles.....	18
4.5	A2C Message Sequencing .....	18
4.6	Service Level Recommendations.....	22
5	Examples .....	23
5.1	Out-of-Sequence Scenario .....	24
5.2	Dealing with messages which are out of sequence for a defined time interval	25
5.3	Constructing an Out-of-Sequence XML message.....	27
5.4	Messages not conforming to the prescribed schema .....	28
5.5	Constructing a schema validation failure XML message.....	28
6	A2C Recovery Strategy.....	31
6.1	Recovery of lost messages.....	31
6.2	Repeated Messages.....	32
6.3	Reconciliation Strategy .....	32
6.4	Service Level Recommendations.....	33
7	Data interchange between Transport and Message Processing Component.....	34
7.1	Context and Scope .....	34

---

8	Hosted MIS Polling Solution.....	41
8.1	Overview .....	41
8.2	Context and Scope .....	41
8.3	Hosted MIS Processes .....	42
8.4	XML schema for hosted MIS data interchange.....	43
8.5	Example Messages .....	44
9	Centre Switch Over to A2C .....	45
9.1	Pre-A2C Transition Period .....	45
9.2	Post-A2C Transition Period.....	46
9.3	Final State: A2C Switch Over complete .....	47
10	Centre Set-up Notification and Request Product Catalogue .....	48
10.1	Rules for use of Centre Set-up Notification and Request Product Catalogue ...	48
10.2	Scenarios for Centre Set-up Notification and Request Product Catalogue .....	49
11	XML Schema Design.....	53
11.1	XSD Overview .....	53
11.2	Managing XSD upgrades.....	53
11.3	Element optionality.....	53
11.4	TransactionName .....	54
11.5	Feedback Messages.....	54
12	XML Schema Usage .....	56
12.1	Usage Rules.....	56
12.2	Definitions for data and feedback messages.....	59

# 1 Solutions Architecture

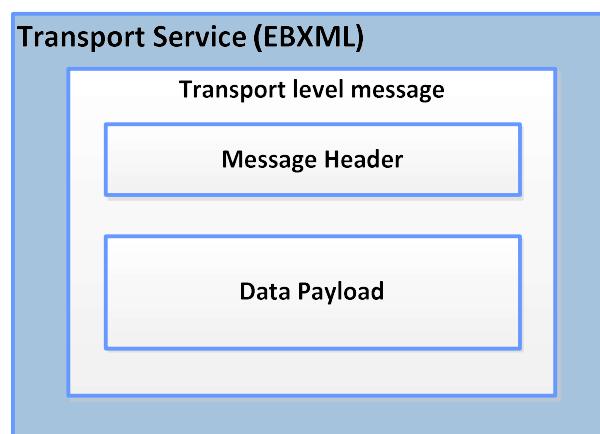
## 1.1 Overview

This document specifies how A2C messages and transactions are used to control and audit the data exchanges between centres and awarding organisations. This provides the underlying mechanism to ensure that the required business processes can exchange data in an efficient and consistent manner.

Note that where the A2C XML schema is referenced diagrams, descriptions and code are provided purely for the purposes of illustration. See the XSD file *A2CEntityBlockSchema*.

## 1.2 Context and Scope

A2C messages are constructed from a number of layers, each responsible for meeting a defined set of requirements. This is depicted in the following diagram:



- The A2C Transport Service is responsible for reliably transferring messages between systems. This service is based on the OASIS ebXML Messaging Services open standard and includes message sending and receipt acknowledgement protocols, where each party knows when a message has been successfully delivered. Included are guaranteed delivery Service Level Recommendations, and checks for integrity and duplication of messages.
- Transport level messages are the units of information conveyed by the A2C Transport Service.

The message consists of headers which are used for transport purposes; this header includes information related to the transport of data and security. The message also has a data payload attachment containing the real business data.

## 1.3 Data Protection

Data protection for data 'at rest' in an MIS is the responsibility of the MIS and centre. The AO assumes responsibility for data in transit.

## 2 Messages

### 2.1 Description

A business level message is carried in the transport as a data payload from one party to another.

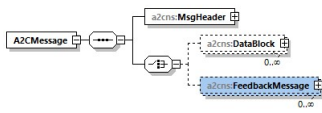
The content of the payload is referred to as a message in this section. A message is the unit of data transfer across the A2C Transport from one party to another. It is identified by a unique message identifier and includes a timestamp when the message was created in the source system and a sequence number that indicates the order of message generation.

Every message contains either a transaction request or response, which contains the business data required for the specific business process. Only one such request or response is allowed in a message.

A2C messages must be encoded using utf-8. If any other encoding is used or if the xml contains any non utf-8 characters the receiving system would return an NCMN.

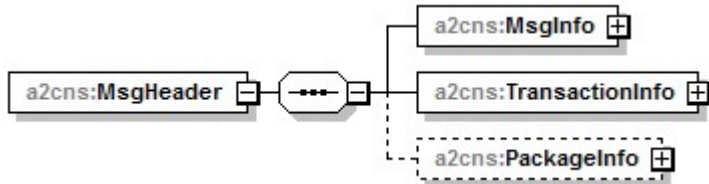
### 2.2 Message Schema

element **A2CMessage**

diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
properties	content complex
children	<b>a2cns:MsgHeader a2cns:DataBlock a2cns:FeedbackMessage</b>
annotation	documentation

### 2.3 Message Header

element **A2CMessage/MsgHeader**

diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
properties	content complex
children	<b>a2cns:MsgInfo a2cns:TransactionInfo a2cns:PackageInfo</b>

element **A2CMessage/MsgHeader/MsgInfo**

diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
properties	content complex
children	<b>a2cns:MessageID a2cns:TimeStamp a2cns:Sequence a2cns:Initiator_Data a2cns:Receiver_Data a2cns:SchemaVersion a2cns:Exchange_Name a2cns:ExchangeSpec_Version</b>

element **A2CMessage/MsgHeader/MsgInfo/MessageID**

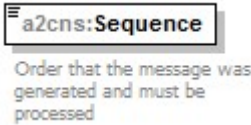
diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
type	<b>xs:string</b>
properties	content simple
annotation	Documentation Unique identifier for this message as a UUID string

element **A2CMessage/MsgHeader/MsgInfo/TimeStamp**

diagram	
---------	--


namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
type	<b>xs:dateTime</b>
properties	content simple
annotation	Documentation Date and time this message was created

#### element **A2CMessage/MsgHeader/MsgInfo/Sequence**


diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
type	<b>xs:integer</b>
properties	content simple
annotation	Documentation Order that the message was generated and must be processed

See 4.5 A2C Message Sequencing for details of sequencing.

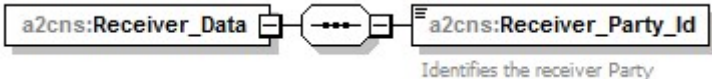
#### element **A2CMessage/MsgHeader/MsgInfo/Initiator\_Data**

diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
properties	content complex
children	<b>a2cns:Initiator_Party_Id</b>

#### element **A2CMessage/MsgHeader/MsgInfo/Initiator\_Data/Initiator\_Party\_Id**


diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
type	<b>xs:string</b>
properties	content simple
annotation	documentation Identifies the sender Party

#### element **A2CMessage/MsgHeader/MsgInfo/Receiver\_Data**

diagram	
namespace	<a href="http://jcq.org.uk/a2c">http://jcq.org.uk/a2c</a>
properties	content complex

children	<b>a2cns:Receiver_Party_Id</b>
----------	--------------------------------

element **A2CMessage/MsgHeader/MsgInfo/Receiver\_Data/Receiver\_Party\_Id**

diagram	
namespace	http://jcq.org.uk/a2c
type	<b>xs:string</b>
properties	content simple
annotation	documentation Identifies the receiver Party

The SchemaVersion to be used in all A2C messages must match that found in the Current Version Block annotation within the XSD. The actual text to be used for SchemaVersion is shown in quotes against XSD Version. See the example below from schema 10.5. Note that this value is a literal and could in future be populated with a text string such as 'XSD for the 2020/21 Academic Year'.

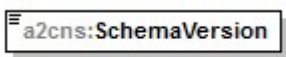
```

===== Current Version Block =====
Specification = JCQ A2C Data Standards Specification
XSD Name = A2CEntityBlocksSchema.xsd
XSD Release Date - 18 January 2018 Chris Wiltshire
SchemaVersion = "10.5"
Exchange_Name = "JCQ-A2C"
ExchangeSpec_Version = "2018"
=====

```

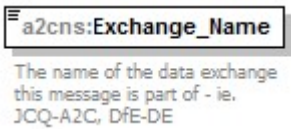
*Figure 1 Current Version Block*

element **A2CMessage/MsgHeader/MsgInfo/SchemaVersion**

diagram	
namespace	http://jcq.org.uk/a2c
type	<b>xs:string</b>
properties	content simple

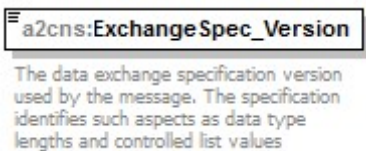


element **A2CMessage/MsgHeader/MsgInfo/Exchange\_Name**

diagram	 <p>The name of the data exchange this message is part of - ie. JCQ-A2C, DfE-DE</p>
namespace	http://jqc.org.uk/a2c
type	<b>xs:string</b>
properties	content simple
annotation	documentation The name of the data exchange this message is part of - ie. JCQ-A2C, DfE-DE

The ExchangeSpec\_Version to be used in all A2C messages must match that found in the Current Version Block annotation within the XSD, as in Figure 1 Current Version Block. The actual text to be used for ExchangeSpec\_Version is shown in quotes against Specification Version. Note that this value is a literal and could in future be populated with a text string such as 'Version for the 2020/21 Academic Year'.

element **A2CMessage/MsgHeader/MsgInfo/ExchangeSpec\_Version**

diagram	 <p>The data exchange specification version used by the message. The specification identifies such aspects as data type lengths and controlled list values</p>
namespace	http://jqc.org.uk/a2c
type	<b>xs:string</b>
properties	content simple
annotation	documentation The data exchange specification version used by the message. The specification identifies such aspects as data type lengths and controlled list values

element **A2CMessage/DataBlock**

diagram	
namespace	http://jcq.org.uk/a2c
type	<b>a2cns:DataBlock</b>
properties	minOcc 0 maxOcc unbounded content complex
children	<b>a2cns:DataBlockName</b> <b>a2cns:Locator_DS</b> <b>a2cns:Party_DS</b> <b>a2cns:PartyName_DS</b> <b>a2cns:PartyRelationship_DS</b> <b>a2cns:PartyRelationshipRole_DS</b> <b>a2cns:PartyRelationshipContact_DS</b> <b>a2cns:PartyRelationshipName_DS</b> <b>a2cns:QualificationElement_DS</b> <b>a2cns:QELearnerIdentifier_DS</b> <b>a2cns:QualificationElementAgeRange_DS</b> <b>a2cns:QualificationElementFramework_DS</b> <b>a2cns:QEAavailability_DS</b> <b>a2cns:QESubjectClassification_DS</b> <b>a2cns:QEBooking_DS</b> <b>a2cns:QELearnerBooking_DS</b> <b>a2cns:QEGradeSet_DS</b> <b>a2cns:QEOutcome_DS</b> <b>a2cns:ContributingQEOutcome_DS</b> <b>a2cns:QEAavailabilityGradeBoundary_DS</b> <b>a2cns:QEPreference_DS</b> <b>a2cns:QEPPerformanceMeasure_DS</b> <b>a2cns:QEQualificationCategory_DS</b> <b>a2cns:QELearningHours_DS</b> <b>a2cns:QEObjectiveStatement_DS</b>

element **A2CMessage/FeedbackMessage**

diagram	
namespace	http://jqc.org.uk/a2c
properties	minOcc 0 maxOcc unbounded content complex
children	<b>a2cns:Ref_Message_Id a2cns:FB_MessageLevel a2cns:FB_TransactionOrRecordLevel</b>

element **A2CMessage/FeedbackMessage/FB\_MessageLevel/FB\_Feedback**

diagram	
namespace	http://jqc.org.uk/a2c
type	<b>a2cns:FB_Feedback</b>
properties	minOcc 1 maxOcc unbounded content complex
children	<b>a2cns:FB_FeedbackCode a2cns:FB_FeedbackSeverity a2cns:FB_FeedbackText a2cns:FB_AdditionalInformation</b>

## element

**A2CMessage/FeedbackMessage/FB\_TransactionOrRecordLevel/FB\_TransactionOrRecord/FB\_Pri**

**maryRecord**

diagram	<p>This can be for either the Transaction data block (JCQ or ISB) or Record datablock (ISB). For a Transaction based message feedback this is always the Transaction identifying record within the primary datablock (eg. QELearnerBooking in a Named Order).</p>
namespace	http://jcq.org.uk/a2c
type	<b>a2cns:FB_Record</b>
properties	content complex
children	<b>a2cns:FB_DataBlockName</b> <b>a2cns:FB_EntityName</b> <b>a2cns:FB_RecordIDs</b> <b>a2cns:FB_Feedbacks</b> <b>a2cns:FB_AttributeLevel</b>

**element****A2CMessage/FeedbackMessage/FB\_TransactionOrRecordLevel/FB\_TransactionOrRecord/FB\_RelatedRecords/FB\_RelatedRecord**

diagram	
namespace	http://jcq.org.uk/a2c
type	<b>a2cns:FB_Record</b>
properties	minOcc 1 maxOcc unbounded content complex
children	<b>a2cns:FB_DataBlockName</b> <b>a2cns:FB_EntityName</b> <b>a2cns:FB_RecordIDs</b> <b>a2cns:FB_Feedbacks</b> <b>a2cns:FB_AttributeLevel</b>

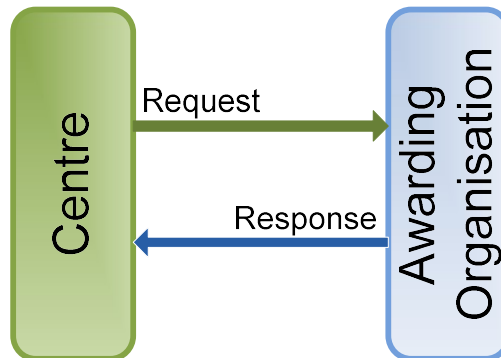
For information on how to use the schema to produce XML messages, refer to the *XML Schema Usage* section later in this document.

