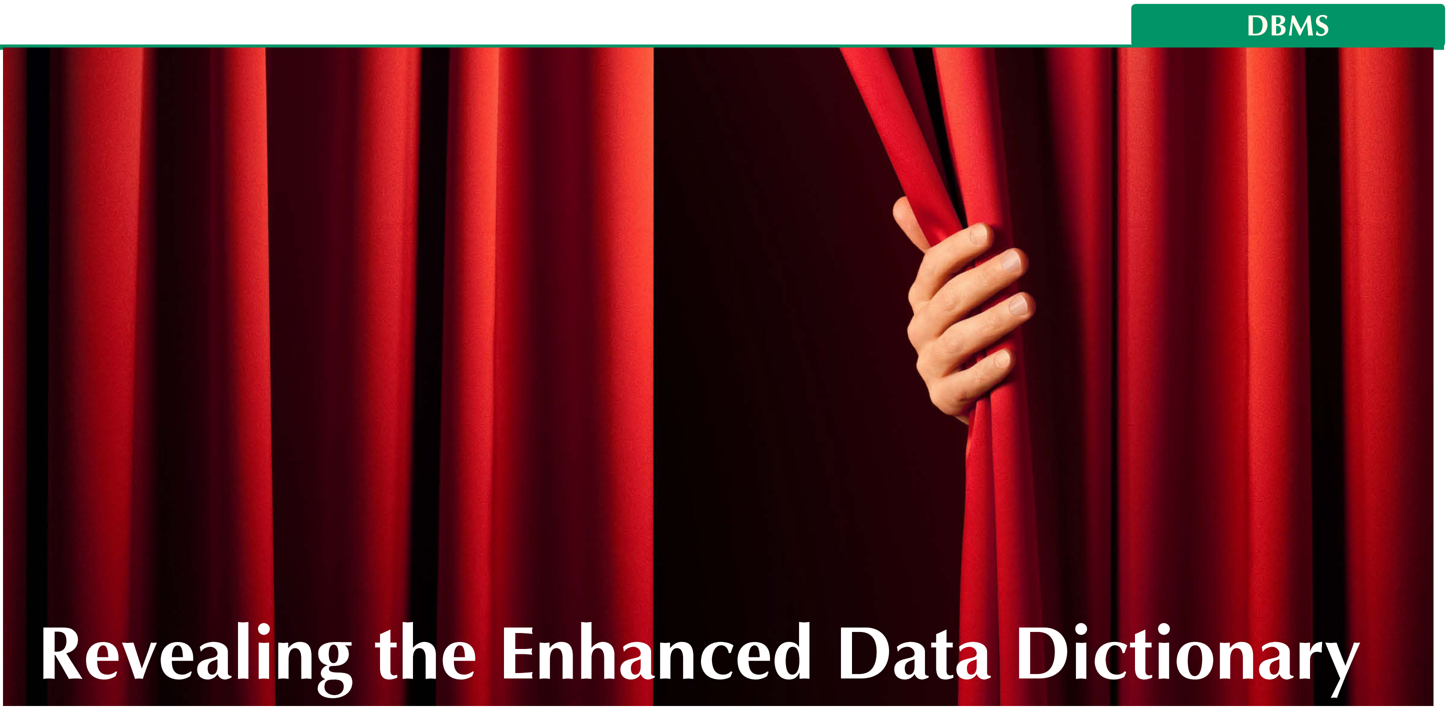


Revealing the Enhanced Data Dictionary

BBBj® is a very powerful and flexible language, giving developers countless options for accessing and managing data. READ/WRITE RECORD provides the ability to control the appearance of the data down to the individual byte level. String templates add the ability to describe file record data as a set of fields of particular data types to make it easier to build and interpret the byte-oriented records used by the READ/WRITE RECORD calls. SQL goes a step further in providing a simple, high level, but very powerful querying language that makes it easier to acquire specific data and to gather that data from multiple sources into a single result. At the lowest level, all of these methods manipulate and access data in essentially the same fashion (with the exception of ESQL tables), writing and reading arrays of bytes to and from data files.

The Data Dictionary

A data dictionary is a layer that sits between the data file and the SQL engine. The data dictionary describes the entities within a particular database. Each dictionary describes a single database and includes information the SQL engine uses to perform queries and other database related operations on tables, views, stored procedures, sequences, user permissions, and roles (groups). In BBj 13.0 and higher, BASIS provides two dictionary formats to choose from: legacy and enhanced.

