



## CCAPRINT

A Newsletter Excerpt for Model 204 Users

April 2008

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.

### ***Large Objects and Table E in V6R1.0***

*By James Damon*

Hundreds, if not thousands, of different types of data objects can now be created and manipulated by an ever increasing number of software packages and hardware devices. Files with extensions representing text documents, photographs, spreadsheets, videos, slide shows, and sounds --the list seems endless--have proliferated in every environment. Figure 1 shows the icons associated with a few of the applications used to process these objects.



**Figure 1. Application icons and file extensions**

Many Model 204 customers wanted to be able to store, manage, locate and retrieve some of these data objects using a Model 204 file. This would provide the ability to link a data object to text data: for example, a picture of a house as a .jpg file linked to a street address. Responding to this need, in V6R1.0 CCA implemented support for these data objects, calling them large objects. Now, any data object, regardless of length or type of data (binary, character or a mixture) can be stored, retrieved and updated in a Model 204 file. Support for large objects is provided through the following:

1. New file parameters and commands
2. New User Language statements
3. A new table, Table E, in a Model 204 file

When stored in a Model 204 file, large objects, with the exception of unformatted xml and text documents, generally contain binary data that is in a format unknown to Model 204. A Model 204 file is primarily a storage and retrieval facility for these large objects. Editing, displaying and manipulating large objects is still handled by well-known, third-party software packages that are readily available.

## Table E Architecture

When a file is created under V6R1.0 and the new parameter, ESIZE, is greater than or equal to 20, Table E will be created and initialized for use as a large object repository.

```
CREATE FILE LOBFILE
  PARAMETER ESIZE=50000
END CREATE
OPEN LOBFILE
INITIALIZE
```

Viewing all of the Table E parameters—ESIZE, EHIGHPG, EPGSUSED--immediately after the file is initialized, but before any data has been loaded, would show the following:

```
VIEW ESIZE,EHIGHPG,EPGSUSED
ESIZE      50000      PAGES IN TABLE E
EHIGHPG    16        TABLE E HIGHEST PAGE USED
EPGSUSED   17        TABLE E PAGES IN USE
```

The first 17 pages of Table E, numbered zero through 16, are reserved for the segment bit map and free page bit maps. Additional bit map pages are allocated throughout Table E when required.

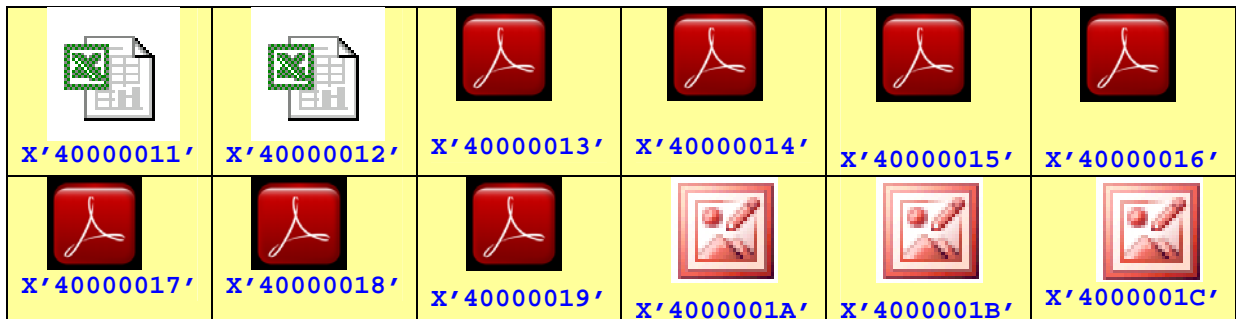
Each page in Table E is formatted as part of initialization. The page trailer is the last 40 bytes of each page and the final four bytes contains the table and page number of that page. The 1-byte table number for Table E is X'40'. The six remaining bits in this byte as well as the remaining three bytes on the page are available for constructing the page number. This means that the maximum number of pages in Table E is X'3FFFFFFF', or 1,073,741,823 pages.

## Large Objects

A large object occupies contiguous pages, as many as required, in Table E. There is no limit, except for ESIZE, to the number of pages a large object may consume. For example, a 30 second .avi video clip, in low resolution, can consume about six megabytes or about 975 Table E pages. Pictures taken with a digital camera can be anywhere from 32K to 3M in length, depending on resolution, or five to 480 Table E pages.

Figure 2 is a schematic of what the first twelve large object pages might contain in Table E.

Figure 2. Table E large objects



These pages could represent twelve different large objects or each collection of icons could represent the pages required to store that single, large object. This collection of Table E pages shows that the first page available is page number X'11' = 17 which is the 18th page in the file. At this point, a VIEW command would show the following:

```
VIEW ESIZE,EHIGHPG,EPGSUSED
ESIZE      50000          PAGES IN TABLE E
EHIGHPG    28            TABLE E HIGHEST PAGE USED
EPGSUSED   29            TABLE E PAGES IN USE
```

## Enlarging Table E

If additional pages are required, you can issue an INCREASE TABLEE command. Like INCREASE TABLEB and INCREASE TABLED, the INCREASE TABLEE command will allocate pages in FREESIZE to Table E. If there are no pages in FREESIZE, the INCREASE DATASETS command can be used to add additional datasets to the file. This will increase the number of pages in FREESIZE making those pages available for the INCREASE command.

## In Summary

If you have an application in mind that could combine text data with graphics, spreadsheets, pictures, maps, and so on, then large objects provide the solution.

## Coming attractions

In part 2 of this article, I'll review the facilities available for storing, locating, retrieving, updating, printing and deleting large objects as well as the tools available for reorganizing files containing large objects.

© 2008 Computer Corporation of America  
200 West Street, 3<sup>rd</sup> Floor West, Waltham, MA 02451