Emulator

Logical Design

2/19/2009

Version: 1.7

Revision & Sign-off Sheet
Change Record

Date	Author	Version	Change Reference
1/30/08	Sri Koka	1.0	Initial Draft
2/15/08	Chris Snipes / Sri Koka	1.1	High Level Architecture
2/22/2008	Parag Tengshe	1.2	Changes in Tables and Stored Procedures
2/27/2008	Parag Tengshe	1.3	Columns added to Patient Master table
3/10/08	Parag Tengshe	1.4	New table and new stored procedures added
4/2/2008	Parag Tengshe	1.5	New table and new stored procedures added
6/10/2008	Nagendra Karri	1.6	Reviewed and re-structured.
02/19/09	Lupita Figueroa	1.7	Updated with corrected standard (CDA) Updated to reflect new Emulator architecture

Reviewers

Name	Version Approved	Position	Date

Distribution

Name	Position

AHCCCS

AHCCCS

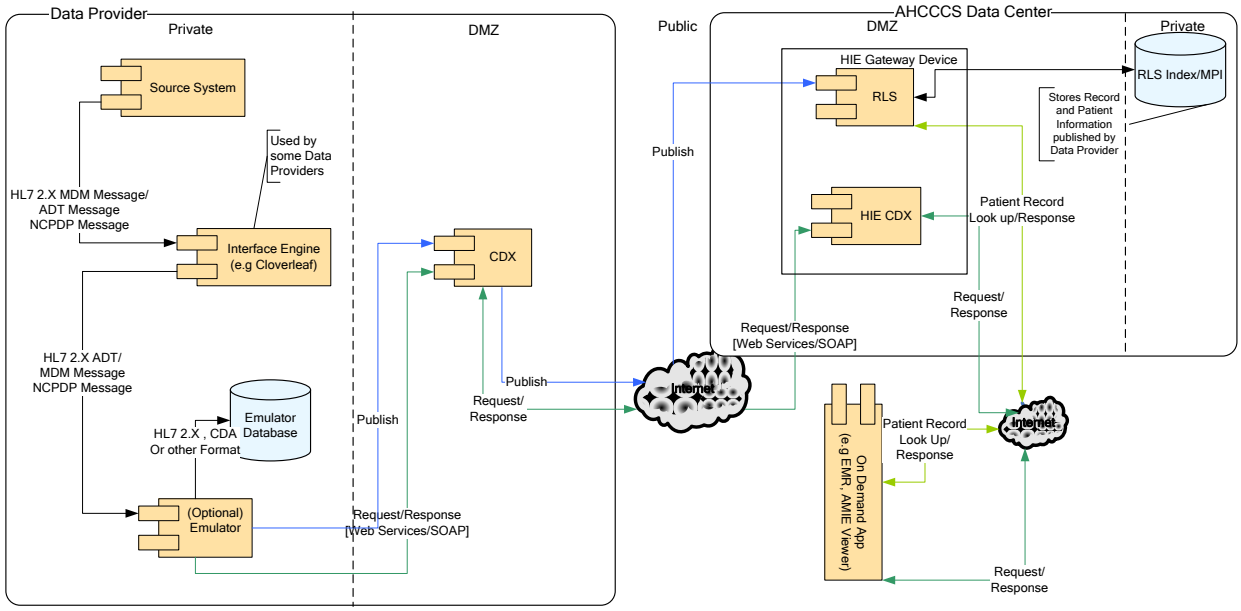
Introduction

The Emulator is an application component that built on Microsoft BizTalk Server and Microsoft BizTalk HL7 Accelerator. The Emulator processes the data received from the Data Provider systems. It will transform data delivered into a CDA format and persist it in the emulator database. Furthermore, it will publish all transformed and persisted records to the AHCCCS HIE. Also, it will provide the records to the AHCCCS HIE upon request either in CDA format or in the same format as it was received.

The Emulator is architected to support the needs of a Data provider that does not have the ability to integrate with the AHCCCS HIE natively. This solution can be deployed either at the data provider location or at the AHCCCS data center. The Emulator would support many different methods of transferring the data from the Data provider, and will connect to the CDX/PDX gateway using Web Services over HTTPS. The gateway itself can be either at the data provider or at AHCCCS.

