



## CCAPRINT

A Newsletter Excerpt for System 1032 Users

August 10, 2004

USE OF AND ACCESS TO PRODUCTS AND FEATURES ARE IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THE USER'S SOFTWARE LICENSE. THE PRESENTATION OF MATERIAL HEREIN DOES NOT, IN ANY MANNER, MODIFY SUCH TERMS AND CONDITIONS.

## Back to Basics, Part 2 – Data Values

By Tym Stegner

In the June 2004 article of this series, we reviewed the components of the System 1032 environment. In this article, we focus on the basic unit of the DBMS environment, the data value. While a data value is the basic building block of the environment, each data value is defined and stored within a dataset (or table) structure comprised of columns and rows. This article examines your data values and helps you determine what columns you want in your datasets.

### Describing Data Values

To store data values, such names, addresses, appointments, salary figures, sales figures, photo IDs, results of tests, temperatures, and taxes, you must have a description of the data you are storing. These descriptions are called data definitions and are coded using PL1032 DDL, the command language of the System 1032 DBMS and its Data Definition Language. In System 1032 the value definitions are called attribute definitions, where the **attribute** is the column or field of a tabular table.

At a minimum each attribute has a primary name and a data type. Many other options exist to further quantify the type of data stored by the attribute. Before these qualities can be defined, however, you must first decide the most appropriate data type for each attribute.

### Choosing the Appropriate Data Type

How efficiently System 1032 can store, manipulate, and retrieve data values is based on the attribute types defined. Although you may define attributes in a particular order, this might not correlate to the arrangement of data within the dataset. System 1032 compacts data elements together to achieve efficient storage, thus maximizing the amount of data retrievable in the least number of IO operations. In light of this capability, it behooves you to choose appropriate data types to store your data values.

## Reviewing Data Types

The data type of an attribute determines what and how the data value for that attribute is stored and displayed and how it is handled when you make queries against your data. It also determines default validations that System 1032 can perform while you enter or update data.

Whether you are creating a dataset for the first time or from data you already have, or you are reviewing how the datasets you have had for years at your site are defined, think of data as falling into roughly the following classifications:

- Text
- Numeric
- Time
- Composite

## Considering Text Data

While you can store most any data as text, it is wise to make use of the intrinsic System 1032 data validation capabilities to ensure proper data is input, by choosing the data type that best describes the data values you want to store—even text data can be handled in several ways. The System 1032 text and related data types are described in Table 1.

**Table 1. Text data types**

<b>Data type</b>	<b>Stores...</b>	<b>Especially good for defining...</b>
Text <i>n</i>	A series of characters up to a fixed number of characters. Stores the characters and additional spaces, as needed, to fill the space.	All kinds of names: addresses, breeds of dogs, course descriptions, titles of books, drug names, test names, location points, model numbers.
Text Varying	Any series of characters, including embedded formatting. Stores only the characters specified.	Narrative descriptions of: customers complaints, patient health treatment, airplane specifications, drug trial setup, and letters written.
Binary Varying	Unstructured data, particularly large binary data. Parallel in function to Text Varying, but of larger capacity.	Pictures of the galaxies, photo IDs, internal images of the human body.

## Text *n* and Text Varying Data

When choosing text data types, you must decide between fixed length text (Text *n*) or varying length text (Text Varying). Storing data as Text Varying is convenient, but not necessarily efficient. Fixed text values are stored as part of the internal record, text varying data is stored in a completely separate structure within the dataset, tagged with the proper record number, because System 1032 does not know in advance how much space to set aside.

The intrinsic overhead associated with Text Varying amounts to about 16 characters of space per attribute, per record. Keep in mind:

- If the values to be stored in the attribute vary widely in length, Text Varying saves space in the dataset.
- If there is only a small variation, fixed-length Text noticeably saves time and processing, although not necessarily disk space.

In a study of this topic, I found the break-even point to be around the 80-character mark. If your text values are 80 characters or fewer, just use fixed-length Text for your data, instead of incurring the overhead of Text Varying.

### Binary Varying Data

Binary Varying, also known as a Binary Large Object (BLOB, and similar to a MS Memo field), is the catch-all data type for very large value sizes. There is much flexibility in the storage arrangements; you can choose to store the entire data in a single segment or use multiple same-sized segments for the data value.

