



<p>Manuel d'utilisation NS-Calc</p>
--

Table des matières

Introduction	7
Intégration dans NS-DK.....	9
Installation.....	9
Intégration dans un projet NS-DK	10
Génération	12
Livraison de l'application générée	13
Intégration dans NatStar	15
Installation.....	15
Intégration dans un projet NatStar	16
Génération	19
Livraison de l'application générée	20
Utilisation du tableur	21
Présentation de la feuille de calcul	22
Ligne de statut.....	22
Matrice de calcul	23
Notions de base	24
Ligne, colonne, cellule, bloc	24
Contenu des cellules	27
Définition et sélection de la cellule active	28
Production d'une feuille de calcul.....	29
Insertion de texte dans les cellules	30
Mise en place de formules de calcul.....	31
Saisie des données	32
Modification du contenu d'une cellule	33
Modification de la dimension des cellules.....	34
Sélection d'un bloc.....	36
Sélection de plusieurs blocs	38
Illustration de mise en forme	39
Formules de calcul	41
Affectation de formules aux cellules	42
Contenu d'une formule de calcul.....	43
Valeurs numériques.....	44
Valeurs alphanumériques.....	45
Opérateurs	46
Coordonnées de cellules	47
Fonctions.....	48
Fonctions classées par thème	49
Fonctions mathématiques	50
Fonctions trigonométriques	60

Fonctions statistiques.....	64
Fonctions logiques	70
Fonctions date & heure	71
Fonctions sur les chaînes	74
Fonctions de formatage.....	78
Fonctions sur événement.....	79
L'exemple NS-Calc	81
Introduction.....	81
Composantes du projet NS-Calc	82
Fenêtres.....	82
Librairies	85
Segments et variables.....	86
Icônes et bitmaps.....	87
Autres ressources	88
Utilisation de la librairie NS Calc au sein de l'exemple	89
Thèmes de l'API	99
Affichage de la fenêtre NS-Calc	99
Dimension de la feuille de calcul	100
Feuille à cellules variables	101
Dimension des cellules	102
Blocage des cellules	103
Cellule active, sélections, coordonnées	104
Noms logiques	105
Contenu des cellules	106
Format des cellules et de la feuille de calcul	107
Liens entre les cellules	110
Présentation	111
Modification des cellules et des blocs	112
Déplacement de blocs.....	113
Chargement, sauvegarde et impression	114
Gestion du presse-papiers	115
Gestion de la mémoire	116
Presentation Space	117
Référence de l'API	119
Avertissement 1 Au sujet de HCalc%, le handle Calc	119
Avertissement 2 Au sujet des paramètres !V!	120
Rappel Paramètre coord bloc.....	121
Evénements Classiques	122
Fonctions et Instructions Classiques	123
Fonctions Ns-Calc.....	124
Instructions Ns-Calc.....	223

Segments Ns-Calc.....	324
Constantes Ns-Calc.....	333
Index	353

INTRODUCTION

Le tableur NS Calc (Nat System Calculator) est une fenêtre contenant une feuille de calcul qui peut être intégrée dans toute application conçue avec NS DK (Nat System Development Kit) ou NatStar, et qui offre les fonctionnalités de base d'un tableur :

- organiser les informations en lignes et en colonnes (ce qui permet de voir d'un seul coup d'oeil toutes les données rassemblées),
- créer des structures (titres, en têtes, ...),
- rentrer ses propres formules de calcul,
- rentrer des données (dont le résultat est immédiatement affiché dans les cellules comportant des formules de calcul),
- imprimer les feuilles de calcul.

Comme tous les tableurs, NS Calc est organisé autour d'une feuille de calcul qui est opérationnelle dès l'ouverture de la fenêtre NS Calc et qui offre immédiatement les fonctions de calcul élémentaires.

NS-Calc - C:\NSCALC\SAMPLE\LOCAT.CAL									
File Edit Search View Format Options Windows Help									
R16C4		✓	✗	3320					
	1	2	3	4	5	6	7	8	9
1									
2	Mois	Intitulé	Date	Crédits	Crédits	Débets	Débets	Débets	Balance
3				Loyer	Rembt	Charges	Rép & div.	Impôts	
4	Janvier		02/01/92	3 250,00					3 250,00
5		1er trim.	09/01/92			4 810,20			-1 560,20
6		Concierge					200,00		-1 760,20
7	Février		09/02/92	3 250,00					1 489,80
8		Assur.	11/02/92		312,21				1 802,01
9		Taxe fonc.	21/02/92					2 990,00	-1 187,99
10	Mars		05/03/92	3 250,00					2 062,01
11		Ord. mén.	05/03/92		322,00				2 384,01
12	Avril		03/04/92	3 320,00					5 704,01
13		2ème trim.	05/04/92			4 810,20			893,81
14	Mai		11/05/92	3 320,00					4 213,81
15		Arr. Dupont	15/05/92		453,20				4 667,01
16	Juin		09/06/92	3 320,00					7 987,01
17		Plomb.	11/06/92				512,81		7 474,20
18	Total	Semestre		19 710,00	1 087,41	9 620,40	712,81	2 990,00	7 474,20
19									
20									
21									

Contrairement aux autres tableurs du commerce qui sont figés, NS Calc est un outil qui permet :

- de personnaliser l'application Tableur,
- de l'ajuster aux besoins réels de l'utilisateur final,
- de l'intégrer à toute autre application NS DK/NatStar,

ceci, grâce à la librairie de fonctions NCL fournie avec NS Calc.

Certains tableurs du marché proposés aujourd'hui sont très complets et offrent un grand nombre de fonctions. En contre partie, cela entraîne pour l'utilisateur une longue période de formation, pour exploiter rarement plus de 50% des possibilités offertes.

NS Calc permet au contraire de créer un tableur à la dimension des besoins de l'utilisateur final, en n'implémentant que les fonctions exploitées, d'appliquer le standard propre à la société utilisatrice, ce qui autorise une formation rapide de l'utilisateur, et d'intégrer des facilités de saisie.

Par exemple, `CALC_SETTHREAD` permet de passer automatiquement le focus (cellule active) à la cellule devant recevoir la prochaine donnée. L'exploitation de cette instruction permettra à l'utilisateur de gagner du temps et limitera les risques d'erreur lors de la saisie !

Ce manuel traite à la fois de l'utilisation et de la programmation du tableur NS Calc.

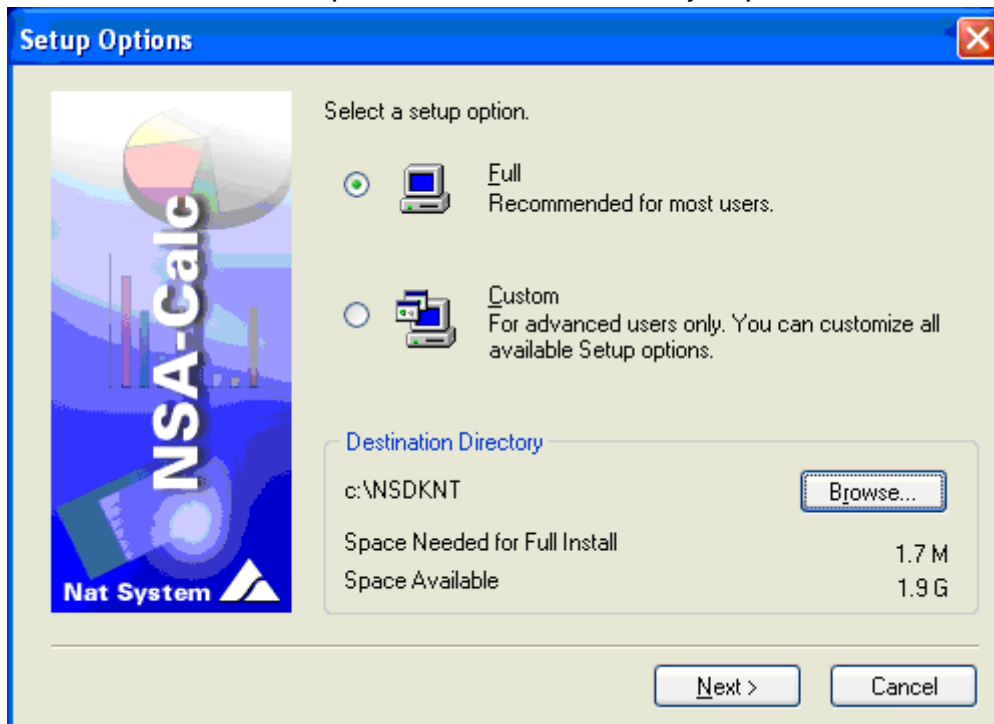
Le lecteur est supposé connaître NS-DK ou NatStar et être familier avec la programmation NCL.

INTEGRATION DANS NS-DK

Installation

Pour installer NS Calc :

1. Insérez le CD-ROM NS-Calc dans votre lecteur CD ROM. Activez Démarrer \ Exécuter. La boîte Exécuter s'ouvre.
2. Dans le champ de saisie, tapez : "E:\SETUP" et cliquez sur OK.
3. Sélectionnez le répertoire de NS-DK afin d'y copier les fichiers de NS Calc.

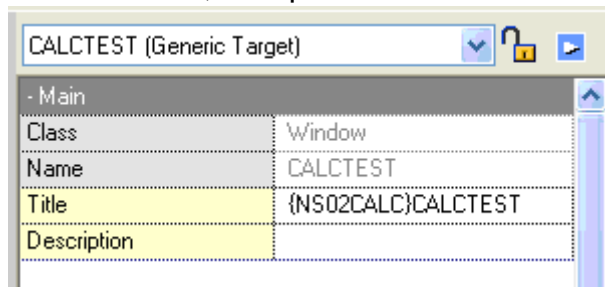


Un ensemble de possibilités vous est ensuite proposé. En fonction des choix que vous aurez effectués, tout ou une partie des fichiers (DLL, aide en ligne, exemple) seront copiés.

Intégration dans un projet NS-DK

Pour ajouter une fenêtre NS-Calc au sein d'un projet NS-DK, il faut, depuis NS-Design :

1. Ajouter la librairie NSCALC dans le panneau NSDK Resources,
2. Créer une nouvelle fenêtre de classe Window (par exemple CALCTEST),
3. Ouvrir le volet de propriétés de cette fenêtre, et rajouter {NS02CALC} en tête du champ Title (par exemple : {NS02CALC}CALCTEST), et en sélectionnant, si nécessaire, les options Horz. Scroll Bar et Vert. Scroll Bar.

