



|  |
|--|
| <p><b>Manuel d'utilisation<br/>NSFCTUTIL</b></p> |
|--|



## Table des matières

|   |    |
|---|----|
| Résumé.....                                   | 5  |
| Introduction .....                            | 7  |
| Nom de la DLL utilisateur .....               | 9  |
| Installation et utilisation .....             | 11 |
| Référence de la librairie NSFCTUTIL .....     | 13 |
| Constantes CST_NS_USERFUNCT_*% .....          | 13 |
| Constantes de gestion des erreurs.....        | 14 |
| Fonction NS_USERFUNCT_BUILD% .....            | 15 |
| Fonction NS_USERFUNCT_GLOBAL% .....           | 17 |
| Fonction NS_USERFUNCT_NCLCHECK% .....         | 19 |
| Fonction NS_USERFUNCT_NCLEDITOR% .....        | 20 |
| Fonction NS_USERFUNCT_QUERYACTIONNAME\$ ..... | 22 |
| Segment SEG_FCTPARAMIN .....                  | 23 |
| Segment SEG_FCTPARAMOUT .....                 | 24 |
| Index .....                                   | 25 |



## RESUME

---

Ce chapitre décrit la librairie de fonctions utilisateurs NSFCTUTIL fournie avec l'outil NS-DK.

Cette librairie est spécifique à NS-DK.



## INTRODUCTION

---

Afin de personnaliser les contrôles sur les projets clients, il est désormais possible de définir des fonctions qui seront appelées par NS-Design pendant les périodes de développement.

Ces fonctions de contrôles seront appelées :

- Après une vérification sur du code NCL (touche [F8]) appartenant à une librairie ou à un évènement.
- Avant et/ou après le lancement de la construction de l'application (Build).
- A partir de l'éditeur de code NCL (deux fonctions).
- Quelque soit le contexte (édition NCL, édition scr, compilation ...) (deux fonctions utilisateurs globales).

Les prototypes de ces fonctions ainsi que leurs noms sont imposés par Nat System. Le nom de la DLL dans laquelle devra se trouver ces fonctions sera paramétrable au niveau de l'outil NS-Design dans la boîte de dialogue Setup. Le paramétrage doit se faire sur chaque poste (stocké dans le fichier NSDKLOC.INI).

Il n'est pas nécessaire aux utilisateurs de coder toutes les fonctions potentiellement appelables. L'outil gèrera les menus ainsi que les appels des différentes fonctions suivant leurs existences ou non dans la DLL utilisateur.

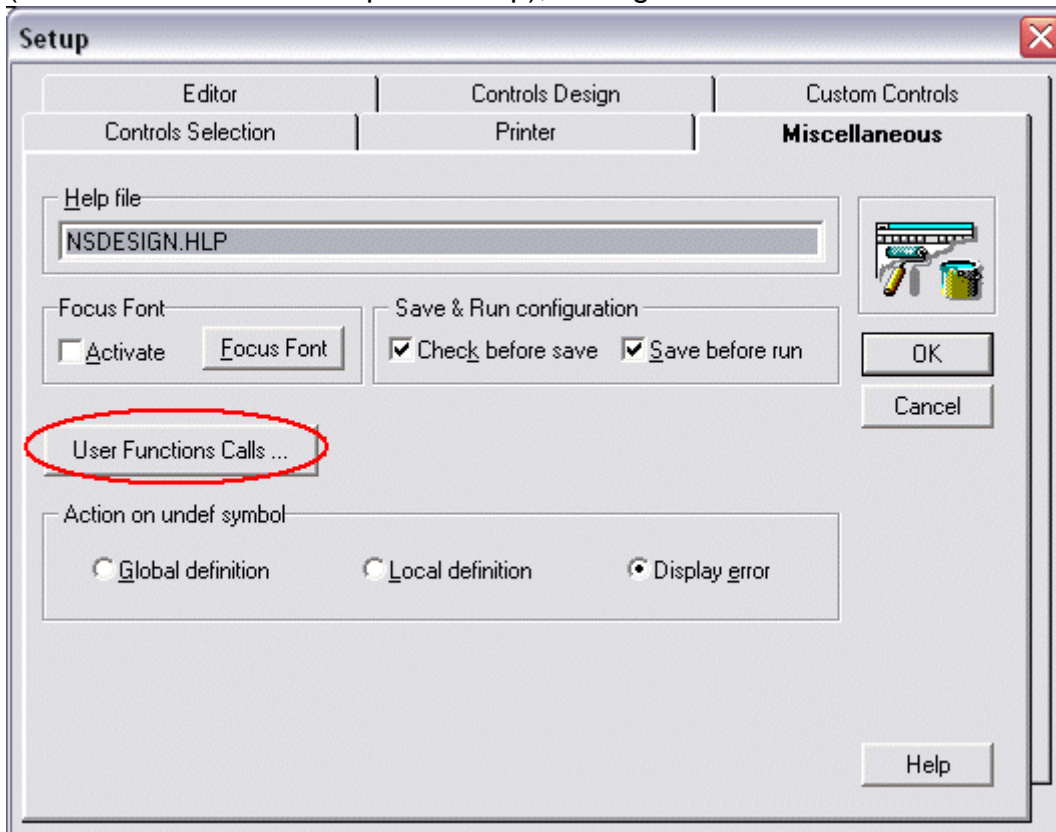
La définition des constantes et des segments utilisés pour les appels des fonctions utilisateurs sont présentes dans le fichier NSFCTUTIL.NCL.



