



## CCAPRINT

A Newsletter Excerpt for Model 204 Users

April 2007

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.

## Preserving Field Definitions with INITIALIZE KEEPDEFS

By James Damon

When you test a new application, you often need test files that can be recreated easily. This can speed up your testing process by letting you focus on the application instead of the files, which for the most part, have already been designed for the data they will contain. There are many instances when it would be convenient to take an existing file and, while preserving field definitions, re-initialize it.

The CREATE command is first required to create a file and establish DDNAMES and datasets, table sizes and file parameters. CREATE is then followed by INITIALIZE. INITIALIZE is required before the file can be used. Once a file has been initialized, however, it can be initialized again and again without a CREATE.

Prior to V6R1.0, the INITIALIZE command issued against a file populated with data or procedures, completely eliminated all data, index entries in Table C and Table D, stored procedures, and all data in Table A including field definitions. Only the file parameter settings were preserved. This meant that in order to clear the file and start over, you would issue an INITIALIZE command and then redefine all fields and their attributes. In V6R1.0 the INITIALIZE command has a new option, KEEPDEFS, which preserves the field definitions--field names and attributes--including preallocated fields.

### The New INITIALIZE Command

Figure 1 shows the complete syntax of the V6R1.0 INITIALIZE command.

Figure 1. V6R1.0 INITIALIZE command syntax

```
INITIALIZE [KEEPDEFS]
           { [SORT | HASH] field-description
           | RECSCY          field-description }
```

An existing file that is not hashed or sorted can be initialized and have field definitions preserved as shown in Figure 2. In this case, you would not issue the CREATE command prior to the INITIALIZE command.

Figure 2. Example with KEEPDEFS option

```
OPEN DSNLIST
INITIALIZE KEEPDEFS
*** M204.0763: BEGIN INITIALIZATION: FILE DSNLIST
*** M204.0764: END INITIALIZATION: FILE DSNLIST - FIELD DEFINITIONS
                PRESERVED
```

## INITIALIZE Command Processing

INITIALIZE command processing is generally very fast. Unless you have a large Table A or a large Table C, the initialization process usually completes in less than one or two seconds. The entire process involves:

1. Formatting and rewriting all Table A pages with hexadecimal zeroes. However, when the KEEPDEFS option is specified, only the FEW VALUED and MANY VALUED pages are formatted and rewritten. The attribute pages, which contain field definitions, are untouched when KEEPDEFS is specified.
2. Formatting and rewriting all Table C pages with hexadecimal zeroes. If you have eliminated all KEY and NR fields and set CSIZE=1, this process should complete in less than 1 second depending on I/O response times and CPU availability.
3. Resetting parameters in the File Parameter List (FPL) page to indicate that the file is empty; that is it contains no data and no procedures. Some parameters are reset to zero (0), others to minus one (-1).
4. Initializing free page bit maps--Table D and Table E--to hexadecimal zeroes.

## In Summary

If you are testing an application that stores records and you need a clean file each time you run your test, consider using the very fast and convenient new option on the INITIALIZE command, KEEPDEFS. This will preserve your field definitions and all parameter settings and let you repeatedly test your application against a clean file with little delay between tests.