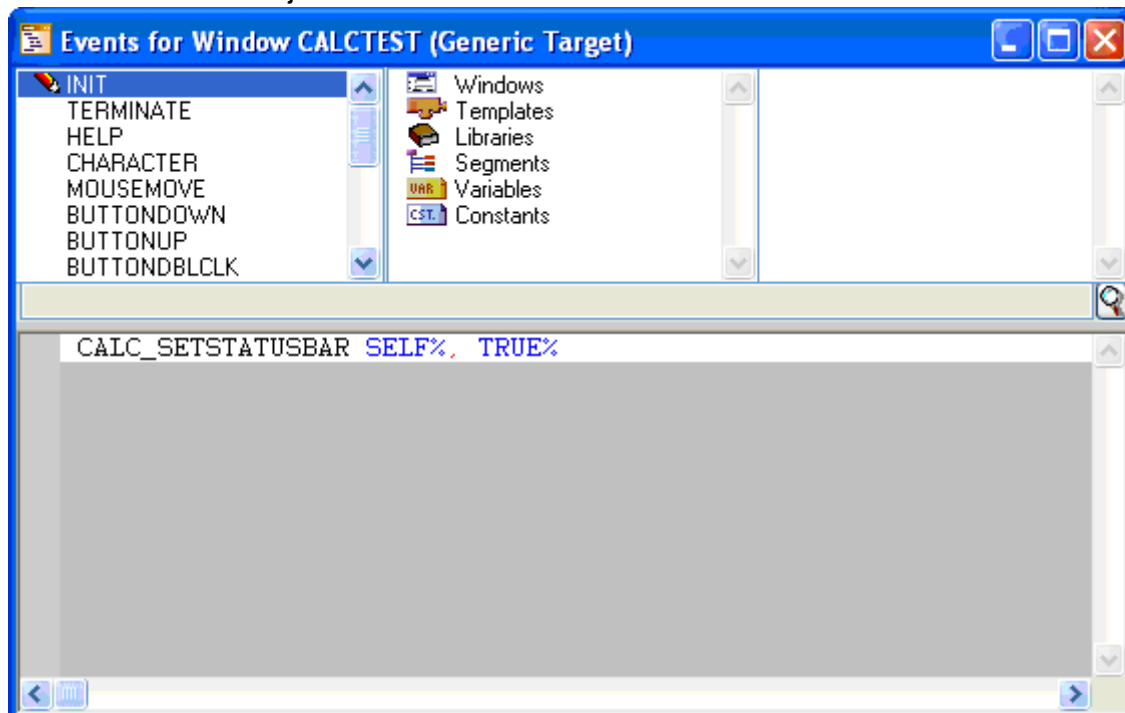


Pour que la fenêtre NS Calc apparaisse avec la ligne de statut :

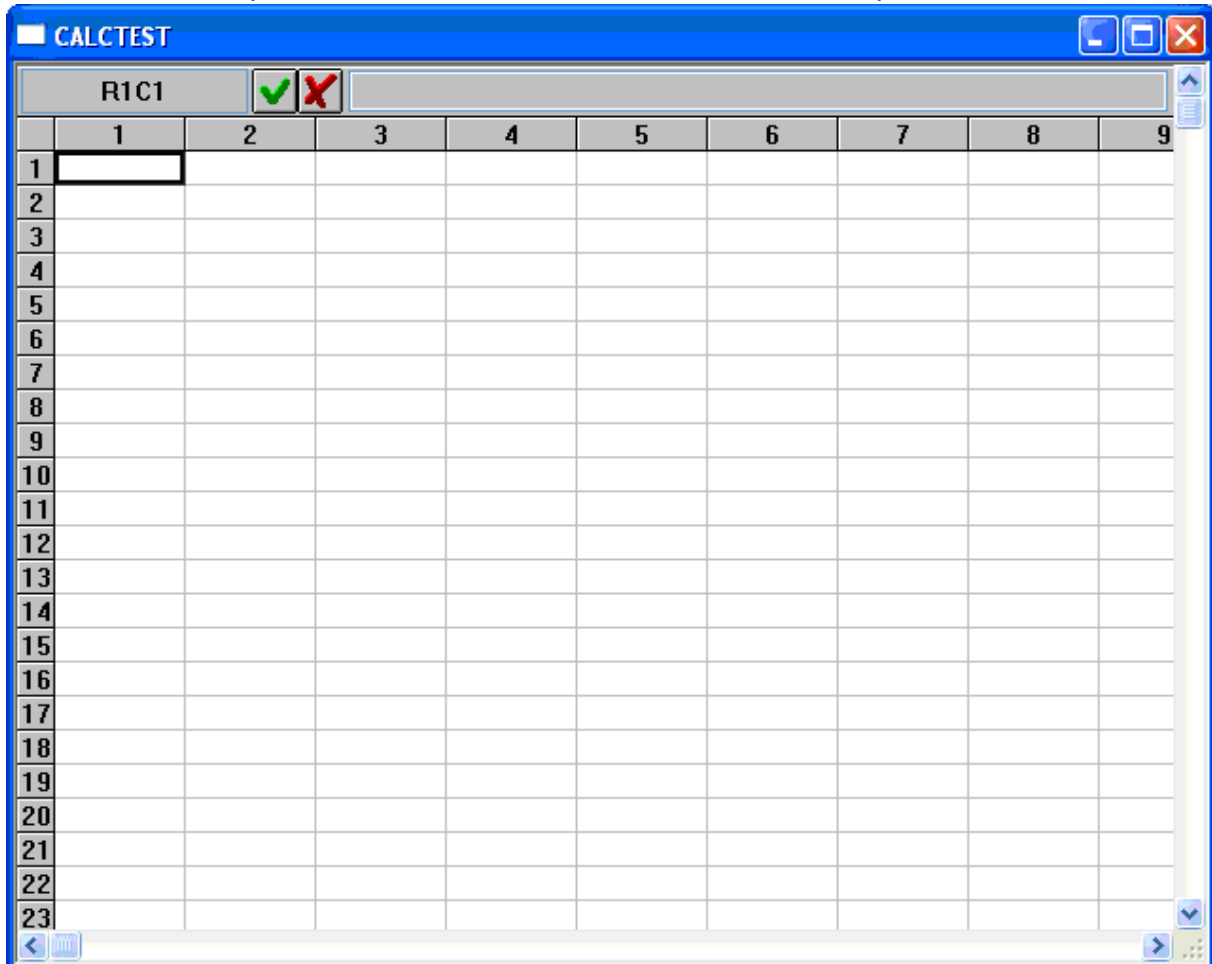
4. Ouvrir la boîte Evénements de cette fenêtre,
5. Coder sur l'événement INIT :

```
CALC_SETSTATUSBAR SELF%, TRUE%
```

L'instruction CALC_SETSTATUSBAR fait partie de l'A.P.I. rendue disponible par la librairie NSCALC rajoutée aux ressources.



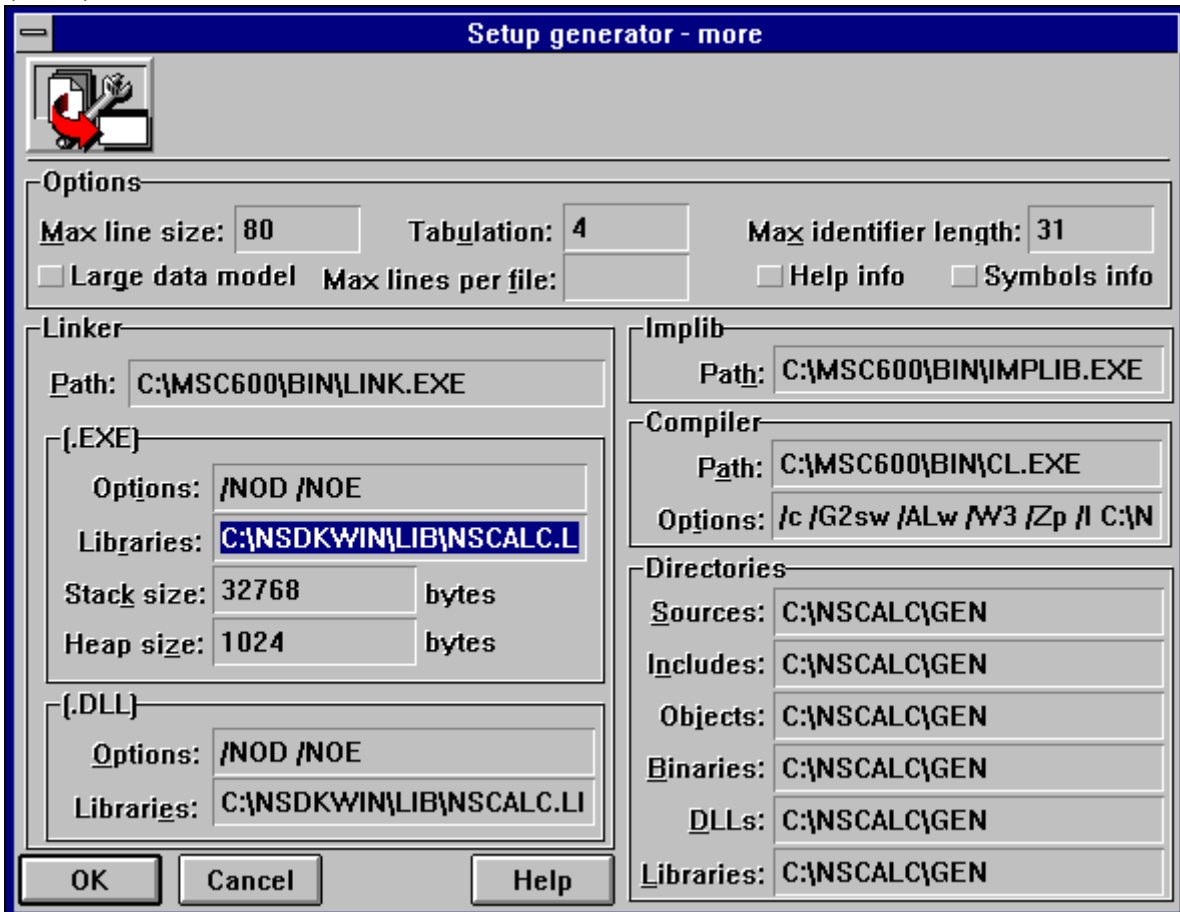
6. Appeler maintenant la touche [F5] pour tester cette fenêtre. La feuille de calcul apparaît, prête à l'emploi. La ligne de statut affiche les coordonnées de la cellule active et il est possible de saisir les données dans le champ de saisie.



