Exchanging Patient Demographics Information using ANSI/HL7 v2.8.2

Created by:
National Resource Centre for EHR Standards,
Centre for Development of Advanced Computing (C-DAC), Pune, India

REVISION HISTORY

Date	Version	Description
09/12/2019	2.0	Document describing exchange of patient demographic information using HL7 registration message (ADT_A04)
08/08/2018	1.0	Initial release describing exchange of patient demographic information using HL7 Patient Visit/Admit message (ADT_A01)

TABLE OF CONTENTS

INTRODUCTION

There are many existing purposes specific Healthcare IT applications which needs to communicate with each other in order to have interoperability. Patient Demographic is one of the basic information which is captured and required for each application and hence it is important for interoperability.

ANSI/HL7 v2.8.2 is one of the exchange standards specified for adherence in EHR Standard for India (2016). The standard enables transfer of clinical and administrative data between software applications used by various healthcare providers. These messages are simple text files stored with '.hl7' file extension.

The standard specifies structure for transmission of new or updated demographic and visit information about patients. Generally, information entered into one system can be shared with other systems either in the form of a new message (record) or response to a query.

PURPOSE

The purpose of this document is to provide glimpse of exchanging patient information using ANSI/HL7 v2.8.2 standard.

SCOPE

The scope of the document is to construct Register a Patient message of ANSI/HL7 v2.8.2 to share patient demographics information using C-DAC's Medical informatics SDK for ANSI/HL7 v2.8.2. For the purpose of demonstration, HL7's Register a Patient (ADT_A04) message is taken as an example here. All other HL7 messages can be constructed in similar way using SDK.

This document is only for reference purpose. The approach for implementation may vary from application to application based on the requirements, resources, and scope. Also, the application may have different mechanism to share HL7 file based on the design of application e.g., socket-based communication, using import-export facility, web services, etc. User may choose to use different tool/SDK rather than using C-DAC's SDK for ANSI/HL7 v2.8.2.

This document **does NOT**, in any way, provide or substitute requirements for evaluation, certification or compliance to ANSI/HL7 v2.8.2 standard.

INTENDED AUDIENCE

The intended audience for this document includes Healthcare Providers, Central and State Health Departments, Hospital Management, Policy Makers, Healthcare IT professionals, and Medical Software and Device Manufacturers.

ADT_A04/ACK_A04 – REGISTER A PATIENT

An ADT_A04 signals that the patient has arrived or checked in as a one-time, or recurring outpatient, and is not assigned to a bed. As an example, it might be used to signal the beginning of a visit to the Emergency Room.

Some systems refer to these events as outpatient registrations or emergency admissions by providing visit start date/time.

For more information refer the Section 3.2.4 in Chapter 03 of HL7 v2.8.2 specification.