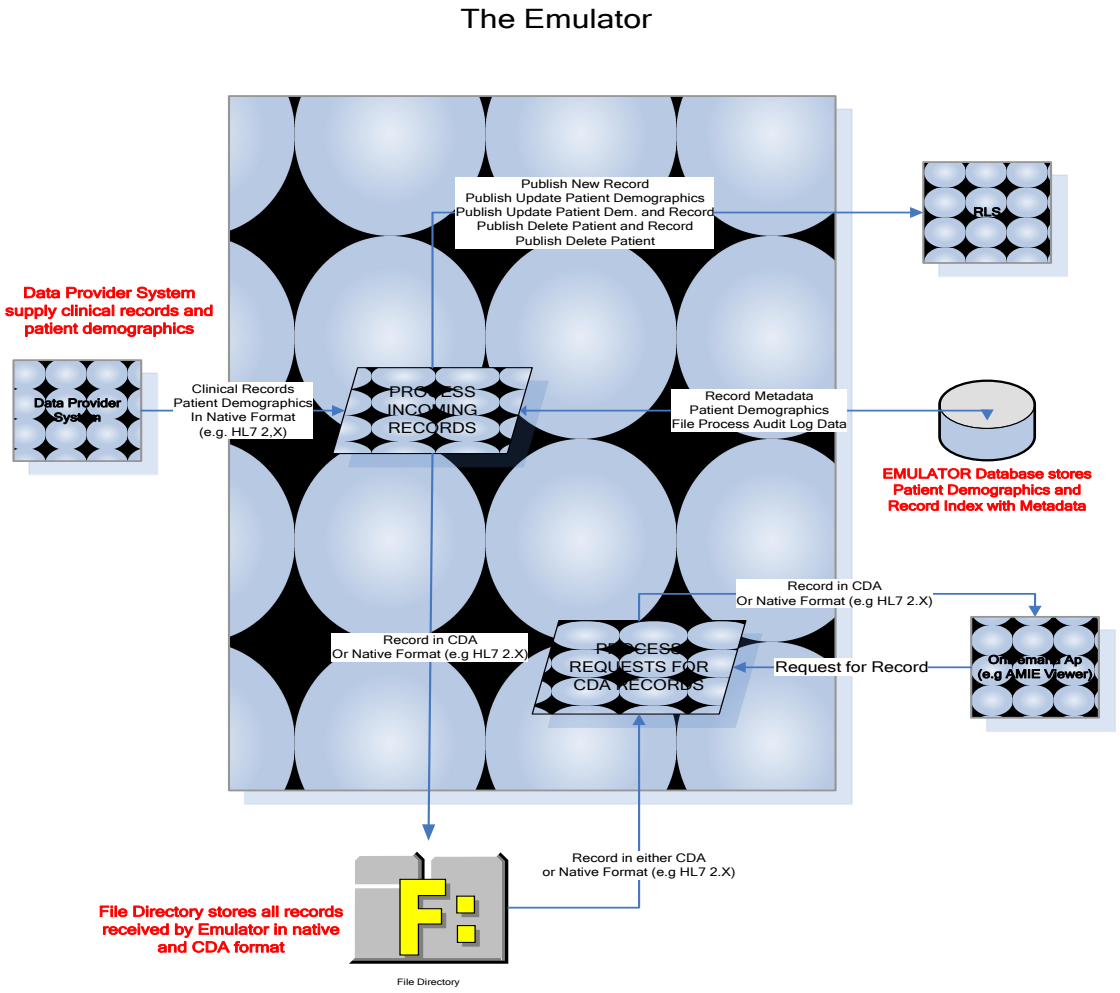
The Emulator is configured to accept large “bundles” of records over MLLP, FTP, and Web Services. It processes those “bundles” and publishes the availability of these records in the record locator service (RLS). When one of the records is requested by the either the AMIE viewer or other On Demand application, the emulator will process that request and return the record through the Gateways to the requesting application.

The following diagram illustrates how the Emulator fits within the AMIE architecture.



Emulator Design

The following context diagram illustrates the emulator system interfaces and core functionality.



The emulator interfaces with three systems either though CDX gateways, data provider interface engines or directly as described and shown in the AMIE architecture diagram above.