La fenêtre NS Calc ne dispose pas, par défaut, des barres de défilement horizontal et vertical. Il faut les demander dans la boîte Window Info. Le titre réel apparaît, sans le {NS02CALC} qui a été rajouté dans le champ Title et qui permet de charger la DLL correspondante.

Génération

Pour générer une application NS DK (ou NatStar) comportant une (ou plusieurs) fenêtre(s) NS Calc, il est nécessaire de rajouter NSCALC.LIB dans les bibliothèques à utiliser lors de l'édition de liens. Pour cela, ouvrir la boîte Setup Generator More, et rajouter NSCALC.LIB dans les deux champs Libraries du Link, pour (.EXE) et pour (.DLL).



Setup generator - more

Options

Max line size: 80 Tabulation: 4 Max identifier length: 31

☐ Large data model Max lines per file: ☐ Help info ☐ Symbols info

Linker

Path: C:\MSC600\BIN\LINK.EXE

[.EXE]

Options: /NOD /NOE

Libraries: C:\NSDKWIN\LIB\NSCALC.L

Stack size: 32768 bytes

Heap size: 1024 bytes

[.DLL]

Options: /NOD /NOE

Libraries: C:\NSDKWIN\LIB\NSCALC.LI

Implib

Path: C:\MSC600\BIN\IMPLIB.EXE

Compiler

Path: C:\MSC600\BIN\CL.EXE

Options: /c /G2sw /ALw /W3 /Zp /I C:\N

Directories

Sources: C:\NSCALC\GEN

Includes: C:\NSCALC\GEN

Objects: C:\NSCALC\GEN

Binaries: C:\NSCALC\GEN

DLLs: C:\NSCALC\GEN

Libraries: C:\NSCALC\GEN

OK Cancel Help

Livraison de l'application générée

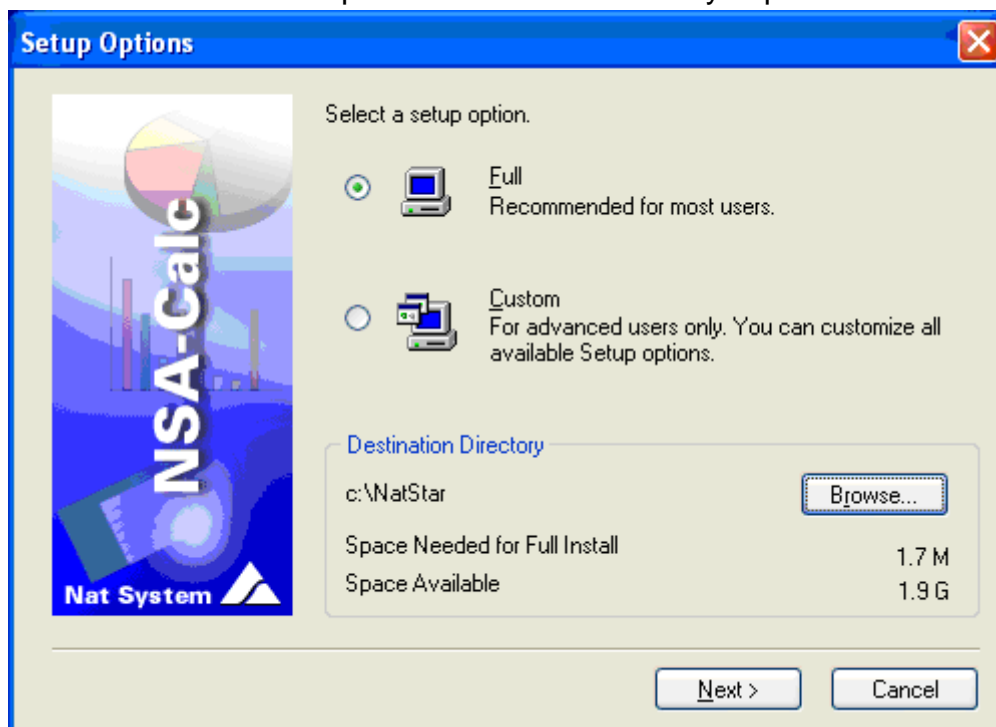
En plus de l'exécutable généré, le fichier NS02CALC.DLL doit être livré de même que les autres DLLs utilisées par l'application NS DK (NS02LIB.DLL et autres NS*.DLL).

INTEGRATION DANS NATSTAR

Installation

Pour installer NS Calc :

1. Insérez le CD-ROM NS-Calc dans votre lecteur CD ROM. Activez Démarrer \ Exécuter. La boîte Exécuter s'ouvre.
2. Dans le champ de saisie, tapez : "E:\SETUP" et cliquez sur OK.
3. Sélectionnez le répertoire de NatStar afin d'y copier les fichiers de NS Calc.

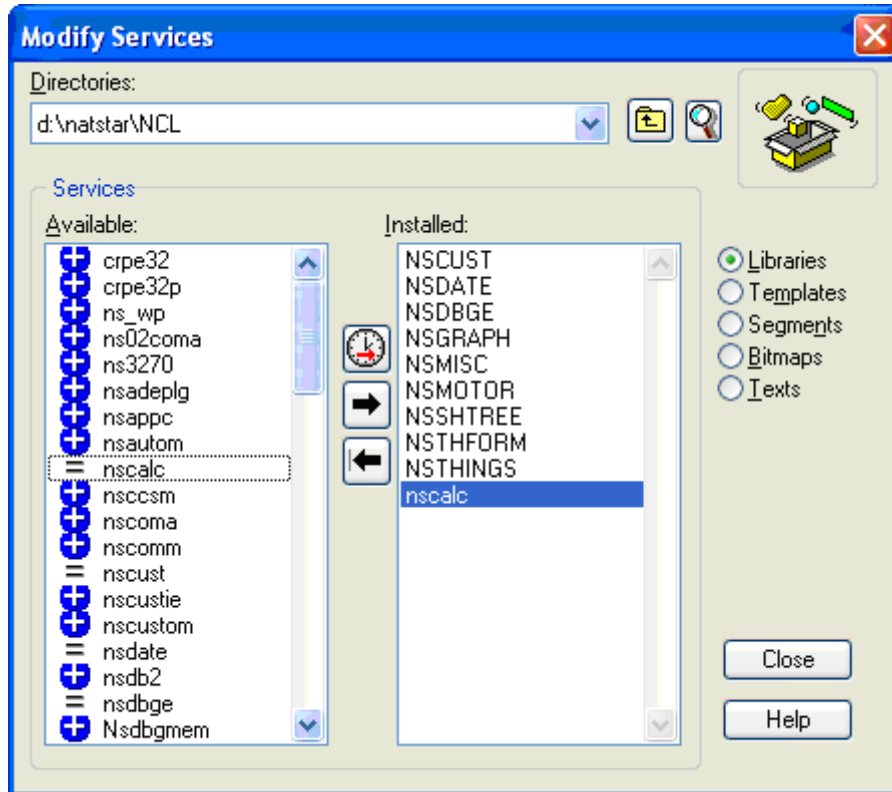


Un ensemble de possibilités vous est ensuite proposé. En fonction des choix que vous aurez effectués, tout ou une partie des fichiers (DLL, aide en ligne, exemple) seront copiés.

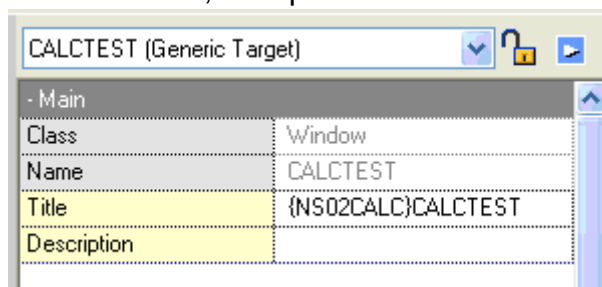
Intégration dans un projet NatStar

Pour ajouter une fenêtre NS Calc au sein d'un projet NatStar, il faut, depuis NatStar :

1. Ajouter la librairie NSCALC dans le menu Options\Services,



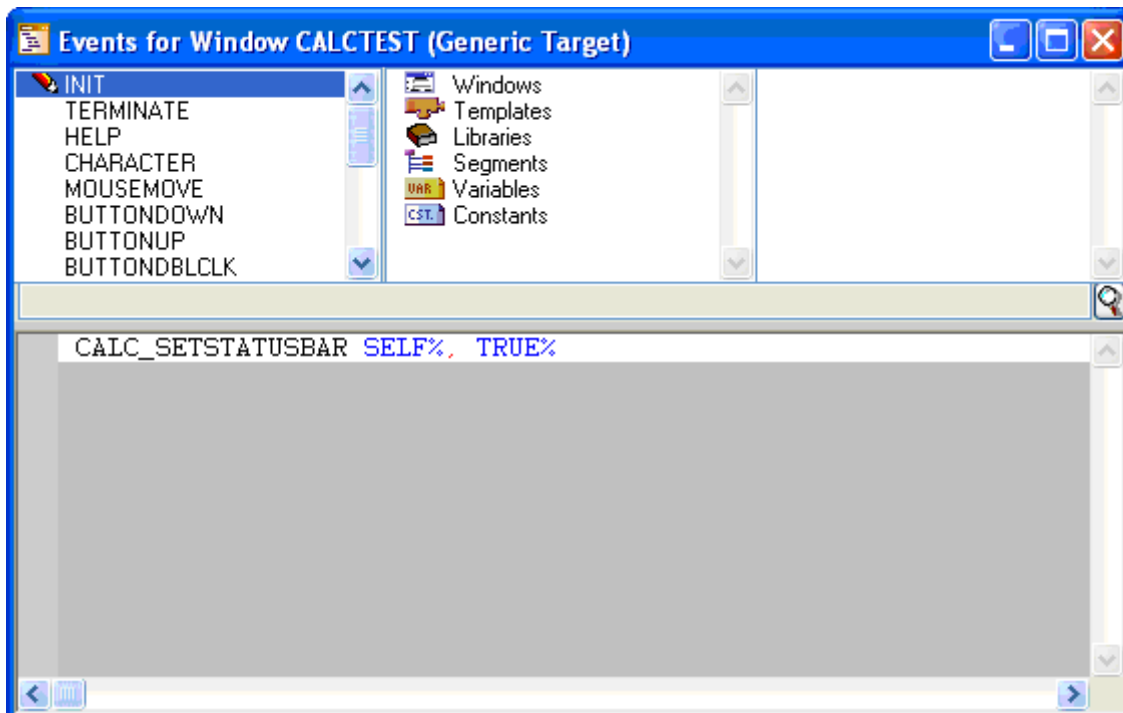
2. Créer une nouvelle fenêtre de classe Window (par exemple CALCTEST),
3. Ouvrir le volet de propriétés de cette fenêtre, et rajouter {NS02CALC} en tête du champ Title (par exemple : {NS02CALC}CALCTEST), et en sélectionnant, si nécessaire, les options Horz. Scroll Bar et Vert. Scroll Bar.



