



STANDARDS-BASED DATA INTEGRATION

US DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT



TABLE OF CONTENTS

INTRODUCTION	3
BACKGROUND.....	4
THE PRESENTING CONDITIONS.....	4
THE DYNACC·SYS APPROACH USING THE XABRA TOOL SUITE	5
DETAILED DESCRIPTION OF THE XABRA SOLUTION.....	6
CONCLUSION.....	9

Introduction

A major problem facing today's organizations is the challenge to produce the same answer for the same question for people working in different parts of the same organization or in different systems. Different units within the same organization use incompatible software systems, leading to multiple data silos that do not communicate with one another—they may represent the same basic data but differently. Data silos emerge organically because different business units implement their own projects independently, they occur due to mergers and acquisitions, or because of a need to exchange data with a Business Process Outsource (BPO) provider.

eXtensible Business Reporting Language (XBRL™¹) offers a solution that enables organizations to successfully integrate and exchange their data to achieve a unified view of all organizational data.

XBRL is an XML-based meta-data standard used to communicate business and financial information electronically. Created and maintained by XBRL International, Inc., the standard enables data to be entered once and flow seamlessly throughout financial business processes. Corporate data is mapped to authoritative taxonomies, then formatted into standardized output “instance documents.” Instead of treating financial information as a block of text — as in a standard internet page or a printed document — the XBRL identifying tag for each individual item of data is computer readable. The introduction of XBRL tags enables automated processing of business information by computer software, cutting out laborious and costly processes of manual re-entry and comparison. As described by XBRL International, “because computers can treat XBRL data ‘intelligently’, they can recognize the information in an XBRL document, select

A shared and structured vocabulary enables software to recognize the data for easy reuse and exchange.

it, analyze it, store it, exchange it with other computers and present it automatically in a variety of ways for users.”

(<http://www.xbrl.org/WhatIsXBRL/>)

Recent steps taken by the Securities and Exchange Commission (SEC) underscore the importance of

XBRL to the business community. In addition to establishing a voluntary filing program using XBRL, they are investing millions in the full development of XBRL taxonomies, modernizing systems to accept XBRL data and providing tools to investors to be able to examine corporate reports easily.

XBRL-based data integration provides multiple advantages:

- XBRL is a vendor-neutral, royalty-free standard. There is no vendor lock-in based on proprietary taxonomies.
- XBRL creates intelligent data that is usable by software programs for multiple purposes.
- XBRL has built-in data validation that minimizes errors during data transfer.

XBRL can enable legacy systems to operate through their intended life at low cost.

¹ The XBRL™ logo is a trademark or service mark of XBRL International, Inc. and is registered in the United States and other countries.



- XBRL minimizes the cost of adding new data integration points because mapping always conforms to the same XBRL meta-data standard.
- XBRL provides contextual information, such as the relevant timeframe, for each data item.

DynAccSys implemented an XBRL-based data integration solution for the US Department of Housing and Urban Development's (HUD) Federal Housing Administration (FHA). This paper discusses this solution.

Background

The US Department of Housing and Urban Development's (HUD) Federal Housing Administration (FHA) had two presenting processes that needed to be upgraded.

First, FHA maintains a general ledger that conforms to US government accounting standards using PeopleSoft Enterprise solutions. Disparate financial systems feed this general ledger—the FHA Subsidiary Ledger or FHASL. Since these feeder systems use a commercial chart of accounts, the format differs from the US Standard General Ledger (USSGL). To address this disparity, data from the feeder systems were converted to USSGL through a data warehouse, where custom interfaces were maintained for each USSGL non-compliant subsidiary ledger.

Two of these feeder systems are maintained by DynAccSys as an outsourced service provider. These systems are an accounts receivable (loan servicing) system and an accounts payable (property accounting and contract management) system. They maintain business and financial information on departmental assets.

Second, HUD maintains a centralized procurement management system that monitors commitments and obligations for federal contracts. The procurement management system must be updated with commitment, obligation, and disbursement data daily and must be in sync with the FHA general ledger. (This is a new requirement for FHA.)