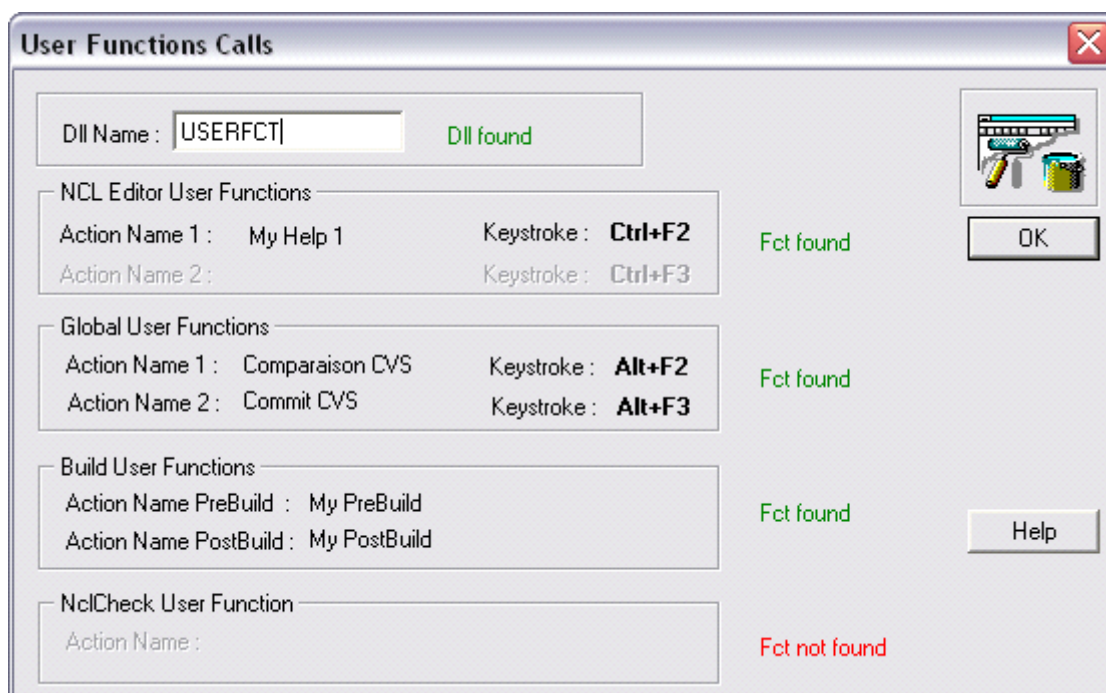


## NOM DE LA DLL UTILISATEUR

La fenêtre de détail de ces fonctions utilisateurs est appelée depuis la fenêtre Setup (obtenue avec le menu Option/Setup), à l'onglet Miscellaneous.



