

# The Bridge to Nowhere

an eprentise white paper



tel: 407.591.4950 | toll-free: 1.888.943.5363 | web: [www.eprentise.com](http://www.eprentise.com)

Author: Helene Abrams  
[www.eprentise.com](http://www.eprentise.com)

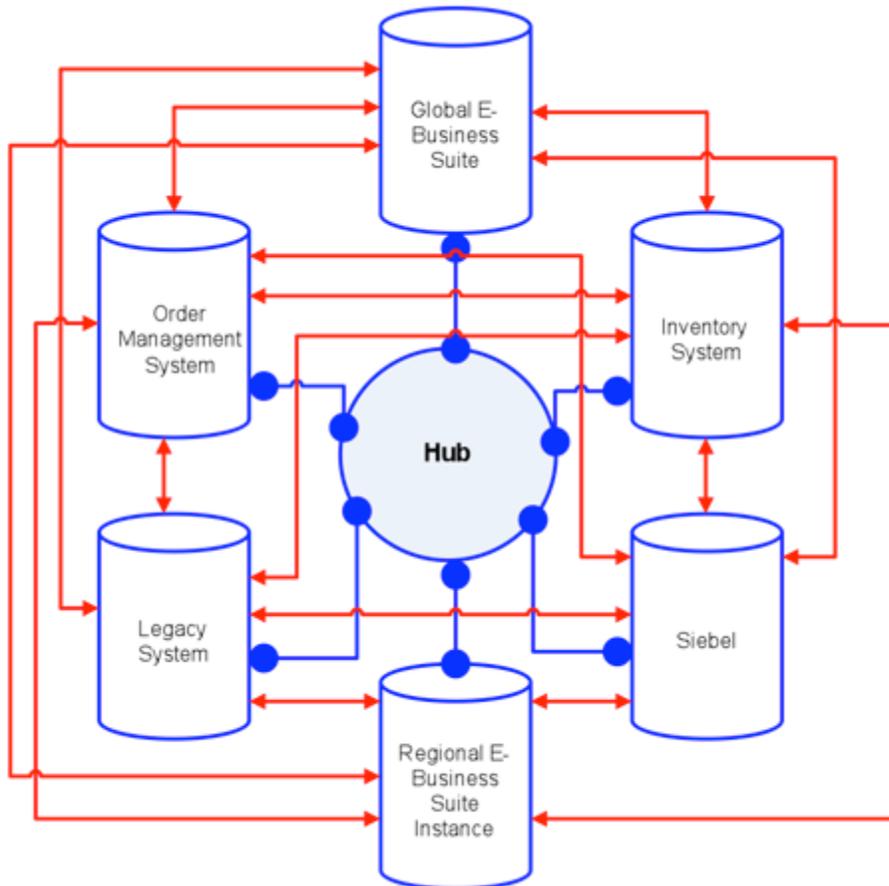
© 2022 eprentise, LLC. All rights reserved.

*eprentise, FlexField Express and FlexField are registered trademarks of eprentise, LLC.*

*Oracle is a registered trademark of Oracle Corporation.*

*All other company or product names are used for identification only and may be trademarks of their respective owners.*

Integration of diverse applications means building bridges that connect one application to another in order to pass data between them. There are several ways of integrating data, from writing code to insert data that is generated in one system into another system to using a hub-type technology with several adaptors that also includes a messaging system and a broker for routing and transformation of the data. In the following diagram, the blue lines represent data movement and messages that are passed through adaptors to other systems. The blue circles represent adaptors that are connected to a common interface table in a system. The red lines represent interfaces directly between any two systems.



These interfaces are generally SQL code used to extract the data from one system and load it into another system. As is obvious, both methods of integration can be very complex and difficult to maintain. The data may be in different formats in each of the systems, the interface code or the adaptors may need to change as each system is upgraded, the loads have to be done in a particular sequence to obtain the correct results, and the data itself may be inconsistent. Decisions have to be made regarding which application contains the correct data, how to deal with conflicts, and the frequencies of loads. There are some basic principles that will help streamline the process of integrating data among disparate systems.

1. **Try to keep the same type of data within a single application**, or at best, identify a single place where data is created and updated. This is the underlying concept of master data management efforts. All applications that reference that data should be "read only".
2. **Set up data standards.** Create naming standards and formatting standards for all systems across the enterprise. For example, all descriptions should be the same field length, telephone numbers should

all be in the same format (for example, countrycode.areacode.number.extension), punctuation should be eliminated, and abbreviations should be standardized.

3. **Create a Data Map.** This can be done in a spreadsheet, in a database, or by using database design software. The purpose of the Data Map is to show what each data element is mapped to in other systems and the “load instructions” for that data element. The data map is cross referenced for two-way interfaces. If using a spreadsheet, you would have a worksheet for each table with the attributes or columns of the table on the left of the spreadsheet (column A) with each interfaced system/table going across the top (Row 1). The first Application should be the current system. In the first intersection cell (B2), put the format of the data of the current system (i.e. varchar 10). After the current system is documented, allow 3 columns for each application to be integrated with the current system. In the second intersection cell (C2), put the table/column name that is the destination for the first data element in the first application to be integrated. In the third column (D2), put the format required for the first system to be integrated (i.e. varchar 25). In the fourth column (E2), you will document the transformation code required to get the data from the format in column B into the format required for Application B, Column D (i.e. rpad 15). Continue on until you have all the interfaces mapped and the transformations documented for each application to be integrated. Keep the data map current as systems are updated.
4. **Limit the interface to a “need to know” interface.** In other words, if an application does not need to use the information as a trigger for a procedure or an action within that system, do not bring it into the new system.
5. **Define the processes that create, read, or update each type of data** and put security and access controls in place so that the governance and ownership of the data is unambiguous.

Finally, evaluate all data that is integrated for completeness, consistency, and correctness between each source and each target. Validate that the correct number of records are transferred, the resulting data, and the reconciliation between each source and target so that the bridges you are building are stable enough to withstand change.

---

#### **Curious?**

For more information, please call **eprentise** at **1.888.943.5363** or visit **www.eprentise.com**.



#### **About eprentise**

**eprentise** provides transformation software products that allow growing companies to make their Oracle® E-Business Suite (EBS) systems agile enough to support changing business requirements, avoid a reimplementation and lower the total cost of ownership of enterprise resource planning (ERP). While enabling real-time access to complete, consistent and correct data across the enterprise, **eprentise** software is able to consolidate multiple production instances, change existing configurations such as charts of accounts and calendars, and merge, split or move sets of books, operating units, legal entities, business groups and inventory organizations.