



HL7 Generator Tool Developer Guide

Author: John Park

Version 1.0

Revision Log

Version	Changes
1.0	Initial version



Table of Contents

Description of Tabs in the HL7 Generator Tool	3
Message processing overview	4
Admin tab configuration reference	5
Performance considerations and parallel processing	11

Table of Tables

Table 1: General Tool Configuration Items	5
Table 2: HL7 Mapping Types	8
Table 3: Configuration Items.....	9
Table 4: Validation Items	10
Table 5: Configuration Detail Items	11



This tool is built into Microsoft Excel and has various tabs, some of which are for data input and some of which are for configuration data.

It has been tested in Windows 10 and Windows 7 environments with Excel version 16 and Office 365. Otherwise, run in a virtual windows environment.

Description of Tabs in the HL7 Generator Tool

Instructions Tab: Contains overall instructions and buttons to execute the HL7 batch message generation.

Gap_Analysis_7-8 Tab: Contains the overall data guidance/constraints provided by CDC and HHS on 7/8/2020; it also defines the data elements for the flatfile and how they are mapping into an HL7 message. It can be used to assess, if the lab has all required data elements and to document where each elements is drawn from in their system.

Flatfile_Result_Use Tab: Data entry tab which allows the user to manually enter data into any of the fields. If data in fields required to generate an HL7 v.2.5.1 compliant message is missing the tool will provide an error report upon execution of the conversion. This provides the file layout and column headers that the imported file must contain – the tab itself is not used when using csv file input option.

Configuration Tab: Contains two parts that determine data mapping for each message generated by the tool.

1. Static values to be included in specific data fields for all messages generated
 - a. If Column E – “Corresponding input column name (to replace if empty)” is populated, the tool will only populate this entry if the corresponding input column is empty.
2. References that map valid values from the *dropDownValues* tab to the HL7 replacement value for each column configured. The dropdowns are manually configured to the data input fields on the *Flatfile_Result_Use* tab using data validation.

dropDownValues Tab: Defines specific value sets that are valid for certain HL7 elements and fields. Referenced by *Configuration* tab and *Flatfile_Result_Use* tab and allows for mapping local codes used in the csv file that is being imported to the standard codes to be included in the HL7 messages. If the lookup of the local value fails, it will then check to see if the input is in Column D (Standard code). If it matches the value in Column D, it will map the replacement value to the HL7 message for that field.

Admin Tab: Defines overall flat file configuration (input/output directories for processing, flatfile naming parameters) as well as validations and lookups to be used for various fields. The



admin tab also defines how the flatfile field column maps to the HL7 message and what functionality to invoke for each data element.

Message processing overview

The HL7 Generator consists of the input and configuration tabs and two Macro functions or “Subs”. They are the “LoadCSV” sub and the “GenerateHL7” sub.

The “LoadCSV” sub can be called by pushing the “Process CSV and generate HL7” button on the instruction tab. It can also be called using CScript.exe (built into windows) to call the macro via the vbscript “macroLauncher.vbs”. It can then be run as a scheduled task. Instructions to do this are in the document “HL7generator Instruction_CSV Input.docx.” The “LoadCSV” Sub does the following (see Table 1: General Tool Configuration Items for details on where to locate each of the items):

1. It opens the CSV file located in the configured Input directory and named per the configuration “Input file name” on the *Admin* tab.
2. If there are no errors on load, it will copy the data to a temporary sheet within the Excel Macro and call the Sub “GenerateHL7”. If it cannot open the file it will create an error log file in the error directory and stop processing.
3. The Sub “GenerateHL7” will load all the data from the temporary sheet into memory, and if there are no errors, it will rename the input file with “prc” and the timestamp pre-pended to the input name and move it to the “Processed” directory. The temporary sheet is deleted. Note that there may be errors processing the data to generate the HL7 after this that will prevent HL7 batch creation.
4. At this point it will process the input in the Sub “GenerateHL7” the same way it processes the data if it were invoked pushing the “Generate HL7 from Manual Entry” button.

Once CSV to HL7 processing starts, it does the following:

1. The GenerateHL7 sub loops through each row, stores each row in memory, and captures the “Testing_lab_accession_number” and adds it to a list of Distinct HL7 records. This is because each row represents a test result and there can be multiple test results per HL7 message. They are all linked with the value in “Testing_lab_accession_number” as it is the primary key for each HL7 message. It also adds a list of row numbers that have this primary key into an internal lookup table.
2. If the “Testing_lab_accession_number” is empty, it will look in “Testing_lab_specimen_ID” and use that field as the primary key and add the list of rows with this key to the dictionary. If both fields are empty, the batch will error.
3. Each column is evaluated, and an internal reference is created for each column. The reference stores the column and maps the following values based on Columns D through H on the *Admin* tab.