Pour que la fenêtre NS Calc apparaisse avec la ligne de statut :

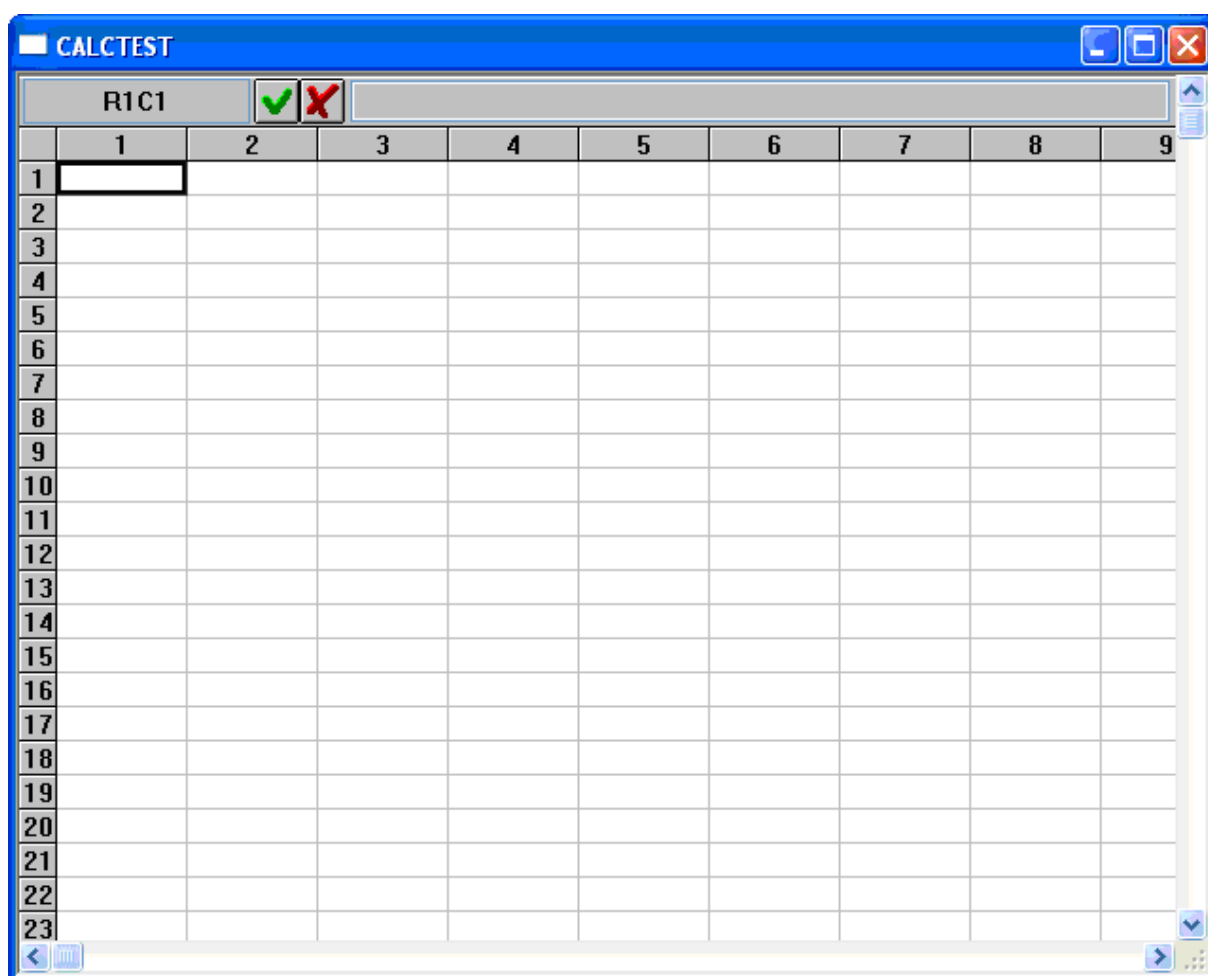
4. Ouvrir la boîte Événements de cette fenêtre,
5. Coder sur l'événement INIT :

```
CALC_SETSTATUSBAR SELF%, TRUE%
```

L'instruction `CALC_SETSTATUSBAR` fait partie de l'API rendue disponible par la librairie `NSCALC` rajoutée aux ressources.

6. Appeler maintenant la touche `[F5]` pour tester cette fenêtre. La feuille de calcul apparaît, prête à l'emploi. La ligne de statut affiche les coordonnées de la cellule active et il est possible de saisir les données dans le champ de saisie.

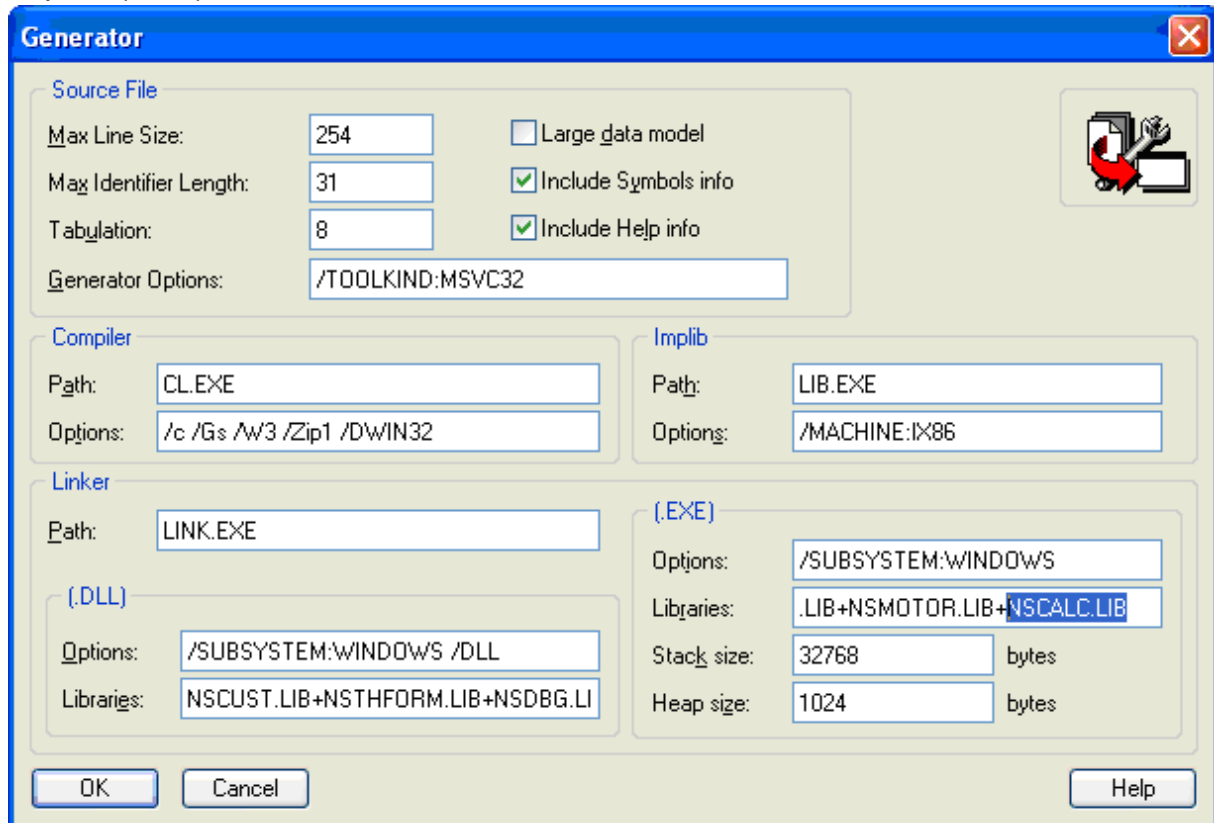


La fenêtre NS Calc ne dispose pas, par défaut, des barres de défilement horizontal et vertical. Il faut les demander dans le volet de propriétés. Le titre réel apparaît, sans le {NS02CALC} qui a été rajouté dans le champ Title et qui permet de charger la DLL correspondante.

Génération

Pour générer une application NatStar comportant une (ou plusieurs) fenêtre(s) NS Calc, il est nécessaire de rajouter NSCALC.LIB dans les librairies à utiliser lors de l'édition de liens.

Pour NatStar, ouvrir la boîte Select Default Configuration, puis cliquer sur le bouton Gen ... et rajouter NSCALC.LIB dans les deux champs Librairies du Link, pour (.EXE) et pour (.DLL).



Generator

Source File

Max Line Size: 254 ☐ Large data model

Max Identifier Length: 31 ☒ Include Symbols info

Tabulation: 8 ☒ Include Help info

Generator Options: /TOOLKIND:MSVC32

Compiler

Path: CL.EXE

Options: /c /Gs /W3 /Zlp1 /DWIN32

Linker

Path: LINK.EXE

(.DLL)

Options: /SUBSYSTEM:WINDOWS /DLL

Libraries: NSCUST.LIB+NSTHFORM.LIB+NSDBG.LI

(.EXE)

Options: /SUBSYSTEM:WINDOWS

Libraries: .LIB+NSMOTOR.LIB+NSCALC.LIB

Stack size: 32768 bytes

Heap size: 1024 bytes

OK Cancel Help

Livraison de l'application générée

En plus de l'exécutable généré, le fichier NS02CALC.DLL doit être livré de même que les autres DLLs utilisées par l'application NatStar (NS02LIB.DLL et autres NS*.DLL).

UTILISATION DU TABLEUR

Ce chapitre rappelle les définitions de base nécessaires à l'utilisation d'un tableur, les différentes fonctionnalités offertes par le tableur NS Calc et les connaissances nécessaires à son exploitation.

Vous trouverez une présentation :

- des définitions concernant une feuille de calcul,
- des notions de base,
- des informations concernant la production d'une feuille de calcul.

