AIRA
AMERICAN IMMUNIZATION REGISTRY ASSOCIATION

Overview and Functional Requirements

The Message Quality Evaluation (MQE) Tool

September, 2018

Revisions/Change Log

Version	Date	Approved By	Changes/Updates
2.5 Beta	1/28/2018	AIRA-MQE	Initial version of draft document
Final	9/30/2018	AIRA Team	Reformatted document

Table of Contents

Introduction

The Message Quality Evaluation (MQE) tool was initially conceived as an open-source project managed by Dandelion Software and Research in 2012. The application, originally called the Data Quality Assurance (DQA) tool, has been available in the open source marketplace for several years and has been used by several IIS programs across the country. In 2017, AIRA selected the development of an initial version of the MQE, along with accompanying governance, as the next project for the American Immunization Registry Association's (AIRA's) Joint Development & Implementation (JDI) Initiative. As part of this process, prioritization of new requirements will shift to the JDI-MQE Governance Group, while agile development will continue to be implemented by the MQE Technical Team.

This document provides a summary of the functional requirements that drive the MQE tool. The MQE is an open-source application that will be freely available to the members of the IIS community that are seeking a tool to consistently evaluate and improve the quality of data coming into their Immunization Information System (IIS). The primary function of the MQE tool is to quickly and easily generate a series of reports that describe the quality of incoming immunization data as evaluated against national standards. The purpose of this document is to provide a high-level list of requirements that describe the primary functions of the tool. Details, technical requirements, prioritization, and status of enhancements will be enumerated in GitHub (https://github.com/immregistries/).

Background

AIRA's Modeling of Immunization Registry Operations Workgroup (MIROW) Steering Committee produced two documents focused for best practices regarding data quality assurance for Immunization Information Systems (IIS) specific to incoming data. These guides, Data Quality Assurance in IIS: Incoming Data (also referred to as Chapter 3) and MIROW Best Practices for Data Quality Assurance: Selected Aspects, outline how an IIS can improve overall data quality (completeness, accuracy, and timeliness). The Message Quality Evaluation (MQE) tool is designed based on the principles and metrics delineated in that guide, including the ability to evaluate:

- Presence of mandatory data fields
- Estimated duplicate records count
- General completeness of data
- Timeliness of data submission
- Inter-record conflict (i.e. vaccination date before patient birth date, combinations and single antigens recorded on the same date, etc.)

The core function of the MQE tool is to ingest HL7 v2 VXU messages and parse them to pull out relevant data. The tool focuses on evaluating relevant data and will generally ignore

minor message construction rule violations, which differentiates it from other tools like the publicly available NIST validator, although it does leverage the NIST message validation tool to identify HL7 conformance errors. The MQE tool can also process messages that meet different versions of the HL7 guide. The list of data quality checks (see Appendix B) the MQE conducts, coupled with the ability to adjust the weight given to issues discovered (thus giving the user control over how the reports work) is the real value of the tool.

Data quality has long been a focus of the Centers for Disease Control's (CDC's) IIS Functional Standards. The MQE tool will help programs meet those functional standards by giving staff a resource to assist them in making sure that the goals and standards are met. The value of accurate, complete, and timely data in being able to support clinical decisions, public health response, adverse event reporting, and recall reporting cannot be overstated. As an IIS matures, the importance of ensuring data quality becomes more pronounced. Data quality is the cornerstone of successfully reaching immunization-related goals.

Scope of this document

This document describes technical decisions made based on requirements gathered prior to JDI involvement, and will serve as a living document that will include updated requirements as they are gathered from the community. This document will not enumerate enhancement requests that have not been incorporated in the tool. Rather, this document serves to describe the current technical decisions and existing functionality of the currently released version[1]. Furthermore, JDI-MQE Governance processes are beyond the scope of this document, but those processes will help to direct work as new requirements are proposed and prioritized.

This document should serve as a reference for IIS technical staff as well as IIS program and management staff that are interested in reviewing the more technical components of this application[2]. Core IIS goals for data quality are woven into the IIS Functional Standards and therefore will not be repeated in this document. Similarly, the core business rules and principles of data quality assurance are detailed in the MIROW documents on data quality and will not be repeated here. That said, minimum MQE tool functionality should always align with these documented principles and are expected to do so unless otherwise documented.

