C-DAC's Medical Informatics SDK for Continuity of Care Document (CCD) Tutorials

1. Introduction to SDK for CCD

The Continuity of Care Document (CCD) is an electronic document exchange standard for sharing patient summary information. Summaries include the most commonly needed pertinent information about current and past health status in a form that can be shared by all computer applications. The Continuity of Care Document is a compromise reached by two standards groups, ASTM International and Health Level 7 (HL7).

C-DAC's Medical Informatics SDK for CCD is a set of object-oriented APIs which can be used to make the Healthcare Applications compliant to HL7/ASTM CCD Release 1 specification. The SDK is implemented using JDK 8, which facilitates incorporation of HL7/ASTM CCD in healthcare applications. It provides set of APIs which can be integrated in application to generate CCD and APIs for read and validate already existing CCD.

Back to top

2. Programming with SDK for CCD

2.2 Basic Configuration

2.2.1 How to enable logging

Log can be generated in CCD SDK for different events occurred while processing. The SDK describes two levels of logging:

SEVERE - At this level of logging any exception's stack trace is completely logged into a log file. Along with the stack trace the data that are failed are also logged.

INFO - The failed data elements with status WARNING are logged in the log file.

Ideally this should be the first statement before using CCD SDK.

For e.g.:

```
// log files will be created in ' C:\\CCDLogs' directory.
CCDLogger.getLogger("C:\\CCDLogs");

// If no directory location is defined by user then user's
//temp directory will be taken as default.
```

Back to top

2.2 Working with CCD Header

2.2.1 CCD Header Description

The CCD Header contains information about Patient, document related information (language, version, date/time), CCD Author related information (who has generated), and to whom the CCD is directed. The purpose of the CCD header is to facilitate clinical document management and compilation of a patient's clinical documents into a lifetime electronic record. CCD Header is having following elements:

- Legal Authenticator
- Custodian
- Informant
- Participant
- Performer
- Record Target
- Author
- DocumentationOfInfo
- OrganizationInfo
- Identifiers

Back to top

2.2.2 How to Populate CCD Header

CCD Header can be populated like this

```
//Populate CCD Header
CCDHeader objCCDHeader= objCCDWriterTestCode.populateCCDHeader();

//Populates other header parts such as AuthorInfo, Patient Details and add in
objCCDHeader

List<PatientDetails> objPatient= populatePatientDetails() ;
objCCDHeader.getPatientDetails().addAll(objPatient);

List<AuthorInfo> objAuthor = populateAuthor();
objCCDHeader.getAuthor().addAll(objAuthor);
```

To Populate a CCD Header's fields consider *PatientDetails*, Following are the populatePatientDetails function code to populate patient details.

```
List<PatientDetails> objListPatient = new ArrayList<>();

PatientDetails objPatient= new PatientDetails();
objPatient.setName("Rahul");
objPatient.setSuffix("Singh");
objPatient.setBirthTime("19580304");
objPatient.setMaritalStatus("married");
                    .
                    .
.

objListPatient.add(objPatient);
```

Back to top

2.3 Working with CCD Sections

CCD contains the patient-related data, such as problems, and procedures, current and past medications, Data are clustered into sections based on common clinical gathering.

Back to top

2.3.1 How to Populate CCD Sections

2.3.1.1 How to Populate Summary-Purpose Section

This Part of Clinical Document represents the reason for which the CCD Document was generated. Purpose may be transfer, referral, or patient request.

CCD Summary-Purpose Section can be populated like this

```
//Populates Summary-Purpose Section

CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<SummaryPurpose> objPurpose =
objCCDWriterTestCode.populateSummaryPurpose();
        objCCDWriter.addSummaryOrPurpose(objPurpose);

Here, objCCDHeader is CCD Header's Object
```

To Populate CCD Summary Purpose section's fields from populateSummaryPurpose function, Instructions are

```
// Populates Values of Summary Purpose Section
List<SummaryPurpose> objPurpose = new ArrayList<>();
SummaryPurpose objSP = new SummaryPurpose();
objSP.setPurpose("Transfer of care");

objPurpose.add(objSP);
```

Back to top

2.3.1.2 How to Populate Payer Section

Payers Section contains patient's payers related information, whether self-pay, or third-party insurance, or some combination of payers or other pay. It also contains information regarding policy type or coverage type.