- a. To which HL7 Segment it will map (MSH, PID, ORC, OBR, Test OBX, SPM or NTE) or whether it creates an entirely new OBX segment for Ask at Order Entry (AOE) questions. This is read from Column E (HL7 mapping).
 - b. What action to take whether it's to look up and insert a replacement value, pass through the value unchanged to the HL7, convert the field (date, county, zip, phone number, etc.), create a new OBX segment for AOE questions, or custom handling from Column F (Config).
 - c. What kind of validation needs to be completed: part of a dropdown list, required, one of a group of fields that require a value, etc. from Column G (Validation).
 - d. Certain configurations require additional configuration detail from Column H (Config Detail). For example this column holds the template OBX segment for AOE questions.
4. Every value is stored, and its action defined. For each distinct HL7, it will copy the template HL7 segments defined in Column K (Segment Template) and create a new HL7 message for each distinct HL7 key copied from the segment templates.
 5. It will loop through all the values stored and "string replace" the HL7 template element names with values from the stored input. For example, in the PID segment `_PID-11.5_` is replaced with "46203" from input. The underlines before and after were added within the code to avoid replacing part of a number (23.1 vs 23.10)
 6. Once all the columns for the rows that pertain to one distinct HL7 record is processed, it adds the newly populated segment templates to the Batch and creates new instances of the segment template to start a new HL7 message.
 7. If there are errors, they are logged to the error log file and output to the Error directory indicating that processing is finished with errors. It will not output the HL7 batch.
 8. If there are no errors the error log file will be output with the message: "HL7 batched produced with no errors". An HL7 file with InterPartner naming will be produced in the configured output directory configured: "InterPartner~CentralizedELR~<sender name>~<receiver name>~<test or prod>~<test or prod>~<timestamp>~STOP~<input file name>". For example:
"InterPartner~CentralizedELR~eTrueNorth~AIMSPlatform~PROD~PROD~20200814123715~STOP~NationalELR_csvToConvert.hl7"

[Admin tab configuration reference](#)

The below tables describe each configuration of the Admin tab.

Table 1: General Tool Configuration Items

Configuration Item	line	Sample Value	Description
Input Directory	2	C:\runMacro\Input	This is the directory on the system where input CSV files will be placed. Not required for manual entry. When placing files in this directory make sure the file has the same name configured in "Input file name" below.



Configuration Item	line	Sample Value	Description
Output Directory	3	C:\runMacro\Output	The directory that will output the generated HL7 batch.
Processed Directory	4	C:\runMacro\Processed	Once the CSV is loaded into the tool, the input CSV is moved to this directory and renamed with “prc” and timestamp pre-pended to the original name.
Error directory	5	C:\runMacro\Error	At the end of processing, the tool will output a log file indicating whether the process was successful or indicating what errors prevented processing.
Timer on	9	FALSE	For developer use. The Tool will create a log of timestamps at each point in processing (after each row, after each distinct HL7, beginning and end of processing) This is to measure performance and identify bottlenecks. The timestamps will be appended to the log in the Error directory.
Number of Errors before Exit	10	0	For debugging errors. For a large file, it may take many minutes to complete processing. The output error log file may have redundant errors for each row. If the number of errors specified is reached, it will exit processing thereby saving time in identifying and addressing errors that prevent generation of the HL7 batch. Enter “0” to log all errors. 100 is a good number to start debugging.
Current usage (and MSH-11 Mapping)	15	Prod	This indicates whether this is a test or production message. It configures “Test” or “Prod” on the output file name for InterPartner file naming. It also populates MSH-11 in each HL7 message with “T” for “Test” or “P” for “Prod”.
InterPartner “From” value	16	eTrueNorth	For InterPartner file naming of the HL7 output file. This configures the sender name. This value is provided by the InterPartner exchange partner you are working with.
InterPartner “To” Value	17	AIMSPlatform	For InterPartner file naming of the HL7 output file. This configures the receiver name. This value is provided by the InterPartner exchange partner you are working with.



Configuration Item	line	Sample Value	Description
Test result exclude from normal processing	18	Test_kit_EUA_ID, Test_kit_model_ID	These are columns that are not handled by normal processing as configured in columns D through H of the Admin Tab. Custom code has been written for these columns to support grouping of data and specific functionality specified in the Gap Analysis Tab. Do not change this configuration.
Multiple value separator	19	;	For certain inputs that can support multiple values (e.g Disease_symptoms), this specifies the separator. This is for columns that have a config value of "REPEAT_OBX_MAP_MULTIPLE" in Column E of the Admin tab.
Input file name	20	csvToConvert.csv	The name of the input file to be read by the "LoadCSV" sub for processing.
Debug mode	21	FALSE	If this is set to true, the out file in the Error directory will include verbose debugging statements. It should be set to "FALSE" for normal processing.
Time zone	22	ET	For columns that are configured with "CONVERT_TIMESTAMP" in Column F this value specifies the HL7 time zone offset to be appended to the end of the timestamp value. For example, it will take the HL7 timestamp value of "202007130000" and append the Eastern time offset value of "-0500": "20200713000000-0500"

Mapping configuration:

Columns D-H define how each value in the input CSV or Flatfile_Result_Use column is handled.