---

[1] Since this document precedes a publicly available release of the MQE tool, the information contained in the current version of this document reflects functionality that will exist once the initial release is complete.

[2] For a higher-level understanding of the use cases for this application, please review the Message Quality Evaluation Tool Business Case.

Business Requirements Overview

The MQE is an application that provides users with standardized reports that indicate the level of quality in a batch of incoming immunization data. MQE reports not only summarize the quality of incoming data, but also provide a quality score which can be used to both gauge data quality as well as help set automatic trigger points for monitoring data quality. In this way, reports can facilitate quick and consistent decisions on which interfaces are candidates for production, need more review, or have major data quality problems. Use of this tool can give IIS staff confidence that data quality is high or, if not, a consistent framework for identifying issues.

Data quality is an essential and critical aspect of an IIS. The MQE tool is designed to assist sites to consistently evaluate and improve the quality of data coming into their IIS. It will allow users with varying levels of expertise to quickly and easily generate a series of reports that describe the quality of incoming immunization data. This tool facilitates efficiencies in onboarding, allowing IIS to respond to submitters more quickly after evaluating data, leading to the receipt of data that are timelier and more accurate. This will in turn improve the overall utility of the IIS for both public health and for providers/end users as they care for patients.

Business Rules

Data quality is assessed by comparing actual versus expected values. The immunization community, including CDC, AIRA, state and local IIS and immunization program staff, and immunizing providers, set the expectations for immunization data stored and produced by IIS. It is this community of subject matter experts that inform the standards developed and used by EHRs, IISs, and ultimately the MQE tool. With that in mind, the tool leverages the Implementation Guide for Immunization Messaging HL7 2.5.1 release 1.5 and its accompanying addendum (including code sets referenced therein), as well as the AIRA Guidance for HL7 Acknowledgement Messages to Support Interoperability.

All reports generated by the tool are based on these underlying standards and guiding documents. Data are evaluated based on required elements as specified in the standards, but can be weighted locally.

Assumptions / Constraints

The principal assumption for building a data quality assessment application is that high quality data is the backbone of an immunization program. Use of the MQE tool assumes that implementers have at least a rudimentary understanding of relevant immunization data requirements (e.g., patient demographic data, immunization name and date administered, etc.) and, more specifically, of HL7 v2 VXU messages which are used by nearly all IIS. It is also assumed that implementers understand the requirements of their own IIS.

Additionally, users must have access to incoming data files and understand basic mathematical descriptors (e.g., % complete). While this tool provides completeness and timeliness scores, which are objectively calculated, it also provides a subjective score that addresses the concept of data quality. This measure is configurable and is a key part of the tool's utility: rather than detailing an individual percent, the tool provides a quality indicator (e.g., "Good") to describe the evaluation of a batch of incoming data. Further, while the MQE tool offers a quality score and provides individual message-level feedback through a drill down feature, it does not prevent or promote data ingestion into an IIS. The MQE tool offers a summary for end-users to be able to evaluate incoming data so they can make decisions during onboarding or ongoing data quality assessment activities (e.g., whether to suspend a submitter's production feed or to promote a testing submitter to production status).

Functional Requirements

The core functional requirement of the MQE tool is to generate reports for three specific areas:

- Completeness: Are all required and most recommended fields being sent as expected?
- Quality: Are there errors or warnings in the received data?
- Timeliness: Are reports of vaccination administration being received in an appropriate time frame?

Each of these areas is comprised of several elements, which are calculated in Boolean fashion (e.g., completeness: is a required field populated; quality: is a vaccine appropriate for the patient; timeliness: was the message delivered within one day of administration). From there, each element has a calculated percentage for completeness across the message set (e.g., for this batch, patient name is complete in 95% of messages). These elements are weighted (which is configurable) and summarized to provide an overall quality score (which is also configurable) for the batch[3].