Flow diagram
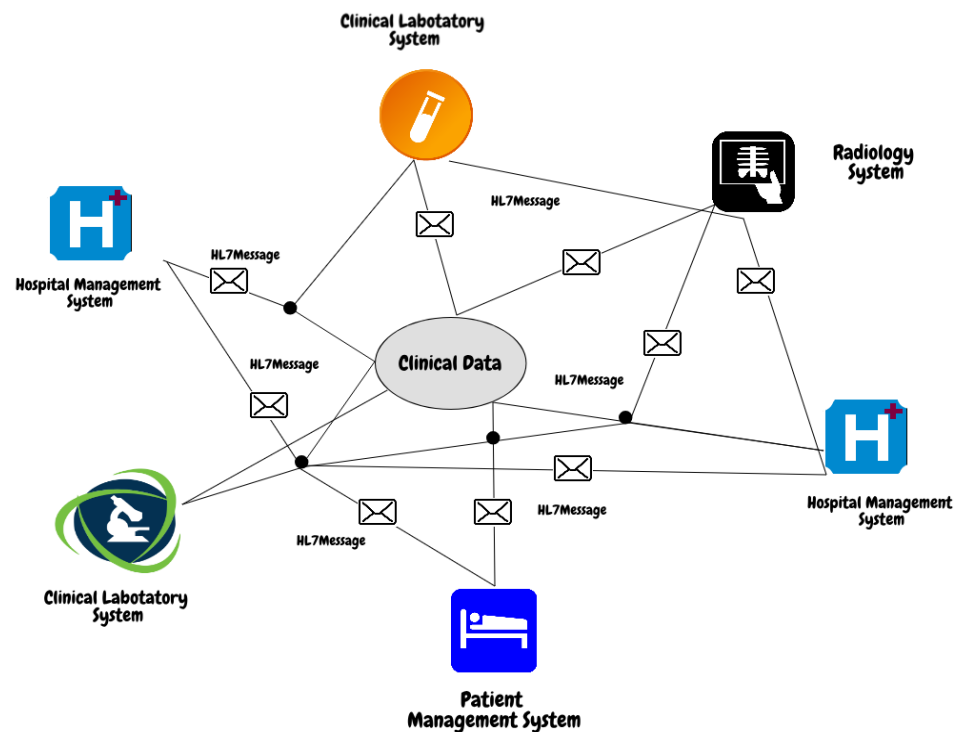
Description

If a patient information exists in one system which need to be transferred to other system to support interoperability, HL7 messages can be exchanged within the systems.

ADT_A04 Message Structure

As per the HL7 Message structure given in Section 3.2.4 of Chapter 03. Required Segments in ADT_A04 message structure are MSH, EVN, PID, PV1.

- The message header (MSH) - segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. (Refer Section 2.14.9 in Chapter 02).
- The EVN segment is used to communicate necessary trigger event information to receiving applications. (Refer Section 3.3.1 in Chapter 03).
- The PID segment contains permanent patient identifying and demographic information that, for the most part, is not likely to change frequently. (Refer Section 3.3.2 in Chapter 03).
- The PV1 segment is used by Registration/Patient Administration applications to communicate information on an account or visit-specific basis. The default is to send account level data. To use this segment for visit

level data PV1-51 - Visit Indicator must be valued to "V". (Refer Section 3.3.3 in Chapter 03).

For detailed information of message structure refer the Section 3.2.4 in Chapter 03 of HL7 v2.8.2 specification.

Sample ADT_A04 Message

```
MSH|^~\&#|Patient Care Application |Sharma Hospital|Laboratory
Application|Wellness Diagnostic Centre
|20181221||ADT^A04^ADT_A01|MSG0001||2.8.2|123||AL|AL||ASCII
EVN||201812211123
PID|1||458762586895^^^UIDAI^CZ||Joshi^Atul^^^Mr||19850304|^Male||20
28-9^Asian^HL70005|&MG Road&C-
601^^Pune^Maharastra^411007^India^Home||^PRS^CP^Home^+91^^7898
586959|^WPN^PH^Work^+91^020^123456^234||M^Married|HIN^Hindu^H
L70006|2546412315^^^^AN^^^^^&Bank of India|||||||||^Indian
PV1|1|O^Outpatient^HL70396||||||||12369^Joshi^Anurag^^^DR^^^Maharash
tra Medical Council
OBX|1|ST|274640006^Fever with chills^SCT||38|258710007^Degrees
Celsius^SCT|36.1°C-37.2°C
AL1|1|EA^Environmental Allergy^HL70127|390952000^Dust Allergy^SCT
DG1|1||420079008^Recurrent fever^SCT||20181221|89100005^Final
Diagnosis^SCT
IN1|1|HEALTHPLAN001^ Star Health Insurance Plan
^L|ICID12345^^^^NPI|Star Health Insurance Company Ltd
```

A 29-year married male Mr Atul Joshi resident of Pune attended OPD on December 21, 2018, at 11:23 a.m. by DR. Anurag Joshi Attending (Reg. No. 12369) for general Medicine. chief complains of fever with chills for 3 days, having Rashes on dust allergy and rashes. Blood sample send for viral infection investigation. Patient diagnosed with recurrent fever.

The message was sent from system "Patient Care Application" at the "Sharma Hospital" to "Laboratory Application" at "Wellness Diagnostic Centre". Patient has "Star health insurance plan" under insurance company "Star Health Insurance Company Ltd".