The Presenting Conditions

There were several challenges associated with the old data integration solution:

- The data warehouse added a middle layer between the feeder systems and the FHA systems that was costly and error-prone. First, data was not sent directly from feeder systems—causing delays in update to the general ledger. Additionally, while HUD had developed a common interface to the data warehouse, each feeder system had to develop its own custom interface to feed it. Altering the format of the data that was exchanged required expensive and time-consuming re-programming to modify the custom interfaces.
- Data was sent only monthly, causing time-consuming, end-of-month reconciliation problems, given the construct and operation of the data warehouse.
- The FHA's general ledger data is maintained in a highly summarized form, without the supporting details. FHA staff had to rely on the organizations that maintain the feeder



systems to get detailed information for further analysis. Not only was this an inefficient process but it was prone to reconciliation challenges. Posting Models were evolving continually and all parties were not informed of changes, leading to unreconciled or improperly recorded accounts.

FHA wanted to:

- Eliminate the data warehouse to improve the efficiency and reduce the cost of its financial processes.
- Improve the quality and integrity of data.
- Improve the timeliness of the data by going from a monthly gathering of accounting activity to a daily receipt of cash transactions. Daily reconciliation of FHA cash accounts with Treasury provides better controls and enables a quicker month-end and year-end close.
- Report commitment and obligation data to the Departmental procurement system at the same time that data was entered into FHASL
- Have access to detail data that goes with its general ledger summary data for submission to other systems and for analysis.

The DynAccSys Approach Using the Xabra Tool Suite

In the presenting environment, data integration and re-use was hampered by the absence of a common standard. DynAccSys has leveraged the XBRL standard to solve this problem through its XabraTools Suite. Xabra enables data to be easily mapped to XBRL from detail-level transactions, journals, and business information in legacy systems. The data, once mapped, is easily re-purposed and can be consumed by other proprietary systems to create a complete customized, end-to-end data integration solution.

The Xabra Tools include:

- The Binder module to create XBRL data instances from a data source
- The Loader module to load the data from XBRL data instances to destination databases, and to generate fixed files or csv files
- The XReports module to create HTML or PDF reports from the data in XBRL data instances
- The Transformer module to summarize/translate/filter data in XBRL data instance documents
- The Xabra Engine execution environment to enable the above modules to work together to create a custom data integration solution



Xabra modules are customized by configuration files generated using a Graphical User Interface (GUI): the Xabra Constructor. Figure 1, shows how the Xabra Constructor is used to create the bindings from the database columns to the XBRL taxonomy elements, in order to create XBRL data instance documents from database data. The binding between the ‘AMT’ column and the ‘gl-cor:amount’ taxonomy element is highlighted.