Looking Back at the Legacy Format

The legacy format is the dictionary format used by the SQL engine in PRO/5®, the BASIS ODBC Driver®, and earlier versions of BBj. The format has served its purpose well for many years providing BASIS developers with SQL access to their data files. However, this format has some limitations. Four of the most notable examples are **1)** a 16-character limit on table, view, and column names, **2)** no way to define foreign key relationships (with the exception of ESQL tables), **3)** the need to specify a date format and date suffix to define columns as DATE type, and **4)** the inability to specify a specific precision and scale on NUMERIC type columns.

By Jeff Ash
Software Engineer

Introducing the Enhanced Data Dictionary

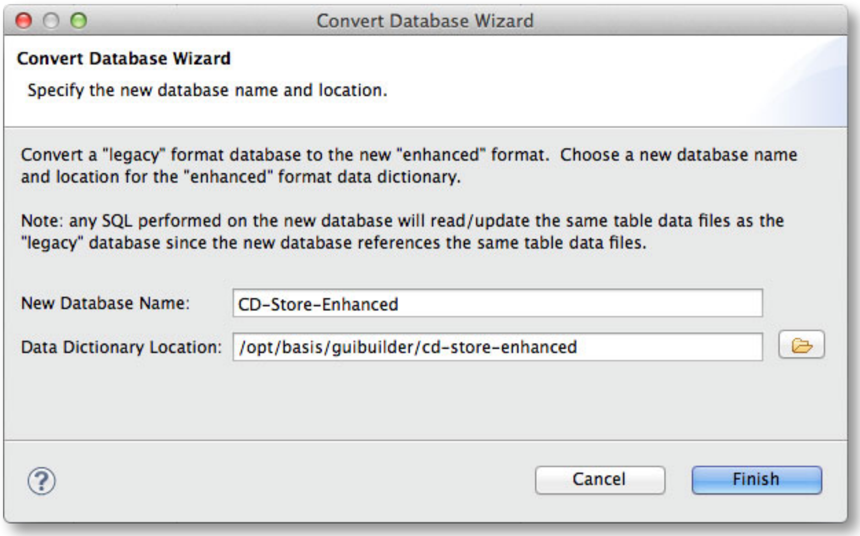
The enhanced format data dictionary overcomes the limitations of the legacy format. The following is a list of some of the key features in the enhanced format:

- **A more than 15-fold increase, to 255 characters, of the limit for entity names** – makes it possible to specify more descriptive table, view, and column names.
- **The ability to define foreign key relationships between the tables in the database** – information very useful for query building tools and to assist users in understanding the relationship between the tables in a database. It is important to note that this does not provide referential integrity (except on ESQL tables), but is simply informational.
- **Date formats defined at the column level** – embedded within the data dictionary so there is no need to configure the database with date format information. Another benefit is that column names do not need to adhere to any particular naming convention. For those who have multiple date formats used in their database, this makes it easy to accommodate that structure.
- **NUMERIC columns get precision and scale** – NUMERIC type columns contain a dictionary length that describes the length of the raw data at the file record level, but now additionally contain a precision and scale. This gives the database administrator the ability to describe the raw data, and exactly how the numeric values should look to the outside world. For example, a NUMERIC value in a BBj data file could be defined with a dictionary length of 12. But since BBj stores NUMERIC values as a string representation of that number, how would a third party application know how many decimal places were allowed? The answer is to provide precision and scale that allows a way to define this metadata.

Using the Enhanced Format

To begin using the enhanced format, create a new database from scratch, or convert an existing legacy format database to the enhanced format. Conversion is simple using the Enterprise Manager's database conversion wizard, shown in **Figure 1** (you can also run a conversion using the Admin API that is outside of the scope of this article). Converting your existing data dictionary to an enhanced data dictionary is as simple as using the following steps in the database conversion wizard:

1. Log into the Enterprise Manager.
2. Double-click on the Databases node.
3. Select the database to convert.
4. Click the "Convert to Enhanced" button.
5. Specify a name for the new database.
6. Specify a location for the new data dictionary.



Convert Database Wizard

Convert Database Wizard

Specify the new database name and location.

Convert a "legacy" format database to the new "enhanced" format. Choose a new database name and location for the "enhanced" format data dictionary.

Note: any SQL performed on the new database will read/update the same table data files as the "legacy" database since the new database references the same table data files.