En cliquant sur le bouton User Functions Calls ..., la fenêtre suivante apparaît :



Le champ Dll Name indique le nom de la DLL utilisateur dans laquelle des fonctions utilisateur devront être codées.

Dans un souci de cohérence avec l'outil NS-DK, les différents raccourcis claviers correspondants aux quatre fonctions utilisateurs potentiellement présentes dans la DLL utilisateur sont imposés et ne sont pas modifiables.

Dans le groupe Ncl Editor User Functions, les champs Action Name 1 et Action Name 2 correspondent au libellé qui sera affiché dans le menu Edit lorsque le focus est sur l'éditeur de NCL.

Dans le groupe Global User Functions, les champs Action Name 1 et Action Name 2 correspondent au libellé des bulles d'aides des boutons des fonctions utilisateurs.

Dans notre exemple, l'utilisateur choisit le nom de sa DLL (USERFCT.DLL) et ensuite sont affichés les fonctions qui sont présentes. On note donc que la deuxième fonction dans l'éditeur NCL n'a pas été codée (ActionName2 est grisé) ainsi que la fonction utilisateur appelée après la vérification de code.

En fonction du contexte dans lequel ces fonctions seront appelées, deux segments d'échanges de données devront être passés en paramètre :

- SEG\_FCTPARAMINT
- SEG\_FCTPARAMOUT

## INSTALLATION ET UTILISATION

---

Après l'installation de NS-DK, vous trouverez NSFCTUTIL.NCL dans le répertoire <NSDK>\NCL, <NSDK> représentant le répertoire spécifié lors de l'installation.

Pour l'utilisation, vérifiez que NSFCTUTIL.NCL est bien dans le répertoire correspondant à \SERVICES\NCL.

Lors de la génération sous forme de DLL de vos composants NS-DK (Librairies et classes), vérifiez que NSFCTUTIL.LIB est bien spécifiée dans les champs .DLL Libraries et .EXE Libraries de la boîte Configurations à l'onglet Compiler/Linker de NS-DK.

Pour consulter le fichier NSFCTUTIL.NCL :

1. Placez-vous dans le répertoire <NS-DK>\NCL

<NS-DK> représente le répertoire que vous avez choisi au moment de l'installation de NS-DK.

2. Editez le fichier NSFCTUTIL.NCL avec n'importe quel éditeur de texte.



## REFERENCE DE LA LIBRAIRIE NSFCTUTIL

### Constantes CST\_NS\_USERFUNCT\_\*\*%

Les constantes CST\_NS\_USERFUNCT\_\*\*% correspondent à l'identifiant des différentes fonctions utilisateurs potentiellement créées par l'utilisateur. Elles sont à utiliser notamment en paramètre de la fonction NS\_USERFUNCT\_QUERYACTIONNAME\$.

| Constantes                   | Description interne | Description   |
|------------------------------|---------------------|---|
| CST_NS_USERFUNCT_NCLCHECK%   | 1                   | Identifie la fonction utilisateur lancée après la vérification de code NCL (touche [F8])      |
| CST_NS_USERFUNCT_PREBUILD%   | 2                   | Identifie la fonction utilisateur lancée avant le Build.                                      |
| CST_NS_USERFUNCT_POSTBUILD%  | 3                   | Identifie la fonction utilisateur lancée après le Build.                                      |
| CST_NS_USERFUNCT_NCLEDITOR1% | 4                   | Identifie la fonction utilisateur lancée lors de l'édition d'un NCL par le raccourci CTRL+F2. |
| CST_NS_USERFUNCT_NCLEDITOR2% | 5                   | Identifie la fonction utilisateur lancée lors de l'édition d'un NCL par le raccourci CTRL+F3. |
| CST_NS_USERFUNCT_GLOBAL1%    | 6                   | Identifie la fonction utilisateur lancée par le raccourci ALT+F2.                             |
| CST_NS_USERFUNCT_GLOBAL2%    | 7                   | Identifie la fonction utilisateur lancée par le raccourci ALT+F2.                             |

