
Titre : Impressions dirigées par le mot clé *INFO* des commandes

Auteurs : **J. PELLET**

Clé : D6.04.02

Date : 02/05/1997

Page : 1/4

Département Mécanique et Modèles Numériques

Indice : A

Diffusion : Développeurs

Manuel de Descriptif Informatique

Fascicule D6.04 :

Document D6.04.02

Impressions dirigées par le mot clé *INFO* des commandes (paquet *INFXXX*)

Résumé :

Ce document indique comment les développeurs doivent réaliser les impressions dirigées par le mot clé *INFO* des commandes.

1 Introduction

Les informations dirigées par le **mot clé INFO** des commandes sont des messages d'information (INFO) concernant le déroulement du calcul. Elles sont **contractuelles** (pour le niveau 1) et documentées dans la documentation d'utilisation.

Remarque historique :

Jusqu'en 1995, de nombreuses commandes comportaient un mot clé "IMPR" dont l'argument pouvait valoir : 0, 1, 2, 3, L'effet de ce mot clé était varié : écriture d'informations, de résultats, de données ... sur le fichier 'MESSAGE' ou 'RESULTAT'. Pour clarifier la situation, nous avons décidé de résorber progressivement ces mot clé IMPR par le mot clé INFO (pour les INFO) et d'éventuels mot clés d'impressions pour l'impression des résultats.

La description plus détaillée de ce qu'est (ou ce que n'est pas) une INFO est faite dans [D2.07.01 §6], en résumé :

- une INFO n'est ni un résultat, ni une alarme ... ,
- le mot clé **INFO** peut prendre 2 valeurs : 1 ou 2 ; le défaut est 1,
- les impressions sont faites par "WRITE" et non des UTMESS <I> [D6.04.01 §1.5],
- les INFO sont écrites sur le fichier 'MESSAGE'

Les impressions étant faites par des WRITE formatés, les seuls problèmes à régler sont :

- sur quelle unité logique ?
- comment respecter le choix de l'utilisateur : INFO = 1 ou 2 ?

Ceci fait l'objet du paragraphe 2.

Un autre problème survient lorsqu'une commande **ne veut pas** d'un message imprimé systématiquement par une routine utilitaire. Ceci fait l'objet du paragraphe 3.

2 Comment émettre un message d'information ?

- La routine **INFMAJ** sert à informer le "paquet" **INFXXX** du niveau d'impressions demandé par l'utilisateur,
- la routine **INFNIV** sert à récupérer le niveau d'impression demandé par l'utilisateur ainsi que l'unité logique du fichier 'MESSAGE'.
- le programmeur d'une commande fera donc :
 - a- mettre le mot clé **INFO** dans le **catalogue** de la commande, (2 valeurs possibles 1 et 2, valeur par défaut : 1),
 - b- dans la routine **OP000I**

```
CALL GETVIS (' ', 'INFO', 0, 1, 1, NIV, IBID)
CALL INFMAJ (NIV)
```
 - c- dans une routine voulant imprimer des INFO :
 - `CALL INFNIV (IFM, NIV),`
 - pour une INFO de niveau 1 (contractuelle) :

```
IF (NIV.GE.1) WRITE (IFM, ...)
```
 - pour une INFO de niveau 2 :

```
IF (NIV.EQ.2) WRITE (IFM, ...).
```

Remarques :

On pourrait penser que les INFO de niveau 1 peuvent être écrites sans se protéger par (IF (NIV.GE.1)...) car ces impressions se déclenchent par défaut. On verra au [§3] pourquoi il faut les protéger.

*Pour se protéger des commandes qui n'ont pas le mot clé **INFO** dans leur catalogue, le superviseur met à jour le **COMMON avant** chaque commande :*

| *CALL INFMAJ* (1).

3 Se protéger d'un excès de messages

Le responsable d'une commande est responsable des impressions de niveau 1 de cette commande. Il doit les documenter et s'arranger pour que le fichier message reste lisible.

Un problème se pose quand une commande fait appel à un utilitaire imprimant des INFO de niveau 1 que le responsable de la commande juge sans intérêt (ou trop nombreuses ... par exemple dans une boucle). Il faut donner à la commande le moyen de "faire taire" l'utilitaire. Pour cela, on dispose des 2 routines :

INFMUE () pour rendre le code "MUET" jusqu'à nouvel ordre,
INFBAV () pour remettre le code en mode "BAVARD".

Ces 2 routines ne sont efficaces que si le niveau d'impression demandé est 1 : on ne peut empêcher le code de "parler" si INFO = 2.

Exemple :

La commande OPOOOI ne veut pas entendre ce que dit la routine B

```
SUBROUTINE OPOOOI
CALL INFMAJ
...
CALL A (...)
...
CALL INFMUE ( )
CALL B (...)
CALL INFBAV ( )
...
END
```

Remarques :

- *Ce système n'est pas parfait car il permet seulement d'interrompre sans discernement le flot des impressions. Il ne permet pas par exemple, d'interdire à B de parler tout en laissant C (appelé par B) parler !*
- *pour que INFMUE/INFBAV soit efficace, il faut que les routines que l'on veut faire taire aient programmé leurs impressions de niveau 1 de la manière suivante : IF (NIV.GE.1) WRITE*

4 Fonctionnement des utilitaires INFXXX

Ce paragraphe a pour seul but d'aider à comprendre (si nécessaire) l'utilisation de ces routines :

- COMMON/INF001/NIVUTI, NIVPGM, UNITE
INTEGER NIVUTI : niveau demandé par l'utilisateur : 1 ou 2
INTEGER NIVPGM : niveau accessible au programmeur : 0, 1, ou 2
INTEGER UNITE : unité logique du fichier 'MESSAGE'
- le COMMON INF001 **n'est utilisé que par** les routines INFXXX
- SUBROUTINE INFMAJ ()
COMMON/INF001/...
GETVIS (nomCMD, 'I', 'INFO', ..., NIV)
NIVUTI = NIV
NIVPGM = NIV
UNITE = IUNIFI ('MESSAGE')
END
- SUBROUTINE INFNIV (IFM, NIV)
COMMON/INF001/...
INF = UNITE
NIV = NIVPGM
END
- SUBROUTINE INFMUE()
COMMON/INF001/...
IF (NIVUTI.EQ.1) NIVPGM = 0
END
- SUBROUTINE INFBAV()
COMMON/INF001/...
NIVPGM = NIVUTI
END