The MQE tool uses national standards as the basis for comparison, which allows for consistent data quality measures that can be used for training, documentation, etc. Consistent use of standards can further structure onboarding to any IIS. In addition to evaluation of HL7 v2 VXU messages, the MQE tool evaluates against established CDC supported code sets (e.g., CVX, NDC). Further, the MQE tool aligns with MIROW and supports EHR Certification Criteria.

---

[3] Details on data elements evaluated for display are presented in Appendix B. Screen shots exemplifying the report display are presented in Appendix C.

For quality evaluations, the following metrics are considered:

- Valid – code is recognized and may be used.
- Invalid – code is recognized but should not be used.
- Unrecognized – code is not known.
- Deprecated – code is no longer appropriate to use.

Processing

The following section addresses message processing components.

Ingestion

The Core Processing Model is the foundation of MQE processing. The process is built on the processing of a single HL7 VXU message. In the initial release, messages must be uploaded as individual files through the user interface; multiple files may be manually uploaded if they are zipped first. Other methods of ingestion may be incorporated as they are identified and prioritized[4]. As messages are processed, issues are detected, and added to the message store for acknowledgement (ACK) population. Issue detections are fixed in meaning, but can be configured to meet local business rules. These issues are classified as:

- Error: submitter needs to fix and resubmit
- Warn: submitter should fix but does not need to resubmit
- Info: submitter does not need to fix or resubmit, but should be notified
- Accept: issue is noted internally and will drive reports but is not sent back to submitter

Parsing

Relevant data from VXU messages (HL7 2.3.1 or higher) will be parsed and irrelevant data will be ignored. Conformance errors will not stop processing of the message. All relevant parsed data will be evaluated according to configuration. The NIST message validation code has been integrated into the MQE tool allowing it to identify HL7 conformance errors. By default, conformance errors are identified as Warn issues. Conformance errors do not stop processing.

Transformation

Data may be modified or transformed prior to validation. Preliminary data checks and data cleanup can be performed (e.g., address check against address engines such as SmartyStreets), and processing can be extended to allow for local business rules (configurable during installation).

---

[4] The tool functions optimally when messages are fed in through a real-time interface using SOAP/Web Services and the CDC WSDL. However, this level of integration requires additional configuration and is not available as part of a stand-alone deployment.

Issue Detection

The MQE tool uses a logical model of dependent checks that are executed when other checks are successful. This model creates a logical organization for checks. Incoming data will be evaluated against business rules (configured during installation). The weight of each detected issue is configurable by the IIS, as is the functionality to evaluate submitters differently. Issue Detection is run for all messages that pass through the tool, but not all issues require review. For more detail on data checks, see Appendix B.

CDC Vaccine Code Table

The MQE tool evaluates valid vaccines against an updated CDC Code Set table. In addition to the data provided by CDC, this table also contains metadata updated by the IIS community such as the date range that the vaccine is valid through. These data are not updated by CDC so rely on the community to provide updates as new information becomes available for a vaccine. The most updated version of this table is included in the initial release and the MQE supports updates to the table through an XML upload. Local changes to the table are not supported, but we anticipate that as part of the MQE User Community these data can be updated as they are provided and alerts about updated data will be available for end users to consume at the push of a button.

Consuming into Production

Consuming of messages into production is not supported when implemented as a stand-alone tool. The level of consumption will vary by level of integration, and fully integrated deployments could use this step to consume good data, resulting in the creation of the final ACK message. Again, it is anticipated that as this project and collaboration matures, this function will be updated.

Acknowledgement

An ACK is generated for each message indicating all issues identified for that message. The ACK is always available even if not used or needed. On real-time interfaces, it is returned when messages are submitted. For batch processing the user will receive a batch set of ACKs. These may be used for other processes or discarded.

Report

Details of messages received and the issues identified are displayed via web browser. A summary score and issue counts are displayed under a heat map representing the number of messages processed by day. The report groups issues together in batches by day and submitter, and issues can be viewed at the message level by drilling down through processed messages. Additional filters allow users to limit what is displayed by type of issue or error location.

