



4th Dimension 2003.2

ADDENDUM

Welcome to 4th Dimension version 2003.2. This document describes the modifications related to this new version of the 4th Dimension product line.

Compatibility

4th Dimension

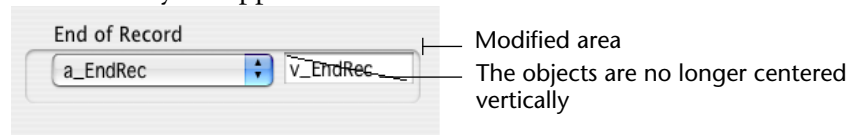
Databases created or opened using 4th Dimension version 2003 or 2003.1 are entirely compatible with 4th Dimension version 2003.2 (structure and data). Moreover, a database opened with 4th Dimension 2003.2 can be reopened with 4th Dimension 2003 or 2003.1. Keep in mind, however, that if the new syntax of the MULTI SORT ARRAY command is used in a 2003.2 database (see next page), an error will be generated when you use this database with a previous version of 4th Dimension.

Plug-ins

Since modifications have been made to the applications of the 4th Dimension product line version 2003.2, it is advisable to upgrade your plug-ins to version 2003.2.

Panther (MacOS 10.3) interface and group boxes

The appearance of objects of the “Group box” type has been modified in the Panther version of MacOS X (10.3): the label is now located above the area. This modification may require some retouching in the interface of your applications.



Handling of relations in quick reports

In order to maintain the compatibility of 4th Dimension 2003.x with former databases, the **Only automatic relations/All relations in automatic** option is no longer displayed in the Quick Report window when it is called via the QR REPORT command.

In fact, in this case, it is preferable that the status of the relations be managed by programming using the AUTOMATIC RELATIONS command as in former versions of 4th Dimension.

Support of substructures

The support of substructures (subtables and subfields) has been modified in 4th Dimension 2003.2, both in the Design and User environments:

- From now on, the Method editor no longer recognizes substructures on more than one level. This type of substructure may have been defined in former versions of 4th Dimension.
- The Quick Report editor no longer lets you carry out calculations (sum, average, etc.) on substructures nor to apply formats to them. However, it does remain possible to use quick reports containing calculations on substructures, if they have been created with a previous version of 4th Dimension, and if they have not been re-saved using 4th Dimension 2003.x.

Language

MULTI SORT ARRAY ("Arrays" theme)

MULTI SORT ARRAY (*arr1{;sort1}{; arr2{; sort2}{;... arrX{; sortX}}}*)

Parameter	Type	Description
arr1...arrX	Array	→ Array(s) to be sorted
sort1... sortX	> or <	→ > to sort by increasing order or < to sort by decreasing order If omitted= no sort

In 4th Dimension 2003.2, the MULTI SORT ARRAY command accepts a new syntax. This new syntax lets the names of the synchronized arrays to which a multi-criteria sort will be applied to be passed directly as parameters.

Note The former syntax is still accepted by the program (on this point, see the 4th Dimension *Language Reference* manual).

You can pass an unlimited number of sets (array;> or <) and/or of array names. All the arrays passed as parameters are sorted in a synchronous manner.

You can pass arrays of any type, except for pointer and picture arrays. You can sort an element of a two-dimensional array (i.e., `t2DArray{$\vlThisElement}`), but you cannot sort the two-dimensional array itself (i.e., `t2DArray`).

To use the contents of an array as the sort criterion, pass the *sort* parameter. The value of the parameter (> or <) sets the order (increasing or decreasing) in which the array will be sorted. If the *sort* parameter is omitted, the contents of the array are not used as the sort criterion.

Note Remember that at least one sort criterion must be passed in order for the command to operate. If no sort criterion is set, an error will be generated.

The sort levels are determined by the order in which the arrays are passed to the command: the position of an array with a sort criterion in the syntax determines its sort level.

- ▼ The following example creates four arrays and sorts them by city (increasing order) then by salary (decreasing order) with the last two arrays, *names* and *telNums*, being synchronized according to previous sort criteria:

```
ALL RECORDS([Employees])
SELECTION TO ARRAY([Employees]City;cities;[Employees]Salary;
                    salaries;[Employees]Name;names;
                    [Employees]TelNum;telNums)
MULTI SORT ARRAY(cities;>;salaries;<;names;telNums)
```

If you want the names array to be used as the third sort criterion, you just need to add > or < after the *names* parameter.

Note that the syntax:

```
MULTI SORT ARRAY (cities;>;salaries;names;telNums)
```

is strictly equivalent to:

```
SORT ARRAY(cities;salaries;names;telNums;>)
```