

Why start using the OCCURS DYNAMIC? How to save memory by replacing your fixed arrays

Every COBOL application has many programs containing definitions with OCCURS in Working Storage or Linkage sections.

These arrays are usually of a fixed size to accommodate the maximum number of references possible in the table.

The result is that the memory consumed for the array is fixed, even when the program doesn't need to use all the occurrences.

For example, this definition:

```
78 max-element value 900. 01 w-group. 03 w-element occurs max-  
element. 05 w-cod pic 9(3). 05 w-desc pic x(20). 05 w-note pic x(100).
```

will allocate the memory for all 900 occurrences at the program startup, even if the program will use only few references.

To reduce the memory consumed by allocating only the memory really necessary, you can change your code to use a dynamic occurs. For example:

```
01 w-group. 03 w-element dynamic capacity max-  
element. 05 w-cod pic 9(3). 05 w-desc pic x(20). 05 w-note pic x(100).
```

The existing code works without any change, so statements like MOVE, IF, SORT... are still supported.

When using occurs dynamic, you cannot move or pass a parent level (w-element above) of a dynamic occurs to a similar structure with a fixed occurs.

For instance, when moving a working storage field to an FD field (where the fixed occurs remains because files were already created with that size) or vice versa, using the parent level (w-group) returns this compiler warning: "Dynamic items will be ignored". It's necessary to write a loop to move all the child fields instead of the parent.

Passing the occurs dynamic group level as a parameter in a CALL statement is supported as long as the called program receives the group in a similar OCCURS DYNAMIC array in the Linkage definition

Online URL: <https://support.veryant.com/phpkb/article.php?id=323>