1. Record Locator Service (RLS)
2. Data Provider System (e.g. hospital information system)
3. On Demand Application (e.g AMIE viewer or other)

The emulator provides the following core functionality:

1. Publishes availability of records and patient information to RLS on behalf of the data provider's source system(s).
2. Maintains current patient demographics on behalf of the data provider's source systems.
3. Transforms standard (e.g. HL7 2.X) or custom formatted data into CDA and stores it onto a database

4. Responds to retrieve record requests from other systems (e.g. AHCCCS viewer) via the CDX Gateways.

Emulator Supported Transmission Protocols

Emulator can send and receive data from source systems using one or all of the following protocols:

1. Local folder location - Data provider should define their local directory structure and notify the same to AHCCCS). The local directory structure must have an incoming records folder where clinical records are placed to process.
2. FTP – Data provider should specify their URL for the In folder location along with the authentication credentials required to access their network.
3. SMTP – Data provider should provide the following information required for the Emulator to deliver secure emails to the data provider:
    - SMTP Server Name.
    - Destination email address.
4. MLLP

Emulator Error Notification Mechanism

If the clinical records sent by the data provider source systems fail to process at the Emulator, the emulator may notify the data provider by using one of the following mechanisms:

1. Emulator drops error files into an error folder as specified by data provider in local folder location.
2. Emulator transfers error files into an error folder in a remote location.