Pour vous exercer à la manipulation de ce tableur, démarrez l'exemple NSCALC.EXE, fourni avec le produit NS Calc.

Présentation de la feuille de calcul



Ligne de statut





La ligne de statut, placée en dessus de la matrice de calcul, est divisée en trois parties :

- La partie gauche est le champ dans lequel s'affichent les coordonnées de la cellule active (R2C3 dans l'exemple). Le contenu de ce champ est automatiquement actualisé par NS Calc lors de la sélection d'une cellule par les touches clavier ou par la souris.

Remarque : au cours de la sélection d'un bloc, ce champ affiche le nombre de lignes et de colonnes sélectionnées sous la forme "nR x mC" dans laquelle n désigne le nombre de lignes et m le nombre de colonnes.

- La partie centrale comporte deux boutons ( et ) permettant respectivement de valider ou d'annuler la dernière saisie effectuée.

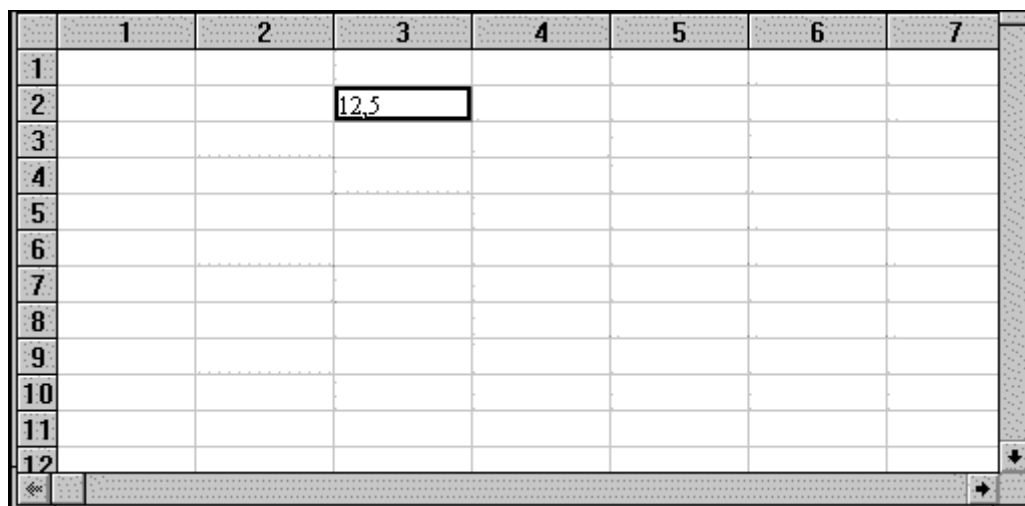
Un clic sur  est équivalent à l'appui sur la touche [Enter], un clic  est équivalent à l'appui sur la touche [Esc].

- La partie droite est le champ de saisie du contenu des cellules.

L'activation de ce champ est automatique après la sélection d'une cellule, même si le point d'insertion clignotant du curseur n'est pas visible.

Ce champ permet d'effectuer directement la saisie du texte, de la valeur numérique ou de la formule qui sera affectée à la cellule active.

Matrice de calcul



	1	2	3	4	5	6	7
1							
2			12,5				
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

La matrice de calcul, souvent appelée, par extension, feuille de calcul, est le tableau situé sous la ligne de statut. Elle permet d'organiser en lignes et en colonnes les informations présentées.

La matrice de calcul de NS Calc peut comporter jusqu'à 16249 colonnes et 16249 lignes.

Par défaut, les numéros de lignes et de colonnes sont affichés. Dans ce cas, la première ligne de la matrice de calcul comporte la numérotation des colonnes (de 1 à 256 par défaut), la première colonne comporte la numérotation des lignes (de 1 à 16249 par défaut).

Comme la totalité de cette matrice ne peut pas être représentée dans la page écran, celle-ci peut comporter à droite, une barre de défilement vertical et en bas, une barre de défilement horizontal. Ces deux barres, munies d'ascenseurs et de flèches de direction permettent d'effectuer le défilement de la matrice à l'aide de la souris.

Notions de base

Ligne, colonne, cellule, bloc

Lors de la création d'une fenêtre NS Calc, la feuille de calcul est initialisée par défaut à 256 colonnes (pour 16249 colonnes possibles) et 16249 lignes.

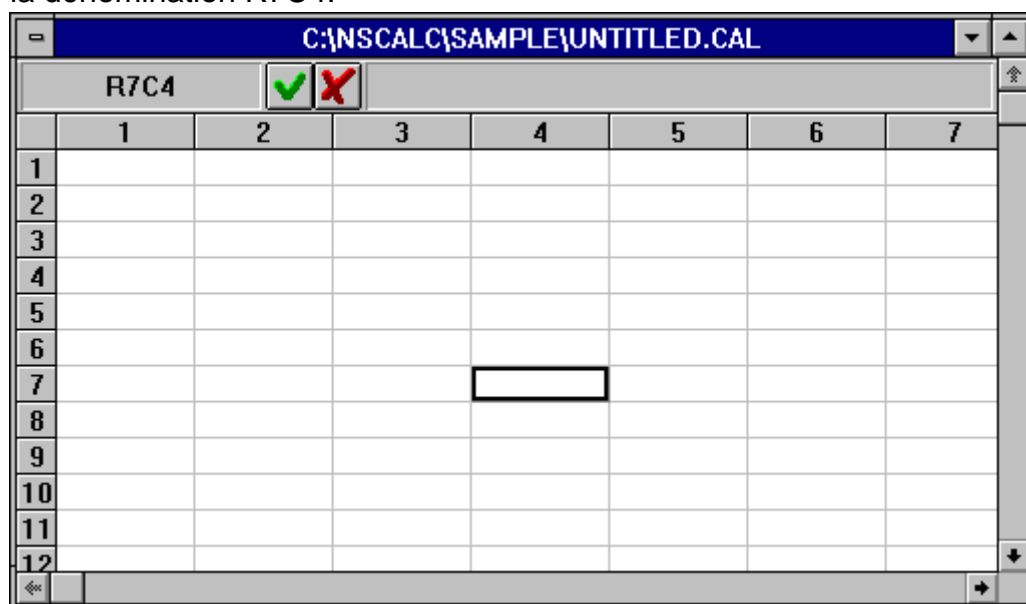
Les lignes sont identifiées par Ry (Rows) où y représente l'ordonnée et peut prendre n'importe quelle valeur entière positive comprise entre 1 et le nombre maximum de lignes de la feuille de calcul (16249 par défaut).

Les colonnes sont identifiées par Cx (Columns) où x représente l'abscisse et peut prendre n'importe quelle valeur entière positive comprise entre 1 et le nombre maximum de colonnes de la feuille de calcul (256 par défaut).

L'intersection d'une ligne et d'une colonne de la feuille de calcul forme une cellule.

Les cellules sont identifiées sous la forme RnCm correspondant à l'intersection de la ligne et de la colonne correspondantes.

Ainsi, la cellule située à l'intersection de la ligne 7 et de la colonne 4 est identifiée par la dénomination R7C4.

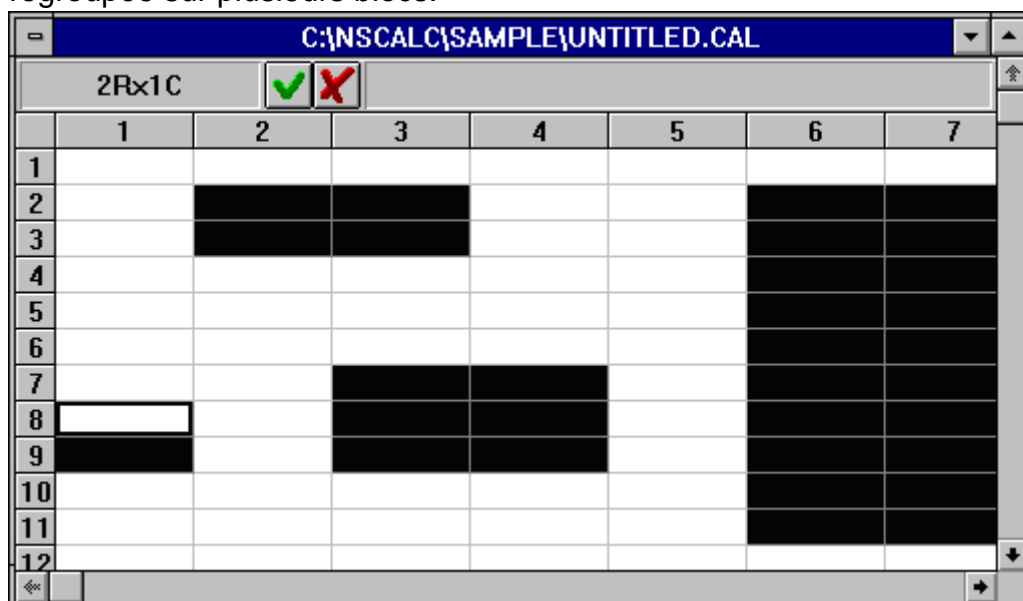


L'identification se fait toujours dans l'ordre "Ligne (donc ordonnée en général représentée par y) Colonne (donc abscisse en général représentée par x)".

NS-Calc permet de grouper plusieurs cellules pour former un bloc. Un bloc peut être créé par l'utilisateur avec la souris en cliquant sur la cellule représentant un des coins du bloc et en déplaçant la souris en gardant le bouton gauche appuyé : la zone parcourue par la souris est sélectionnée sous forme de bloc. La même opération peut être effectuée avec le clavier en appuyant sur la touche [Shift] tout en déplaçant le curseur avec les touches de direction du curseur.

Pour créer plusieurs blocs avec la souris, il suffit de créer un bloc puis déplacer la souris sur le nouveau bloc à créer et cliquer sur un de ses coins en maintenant la touche [Ctrl] appuyée : le nouveau bloc sera sélectionné sans désélectionner l'autre

bloc. Vous pouvez réitérer l'opération et créer ainsi une sélection complexe regroupée sur plusieurs blocs.