CCD Payer Section can be populated like this

```
//Populates Payer Section

CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Payer> objPayer = objCCDWriterTestCode.populatePayer();
objCCDWriter.addPayer(objPayer);

Here, objCCDHeader is CCD Header's Object
```

To Populate CCD Payer section's fields from populatePayer function, Follow the below given instructions

```
// Populates Values of Payer Section List<Payer>
objPayer = new ArrayList<>();

Payer Payer = new Payer();
Payer.setAuthorization("Colonoscopy");
Payer.setCoveredPartyID("2");
Payer.setPayerName("Max Health Insurance");
Payer.setPolicyType("Extended healthcare");
Payer.setCoverageType("Self");

objPayer.add(Payer);
```

Back to top

2.3.1.3 How to Populate Advance-Directive Section

This section contains data defining related to patient's advance directives and any reference to supporting documentation.

CCD Advance-Directive Section can be populated like this

```
//Populates Advance-Directive Section

CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<AdvanceDirectives> objAdvanceDirectives =
objCCDWriterTestCode.populateAdvanceDirectives();
objCCDWriter.addAdvanceDirectives(objAdvanceDirectives);

Here, objCCDHeader is CCD Header's Object
```

Example to Populate CCD Advance-directive section's fields from populateAdvanceDirectives function,

```
// Populates Values of Advance Directive Section List<AdvanceDirectives>
objAdvanceDirectives = new ArrayList<>();

AdvanceDirectives advanceDirectives = new AdvanceDirectives();
advanceDirectives.setDirective("Resuscitation status");
advanceDirectives.setDescription("Do not resuscitate");
advanceDirectives.setVerification("Dr. N. K. Jain");
advanceDirectives.setVerificationDate("19990711");
advanceDirectives.setSupportingDocument("Advance directive");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setLow("19990711");
objEffectiveDate.setHigh("20010711");
advanceDirectives.setEffectiveTimes(objEffectiveDate);
advanceDirectives.setMediaType("pdf");
advanceDirectives.setSupportingDocumentLink("AdvanceDirective.b50b7910-7ffb-
4f4c-bbe4-177ed68cbbf3.pdf");

objAdvanceDirectives.add(advanceDirectives);
```

Back to top

2.3.1.4 How to Populate Functional-Status Section

This section contains data related to the patient's status of normal functioning at the time the Care Record was created. It includes information regarding the patient relative to his Mental status, Ability to care for self, Activities of Daily Living etc.

CCD Functional-Status Sectioncan be populated like this

```
//Populates Functional-Status Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<FunctionalStatus> objFunctionalStatus =
objCCDWriterTestCode.populateFunctionalStatus();
objCCDWriter.addFunctionalStatus(objFunctionalStatus);
```

Setting CCD Functional Status section's fields from populateFunctionalStatus function

```
// Populates Values of Functional Status Section List<FunctionalStatus>
objFunctionalStatus = new ArrayList<>();

FunctionalStatus functionalStatus = new FunctionalStatus();
functionalStatus.setFunctionalCondition("Dependence on cane");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setLow("2008");
functionalStatus.setEffectiveDates(objEffectiveDate);
functionalStatus.setConditionStatus(" Active ");

objFunctionalStatus.add(functionalStatus);
```

Back to top

2.3.1.5 How to Populate Problem Section

This section contains information about clinical problems at the time the summary is generated.