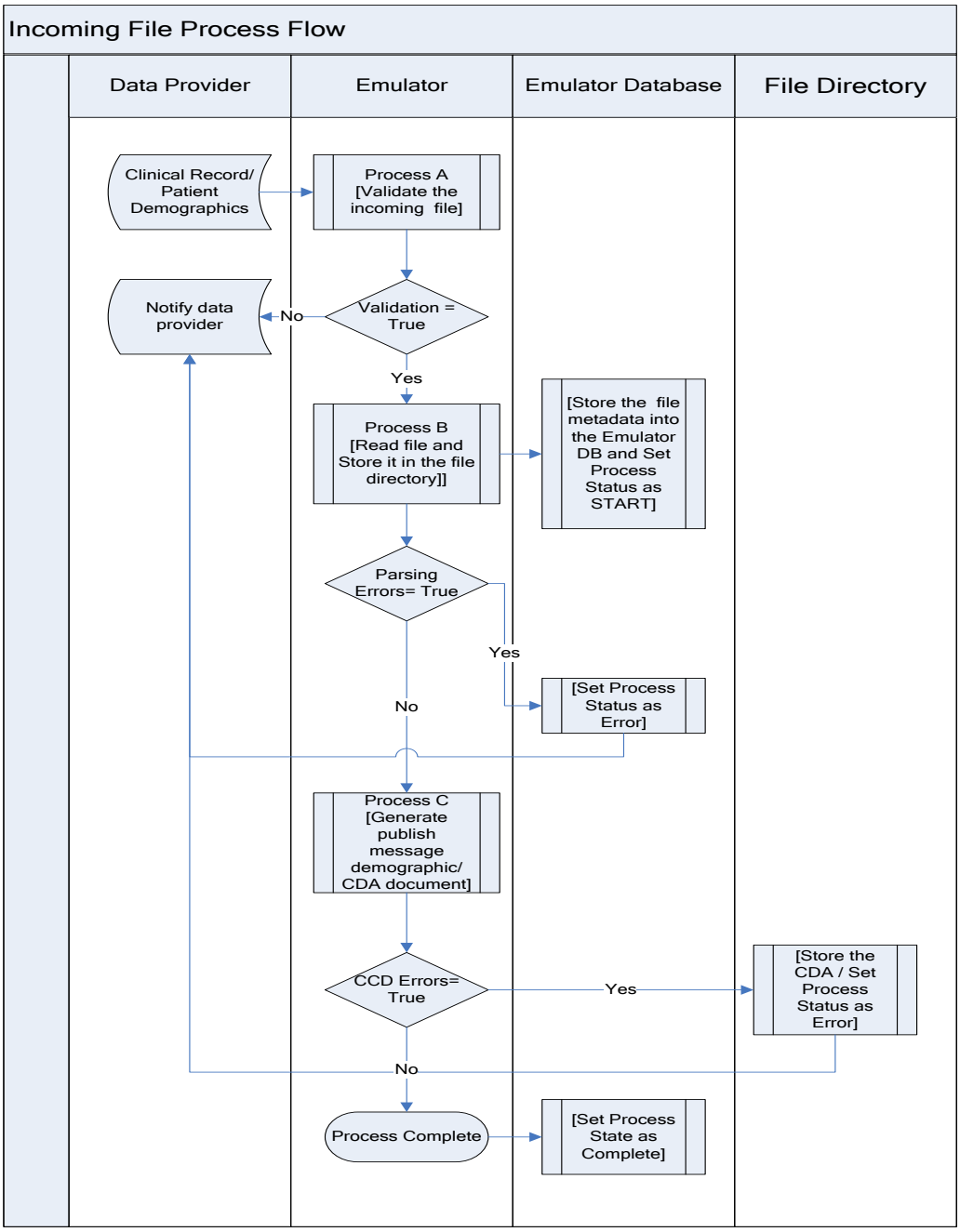
Emulator Incoming file process flow

1. Emulator receives data file from data provider and does the following in Process A:
    a. Validate the file using the BizTalk HL7 Accelerator 2.x
    b. If the message fails to process, Emulator notifies the sender.

2. Emulator continues to process the file and does the following in Process B:
    a. Store the file metadata into the Emulator database.
    b. Set process status to START in audit log.
    c. Process the file and store in configured file directory.
    d. Log the Error message if the file fails for any parsing errors.
    e. If the file fails to process, Emulator notifies the sender.

3. Emulator continues to process the file and does the following in Process C:
    a. Generates appropriate publish message
    b. Generates CDA document
    c. Stores CDA document in pre-configured file directory
    d. Set process status to SUCCESSFUL in audit log database upon successful processing.
    e. If the file fails to process, Emulator logs the CDA errors in audit log and notifies the sender.

AHCCCS


Incoming File Process Flow
Data Provider
Emulator
Emulator Database
File Directory
Clinical Record/ Patient Demographics
Process A [Validate the incoming file]
Notify data provider
No
Validation = True
Yes
Process B [Read file and Store it in the file directory]]
[Store the file metadata into the Emulator DB and Set Process Status as START]
Parsing Errors= True
Yes
No
[Set Process Status as Error]
Process C [Generate publish message demographic/ CDA document]
CCD Errors= True
Yes
[Store the CDA / Set Process Status as Error]
No
Process Complete
[Set Process State as Complete]

AHCCCS

Emulator Database Design

The Emulator database contains all the metadata required to locate a CDA document. The emulator database stores patient demographic information and patient record metadata to facilitate retrieval of records upon request. The Patient Master stores patient demographic information. The Encounter Master stores record metadata.

Database Tables

Encounter Master

The purpose of this table is store available records metadata.

EncounterMaster	
PK	**RecordID**
	CreateDate Published EncounterID DateOfService RecordType Descriptor LastUpdate EncounterHL7 TranslatedCCD
FK1	**PatientMasterID**

Data Element Attributes

RecordID
Integer data type
Required field
Attribute in Primary Key

PatientMasterID
Integer data type
Required field
Foreign Key from Patient Master and attribute in Primary Key

CreateDate
Defaults to the current date/time when the entry is first created
Date/Time field
Required field

Published
Bit data field defaults to 0
Refers to data which is published or not yet published
Required field

IncomingHL7 (This data element is obsolete due to change in Emulator Architecture)
Stores incoming HL7 message passed to emulator

Varchar(max) data type
Required field

TranslatedCDA (This data element is obsolete due to change in Emulator Architecture)
Stores translated HL7 message in CDA format
Varchar(max) data type
Required field

Patient Master

The purpose of this table is to store the patient demographics.

PatientMaster	
PK	**PatientMasterID**
	SourceSystemID **MRN** AHCCCSID SSN DriverLicenseNumber LastName FirstName MotherMaidenName MiddleInitial AddressLine1 AddressLine2 City State Zip CountryCode Gender DOB Race HomePhone OfficePhone BirthPlace EthnicityCode ReligiousAffilicationCode LaguageCode MaritalStatusCode PatientImportanceCode Citizenship Nationality PatientDeathIndicator Published Status Active CreateDate LastUpdate DemographicsHL7

AHCCCS

Data Element Attributes

PatientMasterID
Automatically assigned by the system
Integer data type
Required field
Primary key

SourceSystemID
Varchar data type
Required field
Unique for given Data Provider

MRN
Varchar data type
Required field
It’s a patient number coming from SourceSystem

