



# AIRA

AMERICAN IMMUNIZATION  
REGISTRY ASSOCIATION

## Implementation Guide

---

### The Message Quality Evaluation (MQE) Tool

September, 2018

## Revisions/Change Log

Version	Date	Approved By	Changes/Updates
0.1	05/05/2018	AIRA-MQE	Initial version of draft document
0.2	05/13/2018	AIRA-MQE	Internal revisions
0.3	7/25/2018	N/A	Copy editing, formatting
Final	9/30/2018	AIRA Staff	Reformatted Document

---

**Table of Contents**

- Introduction ..... 1
  - Scope of this document..... 1
  - Application overview..... 1
    - Application description ..... 1
    - Assumptions and constraints ..... 2
- Management Overview ..... 3
  - Description of implementation ..... 3
  - Points of contact..... 4
  - Major tasks and roles..... 4
    - Project team meetings..... 4
    - Development sprints..... 4
    - Software downloads and installation ..... 5
    - Application testing..... 5
    - Application use..... 5
    - Implementation schedule ..... 6
    - Security and privacy ..... 6
- Implementation Support..... 6
  - Hardware ..... 6
  - Software..... 6
  - Documentation ..... 7
  - Implementation impact..... 7
    - Configuration management interface ..... 7
- Risks and contingencies ..... 8
  - Inaccurate reports ..... 8
  - Risk of malicious code..... 8
  - Discontinued support ..... 9

---

## Introduction

The Message Quality Evaluation (MQE) Tool is an open-source application that is freely available to members of the IIS community. The tool is designed to assist sites in evaluating and improving the quality of data coming into their IIS. It allows users with varying levels of expertise to quickly and easily generate a series of reports that describe the quality of incoming immunization data. Individual users are able to take advantage of this stand-alone version of the MQE. With this level of implementation, users are able to evaluate messages as part of their ongoing onboarding activities as well as to augment existing data quality assurance activities in an ad hoc fashion. This level of implementation does not interface with existing IIS infrastructure, nor can it receive data in any way other than manual upload.

## Scope of this document

This document describes the use case and requirements for the stand-alone implementation of the MQE Tool. While more sophisticated implementations, including partial and full integrations with IIS are possible, describing those is beyond the scope of the initial version of this guide. Similarly, a description of the business case, functional requirements, or application upgrades will not be covered in this document. Rather, the scope and purpose of this document is to describe what an IIS program would need in terms of hardware, software, and resources to implement the tool at this level. This and related MQE documentation will be updated as new implementation requirements are identified.

## Application overview

For immunization data, incoming data must be assessed in order to determine whether they meet the quality required for business processes. The MQE Tool looks at whether data are of the right type and quantity to support their intended use. The MQE Tool standardizes and automates message quality evaluation functions, which results in improved throughput and increased data quality. The initial public release of the tool is intended to be deployed as a stand-alone application; it can be installed on an individual workstation or in a shared networked location, and messages are delivered to the tool manually.

## Application description

The application will be available for download on the AIRA website 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. A website will guide users through the installation and configuration setup.

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.

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 10 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.

### Assumptions and constraints

While AIRA does not endorse use of the MQE tool over any other application, AIRA does support community-led initiatives, including implementation of open-source products such as the MQE Tool. As such, and as part of the Joint Development and Implementation (JDI) Initiative, AIRA is responsible for initial drafts of supporting documentation and will continue to work with IIS that are interested in implementing tools like the MQE to improve their data quality and streamline workflow.

Because this tool is available in the open-source marketplace and development is contingent upon sustained investment by the MQE Project Team, the assumptions and constraints below are presented as the current state and best-case scenario. These are subject to change after AIRA's role is reduced.

### Schedule

MQE Project Team meetings will occur monthly and will be hosted by AIRA and facilitated by a community member. Meetings will address documentation updates and unmet needs, demonstrations of new functionality and updates on issue resolution, and planning and prioritization for future work. The two weeks following each Project Team meeting will be reserved for a coordinated development sprint that will be facilitated by a community-appointed Scrum Master. During this window, a dynamic group of developers will work on one or more issues that may include new functionality or bug fixing. These issues will be determined in collaboration with the Project Team. At the end of each sprint, developers and testers will meet to ensure that issues have been implemented as expected, release notes will be updated, and a functional release will be available on GitHub.

### Budget and resources

There is no out-of-pocket expense for adoption or implementation of this tool. However, there will be expenses related to staff time. In addition to the monthly Project Team meeting, developers will also be expected to put in time to ensure the ongoing evolution of the tool meets their program needs. In its initial implementation, technical resources may be required for staff who do not have administrative rights to the machine on which the tool is deployed. Additional budgetary concerns may involve staff time for testing of new releases. It is expected that a program could see an overall cost savings in terms

of staff time as use of the MQE Tool improves onboarding and as routine data quality analysis activities are made more efficient.

### Constraints

The MQE Tool is an open-source application, built in Java, and in its initial release is not suitable for use outside of an IIS. Data that are processed through it are fully identified; therefore, the system must be deployed in a manner that is consistent with the jurisdiction's local privacy and security policies. Users must be able to install the application, have access to a modern web browser, and have access to raw HL7 messages for the tool to work as intended. The Open Source Product Adoption Evaluation Worksheet (available on the AIRA website) may help guide your decision about whether or not use of the MQE Tool is appropriate for your program.

## Management Overview

Because implementation will be locally determined, this section describes a typical stand-alone implementation approach and areas that will need to be addressed by implementers.

### Description of implementation

For the most up-to-date information on versioning and installation requirements, refer to the GitHub wiki page on installation (<https://github.com/immregistries/mqe/wiki>). For developers and other technical personnel interested in accessing and manipulating the source code, see the wiki pages on setting up the development environment and project practices. The stand-alone implantation is straightforward and does not require complex integration with any other technology. It can be installed on both Windows and Macintosh operating systems. As mentioned above and elsewhere in project-related materials, implementing this tool as a stand-alone application requires administrator access to the computer on which the application will be installed, Java version 1.7 or later, access to raw HL7 files, and a modern internet browser<sup>1</sup>. Installation should take less than 10 minutes.

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 users 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

---

<sup>1</sup> At present, there are known display issues with Internet Explorer, which has not been supported by Microsoft since January 1, 2016. If the user experiences issues with a particular browser, it should be reported via GitHub.

assessment processes. Integration of the tool as part of routing data quality efforts is also jurisdiction-specific and should be discussed with appropriate staff and managers within an IIS.

Use the table below to identify information for your organization about installation and accessibility.

Server/IP address for installation	
Web browser(s) supported	
Users with access	

### Points of contact

The positions below are provided for consideration to ensure that appropriate parties are aware of implementation and use of this tool. Add or subtract personnel as appropriate for your organization.

<i>Role</i>	<i>Name</i>	<i>Contact Number</i>
IIS Manager		
IIS Data Quality Analyst		
MQE Project Team Representative		
MQE Implementation Representative		
MQE Tool Developer		
MQE Tool Tester		
MQE Tool User		

### Major tasks and roles

Because this tool is produced and maintained in the open-source space and is available as a package for download, the major tasks associated with implementation are minimal. As specific individuals are identified for each of these major tasks, the table above should be populated with their role and contact information.

### Project team meetings

While use of the tool is not restricted to MQE Project Team participants, it is strongly recommended the IIS appoint someone to attend the monthly meeting to learn about new features, bug fixes, and documentation updates.

### Development sprints

In order for this tool to have longevity and support from the IIS community in the open-source marketplace, active engagement by developers and technical personnel within the community is required. While use of the tool and recommendations for enhancements are not restricted to sites that provide developer resources, the prioritization of site-specific enhancements will be realized only if those sites are engaged in the development process. Sites that implement are encouraged to identify technical resources who have or are willing to develop the skill to work on the tool and to make those source code modifications available to the larger IIS community rather than making local changes that are never integrated upstream.

## Software downloads and installation

Downloads of source code, compiled application, and related documentation are available from the GitHub repository (<https://github.com/immregistries/mqe>) and the AIRA website (<http://www.immregistries.org/message-quality-evaluation-mqe-tool>). There is no cost to download any of the materials. Installation of the application can be done on any machine to which the user has administrative rights. Details about installation and testing can be found on GitHub. In addition to the installation package, Java version 1.7 or later must be running on the machine.

After downloading the zip package from GitHub, the file must be unzipped to expose the contents<sup>2</sup>. The application can be launched with no additional configuration on a PC by double-clicking the start-mqe.bat file (on other systems, click or run start-mqe.command). After one minute (or when the last line of Java code indicates “Tomcat started on port(s):8756”), double-click on the MQE Demo.url to launch the application with the default web browser using the URL <http://localhost:8756/mqe/>.

The download package includes a built-in database file that stores data processed through the application. As new releases are made available, users may choose to use the embedded database or update only the .war file. Sites may choose to use a separate database, but that level of configuration is beyond the scope of the documentation required for the initial release. Please work with the MQE Project Team for more information on customized implementations.

## Application testing

Application testing will not be described in this document, but updated information on testing is available on GitHub. There is a complete test script in the MQE Testing Guidance document, including a description of how to download and install the application with expected results located on the AIRA website.

## Application use

It is recommended that all sites planning to use the application download the Testing Guidance document and step through the testing script after each updated release to verify that no issues were introduced during the sprint. As the application is being used, enhancements, questions, or bugs should be reported in GitHub (details on how to report are provided in the Testing Guidance document). As new releases are made available and depending on the level of workflow integration, users may choose not to update the database or configuration files and to update only the source code. In this case, the application should be stopped, a new package should be downloaded, and the .war file should be replaced before relaunching the application.

The MQE User Guide (available on the AIRA website) describes in detail how the tool is to be used and how to interpret reports. Basic functionality is described in the both the Functional Requirements and Testing Guidance documents.

---

<sup>2</sup> Package contents include an embedded database and configuration files. Refer to the GitHub wiki page for details on each file. For immediate implementation, follow launch instructions.



## Implementation schedule

Because installation of the tool as a stand-alone application takes less than 10 minutes and requires no networking or system integration, implementation at the jurisdiction level can be done as long as installers have administrative rights to the host machine. This may involve coordination with a technical resource. At present, updated releases may be available monthly, and it is recommended that sites coordinate with appropriate staff to ensure that the latest code is being implemented. Because the application will be updated by mid-month and new functionality will be reviewed at the beginning of the month, consider adding this update to other monthly product updates in your jurisdiction.

## Security and privacy

The initial public release of the MQE Tool has no security configurations associated with the tool. Sites should ensure that the application and access to it follow internal security and privacy protocols (e.g., installed behind a firewall, accessible to authorized users only). While the source code is open and available for public use, the MQE Project Team serves as the governing body to determine who can actually commit code changes to the master branch. Access must be granted by the lead developer, and code is always reviewed by at least one other developer (from an organization other than the contributor) before changes are approved and accepted. See the GitHub wiki page on Project Practices for more information.

## Implementation Support

In addition to the roles identified above that are specific to your organization, the MQE Project Team, which includes the community of developers, testers, and business analysts, will serve as external support for implementation.

## Hardware

This application is designed to run on either an individual computer workstation or server. It is not designed for installation on smart phones or tablets at this time. However, if the application is installed on a networked server and a user has networked device with a web browser that can access that domain, reports may be viewed through modern browsers on that device. Please note: because there are no security configurations on this application, accessing data with protected health information on remote devices is subject to the same privacy and security restrictions as other data within your organization.

## Software

As described above, the application, configuration files, and source code are publicly available for download via GitHub. For details on decisions made regarding the code language and development environment specifications, please refer to the Functional Requirements document. The operating system that will be hosting the application must be running Java version 1.7 or later. The tool will render on most modern web browsers, including Safari, Firefox, Chrome, and Microsoft Edge. As issues are identified with certain browsers or versions, those issues will be noted in GitHub.

## Documentation

In addition to this document, AIRA has provided<sup>3</sup> a number of supporting documents and resources to help organizations determine whether or not the application fits into their workflow:

- Open Source Product Evaluation Worksheet
- MQE Tool Business Case
- MQE Tool Functional Requirements and Overview
- MQE Tool Testing Guidance
- MQE Tool User Guide
- MQE Project Decision Framework and Sustainability Plan

At present, these documents describe only the stand-alone implementation of the MQE Tool. More sophisticated implementations may be executed (and are currently under way in some jurisdictions), but it will be the responsibility of the community-led Project Team to update these documents as new implementations are defined. In an effort to maintain flexibility and easy access to the most up-to-date information, the GitHub repository (<https://github.com/immregistries/mqe/>) will always be considered the most up-to-date information.

