CCAPRINT

A Newsletter Excerpt for System 1032 Users

November 15, 2004

Back to Basics, Part 4 – Enabling Indexing

By Tym Stegner

In the previous articles of this *Back to Basics* series, we described the components and options that define attributes for storing correctly in System 1032 datasets. In this article, we examine the class of attribute options that enable indexing. The attribute definition option that supports indexing is KEYED.

Attributes are indexed primarily for use in a FIND command query. A FIND command query performed against one or more attributes defined with the KEYED option enables the fast selection of records without System 1032 reading the underlying data records, because key information is stored separately in optimized structures known as key tables. (Note: The SEARCH command query is used against attributes that are not defined with the KEYED option.

All data types support indexing. The KEYED option has additional options based on data type.

Keying Text Data Types

Text and Text Varying attributes are by far the most commonly indexed attributes, closely followed by Date/Time attributes. You can index fixed-length text attributes from 1 to 80 characters.

```
KEYED [[IGNORE_CASE | USE_CASE][MULTINATIONAL | NO_MULTINATIONAL]] |
[[n:n]]
```

For text attributes, three sets of optional parameters apply:

- The USE_CASE and IGNORE_CASE options let you index the text value in a case-sensitive mode or case-insensitive mode, respectively. The default setting, IGNORE_CASE, causes System 1032 to store the attribute in uppercase mode.
- The MULTINATIONAL and NO_MULTINATIONAL options let you index text values using an alternate character set, as controlled by the System 1032 logical name S1032_NCS. This allows for language-specific sorting and retrieval. The NO_MULTINATIONAL option is the default setting.
- The [n:m] option lets you index a substring of a Text or Text Varying attribute. The total length of the substring can be from 1 to 80 characters. You can specify only one substring per attribute for indexing, which cannot be combined with a standard key.

Keying Compound Data Types

```
KEYED [FOR_ALL | BY_ELEMENT]
```

For arrayed attributes, the FOR_ALL and BY_ELEMENT options specify the dimensional granularity by which an array is keyed. The FOR_ALL option lets you query to match any array element that qualifies. Whereas, the KEYED BY_ELEMENT option lets you query to matach a specific array element. The FOR_ALL and BY_ELEMENT options are mutually exclusive.

Note: For grouped attributes, you cannot key the group itself. If you want to make a query based on the entire group value, you must define a composite attribute that contains all the data from the individual group elements. Use of Record Descriptors and trigger procedures can automate populating and updating composite attribute values, which will be discussed in a later article.

Keying Binary Varying Data Type

```
KEYED [[n:m]]
```

You can key a Binary Varying attribute to index up to 40 consecutive bytes of the first 80 characters of the value.

Keying Other Data Types

The remaining data types--Integer, Real, Decimal, Date/Time, Logical, Special--support indexing as well. Take care in query commands to specify values using the proper precision to locate the desired results.

Performance Considerations

The query results from keyed attributes are returned much more easily and quickly than attributes without keys. However, records with many keyed attributes are updated more slowly than those with fewer keyed attributes, due to the additional data structures that must be updated for the record.

Arrayed attributes defined with the BY_ELEMENT option are updated more efficiently than an array having the FOR_ALL option. However, the BY_ELEMENT option requires additional data storage for the attribute.

In Summary

You can not only store and represent the data you enter in System 1032 with the ATTRIBUTE command options, but you can also specify the data that you want retrieved quickly. The KEYED option of the ATTRIBUTE command enables System 1032 to index attribute values for faster query results. Not all attributes in a record should be keyed, but you might consider keying the values you query most frequently.

500 Old Connecticut Path, Framingham, MA 01701