Enhanced Hospital Safety

Bluebird

HL7 Generator

Table of Contents

Introduction

The Bluebird HL7 generator is a lightweight application that resides in your datacenter and enables your hospital to easily create and securely send HL7 messages to Bluebird. Many hospitals have existing admission, discharge and transfer (**ADT**) solutions as well as pharmacy dispensing solutions. In order for Bluebird to provide value to your hospital Bluebird requires secure access to that data. For example, infection rates are standardized based upon the number of admissions or the number of days a patient resides in hospital and it is important to be able to specify infection rates by specific locations (wards/units) in each hospital. **Pharmacy** data is important for the Bluebird Medication Administration Record (MAR) as well as for Antibiotic Stewardship.

The standard way healthcare organizations send information to one another is via a language called HL7. While Bluebird publishes the HL7 format and transport required, many Hospital Information Systems (HIS) are unable to produce standardized HL7 feeds and many hospitals do not have the IT staff or the budget to employ 3rd party IT consultants to modify their HIS in order to generate those HL7 feeds. In order to solve this problem the software engineers at Bluebird developed a lightweight application that sits in the hospital datacenter, interrogates easily set up **Views** of the HIS database, creates the required HL7 data feed and sends that data in a secure pipe to Bluebird. In order to make use of the Bluebird HL7 generator, your hospital IT staff simply need to create five **Database Views**, install the Bluebird HL7 generator application and open a port in the firewall.

This white paper explains the minimal, once off, Set Up required by your hospital IT.

What is Bluebird

Bluebird is an advanced electronic health record (EHR) specifically designed to **reduce risk** and **optimize inpatient safety**.

Bluebird's many risk attenuation modules include safe medication administration, fall prevention, pressure sore prevention, surgical management (SURPAS) as well as an early warning system that alerts clinicians (in real time) to patient deterioration. Infection control is considered a **sentinel indicator** of patient safety and the Bluebird Infection Control and Antimicrobial Stewardship module has proven extremely successful over the last decade in helping more than 70 Southern African hospitals measure and proactively manage hospital infections and improve antibiotic use. For more detail please see: intelms.com/177

Views

The Bluebird HL7 app requires read-only access to specific data (patient demographics, admission, transfer, discharge and drug dispensing information) in the Hospital Information System (HIS) database. In order to facilitate this and to protect the HIS database five views should be created to give access to this information. The Bluebird HL7 app can then be given read-only access to these views only so that Bluebird has access only to the restricted data it needs (instead of all of the data in the HIS database).

Definition: A **database view** is a virtual table that is based on the result of a SELECT statement. It does not store data itself, but instead provides a way to access data from one or more tables in a specific way. A view can be thought of as a saved SELECT statement that can be referred to by name, just like a table. Views can be used to provide a specific perspective on the data, to restrict access to certain columns or rows, or to simplify the structure of the underlying tables. Views can also be used to join data from multiple tables and to aggregate data, such as calculating a sum or an average. It's important to note that views are **read-only by default** and any changes made through the view will not be persisted in the underlying tables.

In order to make use of the Bluebird HL7 Generator, the hospital IT staff simply need to do the following:

- Create a Bluebird database user and five views

- Set up an ODBC DSN on a Windows or Linux server

- Install the Bluebird HL7 Generator app on the server

- Configure the Bluebird HL7 Generator app

- Alter the firewall rules to allow connections from the HL7 app to https://zeus.intelms.com/ (IP 41.223.32.218, TCP port 443)

The five required views are:

1. bb_demographics

2. bb_adt

3. bb_diagnosis

4. bb_dispensing

5. bb_merges

The field names and descriptions for each of these 5 views are:

1. bb_demographics

This view should contain a list of patient **demographic** transactions.

Each row will contain the patient's demographics. Any time there is any addition or change to the patient demographics, there should be a new record in this view.

There must be an integer primary key that always increases so that it's possible to keep track of the last record processed. There must never be changes made to any existing records. Any changes must only ever be reflected in new records.

Fields

- **id:** Monotonically increasing integer
- **event_at:** The date/time that this transaction happened. yyyy-mm-dd HH:MM:SS
- **recorded_at**: The date/time that the transaction was recorded in the HIS: yyyy-mm-dd HH:MM:SS
- **transaction:** A string indicating the type of transaction. New patient, Update
- **patient_number:** A string containing the unique identifier for the patient across the whole hospital group
- **first_name:** The patient's first name(s)
- **last_name:** The patient's last name/surname
- **date_of_birth:** The patient's date of birth (if known), in the format yyyy-mm-dd. If it is unknown, this should be NULL
- **sex:** The patient's sex. One of "male", "female", "unknown" or "other"
- **address1**: First line of patient's home address
- **address2**: Second line of patient's home address