In segment PID, the PID-17 field is to refer patient's religion. Its values are defined in 'User-defined Table 0006 – Religion'. In above message PID-17 is having value "HIN^Hindu^HL70006", where HIN is code for religion Hindu and name of coding system is HL70006 in which 0006 is table number for codes of religion.

In Segment OBX, the OBX-3 field is for observation identifier. In above message OBX-3 is having value "274640006^Fever with chills^SCT", where "274640006" is a SNOMED CT code for finding "Fever with chills" and SCT is the name of coding

system. The OBX-2 is used to define the data type of OBX-5 field which is ST in above message. The OBX-5 field contains observation value, which is 38 in above message. The OBX-6 field contains units, it is having value "258710007^Degrees Celsius^SCT" where "258710007" is SNOMED CT code for "Degrees Celsius".

CONSTRUCTING ANSI/HL7 V2.8.2 MESSAGE USING C-DAC'S MEDICAL INFORMATICS SDK FOR ANSI/HL7 V2.8.2 (E.G. ADT_A04)

In SDK, for creating HL7 messages there are two entities are used HL7 Source and Hl7 Recipient. Both provides capability to create HL7 Messages for defined systems. HL7 Source and HL7 Recipient can send and receive HL7 messages depending on what kind of message it is.

Capabilities of HL7 source and recipient,

1. HL7 Source

    Can send – Query and Acknowledgement Messages

    Can receive – Event, Response and Acknowledgement Messages

2. HL7 Recipient

    Can send – Event, Response and Acknowledgement Messages

    Can receive – Query and Acknowledgement Messages

Steps to create hl7 message

1. Identify the type of message whether it is Event, Query, response or an acknowledgement to create the message object using hl7 source or hl7 recipient entity. E.g. for above example message ADT_A04 create object by using **ADTRecepient**.

> *IMessageSource objADTRecepient = new ADTRecepient ();*
> *Message objADT_A04 =*
> *objADTRecepient.createMessage(EnumMessageCode.ADT,*
> *EnumTriggerEvent.A04);*

2. Populate Message (Fill the information in object created in step 1) by populating segments (Required segments must be populated).

```
IMessageSource objADTRecepient = new ADTRecepient ();
Message objADT_A04 = objADTRecepient.createMessage(EnumMessageCode.ADT,
EnumTriggerEvent.A04);
//MSH segment
MSH objMSH = populateMSH();
objADT_A01.setHeader(objMSH);

//EVN segment
EVN objEVN = populateEVN();
objADT_A01.setEventTypeSegment(objEVN);

//PID segment
PID objPID = populatePID();
objADT_A01.setPatientIdentificationSegment(objPID);

//PV1 segment
PV1 objPV1 = populatePV1();
objADT_A01.setPatientVisitSegment(objPV1);

//OBX Segment
OBX objOBX = populateOBX();
objADT_A04.addObservationOrResultSegment(objOBX);

//AL1 Segment
AL1 objAL1 = populateAL1();
objADT_A04.addAllergyInformationSegment(objAL1);

//DG1 Segment
DG1 objDG1 = populateDG1();
objADT_A04.addDiagnosisInformationSegment(objDG1);

IMap objMessageMap =
MessageMapReader.createInstance().getMessageMap(EnumMessageCode.ADT,
EnumTriggerEvent.A04);

IMap objInsuranceGroupMap =
objMessageMap.getGroupMap(EnumSegments.HL7_GROUP_INSURANCE);

IGroup objInsuranceGroup = new
Group(EnumSegments.HL7_GROUP_INSURANCE, objInsuranceGroupMap);

//IN1 Segment
IN1 objIN1 = populateIN1();
objInsuranceGroup.addSegment(EnumSegments.HL7_SEG_IN1, objIN1);

objADT_A04.addInsuranceGroup(objInsuranceGroup);
```