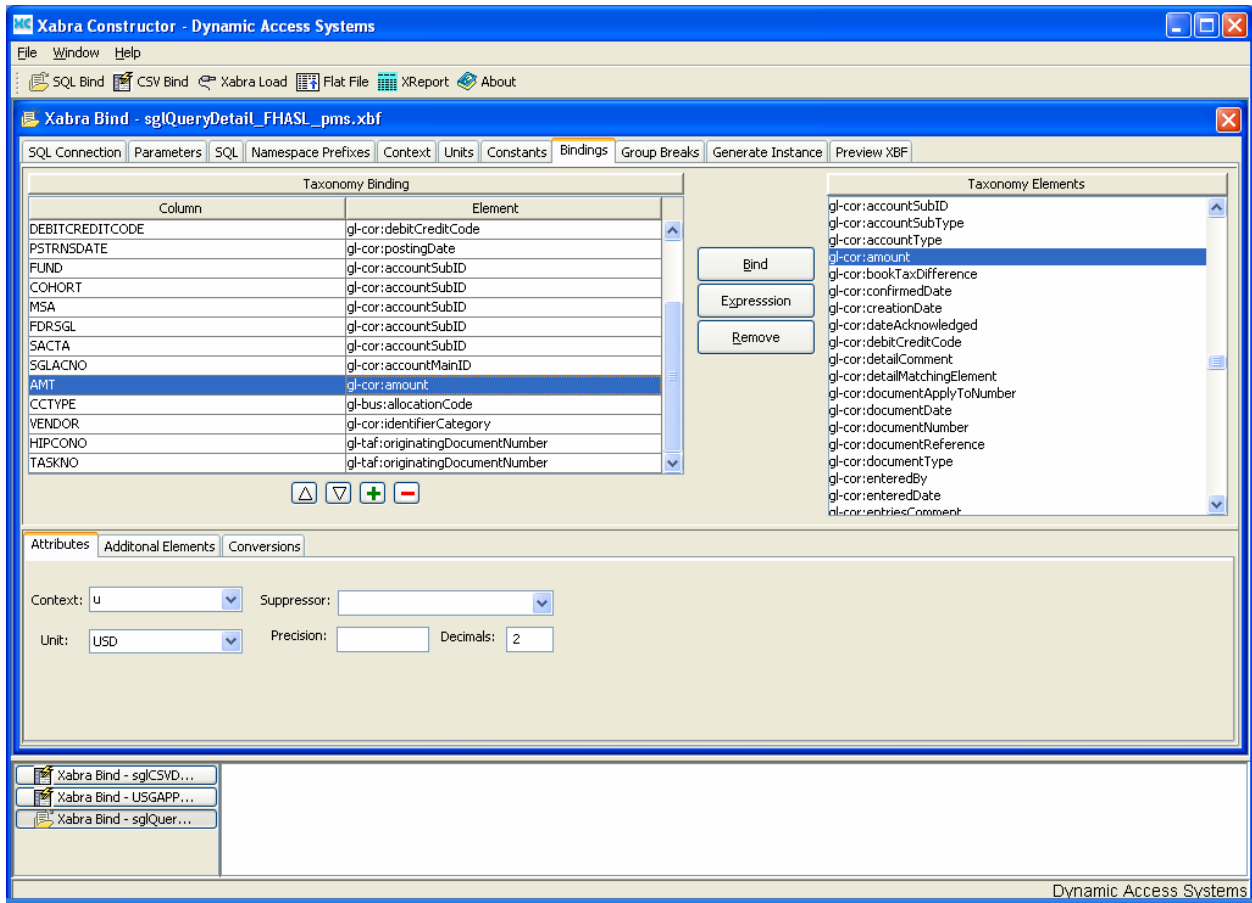


Figure 1 – The Xabra Constructor

Detailed Description of the Xabra Solution

For the aforementioned FHA implementation, the Xabra solution is used to standardize data, using the XBRL GL taxonomy (or the Global Ledger taxonomy) from the accounts receivable and accounts payable feeder systems. The XBRL GL taxonomy provides the meta-data needed to standardize business and financial information at the detail level. This taxonomy is used to standardize summary journal entries in an XBRL GL data instance document that is submitted directly to FHA’s PeopleSoft general ledger on a daily basis. The standardized detail data is retained for analysis. Further, a subset of these data are re-purposed and fed directly to the



procurement management system at HUD on a daily basis, to monitor commitments and obligations for Federal contracts.

As a result of the DynAccSys Xabra solution, FHA achieves:

- Daily cash activity information directly loaded to FHA’s general ledger, bypassing the data warehouse. The previous system only loaded monthly accounting activity to the general ledger.
- Daily funds control activity loaded directly to FHA’s general ledger and to HUD’s procurement feeder system, bypassing the data warehouse.
- Monthly, quarterly, and annual reporting at the summary and detail levels.
- Automated reconciliation with multifamily mortgage claims data, eliminating manual steps and data re-entry.
- Single source use for multiple reporting requirements (contracts, budget execution, and business analysis), eliminating duplicate data entry, duplicate data processing, and reducing time-consuming reconciliations.

An example of using Xabra modules to feed the FHA’s general ledger is shown in Figure 2.

