

UIDAI

Unique Identification Authority of India
Planning Commission, Govt. of India (GoI),
3rd Floor, Tower II,
Jeevan Bharati Building,
Connaught Circus,
New Delhi 110001



REGISTRAR INTEGRATION MANUAL PRE-ENROLMENT DATA, SECURITY & KYR+ DATA VERSION 1.1

Table of Contents

1. INTRODUCTION.....	3
1.1 TERMINOLOGY	4
1.2 OBJECTIVE OF THIS DOCUMENT	4
2. INTEGRATION PROCESS	5
2.1 PRE-ENROLMENT DATA INTEGRATION	5
2.2 REGISTRAR SECURITY INTEGRATION	8
2.3 KYR+ DATA INTEGRATION	9
3. DATA TYPE AND DATA FORMAT.....	12
4. ENROLMENT CLIENT XSDS	14
4.1 DEMOGRAPHIC	14
4.2 BIOMETRICS	16
4.3 MANIFEST	17
5. APPENDIX	20

1. Introduction

UIDAI was created with the responsibility to plan and implement a Unique Identification scheme, own and operate the CIDR and be responsible for updates and maintenance on an ongoing basis.

Since the capture of biometrics using sophisticated biometric devices by highly trained professionals could be the bottle-neck in the entire enrolment processes, it is desirable to complete the other demographic data capture and verification ahead of the biometric capture. This step is called pre-enrolment. It is conceivable that through existing databases available to the Registrars (For example, state governments BPL, PDS, or NREGA databases) a pre-enrolment workflow can import data into the AADHAAR Enrolment Client module, and only call upon residents to capture biometric data so that it can be linked to the existing demographic record for that resident. Note that, even with pre-enrolment, final verification is done at the time of enrolment during which biometric data will also be captured.

Pre-enrolment data should be in a CSV file format as prescribed in this document. AADHAAR Enrolment Client software collects basic AADHAAR fields which are known as KYR (Know Your Resident) fields where as Registrar collects additional fields known as KYR+. AADHAAR Enrolment Client software provides a loosely coupled way to integrate KYR data into Registrar's software.

At the end of every enrolment, Enrolment Client software writes KYR fields into a pre-configured directory as per software manual as a name- value pair file which Registrar's software can load and continue to collect additional KYR+ fields. This document also lists the name- value pair file format required for the Registrars for launching their KYR+ application. The allowed data format and the allowed character length have been covered in the subsequent sections of this document.

The document also covers the data format and the data length (maximum) supported with various labels. The registrars are expected to input the data in the suggested data format and length.

1.1 Terminology

Enrolment: is the process of capturing resident data (including demographic and biometric data). The enrolment is done by the enrolment operator/agent.

Enrolment centre: is the location where the enrolment happens. Each enrolment station has the required enrolment set-up to make the enrolment possible.

Enrolment station: is the system which does the enrolment capture. The enrolment set-up includes a computer, the biometric devices and some accessories.

Resident: is a person who undergoes the enrolment capture process and gets a UID from the government.

AADHAAR Number: is a 12-digit number issued by the government as proof of identity and residence in India.

Central ID Data Repository (CIDR): a repository regulated and managed by the UIDAI. It issues AADHAAR numbers, updates resident information and authenticates the identity of residents as required.

1.2 Objective of this document

The objective of this integration manual is to educate the intended audience, the Registrar to integrate the pre-enrolment data with the AADHAAR Enrolment Client application as well as to integrate the KYR+ related fields to KYR+ application.

The objective of this document is not to cover the installation and configuration of the AADHAAR Enrolment Client. For the intended users of this document, the AADHAAR Enrolment Client application should be seamless and hence the user didn't bother about the internal demarcations of the system.