Population of segment

MSH - Message Header Segment

The MSH segment defines the intent, source, destination, and some specifics of the syntax of a message.

```
MSH objMSH = new MSH();
objMSH.setFieldSeparator('|');//ST
objMSH.setEncodingChars('^','&','~','\\','#');//ST
objMSH.setDateTimeOfMessage("201812211123");//TS
objMSH.setMessageType("ADT","A04","ADT_A01");
objMSH.setMessageControlID("MSG0001");//ST
objMSH.setVersionID("2.8.2", null, null);//VID
objMSH.setSequenceNumber("123");//NM
objMSH.setAcceptAcknowledgmentType("AL");//ID AL-Always
objMSH.setApplicationAcknowledgmentType("AL");//ID AL-Always
objMSH.addCharacterSet("ASCII");//ID
objMSH.setReceivingApplication("Laboratory Application", null, null);
objMSH.setSendingApplication("Patient Registration Application ", null, null);
objMSH.setSendingFacility("Sharma Hospital", null, null);
objMSH.setReceivingFacility("Wellness Diagnostic Centre ", null, null);
```

EVN - Event Type Segment

The EVN segment is used to communicate necessary trigger event information to receiving applications.

```
//EVN segment
EVN objEVN = new EVN();
objEVN.setRecordedDateTime("201812211123");//DTM
```

PID - Patient Identification Segment

The PID segment contains permanent patient identifying and demographic information that, for the most part, is not likely to change frequently.

```java
PID objPID = new PID();
objPID.setSetID_PID("1");
//To Populate Patient Identifier List use CX data type
CX objCX = new CX(0,0,null, HL7Constants.HL7_COMPONENT);
objCX.setIdNumber("458762586895");
HD objAssigningAuthority=new HD(0, 0, null,
HL7Constants.HL7_SUBCOMPONENT);
objAssigningAuthority.setNamespaceId("UIDAI");
objCX.setAssigningAuthority(objAssigningAuthority);
objCX.setAssigningAuthority(objAssigningAuthority);
objCX.setIdentifierTypeCode("CZ"); // Citizenship Card
objPID.addPatientIdentifierList(objCX);//CX

//To Populate Patient Name use XPN data type
XPN objXPNPatientName=new XPN(0, 0, null,
HL7Constants.HL7_COMPONENT);
FN objFNfamilyname = new FN(0,0,null,
HL7Constants.HL7_SUBCOMPONENT);
objFNfamilyname.setSurname("Joshi");
objXPNPatientName.setFamilyName(objFNfamilyname);
objXPNPatientName.setGivenName("Atul");
objXPNPatientName.setPrefix("Mr");
objPID.addPatientName(objXPNPatientName);//XPN

objPID.setDateTimeOfBirth("19890720");

//To Populate Administrative Sex use CWE data type
CWE objCWESex = new CWE(0, 0 , null, HL7Constants.HL7_COMPONENT);
objCWESex.setText("Male");
objPID.setAdministrativeSex(objCWESex);

//To Populate patient Address use XAD data type
XAD objXADHomeaddress = new XAD(0,0,null,
HL7Constants.HL7_COMPONENT);
SAD objSADstreetaddr = new
SAD(0,0,null,HL7Constants.HL7_SUBCOMPONENT);
objSADstreetaddr.setStreetName("MG Road");
objSADstreetaddr.setDwellingNumber("C-601");

objXADHomeaddress.setStreetAddress(objSADstreetaddr);
objXADHomeaddress.setCity("Pune");
objXADHomeaddress.setStateOrProvince("Maharastra");
objXADHomeaddress.setZipOrPostalCode("411007");
objXADHomeaddress.setCountry("India");
objXADHomeaddress.setAddressType("Home");
objPID.addPatientAddress(objXADHomeaddress);//XAD
```

```java
//To Populate Race use CWE data type
CWE objCWERace = new CWE(0, 0 , null, HL7Constants.HL7_COMPONENT);
objCWERace.setIdentifier("2028-9");
objCWERace.setText("Asian");
objPID.addRace(objCWERace);

//To Populate Phone Number - Home use XTN data type
XTN objXTNHomePhoneNo = new XTN(0,0,null,
HL7Constants.HL7_COMPONENT);
objXTNHomePhoneNo.setTelecommunicationUseCode("PRS");
objXTNHomePhoneNo.setTelecommunicationEquipmentType("CP");
objXTNHomePhoneNo.setCommuinicationAddress("Home");
objXTNHomePhoneNo.setCountryCode("+91");
objXTNHomePhoneNo.setLocalNumber("7898586959");
objPID.addPhoneNumberHome(objXTNHomePhoneNo);//XTN

//To Populate Phone Number - Business use XTN data type
XTN objXTNBusinessPhoneNo = new XTN(0,0,null,
HL7Constants.HL7_COMPONENT);
objXTNBusinessPhoneNo.setTelecommunicationUseCode("WPN");
objXTNBusinessPhoneNo.setTelecommunicationEquipmentType("PH");
objXTNBusinessPhoneNo.setCommuinicationAddress("Work");
objXTNBusinessPhoneNo.setCountryCode("+91");
objXTNBusinessPhoneNo.setAreaCityCode("020");
objXTNBusinessPhoneNo.setLocalNumber("123456");
objXTNBusinessPhoneNo.setExtension("234");
objPID.addPhoneNumberBusiness(objXTNBusinessPhoneNo);//XTN

//To Populate Marital Status use CWE data type
CWE objCWEMaritalStatus = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEMaritalStatus.setIdentifier("M");
objCWEMaritalStatus.setText("Married");
objPID.setMaritalStatus(objCWEMaritalStatus);

//To Populate Religion use CWE data type
CWE objCWEReligion = new CWE(0, 0 , null, HL7Constants.HL7_COMPONENT);
objCWEReligion.setIdentifier("HIN");
objCWEReligion.setText("Hindu");
objPID.setReligion(objCWEReligion);

//To Populate Citizenship use CWE data type
CWE objCWEcitizenship = new CWE(0, 0 , null, HL7Constants.HL7_COMPONENT);
objCWEcitizenship.setText("Indian");
objPID.addCitizenship(objCWEcitizenship);
```

```
//To Populate Patient Account Number use CX data type
CX objCXAccountNo = new CX(0,0,null, HL7Constants.HL7_COMPONENT);
objCXAccountNo.setIdNumber("2546412315");
objCXAccountNo.setIdentifierTypeCode("AN");//Account Number
CWE objCWEAssigAgencyDept=new CWE(0,0,null,
HL7Constants.HL7_SUBCOMPONENT);
objCWEAssigAgencyDept.setText("Bank of India");
objCXAccountNo.setAssigningAgencyOrDepartment(objCWEAssigAgencyDept);
objPID.setPatientAccountNumber(objCXAccountNo);//CX
```