- **city**: The city/town of the patient's home address
- **province**: The province/state of the patient's home address
- **country**: The ISO 3166-1 alpha-2 (2 letter) country code of the patient's home address
- **home_phone**: A string containing the patient's home phone number in International format. e.g. +27 72 123 4567
- **home_email**: The patient's home email address, if available
- **work_phone**: A string containing the patient's work phone number in International format, if available. e.g. +27 72 123 4567
- **work_email**: The patient's work email address, if available
- **id_number**: A string containing the patient's unique national ID number
- **deceased_at**: The date and time of the patient's death, if deceased. yyyy-mm-dd HH:MM:SS
- **vip**: This is a boolean value. True if the patient is a VIP patient. False otherwise. If this data is unavailable, this should be NULL
- **other_patient_number**: If two patient records have been merged in the HIS, then this field should contain the unique patient identifier of the patient record that was merged into this one. i.e. the "patient_number" field contains the canonical patient identifier. The "other_patient_number" field contains the patient identifier of the merged patient
- **ma_administrator**: Patient's medical aid administrator
- **ma_scheme**: Patient's medical aid scheme
- **ma_option**: Patient's medical aid option
- **ma_main_member**: The name of the main member of the medical aid
- **ma_address1**: The first line of the main member's home address
- **ma_address2**: The second line of the main member's home address

- **ma_city**: The city/town of the main member's home address
- **ma_province**: The province of the main member's home address
- **ma_country**: The ISO 3166-1 alpha-2 (2 letter) country code of the main member's home address
- **ma_dependent_no**: The dependent number
- **ma_member_no**: The patient's medical aid member number
- **ma_id_number**: The unique national ID number of the main member
- **ma_employer**: The name of the employer of the main member
- **ma_home_phone**: The home phone number of the main member in International format. e.g. +27 72 123 4567
- **ma_home_email**: The home email address of the main member, if available
- **ma_work_phone**: The work phone number of the main member in International format, if available. e.g. +27 72 123 4567
- **ma_work_email**: The work email address of the main member, if available

2. bb_adt

This view should contain a list of **ADT** (Admission, Discharge, Transfer) transactions.

Each row will contain the details of the patient's current location and, in the case of transfers, the patient's prior location. Any time the patient is registered in ER or similar location, admitted, transferred, discharged, or an admission or transfer or discharge is cancelled, there should be a new record in this view.

There must be an integer primary key that always increases so that it's possible to keep track of the last record processed. There must never be changes made to any existing records. Any changes must only ever be reflected in new records.

Fields

- **id**: Monotonically increasing integer
- **event_at:** The date/time that this transaction happened. yyyy-mm-dd HH:MM:SS
- **recorded_at**: The date/time that the transaction was recorded in the HIS: yyyy-mm-dd HH:MM:SS
- **transaction**: A string indicating the type of transaction. Admission, Transfer, Discharge, Cancel Admission, Cancel Transfer, Cancel Discharge
- **visit_number**: A string containing an identifier for the patient that is unique for every case/visit.
- **patient_status**: This is "I" for an inpatient, "O" for an outpatient, "E" for an emergency patient
- **current_ward**: The code of the patient's current ward. This is the unique code for this ward in the HIS
- **current_room**: The code for the room that the patient is currently in. This is the unique code for this room in the HIS
- **current_bed**: The code for the bed that the patient is currently in. This is the unique code for this bed in the HIS
- **prior_ward**: The code for the ward that the patient was transferred from. This is the unique code for this ward in the HIS
- **prior_room**: The code for the room that the patient was transferred from. This is the unique code for this room in the HIS

- **prior_bed**: The code for the bed that the patient was transferred from. This is the unique code for this bed in the HIS

- **attending_doctor_code**: A string containing the unique registration number for the admitting doctor. e.g. "mp0123456"

- **attending_doctor_first_name**: The attending doctor's first name(s)

- **attending_doctor_last_name**: The attending doctor's last name/surname

- **referring_doctor_code**: A string containing the unique registration number for the referring doctor

- **referring_doctor_first_name**: The referring doctor's first name(s)

- **referring_doctor_last_name**: The referring doctor's last name/surname

- **consulting_doctor_code**: A string containing the unique registration number for the consulting doctor

- **consulting_doctor_first_name**: The consulting doctor's first name(s)

- **consulting_doctor_last_name**: The consulting doctor's last name/surname

- **admitted_at**: date/time of the patient's admission. yyyy-mm-dd HH:MM:SS

- **discharged_at**: date/time of the patient's discharge. yyyy-mm-dd HH:MM:SS

3. bb_diagnosis

This view should contain the admitting diagnosis or diagnoses for each patient visit/case. Each row will contain the details of one admitting diagnosis associated with a patient visit/case. Each row must contain a reference to the corresponding record in the bb_adt view, using the bb_adt view's "id" field.