2. Integration Process

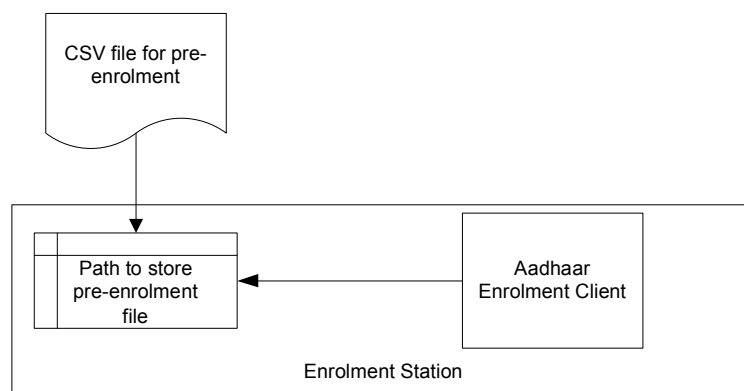
This section covers how to integrate the pre-enrolment data to the AADHAAR Enrolment Client. Also, this document well illustrates the integration of the KYR+ data fields required for the registrar’s application from the Enrolment Client.

The subsequent sections cover the file templates required for the Registrar as well the KYR+ application. The fields/labels listed in the table shall not necessary feature in the same order as mentioned in the file format. The file shall contain all the labels as defined and no column with that label should be missing from the file. The Registrar can pass on the other fields, the details of which are not a part of the template file. The Enrolment Client application is intelligent enough to pick the labels required for running the application. The fields defined as part of pre-enrolment file is adhere to the DDVSP committee report circulated by UIDAI.

2.1 Pre-Enrolment Data Integration

Since the capture of biometrics using sophisticated biometric devices by highly trained professionals could be the bottle-neck in the entire enrolment processes, it is desirable to complete the other demographic data capture and verification ahead of the biometric capture. This step is called pre-enrolment.

In the case where Registrar has a good database, it can be used to pre-populate the AADHAAR Enrolment Client. This data is required to reduce the effort and time of the enrolment operators during enrolment capture process.



The typical process to integrate the pre-enrolment data file with the AADHAAR Enrolment Client application is as follows:

- ❖ Prepare a CSV (Comma separated Value) file for the pre-enrolment data in the template suggested in the subsequent section
- ❖ Ensure that the CSV file is stored in the pre-defined location as suggested in the installation guide in the enrolment station

The CSV file will contain the headers and all these headers should feature in the same CSV file. The registrar should make sure that the labels are not missing from the CSV file though the field level value may or may not be available for that label.

The table has been broken down in two fields, mandatory and optional fields.

Mandatory	PreEnrolmentID
	LocalLanguageCode
Optional	FullName
	FullName_LL
	DateOfBirth
	Gender
	AddrCareOf
	AddrCareOf_LL
	AddrBuilding
	AddrBuilding_LL
	AddrStreet
	AddrStreet_LL
	AddrLandmark
	AddrLandmark_LL
	AddrLocality
	AddrLocality_LL
	AddrVTC
	AddrVTC_LL
	AddrDistrict
	AddrDistrict_LL
	AddrState
	AddrState_LL
	PinCode
	RelativeName
	RelativeName_LL
RelativeUID	
RelationType	
Mobile	
Email	

The labels which shall be part of the pre-enrolment CSV file are as explained as below:

- **PreEnrolmentID**- This is the document ID for the document which the resident has. For ex. ration card number, Passport number etc.This is a mandatory field.
- **LocalLanguageCode**- This is the local language code for all the Indian state official language. The registrar may select the appropriate language code in

accordance to his data and should ensure that the details to be mentioned in all the labels suffixed with 'LL' should be in the same local language. This is a mandatory field. For ex. 06-Hindi. The local language code shall be followed as per MDDS: 01 Version: 1.0 specification for the languages as per the ISO 639-3.¹