CCD Problem Section can be populated like this

```
//Populates Problem Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Problems> objProblems = objCCDWriterTestCode.populateProblems();
objCCDWriter.addProblems(objProblems);


Here, objCCDHeader is CCD Header's Object
```

Setting CCD Problem section's fields from populateProblems function

```
//Populates Values of Problem Section
List<Problems> objProblems = new ArrayList<>();
Problems problems = new Problems();
problems.setCondition("Asthma");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setValue("1978");
problems.setEffectiveDates(objEffectiveDate);
problems.setConditionStatus("Active");

objProblems.add(problems);
```

Back to top

2.3.1.6 How to Populate Family History Section

This section contains data related to patient's genetic relatives and their clinical findings.

CCD Family-History Section can be populated like this

```
//Populates Family-History Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<FamilyHistory> objFamilyHistory =
objCCDWriterTestCode.populateFamilyHistory();
objCCDWriter.addFamilyHistory(objFamilyHistory);


Here, objCCDHeader is CCD Header's Object
```

Setting CCD Family History section's fields from populateFamilyHistory function

Back to top

```
// Populates Values of Family-History Section
List<FamilyHistory> listFamilyHistory=new ArrayList<FamilyHistory>();
FamilyHistory objFamilyHistory1=new FamilyHistory();
objFamilyHistory1.setRelationWithPatient("Father");
objFamilyHistory1.setVitalStatus("Deceased");
objFamilyHistory1.setAgeAtOnset("40");
objFamilyHistory1.setDiagnosis("Hypertension");
objFamilyHistory1.setGender("male");
objFamilyHistory1.setBirthTime("1932");

listFamilyHistory.add(objFamilyHistory1);
```

Back to top

2.3.1.7 How to Populate Social History Section

This section contains information related to patient's lifestyle.

Social-History Section can be populated like this

```
//Populates Social-History Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<SocialHistory> objSocialHistory =
objCCDWriterTestCode.populateSocialHistory();
objCCDWriter.addSocialHistory(objSocialHistory);

Here, objCCDHeader is CCD Header's Object
```

Setting Social-History section's fields from populateSocialHistory function, Follow the below given way

```
// Populates Values of Social-History Section
List<SocialHistory> objListSocialHistory = new ArrayList<>();
SocialHistory objSocialHistory = new SocialHistory();
objSocialHistory.setDescription("1 pack per day");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setLow("1947");
objEffectiveDate.setHigh("1972");
objSocialHistory.setEffectiveDates(objEffectiveDate);
objSocialHistory.setSocialHistoryElement("Cigarette smoking");

listFamilyHistory.add(objFamilyHistory1);
```

Back to top

2.3.1.8 How to Populate Alerts Section

This section contains information related allergies and adverse reactions that are pertinent to the patient's current or past medical history.

CCD Alerts Section can be populated like this

```
//Populates Alerts Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Alerts> objAlerts = objCCDWriterTestCode.populateAlerts();
        objCCDWriter.addAlerts(objAlerts);

Here, objCCDHeader is CCD Header's Object
```

Setting CCD Alert section's fields from populateAlerts function

```
// Populates Values of Alerts Section
List<Alerts> objAlerts = new ArrayList<>();
Alerts objAlert = new Alerts();
objAlert.setSubstance("Penicillin");
objAlert.setReaction("Hives");
objAlert.setStatus("Active");

objAlerts.add(objAlert);
```

Back to top

2.3.1.9 How to Populate Medication Section

The section contains detail about patient's current medications and medication history.

CCD Medication Section can be populated like this

```
//Populates Medications Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Medications> objMedications = objCCDWriterTestCode.populateMedications();
objCCDWriter.addMedication(objMedications);

Here, objCCDHeader is CCD Header's Object
```

Back to top

Setting CCD Medication section's fields from populateMedications function

```
// Populates Values of Medication Section
List<Medications> objMedications = new ArrayList<>();
Medications medications1 = new Medications();
medications1.setMedication("Prednisone");
medications1.setInstructions("20mg PO daily");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setLow("199911");
objEffectiveDate.setHigh("200401");
medications1.setDate(objEffectiveDate);
medications1.setStatus("No Longer Active ");
medications1.setDoseQuantity("1");
medications1.setPeriodValue("8");
medications1.setPeriodUnit("h");

objMedications.add(medications1);
```