Voir aussi NS\_USERFUNCT\_QUERYACTIONNAME\$

## Constantes de gestion des erreurs

Les constantes de gestion d'erreur permettent de préciser le niveau de sévérité de l'erreur dans le segment SEG\_FCTPARAMOUT.

| Constantes      | Déclaration interne | Description            |
|-----------------|---------------------|------------------------|
| NSINFOERROR%    | 0                   | Information            |
| NSWARNINGERROR% | 1                   | Erreur d'avertissement |
| NSFATALERROR%   | 2                   | Erreur fatale          |

Lors de l'affichage des erreurs dans la fenêtre Log, si la constante NSINFOERROR% est positionnée, le libellé de l'erreur sera alors précédé par "---". Si la constante est NSWARNINGERROR%, le libellé de l'erreur sera alors précédé par une icône d'avertissement jaune. Enfin, si NSFATALERROR% est positionnée, le libellé de l'erreur sera alors précédé par une icône d'avertissement rouge.

**Voir aussi** [SEG\\_FCTPARAMOUT](#)

## Fonction NS\_USERFUNCT\_BUILD%

Cette fonction permet d'ajouter des fonctions de contrôles du projet avant et/ou après sa construction (Build).

| Syntaxe          | NS_USERFUNCT_BUILD% ( <i>BuildPos%</i> , <i>NsParamIn</i> , <i>ppointer</i> ) |   |   |  |
|------------------|---|---|---|--|
| Paramètres       | BuildPos%   | INT   | I | entier pouvant prendre la valeur CST_NS_USERFUNCT_PREBUILD% ou CST_NS_USERFUNCT_POSTBUILD% |
|                  | NsParamIn   | SEGMENT                                       | O | segment SEG_FCTPARAMIN   |
|                  | ppointer  | SEGMENT                                       | O | segment typé sur une séquence contenant des segments de type SEG_FCTPARAMOUT               |
| Valeur retournée | INT   | TRUE%, la compilation s'effectue normalement. |   |  |

1. Lors de cet appel seul les champs contenant le nom du projet ainsi que la configuration de compilation en cours seront renseignés.
2. Si la fonction NS\_USERFUNCT\_BUILD% (avec le paramètre BuildPos% = CST\_NS\_USERFUNCT\_PREBUILD%) retourne FALSE%, la compilation ne sera pas effectuée, la fenêtre de Log passera en premier plan et les erreurs renseignées dans la séquence de sortie seront affichées.
3. Si la fonction NS\_USERFUNCT\_BUILD% (avec le paramètre BuildPos% = CST\_NS\_USERFUNCT\_POSTBUILD%) retourne FALSE%, la fenêtre de Log passera en premier plan et les erreurs renseignées dans la séquence de sortie seront affichées.
4. Le troisième paramètre est un pointeur typé sur une séquence (liste chaînée) qui est allouée et désallouée par NS-Design. Lorsque la fonction utilisateur renseigne les éléments de type SEG\_FCTPARAMOUT, le paramètre Severity% doit être renseigné.
5. L'allocation et la désallocation des différentes structures passées en paramètre sont effectuées par NS-Design.

