

Operator TO COPY

1 Goal

To duplicate a concept utilisator under another name.

This order is seldom useful and it presents the disadvantage of doubling space "disc" of the concept in the total base of the user. The produced concept can be destroyed (order TO DESTROY) to recover the disk space.

Only certain types of concepts are accepted.

2 Syntax

```
CO2 = TO COPY (  ◆  CONCEPT = co1,          [concept]
                 ◇  INFORMATION = / 1,
                 / 2,
                 )
```

3 Operands

3.1 Keyword **CONCEPT**

◆ CONCEPT = co1

Name of the concept which one wants to duplicate.
The result of the order is of the same type as the duplicated concept.

3.2 Types of **CONCEPT** authorized

One cannot duplicate all the types of concepts.
Only are currently treated:

```
cabl_precont,
listr8_sdaster,
listis_sdaster,
fonction_sdaster,
nappe_sdaster,
table_sdaster,
maillage_sdaster,
modele_sdaster,
evol_elas,
evol_noli,
evol_ther,
```

If the type of the concept that you want to duplicate does not appear in this list, it is necessary to emit a request for evolution.

3.3 Remarks

The need to duplicate a concept is rare because it is a priori useless to have the same object under two different names.

The produced concept (CO2) corresponds to a “deep” copy of the initial concept (CO1), i.e. that one duplicated ALL objects JEVEUX of CO1.

The “volume” of CO2 in the total base is identical to that of CO1.

The use of the order conduit thus always with a loss in the disk space.

3.4 Operand **INFORMATION**

◇ INFORMATION = information

This keyword does not have any influence.

4 Example

```
Lr1 = DEFI_LIST_REEL ( ... )  
Lr2 = TO COPY (CONCEPT = Lr1)
```