Installation and Implementation

While the MQE tool can be deployed and implemented at various levels of IIS integration, the JDI project is only scoped to address implementation at the stand-alone level of deployment. The application will be available for download on the AIRA website[5] and directly from GitHub wiki page for the MQE (https://github.com/immregistries/mqe) and installation will require administrator access to either the installer's local workstation or server. Users interact with the application via web browser.

An in-application data store is available to facilitate immediate use of the product. The table below describes minimum requirements for immediate use.

Operating System	Windows, Macintosh, or Linux variant
Java Version	1.7 or higher
Memory	1 GB available
Disk Space	Base Application: 40 megabytes

* disk space requirements will increase based on data submitted

---

[5] URL not yet available.
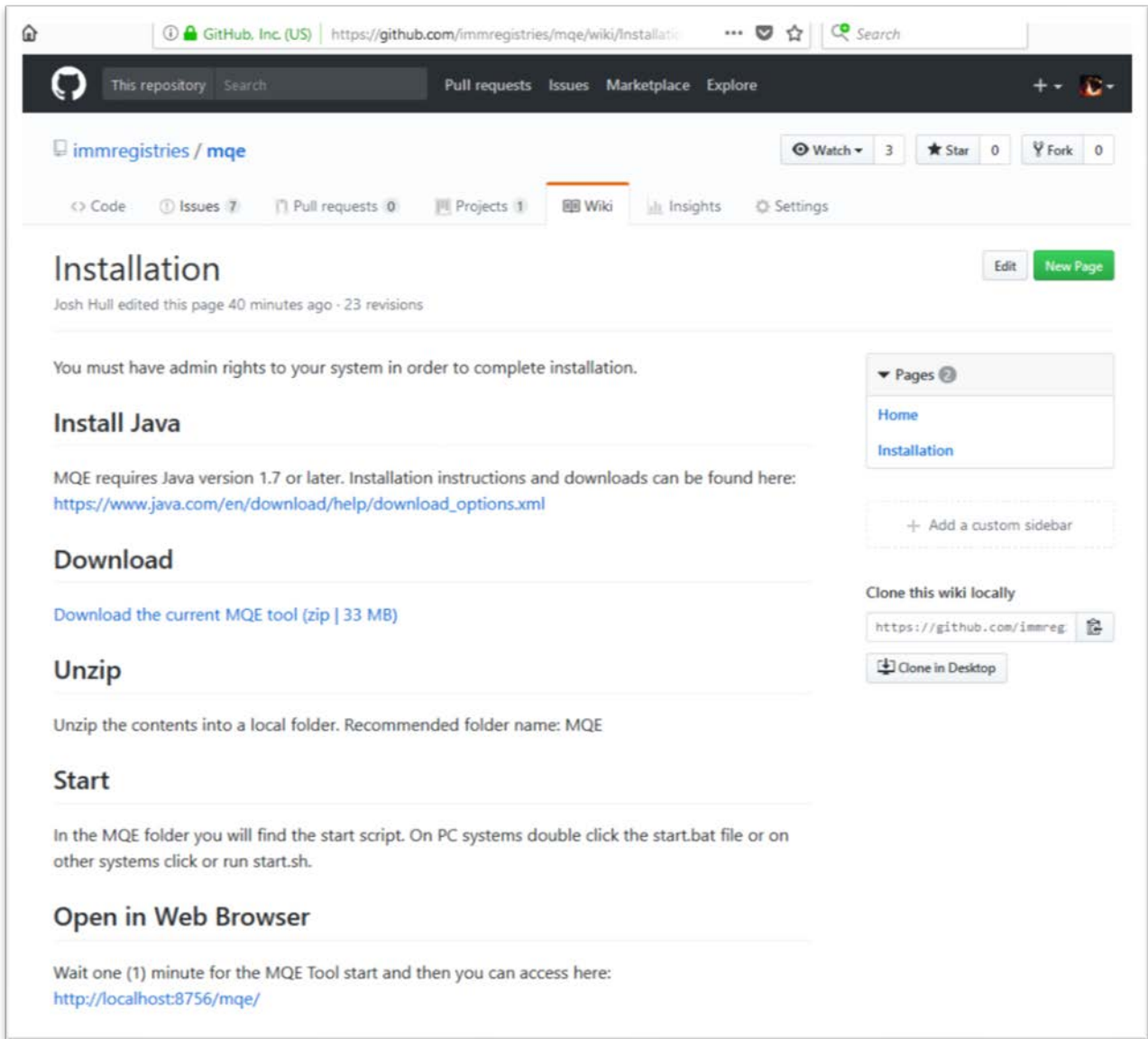
The installation process includes two major pieces:

- Installing the latest version of Java.
- Downloading, unzipping, and starting the MQE application

This entire process is expected to take less than 15 minutes for most systems operating at normal Internet connection speed using a modern operating system.

Once running, the admin user will navigate to the application using their web browser where they will be prompted to set an access username and password.

Level 1: Stand-Alone Deployment

In this level of implementation, the MQE tool may be installed on a desktop or server location for ad-hoc use, producing reports on demand. While the tool is configurable, default settings are consistent with the current CDC HL7 2.5.1 Implementation Guide so very little configuration is needed before a user can get their first report. Furthermore, while the tool is configured based on the HL7 2.5.1 Release 1.5 standard, it will also accept HL7 2.3.1 and 2.4 messages.

With no additional configuration, messages are uploaded to the MQE tool manually for individual or batched messages, so the user controls the amount of data processed. This option needs very little planning but does little to automate the data quality assessment processes.
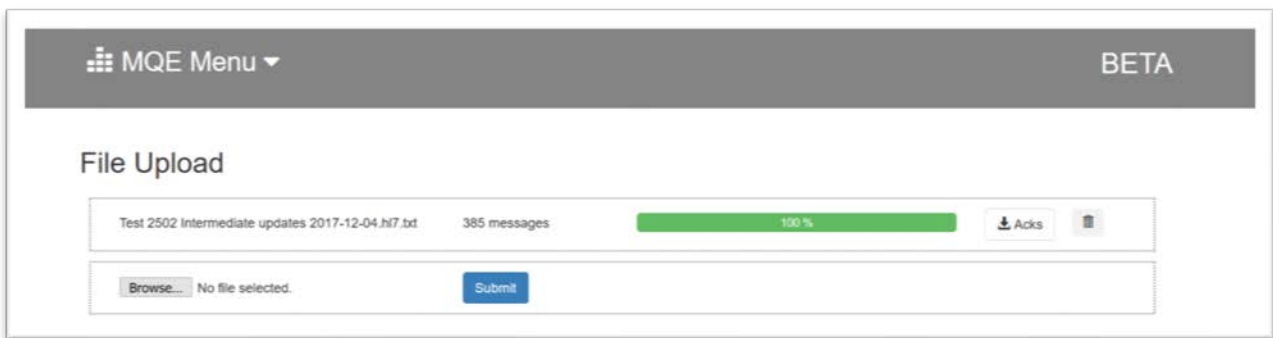
Data are stored in the in-application database and are therefore available as long as the application is running[6].

---

[6] *More sophisticated implementations may configure the MQE to reference an external database, however, this level of configuration is not described for stand-alone implementations.*

Level 2: Partial IIS Integration

This level of implementation is out of scope for the JDI initiative, but is included here for completeness. In a partially integrated deployment, the MQE tool is installed on a server so that it is always running and ready to process new messages received. Copies of messages received are sent from the inbound IIS interface to the MQE tool. In some IIS this may be an extra feed off an integration server while in others it might be an after-the-fact extract of messages received by the IIS. In either case the MQE tool is evaluating the data but has no effect on the final processing. This level of deployment does require additional planning and some technical expertise to access and configure automation on the server.

Level 3: Full IIS Integration

This level of implementation is out of scope for the JDI initiative, but is included here for completeness. In this level of deployment, the MQE tool is integrated either in front of the IIS or into the IIS such that it manages and augments responses from the IIS. In this setup, all of the information about data quality issues and the processing of messages by the IIS are returned in the ACK to the original submitter. The MQE tool is fully engaged with the sender and can communicate to the sender about data quality problems that need to be solved. This level of integration requires extensive planning, technical expertise, and software development.

Training

There are no formal training requirements to use this application. If a user has access to an input file and the processing directories associated with this tool, interpretation of the reports is straightforward. Training on configuration is beyond the scope of this document

at this time. A User Guide will be developed subsequent to the initial release and will be updated with input from the MQE Technical Team, User Group, and Governance Group.

Security

For this current AIRA-supported release of the MQE tool, there are no security requirements beyond those imposed by the IIS. While this system is a web-enabled application, we do not recommend that it is deployed with a public facing interface because it is still under development. As the application matures and security requirements are identified and implemented, deployment recommendations will change accordingly.

As an open-source application, there will be no restrictions on who can download the application package from the AIRA website, nor on the ability to retrieve the source code from GitHub. Furthermore, while anyone can branch their own version of the repository, only community developers vetted[7] by the MQE Technical Team will be able to commit changes to the shared source code.

Interface

The MQE tool has a user interface that is accessed via web browser and is being designed for IIS staff to use on a routine basis. Users of the system interact with the tool by submitting files and reviewing reports that are displayed on the browser. These reports can be filtered and drilled down at the user's discretion, but also have default settings that are configured during installation. See Appendix C for examples of interface defaults and drill down functions.

Assumptions / Constraints

The MQE tool application continues to be developed as an open collaboration between several independent entities. The Joint Development & Implementation (JDI) initiative considers this project a "found pilot," in that work was well underway through a community process prior to JDI involvement. The process for identifying and prioritizing requirements are constrained to improved functionality, bug fixes, and enhancement requests. Processes for prioritization are being developed with the MQE Governance Group and will be documented in an appendix. The list below is a summary of technical decisions made to date[8]:

- Source code will be public and available in an open repository (GitHub).
- Contributions are welcomed from all interested parties, but only contributors vetted by the MQE Technical Team will be able to modify the shared source code.

---

[7] The MQE Technical Team has three principal members as of December 2017: Nathan Bunker, Joshua Hull, and Jason Narine. While developers are self-identified, at this time, commit-rights must be agreed upon by these originating members unanimously.

[8] The Technical Decisions Summary Document will be available on the AIRA webpage.

- *There will be no legal or technical barriers for use of this system.*

The MQE tool does not depend on a specific implementation of an IIS to be useful; in fact, it operates completely independently of an IIS. A significant constraint in the initial release is access to incoming messages. One of the goals of this collaborative effort is that the IIS community will continue to engage with the development of this tool in such a way that partial and full integration may be realized by anyone.

Appendix A: Base Technology Decisions

This table highlights decisions specific to a shared development environment for an open-source application that relies on a distributed team. There may be additional considerations, but these decisions have been implemented for the initial release of the MQE Tool.

Decision Area	Requirement	Decision	Rationale	Cost
Code Repository	Must be centrally stored for team members to access, modify and share.	GitHub	Allows for single accessible location for shared work that is particularly well suited for dispersed development and open source applications.	Free
Programming Language	Must support deployment in different types of environments and systems.	Java	Can be deployed on a number of systems. Easy for developers with varying levels of experience to use.	Free
Build Process	Must be simple to implement and simple to maintain.	Maven	Build process supported by programming language.	Free
Technology Stack	Must use technologies and code from common libraries.	Spring	Aligns with the programming language.	Free
Bug Tracking	Must keep track of issues, stories, tasks, and feature requests.	GitHub	Has built in issue tracking available to the entire community. Given the maturity of the tool and the expected user base, GitHub is more than sufficient.	Free
Technical Team Communication	Must allow for distributed team communication and store conversations for future reference.	Slack	Can be integrated with GitHub. Used by AIRA for other projects.	Free

Decision Area	Requirement	Decision	Rationale	Cost
Development Process	Must follow a common methodology.	Agile, Scrum	Iterative development by distributed and dynamic team. Supports rapid development cycles with a focus on functioning releases. Issues tracking, prioritization, sprints, and implementation details stored in GitHub.	N/A
Software License	Must acknowledge what the tool can and cannot be used for.	LGPL[9]	Accommodates use by anyone for any purpose. Can be integrated with proprietary systems.	Free

---

[9] Lesser General Public License

Appendix B: Data Quality Checks

The below tables represent data that are evaluated in the initial implementation of the tool. These are subject to change following broader community engagement and input.