AHCCCSID
Varchar data type
Optional field

SSN
Varchar data type
Optional field

DriverLicenseNumber
Varchar data type
Optional field

LastName
Varchar data type
Optional field

FirstName
Varchar data type
Optional field

MotherMaidenName
Varchar data type
Optional field

MiddleInitial
Char data type
Optional field

AddressLine1
Varchar data type
Optional field

AddressLine2
Varchar data type
Optional field

City

Varchar data type
Optional field

State
Varchar data type
Optional field

Zip
Varchar data type
Optional field

CountryCode
Varchar field
Optional field

Gender
Char field
Optional field

DOB
Smalldatatime field
Optional field

Race
Varchar field
Optional field

HomePhone
Varchar field
Optional field

OfficePhone
Varchar field
Optional field

BirthPlace
Varchar field
Optional field

EthnicityCode
Varchar field
Optional field

ReligiousAffilicationCode
Varchar field
Optional field

LaguageCode
Varchar field
Optional field

MaritalStatusCode
Varchar field
Optional field

PatientImportanceCode
Varchar field
Optional field

Citizenship
Varchar field
Optional field

Nationality
Varchar field
Optional field

PatientDeathIndicator
Varchar field
Optional field

Published
Bit data type
Optional field (default value: 0)

Active
Bit data type
Optional field (default value: 1)

Status
Char data type
Optional field (permissible values: N, U or D)

CreateDate
Date/Time field
Optional field

UpdateDate
Date/Time field
Optional field

DemographicsHL7 (This field has become Obsolete due to change in Emulator architecture)
Stores HL7 message containing patient demographics
Varchar(max) data type
Optional field

DataProvider

This table has information about Data Providers.

DataProvider	
PK	**DataProviderID**
	DataProviderName Template

Data Element Attributes

DataProviderID
Automatically assigned by the system
Integer data type
Required field
Primary key

DataProviderName
Varchar data type
Optional
Template
XML data type
Optional

EmulatorProcessLog

This table stores audit log information for Emulator. There is an entry for each file processed by the emulator.

EmulatorProcessLog	
PK	**ProcessLogID**
	MessageType CBOxml ProcessStatus ErrorDetails **ProcessDate** **UpdateDate**

Attributes

ProcessLogID
Automatically assigned by the system
Integer data type
Required field
Primary key

MessageType
Varchar data type
Optional
CBOxml
Varchar(max) data type
Optional

ProcessStatus
Char data type
Optional

ErrorDetails
Varchar data type

Optional

ProcessDate
Defaults to the current date/time
Date/Time field
Required field

UpdateDate
Defaults to the current date/time
Date/Time field
Required field

Relationships

Encounter Master object is a master object and has no parent relationship. It will have one or more **Patient Master** objects.

AHCCCS

Stored Procedures

usp_CheckStatus

Input parameters:

```
SourceSystemID,
            MRN,
            AHCCCSID,
            SSN,
            DriverLicenseNumber,
            LastName,
            FirstName,
            MotherMaidenName,
            MiddleInitial,
            AddressLine1,
            AddressLine2,
            City,
            State,
            Zip,
            CountryCode,
            Gender,
            DOB,
            Race,
            HomePhone,
            OfficePhone,
            BirthPlace,
            EthnicityCode,
            ReligiousAffilicationCode,
            LaguageCode,
            MaritalStatusCode,
            PatientImportanceCode,
            Citizenship,
            Nationality,
            PatientDeathIndicator,
            DemographicsHL7,
            EncounterID,
            DateOfService,
            RecordType,
            Descriptor,
            EncounterHL7,
            ProcessLogID
```

Result: Depending on status it stores Patient demographics information and also Encounter information and sends a flag back to Emulator about publishing record to RLS.

usp_PatientMaster_Get

Input parameters:

```
            SourceSystemID,
```

```
MRN
```

Result: Returns patient demographics details from PatientMaster table.

usp_EncounterMaster_Get

Input parameters:

```
SourceSystemID,
MRN
```

Result: This stored procedure gets PatientMasterID from PatientMaster table based on SourceSystemID and MRN. Then it returns all columns from EncounterMaster table.

usp_EncounterMaster_GetCDA

(This procedure is not used in the current Emulator architecture)

Input parameters:

```
RecordID
```

Result: This stored procedure returns a CDA message for earlier inserted HL7 based on RecordID from EncounterMaster table.

