

VKEYED Files – New Features for a New File Type

By *Chris Hardekopf*

BASIS is currently designing a new file type called VKEYED, which will provide several important features:

- Define variable length records
- Dynamically insert, remove, and change key definitions
- Address keys by name rather than by key number
- Store the record template inside the file for programmatic retrieval

Variable length records, as the name implies, allows records to be of different lengths within the same file. As a result, the file may be smaller than if all of the records were the same size, especially when the contents of the records differ in size by significant amounts. This new field type is a perfect choice for storing BLOBs since the type of information stored in the binary field typically varies significantly in size. Storing video clips in a BLOB field is a good example; video clips can vary greatly in size because of their resolution, duration, compression, etc. Storing video clips in an MKEYED file requires that the size of each record accommodate the largest clip, which could waste valuable disk space. By comparison, the new VKEYED file is significantly smaller since each record is sized to fit only the clip stored in that record. It is worthwhile to mention that record access may be slower using variable length records rather than using fixed size records.

In order to perform key manipulations on the new file type, developers must lock the file to avoid potentially serious record locking problems and error recovery that could occur if a developer modified the file's keys while other users were accessing the file. Two basic rules for key numbers are **1)** key zero must be unique and **2)** usable key numbers are continuous. Furthermore, when removing a key, all keys numbers greater than the one removed will reduce by one; when inserting a new key, all keys with numbers equal or greater than the inserted key increase by one. Replacing a key will have no effect on key numbers.

Since key manipulations can change the key numbers for existing keys, clearly they can cause problems with reading records using key numbers. The introduction of key names for the new file type addresses this problem by optionally using a key name in place of previously used key numbers. Basically, key names deliver developers

two great benefits. First, using descriptive key names helps developers clarify code and improve its readability. Second, since key names do not change when removing and inserting keys, developers can trust the preserved code even after modifying unrelated keys.

The new file type will also allow developers to store additional information in the file; any information related to the file but not part of the file data or keys. A perfect example of this auxiliary information is the template for records in the file. Storing the template with the file provides an easy way to document the record format and use the file in general code, and greatly reduces the maintenance of applications that dynamically access the record template. When adding a new field to a database, only those programs that reference the new field need modification.

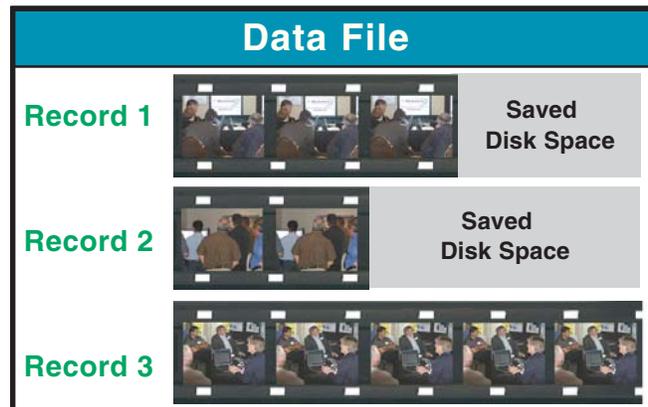


Figure 1. Data file showing saved disk space

Summary

As technology changes and impacts the types of data that customers need to store, BASIS meets this challenge in the 6.0 release of BBj®. The combination of MKEYED, XKEYED, JKEYED, and now VKEYED file types provides BASIS developers with the full myriad of today's options. Using the appropriate file type for your data will maximize performance and efficiency in your day-to-day operations. 



Chris Hardekopf
Software Engineer