PV1 - Patient Visit Segment

The PV1 segment is used by Registration/Patient Administration applications to communicate information on an account or visit-specific basis.

```
PV1 objPV1 = new PV1();
objPV1.setSetID("1");

//To Populate Patient Class use CWE data type
CWE objCWEPatientClass = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEPatientClass.setIdentifier("O");
objCWEPatientClass.setText("Outpatient");
objCWEPatientClass.setNameOfCodingSystem("HL70396");
objPV1.setPatientClass(objCWEPatientClass);

//To Populate Consulting Doctor information use XCN data type
XCN objXCNCounsulDoctor = new XCN(0,0,null,
HL7Constants.HL7_COMPONENT);
objXCNCounsulDoctor.setIdNumber("12369");
FN objFNDocfamilyname = new FN(0,0,null,
HL7Constants.HL7_SUBCOMPONENT);
objFNDocfamilyname.setSurname("Joshi");
objXCNCounsulDoctor.setFamilyName(objFNDocfamilyname);
objXCNCounsulDoctor.setGivenName("Anurag");
objXCNCounsulDoctor.setPrefix("DR");
HD objdocAssigningAuthority=new HD(0,0,null,
HL7Constants.HL7_SUBCOMPONENT);
objdocAssigningAuthority.setNamespaceId("Maharashtra Medical Council");
objXCNCounsulDoctor.setAssigningAuthority(objdocAssigningAuthority);

objPV1.addConsultingDoctor(objXCNCounsulDoctor);
```