usp_DataProvider_Get

Input parameters:

```
DataProviderName
```

Result: Returns the XML template for a given Data Provider

9. usp_DataProvider_Store

Input parameters:

```
DataProviderName,
Template
```

Result: Stores the XML template for a given Data Provider

usp_ProcessLog_Status

Input parameters:

```
ProcessID int,
ErrorDetails,
CBOxml
```

Result: Stores the information about Emulator error logs

usp_ProcessLog_Start

Input parameters:

```
MessageType
```

Result: Stores the startup information about Emulator into EmulatorProcessLog table

Emulator Audit Log Table: Used for logging the process information

Column Name	Data Type	Allow Nulls
ProcessLogID	int	☐
MessageType	varchar(50)	☑
CBOxml	varchar(MAX)	☑
ProcessStatus	char(1)	☑
ErrorDetails	varchar(500)	☑
ProcessDate	datetime	☐
UpdateDate	datetime	☐
		☐

Data dictionary:

ProcesslogID:
Used to identify each file
Message Type:
File type specify if it is a CSV or ADT or MDM or ORU
CBOxml:
Actual message that is stored while the file failed.
Process status:
S – Started
C – Completed
E – Error
Error Details:
Reason for the Failure
Process Date:
File process date
Update Date:
Update date

AHCCCS

Emulator Directory

The Emulator file directory is the location that stores the following types of files:

1. Original patient demographic files (e.g HL7 2.X ADT files, .CSV files)
2. Original record files (e.g. HL7 2.X ORU, HL7 2.X MDM files)
3. Transformed CDA documents

The rest of the information is stored in the Emulator database. The root location for this directory structure is configurable.

All files in this directory as stored as compressed files. The emulator compresses the file when storing the files and decompresses the files when responding to requests for the files.

Folder and File Naming Convention

All top folders within the root directory are named using the following naming conventions:

1. The folders contain patient demographic must be named as follows:

    - **PatientDemographics{Data Format},** where Data Format is the format of the patient demographics files (e.g. Custom, HL7).

    - For example, if patient demographics are in HL7 2.X ADT format, then the folder name would be “PatientDemographicsHL7”.

2. The patient demographic files must be named as follows:

    - **PatientDem{Data Format}_{PatientID}.gz**, where Data Format is the format of the patient demographics files (e.g. Custom, HL7) and patient id in the Emulator database padded with zeros on the left for a fixed length of 9 digits.

    - For example, if patient demographics file is in HL7 2.X ADT format and its record id is 123456, then the filename would be “PatientDemHL7_000123456.gz”.

3. The folders that contain original records must be named as follows:

    - **Encounter{Data Format},** where Data Format is the format of the original record file (e.g. Custom, HL7, NCPCP).

    - For example, if patient clinical records are in HL7 2.X ORU format, then the folder name would be “EncounterHL7”.

4. The original record files must be named as follows:

    - **E{Data Format}_{RecordID}.gz**, where Data Format is the format of the clinical record files (e.g. Custom, HL7) and RecordID is the record id in the Emulator database padded with zeros on the left for a fixed length of 9 digits.

    - For example, if clinical record file is in HL7 2.X ORU format and its record id is 123456, then the filename should be “EHL7_000123456.gz”.

AHCCCS

5. The folder and files that contains patient translated CDA documents must be named as follows:

    - TranslatedCCD

6. The CDA document files must be named as follows:

    - **TransCCD_{RecordID}.gz**, where RecordID is the record id in the Emulator database padded with zeros on the left for a fixed length of 9 digits.

    - For example, if clinical record file is in HL7 2.X ORU format and its record id is 123456, then the filename should be “TransCCD_000123456.gz”.

Folder Directory Structure

In order to optimize directory performance, all files are stored on a multilevel directory structure. The 9 digit identification on the filenames is broken down into three 3-digit strings to create a three level directory structure down from the root directory.

- The first level is defined by the top folders for each of the file types (e.g TransCCD, EncounterHL7, PatientDemographicsHL7)
- The second level is defined by the folders named after the leftmost 3-digit string.
- The third level is defined by the folders named after the middle 3-digit string.

Emulator stores the file in the appropriate folder based on file type and 9-digit identification (e.g. record id, patient id).

All files are stored in a directory structure that is defined by the 9-digit identification (e.g record id or patient id) contained within the filename.

For example in order to store or access the a filed named **TransCCD_000123456.gz**, the following path must be used,

DIR_ROOT://TranslatedCCD/000/123/TransCCD_000123456.gz