Enterprise Computing Insights

What is a Data Set?

#6 IN A SERIES

All computer systems deal with two broad categories of soft copy materials – programs and data. Programs are the instructions that a computer follows to manipulate or process data. Just as a food processor makes raw ingredients more useful by processing them, so too a computer makes “raw” data more useful by ordering or manipulating it by following the instructions in a computer program; hence the term “data processing”. Both programs and data are usually kept on a storage device of some kind (magnetic disk, magnetic tape, CD, DVD, memory stick, etc.). On a personal computer they are stored on the device in the form of a file. The file is simply the named collection of characters that represents the program or data. But not a random collection of characters; instead a file has structure to it. The internal structure ensures that the computer can “read” the instructions in the program or the data items in the data file. It also helps ensure that the storage medium is used efficiently.

Personal Computer Sequential File

Most personal computer operating systems use byte stream or byte oriented files to store programs and data as files. A byte oriented file uses special characters to denote the end of a “line” of data. For example, if you write some text using a word processing program, the end of that line of text will be marked by a special character. It doesn’t matter if the line of text is 20 characters long or 60 characters long – the end of the line is marked by a special character. And within the file, the next line begins immediately after the special character. So, a 20 character line is actually 21 characters long and a 60 character line is actually 61 characters long – the lines internally are as long as they need to be and can vary from line to line. If you are familiar with using a word processing program, you insert a special character by pressing the “enter” key to go on to the next line. As you type, the structure of the data within the file is created as characters (including special characters) are added to the file. Most of the files stored on a personal computer like a word processing file, are sequential files. That is, the only thing known about the file for sure is where it begins. To find anything in the file, you always start at the beginning and process sequentially forward from there – to search for a character string for example.

A file is given a name in order to be able to refer to it. There are rules for naming files. The rules determine what characters can or cannot be used in the name, the length of the name, and possibly the structure of the name. On a personal computer, the structure of a file name is basically a name followed by a period followed by a three character “extension” (i.e. “name”.”extension”). The three character file extension is usually used to provide an indication as to the content (and therefore the structure) of the file. So an extension of “doc” is typically used to indicate a word processing document whereas an extension of “jpg” is used to indicate a file contains compressed data that represents an image. The extension names represent conventions and are not enforced by the computer or its operating system.

An enterprise server needs to store programs and data just as a personal computer does. But the naming conventions and internal structure used to store data is different from that used to store data on a personal computer. Enterprise server data is stored in a “data set”, not a “file”. An enterprise server and its operating system provide for several different types of data sets. In this article, we will examine two widely used data sets – sequential data sets and partitioned data sets. And we will briefly discuss VSAM data sets[1] and note that there are some “special” data sets as well.

[1] See ECI No. 7

Sequential Data Set

A sequential data set on an enterprise server shares many of the characteristics of a sequential file on a personal computer. They both have an external name for reference and are organized in sequence so that in order to find anything within the data set, you must always begin processing at the beginning of the data set. However, the naming convention for a data set is different – the structure of a data set name is a series of one to eight character name segments separated by periods. The first (or leftmost) name segment is referred to as the High Level Qualifier (HLQ). This is often something such as a name belonging to the user that created the data set (e.g. his "userid" on the server). The last (or rightmost) name segment is referred to as the Low Level Qualifier (LLQ) and is similar to the extension portion of a personal computer file name. It consists of a character string that, by convention, gives an indication of what is stored in the data set. The meaning associated with an LLQ name is different from the meaning of personal computer extensions and the LLQ is not limited to three characters. A low level qualifier of "CNTL", for example, indicates that a file contains control information, such as job control statements. When a data set is created and named, the name can be entered into a catalog so that anyone that needs the file can easily find it simply by referring to its name.