Back to top

2.3.1.10 How to Populate Medical-Equipment Section

This section contains implanted and external medical devices that their health status depends on

CCD Medical-Equipment Section can be populated like this

```
//Populates Medical-Equipment Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<MedicalEquipment> objMedicalEquipment =
objCCDWriterTestCode.populateMedicalEquipment();
objCCDWriter.addMedicalEquipment(objMedicalEquipment);
```

Setting CCD Medication section's fields from populateMedicalEquipment function

```
// Populates Values of Medical-Equipment Section
List<MedicalEquipment> objListMedicalEquipment = new ArrayList<>();
MedicalEquipment objMED = new MedicalEquipment();
objMED.setDateSupplied("199911");
objMED.setSupplyDevice(" Automatic implantable cardioverter/defibrillator");

    objListMedicalEquipment.add(objMED);
```

Back to top

2.3.1.11 How to Populate Immunization Section

This section contains information related to a patient's immunization history and current immunization status.

CCD immunization Section can be populated like this

```
//Populates immunization Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Immunizations> objImmunizations =
objCCDWriterTestCode.populateImmunizations();
objCCDWriter.addImmunization(objImmunizations);

Here, objCCDHeader is CCD Header's Object
```

Setting CCD Immunization section's fields from populateImmunizations function

```
// Populates Values of immunization Section
List<Immunizations> objImmunizations = new ArrayList<>();
Immunizations immunization = new Immunizations();
immunization.setVaccine("Influenza virus vaccine");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setCenter("199911");
immunization.setDate(objEffectiveDate);
immunization.setStatus("Completed");
immunization.setMoodOfVaccine("IM");

objImmunizations.add(immunization);
```

Back to top

2.3.1.12 How to Populate Vital-Signs Section

This section contains information relevant to vital signs, such as, height, weight, blood pressure,

heart rate. CCD Vital-Sign Section can be populated like this

```
//Populates Vital-Signs Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<VitalSigns> objVitalSigns = objCCDWriterTestCode.populateVitalSign();
objCCDWriter.addVitalSigns(objVitalSigns);

Here, objCCDHeader is CCD Header's Object
```

Setting Vital-Sign section's fields from populateVitalSign function

```
// Populates Values of Vital-Sign Section
List<VitalSigns> objVitals = new ArrayList<>();
VitalSigns objV = new VitalSigns();
objV.setDate("19991114");
objV.setVitalSigns("Height");
objV.setValue("177");
objV.setUnit("cm");

objVitals.add(objV);
```

Back to top

2.3.1.13 How to Populate Result Section

This section contains information or reports of observations generated by imaging procedures, and other procedures, laboratories.

CCD Results Section can be populated like this

```
//Populates Results Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Results> objResults = objCCDWriterTestCode.populateResults();
objCCDWriter.addResults(objResults);

Here, objCCDHeader is CCD Header's Object
```

Setting Populate CCD Results section's fields from populateResults function

```
// Populates Values of Result Section
List<Results> objListResults =  new
ArrayList<>(); List<TestReport> test = new
ArrayList<>();
test.add(new TestReport("Hematology","HGB","M 13-18, F 12-16","g/dl" ,"13.2"));
test.add(new TestReport("Hematology","WBC","4.3-10.8 ","10+3/ul" ,"6.7"));
test.add(new TestReport("Hematology","PLT","135-145 ","meq/l" ,"123* "));

Results objRD = new Results();
objRD.setTestDate("20000323");
objRD.setTestDetails(test);

objListResults.add(objRD);
```

Back to top

2.3.1.14 How to Populate Procedure Section

This section contains information of treatment-methods.