## Implementation impact

For details on how implementing this application is expected to influence workflow within your organization, please refer to the MQE Tool Business Case. As other impacts or benefits are identified, the MQE Project Team will be responsible for making sure materials are updated to reflect new information.

## Configuration management interface

As described elsewhere, the information included in this document describes the initial version of the MQE Tool, which does not presently support site-specific configuration changes. However, because the source code is open, sites with requisite skill in Java programming may make site-specific configuration changes at their discretion<sup>4</sup>.

The MQE Tool does not require any additional configuration to work. The code base reflects community-supported thresholds for metrics including metadata regarding various vaccine codes. However, while the MQE Project Team is developing a process for community members to make changes to the Vaccine Code metadata (e.g., time frames of validity for various vaccines), a site may choose to update its own vaccine code file by modifying the Configuration.xml file directly. Similarly, for sites that have

---

<sup>3</sup> Some of these documents might not be completed prior to the publication of this document but are expected to be available by the end of the JDI supported project period: September 30, 2018.

<sup>4</sup> Because this tool is intended for community use, sites that make local changes will lose those changes if they stay up to date with versions of the software available on GitHub. It is strongly recommended that these sites join the MQE Project Team to make changes following the project practices outlined on GitHub Wiki so the entire community can benefit from such functionality.

active accounts with SmartyStreets for address validation, the validation.property file may be updated with site-specific credentials to ensure addresses are being evaluated as part of the message quality evaluation report.

As this project matures, additional configuration files or processes will be defined and described by the MQE Project Team.

### Risks and contingencies

With the stand-alone implementation, risks of a failed implementation are minimal. This application is not integrated with any other technology and therefore poses no threat to data integrity for messages being delivered to the IIS. The sections below describe potential risks or areas of concern and a proposed contingency plan should they be encountered for sites that have integrated this technology into their workflow.

#### Inaccurate reports

AIRA and the IIS community have taken great care and have an outstanding track record with regard to data quality standards. They have been instrumental in drafting numerous guidance documents and standards that outline which elements in an immunization message are important and what data are considered valid. The MQE Tool was created with those standards in mind, and every effort has been made to ensure that calculations for timeliness and completeness align with those standards. One purpose of the JDI initiative is to deliver a product that meets the community's needs specific to evaluating HL7 message quality against these standards. There is a process in place for code review and testing, and the MQE Project Team will make every effort to address any errors in the code that might result in an inaccurate report within the development cycle in which it is identified.

An inaccurate report at this level of implementation may affect the program's ability to provide timely feedback to providers that are trying to onboard with a program and may impact the program's overall trust in the product. However, the program could simply discontinue using the product until the issue is resolved; if resources permit, the site could manipulate the source code of their own implementation; or—if the error is minor— it could disregard that section of the report until the issue is resolved.

#### Risk of malicious code

The risk of the introduction of malicious code into the source code prior to release is also minimal. The developers with access to the code are part of the immunization registry community and are likely using the tool in their own implementations. This team has control over code management, including who is allowed to commit changes and the process by which those changes are validated. However, if malicious code is introduced during the code review sessions prior to a release, it should be discovered through the testing process, and the commit will be denied.

Because the application source code is available for manipulation locally, a bad actor in your jurisdiction could introduce malicious code. In that instance, your organization's policies for application removal and technology destruction or recovery should be followed.

### Discontinued support

Probably the greatest risk following adoption of this tool is discontinued support of its evolution. In order for this tool to provide maximum value, sites may choose to improve their data quality by evaluating all incoming messages. For this initial implementation, files must be delivered manually. Should sites continue toward more advanced levels of implementation, the tool may be more integrated with the IIS. As with any software adoption, investment in the tool presumes ongoing support and deferred depreciation.

This project, while initially supported by AIRA, relies on community ownership, development, and governance. Without those components, a site may not be able to engage in a more robust implementation. If the tool does not meet your current need and the promise of enhancement fades, the project will not survive. If this happens, the program will be forced to resort to whatever tools and methods it is currently using for HL7 message quality evaluation and data quality analysis.