Flatfile_Column-Column D

This is a list of all column headers present in both the "Flatfile_Result_Use" tab for manual input and the csv file to be processed.

HL7 Mapping-Column E

The HL7 mapping type entries in the HL7 Mapping column define where the data should be mapped.



Table 2: HL7 Mapping Types

HL7 mapping type	Sample Value(s)	Description
Direct to HL7 Mapping	OBR-4.1 ORC-15 PID-11.3	The value maps directly to an HL7 element as defined in the segment templates in Column K (Segment Template). For example, if this configuration specifies “OBR-4.1” it replaces the text “_OBR-4.1_” in the segment template. Underscores are added in the code to avoid partial number mappings.
Ignore	IGNORE	This means that that value in this column will be ignored and not mapped to the HL7
AOE mapping	REPEAT_OBX_MAP	This value will trigger the addition of an AOE OBX segment as entered in Column H (Config Detail). A lookup replacement value will be found based on the range in the configuration tab for this column. The replacement value goes into the OBX-5, which has the column name specified in the template.
	REPEAT_OBX_NM REPEAT_OBX_ST REPEAT_OBX_DT	This value will trigger the creation of an AOE OBX segment as entered and directly pass the value. For OBX_DT, it will convert the date value to HL7 format.
	REPEAT_OBX_MAP_OTHER	This configuration has the same functionality as REPEAT_OBX_MAP. However, in addition, if the replacement value is not found in the lookup based on the configuration tab, it will insert “OTH^Other^NULLFL” in OBX 5.1, 5.2, and 5.3. Then it will place the column value in OBX-5.9
	REPEAT_OBX_MAP_MULTIPLE	This configuration has the same functionality as REPEAT_OBX_MAP. However, if there are multiple entries separated by the multiple value separator configured in cell B19, it will create an AOE OBX entry for each value entered.
Map to multiple segments	ORC-12.3/OBR-16.3	Some values map to multiple elements in the HL7. If CUSTOM_MULTIPLE_MAPPING is configured in Column F (Config), the HL7 elements are separated by “/”

Config-Column F

These are entries that define where the data should be mapped. The code searches for keywords such as “CUSTOM” or “REPLACE”. Also, multiple configurations can be supported by separating them with “|”



Table 3: Configuration Items

Config item	Sample Value(s)	Description
Replace element	REPLACE_NTE_ELEMENT REPLACE_TEST_OBX_ELEMENT REPLACE_ORC_ELEMENT	The value in the column is to be mapped to the HL7 element specified in Column E (HL7 mapping) “as is” without any lookup or conversion.
Lookup element	LOOKUP_ORC_ELEMENT LOOKUP_PID_ELEMENT LOOKUP_OBR_ELEMENT	The value in the column is mapped with a LOOKUP value. The actual values are configured in the <i>dropDownValues</i> tab and the configuration of the lookup is in the <i>Configuration</i> tab starting at row 47.
Convert	CONVERT_TIMESTAMP_OBR_ELEMENT CONVERT_PHONE	This takes the input value and converts it to the proper HL7 timestamp or phone number based on HL7 specifications
Map one element to multiple HL7 elements	CUSTOM_MULTIPLE_MAPPING	By entering this configuration, one input value can be mapped to two elements in different segments in the HL7 message
Handled in custom code	CUSTOM	This means that mapping and configuration is handled in custom code and the functionality is not configurable on the Admin tab.
Ask At Order Entry (AOE)	REPEAT_OBX_MAP REPEAT_OBX_YNU	The value in this column will be mapped to an AOE OBX segment. The template for this OBX segment is specified in Column H (ConfigDetail). REPEAT_OBX_MAP does a look up based on the column name as configured on the configuration tab. REPEAT_OBX_MAP has a value set of Yes, No, or Unknown that are replaced with the HL7 triplet values.
Manual entry or lookup of coded value	TRIPLET GROUPED	If an element is configured with “TRIPLET”, it will do a lookup of that value. If a lookup value is found, it generally includes the description and coding system, which is then mapped to the related fields (that are configured with GROUPED) and the data entered on input is ignored for description and code system. If there is no match, it passes the value assuming that it is a local code, the “GROUPED” items are mapped from input, and the code system is configured to be validated.