- **FullName**- This is the Name of the resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **FullName_LL**- This is the Name of the resident in the local language as per the local language code mentioned.
- **DateOfBirth**- This is the Date of Birth of the resident. This shall adhere to the data type format and length as described in section 3 of this document. This should be in DDMMYYYY format.
- **Gender**- This is the Gender of the resident. This shall adhere to the data type format and length as described in section 3 of this document. For ex. M-Male, F-Female, T-Transgender.
- **AddrCareOf**- This is the 'Care of' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrCareOf_LL**- This is the 'Care of' detail in the local language as per the local language code mentioned.
- **AddrBuilding**- This is the 'building' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrBuilding_LL**- This is the 'building' detail in the local language as per the local language code mentioned.
- **AddrStreet**- This is the 'Street' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrStreet_LL**- This is the 'Street' detail in the local language as per the local language code mentioned.
- **AddrLandmark**- This is the 'landmark' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrLandmark_LL**- This is the 'landmark' detail in the local language as per the local language code mentioned.
- **AddrLocality**- This is the 'locality' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrLocality_LL**- This is the 'locality' detail in the local language as per the local language code mentioned.
- **AddrVTC**- This is the 'Village/Town/City' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrVTC_LL**- This is the 'Village/Town/City' detail in the local language as per the local language code mentioned.
- **AddrDistrict**- This is the 'District' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrDistrict_LL**- This is the 'District' detail in the local language as per the local language code mentioned.

¹ Refer document Metadata and Data Standards for Person Identification and Land Region Codification MDDS: 01 Version: 1.0 December 24, 2009

- **AddrState**- This is the 'State' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrState_LL**- This is the 'State' detail in the local language as per the local language code mentioned.
- **PinCode**-this is the Pincode details of the address of the resident.
- **RelativeName**- This is the 'Relative Name' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **RelativeName_LL**- This is the 'Relative Name 'detail in the local language as per the local language code mentioned.
- **RelativeUID**- This is the 'Relative UID or Enrolment ID' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **RelationType**- This is the 'Relative Name detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
 - F-Father
 - M-Mother
 - H-Husband
 - W-Wife
 - G-Guardian
- **Mobile**- This is the Mobile number of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **Email**- This is the Email address of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.

2.2 Registrar Security Integration

The AADHAAR Enrolment Client application captures the demographic and biometric data of the residents. All the data collected by UIDAI will be encrypted with a 1024 bit public key. In order for the registrars to get access to the enrolment packets specific to registrars, it's essential for them to provide their public key to the UIDAI. The public key file should be digitally signed by the CA for validation. UIDAI would integrate the given public key with the enrolment client. A release would be made with the available key to the respective registrars.

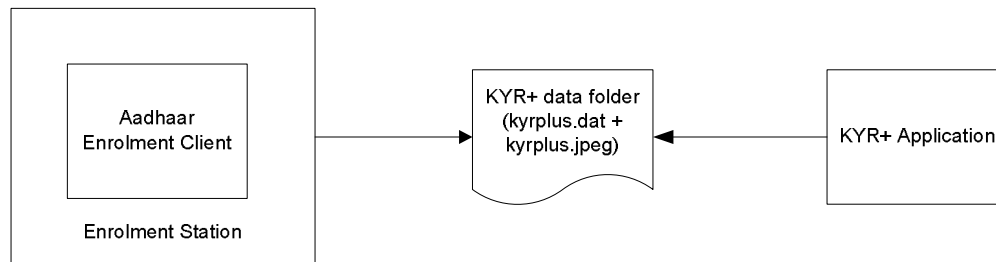
UIDAI recommends using multiple public keys to strengthen the security of the resident data. The public keys should be provided by the registrar before the registrar on-boarding process. The public keys will be integrated into the enrolment client. If a public key is not provided by the registrar, the registrar packets would not be created. The accepted **Public Key Format** for the AADHAAR enrolment client application is X509.

The Registrar code '000' to '010' are reserved for internal/training use. Hence, it is advised to not to use these codes as their registrar code.

2.3 KYR+ Data Integration

The AADHAAR Enrolment Client application captures the KYR (Know Your Resident) data. The registrars may require capturing some other registrar specific fields related to residents called as KYR+ data. For example, In case of PDS data, information such as APL (Above poverty line), BPL (Below poverty line), Family Details, etc. may be collected as part of KYR+ data.

Following diagram depicts the relation between AADHAAR Enrolment Client and Registrar's KYR+ Application their integration.