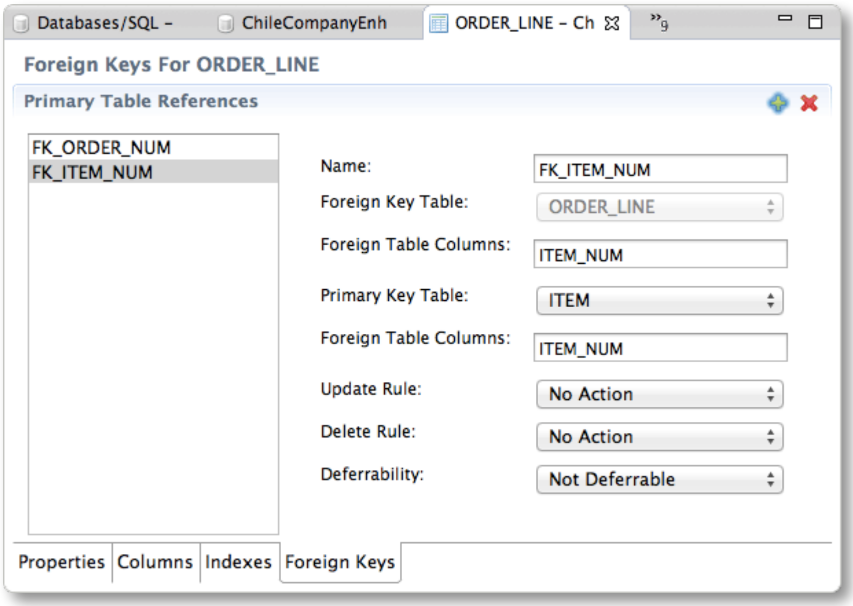
New Database Name: CD-Store-Enhanced

Data Dictionary Location: /opt/basis/guibuilder/cd-store-enhanced

Figure 1. The database conversion wizard

The conversion creates a new data dictionary in a new location, while preserving your original data dictionary. (it does not modify or overwrite your legacy format dictionary). It will set the date format on all columns that match any date format configuration settings on the legacy database and store these in the enhanced format column definitions automatically. This conversion process creates the new database and points it to the existing table data files. As a result, you can continue to use the legacy database while testing the new enhanced format database. Just keep in mind any UPDATE, DELETE or INSERT statements will modify the table data for both database definitions since they both use the same table data files.

Exploring the features of the enhanced format dictionary is as easy as double-clicking on the enhanced format database, then selecting the Tables tab. Double-click on a table to open up the table editor where you can set the precision and scale, add foreign key references, and specify date formats for specific columns. **Figure 2** shows Enterprise Manager's foreign key management user interface for an enhanced version of the Chile Company database. This screen lists the currently defined foreign keys and allows developers to add, remove, and modify existing defined relationships between tables.