B.1 Message Completeness

Data elements presented below are evaluated for completeness. These checks align with expectations for required, expected, and recommended elements based on the HL7 v2 standard and may be further configured to accommodate IIS specific requirements. The number populated and percent complete is calculated across the batch for each element. Each element must be present in accordance with IIS Functional Standards and MIROW best practice recommendations. Other data checks required by these standards are presented in the Rule column.

B.1.1 Patient and Vaccine Record

Element	HL7	Rule	Reference
Required[10]			MIROW Ch. 3
Patient ID	PID-3		
First Name	PID-5.2		
Last Name	PID-5.1		
Birth Date	PID-7	Vaccination date must be after patient date of birth Birthdate must not be in the future	BR101 - MIROW DQA 2013
Sex	PID-8		
Address	PID-11		
- Street	PID-11.1		
- City	PID-11.3		

---

[10] Required elements are those that must be populated in order for the message to be accepted by the IIS. These fields must be sent in every message, every time.

Element	HL7	Rule	Reference
- State	PID-11.4		
- Zip	PID-11.5		
Vaccination Date	RXA-3	Must be after patient date of birth Must not be in the future	BR102 - MIROW DQA 2013
Vaccination Code	RXA-5	Must be aligned with vaccine manufacturer, product information, description Must be valid per metadata provided in CDC Code table	BR116 - MIROW DQA 2013
Information Source	RXA-9		
VFC Status	OBX-5		
- Zip	PID-11.5		
Expected[11]			MIROW Ch. 3
Middle Name	PID-5.3		
Phone	PID-13		
Mother's Maiden Name	PID-6		
CVX Code	RXA-5	Must be aligned with vaccine manufacturer,	BR116 - MIROW DQA 2013

---

[11] Expected elements are those that are expected by the IIS, but there may be legitimate reasons for a missing field (e.g., a person may not have phone number).

Element	HL7	Rule	Reference
		product information, description Must be valid per metadata provided in CDC Code table	
Lot Number	RXA-15	Must be aligned with vaccine manufacturer, product information, description Must be valid per metadata provided in CDC Code table	BR116 - MIROW DQA 2013
Manufacturer	RXA-17	Must be aligned with vaccine manufacturer, product information, description Must be valid per metadata provided in CDC Code table	BR116 - MIROW DQA 2013
Mother's Maiden Name	PID-6		
Recommended[12]			MIROW Ch. 3

---

[12] Recommended fields are requested by the IIS, but they may not be sent. Missing these fields would not normally prevent a submitter from moving to production, but would be identified as an area for improvement.

Element	HL7	Rule	Reference
Responsible Party Address	NK1-4		
Admin Amount	RXA-6		
Completion Status	RXA-20		
Optional[13]			MIROW Ch. 3
Action Code	RXA-21		
Refusal Reason	*RXA-18*		
Vaccination ID	*ORC-3*		

B.1.2   Vaccine Code (Example)

The table below shows a section of the Vaccine Code table and extended metadata that are included in the MQE. The origin and update to the core data (CVX, status, notes) is available directly from CDC. However, the MQE has functionality to import updates to the table through a Code Base Utility available from GitHub. Users will be able to update these data by clicking a button from the Utilities menu. Because updates to metadata are not provided by CDC, but rather the IIS Community, updates to those data will be done through a submission process (also available on the Utilities page). The MQE User Group will confirm these updates and the update package will be available as the Code Base is updated.

Group	CVX	Label	Status	Valid Start	Valid End	Notes
		Expected				
DTaP	20	DTaP				
Hep B						
Polio	10	IPV				
Hib	49	Hib (PRP-OMP)				
Hib	48	(PRP-T)				

---

[13] Optional fields are not expected and the IIS may choose to receive them, use them, or ignore them. They are listed for completeness, but do not affect the overall completeness score.