	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

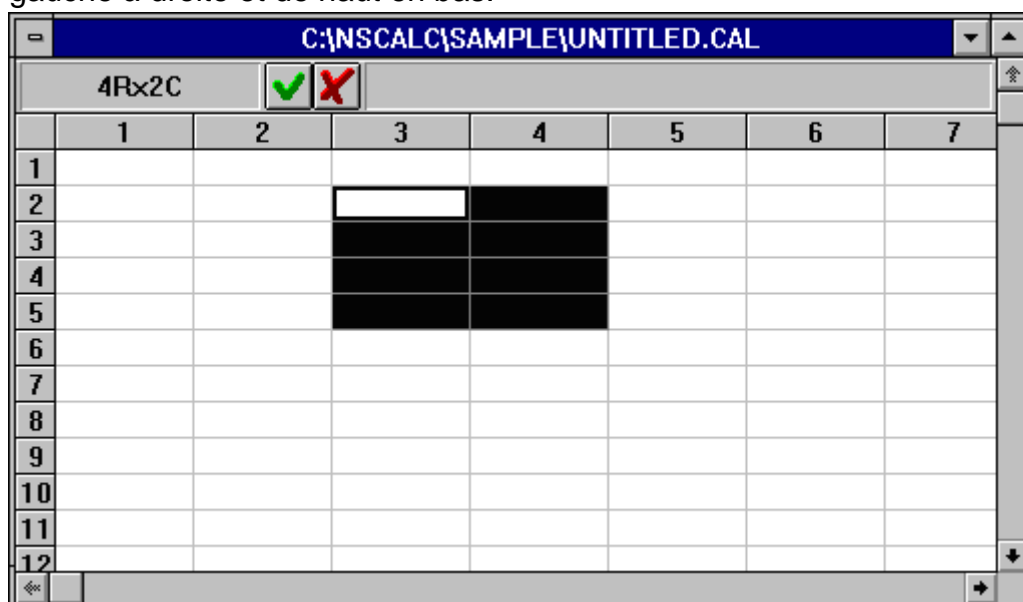
Un bloc est identifié par la définition des coordonnées de deux cellules, séparée par le caractère deux points ":". Ces coordonnées représentent respectivement :

- la première cellule du bloc (située en haut et à gauche).
- la dernière cellule du bloc (située en bas et à droite).

Exemple :

R2C3:R5C4 définit simultanément à NS Calc les 8 cellules : R2C3, R2C4, R3C3, R3C4, R4C3, R4C4, R5C3, R5C4

Remarquer l'ordre de définition en "Z", identique à l'ordre de lecture occidentale : de gauche à droite et de haut en bas.



	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

1. Si une définition de bloc est erronée, (par exemple inversion de l'ordre des coordonnées de cellules "R5C4:R2C3") NS Calc n'affectera aucun bloc.

2. Il est possible d'étendre la notion de bloc à une seule rangée ou à une seule colonne de cellules. Dans ce cas, la définition se fera en désignant seulement la rangée ou la colonne correspondante.

Exemple :

"R2" désigne toutes les cellules de la deuxième ligne et "C4" désigne toutes les cellules de la quatrième colonne.

3. Pour les instructions et fonctions NS Calc, la définition d'une cellule ou d'un bloc est toujours passée sous forme de chaîne de caractères.

Exemple :

"R2C3" pour une cellule ou "R2C2:R4C8" pour un bloc.

4. Certaines instructions et fonctions NS Calc telles que CALC_CLEAR ou CALC_PUT acceptent que plusieurs blocs soient définis dans la même chaîne de caractères. Dans ce cas, chaque bloc est séparé par le caractère point virgule ";".

Exemple :

"R2C2:R3C4;R6C10:R6C12;C14:C15;R18C20" définit simultanément :

les cellules R2C2, R2C3, R2C4, R3C2, R3C3, R3C4

les cellules R6C10, R6C11, R6C12

toutes les cellules des colonnes 14 et 15

la cellule R18C20

Contenu des cellules

Si le programme d'application ne comporte pas de paramètres filtrant les données saisies, une cellule NS Calc peut recevoir :

- Du texte

Tous les caractères alphanumériques sont admis. Les textes seront en général utilisés pour définir des titres, ou pour expliciter les résultats des cellules adjacentes.

- Des données numériques

L'entrée de données numériques permet d'effectuer les calculs et de modifier immédiatement le contenu des cellules concernées, c'est bien sûr la principale fonction du tableur.

- Des formules

Les formules contenues dans les cellules peuvent être soit saisies, soit établies par programmation à partir de NS Calc lors de la création de l'application.

- Des variables

La définition des variables affectées aux cellules est faite en général par programmation à partir de NS Calc lors de la création de l'application.

Définition et sélection de la cellule active

La cellule active est la cellule à laquelle seront affectées les données provenant du champ de saisie de la ligne de statut ou depuis laquelle seront extraites les données affichées dans ce champ.

Sur la feuille de calcul, la cellule active est représentée par un trait d'encadrement plus épais.

Une seule cellule peut être active à un instant donné.

L'activation d'une cellule de la feuille de calcul se fait :

1. En déplaçant le curseur souris sur la cellule à activer puis en cliquant avec le bouton gauche de la souris (pour activer une cellule qui n'est pas visible dans la fenêtre, faire d'abord défiler la feuille de calcul à l'aide des ascenseurs).
2. A partir du clavier à l'aide des touches de déplacement curseur (voir en annexe l'action des touches clavier et de leurs différentes combinaisons).



Production d'une feuille de calcul

Toute feuille de calcul peut comporter des cellules contenant :

- des titres et du texte (en général pour expliciter les résultats),
- des résultats (donc contenant des formules),
- les données saisies.

Nous avons choisi d'illustrer la création d'une feuille de calcul par un exemple simple.

Dans cette première création, nous exploitons les possibilités brutes de la feuille de calcul (sans formater les valeurs numériques). C'est ce qui explique l'absence de présentation des résultats dans l'exemple.

C:\NSCALC\SAMPLE\EX1.CAL*						
R8C3				=R7C2+R7C3		
	1	2	3	4	5	6
1	VENTES					
2		Article n°1	Article n°2			
3	Prix unitaire	512,32	1420,1			
4	Nombre d'articles	11	4			
5	Total H.T.	5635,52	5680,4			
6	T.V.A. 18.6%	1048,20672	1056,5544			
7	TOTAL T.T.C.	6683,72672	6736,9544			
8	TOTAL Général		13420,68112			
9						
10						
11						
12						

Insertion de texte dans les cellules

1. Sélectionner la cellule dans laquelle devra s'inscrire le titre.
2. Entrer directement au clavier le texte à inscrire dans cette cellule.
3. Valider par la touche [Enter], ou sélectionner avec la souris la prochaine cellule à activer.

Dans notre exemple nous avons respectivement entré :

- VENTES dans la cellule R1C1
- Article n°1 dans la cellule R2C2
- et Article n°2 dans la cellule R2C3
- Prix unitaire dans la cellule R3C1
- etc...

Contrairement à la plupart des tableurs, il ne faut pas déclarer un texte par l'ouverture de guillemets (") car ceux ci seraient représentés dans le texte. Le contenu du texte peut comporter tous les caractères alphanumériques. En fonction des caractères rentrés, NS Calc attribue automatiquement un type au contenu de la cellule. Si une cellule doit contenir des nombres au format texte, ceux ci ne doivent pas représenter de valeur numérique. Exemple : 1 2 3 est considéré comme texte car il y a au moins un caractère non numérique (espace)

Mise en place de formules de calcul

1. Sélectionner la cellule dans laquelle devra s'inscrire la formule,
2. Entrer au clavier le signe égal (=) pour déclarer une formule, puis la formule de calcul correspondant au résultat à obtenir,
3. Valider par la touche [Enter], ou sélectionner avec la souris la prochaine cellule à activer.

Dans l'exemple, nous avons respectivement entré :

=R3C2*R4C2 dans R5C2 et =R3C3*R4C3 dans R5C3

=R5C2*0.186 dans R6C2 et =R5C3*0.186 dans R6C3

=R5C2+R6C2 dans R7C2 et =R5C3+R6C3 dans R7C3

=R7C2+R7C3

dans

R8C3

Si une cellule doit contenir une formule de calcul, le premier caractère à saisir pour déclarer une formule sera toujours le signe égal (=). La suite de la formule pourra contenir des valeurs numériques, des opérateurs arithmétiques, des références de cellules.

Voir chapitre "Formules de calculs" pour un complément d'informations.

Saisie des données

1. Sélectionner la cellule dans laquelle devra s'inscrire la donnée correspondante,
2. Entrer au clavier la valeur numérique,
3. Valider par la touche [Enter], ou sélectionner avec la souris la prochaine cellule à activer.

Dans l'exemple nous avons uniquement saisi les données contenues dans les cellules R3C2, R3C3, R4C2 et R4C3 : NS Calc effectue automatiquement les calculs et affiche les résultats. Si une cellule doit contenir des données, celles ci doivent être entrées sous forme de valeurs numériques (ces valeurs seront considérées comme entières ou réelles en fonction des filtres prévus dans l'application).

Modification du contenu d'une cellule

Pour modifier le contenu d'une cellule, cliquer (ou sélectionner par les touches curseur) la cellule concernée : son contenu est automatiquement transféré dans le champ de saisie.

Dès l'appui d'une touche, le contenu de ce champ est immédiatement effacé. Pour éviter une nouvelle saisie complète, cliquer dans le champ de saisie avec la souris, son contenu apparaît en vidéo inverse. Le déplacement du curseur (ou un nouveau clic de la souris) à l'emplacement où doit avoir lieu la modification annule l'inversion vidéo, et autorise la modification.

Si le contenu de la cellule a été effacé après l'appui d'une touche, celui ci est récupérable par l'appui de la touche [Esc] ou par un clic sur le bouton d'annulation de la ligne de statut.

Modification de la dimension des cellules

Hauteur

Le dimensionnement en hauteur affecte toutes les cellules appartenant à la même ligne.