Foreign Keys For ORDER_LINE

Primary Table References

FK_ORDER_NUM
FK_ITEM_NUM

Name: FK_ITEM_NUM

Foreign Key Table: ORDER_LINE

Foreign Table Columns: ITEM_NUM

Primary Key Table: ITEM

Foreign Table Columns: ITEM_NUM

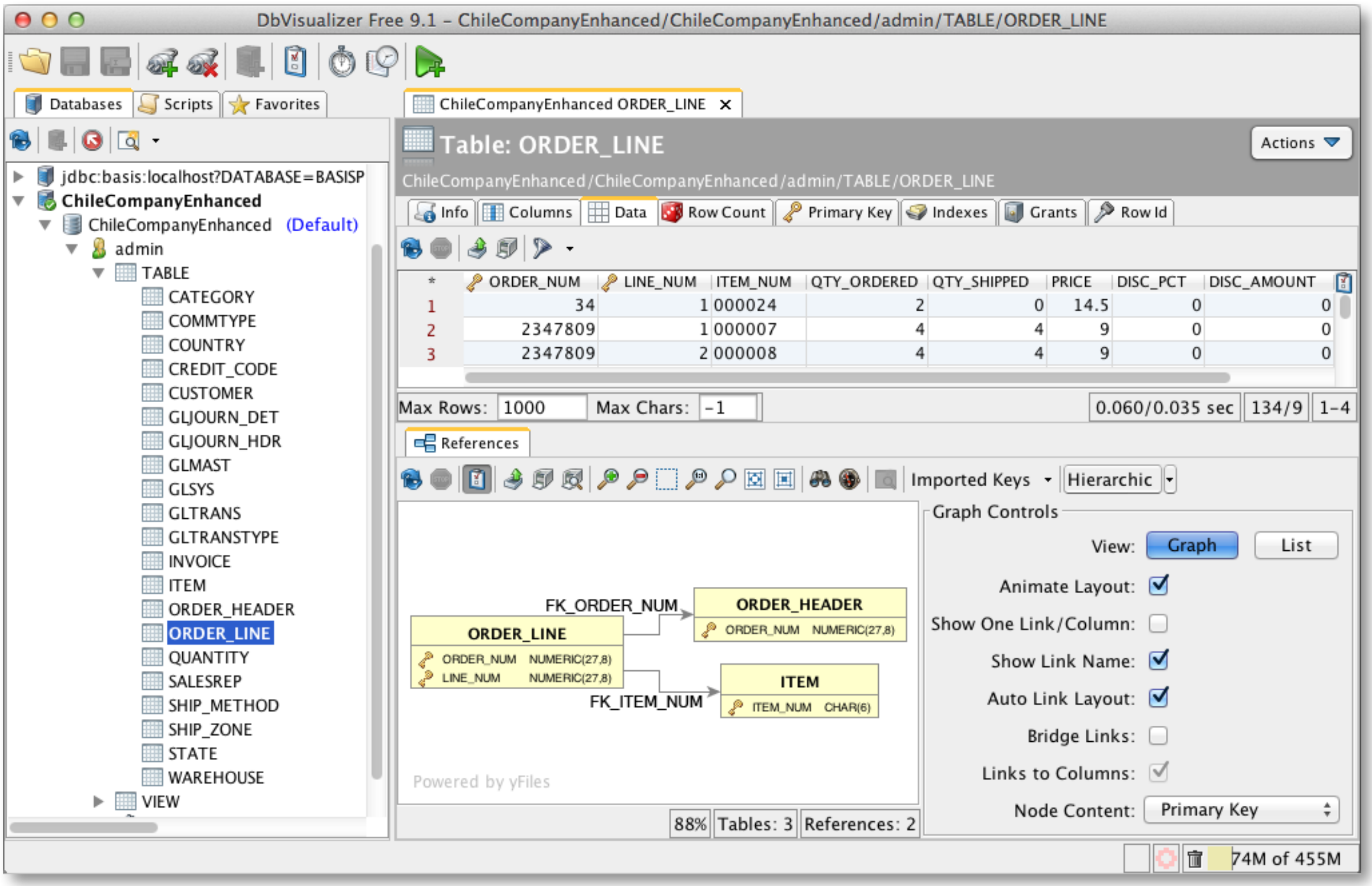
Update Rule: No Action

Delete Rule: No Action

Deferrability: Not Deferrable

Figure 2. Enterprise Manager's foreign key management

After defining foreign key relationships between the ORDER_LINE table and the ORDER_HEADER and ITEM tables on their common columns, we can view the relationships in a third party tool, such as DbVisualizer (www.dbvis.com). **Figure 3** shows a screenshot of DbVisualizer examining the ORDER_LINE table, displaying its data and a graphical representation of the foreign key relationships that we just created.



DbVisualizer Free 9.1 – ChileCompanyEnhanced/ChileCompanyEnhanced/admin/TABLE/ORDER_LINE

Table: ORDER_LINE

ChileCompanyEnhanced/ChileCompanyEnhanced/admin/TABLE/ORDER_LINE

*	ORDER_NUM	LINE_NUM	ITEM_NUM	QTY_ORDERED	QTY_SHIPPED	PRICE	DISC_PCT	DISC_AMOUNT
1	34	1	000024	2	0	14.5	0	0
2	2347809	1	000007	4	4	9	0	0
3	2347809	2	000008	4	4	9	0	0

Max Rows: 1000 Max Chars: -1 0.060/0.035 sec 134/9 1-4

Graph Controls

View: Graph

Animate Layout: ☑

Show One Link/Column: □

Show Link Name: ☑

Auto Link Layout: ☑

Bridge Links: □

Links to Columns: ☑

Node Content: Primary Key

88% Tables: 3 References: 2

74M of 455M

Figure 3. DbVisualizer showing the foreign key references for the ORDER_LINE table

Summary

BBj 13.0 introduced a new data dictionary format to expand the capabilities of the BBj database. These new features provide better interoperability with third party applications with the addition of foreign key relationships and precision and scale, increased flexibility with the removal of name limits, and simplified database configuration by storing date format information in the column definitions. This suite of new features makes the new enhanced format data dictionary the choice for new database projects, and something to seriously consider with existing legacy applications as it further empowers third party access to existing BBj data. The biggest benefit of converting your existing data dictionary today, is that your data dictionary will now be ready to handle any and all of the new features that your organization will need in the future, but will continue to work exactly like the original data dictionary until you begin to utilize the new features and functions. ■