Exemple :

```
Function NS_USERFUNCT_BUILD% (Int BuildPos%,SEG_FCTPARAMIN @NSPARAMIN,
SEQ_PPOINTER @ppointer) return int
Local SEG_FCTPARAMOUT @NSPARAMOUT
Evaluate BuildPos%
Where CST_NS_USERFUNCT_PREBUILD%

Message NSPARAMIN.Proj_Name$ ,NSPARAMIN.Config_Name$

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur before 1"
NSPARAMOUT.Severity% = 0
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1
```

```

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur before 2"
NSPARAMOUT.Severity% = NSWARNINGERROR%
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur before 3"
NSPARAMOUT.Severity% = NSFATALERROR%
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1
Return FALSE%
EndWhere
Where CST_NS_USERFUNCT_POSTBUILD%

Message NSPARAMIN.Proj_Name$ ,NSPARAMIN.Config_Name$

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur after 1"
NSPARAMOUT.Severity% = NSWARNINGERROR%
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur after 2"
NSPARAMOUT.Severity% = NSFATALERROR%
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1

@NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
NSPARAMOUT.Err_Message$ = "Erreur after 3"
NSPARAMOUT.Severity% = 0
NSPARAMOUT.Col_Number% = -1
NSPARAMOUT.Line_Number% = -1
Return False%
EndWhere
ELSE
Return False%
EndEvaluate
Return True%
EndFunction

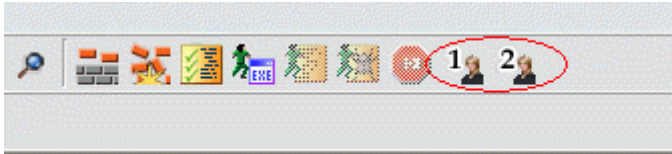
```

**Voir aussi Constantes de gestion d'erreur, CST\_NS\_USERFUNCT\_POSTBUILD%, CST\_NS\_USERFUNCT\_PREBUILD%, SEG\_FCTPARAMIN, SEG\_FCTPARAMOUT**



## Fonction NS\_USERFUNCT\_GLOBAL%

Il est possible de lancer quelque soit les fenêtres ouvertes sous NS-Design deux fonctions utilisateurs. Ces fonctions si elles sont présentes dans la DLL utilisateur apparaîtront au niveau du menu Build ainsi que par la présence ou non de deux boutons supplémentaires dans la barre d'outil de NS-Design.



Elles pourront être appelées par les raccourcis clavier [Alt]+[F2] et [Alt]+[F3]. Dans un souci de cohérence avec l'outil NS-Design ces raccourcis clavier sont imposés par Nat System et ne sont pas modifiables.

| Syntaxe          | NS_USERFUNCT_GLOBAL% ( <i>FctNum%</i> , <i>NsParamIn</i> , <i>ppointer</i> ) |                 |   |  |
|------------------|--|-----------------|---|--|
| Paramètres       | FctNum%  | INT             | I | entier identifiant les deux fonctions utilisateurs pouvant être appelées           |
|                  | NsParamIn  | SEGMENT         | O | segment typé sur les paramètres d'entrée renseignés par NS-Design (SEG_FCTPARAMIN) |
|                  | ppointer   | SEGMENT         | O | segment typé sur une séquence contenant des segments de type SEG_FCTPARAMOUT       |
| Valeur retournée | INT  | TRUE% ou FALSE% |   |  |

1. Le paramètre FctNum% permet d'identifier quelle est la fonction utilisateur 1 ou 2 appelée (respectivement identifiée par les constantes CST\_NS\_USERFUNCT\_GLOBAL1% et CST\_NS\_USERFUNCT\_GLOBAL2%)
2. Lors de cet appel seul le champ contenant le nom du projet sera renseigné.
3. Si la fonction appelée retourne FALSE%, la fenêtre de Log passera au premier plan et les erreurs renseignées dans la séquence de sortie seront affichées.
4. Le paramètre ppointer est un pointeur typé sur une séquence (liste chaînée) qui est allouée et désallouée par NS-Design. Lorsque la fonction utilisateur renseigne les éléments de type SEG\_FCTPARAMOUT, le paramètre Severity% doit être renseigné. Les valeurs possibles sont NSINFOERROR% (dans ce cas le libellé de l'erreur sera précédé par '---'), NSWARNINGERROR% (dans ce cas le libellé de l'erreur sera précédé par une icône d'avertissement jaune), ou NSFATALERROR% (dans ce cas le libellé de l'erreur sera précédé par une icône d'avertissement rouge).
5. L'allocation et la désallocation des différentes structures passées en paramètre sont effectuées par NS-Design.