OBX - Observation/Result Segment

The OBX segment is used to transmit a single observation or observation fragment. It represents the smallest indivisible unit of a report.

```
OBX objOBX = new OBX();
objOBX.setSetID_OBX("1");
objOBX.setValueType("ST");

//To Populate Observation Identifier use CWE data type
CWE objCWEPatientClass = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEPatientClass.setIdentifier("274640006 ");
objCWEPatientClass.setText("Fever with chills");
objCWEPatientClass.setNameOfCodingSystem("SCT");
objOBX.setObservationIdentifier(objCWEPatientClass);

//To Populate Observation Value using ST data type as provided in OBX-2
field
ST objSTObservationValue= new ST();
objSTObservationValue.setValue("38");
objOBX.addObservationValue(objSTObservationValue);

//To Populate Units use CWE data type
CWE objCWEUnits=new CWE(0, 0 , null, HL7Constants.HL7_COMPONENT);
objCWEUnits.setIdentifier("258710007");
objCWEUnits.setText("Degrees Celsius");
objCWEUnits.setNameOfCodingSystem("SCT");
objOBX.setUnits(objCWEUnits);

objOBX.setReferencesRange("36.1°C-37.2°C");
```

AL1 - Patient Allergy Information Segment

The AL1 segment contains patient allergy information of various types.

```
AL1 objAL1 = new AL1();
objAL1.setSetID_AL1("1");

//To Populate Allergen Type Code use CWE data type
CWE objCWEallergenTypeCode = new
CWE(0,0,null,HL7Constants.HL7_COMPONENT);
objCWEallergenTypeCode.setIdentifier("EA");
objCWEallergenTypeCode.setText("Environmental Allergy");
objCWEallergenTypeCode.setNameOfCodingSystem("HL70127");
objAL1.setAllergenTypeCode(objCWEallergenTypeCode);

//To Populate Allergen Code Mnemonic Description use CWE data type
CWE objCWEAllergen = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEAllergen.setIdentifier("390952000");
objCWEAllergen.setText("Dust Allergy");
objCWEAllergen.setNameOfCodingSystem("SCT");

objAL1.setAllergenCodeMnemonicDescription(objCWEAllergen);
```

DG1 - Diagnosis Segment

The DG1 segment contains patient diagnosis information of various types, for example, admitting, primary, etc.

```
DG1 objDG1 = new DG1();
objDG1.setSetID_DG1("1");

//To Populate Diagnosis Code use CWE data type
CWE objCWEDiagnosis = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEDiagnosis.setIdentifier("420079008");
objCWEDiagnosis.setText("Recurrent fever");
objCWEDiagnosis.setNameOfCodingSystem("SCT");
objDG1.setDiagnosisCode_DG1(objCWEDiagnosis);

objDG1.setDiagnosisDateTime("201812211150");//DTM

//To Populate Diagnosis Type use CWE data type
CWE objCWEDiagnosisType = new CWE(0, 0 , null,
HL7Constants.HL7_COMPONENT);
objCWEDiagnosisType.setIdentifier("89100005");
objCWEDiagnosisType.setText("Final Diagnosis");
objCWEDiagnosisType.setNameOfCodingSystem("SCT");
objDG1.setDiagnosisType(objCWEDiagnosisType);
```

IN1 - Insurance Segment

The IN1 segment contains insurance policy coverage information necessary to produce properly pro-rated and patient and insurance bills.

```
IN1 objIN1 = new IN1();
objIN1.setSetID_IN1("1");

//To Populate Health Plan ID use CWE data type
CWE objCWEHealthPlanId = new CWE(0, 0 ,
null,HL7Constants.HL7_COMPONENT);
objCWEHealthPlanId.setIdentifier("HEALTHPLAN001");
objCWEHealthPlanId.setText("Star Health Insurance Plan ");
objCWEHealthPlanId.setNameOfCodingSystem("L");
objIN1.setHealthPlanID(objCWEHealthPlanId);

//To Populate Insurance Company ID use CX data type
CX objCXInsuranceCompanyID = new CX(0,0,null,
HL7Constants.HL7_COMPONENT);
objCXInsuranceCompanyID.setIdNumber("ICID12345");
objCXInsuranceCompanyID.setIdentifierTypeCode("NPI");
objIN1.addInsuranceCompanyID(objCXInsuranceCompanyID);//CX

//To Populate Insurance Company name use XON data type
XON objXONinsurancecompname = new XON(0,0,null,
HL7Constants.HL7_COMPONENT);
objXONinsurancecompname.setOrganizationName("Star Health Insurance
Company Ltd");
objIN1.addInsuranceCompanyName(objXONinsurancecompname);//XON
```