### 3 Transactions

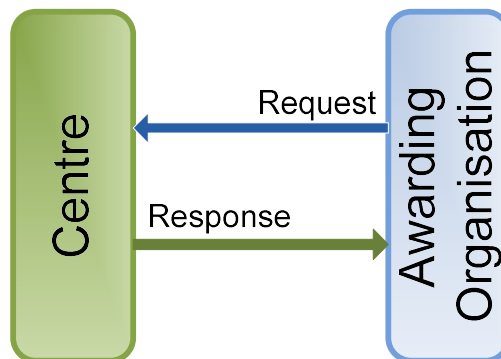
#### 3.1 Description

Transaction refers to a set of data relating to a business process which needs to be communicated by one party to another. One message can contain data related to one transaction type only.

A request message is always followed by a matching response. For example, a centre may send an order request to the awarding organisation, and the awarding organisation then sends a response back to confirm or refuse the order:



The same pattern is used in reverse, eg an awarding organisation sends a product catalogue update and the centre responds to indicate it has loaded the update:



Each transaction also has a unique identifier, which enables the request and response to be associated. The transaction identifier must be generated as a unique value by the originator, but only in the scope of that system. For consistency, a UUID-based identifier is required.

One transaction can contain one or more blocks of data. For example Attendance data can consist of data blocks for Assessment Event, Party Role and Attendance.

A response message carries the reference to the original request message as reference.

The Message Header contains information about the transaction: eg initiator and receiver of the data and transaction name.


#### 3.2 Transaction Types

element **A2CMessage/MsgHeader/TransactionInfo**

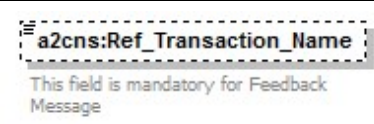


namespace	<a href="http://jqc.org.uk/a2c">http://jqc.org.uk/a2c</a>
properties	isRef 0 content complex
children	<b><u>a2cns:TransactionName</u></b> <b><u>a2cns:Ref Transaction Name</u></b>

### element **A2CMessage/MsgHeader/TransactionInfo/TransactionName**

diagram	 A diagram showing a dashed box around the text 'a2cns:TransactionName'. Below it, a note says 'Business Process Name - eg. ProcessCentreSetupNotification, ProcessNamedOrder etc.'
namespace	<a href="http://jqc.org.uk/a2c">http://jqc.org.uk/a2c</a>
properties	isRef 0
annotation	documentation eg ProcessCentreSetupNotification, ProcessNamedOrder etc

### element **A2CMessage/MsgHeader/TransactionInfo/Ref\_Transaction\_Name**

diagram	 A diagram showing a dashed box around the text 'a2cns:Ref_Transaction_Name'. Below it, a note says 'This field is mandatory for Feedback Message'.
namespace	<a href="http://jqc.org.uk/a2c">http://jqc.org.uk/a2c</a>
properties	isRef 0 minOcc 0 maxOcc 1
annotation	documentation This field is mandatory for Feedback Message

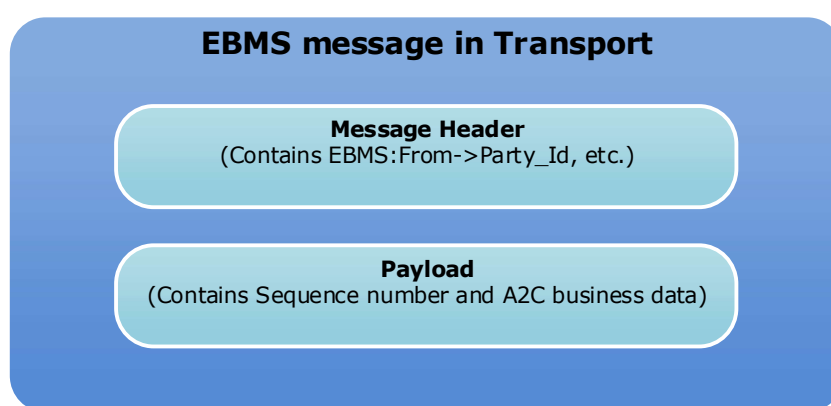
## 4 Message Sequencing, Consistency and Recovery

This document specifies how A2C will achieve the sequencing and control of data exchanges between centres and awarding organisations. This control provides the underlying mechanism to ensure that the required business processes can exchange data in an efficient and consistent manner.

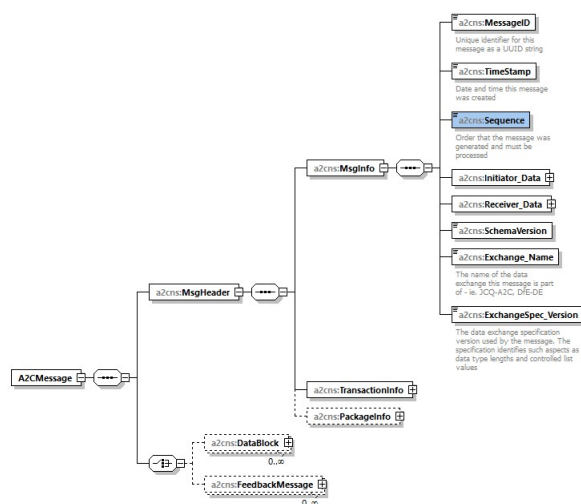
In considering various alternatives, this solution has been specified as the best balance between the control and consistency requirements versus complexity of implementation. Alternatives such as data-replication or data-locking strategies were considered too complex given the target environments.

### 4.1 Context and Scope

A2C business data messages are conveyed using A2C transport. The structure of the transport message carrying the A2C business level message is indicated below:



The payload contains the actual business data. The data within the payload is formatted as XML, the structure of which is indicated below:



The Transport Service is responsible for reliably transferring messages between systems. This service is based on the OASIS AS4 standard which is an open standard and includes message sending and receipt acknowledgement protocols, where each party knows when a message has been successfully delivered. The messages interchanged are called ebMS messages. This type of messaging ensures guaranteed delivery, and checks for integrity and duplication of ebMS messages. Hence a sending transport system cannot send the same ebMS message (containing the same ebMS MessageId) more than once.

The transport does not check for duplicate payloads, since it is content agnostic. Therefore a payload containing exactly the same data can be potentially sent more than once through different ebMS messages.

It is important that the receiving system processes the messages in the order they are generated at the source system, not in the order the messages are sent nor the order the messages are received. This ensures data consistency and avoids any risk of overriding new data with old.



## 4.2 Assumptions

The following assumptions have been made:

- MIS and awarding organisation systems must conform fully to this Specification.
- The principle of "I've started so I'll finish" applies once message exchanges are implemented between a centre and an awarding organisation using the A2C Data Standards. This means that if an order is received using the old EDI format then a result must be sent using the old EDI format: and if an order is received in A2C format then a result must be sent in A2C format. CAOs and other amendments will need to be transmitted in the format of the original order. There is no change to the Specification for EDI format files and these formats will not follow any sequencing rules.
- The A2C Transport provides a reliable and guaranteed delivery of messages, within its Service Level Recommendation (SLR). The sequence capability does not duplicate functionality of the Transport Service.
- An A2C business transaction does not span more than one message. There is no restriction to the size of messages or transactions implied by this Specification.

## 4.3 Managing XSD upgrades

- Awarding organisations will always upgrade XSD in advance of MIS and will support both the upgraded and previous version.
- There is currently no requirement for MIS to support more than one XSD version.
- Awarding organisations will track the XSD version for each centre and outgoing messages from the awarding organisation to the centre will always use the XSD version indicated in the last incoming message from the centre.
- The proposed use of the RPC message described in 10.2.6 below, will allow awarding organisations to track the centre XSD version and create outgoing messages to match the appropriate schema version.
- Where a centre upgrades XSD but fails to submit the RPC message as described in 10.2.6 below, the awarding organisation will assume that they are still operating with the previous XSD version. This may be rectified if the next message is a centre to awarding organisation message, and the header includes the XSD upgrade information.
- If the first message exchanged, following an upgrade of the centre XSD, is an awarding organisation to centre message, then a mismatch between awarding organisation and centre XSD is anticipated. The awarding organisation is unaware of the XSD upgrade and will have built the message according to the old schema. In those circumstances it is assumed that the centre will reject the awarding organisation message and will respond with feedback 0016.
- Where awarding organisations receive feedback 0016, they can identify, by reference to the header, whether the rejection is due to a mismatch of XSD version. Awarding organisations should then update their records for that centre and recreate the rejected message according to the appropriate schema. Note that the original message can be identified using the 'Ref\_Message\_Id' in the 'FeedbackMessage' block which is mandatory for a feedback message.

## 4.4 Roles

Three roles are defined for A2C message exchanges – centre, awarding organisation and hosted MIS:

1. The centre role represents the system being used by examination officers and related functions within learning centres. These systems are typically provided in the form of commercial Management Information Systems (MIS).
2. The awarding organisation role represents the system used to manage assessment and awarding-related functions with which the centre role interacts.
3. Hosted MIS role represents an MIS setup which manages the database for one or more centres.

These roles are used throughout this document as indicative of data exchanges required.

## 4.5 A2C Message Sequencing

The transport service does not guarantee delivery in the same order as the message is generated or sent, so sequence order must be used by the receiver to control the order of processing, so that the receiver can process the data in the order it was generated in the source system rather than in the order in which it is received. This will help the receiving system maintain data accuracy and consistency.

Every message (except the Hosted MIS-specific Poll for Centres of Interest) sent between A2C parties shall include an integer sequence number:

- A centre will maintain and own one sequence number for each awarding organisation for the outgoing messages. When the centre sends one message to a particular awarding organisation, the outgoing sequence number for that centre is increased by one. A centre will also maintain one sequence number for each awarding organisation for the incoming messages. This incoming sequence number will always reflect the sequence number of the last successfully processed incoming message from an awarding organisation. The incoming sequence number for an awarding organisation is increased by one only when an in-sequence message is received from the awarding organisation and the incoming message passes XML schema validation. When either an out-of-sequence message is received from an awarding organisation, or the message fails schema validation, the centre will not increment the incoming sequence number for that message.
- An awarding organisation will similarly maintain and own one sequence number for each centre for the outgoing messages. The sequence number relating to a centre is increased by one when a message is sent to that centre. An awarding organisation will also maintain one sequence number for each centre for the incoming messages. This information will always reflect the sequence number of the last successfully processed incoming message from a centre. The incoming sequence number for a centre is increased by one only when an in-sequence message is received from the centre and the incoming message passes XML schema validation. When either an out-of-sequence message is received from a centre, or the message fails schema validation, the awarding organisation will not increment the incoming sequence number for that message.
- Each centre managed by a hosted MIS will maintain sequence numbers in the same way they would if not managed by a hosted MIS. This will allow a centre to join a hosted MIS at any time without changing the sequence number and without informing the awarding organisations. This will also allow a centre which is already managed by a hosted MIS to adopt self-management. See heading 8: Hosted MIS Polling Solution below for more details.

A centre (centre number = 12345) dealing with Pearson (AO Id = 10) and AQA (AO Id = 70) can maintain a database similar to the following (indicative values for sequence numbers):

AO_Id	Outgoing_Sequence_No	Incoming_Sequence_No
10	100	20
70	120	45

An awarding organisation (AO Id = 10) can maintain a database similar to (indicative values for sequence numbers):

Party_Type	Party_Id	Outgoing_Sequence_No	Incoming_Sequence_No
Centre	12345	20	100
Centre	23456	45	120

#### 4.5.1 Sequence Number

The following must be included at the front of any message sent:

Attribute	Type	Description
Sequence	Unsigned Integer (32-bit)	Value 0 to 4,294,967,295 indicating the source creation order. First message sequence number for new connections is 1.