Procedure Section can be populated like this

```
//Populates Procedure Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Procedures> objProcedures = objCCDWriterTestCode.populateProcedures();
        objCCDWriter.addProcedures(objProcedures);
```

Setting CCD Procedure section's fields from populateProcedures function

```
// Populates Values of Procedure Section
List<Procedures> objProcedures = new ArrayList<>();

Procedures procedures = new Procedures();
procedures.setProcedure("Total hip replacement");
procedures.setObserve("Left");
procedures.setLaterality("Left");
procedures.setDate("2008");
procedures.setProcedureActivityName("Observation");
procedures.setProcedureStatusCode("active");

objProcedures.add(procedures);
```

Back to top

2.3.1.15 How to Populate Encounters Section

This section contains information of any healthcare, An Encounter is an interaction between a practitioner and a patient.

CCD Encounter Section can be populated like this

```
//Populates Encounter Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Encounters> objEncounters = objCCDWriterTestCode.populateEncounters();
objCCDWriter.addEncounters(objEncounters);
```

Setting CCD Encounter section's fields from populateEncounters function

```
List<Encounters> objEncounters = new ArrayList<>();

Encounters Encounters = new Encounters();
Encounters.setEncounter("Checkup Examination");
Encounters.getLocation().add("XYZ Health Clinic");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setValue("20000407");
Encounters.setDate(objEffectiveDate);
objEncounters.add(Encounters);
```

Back to top

2.3.1.16 How to Populate Plan of Care Section

The section contains information of pending orders, encounters, and procedures for the patient.

CCD Plan of Care Section can be populated like this

```
//Populates Plan Section
CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
CCDWriter objCCDWriter = new CCDWriter(objCCDHeader);
List<Plan> objPlan = objCCDWriterTestCode.populatePlan();
        objCCDWriter.addPlan(objPlan);

Here, objCCDHeader is CCD Header's Object
```

Setting CCD Plan section's Value from populatePlan function

Back to top

```
List<Plan> objListPlan = new ArrayList<>();

Plan objPlan = new Plan();
objPlan.setPlannedActivity("Pulmonary function test");
EffectiveDateTime objEffectiveDate=new EffectiveDateTime();
objEffectiveDate.setCenter("20000421");
objPlan.setPlannedDate(objEffectiveDate);
objPlan.setPlanOfCareActivityName("Observation");
objPlan.setActivityMoodCode("RQO");

objListPlan.add(objPlan);
```

Back to top

2.4 Working with CCD Reader

The *CCD* parses and validates Continuity of Care Document (CCD) which is in XML format and provides information of Patient ‘s medical summary, if form of header *and sections*.

To parse a CCD Document, first, provide CCD Document Path in CCDReader which gives information of Patient ‘s medical summary.

Initializing CCDReader Object

```
String strFilePath = “C:/ClinicalDocument.xml”;
CCDReader objCCDReader = new CCDReader(strFilePath);

Now retrieve the information of Patient
List<PatientDetails> objPatientList = objCCDReader.getPatientDetails();
```

Fetching information of various Section from CCD

```
//Populates Section (consider Payers Section)

List<Payer> objList =objCCDReader.getPayers();
System.out.println("Payer Details ");
for(Payer a :objList)
{
        System.out.println("Authorization : "+a.getAuthorization());
        System.out.println("CoveredPartyID : "+a.getCoveredPartyID());
        System.out.println("PayerName : "+a.getPayerName());
        System.out.println("PolicyType : "+a.getPolicyType());
}
```

Back to top

2.5 Working with CCDWriter

CCD Writer serializes and validates generated CCD XML document. To Serialize a CCD, First Populates various Sections & Header and add information into Continuity of Care Document in xml Format.

Adding Section into CCDWriter

```
//Populates Section (consider Problems Section)

CCDWriterTestCode objCCDWriterTestCode = new CCDWriterTestCode();
List<Problems> objProblems = objCCDWriterTestCode.populateProblems();
objCCDWriter.addProblems(objProblems);
```

Back to top

2.6 Working with CCDValidator

CCD Validator is used to validate Existing Continuity of Care Document.

To Validate a CCD Document

```
//Validates Continuity of Care Document, Provide CCD File Path
String strFilePath = "C:/ClinicalDocument.xml";
CCDValidator.validate(strFilePath);
```

Back to top