The typical process to integrate the pre-enrolment data file with the AADHAAR Enrolment Client application is as follows:

- ❖ Resident comes to enrol
- ❖ The enrolment operator finishes the KYR data capture using AADHAAR Enrolment Client of the resident
- ❖ The KYR+ data as per the template gets saved into a .dat file (name-value pair) at a pre-defined location on the enrolment station
- ❖ The Resident's photograph gets saved into a .jpeg file) at a pre-defined location on the enrolment station
- ❖ The operator switches to the KYR+ application with the help of Alt+Tab key
- ❖ The KYR+ data of the resident can be pulled to the KYR+ application with the help of a function key for e.g. F5.
- ❖ KYR+ data is captured and saved by the KYR+ application

The KYR+ data gets captured from the AADHAAR client application after each enrolment capture and gets saved into a pre-defined location. At the end of every enrolment, a unique "Enrolment ID" is generated. This edited KYR data along with Enrolment ID is shared using a Name-Value pair file which will typically be a .dat file (kyrplus.dat) to Registrar's software for collecting KYR+ fields.

The KYR+ application requires almost the same set of data what AADHAAR system collects from the pre-enrolment data from registrar. The only extra column added is that of the Enrolment ID which is generated as a part of the enrolment from the application itself. The "Name" and the "Value" field will be separator with delimiter '=' (equal to) and name will carry the labels as mentioned below. For every new name-value pair, 'new Line' will be the separator, ex.

FullName=Ram Prakash

DateOfBirth=10/04/1980

Mandatory	PreEnrolmentID
	LocalLanguageCode
Optional	EnrolmentClientVersion
	DocumentSpecificationVersion
	RegistrarCode
	EnrolmentAgencyCode
	StationID
	EnrolmentID
	ApplicationNumber
	NPR Receipt Number
	Bank Account Number
	FullName
	FullName_LL
	DateOfBirth
	DateOfBirthType
	Gender
	AddrCareOf
	AddrCareOf_LL
	AddrBuilding
	AddrBuilding_LL
	AddrStreet
	AddrStreet_LL
	AddrLandmark
	AddrLandmark_LL
	AddrLocality
	AddrLocality_LL
	AddrVTC
	AddrVTC_LL
	AddrDistrict
	AddrDistrict_LL
	AddrState
	AddrState_LL
	PinCode
	RelativeName
	RelativeName_LL
	RelativeUID
RelationType	
Mobile	
Email	

Note: The fields have been broken down to accommodate in this document.

The labels mentioned above are generated out of the AADHAAR Enrolment Client application. The description for all these labels are same as discussed in the Pre-enrolment integration section. The labels suffixed with 'LL' are the respective details of the resident in the local language in which the AADHAAR Enrolment Client is configured for.

The **EnrolmentClientVersion** is the AADHAAR client application version number.

The **DocumentSpecificationVersion** is the registrar integration manual document version number.

The **RegistrarCode** is the 3 digit code of the Registrar.

The **EnrolmentAgencyCode** is the 4 digit code for the enrolment agency.

The **StationID** is the 5 digit code of the enrolment station.

The **EnrolmentID** label is the enrolment Id of the resident. This Id was generated as a part of enrolment capture from the AADHAAR Enrolment Client application. This shall adhere to the data type format and length as described in section 3 of this document.

The **NPR Receipt Number** label is the National Population Register acknowledgement number of the resident. This shall adhere to the data type format and length as described in section 3 of this document.

The **Bank Account Number** is the bank account number of the resident. This should be a valid account number. This shall adhere to the data type format and length as described in section 3 of this document.

In case when the resident provides the age in years, the year of birth is calculated and passed on to the registrar specific application with the **Date of Birth** name value pair.

3. Data Type and Data Format

The data type and the data length for the various labels mentioned in the file format as mentioned in the previous sections are described below. The registrars are expected to follow the norms and provide the input file in the suggested data formats only.