#### 4.5.2 Sequencing Rules

The following rules apply:

Rule	Description
Sequence	Messages must be created with contiguous sequence numbers $N$ , $N+1$ , $N+2$ etc for each destination. When the maximum is reached, the next value shall be 1.
Message Sequence	A single sequence number will occur within a message.
Business Significance	Messages must be created in business-significant sequence order at the source system, however the sequence number itself is only transitory and has no business significance.
Sending Messages	Any number of messages may be created and sent in rapid succession by one party to another party, providing sequence numbers are correctly included. No waiting for feedback message is required, eg an amendment may be allowed before the feedback to an order is received.
Receiver Processing Order	Messages must be processed by the receiver in sequence number order, to maintain sender's business-significant sequence.
Out-of-Sequence Messages	If a message is received out of sequence, it must not be processed until all earlier messages have been received and

Rule	Description
	processed. These messages may be kept in a holding area until all the earlier messages are received.
Duplicate Messages	<p>Duplicate messages can be tracked at both transport and at message processing component level. If a message is received with the same ebMS message identifier, transport safely discards it and it does not get passed on to the message processing level.</p> <p>If a message is received with the same sequence number as one of the previously received messages then the message is not processed further. A feedback message is sent back to the sender with the error '0009 - Message ignored: A message with same sequence number already received earlier'. The Transaction Name should be the normal transaction name for the feedback message, not ProcessOOSNotification.</p>
Multi-Centre Connections	Where several centres' messages share a single transport connection, eg through a hosted MIS solution, then each centre number/awarding organisation combination should still maintain its own sequence number.
Centres Merging	<p>If two or more centres merge and retain the centre number for one of the centres, then the sequence number for that centre should be used. For example centres 12345, 23456 and 34567 merge and the new centre number is 12345. In this case if 12345 has completed sequence number 100 for outbound messages before the merge, then the new centre should start from 101. Similarly, for inbound messages. All sequence numbers belonging to 23456 and 34567 become obsolete.</p> <p>If the centres merge to a completely new centre number instead of 12345, then a new sequence number should start from 1 which would contain a Centre Set-up Notification.</p>
Centres Splitting	<p>If a centre is split into two or more centres, then the following rules apply.</p> <ul style="list-style-type: none"> <li>• Supposing centre 54545 is split into 69991 and 69992 (assuming 69991 and 69992 did not previously exist) then both centres will start with sequence number 1 and the sequence number for 54545 will become obsolete</li> <li>• If 54545 is split into 54545 and 69991 then 54545 will continue the sequence numbering from where it was left before the split and 69991 will start a new sequence number from 1.</li> </ul>

Rule	Description
Awarding Organisations Merge	<p>If two awarding organisations merge, then the following rules apply.</p> <p>Assuming awarding organisations 10 and 11 merge to become 10:</p> <ul style="list-style-type: none"> <li>All centres continue to use the same sequence number as they were using for 10 for their outbound data to the awarding organisation. Awarding organisation 10 will follow on from the same sequence number as it was using with the centres for their outbound data to the centres</li> <li>If a centre is new to 10, whereas it was transacting with 11 previously, then that centre should start a new sequence number starting at 1 with 10, for their outbound data to the awarding organisation. Similarly the awarding organisation will start a new sequence number for that centre for its outbound data to the centre.</li> </ul> <p>If awarding organisations 10 and 11 merge to a new number 50, then:</p> <ul style="list-style-type: none"> <li>All centres will start a new sequence number starting at 1 for the outbound data to the awarding organisation</li> <li>The awarding organisation will also start a new sequence number for each centre for the outbound data to the centres.</li> </ul>
Centre Changes MIS Software	<p>When a centre changes from one MIS software to another, it makes a note of the last sequence numbers used with respect to all the awarding organisations with which it is dealing. The new MIS application is configured to commence the sequence numbering from where it was left.</p>
Sequencing of Non-Received Messages	<p>Where a centre does not provide a transport-level receipt for a received user message from an awarding organisation, on subsequent pull requests, the same message with identical sequence number will be retrieved from the awarding organisation, until the receipt is sent to the awarding organisation. Centres should not expect incrementing of sequence numbers in this scenario.</p>

Implementers should be aware that the rule for sending messages above may give rise to the following scenario:

1. Awarding organisation receives a Named Order from a centre containing a new learner at 10:00am with sequence number 100
2. Centre sends an amendment to the same learner at 10:02am with sequence number 101
3. Awarding organisation processes that Named Order (sequence 100) and rejects the transaction due to some problem with the learner data at 10:05am
4. Awarding organisation then processes the message containing the Amend Learner Detail at 10:06am (sequence 101). In this case the awarding organisation will reject the transaction saying that the learner does not exist.

This is a valid scenario. In this case the centre would fix the learner data as per the feedback received from the awarding organisation against the message with sequence number 100 and then send the Named Order transaction back to the awarding organisation.

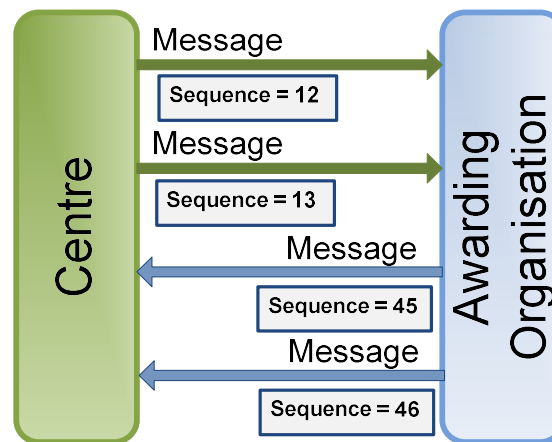
## 4.6 Service Level Recommendations

The messaging must conform to the following SLRs:

Service Level	Conformance
Out-of-Sequence Time Interval	<p>A receiving system (awarding organisation or centre) must wait until all the previous messages arrive, when an out-of-sequence situation persists.</p> <p>If an out-of-sequence situation persists for more than two hours, the receiving system sends a message containing the code '0003 – Messages out of sequence', informing the sender of all the messages received after the first out-of-sequence message received two hours ago.</p> <p>The sender must send all the missing messages, using their original sequence numbers, before sending any new messages (i.e. messages with a sequence number greater than the message that caused the OOS).</p> <p>The receiver will not accept any messages with a sequence number greater or equal to than the message that caused the OOS until the missing messages have been received and will respond with further 0003 feedback messages if any are received.</p> <p>If an OOS situation can not be resolved by resending missing messages, the AO and centre will agree to reset the sequence numbers once the integrity of both systems has been checked and they have agreed what needs to be done to restore synchronisation.</p>
Send Message Sequence Guarantee	<p>A sending system must guarantee that it sends all messages to the receiving system, regardless of interruptions. To achieve this, it must retain all messages indefinitely until they are successfully transferred and the receiver's transport has acknowledged receipt.</p> <p>This ensures no message sequence can be lost under normal conditions and connection interruptions.</p>

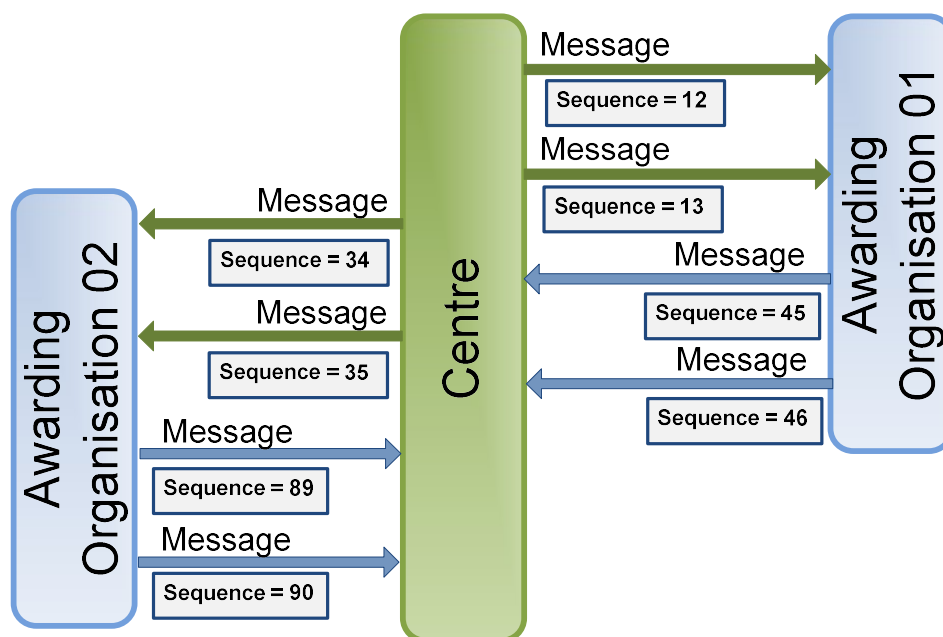
## 5 Examples

The following example shows a simple exchange of messages between a centre and an awarding organisation:



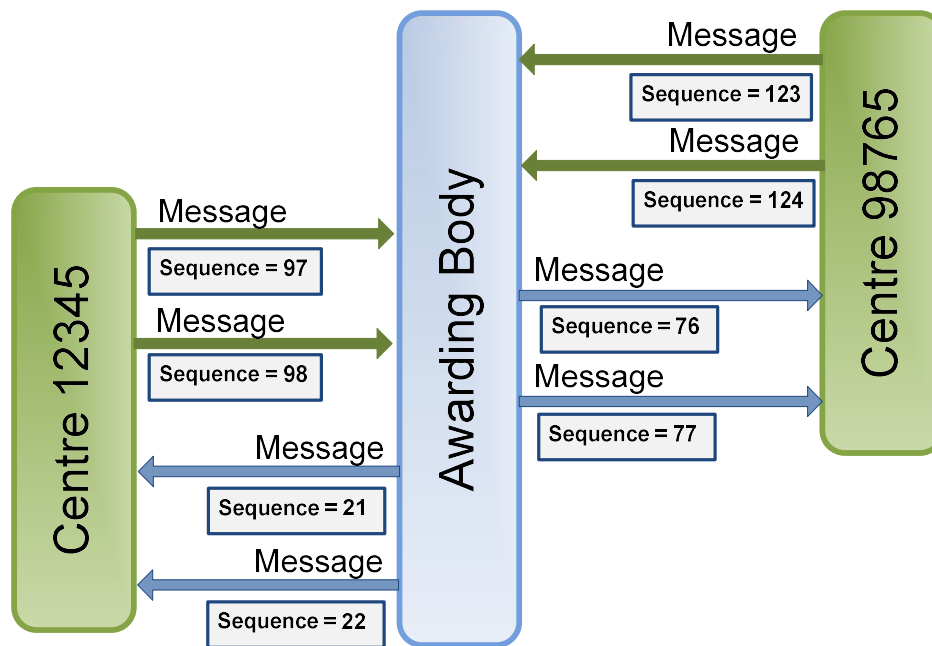
The messages sent from the centre have sequence numbers 12 then 13. The response messages from awarding organisation have a separate sequence of 45 then 46.

In the case of a centre dealing with two awarding organisations, separate sequences are maintained. For example:



Messages to the second awarding organisation are sent with sequence 34 and 35 and response messages have sequence 89 and 90. These sequence numbers are completely independent of the sequence numbers for the first awarding organisation exchanges.

The same situation occurs with an awarding organisation dealing with multiple centres.



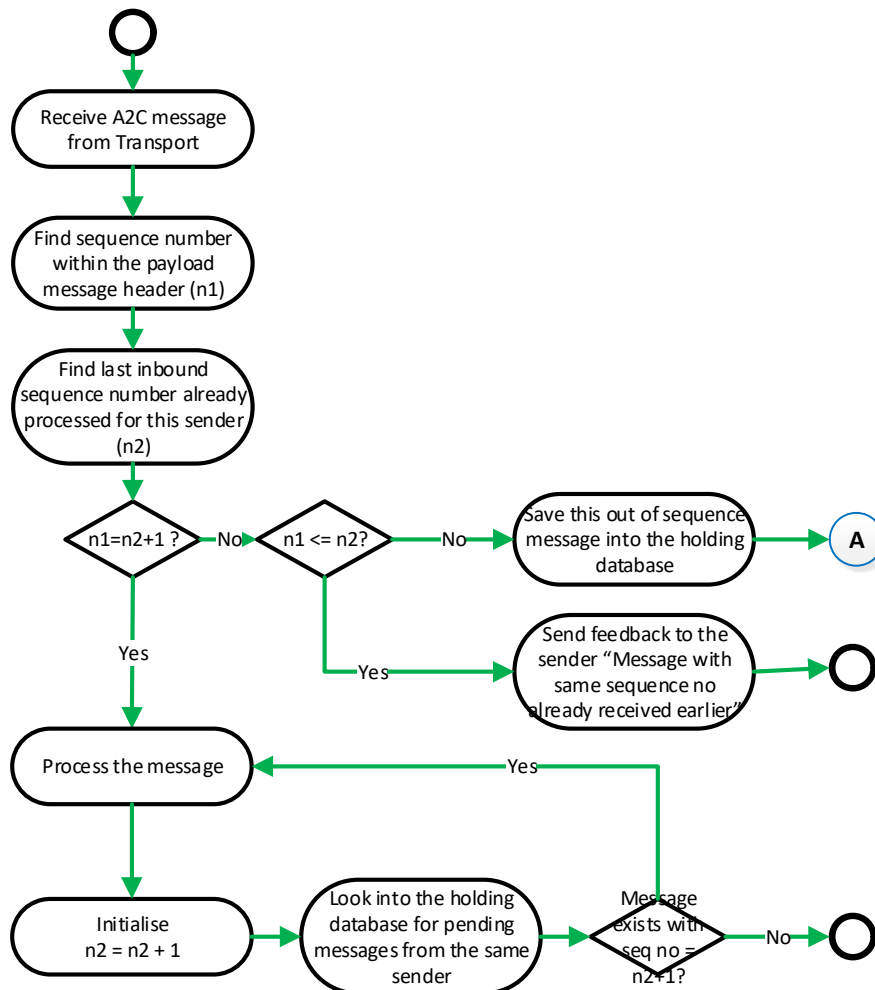
## 5.1 Out-of-Sequence Scenario

The out-of-sequence scenario only relates to the receiving system. The sending system is not considered here as it allocates the sequence numbers in order of business significance.

Where a message received is out of sequence it is saved into a holding database until all the previous messages are received and processed.

Guidelines on the order in which the messages are processed in a receiving system are given in the diagram below. Note that if at any stage it is identified that the message format is invalid (ie '0004 feedback is needed') then this overrides any considerations regarding the sequencing: sequencing is not incremented for invalid format messages.



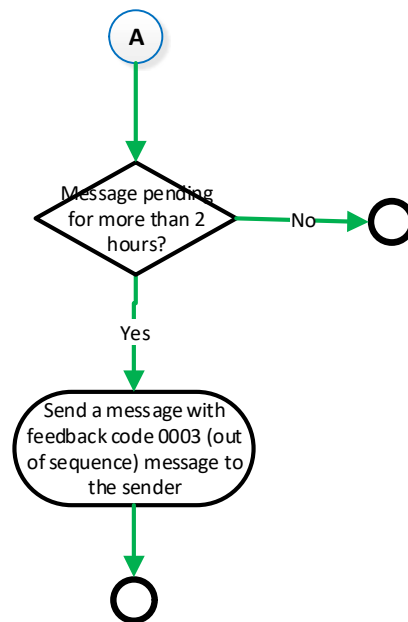


The above diagram depicts the following scenarios:

1. Message received from a sending system has sequence number 101. The last processed message is 100. Action: The message 101 is processed immediately and the last processed counter is initialised to 101.
2. If the last processed message is 100, and a new message received has sequence number 102, then this message will be parked in a holding database until the missing message(s) arrive.
3. If the last processed message is 100, and there are two messages in the holding database already from the same sender having sequence numbers 102 and 103, now if the message with sequence number 101 is received, all three messages (101, 102 and 103) are processed consecutively in order and the last processed counter is initialised to 103.