Fields

- **bb_adt_id**: The ID of the corresponding record in the bb_adt view
- **icd10_code**: The ICD-10 code of the diagnosis
- **description**: The description of the diagnosis

4. bb_dispensing

This view should contain a list of **drug** transactions. Each row will contain the details of a drug that was dispensed for a patient. Any time a drug is dispensed for the patient there should be a new record in this view.

There must be an integer primary key that always increases so that it's possible to keep track of the last record processed. There must never be changes made to any existing records. Any changes must only ever be reflected in new records.

Fields

- **id**: Monotonically increasing integer
- **event_at:** The date/time that this drug was dispensed: yyyy-mm-dd HH:MM:SS
- **ordered_at**: The date/time that that the drug was ordered: yyyy-mm-dd HH:MM:SS
- **placer_order_number**: A string containing the number/code identifying the drug order
- **drug_name**: The name of the drug
- **drug_nappi_code**: A string containing the NAPPI code for the drug
- **drug_atc_code**: The ATC code for the drug
- **dose**: The dose amount
- **units**: A string containing the dose units. mg, g, ug or mcg, ml, L, cc, mol, mmol
- **administration_instructions**: The instructions for how the drug should be administered
- **ward**: The code of the ward that the drug will be delivered to. This is the unique code for this ward in the HIS
- **dispense_amount**: The amount of the drug that was dispensed

- **dispense_units**: A string containing the units for the dispensed amount. mg, g, ug or mcg, ml, L, cc, mol, mmol
- **ordering_doctor_code**: A string containing the unique registration number for the ordering doctor
- **ordering_doctor_first_name**: The first name(s) of the ordering doctor
- **ordering_doctor_last_name**: The last name/surname of the ordering doctor
- **pharmacist_code**: A string containing the unique registration number of the pharmacist
- **pharmacist_first_name**: The pharmacist's first name(s)
- **pharmacist_last_name**: The pharmacist's last name/surname
- **prescription_number**: The prescription number for this drug
- **patient_number**: The unique identifier for the patient across all hospitals in the group
- **case_number**: The identifier for the patient's visit/case

5. bb_merges

This view should contain a list of records recording patient records that have been merged. Each row will contain the the details of one merge. Any time patient records are merged in the HIS there should be a new record in this view.

There must be an integer primary key that always increases so that it's possible to keep track of the last record processed. There must never be changes made to any existing records. Any changes must only ever be reflected in new records.

Fields

- **id:** Monotonically increasing integer
- **event_at:** The date/time that this merge happened. yyyy-mm-dd HH:MM:SS
- **patient_number:** The patient number of the patient record that the other record was merged into. i.e. this is the correct patient number
- **other_patient_number:** The patient number for the record that was merged

Installation

Install the Bluebird HL7 Generator app on the server by double clicking the installation package and following the prompts

Configuration

The app comes with a configuration file that specifies the ODBC data source to use to connect to the database and also some information identifying the hospital group and hospital.

The app config will be installed to

C:\Program Files\Bluebird\HL7\bbhl7.conf on **Windows**, or

/opt/Bluebird-HL7/etc/bbhl7.conf on **Linux**.

Bluebird Support will supply the group code, facility code and token.

Sample config file:

```
[main]
data_source = "ODBC datasource name"
group_code = "ABC"
facility_code = "H999"
token = “xb3J9ah3Z"
```

ODBC DSN

A system ODBC DSN will need to be created to allow the Bluebird HL7 Generator to connect to the database in order to access the views. The name of the DSN must match the “data_source” name in the config file. The credentials to access the database must also be saved in the ODBC DSN.

Windows

On Windows, this will be done through the ODBC Data Source Administrator as described here:

https://support.microsoft.com/en-us/office/administer-odbc-data-sources-b19f856b-5b9b-48c9-8b93-07484bfab5a7

Linux

On Linux, it is necessary to install the Unix ODBC package and configure it as per the Unix ODBC documentation.

The drivers will need to be defined in /etc/odbcinst.ini, e.g.:

```
[IRIS]

Driver = /opt/IRIS/bin/libirisodbcur6435.so

Description = IRIS ODBC driver
```

The ODBC DSN will need to be defined in /etc/odbc.ini, (the options will depend on the type of database) e.g.:

```
[ODBC_DSN_NAME]

Driver = IRIS

Description = Bluebird HL7 Generator access
```

```
Host = 192.168.99.123
Port = 1972
Namespace = TheNamespace
UID = Bluebird
Password = ThePassword
Protocol = TCP
Query Timeout = 1
Static Cursors = 0
Trace = off
TraceFile = iodbctrace.log
Authentication Method = 0
Security Level = 2
Service Principal Name = iris/localhost.domain.com
```