Exemple :

```

Function NS_USERFUNCT_GLOBAL% (Int FctNum%,SEG_FCTPARAMIN @NSPARAMIN,
SEQ_PPOINTER @PPointer) return int
Local pointer hvisu%
Local Int ret%(2)
Local SEG_FCTPARAMOUT @NSPARAMOUT

Evaluate FctNum%
Where CST_NS_USERFUNCT_GLOBAL1%
    Message "Ctrl Glob1", NSPARAMIN.Proj_Name$

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "Erreur Glob 1 1"
    NSPARAMOUT.Severity% = NSWARNINGERROR%
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = -1

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "ErreurGlob 1 2"
    NSPARAMOUT.Severity% = NSFATALERROR%
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = -1

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "Erreur Glob 1 3"
    NSPARAMOUT.Severity% = NSWARNINGERROR%
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = -1
EndWhere
Where CST_NS_USERFUNCT_GLOBAL2%

    Message "Ctrl Glob2", NSPARAMIN.Proj_Name$

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "Erreur Glob 2 1"
    NSPARAMOUT.Severity% = 0
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = 0

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "ErreurGlob 2 2"
    NSPARAMOUT.Severity% = NSFATALERROR%
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = -1

    @NSPARAMOUT = SEG_FCTPARAMOUT(seq_INSERT_ENTRY%(@PPointer,0))
    NSPARAMOUT.Err_Message$ = "Erreur Glob 2 3"
    NSPARAMOUT.Severity% = NSWARNINGERROR%
    NSPARAMOUT.Col_Number% = -1
    NSPARAMOUT.Line_Number% = -1
EndWhere
EndEvaluate
Return ret%
EndFunction

```

**Voir aussi [CST NS USERFUNCT GLOBAL1%](#), [CST NS USERFUNCT GLOBAL2%](#), [SEG FCTPARAMIN](#), [SEG FCTPARAMOUT](#)**

## Fonction NS\_USERFUNCT\_NCLCHECK%

Cette fonction est appelée (si elle existe) lors de la vérification du code NCL (touche [F8]).

|                         |  |   |   |                         |
|-------------------------|--|---|---|-------------------------|
| <b>Syntaxe</b>          | <b>NS_USERFUNCT_NCLCHECK%</b> ( <i>NsParamIn</i> , <i>NsParamOut</i> ) |   |   |                         |
| <b>Paramètres</b>       | NsParamIn  | SEGMENT   | O | segment SEG_FCTPARAMIN  |
|                         | NsParamOut   | SEGMENT   | O | segment SEG_FCTPARAMOUT |
| <b>Valeur retournée</b> | INT  | TRUE% ou FALSE%, si une erreur a été trouvé dans le code. |   |                         |

1. Lors de l'appel de cette fonction tous les champs du segment seront renseignés (sauf la configuration de compilation).
2. Si une erreur dans le code a été trouvé, les champs Err\_Message\$, Col\_Number% et Ligne\_Number% du segment SEG\_FCTPARAMOUT devront être renseignés. Le message d'erreur sera ainsi affiché dans la barre d'état de l'éditeur NCL et le curseur sera positionné aux coordonnées fournies par les deux champs Col\_Number% et Ligne\_Number%.
3. La fonction utilisateur NS\_USERFUNCT\_NCLCHECK% ne sera appelée après l'appui sur la touche [F8] que si ce dernier n'a pas décelé une erreur syntaxique ou lors de la sauvegarde (si l'option Check before Save est cochée dans la boîte de dialogue Setup à l'onglet Miscellaneous).
4. L'allocation et la désallocation des segments passés en paramètres sont effectuées par NS-Design.