Config item	Sample Value(s)	Description
Patient ID assigner OID lookup	CUSTOM_PAIR	This is to support custom functionality for the patient ID assigner, which is populated from a dropdown and mapped to the HL7 segment configured in Column E (HL7 Mapping). The replacement value is the OID for the patient ID assigner which is found via lookup of patient id assigner on the <i>dropDownValues</i> tab. The lookup value is mapped to the HL7 element configured in Column H (ConfigDetail) for this entry.

Validation-Column G

These are entries that define how the data entered should be validated and/or replaced with some other value.

Table 4: Validation Items

Validation item	Sample Value(s)	Description
No validation	NONE	No validation is done for the data element configured
default empty timestamp	0000_IF_EMPTY	Some timestamp fields cannot be empty. If configured, it will replace the empty value with "0000" in the HL7 timestamp field.
Populate empty field from another column	COPY_FROM_<column name>	The gap analysis specifies that if certain fields are empty, the value should be copied from another field. This supports that requirement by copying the value from one field to another if the target column is empty.
Validate dropdown	VALIDATE_DROPDOWN	For entries that require a certain value set, the field is evaluated to see if it exists in a range of values of the dropdownValues tab configured in the Column H (ConfigDetail). If it does not, the message will error.
County and zip format	WARN_COUNTY_FORMAT VALIDATE_COUNTY_FORMAT WARN_ZIP_FORMAT VALIDATE_ZIP_FORMAT	This validates that the county code is 5 digits and numeric and that zip code is either 5 digits or 9 digits and numeric (with a dash). If it is set to WARN_COUNTY_FORMAT or WARN_ZIP_FORMAT, it will process the HL7 batch but leave a warning in the output file in the error directory. If it is set to VALIDATE_COUNTY_FORMAT or VALIDATE_ZIP_FORMAT, it will error and the HL7 batch will not be produced.



Validation item	Sample Value(s)	Description
Required combo	REQUIRED_COMBO_4 REQUIRED_COMBO_5	There are groupings of data elements that require that at least one of them is populated. Fields with REQUIRED_COMBO configured that have the same number appended will be evaluated to make sure one of them is populated, and if not, it will error and not process the HL7 batch
Time stamp if empty	TIMESTAMP_IF_EMPTY	If this field, which expects a timestamp value is empty, it will use the current time function to insert current time into the HL7 message for this field.

ConfigDetail-Column H

For certain columns additional configuration is required. Most columns do not require this column.

Table 5: Configuration Detail Items

Config item	Sample Value(s)	Description
AOE OBX segment template	OBX OBXIDX CWE 75325-1^Symptom^LN _Disease_symptoms_ F _OBX-15.1_^_OBX-15.2_^CLIA _OBX-23_ _OBX-24_ QST	For each distinct HL7 entry and for each ask on order question, an OBX segment is created. The OBXIDX is replaced with an incrementor. OBX 5 is replaced based on the configuration of the column name and OBX 23 and OBX 24 are populated from test result information.
Patient age	Patient_age	This configuration ties Patient_age_units to Patient_age in the custom code for this specific AOE
Patient ID assigner oid	_PID-3.4.2-2_ _PID-3.4.2-1_	This configuration supports the CUSTOM_PAIR configuration. The lookup value returned is an OID that should be placed in this configured field in the HL7 PID segment template.

Segment Template-Column K

The CSV to HL7 converter takes these configured segment templates and does a string search and replace based on the configuration of the *Admin* tab. These segment templates can be modified if the specifications change or if structural issues need to be addressed.

Performance considerations and parallel processing

The CSV to HL7 Generator was benchmarked to process 2000 records in about 2 minutes on a computer with a Core i7 2.6Ghz processor and 16 GB of RAM. Performance will vary based on processor and



number of rows per distinct HL7 record. Macros have a limitation where it cannot fully utilize the resources of multi core processor computers. If more than 2000 rows at a time need to be processed, please implement parallel processing with the CSV file. Parallel processing involves splitting the csv file into smaller files and processing them in parallel, with multiple instances of the vb script that invoke multiple copies of the macro in excel.

Parallel processing

It was observed that when the baseline file of 2000 records were being processed, the processor usage peaked at 15%. Excel macros are single threaded and cannot take advantage of multiple processors. By running the LoadCSV sub in parallel by calling multiple instances of the same vbscript to execute the macro, processor capacity can be fully utilized. In the baseline environment 5 parallel executions of the vbscript ran and 10,000 records could be processed in under 3 minutes. To benchmark, open your task manager and start “Process CSV and generate HL7” with a large file. Go to the “Processes” tab in the task manager and sort by CPU. Once the processing has started, note the processor usage: