# Challenges and Solutions in Electronic Health Records Data Import & Export


## CONTACT US

**Leidos Health**
Interoperability@leidos.com
770.986.3526


leidos

Leidos Health helps its clients use technology to deliver care more efficiently, improve care quality and safety, and better protect the health of national and global populations.

# Contents







leidos

## INTRODUCTION

With the increased adoption of electronic health record (EHR) systems, comes the additional challenge of making the data collected by those EHRs available to other systems, both inside and outside of a healthcare enterprise. In such enterprises, transferring data between systems can be difficult. An even greater challenge manifests when sets of historical data from existing or "legacy" systems must be imported or "backloaded" into other applications. Reasons for these backloads include the following:

- Migration from legacy applications to replacement applications
- Pre-populating longitudinal patient record system
- Pre-population of a Health Information Exchange
- Pre-populating an analytics database.

In all of these situations, a destination system—or "data repository"—requires historical data from one or more applications in the healthcare enterprise. In large enterprises with a mix of systems, data-backload requirements grow in size, frequency, and complexity. This paper explores these issues and standard solutions, and the role that the Leidos Data Exchange Framework can play in the market.

## BACKLOADING DATA: CHALLENGES

Deciding on a data-exchange solution for data-repository backloads requires consideration of the types and sheer variety of obstacles and possible solutions available.

One of the critical challenges to achieving a fully functioning data repository backload solution is the diverse formats used to store and transmit data. Key data are stored in systems created at different times by different developers for different purposes. In many cases, these systems have been further customized by organizations for internal use, and the systems have worked beautifully—so long as the data remained internal. In some cases, however, the systems were never designed to work together or to exchange data. And of course, data input has varied as well: different users have entered data using different rules, and in some cases, they have interpreted and applied rules differently, even when the rules were supposedly uniform.

Drawing fragmented, non-homogenous data together into data repositories often requires that the data be normalized to the standards—e.g., the structures, attributes, vocabulary—of the recipient system. Alternatively, the recipient system can be made to convert noncompliant data, but this approach has sometimes involved the use of proprietary software—which tends to become increasingly costly over time and often leads to still more fragmentation.

Even so, some widely used standards have emerged, and each offers advantages and disadvantages as a solution for backloading data.

## HL7 VERSION 2 MESSAGE-BASED BACKLOAD

With this option, clients provide a series of HL7 V2 messages.

### Pros:

Use of HL7 V2 messaging is ubiquitous in the United States, particularly in the acute-care setting. Organizations will most likely have applications that support HL7 V2 messaging as well as personnel familiar with use of the standard.

Some vendor and commercial software is capable of providing HL7 V2-based backloads of data.

Several approaches to mapping data contained in HL7 V2 messages to the appropriate data elements in a data repository are available. They range from identifying appropriate HL7 messages and segments to convey particular data elements to simply treating most elements as name/value pairs in an HL7 V2 message.

### Cons:

- Many clients' suite of applications do not support all the types of HL7 V2 messages necessary to convey the entire set of data.
- HL7 V2 message instances may contain data beyond what is needed and may pose privacy issues (e.g., both PII and PHI data), depending on the sort of agreements in place. This additional information may need to be stripped out of the messages prior to any backloading.
- HL7 V2 comes in a large number of versions, ranging from v2.1 through 2.7 (v2.7.1 and v2.8 are in the works at HL7). Most implementations in the United States are based on earlier v2.x versions, v2.1 through v2.5.1; v2.2 and v2.3 are probably the most common. To complicate matters, the specific versions supported by various applications in an organization's enterprise will likely vary. That means a variety of potential version and message-type conflicts may arise. Worse, virtually all V2 interfaces have been customized or tailored to specific organizational requirements.

## CLINICAL DOCUMENT ARCHITECTURE–BASED BACKLOAD

With this option, the enterprise provides one or more Clinical Document Architecture (CDA) document(s) with relevant elements. The CDA R2 specification has the flexibility to convey a wide variety of clinical information in a document format.

### Pros:

- "Meaningful Use" rules defined by the Centers for Medicare & Medicaid Services have driven adoption of the HL7 CDA standard, particularly the consolidated CDA (C-CDA) specification. Consequently, many healthcare enterprises are already familiar with CDA implementation.
- The HL7 C-CDA specification covers a large percentage of the data elements used in many data repositories, though there will undoubtedly be gaps that will require additional specification.

### Cons:

- Data residing applications not targeted by Meaningful Use rules (such as pharmacy, lab, radiology, etc. applications) may not be capable of generating CDA documents.
- The CDA standard, based in XML, is quite verbose, so considerable overhead is built into the format. A fully populated CDA instance containing all the data needed for a particular exchange could be very large.
- Applications that have implemented C-CDA for Meaningful Use may not be as flexible—in terms of what content can be placed in the CDA document—as necessary.
- CDA has some serious problems conveying workflow-related information as well as information related to patient-location tracking in hospitals. CDA was not designed to deal with these sorts of information, so workarounds to collect such data in this format would have to be developed.