## 5.2 Dealing with messages which are out of sequence for a defined time interval

See Service Level Recommendations above



The above diagram specifies the action to be taken when a message is out of sequence for more than the recommended time interval.

As described above, an out-of-sequence message is sent to the sender if one or more messages have gone out of sequence for a period of time specified in the Service Level Recommendations. The message carries enough information so that the sender can work out the missing messages. This message will contain:

- Last message sequence number successfully processed
- List of missing message sequence number(s)  
(This lists the sequence numbers of all the missing messages after the last successfully processed message)
- List of message sequence number(s) parked in the holding area.  
(This specifies the sequence numbers of the messages which became out-of-sequence due to missing messages since the last successfully processed message).

Example:

Assume that the following messages are received at the awarding organisation system from centre 12345.

Received from Centre	Received Timestamp	Message Id	Sequence No	Status
12345	2014-08-14 10:00am	1212	100	Processed
12345	2014-08-14 10:20am	2323	102	Out of Seq
12345	2014-08-14 11:30am	3434	103	Out of Seq
12345	2014-08-14 12:10pm	4545	105	Out of Seq

The awarding organisation sends an out-of-sequence message to the centre after 2014-08-14 12:20pm.

The message contains:

- Sequence number of last successfully processed message = 100

- Sequence number(s) of missing message(s) = 101, 104
- Sequence number(s) of parked message(s) = 102, 103, 105.

Unlike feedback messages, an out-of-sequence message is not sent in response to a particular single message received from the sender. This message is considered a special type of notification to the sending system. Hence a specific transaction type is used for constructing this message. The transaction type used is '30 Out of Sequence Message Notification' (Refer Appendix 3). The action code used for this message is specified in Appendix 3 against this transaction type.

When an OOS message has been received, the system should not send any further messages until the situation has been resolved. Nevertheless, it is possible that another message has already 'crossed in the post'. If any further out of sequence messages are received after an OOS message has been returned, the receiver should respond with an additional OOS message for each message received.

The sending system takes responsibility for delivering any missing messages notified by the receiving system as soon as possible. If the missing messages do not arrive within a time period specified in the Service Level Recommendations, the receiving system takes manual action to contact the sender.

### 5.3 Constructing an Out-of-Sequence XML message

Refer to Appendix 3 and XML Schema.

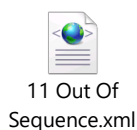
1. Message header (MsgHeader node in the Schema) is mandatory for all messages
  - a. Value of TransactionName will be as specified in Appendix 3 for Transaction Type 30 - Out of Sequence Message Notification
2. FeedbackMessage node is used in this message
  - a. Ref\_Message\_Id is not used for out-of-sequence messages
  - b. FB\_TransactionOrRecordLevel node is not used
  - c. FB\_MessageLevel node is used
  - d. FB\_FeedbackCode will be '0003'. For FB\_FeedbackSeverity and FB\_FeedbackText, refer to Appendix 7
  - e. For additional information specified above, FB\_AdditionalInformation node is used.

The possible values for FB\_InfoName are specified in Appendix 2. For the purpose of out-of-sequence messages the following values will be used:

- LastProcessedSequenceNumber
- MissingSequenceNumber
- PendingSequenceNumber
- DuplicateSequenceNumber

MissingSequenceNumber and PendingSequenceNumber can be repeated to specify more than one value. The number of MissingSequenceNumbers returned should be capped (recommended cap is 10) in case a message with a very large sequence number is received.

An example out-of-sequence XML message is attached below:



The above example also demonstrates that although there are two messages identified as missing (101 and 104), only one out-of-sequence message is enough to intimate these missing messages to the sending system.

## 5.4 Messages not conforming to the prescribed schema

It is possible that a message received does not conform to the A2C schema. This may be due to several reasons such as malformed XML, mandatory data missing and invalid format of the data, etc. It can also be due to a corrupt payload received. All these scenarios can be generalised as unreadable message payload. Even though some part of the message can be retrieved successfully, it does not guarantee accuracy of the data since it could have occurred due to several technical reasons.

It is also possible that, before the schema validation is performed, the schema version which is specified in the message header is either not readable or the version is not supported by the receiving system.

These scenarios are handled by completely rejecting the message and informing the original sender that an unreadable message is received. The sender information is always retrieved from the transport metadata. The sender is also provided with additional information to help it identify all the messages which are either missing or rejected due to schema validation.

This unreadable message creates a possible out-of-sequence scenario, because any further messages from the same sender may not be processed unless the new messages are next in sequence to the last successfully processed message. A special type of response message is used to help resolve this issue. A message with code 0004 'Message structure does not conform to A2C schema' is sent immediately as a follow up to the received unreadable message. Similar to an out-of-sequence message, this message will contain the last successfully processed message, a list of any currently out-of-sequence messages and a list of any pending messages waiting to be processed due to out-of-sequence scenarios for that sender.

A message with code 0004 advises the original sender that a message sent by it is not readable, and also instructs it to resend the missing messages, if any.

A schema validation failure message or a message with code 0004 is different from an out-of-sequence message in the following ways:

- Unlike an out-of-sequence message, the message validator at the receiving system does not wait for two hours to send the schema validation failure message when a message received fails schema validation.
- If an unreadable message is received when there is one or more messages from the same sender already pending due to an out-of-sequence situation, then the message with code 0004 also informs the sender about the missing messages. In this case a further out-of-sequence message with code 0003 need not be sent to the original sender.

Sending systems must validate outgoing messages against the schema before sending the messages to another system to avoid schema validation failure situations at the receiving system.

## 5.5 Constructing a schema validation failure XML message

This is similar to an out-of-sequence message.

Refer to Appendix 3 and XML Schema.

1. Message header (MsgHeader node in the schema) is mandatory for all messages
  - a. Value of TransactionName will be as specified in Appendix 3 for Transaction Type 31-Non-Conforming Message Notification
2. FeedbackMessage node is used in this message
  - a. Ref\_Message\_Id is not used for this message

- b. FB\_TransactionOrRecordLevel node is not used
- c. FB\_MessageLevel node is used
- d. FB\_FeedbackCode will be '0004'. For FB\_FeedbackSeverity and FB\_FeedbackText, refer to Appendix 7
- e. For additional information as specified above, FB\_AdditionalInformation node is used.

The possible values for FB\_InfoName are specified in Appendix 2. The schema validation failure message will use the following values:

- LastProcessedSequenceNumber
- MissingSequenceNumber
- PendingSequenceNumber

LastProcessedSequenceNumber should always be returned, to help the sending system reset their sequence numbering.

MissingSequenceNumber and PendingSequenceNumber should be sent only if the system is already out of sequence and there are messages being held in a queue.

Some example messages are as below:

Example 1: This example is based on the following scenario.

Received from Centre	Received Timestamp	Message Id	Sequence No	Status
12345	2014-08-14 10:00am	1212	100	Processed
12345	2014-08-14 10:30am	Unreadable message	Unreadable message	Schema validation failed

The XML message is attached below



11 Schema

Validation Failed Exa

Example 2: This example is based on the following scenario.

Received from Centre	Received Timestamp	Message Id	Sequence No	Status
12345	2014-08-14 10:00am	1212	100	Processed
12345	2014-08-14 10:20am	2323	102	Out of Seq
12345	2014-08-14 11:00am	3434	103	Out of Seq
12345	2014-08-14 11:30am	Unreadable message	Unreadable message	Schema validation failed

The XML message is attached below



11 Schema

Validation Failed Exa

In this case the receiving system does not further send an out-of-sequence message at 12:20pm, since the missing message details are already passed on to the original sending system through the above attached message.

## 6 A2C Recovery Strategy

The sequence mechanism and the underlying reliable Transport Service ensure that messages are exchanged in a robust and predictable manner. However, there are a number of exceptional circumstances that may require a message recovery to be necessary. These include:

- System fault where one or more messages get 'lost', even though the sender thought they were delivered successfully
- Manual processing loses a message, eg a user accidentally deletes a message before it is processed
- Following a system failure, a system restore sets the data back to an earlier state.

These situations may result in either a deadlock where data cannot be successfully delivered, or a condition that repeats message sequences.

### 6.1 Recovery of lost messages

Where a sending system has a higher sequence number than the next sequence expected by the receiver, the message exchange ends up in a deadlock. The sequencing rules do not allow the receiver to process the later data, and the sender is unaware that it needs to resend messages that were previously successfully delivered.

This section specifies the recovery strategy for this situation:

Strategy	Description
Sender Sequence Recovery	Recovery must be achieved in the sending system, to avoid lost data.
Manual Recovery Initiation	No automatic recovery capability is allowed. It is assumed that this would be required only rarely and must be initiated by manual actions.  A suitable administration user interface must be provided for this purpose.
Exceptional Recovery	Recovery must be the last resort when deadlocks are identified and confirmed. It is not intended as a normal user activity or work-around for unreliable connections.
Message Resend	In order to recover, messages must be resent from an earlier point in time. Any system must implement one of two strategies: <ul style="list-style-type: none"> <li>• Replay previous messages from the last known sequence or</li> <li>• Recreate equivalent messages from a particular point in time.</li> </ul>

A sending system is required to adopt one of the strategies described below.

#### 6.1.1 Store and Replay Previous Messages

Under this strategy, the sending system stores an exact copy of all messages sent (see SLRs). Recovery is achieved by identifying the last message sent, then resending all messages with larger sequence numbers. Note however that it may not be possible to replay stored messages created under a previous XSD version.

The sending system must also handle the potentially duplicated responses that may ensue.

### 6.1.2 Recreate Equivalent Messages

Under this strategy, exact messages need not be stored. Instead, the sequence number is first reset so the sender and receiver sequence numbers agree. Once done, all transactions that occurred after the problem point must be recreated by the sender as new messages and sent.

This strategy allows more flexibility, for example if a product order and two amendments were originally sent, then only the amended product order need be recreated. This achieves the correct data state in the receiving system.

## 6.2 Repeated Messages

Where a sending system has a lower sequence number than the next sequence expected by the receiver, new messages will be sent repeating sequence numbers already sent. This can potentially cause a number of issues that need to be addressed.

Strategy	Description
Receiver Sequence Recovery	Recovery must be achieved in the receiving system, to avoid undetected double-processing.
Manual Confirmation	<p>Where a receiver sees a sequence number that it has already processed, it must halt processing further messages and prompt the system user/administrator before continuing.</p> <p>Re-processing of messages may cause unexpected effects, so must not occur automatically.</p>
Duplicate Processing	<p>Following confirmation, when the awarding organisation processes a sequence that it has already processed, it must return a normal response according to the prevailing business process.</p> <p>In most cases this will be either a warning (duplicate) or error (no longer valid) result, depending upon the process (see process documentation for further detail).</p>

## 6.3 Reconciliation Strategy

This Specification details the sequencing and recovery mechanisms designed to prevent inconsistencies between systems. Additional mechanisms to allow for consistency checks and reconciliation may be provided through separate business processes, but this is out of scope for this document.



## 6.4 Service Level Recommendations

The following Service Level Recommendations (SLRs) must be adhered to:

Service Level	Conformance
Data Retention	A sending system must be able to resend data from a point within the last 28 days (unless the XSD has changed). Recovery beyond this is not required.
Deadlock Alerts	A deadlock situation must be alerted to the system user/administrator, such that corrective action can be taken. Deadlocks can be detected by an extended period waiting to handle out-of-sequence messages by a receiver.

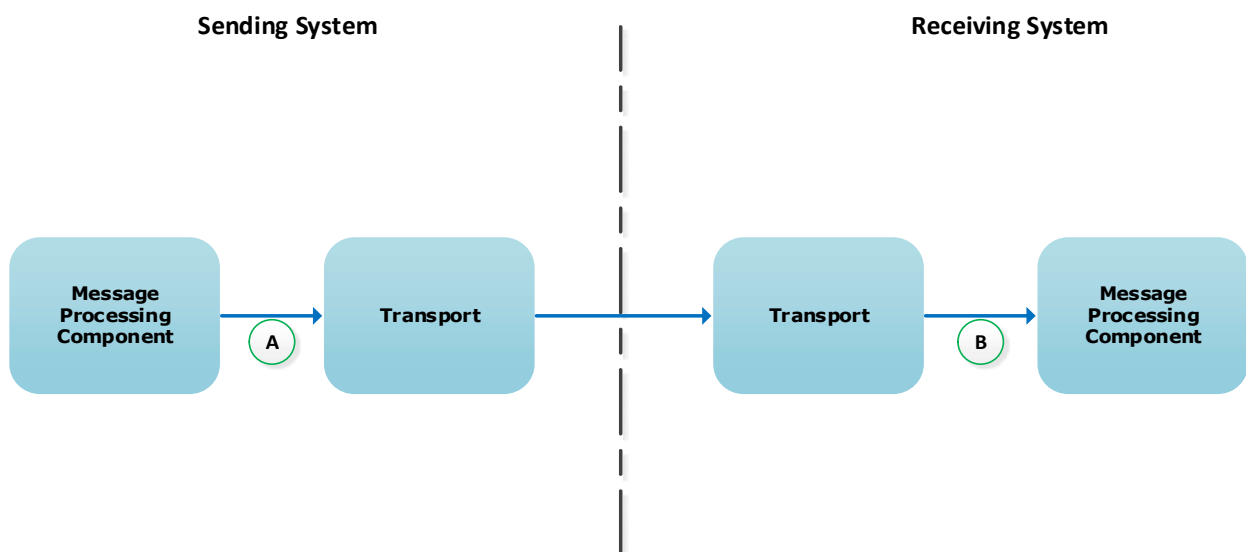
## 7 Data interchange between Transport and Message Processing Component

This section specifies how the A2C Transport Service is used in conjunction with the A2C Data Standards. It specifies the values and data model entities that must be used in order to enable the A2C Transport Service to successfully deliver the required business messages. It also specifies the transport metadata values that must be used by the message processing component in order to process the business data successfully.