Voir aussi SEG\_FCTPARAMIN, SEG\_FCTPARAMOUT

## Fonction NS\_USERFUNCT\_NCLEDITOR%

Il est possible de lancer depuis l'éditeur de NCL deux fonctions utilisateurs. Ces fonctions si elles sont présentes dans la DLL utilisateur apparaîtront au niveau du menu Edit. Elles pourront être appelées par les raccourcis clavier [Ctrl]+[F2] et [Ctrl]+[F3]. Dans un souci de cohérence avec l'outil NS-Design ces raccourcis clavier sont imposés par Nat System et ne sont pas modifiables.

| Syntaxe          | NS_USERFUNCT_NCLEDITOR% ( <i>FctNum%</i> , <i>NsParamIn</i> , <i>ParamOut</i> ) |                 |   |  |
|------------------|---|-----------------|---|--|
| Paramètres       | FctNum%   | INT             | I | entier identifiant les deux fonctions utilisateurs pouvant être appelées             |
|                  | NsParamIn   | NsParamIn       | O | segment typé sur les paramètres d'entrée renseignés par NS-Design (SEG_FCTPARAMIN)   |
|                  | ParamOut  | SEGMENT         | O | segment typé sur les paramètres de sortie renseignés par NS-Design (SEG_FCTPARAMOUT) |
| Valeur retournée | INT   | TRUE% ou FALSE% |   |  |

1. Le paramètre FctNum% permet d'identifier quelle est la fonction utilisateur 1 ou 2 appelée (respectivement identifiée par les constantes CST\_NS\_USERFUNCT\_NCLEDITOR1% et CST\_NS\_USERFUNCT\_NCLEDITOR2%)
2. Lors de cet appel tous les champs sont renseignés sauf la configuration de compilation.
3. En retour ces deux fonctions utilisateur devront renseignées le code retour de la fonction à TRUE% si tout est OK ou à FALSE% si une erreur dans le code a été trouvée. Dans ce cas, les champs Err\_Message\$, Col\_Number% et Ligne\_Number% du segment SEG\_FCTPARAMOUT devront être renseignés. Le message d'erreur sera ainsi affiché dans la barre de statut de l'éditeur NCL et le curseur sera positionné aux coordonnées fournies par les deux champs Col\_Number% et Ligne\_Number%.
4. L'allocation et la désallocation des différentes structures passées en paramètre sont effectuées par NS-Design.

Exemple :

```
Function NS_USERFUNCT_NCLEDITOR% (Int FctNum%, SEG_FCTPARAMIN @NSPARAMIN,
SEG_FCTPARAMOUT @NSPARAMOUT) Return Int
  Local pointer hvisu%
  Local Int ret%(2)

  Evaluate FctNum%
  Where CST_NS_USERFUNCT_NCLEDITOR1%
  ...
  EndWhere
  Where CST_NS_USERFUNCT_NCLEDITOR2%
  ...
```

```
EndWhere  
EndEvaluate
```

```
Return ret%  
EndFunction
```

**Voir**     **aussi**     *CST NS USERFUNCT NCEDITOR1%*,     *CST NS USERFUNCT NCEDITOR2%*,  
*SEG FCTPARAMIN*, *SEG FCTPARAMOUT*

## Fonction NS\_USERFUNCT\_QUERYACTIONNAME\$

Cette fonction permet à NS-Design de paramétrer au niveau de l'outil le libellé des menu utilisateur (ActionName1 ...) mais aussi d'appeler ou non les différentes fonctions.

|                         |   |     |  |
|-------------------------|---|-----|--|
| <b>Syntaxe</b>          | <b>NS_USERFUNCT_QUERYACTIONNAME\$</b><br>(CST_NS_USERFUNCT_TYPE%) |     |  |
| <b>Paramètres</b>       | CST_NS_USERFUNCT_TYPE%  | INT | une des constantes<br>CST_NS_USERFUNCT_* |
| <b>Valeur retournée</b> | CSTRING   |     |  |