Labels	Data Type	Data Length(max)
PreEnrolmentID	alpha-numeric (a-z, A-Z, 0-9) all special characters allowed	40
EnrolmentID	alpha-numeric (a-z, A-Z, 0-9, space,colon,hyphen,Front slash) <ul style="list-style-type: none"> • Registrar Code- (a-z, A-Z) • Enrolment Agency code- (a-z, A-Z) • Station ID- (0-9) • Sequence Number(system generated 5 digit number) • Date Time Stamp-(0-9,hyphen,front slash) in DDMMYYYYHHMMSS format 	40 [3] [4] [5] [16]
LocalLanguageCode	numeric (0-9)	2
ApplicationNumber	alpha-numeric (a-z, A-Z, 0-9, space, hyphen)	40
NPR Receipt Number	numeric (0-9)	99
Bank Account Number	alpha-numeric (a-z, A-Z, 0-9, space,hyphen,Front slash)	99
FullName	alphabetic (a-z, A-Z, space, dot)	99
FullName_LL	Indian language value of "Full Name"	99
DateOfBirth²	numeric (0-9)	8
DateOfBirthType³	alphabetic (a-z, A-Z)	15
Gender	char[M,F,T]	1
AddrCareOf	alpha-numeric (a-z, A-Z, 0-9, space, #,,comma,Dot,Front slash, hyphen)	60
AddrCareOf_LL	Indian language value of "AddrCareOf"	60
AddrBuilding	alpha-numeric (a-z, A-Z, 0-9, space, #,,comma,Dot,Front slash, hyphen)	60
AddrBuilding_LL	Indian language value of "AddrBuilding"	60
AddrStreet	alpha-numeric (a-z, A-Z, 0-9, space, #,,comma,Dot,Front slash, hyphen)	60
AddrStreet_LL	Indian language value of "AddrStreet"	60
AddrLandmark	alpha-numeric (a-z, A-Z, 0-9, space,	60

² Date of Birth must be in DDMMYYYY format

³ DOB type can be Approximate, Declared or Verified

	#,comma,Dot,Front slash, hyphen)	
AddrLandmark_LL	Indian language value of "AddrLandmark"	60
AddrLocality	alpha-numeric (a-z, A-Z, 0-9, space, #,,comma,Dot,Front slash, hyphen)	60
AddrLocality_LL	Indian language value of "AddrLocality"	60
AddrVTC	alpha-numeric (a-z, A-Z, 0-9, space, #,,comma,Dot,Front slash, hyphen)	50
AddrVTC_LL	Indian language value of "AddrVTC"	50
AddrDistrict	alpha-numeric (a-z, A-Z, 0-9, space, &)	50
AddrDistrict_LL	Indian language value of "AddrDistrict"	50
AddrState	alpha-numeric (a-z, A-Z, 0-9, space, &)	50
AddrState_LL	Indian language value of "AddrState"	50
PinCode	numeric (0-9)	6
RelativeName	alphabetic (a-z, A-Z, space, dot)	99
RelativeName_LL	Indian language value of "RelativeName"	99
RelativeUID	alpha-numeric (a-z, A-Z, 0-9, space,hyphen,Front slash)	30
RelationType	char[F,M,H,W,G]	1
Mobile⁴	numeric (0-9)	10
Email	alpha-numeric (a-z, A-Z, 0-9, hyphen,underscore,dot,@)	254

Registrars shall make sure that the string values for all the villages/Districts/States names shall match exactly with the appropriate directories.

⁴ The mobile number shall not prefix '0' or '+'

4. Enrolment client XSDs

This section contains the schema design of the various files which are part of the enrolment client packet. The respective files have been embedded as an object in Appendix.