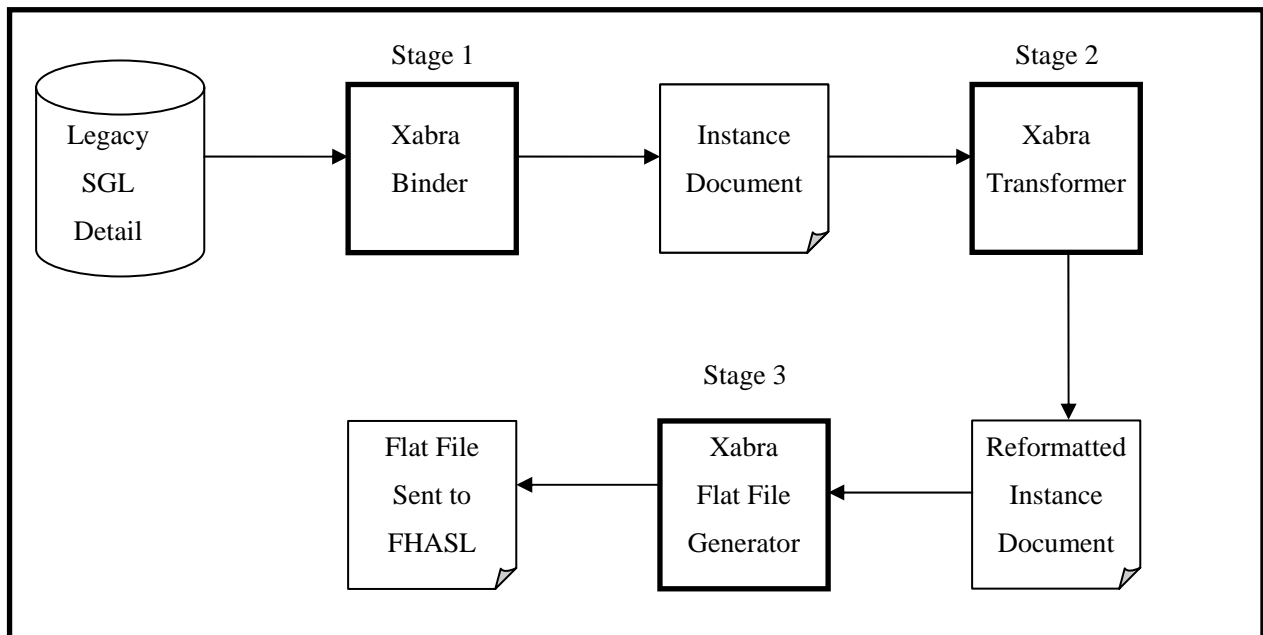
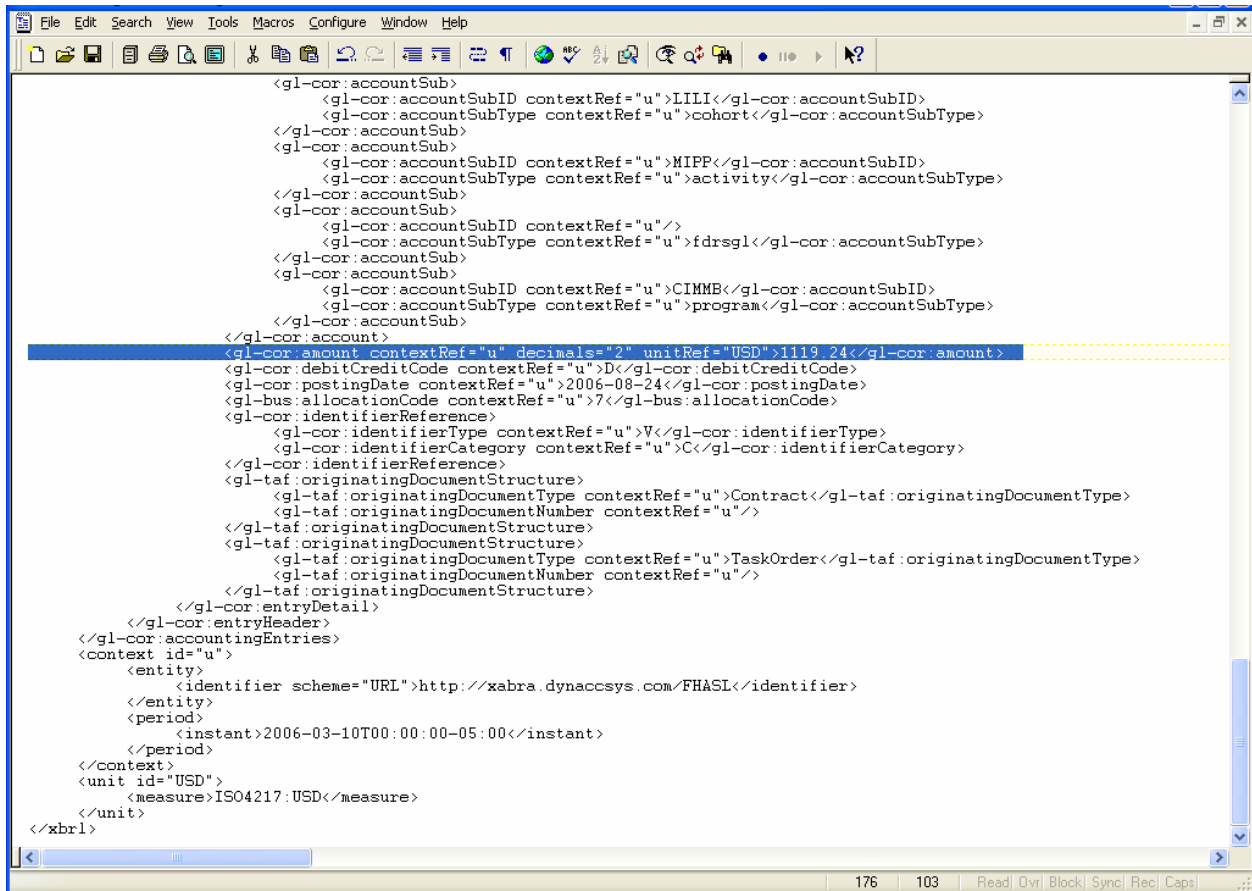