### 7.1 Context and Scope

A2C messages are constructed from a number of layers, each responsible for meeting a defined set of requirements. The Transport Service is responsible for reliably transferring messages between systems. This service is based on the OASIS ebXML Messaging Services open standard and includes message sending and receipt acknowledgement protocols, where each party knows when a message has been successfully delivered. Included are guaranteed delivery Service Level Recommendations, and checks for integrity and duplication of messages.

The Message Processing Component (MPC) is responsible for creating messages containing business data for outbound messages and it is also responsible for validating and consuming inbound data received through transport. The following is the system diagram spanning across the sending and receiving systems, which shows the flow of message from one party to another.



This section is limited to the transport messages carrying business data payloads only (these are called UserMessage in transport).

### 7.1.1 Data required by Transport in the sending system

Refer to 'A' in the diagram

The following data needs to be passed from MPC to the Transport in order to enable Transport to send the data:

1. Party information
  - a. From Party Id
  - b. From Party role (This indicates whether the sender is a centre or awarding organisation or hosted MIS)
  - c. To Party Id
  - d. To Party role
2. Collaboration information
  - a. Service
  - b. Action
3. PackageInfo
  - c. Package Name
  - d. Package Version
4. MimeType.

The above attributes are defined in the *Transport Specification*.

As described in the *Transport Specification* the party identifiers can take the following form:

- a. jcq:ncn:centreNo for centre (eg jcq:ncn:12345 or jcq:ncn:12345A)
- b. jcq:ab:AOId for AO (eg jcq:ab:10)
- c. jcq:mis:MISId for hosted MIS ( eg jcq:mis:1234)

The following image shows how Transport utilises this data to construct the ebMS message:

```
POST https://a2c.pearson.com/A2CEbXmlService/ HTTP/1.0
SOAPAction: "ebms"
Content-Type: multipart/related; boundary=-----0a559ed27f9f4396859785e297979a59-a2c-messagewriter@avco; t
Host: a2c.pearson.com
Content-Length: 4431

-----0a559ed27f9f4396859785e297979a59-a2c-messagewriter@avco
Content-Type: application/soap+xml
Content-ID: <header>

<soap:Envelope xmlns:ebms="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/ ebms-v3.0.xsd">
  <soap:Header>
    <ebms:Messaging id="_1c2a658ef5e74ddd89bdca000d672ba5" soap:mustUnderstand="true">
      <ebms:UserMessage>
        <ebms:MessageInfo>
          <ebms:Timestamp>2013-10-01T09:48:56.5634312Z</ebms:Timestamp>
          <ebms:MessageId>7ec1bf6b6e3c46eb9e437bcf61ae8a96@avco</ebms:MessageId>
        </ebms:MessageInfo>
        <ebms:PartyInfo>
          <ebms:From>
            <ebms:PartyId>jcq:ncn:10102</ebms:PartyId>
            <ebms:Role>centre</ebms:Role>
          </ebms:From>
          <ebms:To>
            <ebms:PartyId>jcq:ab:10</ebms:PartyId>
            <ebms:Role>awardingBody</ebms:Role>
          </ebms:To>
        </ebms:PartyInfo>
        <ebms:CollaborationInfo>
          <ebms:Service>uri:jcq.org.uk/formats/edi/13</ebms:Service>
          <ebms:Action>A</ebms:Action>
          <ebms:ConversationId>1</ebms:ConversationId>
        </ebms:CollaborationInfo>
        <ebms:MessageProperties>
          <ebms:Property name="PackageName">A2C Migration Application</ebms:Property>
          <ebms:Property name="PackageVersion">1.0.0.22880</ebms:Property>
        </ebms:MessageProperties>
        <ebms:PayloadInfo>
          <ebms:PartInfo href="cid:d741126daa7545f7843926604ee98c4f@avco">
            <ebms:PartProperties>
              <ebms:Property name="MimeType">application/EDI-consent</ebms:Property>
              <ebms:Property name="Compressed">true</ebms:Property>
            </ebms:PartProperties>
          </ebms:PartInfo>
        </ebms:PayloadInfo>
      </ebms:UserMessage>
    </ebms:Messaging>
    <wse:Security>...</wse:Security>
  </soap:Header>
  <soap:Body />
</soap:Envelope>

-----0a559ed27f9f4396859785e297979a59-a2c-messagewriter@avco
```

### 7.1.2 Data required by MPC from Transport in the receiving system

Refer to 'B' in the diagram

The following data needs to be passed from Transport to MPC in order to enable MPC to process the business data received:

1. From Party Id
  2. From Party role
  3. To Party Id
  4. To Party role
  5. Received timestamp
  6. Collaboration information
    - a. Service
    - b. Action.
- From Party Id and role are used to route the feedback message generated against the original message back to the original sender.
  - Received timestamp can be used for various purposes. One of the examples is for the awarding organisation to compare this date with the order deadline date and determine whether to impose a penalty.
  - Collaboration information which contains the transaction name can be used to choose the right schema to validate the payload against.
  - From Party Id, To Party Id and the Collaboration information are used in the payload data validation.

### 7.1.3 Usage of Transport metadata in payload data validation

An example payload message received by the receiving system is as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<A2CMessage xmlns="http://jcq.org.uk/a2c" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jcq.org.uk/a2c A2CEntityBlocksSchema_Ver_7.02.xsd">

    *****
    Message Header
    *****

-->
<MsgHeader>
  <MsgInfo>
    <MessageID>xxxxxxxxxx</MessageID>
    <TimeStamp>2014-12-15T19:47:00Z</TimeStamp>
    <Sequence>01</Sequence>
    <Initiator_Data>
      <!-- For a Centre Setup this is a Centre and this example uses the AO Assigned Centre ID number which is 11111 -->
      <Initiator_Party_Id>11111</Initiator_Party_Id>
      <!-- Note - the Initiator IdentifierTypes are no longer used as only the Centre can vary its reference number that
is used and the particular one used is articulated in the below data. Therefore it is no longer necessary to separately supply
the IdentifierTypes and these have been removed in this version -->
      </Initiator_Data>
      <Receiver_Data>
        <!-- For a Centre Setup this is an Awarding Organisation using the
JCQ Awarding Organisation ID and this example uses C&G number which is 73 -->
        <Receiver_Party_Id>73</Receiver_Party_Id>
        </Receiver_Data>
        <!-- SchemaVersion fixed value controlled by JCQ -->
        <SchemaVersion>7.02</SchemaVersion>
        <Exchange_Name>JCQ-A2C</Exchange_Name>
        <ExchangeSpec_Version>1.2.2</ExchangeSpec_Version>
      </MsgInfo>
      <TransactionInfo>
        <!-- Transaction name is sourced from A2C Appendix 3, Transactions & Action Codes - note change
from ProcessA2CCentreSetup to ProcessCentreSetup -->
        <TransactionName>ProcessCentreSetup</TransactionName>
      </TransactionInfo>
      <PackageInfo>
        <!-- fictional supplier system data for example purposes only-->
        <Supplier>Example Supplier</Supplier>
        <System>Example System</System>
        <Version>Example System Version</Version>
      </PackageInfo>
    </MsgHeader>

  <!--
  *****
  *   Party Data Blocks - define all together
  *****
  -->
  <DataBlock>
    <DataBlockName>Party_DS</DataBlockName>
    <Party_DS>
      <Party>
        <!-- Defines Party blocks - order is irrelevant -->
        <!-- Define AO - AO number is 73 for C&G -->
        <Party_ID>
          <Party_Id>73</Party_Id>
        </Party_ID>
        <Party_Type>Organisation</Party_Type>
      </Party>

      <!-- Define Centre using the AO Assigned Centre ID of 11111 for this example -->
      <Party>
        <Party_ID>
          <Party_Id>11111</Party_Id>
        </Party_ID>
        <Party_Type>Organisation</Party_Type>
      </Party>

      <!-- Define JCQ org -->
      <Party>
```

```

    <Party_ID>
    <Party_Id>JCQ</Party_Id>
  </Party_ID>
  <Party_Type>Organisation</Party_Type>
</Party>

</Party_DS>
</DataBlock>
<!--
*****
*   Party Relationship Data Blocks
*****
-->
<DataBlock>
  <DataBlockName>PartyRelationship_DS</DataBlockName>
  <PartyRelationship_DS>

```

<!-- in the Party Relationship - the first party is always the one who assigns values to the second. This becomes more obvious in the Party Relationship Role that uses the Party Relationship structure and then builds on it by assigning the reference number from the first party to the second or a role being assigned to the second party by the first party. Therefore, the Party Relationship Role must be a replica of the Party Relationship in terms of the Party Ids used and their order. So it can be easier to work out the Party Relationship Role first and then cut it down for the related Party Relationship instance. -->

```

  <!-- Join JCQ and AO-->
  <PartyRelationship>
    <PartyRelationship_ID>
      <Party_Id_1st>JCQ</Party_Id_1st>
      <Party_Id_2nd>73</Party_Id_2nd>
    </PartyRelationship_ID>
  </PartyRelationship>

  <!-- Join AO and Centre-->
  <PartyRelationship>
    <PartyRelationship_ID>
      <Party_Id_1st>73</Party_Id_1st>
      <Party_Id_2nd>11111</Party_Id_2nd>
    </PartyRelationship_ID>
  </PartyRelationship>

```

```

</PartyRelationship_DS>
</DataBlock>

```

```

<!--
*****
*   Party Relationship Role Data Block
*****
-->

```

```

<DataBlock>
  <DataBlockName>PartyRelationshipRole_DS</DataBlockName>
  <PartyRelationshipRole_DS>

```

<!-- Using the above Party Relationship Join of JCQ and AO let us expand on that -->

```

  <PartyRelationshipRole>
    <PartyRelationshipRole_ID>
      <Party_Id_1st>JCQ</Party_Id_1st>
      <Party_Id_2nd>73</Party_Id_2nd>
      <Party_Role_Type>Awarding Organisation</Party_Role_Type>
    </PartyRelationshipRole_ID>

```

<!-- The Relationship\_Reference below is the same value as the Party Id we are using for the AO - that must mean that the AO Party\_Id being used is their "JCQ Awarding Organisation ID" number as we can extrapolate that from the next two lines -->

```

    <Relationship_Reference>73</Relationship_Reference>
    <Party_RR_Reference_Type>JCQ Awarding Organisation ID</Party_RR_Reference_Type>
  </PartyRelationshipRole>

```

<!-- Using the above Party Relationship Join of AO and Centre let us expand on that -->

```

  <PartyRelationshipRole>
    <PartyRelationshipRole_ID>
      <Party_Id_1st>73</Party_Id_1st>
      <Party_Id_2nd>11111</Party_Id_2nd>
      <Party_Role_Type>Centre</Party_Role_Type> <!-- This is not as per the ISB Party Role Type list which
has Assessment Centre. JCQ/ISB to resolve -->
    </PartyRelationshipRole_ID>

```

<!-- The Relationship\_Reference below is the same value as the Party Id we are using for the

Centre - that must mean that the AO Party\_Id being used is their "AO Assigned Centre ID" number as we can extrapolate that from the next two lines -->

```
<Relationship_Reference>11111</Relationship_Reference>
<Party_RR_Reference_Type>AO Assigned Centre ID</Party_RR_Reference_Type>
</PartyRelationshipRole>

</PartyRelationshipRole_DS>
```

<!-- Okay that's all we need for the Centre Setup Transaction. -->