4.1 Demographic

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.uidai.gov.in/export/manifest"
  xmlns="http://www.uidai.gov.in/export/manifest"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">

  <xs:element name="uid-enrolment-manifest">
    <xs:complexType>
      <xs:all>
        <xs:element name="registrar-code" type="xs:string" />
        <xs:element name="enrolment-agency-code" type="xs:string" />
        <xs:element name="enrolment-station-id" type="xs:string" />
        <xs:element name="enrolment-machine-id" type="xs:string" />

        <!-- (Running counter number) for the client -->
        <xs:element name="client-serial-number" type="xs:long"
          minOccurs="0" />
        <xs:element name="operator-id" type="xs:string" />
        <xs:element name="device-key" type="xs:string" minOccurs="0" />
        <xs:element name="client-version-number" type="xs:string"
          minOccurs="0" />
        <xs:element name="manifest-name" type="xs:string" />

        <!-- total number of enrolments exported with this manifest file -->
        <xs:element name="enrolments-count" type="xs:long" />
        <xs:element name="device-bar-code" type="xs:string"
          minOccurs="0" />
        <xs:element name="export-date" type="xs:string" />
        <!--
          This element would contain the list of
          enrolment ids captured
          within this manifest file
        -->
        <xs:element name="enrolment-packets" type="PacketListType" />

      </xs:all>
      <xs:attribute name="registrar-code" type="xs:string"

```

```

        use="optional" />
<xs:attribute name="enrolment-agency-code" type="xs:string"
        use="optional" />
<xs:attribute name="enrolment-station-id" type="xs:string"
        use="optional" />
<xs:attribute name="enrolment-machine-id" type="xs:string"
        use="optional" />
<xs:attribute name="client-serial-number" type="xs:long"
        use="optional" />
<xs:attribute name="operator-id" type="xs:string" use="optional" />
<xs:attribute name="device-key" type="xs:string" use="optional" />
<xs:attribute name="client-version-number" type="xs:string"
        use="optional" />
<xs:attribute name="manifest-name" type="xs:string" use="optional" />
<xs:attribute name="enrolments-count" type="xs:long"
        use="optional" />
<xs:attribute name="device-bar-code" type="xs:string"
        use="optional" />
<xs:attribute name="export-date" type="xs:string" use="optional" />
</xs:complexType>
</xs:element>

<xs:complexType name="PacketListType">
  <xs:sequence>
    <xs:element name="enrolment-id" type="xs:string"
      minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!--
      Since the enrolment id is the file name of the exported packet the
      mapping structure is no more required
-->

<!--
  <xs:complexType    name="EnrolmentPacketType">    <xs:all>
<xs:element
      name="enrolment-id"    type="xs:string"    minOccurs="0"    />
<xs:element
      name="packet-file" type="xs:string" minOccurs="0" /> </xs:all>
<xs:attribute    name="enrolment-id"    type="xs:string"    />
<xs:attribute
      name="packet-file" type="xs:string" /> </xs:complexType>
-->
</xs:schema>

```

4.2 Biometrics

```

<?xml version='1.0' encoding="utf-8"?>
<xs:schema          targetNamespace="urn:oid:1.1.19785.0.257.1.7.0"
xmlns="urn:oid:1.1.19785.0.257.1.7.0"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
      attributeFormDefault="unqualified">

  <xs:element name="bir" type="bir" />

  <xs:complexType name="bir">
    <xs:sequence>
      <xs:element name="version" type="RevisionType" minOccurs="0"
/>
      <xs:element      name="cbeff-version"      type="RevisionType"
minOccurs="0" />
      <xs:element name="bir-info" type="BIRInfoType" minOccurs="0"
/>
      <xs:element      name="bdb-info"          type="BDBInfoType"
minOccurs="0" />
      <xs:choice>
        <xs:element name="bir" type="bir" minOccurs="0"
          maxOccurs="unbounded" />
      </xs:choice>
      <xs:choice id="optBdb">
        <xs:element      name="bdb"          type="xs:base64Binary"
minOccurs="0" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="RevisionType">
    <xs:attribute name="major" type="xs:int" />
    <xs:attribute name="minor" type="xs:int" />
  </xs:complexType>

  <xs:complexType name="BIRInfoType">
    <xs:sequence>
      <xs:element name="creator" type="xs:string" minOccurs="0" />
      <xs:element name="index" type="xs:string" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="integrity" type="xs:boolean" use="optional" />
    <xs:attribute name="creation-date" type="xs:string" use="optional" />
    <xs:attribute name="not-valid-before" type="xs:string"
      use="optional" />
    <xs:attribute name="not-valid-after" type="xs:string"
      use="optional" />
  </xs:complexType>

```