Exemple:

```
; Dans notre exemple, la deuxième fonction utilisateur ne sera pas appelée
car le libellé renvoyé est vide.
; On peut visualiser la présence ou non des différentes fonctions
utilisateurs au niveau de la fenêtre de détail des contrôles utilisateurs.
Function NS_USERFUNCT_QUERYACTIONNAME$(Int CST_NS_USERFUNCT_TYPE%) Return
Cstring
Local Cstring ActionName$

ActionName$ = ""
Evaluate CST_NS_USERFUNCT_TYPE%
Where CST_NS_USERFUNCT_NCLCHECK%
    ActionName$ = "Check"
EndWhere
Where CST_NS_USERFUNCT_PREBUILD%
    ActionName$ = "My PreBuild"
EndWhere
Where CST_NS_USERFUNCT_POSTBUILD%
    ActionName$ = "My PostBuild"
EndWhere
Where CST_NS_USERFUNCT_NCLEDITOR1%
    ActionName$ = "My Help 1"
EndWhere
Where CST_NS_USERFUNCT_NCLEDITOR2%
    ActionName$ = ""
EndWhere
Where CST_NS_USERFUNCT_GLOBAL1%
    ActionName$ = "Comparaison CVS"
EndWhere
Where CST_NS_USERFUNCT_GLOBAL2%
    ActionName$ = "Commit CVS"
EndWhere
else
    ActionName$ = ""
EndEvaluate

Return ActionName$
EndFunction
```

Voir aussi CST\_NS\_USERFUNCT\_\*

## Segment SEG\_FCTPARAMIN

Segment utilisé par les fonctions utilisateur pour les données en entrée renseignées par NS-Design.

| Champs           | Valeur  | Description                              |
|------------------|---------|--|
| Ncl_Fic\$        | CSTRING | fichier temporaire contenant le code NCL |
| Proj_Name\$      | CSTRING | nom du projet (.xnp)                     |
| Config_Name      | CSTRING | nom de la configuration de compilation   |
| Selected_topic\$ | CSTRING | sujet sélectionné                        |
| LibNcl_Name      | CSTRING | nom de la librairie NCL                  |
| Scr_Name\$       | CSTRING | nom de l'écran                           |
| Ctrl_Name\$      | CSTRING | nom du contrôle                          |
| Event_Name\$     | CSTRING | nom de l'événement                       |
| Target_Name\$    | CSTRING | nom de la target                         |

Suivant le contexte tous les champs ne pourront pas être renseignés.

***Voir aussi*** SEG\_FCTPARAMOUT

## Segment SEG\_FCTPARAMOUT

Segment utilisé par les fonctions utilisateur pour les données en sortie renseignées par l'utilisateur.

| Champs        | Valeur  | Description                               |
|---------------|---------|---|
| Err_Message\$ | CSTRING | libellé du message d'erreur               |
| Col_Number%   | INT     | numéro de la colonne où se situe l'erreur |
| Line_Number%  | INT     | numéro de la ligne où se situe l'erreur   |
| Severity%     | INT     | niveau de sévérité de l'erreur            |

1. Suivant le contexte tous les champs ne pourront pas être renseignés.
2. Le champ Severity% contient une des constantes NSINFOERROR%, NSWARNINGERROR ou NSFATALERROR%

**Voir aussi** SEG\_FCTPARAMIN, NSINFOERROR%, NSWARNINGERROR%, NSFATALERROR%



## INDEX

### C

Constantes de gestion des erreurs 14  
CST\_NS\_USERFUNCT\_GLOBAL1%  
13  
CST\_NS\_USERFUNCT\_GLOBAL2%  
13  
CST\_NS\_USERFUNCT\_NCLCHECK  
% 13  
CST\_NS\_USERFUNCT\_NCLEEDITOR  
1% 13  
CST\_NS\_USERFUNCT\_NCLEEDITOR  
2% 13  
CST\_NS\_USERFUNCT\_POSTBUILD  
% 13  
CST\_NS\_USERFUNCT\_PREBUILD%  
13

### L

Librairie de fonctions utilisateur  
NSFCTUTIL 5  
Librairie NSFCTUTIL 5  
N  
NS\_USERFUNCT\_BUILD% 15  
NS\_USERFUNCT\_GLOBAL% 17  
NS\_USERFUNCT\_NCLCHECK% 19  
NS\_USERFUNCT\_NCLEEDITOR% 20  
NS\_USERFUNCT\_QUERYACTIONNA  
ME\$ 22  
NSFATALERROR% 14  
NSINFOERROR% 14  
NSWARNINGERROR% 14  
S  
SEG\_FCTPARAMIN 23  
SEG\_FCTPARAMOUT 24