```
</DataBlock>
</A2CMessage>
```

To ensure that the business data received within the payload is processed in the context of correct sender and receiver, the MPC in the receiving system will compare the following data:

1. From Party Id from the transport metadata with Initiator Party Id in the message header of the payload.
2. To Party Id from the transport metadata with Receiver Party Id in the message header of the payload.
3. Action code from the transport metadata with the TransactionName in the message header of the payload.

If these data do not match then the payload data is considered invalid or corrupt. In this case an appropriate feedback is sent to the sending system. (Refer to the Feedback section for details of error code used.)



## 8 Hosted MIS Polling Solution

### 8.1 Overview

Hosted MIS polling enables Management Information System (MIS) providers who run multi-tenanted systems for multiple centres on a converged infrastructure to reduce the burden of making individual polls per centre by using the Poll for Centres of Interest query.

The following should be noted:

- The Poll for Centres of Interest transaction type follows the same transport principles as any other A2C message. It is carried as a payload in a UserMessage within the EBXML SOAP construct, in the same way as any other A2C message. The message from a hosted MIS is signed using an A2C access key supplied by the awarding organisation specific to the hosted MIS system.
- The payload message does not follow the normal A2C XSD schema, eg it has no sequencing, and does not use data blocks. It uses a very brief and simple format set out in 8.4 below.
- It does not generate any message feedback.
- At the awarding organisation system, it will generate either a response message listing centres for whom messages are available, which may be a list with no content, or a transport level error message if the initial query was invalid for some reason. The response messages are delivered to the hosted MIS as a UserMessage with the payload as an attachment to the SOAP message, in the same way as any other A2C message.
- If the format of the Poll for Centres of Interest sent by the Hosted MIS was invalid, the message may simply be ignored by the awarding organisation.
- The Poll for Centres of Interest transaction does not exchange any business data.

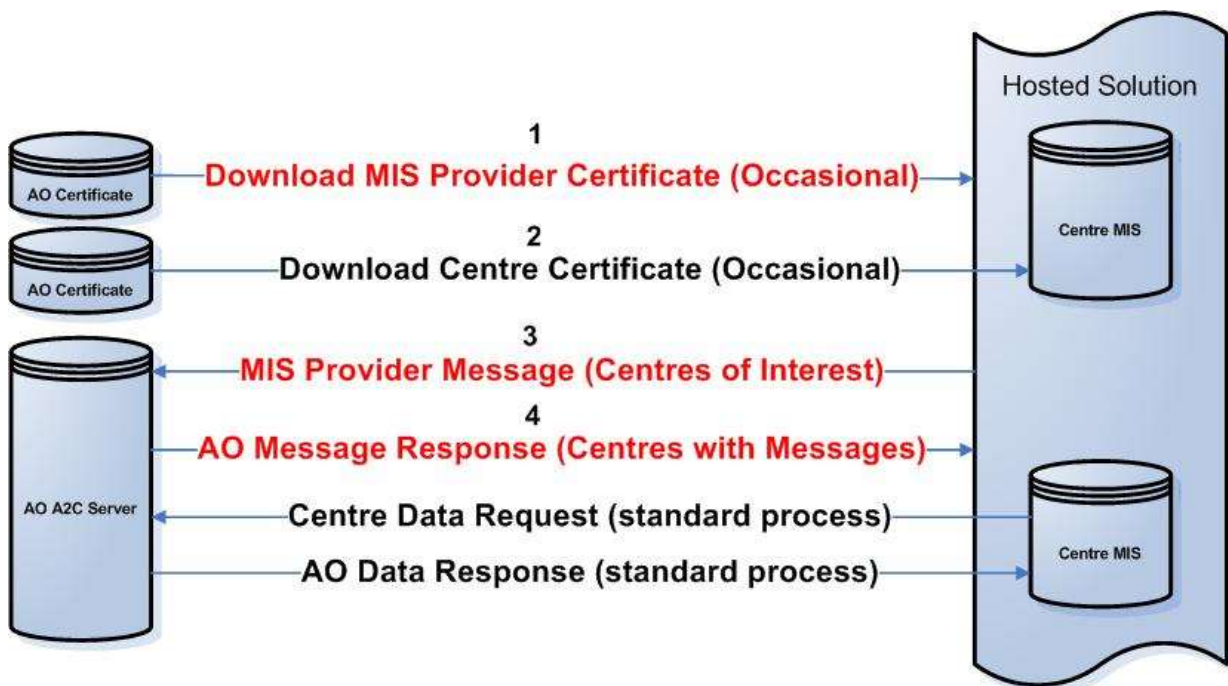
### 8.2 Context and Scope

- The A2C Transport Solution is a point to point relationship between the centre and the awarding organisation.
- Some MIS providers provide their services via a consolidated infrastructure (act as agents of centres).
- The point to point approach has implications for the volume of traffic and the number of credentials required for transactions between these hosted MIS Providers and awarding organisations.
- MIS provider preference is to eliminate the need to poll the awarding organisation system once per each centre separately, by polling only once on behalf of all the centres they are managing, by acting as a proxy in the A2C environment.
- The issue for awarding organisations is to assure the integrity of the end point transmission to the centre with which their business relationship exists.
- Multi-tenanted MIS must implement the Hosted MIS Polling Solution.
- Centres with MIS that are hosted in the cloud but are completely isolated from other centres are not required to use the Hosted MIS Polling Solution.
- On-premises setups for centres with many centre numbers may use the Hosted MIS Polling Solution but are not required to do so.
- The Poll for Centres of Interest must poll each AO ID separately even if certificates cover more than one AO ID.

- Where an individual centre is sending data to an awarding organisation then the centre will also issue a pull request, automatically retrieving any outstanding data. This hosted MIS polling solution is therefore not required for individual centres.
- As this process is outside the business relationship between the awarding organisations and centre, it is not reflected in the business data model.

### 8.3 Hosted MIS Processes

1. On request, the awarding organisation will generate an MIS authentication certificate: this allows the hosted MIS to be validated as a known end point.
2. Centre downloads Access Key (as they would for a stand-alone setup) and provides Centre Access Key to MIS.
3. Hosted MIS solution requests data for all those centres it hosts (Centres of Interest).
4. Awarding organisation responds with a list of centres where data is awaiting collection (Centres with Messages). This list will be a subset of the centres sent by the MIS in step 3.
5. For the Centres with Messages, the MIS defaults to behaving in the same way as non-hosted MIS systems.



For operations 3 and 4, a new EBXML service and action are defined in Appendix 3, Action Codes. For ease of reference these are:

	Service	Action
Hosted MIS to AO	uri:jcq.org.uk/misQuery	ProcessPollCentres
AO to Hosted MIS	uri:jcq.org.uk/misQuery	RespondPollCentres

*Table 1 Service and Action Codes for Hosted MIS Polling Solution*

These operations are only available to the hosted MIS certificate type as well as being the only operations available to those types of certificate.

## 8.4 XML schema for hosted MIS data interchange

For data interchange between hosted MIS and Awarding Organisation an XML document is to be provided as the payload of the EBXML message, conforming to the following schema:

```
<xs:element name="MessagesAvailable" type="CentreListType"/>
<xs:complexType name="CentreListType">
  <xs:sequence>
    <xs:element name="CentreNumber" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CheckForMessages" type="CentreListType"/>
```

This schema is published as misquery.xsd.

### 8.4.1 MIS Provider Certificate

- On request, each awarding organisation will issue MIS providers with a standard A2C certificate as defined in section 12 Transport Specification, heading 9 'Certificates / Access Keys'. However, as stated above, this certificate will relate to an MIS system rather than a centre.
- MIS identifiers will be managed by JCQ<sup>CIC</sup>. Agreed identifiers are listed in the A2C Best Practice document. Once MIS have confirmed their requirements the list will be incorporated in Appendix 2.
- Format of MIS provider identifier for certificate is 'jqc:mis:nnnn'.

### 8.4.2 Messaging Timeline

The MIS requests and responses will be asynchronous which means that a hosted MIS solution would send a request at time T1 with a list of centres. The awarding organisation prepares a response message with a list of centres having data to be delivered at time T2. Then MIS sends a polling message at time T3 and receives the response.

The centres that are queried are provided using their ebMS Party identifier, as specified in the *Protocol Profile* section of the A2C Transport Specification.

### 8.4.3 Service Level Recommendations

Upon receiving a message with a list of centres from the hosted MIS, the awarding organisation will create the response message as soon as possible, but no later than two hours after receiving the message from the hosted MIS.

## 8.5 Example Messages

### 8.5.1 Poll for Centres of Interest

This message consists of a list of the centre identifiers recognised by the awarding organisation. In this example 00003A is an AO Assigned Centre ID.

```
<CheckForMessages xmlns="http://jcq.org.uk/a2c/misquery">
  <CentreNumber>00001</CentreNumber>
  <CentreNumber>00002</CentreNumber>
  <CentreNumber>00003</CentreNumber>
  <CentreNumber>00003A</CentreNumber>
</CheckForMessages>
```

### 8.5.2 Awarding Organisation Response

This message consists of a list of centre identifiers recognised by the awarding organisation where data is currently awaiting collection. This follows the same format as the previous message:

```
<MessagesAvailable xmlns="http://jcq.org.uk/a2c/misquery">
  <CentreNumber>00001</CentreNumber>
  <CentreNumber>00003</CentreNumber>
</MessagesAvailable>
```

### 8.5.3 No messages available

If there are no messages available for any of the centres which were polled for, an empty element is returned:

```
<MessagesAvailable xmlns="http://jcq.org.uk/a2c/misquery"/>
```

## 9 Centre Switch Over to A2C

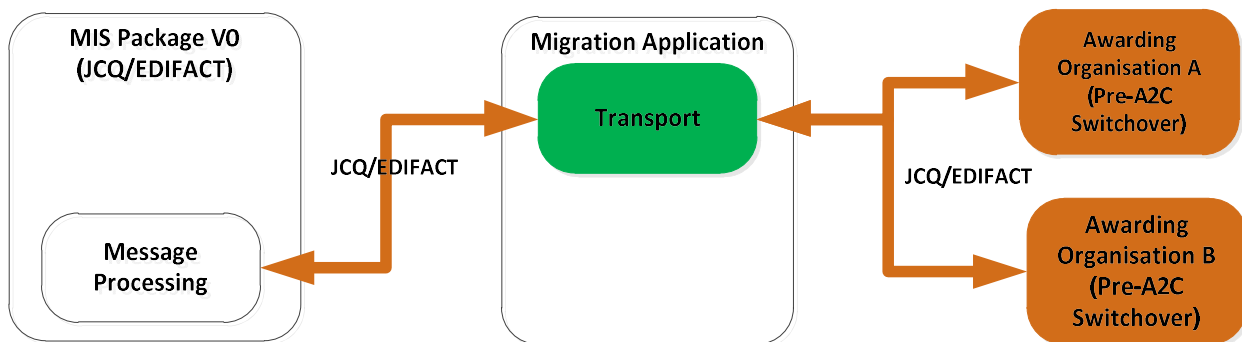
This document explains the processes to be followed for the transition to A2C. Refer to Section 2 *Centre Setup* for additional guidance. This section describes the transition period and the before and after states.

From the start of the transition period the transport component (at either the awarding organisation or in the MIS system) will support inbound and outbound transmission of the old JCQ-EDI/Pearson EDIFACT format files and the new A2C messages. This provides the most flexible way to transition from the old JCQ-EDI/ Pearson EDIFACT format files to the new A2C messages:

- Centres will be able to switch to A2C at the most appropriate time
- Supports the phased awarding organisation approach. Centres can switch to A2C for a particular awarding organisation, one awarding organisation at a time
- Facilitates the decommissioning of the migration application
- Minimises the impact on the awarding organisation by only receiving connections from one transport component
- Minimises administrative overhead of the centre by having only one transport component.

### 9.1 Pre-A2C Transition Period

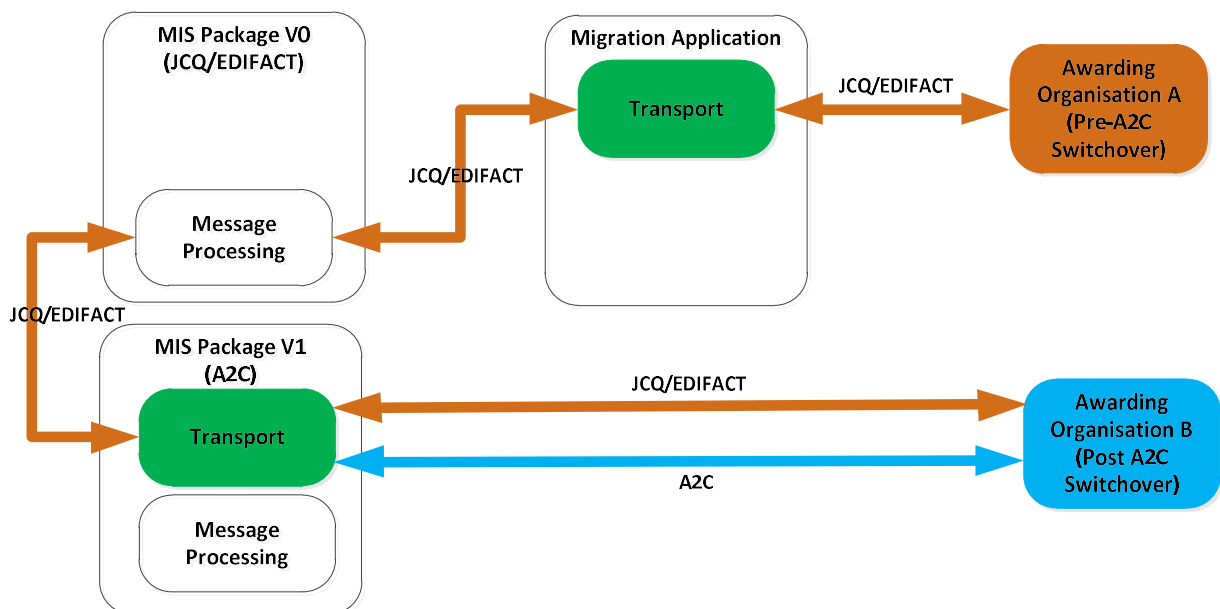
The following describes the state prior to the A2C messaging transition period and shows the Migration Application as the transport component supporting only the old JCQ-EDI/ Pearson EDIFACT format files. The 'MIS Package V0' label is simply to indicate the non-A2C transport version of the MIS system:



## 9.2 Post-A2C Transition Period

The following describes the state after the start of the A2C transition period:

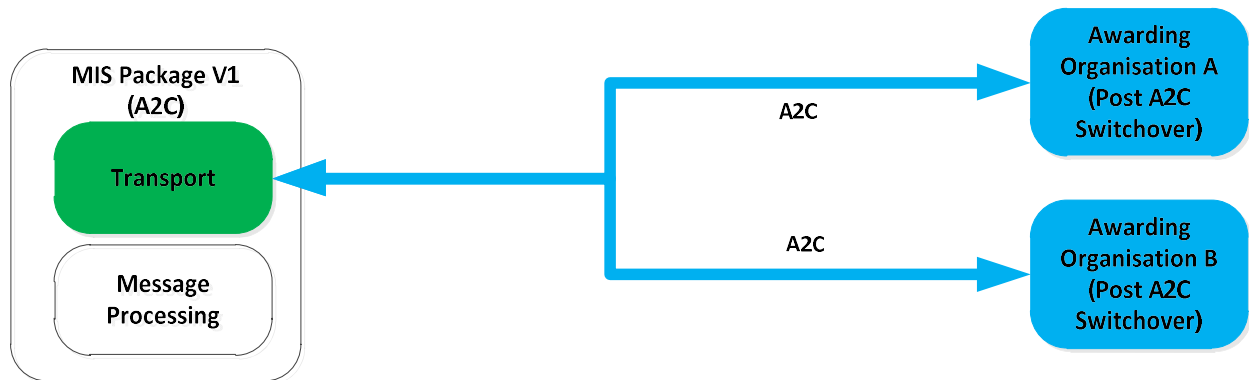
- The centre may decide to switch over to A2C in a phased approach, ie centres can switch to A2C for a particular awarding organisation one awarding organisation at a time. Before switching to A2C for a particular awarding organisation the centre will continue to use the Migration Application to send and receive the old JCQ-EDI/Pearson EDIFACT format files.
- When the centre chooses to switch to A2C for a particular awarding organisation both the A2C messages and the old JCQ-EDI/ Pearson EDIFACT format files will be sent and received through the MIS package alone (labelled 'MIS Package V1' to indicate the A2C transport compliant version of the MIS system).
- For orders placed with an awarding organisation before switching to A2C, all related transmissions including amendments and results will continue to be interchanged in the original format. This requires that the awarding organisation and MIS providers track and use the correct format for the original order.



Having switched over to A2C for all awarding organisations, the centre can decommission the Migration Application.

### 9.3 Final State: A2C Switch Over complete

This is the final state whereby the centre has switched over to A2C for all awarding organisations and all JCQ/Pearson EDIFACT orders have been completed, ie the centre has received results for the original order.



This state will have to be achieved by all centres before the end of the transition period.

## 10 Centre Set-up Notification and Request Product Catalogue

Centre Set-up Notification (CSN) is intended as a one-off message from the centre to the awarding organisation to let the awarding organisation know that the centre is now using an A2C compliant MIS and wishes to migrate to the A2C format for transactions with that awarding organisation.

Request Product Catalogue (RPC) is used by the centre to request a product catalogue from an awarding organisation. An RPC message can be sent by the centre any number of times, and each time the awarding organisation would return a product catalogue, featuring both static and instance data to the centre. The centre would normally use this transaction type in the following scenarios:

- Before actually creating the first order in A2C format, when the centre users want to get acquainted with the new MIS software by obtaining a product catalogue before sending a CSN.
- If the MIS system loses product data due to system failure or some other reason in future, the centre may need to obtain a product catalogue (static and instance) after restoring the system.
- The centre may wish to obtain a product catalogue when they change their MIS provider and a new software package is installed.
- The centre should send an RPC when a new version of the MIS software is installed. This is only necessary if the MIS upgrade includes an XSD change. This allows awarding organisations to track the XSD version in use and allows centres to obtain an up-to-date product catalogue (static and instance).

Refer to Section 2 *Centre Setup* for further details.

The action codes to be used for these transaction types are specified in Appendix 3.

### 10.1 Rules for use of Centre Set-up Notification and Request Product Catalogue

- The first message received by an awarding organisation from a centre must use either the CSN or the RPC transaction types.
- RPC is the only transaction type which may be used in advance of the CSN. A2C messages of any other type received before a CSN will be rejected.
- A CSN is intended to be sent by a centre to an awarding organisation only once.
- An RPC transaction type can be sent by a centre any number of times, before and after sending the CSN, and the awarding organisation will always return the up-to-date product catalogue, featuring the latest static and instance data.
- The first message received by the awarding organisation from the centre, whether an RPC or a CSN, will start the sequence number at 1 for the outbound messages from the centre to the awarding organisation. Any future A2C messages afterwards from the same centre to the same awarding organisation will carry on from the previous sequence number (see scenarios below).
- The awarding organisation will set the outbound initial sequence number from the awarding organisation to the centre upon receiving the first A2C message which can be either a CSN or an RPC. Any future A2C messages afterwards from the same awarding organisation to the same centre will carry on from the previous sequence number (see scenarios below).



## 10.2 Scenarios for Centre Set-up Notification and Request Product Catalogue

### 10.2.1 Scenario 1: Centre sends RPC before CSN

1. Centre wants to access product catalogue prior to becoming A2C compliant so that the users can get a view of the products in the new MIS software
2. Centre currently using legacy JCQ EDI/Pearson EDIFACT for transactions
3. Centre has the A2C compliant MIS software in their system
4. Centre sends an RPC message to the awarding organisation (Centre to AO sequence no. 1) at datetime1
5. Awarding organisation processes the request and sends feedback (AO to Centre sequence no. 1) at datetime2
6. Awarding organisation issues latest product catalogue – static and instance data combined (AO to Centre sequence no. 2) at datetime3
7. Centre processes the product catalogue and sends feedback (Centre to AO sequence no. 2) at datetime4
8. If necessary, the centre can send further RPC messages before submitting a CSN – see scenario 3.
9. When the centre is ready to commence A2C transactions, it sends a CSN to the awarding organisation (Centre to AO sequence no. 3, assuming that no other messages were interchanged after step 5) at datetime5
10. Awarding organisation processes the CSN message and sends feedback (AO to Centre sequence no. 3) at datetime6
11. Awarding organisation issues latest product catalogue at datetime7.

**Note:**

- a. The product catalogue sent at step 8 is a full product catalogue reflecting fully up-to-date static and instance data; this may be different from the product catalogue issued at datetime3
- b. Even if there are no changes to the product catalogue static or instance data since datetime3, a full product catalogue is issued at step 8 /datetime7. A centre can also send an RPC message to the awarding organisation at a later point – see scenario 2.

### 10.2.2 Scenario 2: Centre sends RPC after CSN

1. Centre is already A2C compliant:
  - a. In this case, the centre would have sent a CSN earlier
  - b. Centre and awarding organisation are continuing processing standard transactions using an A2C type of message (Assume the last sequence number for Centre to AO is 100 and AO to Centre is 70)
2. Centre MIS needs an up-to-date copy of a product catalogue eg centre has somehow lost previously-received product catalogue data
3. Centre sends an RPC message to the awarding organisation (Centre to AO sequence no. 101) at datetime1
4. Awarding organisation processes the request and sends feedback (AO to Centre sequence no. 71) at datetime2
5. Awarding organisation issues product catalogue (AO to Centre sequence no. 72) at datetime3
6. Centre accepts product catalogue and sends feedback (Centre to AO sequence no. 102) at datetime4.

**Note:**

After step 6, any further updates to the product catalogue will be sent as separate static and instance product catalogue messages, which will contain changes since datetime3.

### 10.2.3 Scenario 3: Centre sends an RPC more than once before CSN

This scenario is similar to scenario 1. The only difference is the sequence number used.

1. Centre sends an RPC to the awarding organisation (Centre to AO sequence no. 1) on datetime1
2. Awarding organisation processes the request and sends feedback (AO to Centre sequence no. 1) on datetime2
3. Awarding organisation issues latest product catalogue – static and instance data combined product catalogue (AO to Centre sequence no. 2) on datetime3
4. Centre accepts the product catalogue and returns feedback (Centre to AO sequence no. 2) on datetime4
5. Centre sends another RPC (Centre to AO sequence no. 3) on datetime5
6. Then steps 2 to 4 are repeated, ie:
  - a. Awarding organisation processes the request and sends feedback (AO to Centre sequence no. 3) on datetime6
  - b. Awarding organisation creates product catalogue and sends it to the centre (AO to Centre sequence no. 4) on datetime7
  - c. Centre accepts the product catalogue and returns feedback (Centre to AO sequence no. 4) on datetime8
7. Centre sends a CSN to the awarding organisation (Centre to AO sequence no. 5) on datetime9
8. Awarding organisation issues latest product catalogue (AO to Centre sequence no. 5) on datetime10.

This demonstrates that the sequencing starts at the first A2C message from the centre to the awarding organisation and it carries on. A CSN message does not normally reset the sequence number to 1 (see scenario 5 for the exception).

Although RPC messages can be sent multiple times, a second CSN message would normally trigger error feedback; message '0014 Centre set-up transaction cannot be submitted as this process has already been completed successfully' would be sent. Awarding organisations might also contact the centre in these circumstances as this scenario would raise concerns that the centre did not understand the A2C process.

### 10.2.4 Scenario 4: Centre changes MIS software

As specified in the Sequencing section, the new MIS application should be configured to commence the sequence numbers from where it was left in the old MIS application for each awarding organisation with which the centre is dealing.

The centre is not required to send a Centre Set-up Notification when it changes MIS. The MIS application details are always notified to the awarding organisation in the message header of each message. The specific approach adopted regarding change of MIS supplier should be agreed between the new MIS supplier and the centre; the feasibility of maintaining product catalogue data and sequencing from the previous installation will influence the approach and it may be appropriate to contact the awarding organisation before proceeding. The following scenarios might be considered:

1. Where the last sequence number is known and the previous product catalogue information has been maintained A2C data exchange may continue without sending an RPC or CSN message.

2. Where the last sequence number is known and the previous product catalogue information has been lost or discarded, A2C data exchange may continue without sending a CSN message, but an RPC message would be required in order to replace product catalogue data.
3. Where the last sequence number is not known and the previous product catalogue information has been maintained, a CSN message would be required in order to restart sequencing. MIS should be aware that this would automatically make a fully up-to-date product catalogue available to the centre. This would include static and instance data for all currently available products.
4. Where the last sequence number is not known and the previous product catalogue information has been lost or discarded, a CSN message would be required in order to restart sequencing. This would automatically make a fully up-to date product catalogue available to the centre. This would include static and instance data for all currently available products.

#### 10.2.5 Scenario 5: MIS has lost track of sequencing due to unforeseen problem

In exceptional cases, where it is has not been possible to maintain sequencing, an offline discussion to agree how the reset of sequence numbers will be handled is essential. It may require manual intervention to systems. The centre must:

1. First contact the awarding organisation to advise of the problem and obtain agreement to change the sequencing.
  - a. It is necessary to agree a suitable number from which to re-start. It must be a bigger number which is not used so far, and ideally leave a clear gap to give visibility of the reset. For example, if the awarding organisation's record says sequence number 1000 has already been used for the centre-awarding organisation messages, the sequence would be reset starting from 1100 or similar.
  - b. Depending on the circumstances it may be necessary to reset the sequence numbers in both directions – this would also need to be considered and agreed before further messages are sent.
  - c. It may also be necessary to consider the impact on queued messages from the awarding organisation not yet collected by the centre.
2. Agree with the awarding organisation any further steps needed to implement the sequence change, and to address any related issues the centre may be experiencing, such as loss of product catalogue data. Such steps may vary dependent on the exact situation and technical aspects of both the centre's MIS system and awarding organisation's systems.

#### 10.2.6 Scenario 6: Centre upgrades MIS software and XSD version

This scenario is similar to Scenario 4. The only difference is that the MIS software upgrade need not involve a change of supplier, but does involve a change of XSD version.

As specified in the Sequencing section, the upgraded MIS application should be configured to commence the sequence numbers from where it was left in the previous version for each awarding organisation with which the centre is dealing.

1. Centre wants to advise the awarding organisation that they have upgraded to the latest version of XSD.
2. Centre sends an RPC using their upgraded MIS system. This will ensure that the MIS application details and associated XSD version are notified to the awarding organisation. This information is included in the message header.
3. Awarding organisation processes the request and sends feedback using the latest version of the XSD as indicated in the message header of the RPC message.
4. Awarding organisation issues latest product catalogue – static and instance data. This and all subsequent messages will use the latest version of the XSD.

5. The centre may, or may not, choose to load the product catalogue data depending on the approach taken for the MIS upgrade. The specific approach regarding upgrade of MIS software should be agreed between the MIS supplier and the centre and will depend on the impact of XSD changes on product catalogue data. It may be feasible to maintain product catalogue data from the pre-upgrade installation, or the MIS supplier may advise that the product catalogue message returned at 4. above should be loaded to the centre's database. (This is similar to the situation described in 2, under Scenario 4.)
6. If the product catalogue message is not loaded to the centre MIS, the only purpose of the RPC will be to communicate the XSD upgrade to the awarding organisation.

## 11 XML Schema Design

### 11.1 XSD Overview

The XSD used within the JCQ<sup>CIC</sup> A2C project uses a sector wide generic schema managed by the [Education, Skills and Children's Service Information Standards Board \(ESCS ISB\)](#)

The schema is intended to provide a single generic schema for the sector allowing suppliers of MIS systems to use one schema for all data exchange projects rather than having many solution specific data exchange schemas.

The ISB schema covers many parts of the data in the sector that the JCQ<sup>CIC</sup> A2C project does not require and therefore a cut down version of the overall schema has been created for the A2C project. This aids understanding and implementation of the new concept as the A2C project only requires 23 out of the 60 data blocks currently available. Although the full schema could have been used it was considered that the extra data in the schema not required for this project would have led to confusion at this early stage of adoption.

As suppliers start to support both JCQ<sup>CIC</sup> A2C and other sector projects using the ISB schema it is expected that at some time in the future just the one schema will be used.

The JCQ<sup>CIC</sup> schema only varies from the full ISB schema in that:

- It only uses the specific data blocks required for the JCQ<sup>CIC</sup> A2C project
- The root segment name has been amended to reflect the JCQ<sup>CIC</sup> A2C project name
- The namespace is specific to JCQ<sup>CIC</sup>.

Otherwise, the base ISB schema and the JCQ<sup>CIC</sup> version are identical in design and behaviour.

The schema uses a Data Block concept rather than inline process specific schemas. The full details of the schema design and behaviour are explained in the document [ISB Data Block Message Design](#).

### 11.2 Managing XSD upgrades

Schema changes during the A2C rollout will be avoided if possible. Any schema change requires approval by the A2C Change Control Board.

Awarding organisations will support the current and previous specification. As the Spec will be issued each year, this means that awarding organisations will always support at least two full academic years.

If there are XSD changes:

- Awarding organisations will continue to support XSD1 alongside XSD2 for at least one year.
- In the second year of XSD2 or if an XSD3 is implemented in the subsequent year, XSD1 will no longer be required to be supported.
- MIS are not required to support multiple spec versions but must ensure that all of their centres are operating on the current or immediately preceding spec version.

### 11.3 Element optionality

The schema specifies all identifier elements as mandatory, but all other elements as optional. Due to the nature of the XSD design, the same data block can be used for different business processes and each business process will only require the use of certain of the elements in the data block.

Appendix 1 *Business Data Architecture Detail* lists the JCQ<sup>CIC</sup> Data Architecture Data Model attributes defined for the project. Each data model attribute to be exchanged has a corresponding element in the XSD.

## 11.4 TransactionName

The message header has an optional element called <TransactionName>. This is held under the <MsgHeader>TransactionInfo node. The use of this is explained in the *ISB Data Block Message Design* document, but for JCQ<sup>CIC</sup> this element must be completed as per Appendix 3 *TBDUM and Service Codes (Action Codes)*.

## 11.5 Feedback Messages

Data messages can be considered as providing data at various levels and these levels need to be identified so that feedback can be constructed accordingly:

- Record: A record is defined as one row of data in the primary data block. A row in a data block is identified using the identifiers defined in the schema. For example, in the case of a Named Order, if the message contains 100 bookings, the QE Learner Booking data block will have 100 rows uniquely identified by the QELearnerBooking\_Id.
- Transaction: A record will have referenced data in other data blocks. One row in the primary data block may have reference to multiple rows in other data blocks; hence a record by itself is not complete without the referenced data within other data blocks. A single transaction consists of a single record and its associated data in other data blocks.
  - In a named order consisting of 100 orders, there will be essentially 100 different transactions.
- Transaction Type: Transaction types are defined in the transaction to data block matrix by a project. In the case of the A2C project this information is contained within the TDBUM matrix. Each transaction refers to one business process. One message cannot contain data relating to more than one transaction type.
- Batch of Transactions: One A2C message contains a set of transactions relating to one transaction type. The type of transaction within the batch is specified in the TransactionName attribute in the message header.
- Transaction and Record feedback depends upon whether the message is a Transaction based message or Data Driven message. A2C only uses Transaction based messages and therefore will never have Record Level feedback.

Whenever a Transaction, Record or Attribute level feedback is supplied, the entity in the data block that the error is within is identified by the FB\_EntityName element. This supports data blocks that have more than one entity within them as defined by the Data Model.

The structure of feedback messages is defined in the ISB Data Block Message Design Technical Data Standard available from the ISB website at <http://data.gov.uk/education-standards/guidance/isb-data-block-message-design-technical-data-standard>. Version 6 dated 27/3/15.

The table provides details of the elements within the Feedback Message structure including maximum length for string fields allowed in A2C.

Element	Maximum string length	ISB Description	Additional Notes
Ref_Message_Id	250	Identifies the original message Id being referred to in a feedback message. Optional as the issue may be that there was no message Id in the original message	
FB_FeedbackCode	4	The Message, Transaction or Record Level feedback error code that identifies the reason for the feedback	Feedback Codes are defined in Appendix 7 – Feedback Messages Detail
FB_FeedbackSeverity	12	The Message, Transaction or Record Level error severity	Valid values for FB_FeedbackSeverity are defined in Section 12 Feedback Messages Section 3 Definitions.
FB_FeedbackText	250	The Message, Transaction or Record Level error message	
FB_InfoName	250	The text of any additional information example is "last received sequence Number = "	
FB_InfoValue	250	The data value(s) related to the FB_InfoName - example is "12345" (being the last received sequence number as per the FB_InfoName text	
FB_Data BlockName	250	Name of the Data Block as specified within the XSD, ie Party_DS – the name of that Data Block node	
FB_EntityName	250	Name of the data architecture entity within the Data Block that the attribute (FB_DB_ID_AttributeName) in the feedback is contained within	As in Appendix 1 ie with spaces
FB_IdName	250	The name of the data architecture attribute that is part of the identifiers of the Data Block	As in Appendix 1 ie with underscores
FB_IdValue	250	The value of the content of the FB_DB_ID_AttributeName that was submitted in the original message that this feedback refers to	
FB_AL_AttributeName	250	The name of the data architecture attribute that is being fed back	As in Appendix 1 ie with underscores
FB_AL_AttributeValue	250	The original content of the FB_AL_AttributeName that was submitted in the original message is being fed back	

Table 2 Feedback Message Elements

## 12 XML Schema Usage

XML messages are used within the data exchanges between centres and awarding organisations. This is to provide the underlying mechanism to ensure that the required business processes can exchange data in an efficient and consistent manner. The same XML schema also supports feedback messages.

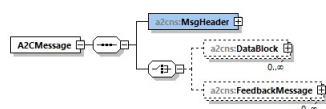
The schema has three main divisions at the top level of the structure.

1. MsgHeader containing the message header
2. Datablock
3. FeedbackMessage containing the feedback information.

Message header contains information about the message itself and should be present in all messages. A message containing business data will contain the message header as well as one or more data blocks as specified in Appendix 3. This message will not contain the feedback message section.

A feedback message will contain the message header and the feedback message section and will not contain the data block section.

Below is a snapshot of the message schema:



**Note:** Each data block may contain one or more data entities. This is specified in Appendix 1.

### 12.1 Usage Rules

- R1. All A2C messages follow a single schema used to validate the message irrespective of transaction type.
- R2. One A2C message will contain data relating to one transaction type only. For example, a single message cannot contain Named Order and Amend Learner Details. However one message can contain more than one transaction of the same transaction type.
- R3. A single message must contain all of the mandatory data blocks for the transaction type identified in the TransactionName attribute in the message header as defined in the TDBUM (Appendix 3), eg A Named Order must contain 'QE Learner Booking' and 'QE Booking' data blocks. The message may also contain one or more optional data blocks.
- R4. A message must only contain data blocks which are related to the transaction type identified in the TransactionName attribute in the message header as specified in the TDBUM (Appendix 3).
- R5. A transaction of any transaction type must not span across multiple messages. Data sufficient to complete a transaction must be contained within a single message.  
Examples:
  - a. A transaction of type Named Order can contain 11 data blocks. For one Named Order, the centre cannot send four of the data blocks in one message and the rest of the data blocks in a subsequent message.
  - b. Multiple transactions of transaction type Named Orders can be sent in one message.
  - c. If the centre has 100 orders to send it may send 20 orders in one message and 80 in a subsequent message.



R6. Each feedback message must relate to only one received message. That is, one feedback message cannot relate to data received in more than one original message. (For detailed behaviours related to feedbacks, please refer to Section 12 *Feedback Messages*.)

R7. In the source system, data blocks in a message must be constructed in the correct order as specified in the schema.

R8. If an invalid feedback message is received by the original sending system (eg feedback message contains erroneous feedback codes) then a manual exception process must be initiated at the centre or awarding organisation.

R9. While constructing the XML message, the source system must trim all leading and trailing spaces in the element values; these include white spaces and carriage returns. This means that an element value can not consist solely of white space. The receiving system does not need to perform any further trimming.

R9.1 If an optional tag is omitted the value is assumed to be null.

Nulls can also be explicitly specified by using an empty tag, such as:

```
<QE_Website_URL_Locator_Id></QE_Website_URL_Locator_Id>
```

Or a self-closing tag, such as:

```
<QE_Website_URL_Locator_Id/>
```

However white space is not acceptable. The following (which has a space between the opening and closing tags) is invalid:

```
<QE_Website_URL_Locator_Id> </QE_Website_URL_Locator_Id>
```

R9.2 Nullifying previously submitted values will only be permitted where specifically stated against the attribute in Appendix 1. Currently, values for Relationship\_Reference associated with Party\_RR\_Reference\_Types of:

- Driving Licence Number
- National Identity Number
- National Insurance Number
- Passport Number
- ULN

may be nullified using the Amend Learner Details transaction type. This is allowed for these values because it may not be possible to provide the corrected value at the time the error is identified, due to delays in receiving these from external parties such as LRS and HMRC.

Blanking out of previously communicated values is permitted in the Results transaction type. It is accepted that it may not be possible to provide updated values immediately. This will apply where a result, including the outcome or any associated value, needs to be updated.

Incorrect values will normally be updated to correct values using the appropriate transaction type, but where nullification is acceptable this should be done by sending any one of the following for the affected attribute:

- a self-closing tag
- an empty tag

Note however, that the real requirement is a null in eg ULN, so that must be generated by the receiving program.

R10. A message must not contain multiple data blocks of the same kind. For example, if a Named Order has to specify multiple parties in a message, the following will apply.

R11. The following use is **not** valid. Here the data block Party\_DS is repeated in a single message):

```
<DataBlock>
  <DataBlockName>Party_DS</DataBlockName>
  <Party_DS>
    <Party>
      <Party_ID>
        <Party_Id>10</:Party_Id>
      </Party_ID>
      <Party_Type>Organisation</:Party_Type>
    </:Party>
  </:Party_DS>
</:DataBlock>

<:DataBlock>
  <:DataBlockName>Party_DS</:DataBlockName>
  <:Party_DS>
    <:Party>
      <:Party_ID>
        <:Party_Id>12345</:Party_Id>
      </:Party_ID>
      <:Party_Type>Organisation</:Party_Type>
    </:Party>
  </:Party_DS>
</:DataBlock>
```

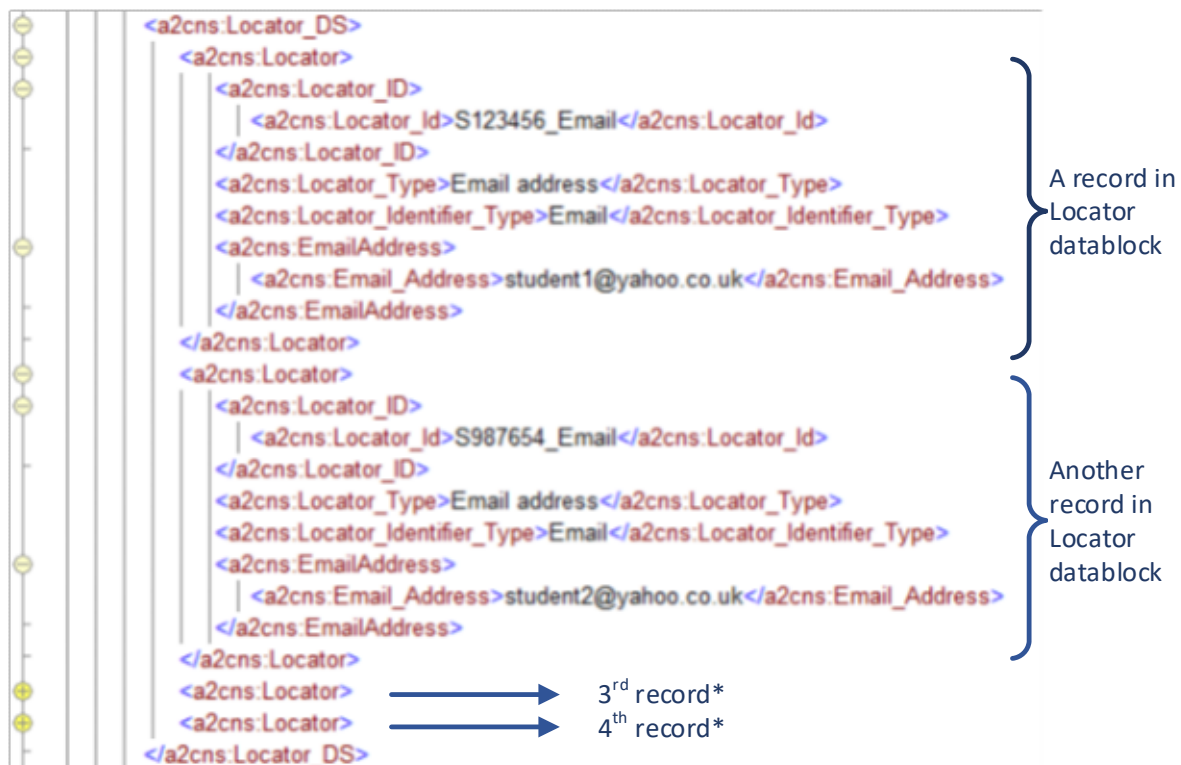
The following use is valid (The data block is not repeated, but more than one parties are specified):

```
<:DataBlock>
  <:DataBlockName>Party_DS</:DataBlockName>
  <:Party_DS>
    <:Party>
      <:Party_ID>
        <:Party_Id>10</:Party_Id>
      </:Party_ID>
      <:Party_Type>Organisation</:Party_Type>
    </:Party>
    <:Party>
      <:Party_ID>
        <:Party_Id>12345</:Party_Id>
      </:Party_ID>
      <:Party_Type>Organisation</:Party_Type>
    </:Party>
  </:Party_DS>
</:DataBlock>
```

## 12.2 Definitions for data and feedback messages

Data messages can be considered as providing data at various levels and these levels need to be identified so that feedback can be constructed accordingly. The following definitions have some associated illustrations taken out of an early Named Order example. For the most up-to-date XML examples please see the [XSD and XML Examples](#) folder on SharePoint.<sup>1</sup>

- Record: A record is defined as one row of data in an entity within a data block. A row in a data block can be uniquely identified using the identifiers defined in the schema.
- Each entity within a data block will have one or more records. The following image shows the Locator data block containing four records:



XML collapsed to save space in the snapshot

- The Named Order message has got four bookings, and hence the QE Learner Booking entity within the primary data block (QE Learner Booking) has four records.
- The relationship between records in different data blocks is defined in Appendix 5.
- Transaction: A record within an entity in the primary data block will have referenced data in other data blocks. A single transaction consists of a single record in the primary data block and its associated data in entities within all other data blocks. For example in the case of Named Order, one record in the primary data block does not alone represent one Named Order, rather the transaction includes data related to the learner, the centre to which the learner belongs to and the qualification to which the order relates.

<sup>1</sup> Note that the XML examples on this and subsequent pages are based on an out-of-date version of the schema (v5.0) however the explanations relating to records, related records etc are still valid. Note also that although all parties must accept incoming messages with the A2CNS namespace prefix against every element, it is recommended that this is not included in outgoing messages; the XSD contains targetNamespace and implicit element qualification is enabled in the XSD.

- The snapshot below illustrates a single transaction. (Data extracted from the above attached message.) This demonstrates a Named Order from the centre 12345 to the AO 10, involving learner S676545 for the award JJ123/1.





- All the above constitute a transaction.

- **Transaction Type:** Transaction type refers to the type of request and/or information a centre would want to provide to an awarding organisation or vice versa. Examples include a Named Order, Centre Assessed Outcomes, Results and Product Catalogue. All the transaction types are defined in the TDBUM matrix in Appendix 3.  
**Message:** A message contains a set of data pertaining to one transaction type. One message may contain one or more transactions.