A sequential dataset also differs from a sequential file internally. The data in a sequential data set is not byte oriented; rather, it is record oriented. A record usually represents an entity such as a customer or a warehouse item and consists of fields defined by a software engineer that describes the entity. So, a customer record might include fields for the customer name, address, phone number, credit limit and other items relevant to accurately representing a customer and meeting the processing needs of a business for a customer. In its simplest form, a record is defined by the order and the fixed length of each field. The record length for the sequential data set record then is simply the sum of the length of each field in the record. The process of defining a record is similar to assigning meaning to the columns in a blank spreadsheet. A record is the fundamental unit of data in a data set and the fundamental unit of data that is transferred between a program in server storage and an Input / Output device.

Let's say that the record length for a customer record is 100 characters. Since there is no special character to mark the end of a record, all the records in the data set need to contain the same amount of data (100 characters), even if some of it is not present. Perhaps a new customer does not yet have a credit limit defined. If that is the case, the credit limit field cannot be simply left out of the record. Instead, the fixed length of the credit limit field must be filled in (or "padded") with some data to indicate that there is no credit limit established for this customer. Our customer record example used a 100 character fixed length record but another possibility is for a file is to use variable length records. If used, each variable length record has to have a "header" that contains information about the record, especially its length.

A sequential data set is an appropriate choice for storing data when records will be processed in the order they have been stored or if we know that all (or substantially all) of the records will be processed by a program each time the data set is used (for example sending a monthly bill to utility customers).

Partitioned Data Set

A partitioned data set (or PDS) is a data set that collects a number of sequential data sets (each referred to as a "member" of the PDS) under one data set name. Usually, the collection of sequential data sets is related in some way. For example, each member might be a chapter in a book or each member might contain the source code for a module in a program. The data set name (as described earlier) is used to refer to the entire collection (or PDS). And if a particular member needs to be referenced, it can be identified by using the PDS name in conjunction with its member name. Internally, each member is a sequential data set as described previously. In addition, a PDS uses an internal directory to keep track of where the individual members are located on a disk. A PDS is always stored on a disk. So, when using a PDS, you can go directly to a member, but since each member is a sequential data set, the member itself must be processed sequentially. Consequently, a PDS can be accessed partially in a direct access fashion (by going directly to the member of interest) and partially in a sequential fashion (by processing each member sequentially). A PDS is sometimes referred to as a "library", especially when accessed by an on line end user accessing an enterprise server.

Other Data Set Types

Two other types of data sets types that may be encountered when working with an enterprise server are VSAM data sets and "special" data sets. The internal structure for a VSAM data set is considerably more complex than a sequential or partitioned data set. This complexity provides significantly more function, processing flexibility, and performance than a sequential or partitioned data set can provide. For example, a program can use VSAM data sets to quickly and efficiently access individual records, insert new records in a data set that has a predetermined record order that needs to be maintained, process data set records sequentially some of the time and directly at other times, etc. VSAM data sets are used extensively by the enterprise server operating system itself and by applications that take advantage of its unique data storage and access capabilities (e.g. for frequent access to individual records in an unpredictable order).

Finally, the enterprise server operating system itself has created some "special" data sets with a unique internal structure and characteristics to meet the special needs of the operating system itself. For example, there are paging data sets that are used in the creation and management of virtual storage. And the Job Entry Subsystem uses a special "spool" data set to temporarily hold print data sets created by batch jobs[2]. So, not every data set encountered on an enterprise server will be a sequential, partitioned or VSAM data set – but the vast majority of them will be one of these three types.

Because there are many types of data sets available and many possibilities for the internal structure of a data set (record length, fixed or variable length, etc.), all of these choices need to be specified when creating or using a data set. There are several ways to make these choices (an on line ALLOCATE command, using job control language, etc.) but the fact that the choices need to be made sometimes puzzles a new user of an enterprise server since this step is not needed prior to working with a personal computer file where such choices are essentially made as a file is created and used by an application.

Summary

So, there you have it. Enterprise server data sets are conceptually similar to personal computer files. But data sets provide more variety in terms of their internal structure used to store and retrieve data. This variety in internal structure, when combined with an application program or operating system function that can take advantage of it, results in a very robust set of capabilities for accessing, and efficiently processing, data for a wide variety of situations.

We will explore other aspects of Enterprise Computing in subsequent articles.

[2] See ECI No. 4