```

<xs:complexType name="BDBInfoType">
  <xs:attribute name="format-owner" type="xs:string" />
  <xs:attribute name="format-type" type="xs:string" />
  <xs:attribute name="encryption" type="xs:boolean" use="optional" />
  <xs:attribute name="creation-date" type="xs:string" />
  <xs:attribute name="type" type="xs:string" use="optional" />
  <xs:attribute name="subtype" type="xs:string" use="optional" />
  <xs:attribute name="level" type="xs:string" use="optional" />
  <xs:attribute name="product-owner" type="xs:string" use="optional" />
  <xs:attribute name="product-type" type="xs:string" use="optional" />
  <xs:attribute name="purpose" type="xs:string" use="optional" />
  <xs:attribute name="quality" type="xs:string" use="optional" />
</xs:complexType>
</xs:schema>

```

4.3 Manifest

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.uidai.gov.in/export/manifest"
  xmlns="http://www.uidai.gov.in/export/manifest"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">

  <xs:element name="uid-enrolment-manifest">
    <xs:complexType>
      <xs:all>
        <xs:element name="registrar-code" type="xs:string" />
        <xs:element name="enrolment-agency-code" type="xs:string" />
        <xs:element name="enrolment-station-id" type="xs:string" />
        <xs:element name="enrolment-machine-id" type="xs:string" />

        <!-- (Running counter number) for the client -->
        <xs:element name="client-serial-number" type="xs:long"
          minOccurs="0" />
        <xs:element name="operator-id" type="xs:string" />
        <xs:element name="device-key" type="xs:string" minOccurs="0" />
        <xs:element name="client-version-number" type="xs:string"
          minOccurs="0" />
        <xs:element name="manifest-name" type="xs:string" />

        <!-- total number of enrolments exported with this manifest file -->
        <xs:element name="enrolments-count" type="xs:long" />
        <xs:element name="device-bar-code" type="xs:string"
          minOccurs="0" />
        <xs:element name="export-date" type="xs:string" />
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

captured This element would contain the list of enrolment ids
within this manifest file

```

-->
  <xs:element name="enrolment-packets" type="PacketListType" />

</xs:all>
<xs:attribute name="registrar-code" type="xs:string"
  use="optional" />
<xs:attribute name="enrolment-agency-code" type="xs:string"
  use="optional" />
<xs:attribute name="enrolment-station-id" type="xs:string"
  use="optional" />
<xs:attribute name="enrolment-machine-id" type="xs:string"
  use="optional" />
<xs:attribute name="client-serial-number" type="xs:long"
  use="optional" />
<xs:attribute name="operator-id" type="xs:string" use="optional" />
<xs:attribute name="device-key" type="xs:string" use="optional" />
<xs:attribute name="client-version-number" type="xs:string"
  use="optional" />
<xs:attribute name="manifest-name" type="xs:string" use="optional" />
<xs:attribute name="enrolments-count" type="xs:long"
  use="optional" />
<xs:attribute name="device-bar-code" type="xs:string"
  use="optional" />
<xs:attribute name="export-date" type="xs:string" use="optional" />
</xs:complexType>
</xs:element>

<xs:complexType name="PacketListType">
  <xs:sequence>
    <xs:element name="enrolment-id" type="xs:string"
      minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!--
  Since the enrolment id is the file name of the exported packet the
  mapping structure is no more required
-->

<!--
  <xs:complexType name="EnrolmentPacketType"> <xs:all> <xs:element
  name="enrolment-id" type="xs:string" minOccurs="0" /> <xs:element
  name="packet-file" type="xs:string" minOccurs="0" /> </xs:all>
  <xs:attribute name="enrolment-id" type="xs:string" /> <xs:attribute

```

```
        name="packet-file" type="xs:string" /> </xs:complexType>
    -->
</xs:schema>
```

5. Appendix



demographics.xsd



biometrics.xsd



manifest.xsd