## OTHER FILE-FORMAT-BASED BACKLOADS

An organization's applications most likely use a number of different file formats. Often, vendors provide their own file formats for data extracts from their software. These formats may be based on standards other than those listed above; they may also be vendors' proprietary formats or even client-configurable formats.

### Pros:

- Backloading using existing formatting is likely to be a simple matter on the client side because it requires little if any adaptation or preparation of the data.

### Cons:

- Using data in a client-configured or vendor's proprietary format is very likely to require ad-hoc development of special parsers to handle the format at some later point and thus cause unforeseen delays and/or costs.

## LISTENING MODE "BACKLOAD"

This option does not really backload data—rather, it involves installing an application that "listens" to traffic on a client's network. As transactions flow through the network, the application examines the transactions and determines whether the data content is relevant to the data repository. When the application detects a transaction of interest, it either stores the transaction for further processing or does the actual processing real time. Over time, this process builds up the same level of data needed for a backload. The specific format of the data depends entirely on what is being used on the client's network. This could be entirely HL7 V2 messages or a mix of formats.

### Pros:

- Perhaps the simplest solution once a listening application is in place.

### Cons:

- Involves considerable delay between the initiation of data capture and completion of the "backload." For example, if 18 months of historical data is needed for the backload, then the listening method necessarily involves a matching delay of 18 months between the start of data capture and the time at which the data repository has sufficient data for use.
- A variety of HL7 versions and possibly distinct formats may be in use, complicating collection and potentially causing delays in gathering all the appropriate data.
- Data elements essential to the data repository may not be available on the hospital's network.

## MIXED BACKLOADING OPTIONS

Any given enterprise is likely to use (or to have used) a wide variety of applications that may or may not be compatible. Some applications may support HL7 2.x messages, while others for whatever reason do not. Some applications may support backloads in specific formats, whereas others do not support backloads in any format. Implementing the mixed approach, then, requires an examination of the organization's particular mix of applications to determine the best approach for collecting the rich set of data necessary for a repository.

### Pros:

- Best option for a mix of commercial and customer-specific configurations.
- A high level of customization is possible if necessary.

### Cons:

- The data collection itself is likely to be very complex and may require multiple applications and/or interfaces.

## THE LEIDOS SOLUTION

The considerations outlined above make clear that while backloading data is necessary, it can be an expensive, complex, and time-consuming exercise. Yet modern healthcare enterprises are under increasing pressure to reduce costs and maximize efficiency.

In light of these pressures, two of the important questions organizations should ask are:

- Which is the best method for backloading data?
- What will it cost?

Until now, organizations trying to select a method for backloading data from multiple systems have had to choose among several relatively expensive but incomplete options.

Introducing the Leidos Data Exchange Framework (LDEF)—a cost-effective, adaptable solution built for flexibility and capacity. Where most commercial and vendor data-exchange solutions impose significant costs in terms of data preparation required for data compliance, LDEF provides—at a reasonable price—a robust package of the key capabilities needed to deal with a variety of data formats.

As the name suggests, LDEF is an integrated framework. It combines the best- production-ready open-source components, and can be configured to use any number of these components to create an interface-engine solution that can be customized for any client's unique situation. And although it brings together multiple components, LDEF can be managed as a single, comprehensive system.

LDEF is a highly extensible and dynamically expandable solution that is cloud and Service Oriented Services (SOA) enabled. It is built around a solutions core that combines orchestrated SOA service integration with a rules, events, and complex-event processing engine, all of which provide both real-time and historical message or data extraction, evaluation, and analytics.

Adding LDEF to the mixed options scenario above provides the capacity and flexibility organizations need to extract and integrate data from a variety of sources and formats—including legacy systems with data from different databases created using different companies' software—into a common format. After carefully analyzing an enterprise to determine the sources of information required by the destination data repository, Leidos will map that data's sources formats to the data repository's format. For example, standard mappings from HL7 V2 can be developed fairly rapidly and then modified to fit any custom implementations in an organization's enterprise.

But the ability to handle HL7 2x messages is only a glimpse of what LDEF has to offer.

LDEF's combination and integration of fully scalable components makes it adaptable to any client's enterprise, which minimizes the client's burden while enhancing its ability to extract, transform, load, generate, store, receive, route—and ultimately, use—data, whatever that data's source and original format, thus maximizing the investment made in your EHR and minimizing data entry.

## CONCLUSION

Mixed Backloading provides the best technical and economical solution for electronic health records data Import and export. The addition of a framework, such as LDEF, to this approach will reduce implementation time by providing a major portion of the solution at a very attractive investment cost, an investment that will continue to pay benefits overtime for this and other efforts. LDEF will also ensure a well tested, managed, highly secure, extensible, and proven code base for the solution.

If you or someone in your organization is actively working in these specific or related areas please contact the Leidos LDEF team for an assessment and demonstration of how LDEF might benefit your institution,  at (770) 986-3526 or Interoperability@leidos.com.