

JD Edwards and EDI

By

Skip Stein

[Management Systems Consulting, Inc.](#)

Background

For over 13 years that I have been involved with the implementation of the JD Edwards (JDE) systems, there has been confusion and misunderstanding of how JD Edwards handles electronic transaction processing and Electronic Data Interchange (EDI).

In the JD Edwards EnterpriseOne Data Interface for Electronic Data Interchange Overview documentation it states:

“Electronic Data Interchange (EDI) is the paperless, computer to computer exchange of business transactions, such as purchase orders and invoices, in a standard format with standard content. As such, it is an important part of an electronic commerce strategy. Electronic commerce is a means to extend business processes to include suppliers, customers, and employees in a fully integrated supply chain.

As an interface between the JD Edwards EnterpriseOne system data and the translator software, the JD Edwards EnterpriseOne Data Interface for Electronic Data Interchange system (code 47) acts as a staging area for moving data in and out of the application systems. In addition to exchanging EDI data, this data interface can be used for general interoperability and electronic commerce needs where a file based interface meets the business requirements.”

Note that these 'code 47' data structures are tables within the overall JDE data base structure and follow the same conventions, structure, formats and data content consistent with the rest of the JDE data structure. It mentions that these tables may be used *“where a file based interface meets the business requirements”*. For this reason, the JDE Architects provided what is termed a 'flat file interface' series of generalized program structures to aid in the importation of file based (as opposed to data based tables) information and likewise to export JDE data table information to external 'file based' third party or external data structures.

Unfortunately, over the years, many of the supplied EDI interfaces have been implemented by individuals who not only didn't understand EDI, but didn't have a good grasp of Data Base structures and table intricacies; especially as architected within the JDE framework. Data files as static fixed format data structures have been around for decades before the advent of Data Base Table architecture. Maybe it is understandable that less well trained individuals would adopt an antiquated form of data handling; EDI, after all, goes back to the mid 1970's well before Data Base Architecture was evolved. EDI data structures themselves are a delimited, highly structured form of data presentation. While they are complex and detailed in their complexity and often hierarchical in nature, they continue to be fixed, variable length

data delimited bits of information enveloped in a highly controlled structural format.

As EDI evolved it became apparent that standardization would be necessary to provide a viable international means of data exchange. Over the years, The American National Standards Institute (ANSI), United Nations Economic Commission for Europe (UNECE), became the international repositories and standard setting organizations. Over the years, specialty standards that pertain to industry specific formats evolved. These are common in industries such as the petrochemical, pharmaceutical, transportation and other industries.

Today, analysts estimate that businesses trade more than a hundred billion in goods and services using EDI. It is predicted that this number will increase as new technologies make EDI more accessible to businesses.

For more information about EDI, please visit <http://ediguy.net/documents/edi-definition.html>.

The EDI Translator

EDI grew and penetrated many industries, institutions and government agencies across the Globe. Because of its structure and nature, highly complex codification structures evolved. These codification structures helped define, data content, type, format and usage. Depending on specific standards format, these codification structures either delimited a data element or functioned as a property designator.

Because of the growing complexity, software companies recognized an opportunity to more fully automate functions and replace custom/proprietary program code that had been developed to decode or translate EDI documents in a standardized way. These became the 'translators' of EDI you see today. They are continually updated to increasingly complex codification schemes and formats as EDI standards evolve.

Initially the EDI Translators accepted EDI formats and converted them to 'flat files' or common fixed width data file formats. Conversely, for outbound documents, the translator accepted a fixed file format and assembled a properly formed EDI document.

As Data Base Table structures and data bases evolved, so to did the EDI translators. They became capable of accessing and integrating with the most common data base systems, while maintaining the backwards compatibility to handle archaic fixed/flat file data structures.

Today there are a host of different program products that do EDI Translation. Some better than others. Some more capable than others. Almost all have standards capabilities to process a huge variety of EDI formats and international and specialized data structures. For the most part they ALL can access, update and retrieve data from a huge variety of data base systems and structures.

JD Edwards Code 47 Tables

As previously discussed the JDE F47 tables (all EDI tables within the JDE systems begin with the prefix 'F47') are used as interim transport mechanisms to handle data documents into and

out from the JDE system. These documents may be purchase orders, invoices, or a host of other documents dependent on functions and functionality of the JDE system.

Most common EDI translators can accommodate, facilitate and work with the JDE F47 tables. Indeed, many have per-codified 'maps' for the most common documents and are per-coded to integrate seamlessly with the JDE system. Tweaking of these standardized mapping forms can accommodate variations in requirements of different customers/suppliers.

In other cases, development of standardized mapping structures by a company that has implemented or is implementing JDE, can be done relatively quickly by mapping directly into and out from the F47 tables. The use of interim flat file structures, programs and formats is totally unnecessary.

Even when there is a requirement by a customer/supplier who cannot or will not work with EDI, flat files, email, XML documents or just about any other data transmission format can be accommodated by most EDI Translators. The 'any to any' format capabilities of most translators make the necessary data manipulations to ready data for insertion into JDE F47 tables straightforward. In many cases the EDI translator can be used for data migrations, conversions and Extract/Transform/Load requirements, often eliminating the need for other third party utilities.

Conclusion

In conclusion, it is seldom if ever necessary to utilize the backward compatibilities provided by the JDE 'Flat File Interface' routines. It complicates the process, increased the cost and requires unnecessary ongoing maintenance. Furthermore, because most EDI document requirements vary dramatically between customers/suppliers, it is necessary to have a different flat file program format for each and every one. With an EDI Translator, it is usually only necessary to have a single or a small selection of mapping formats to accommodate many tens or hundreds of customer/supplier requirements.

For additional information about implementing a cost effective EDI solution for your new or existing ERP system, please contact The EDI Guy at <http://www.ediguy.net> or our parent company, Management Systems Consulting, Inc. at: <http://www.msc-inc.net>.