Figure 2 – Shows How Xabra Modules Use Legacy Accounting Transactions to Create and Reformat XBRL Data Instance Documents to Generate a Flat File That is Consumed by FHA’s General Ledger.

In stage 1, the Xabra Binder consumes legacy Standard General Ledger (SGL) detail transactions to create an XBRL data instance document based on the XBRL GL taxonomy. Data is retrieved from the legacy systems using SQL queries. The legacy data is mapped to the XBRL GL

taxonomy using predefined data mappings that identify how to go from the flat data structure of the legacy systems to the hierarchical structure of the XBRL GL taxonomy. The output of the Xabra Binder is an XBRL GL data instance document that has the SGL detail transactions.



```

<gl-cor:accountSub>
  <gl-cor:accountSubID contextRef="u">LILI</gl-cor:accountSubID>
  <gl-cor:accountSubType contextRef="u">cohort</gl-cor:accountSubType>
</gl-cor:accountSub>
<gl-cor:accountSub>
  <gl-cor:accountSubID contextRef="u">MIPP</gl-cor:accountSubID>
  <gl-cor:accountSubType contextRef="u">activity</gl-cor:accountSubType>
</gl-cor:accountSub>
<gl-cor:accountSub>
  <gl-cor:accountSubID contextRef="u"/>
  <gl-cor:accountSubType contextRef="u">fdrsgl</gl-cor:accountSubType>
</gl-cor:accountSub>
<gl-cor:accountSub>
  <gl-cor:accountSubID contextRef="u">CIMMB</gl-cor:accountSubID>
  <gl-cor:accountSubType contextRef="u">program</gl-cor:accountSubType>
</gl-cor:accountSub>
</gl-cor:account>
<gl-cor:amount contextRef="u" decimals="2" unitRef="USD">1119.24</gl-cor:amount>
<gl-cor:debitCreditCode contextRef="u">D</gl-cor:debitCreditCode>
<gl-cor:postingDate contextRef="u">2006-08-24</gl-cor:postingDate>
<gl-bus:allocationCode contextRef="u">7</gl-bus:allocationCode>
<gl-cor:identifierReference>
  <gl-cor:identifierType contextRef="u">V</gl-cor:identifierType>
  <gl-cor:identifierCategory contextRef="u">C</gl-cor:identifierCategory>
</gl-cor:identifierReference>
<gl-taf:originatingDocumentStructure>
  <gl-taf:originatingDocumentType contextRef="u">Contract</gl-taf:originatingDocumentType>
  <gl-taf:originatingDocumentNumber contextRef="u"/>
</gl-taf:originatingDocumentStructure>
<gl-taf:originatingDocumentStructure>
  <gl-taf:originatingDocumentType contextRef="u">TaskOrder</gl-taf:originatingDocumentType>
  <gl-taf:originatingDocumentNumber contextRef="u"/>
</gl-taf:originatingDocumentStructure>
</gl-cor:entryDetail>
</gl-cor:entryHeader>
</gl-cor:accountingEntries>
<context id="u">
  <entity>
    <identifier scheme="URL">http://xabra.dynaccsys.com/FHASL</identifier>
  </entity>
  <period>
    <instant>2006-03-10T00:00:00-05:00</instant>
  </period>
</context>
<unit id="USD">
  <measure>ISO4217:USD</measure>
</unit>
</xbrl>