Il peut être effectué :

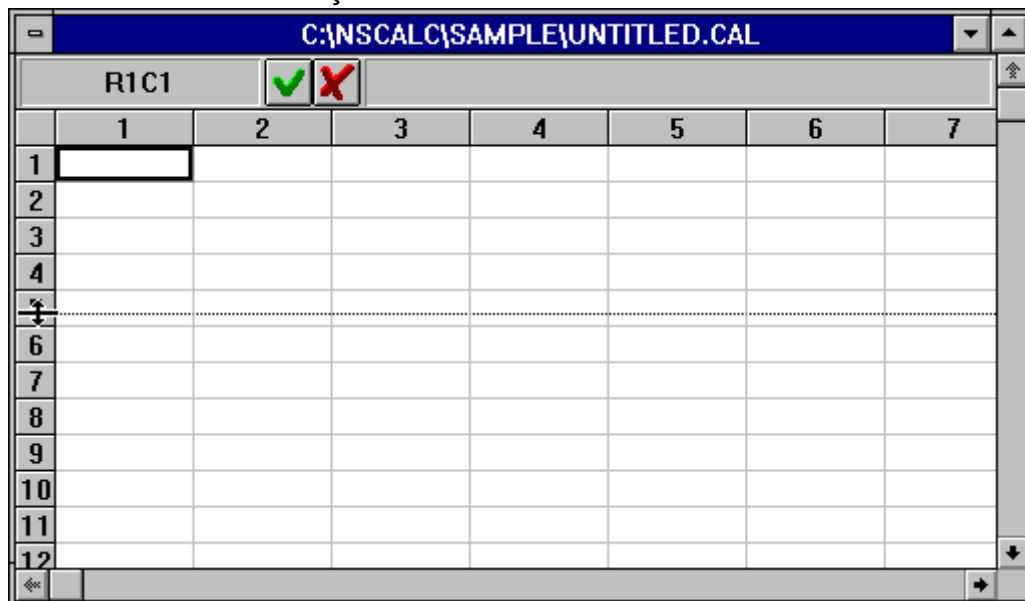
- par programmation par l'instruction `CALC_SETHEIGHT`,
- avec la souris depuis la feuille de calcul :

Si les numéros de lignes et de colonnes sont affichés :

1. Déplacer verticalement le pointeur souris de façon à l'amener sur la colonne affichant les numéros de lignes, entre le bas de la cellule dont la hauteur doit être modifiée et le haut de la cellule suivante, sur la ligne horizontale de séparation, sinon, déplacer verticalement le pointeur souris de façon à l'amener entre le bas de la cellule dont la hauteur doit être modifiée et le haut de la cellule suivante, sur la ligne horizontale de séparation.

Le pointeur prend alors la forme d'une double flèche haut bas séparée par un trait horizontal.

2. Cliquer sur le bouton gauche de la souris et faire glisser le pointeur vers le haut ou vers le bas de façon à obtenir la hauteur désirée.



Dans l'exemple NS-Calc : par l'option Row height du menu Format.

Largeur

Le dimensionnement en largeur affecte toutes les cellules appartenant à la même colonne.

Il peut être effectué :

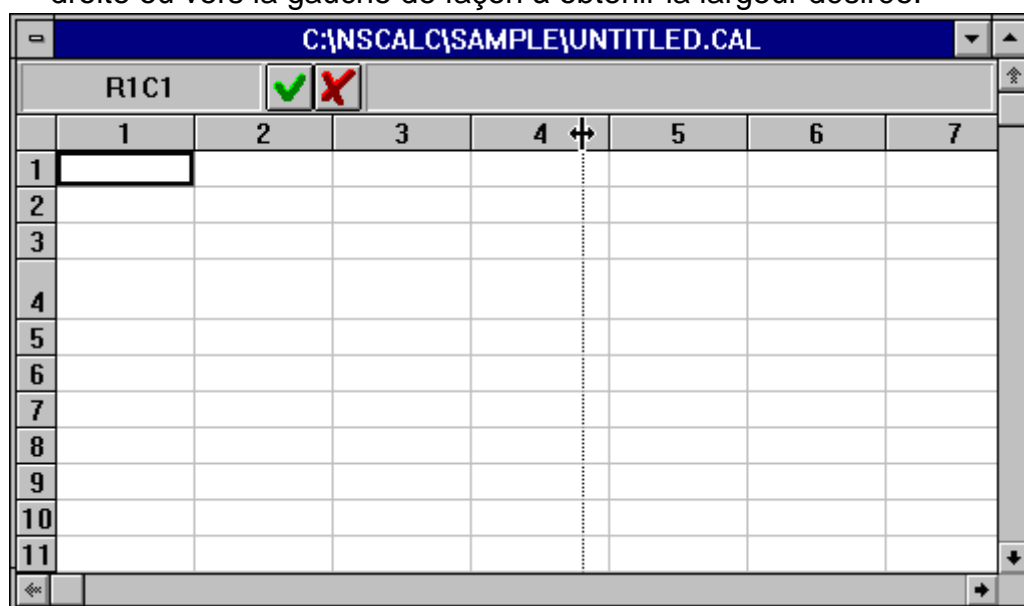
- par programmation par l'instruction `CALC_SETWIDTH`,
- avec la souris depuis la feuille de calcul :

Si les numéros de lignes et de colonnes sont affichés :

1. Déplacer horizontalement le pointeur souris de façon à l'amener sur la ligne affichant les numéros de colonne, entre le bord droit de la cellule dont la largeur doit être modifiée et le bord gauche de la cellule suivante, sur la ligne verticale de séparation, sinon, déplacer horizontalement le pointeur de façon à l'amener entre le bord droit de la cellule dont la largeur doit être modifiée et le bord gauche de la cellule suivante, sur la ligne verticale de séparation.

Le pointeur prend alors la forme d'une double flèche gauche droite séparée par un trait vertical.

2. Cliquer sur le bouton gauche de la souris et faire glisser le pointeur vers la droite ou vers la gauche de façon à obtenir la largeur désirée.



Dans l'exemple NS-Calc : par l'option `Column width` du menu `Format`.

En diminuant la hauteur d'une ligne ou la largeur d'une colonne de cellules à l'aide de la souris, il peut arriver que la ligne ou la colonne disparaisse car trop petite. Elle n'est pas pour autant supprimée de la feuille de calcul.

Sélection d'un bloc

Avec la souris

1. Amener le curseur sur la cellule initiale et cliquer le bouton gauche de la souris, puis, tout en maintenant le bouton gauche enfoncé, déplacer le pointeur souris sur les cellules devant être sélectionnées.

Le même résultat peut être obtenu en cliquant d'abord sur la cellule initiale, puis en cliquant sur la cellule finale tout en maintenant la touche [Shift] enfoncée.

2. La feuille de calcul est automatiquement déplacée dans la fenêtre de façon à garder visible la cellule active même si la sélection passe en dehors de la fenêtre.
 - a) Pour sélectionner une ou plusieurs lignes complètes de cellules, cliquer, si les numéros de lignes et de colonnes sont affichés, sur la colonne de gauche de la feuille de calcul (colonne contenant les numéros de lignes).
 - b) Pour sélectionner une ou plusieurs colonnes complètes de cellules, cliquer, si les numéros de lignes et de colonnes sont affichés, sur la ligne du haut de la feuille de calcul (ligne contenant les numéros de colonnes).

A partir du clavier

1. Choisir la cellule active initiale à l'aide des touches de déplacement curseur.
2. Appuyer sur la touche [Shift] et, tout en la maintenant enfoncée, déplacer le curseur par les touches de déplacement pour inclure les nouvelles cellules dans la sélection.
 - a) Avec la touche [Home], la sélection portera de la cellule R1C1 à la cellule initiale.
 - b) Avec la touche [End], la sélection portera de la cellule initiale à la dernière cellule de la fenêtre.
 - c) Avec la touche [PageDown], la sélection portera de la cellule initiale à la cellule du bas de fenêtre.
 - d) Avec la touche [PageUp], la sélection portera de la cellule initiale à la cellule du haut de fenêtre.

Sélection de plusieurs blocs

Pour sélectionner plusieurs blocs, répéter les actions précédentes en maintenant la touche [Ctrl] appuyée pour chaque nouveau bloc.

Illustration de mise en forme

Voici comment peut être présenté l'exemple vu en début de chapitre (paragraphe 'Production d'une feuille de calcul') :

C:\NSCALC\SAMPLE\SAMPLE.CAL							
R14C1							
	1	2	3	4	5	6	7
1							
2	VENTES						
3		Article n°1	Article n°2	(a)			
4	Prix unitaire	512,32	1 420,10	(b)			
5							
6	Nombre d'articles	11	4	(c)			
7							
8	Total H.T.	5 635,52	5 680,40				
9	T.V.A. 18.6%	1 048,21	1 056,55				
10	TOTAL T.T.C.	6 683,73	6 736,95				
11							
12	TOTAL Général		13 420,68	(d)			
13							
14							

- a) Alignement : Centré Encadrement sur les quatre côtés
- b) Alignement : Défaut (gauche pour les textes, droite pour les nombres)
Format numérique : # ###0.00 Encadrement : Gauche + Droit Type : Numérique
- c) Alignement : Défaut Format numérique : Défaut Encadrement : Gauche + Droit
- d) Alignement : Défaut Format numérique : # ###0.00 Encadrement : Quatre côtés Type : Formule Couleur : Texte blanc, Fond noir

FORMULES DE CALCUL

Comme nous l'avons vu dans les chapitres précédents, une feuille de calcul comporte des cellules pouvant contenir du texte, des données et/ou des résultats.

Le véritable intérêt d'une feuille de calcul étant d'afficher les résultats au fur et à mesure de la saisie des données, certaines cellules devront contenir les formules de calcul permettant d'obtenir automatiquement ces résultats.

Ce chapitre présente les différentes formules de calcul du tableur NS Calc.

Affectation de formules aux cellules

L'affectation d'une formule de calcul à une cellule se fait grâce au signe égal (=) en premier caractère, suivi de la formule.

Exemple

```
=@SUM(R1C1:R2C2)
```

Dès la saisie des données, et sauf interdiction par CALC_UPDATE, la cellule ainsi affectée affichera immédiatement le résultat dans la feuille de calcul.

Contenu d'une formule de calcul

Une formule de calcul peut comporter les paramètres suivants :

- des valeurs numériques,
- des valeurs alphanumériques,
- des opérateurs arithmétiques, logiques, sur les chaînes, ...,
- des coordonnées de cellules et/ou de blocs,
- des fonctions.

Lorsqu'une formule est constituée d'une constante numérique, le signe égal (=) utilisé pour l'affectation des cellules est optionnel.

Exemples :

```
=123 ;équivalent à 123  
=2*2  
=R2C3+R4C5  
=@SUM(R2C3:R4C7)
```

Valeurs numériques

Les valeurs numériques sont des nombres entiers ou réels, éventuellement précédés de leur signe.

Pour une valeur numérique, les seuls caractères admis sont les 10 chiffres de 0 à 9, le séparateur décimal (caractère point "." ou caractère virgule ",") et la négation (" "), ainsi que les caractères "E", "+", "-" pour l'exposant.

Exemple :

12,5

Valeurs alphanumériques

Les valeurs alphanumériques sont tout texte comportant au moins un caractère non numérique.