Steps to write hl7 message

Write populated HL7 messages object in file having extension ".hl7".
To write Messages in file we need to create object of **HL7Serializer** to call **serialize** method.

```
String strHL7File = "C:/ADT_A04Message.HL7";
objSegmentSet = objADT_A04.getSegmentSet();
objHL7Serializer = new HL7Serializer();
objHL7Serializer.serialize(objSegmentSet, strHL7File);
```

Steps to read hl7 message

HL7 Parser: The HL7Parser parses the HL7Message and provides a **SegmentSet**. SegmentSet contains all segments of a HL7 Message in the same order as they appeared.

© NRCeS @ C-DAC

Centre for Development of Advanced Computing, Pune   Page **15** of **18**

1. Retrieve Segments from Message

```
String strFilePath = "C:/ ADT_A04Message.HL7";
FileInputStream objFileInputStream = new
FileInputStream(strFilePath);
HL7FileStream objHL7FileStream = new HL7FileStream();
objHL7FileStream.setInputStream(objFileInputStream);
ISegmentSet objADT_A04SegmentSet = parse(objHL7FileStream);
```

2. Get segments from Segment set object to access data in segment.

```
MSH objMSH =objADT_A04SegmentSet. getHeader();
PID objPID= objADT_A04SegmentSet.
getPatientIdentificationSegment();
```

3. Get data from segments

```
ST objST= objMSH.getMessageControlID();
```

MAPPING ISO/TS 22220:2011 AND HL7 DATA FIELDS

The below table covers mapping of required data elements in ISO/TS 22220:2011 to HL7 v2.8.2 data fields for patient demographics. Implementers are expected to specifically ensure population and exchange of all ISO 22220 data elements in HL7 v2.8.2 message in order to ensure compliance to ISO 22220 standard as part of EHR Standards for India (2016). This is an indicative mapping and it is observed that, most of the data elements of ISO standard are covered in HL7.

ISO/TS 22220 Mapping with HL7 v2.8.2

Sr No.	ISO/TS 22220	HL7 v2.8.2
1.	Patient identifier designation	PID-3 Patient Identifier List (CX) CX-10 Assigning Agency or Department (CWE)
2.	Patient identifier geographic area	PID-3 Patient Identifier List (CX) CX-9 Assigning Jurisdiction (CWE)
3.	Patient identifier issuer	PID-3 Patient Identifier List (CX) CX-4 Assigning Authority (HD)
4.	Patient identifier type	PID-3 Patient Identifier List (CX) CX-5 Identifier Type Code (ID)
5.	Family name	PID-5 Patient Name (XPN) XPN-1 family name (FN)
6.	Family name sequence number	NA (Family name sequence number is deprecated in Hl7 v2.8.2)
7.	Preferred name	PID-5 Patient Name (XPN) XPN-2 Given Name (ST)
8.	Sex	PID-8 Administrative Sex (CWE)

NOTE

- The guide is constructed assuming the implementer has access and knowledge of the 2.8.2 version of the HL7 specification.
- The standard specifications can be accessed from, http://www.hl7.org/implement/standards/product_brief.cfm?product_id=403
- It is also assumed the user of this guide has gone through Chapter 02: Control and Chapter 03: Patient Administration

In evidence of any material error, change, correction, concerns or assistance regarding this document, you are requested to immediately report it at: nrc-help@cdac.in

REFERENCE

- For more information about how to use SDK refer the HL7 Tutorial from below link.
  https://www.cdac.in/index.aspx?id=hi_hs_HL7_Tutorial
- Refer sample test codes in CDAC's SDK v3.1 for HL7 v2.8.2 for more message's population examples from below link.

https://www.cdac.in/index.aspx?id=hi_hs_hl7_sdk_v3.1
- Refer below link to download HL7 v2.8.2 Specification
http://www.hl7.org/implement/standards/product_brief.cfm?product_id=403
- https://www.nrces.in/standards/hl7-international/hl7-v2.8.2