```

Figure 3 – Section of the XBRL Data Instance Document with SGL Detail Transactions.

Figure 3 shows a section of this data instance document to illustrate how the XBRL data instances provide intelligent data and contextual information. One of the elements used to tag the monetary amounts by the name of ‘gl-cor:amount’ is highlighted. This element has the value of ‘1119.24’, which was obtained from the legacy data according to the highlighted binding created using the Xabra Constructor. The element below it is the ‘gl-cor:debitCreditCode’ element, which has a value of ‘D’ for debit indicating that the amount ‘1119.24’ is a debit. This is a simple example of how the XBRL data is ‘intelligent’. The ‘contextRef’ attribute of the ‘gl-cor:amount’ element refers to the context ‘u’ that is defined at the bottom of the data instance. The context ‘u’ states that the value ‘1119.24’ of the ‘gl-cor:amount’ element is associated with the entity (for example a business) with the URL ‘http://xabra.dynaccsys.com/FHASL’, and that the date/time of this value is ‘2006-03-10T00:00:00-05:00’. Other contextual information associated with the ‘gl-cor:amount’ element can be added to this context definition if needed.



In stage 2, the XBRL data instance document with the SGL detail transactions is consumed by the Xabra Transformer module. The Xabra Transformer uses the Xabra Summarizer component to group the data elements and then uses a filtering process to filter out irrelevant elements to create a reformatted XBRL data instance document.

In stage 3, the reformatted XBRL data instance document is consumed by the Xabra Flat File Generator module. The Xabra Flat File Generator module uses predefined mappings from the XBRL GL taxonomy to a fixed-width file format acceptable to the FHA's general ledger to create a fixed-width file that is consumed by this general ledger.

Conclusion

Using XBRL for data integration has advantages:

- No vendor lock-in
- The use of intelligent data that is machine readable
- Built-in data validation
- Reduced cost of adding new data-integration points
- Availability of contextual information.

These advantages result in the ability to create low-cost data integration solutions that provide decision maker with accurate information in a timely manner, to make key decisions in a rapidly changing environment based on current knowledge of organizational processors.

DynAccSys leveraged some of these advantages to create a low cost, highly efficient data integration solution to provide the FHA with daily cash information. As a result, not only is it easier for FHA to receive relevant accounting information directly from feeder systems bypassing the data warehouse, but it has eliminated time-consuming reconciliation and research efforts. Data from feeder system, is easily interfaced to multiple systems without further translation or data re-entry. Accompanying business processes are streamlined, translation changes can be managed quickly and easily, with the overall result of more timely, accurate, complete, and reliable data.

The success of this implementation is proof that XBRL delivers on its promise of being a good choice for electronic data exchange and that the DynAccSys Xabra solution is capable of leveraging the benefits of XBRL to create customized data integration solutions that overcome many of the data integration challenges that exist today.