### Considering Numeric Data

The System 1032 Numeric data types are described in Table 2.

**Table 2. Numeric data types**

Data Type	Stores...	Especially good for defining...
Integer [Double]	Positive and/or negative numbers without fractions.	Any enumerated object: codes, counts of items, bit masks, ages.
Decimal [m.f]	Positive and/or negative numbers with a fraction, such as money values. Values are exact, but have additional storage overhead. Default precision is 12 digits, with a maximum of 31 digits.	Money or precise fractions.
Real [Double]	Positive and/or negative numbers with fractions. Stored value may approximate the actual value. Supports exponents.	Scientific measurements such as: cell size, manufacturing specifications, astronomical distances.
Logical	True or False; Yes or No; 1 or 0.	Binary codes or conditions.

Proper classification of numeric data is very important in System 1032 data definitions. Although System 1032 is very forgiving if a value is not stored in its most efficient state, your applications pay a price of extra overhead from decreased storage efficiency and, sometimes, with decreased precision of values derived from attributes used in calculations.

## Considering Times and Dates

System 1032 can store date and time values, as shown in Table 3. Storing dates and times as such gives you greater flexibility to manipulate and calculate against the values.

**Table 3. Date and time attribute definitions**

Data type	Stores
Date_Time	Calendar date, including time-of-day. Default precision is minutes.
Date	Calendar date. Default precision is days.
Time	A duration of clock time, range limited to 24 hours. Default precision is minutes.
Time_Span	A duration of time. Default precision is minutes.

The advantage of storing dates and times as one of the data types in Table 3, instead of Text *n*, is that you can add limits in the definitions to prevent errors at the time of data entry. You can also add and subtract the date-time values when you make queries, which is not possible when you define them as text. Details of how to do this will be discussed in a later article on System 1032 fundamentals.

## Considering the Composite Data Types

Before the introduction of ODBC with its heavy reliance on relational data, the composite data types were a very important means of storing related data. You may be storing data in Arrays, Groups, or an Array of Group at your site. The System 1032 ODBC Driver can handle these data types, although you probably would not include them in the design of a new application.

**Table 4. Quick overview of the composite data types**

Data Type	Stores...	Especially useful for storing...
Array	A compound attribute containing an enumerated series of data values of the same data type and qualities. Supports multiple bounds and index ranges. Attributes of types Varying Text and Binary Varying cannot be arrayed.	A series of similarly defined values having the same data type and options, or logical sub-records within a record, such as item lists, options, contact lists, component lists.
Group	Aggregates a list of one or more attributes of similar or mixed data types into a named entity. Array of group data type provides in-record data structures.	Data that will always be retrieved as a unit. Unlike normal attribute consolidation, elements of a group are stored together, for example: addresses, key codes.
Array of Group	An enumerated list of aggregated data.	Logical sub-records within a record; for example: list of movie stars, bill of materials.

## Introducing the Special Data Types

The data types listed in Table 5 are called special because you do not supply the data values; System 1032 does that. You simply define the attribute when you define the dataset; the values are filled in during record entry or update. Database-generated data is stored on a per-record basis. You have Read-only access to Special data type values.

Figure 5. System 1032 generated attribute definitions and values

Data type	Purpose
Date[_Time] of Change	Tracks date (and time) of record update. Initially Missing.
Date[_Time] of Delete	Tracks date (and time) of record deletion. Initially Missing.
Date[_Time] of Entry	Specifies date (and time) of record entry.
Integer of Identification	Unique integer value per record.
User of Change	Tracks the user name that last updated the record. Initially Missing, equivalent of Text 12.
User of Delete	Tracks the user name that last deleted the record. Initially Missing, equivalent of Text 12.
User of Entry	Specifies the user name that originally entered the record, equivalent of Text 12.

## In Summary

The appropriate selection of data type to represent the intended use of data values can give you great benefits. The benefits include the use of the Logical data type to store flag data (true, false, yes, no, 1, 0); integers in place of text codes (the codes can be stored in another reference dataset and translated at report time); and the appropriate precision for date-time data types can represent exactly the date-time information desired.

Data values are the be-all and end-all of any database management system, because data is what you are managing, updating, accessing, and displaying. System 1032 offers many features of its dataset definition to enable you to accurately store and manage various types of data.

© 2004 Computer Corporation of America  
500 Old Connecticut Path, Framingham, MA 01701