Group	CVX	Label	Status	Valid Start	Valid End	Notes
Influenza	141	Influenza, seasonal, injectable				
Influenza	140	Influenza, seasonal, injectable, preservative free				
MMR	94	MMRV				
Varicella	94	MMRV				
Pneumococcal	133	Pneumococcal conjugate PCV 13				
Recommended						
HPV	118	HPV, bivalent				
HPV	62	HPV, quadrivalent				
Rotavirus	116	Rotavirus, pentavalent				
Tdap	115	Tdap				
Hep A	83	Hep A, ped/adol, 2 dose				

B.2 Message Quality

Messages are evaluated to determine the number of errors and warnings that are encountered during processing. IIS can identify and set thresholds for errors and warnings that may trigger further action.

B.3 Message Timeliness

Messages are evaluated to determine the number and percent of records that fall into ranges based on the date the message was received and the date that the vaccination was

administered. Submitters should be sending messages as soon as possible after administration. To facilitate this calculation, the MQE identifies the last administered vaccination and compares it to the date the batch was received. It is up to the IIS to configure the threshold for timeliness based on IIS requirements or state law/policy and to weigh this metric according to the importance of the metric locally.

B.4 Codes Received section

Messages are evaluated to determine which codes were received. These data are not compared to expected values, but they are summarized and available for display. Access to these data can be helpful for staff to get an idea of what type of data are being sent and whether they make sense according to expected values that cannot be easily quantified. For example, the report might list all primary languages sent. There is no meaningful way to assess these data against a known metric, however, it might allow the user to determine whether clients from a clinic are non-English speaking which could then be used for messages and other outreach purposes.

Appendix C: Report Examples

C.1 Heat Map Calendar

The heat map is a visual representation of the number of messages processed by day. Clicking on any date in the heat map will generate a report for that date displayed below.
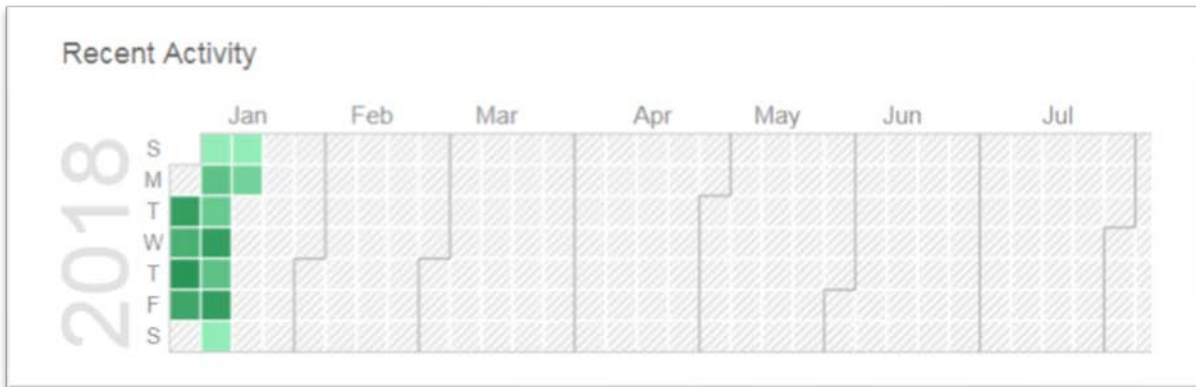
C.2 Message List

When a date is selected from the head map, users can view lists of messages processed for that date. Clicking on any of the individual messages will display a list of issues identified for that message.

C.3 Issues Detected

A detailed list of errors and warnings are displayed for messages processed on the selected date.

C.4 Summary Report

An overall summary of the messages for that date is compiled.

Messages	Errors/Warnings	Codes	Vaccines	Report

REPORT - SCORE: 532/541

COMPLETENESS

Completeness measures how many required, expected and recommended fields have been received and also indicates if expected vaccinations have been reported.

Score

Completeness Score	Description
532/541	Good

Patient

Patient Fields	HL7 Field	Score	Value Present
Overall		289/292	
Patient Id		30 / 30	385 / 385
First Name		20 / 20	385 / 385
Last Name		20 / 20	385 / 385