Exemple :

12 000 FHT

Opérateurs

Les opérateurs arithmétiques reconnus sont :

- la négation
- l'addition +
- la soustraction
- la multiplication *
- la division /

Les opérateurs de concaténation de chaînes de caractères sont :

- la concaténation simple &
- la concaténation avec espace &&

Les opérateurs de comparaison sont :

- égal à =
- différent de <>
- inférieur à <
- inférieur ou égal à <=
- supérieur à >
- supérieur ou égal à >=

Les opérateurs logiques sont :

- AND
- OR
- NOT

Les opérateurs sur chaînes de caractères sont :

- effacement des espaces en début et fin de chaîne SKIP
- effacement des espaces en début de chaîne LSKIP
- effacement des espaces en fin de chaîne RSKIP
- longueur d'une chaîne LENGTH

Enfin, les opérateurs de type sont :

- conversion en entier possible ? ISINT
- conversion en numérique possible ? ISNUM
- conversion en chaîne possible ? ISSTRING
- conversion en date possible ? ISDATE
- conversion en heure possible ? ISTIME
- conversion en type entier INT
- conversion en type numérique NUM
- conversion en type chaîne Pascal STRING
- conversion en type date DATE
- conversion en type heure TIME
- conversion d'un caractère en numérique ASC
- conversion d'un entier en caractère CHR

Tous ces opérateurs correspondent aux opérateurs NCL. Se reporter au Manuel de Programmation NCL pour plus d'informations.

Coordonnées de cellules

Si une formule contient la référence d'une ou de plusieurs cellules sous la forme RnCm, la valeur prise en compte pour l'établissement du résultat est la valeur contenue dans la(les) cellule(s) référencée(s).

Fonctions

Exemple de fonctions :

- Somme @SUM(...)
- Moyenne @AVG(...) @AVG1(...)
- Maximum @MAX(...)

Les fonctions numériques sont précédées d'un caractère 'a commercial' (@), elles reçoivent en paramètres (entre parenthèses) des valeurs numériques, des références de cellules ou des coordonnées de blocs. Les fonctions s'appliquent au contenu des cellules ainsi définies.

Exemple :

En affectant à la cellule R3C3 la formule

```
= @SUM(R1C1:R2C2)
```

le résultat affiché dans R3C3 sera la somme du contenu des cellules R1C1, R1C2, R2C1 et R2C2.

La liste complète, classée par thèmes, des fonctions numériques de NS Calc est donnée et commentée dans le paragraphe suivant.

Si une formule référence une cellule vide, celle-ci est considérée comme contenant la valeur numérique 0. Ainsi, la cellule affichant le résultat indiquera "!DIV/0" si l'on tente de diviser par le contenu d'une telle cellule.

Si une cellule référencée dans une formule ne contient pas de valeur, le résultat affiché dans la cellule à laquelle la formule est affectée contiendra la chaîne "!VALUE".

Si la formule affectée à une cellule comporte la référence de la cellule elle-même, le résultat affiché dans cette cellule sera la chaîne "!REF"
Exemple : il est interdit de placer la formule : =R1C1+R2C2 dans la cellule R2C2 puisque la cellule R2C2 est déjà référencée dans la formule.

Fonctions classées par thème

Dans ce qui suit, les paramètres Val1, ..., ValN représentent soit :

- un nombre 100
- une formule 3*4
- une fonction @MIN(R2C2:R4C4)
- une cellule R3C5
- un bloc R2C2:R4C4

Exemple pour @SUM(Val1;Val2;...;ValN) :

```
@SUM(10; (3*4) ; R2C5; @MAX(R4C3:R4C8) )
```

Par contre, les paramètres Valeur, Valeur1, Valeur2 représentent soit :

- un nombre 5,5
- une formule 3*4
- une fonction @MIN(R2C2:R4C4)
- une cellule R3C5

mais ne peuvent être un bloc.

Exemple pour @SIN(Valeur) :

```
@SIN(R3C7)
```

Fonctions mathématiques

Fonction @EXP

Retourne l'exponentielle du nombre passé en paramètre.

Syntaxe	@EXP (Valeur)
----------------	----------------------

Exemple :

```
=@EXP(R3C2) ; affichera 1 si le contenu de R3C2 vaut 0
```

Fonction @FRAC

Retourne la partie décimale du nombre (réel) passé en paramètre.

Syntaxe	@FRAC (Valeur)
----------------	-----------------------

Exemple :

=@FRAC(R1C3) ; affichera 0.54 si le contenu de R1C3 vaut 3.54

Fonction @LN

Retourne le logarithme népérien du nombre (strictement positif) passé en paramètre.

Syntaxe	@LN (<i>Valeur</i>)
----------------	------------------------------

Exemple :

<code>=@LN(R3C4)</code> ; affichera 0 si le contenu de R3C4 vaut 1
--

Fonction @POWER

Retourne le résultat de Valeur1 puissance Valeur2 (Valeur1Valeur2).

Syntaxe	@POWER (<i>Valeur1</i> ; <i>Valeur2</i>)
----------------	---

Exemple :

```
=@POWER(R5C2;R3C4) ; affichera 32 si R5C2 vaut 2 et R3C4 vaut 5
```

Fonction @RANDOM

Retourne un nombre réel aléatoire entre 0 (inclusif) et 1 (exclusif).

Syntaxe	@RANDOM
----------------	----------------

Fonction @ROUND

Retourne l'entier arrondi le plus proche du nombre (réel) passé en paramètre.

Syntaxe	@ROUND (Valeur)
----------------	------------------------

Exemple :

```
=@ROUND(R3C4) ; affichera 2 si le contenu de R3C4 vaut 1.7
```

Fonction @SQR

Retourne le carré du nombre passé en paramètre.

Syntaxe	@SQR (Valeur)
----------------	----------------------

Ne pas confondre avec @SQRT.

Exemple :

```
=@SQR(R9C2) ; affichera 4 si le contenu de R9C2 vaut 2
```


Fonction @SQRT

Retourne la racine carrée du nombre (positif) passé en paramètre.

Syntaxe	@SQRT (Valeur)
----------------	-----------------------

Ne pas confondre avec @SQR.

Exemple :

<code>=@SQRT(R1C3)</code> ; affichera 3 si le contenu de R1C3 vaut 9
--

Fonction @SUM

Retourne la somme des valeurs numériques passées en paramètres.

Syntaxe	@SUM (<i>Val1</i> ; <i>Val2</i> ; ...; <i>ValN</i>)
----------------	--

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@SUM(R1C1:R2C2) ; affichera 60 (=10+20+30)
```

Fonction @TRUNC

Retourne la partie entière du nombre (réel) passé en paramètre.

Syntaxe	@TRUNC (<i>Valeur</i>)
----------------	---------------------------------

La chaîne de caractère @TRUNC ajoutée à un format (par exemple @TRUNC 0.00) permet d'afficher une valeur numérique tronquée tout en conservant la valeur d'origine en mémoire.

Exemple :

```
=@TRUNC(R5C2) ; affichera 2 si le contenu de R5C2 vaut 2.8
```

Fonctions trigonométriques

Fonction @ATAN

Retourne l'arc tangente (en radians) du nombre passé en paramètre.

Syntaxe	@ATAN (<i>Valeur</i>)
----------------	--------------------------------

Exemple :

```
=@ATAN(R4C3) ; affichera 0.78539... (Pi/4)  
; si le contenu de R4C3 vaut 1
```

Fonction @COS

Retourne le cosinus du nombre, en radians, passé en paramètre.

Syntaxe	@COS (<i>Valeur</i>)
----------------	-------------------------------

Exemple :

=@COS(R2C4) ; affichera 0.54030... ; si le contenu de R2C4 vaut 1

Fonction @PI

Retourne la valeur de Pi, soit 3.141592653...

Syntaxe	@PI
---------	-----

Fonction @SIN

Retourne le sinus du nombre, en radians, passé en paramètre.

Syntaxe	@SIN (<i>Valeur</i>)
----------------	-------------------------------

Exemple :

=@SIN(R2C4) ; affichera 0.84147... ; si le contenu de R2C4 vaut 1

Fonctions statistiques

Fonction @AVG

Retourne la moyenne arithmétique (AVerage) des valeurs numériques passées en paramètre.

Syntaxe	@AVG (<i>Val1</i> ; <i>Val2</i> ; ...; <i>ValN</i>)
----------------	--

Dans le cas de définition de blocs, toutes les cellules définies dans un bloc sont prises en compte contrairement à la fonction @AVG1.

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@AVG(R1C1:R2C2) ; affichera 15 ( (10+20+30)/4 )
```


Fonction @AVG1

Retourne la moyenne arithmétique (AVerage) des valeurs numériques passées en paramètre.

Syntaxe	@AVG1 (<i>Val1</i> ; <i>Val2</i> ; ...; <i>ValN</i>)
----------------	---

Dans le cas de définition de blocs, seules les cellules contenant des valeurs numériques sont prises en compte contrairement à la fonction @AVG qui effectue la moyenne sur toutes les cellules.

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@AVG1(R1C1:R2C2) ; affichera 20 ( (10+20+30)/3 )
```

Fonction @COUNT

Retourne le nombre de cellules contenues dans les blocs passés en paramètre quel que soit leur contenu.

Syntaxe	@COUNT (<i>Bloc1</i> ; <i>Bloc2</i> ; ...; <i>BlocN</i>)
----------------	---

Toutes les cellules des blocs sont prises en compte.

Exemple :

```
=@COUNT(R10C10:R20C20) ; affichera 121 (=11*11)
```

Fonction @COUNT1

Retourne le nombre de cellules des blocs passés en paramètre contenant des valeurs numériques.

Syntaxe	@COUNT1 (<i>Bloc1; Bloc2; ...; BlocN</i>)
----------------	--

Seules les cellules contenant des valeurs numériques sont comptabilisées contrairement à la fonction @COUNT qui dénombre toutes les cellules.

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@COUNT1(R1C1:R2C2) ; affichera 3
```

Fonction @MAX

Retourne la valeur numérique maximum trouvée dans la liste des paramètres.

Syntaxe	@MAX (<i>Val1</i> ; <i>Val2</i> ; ...; <i>ValN</i>)
----------------	--

Seules les cellules contenant des valeurs numériques sont prises en compte.

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@MAX(R1C1:R2C2) ; affichera 30
```

Fonction @MIN

Retourne la valeur numérique minimum trouvée dans la liste des paramètres.

Syntaxe	@MIN (Val1; Val2; ...; ValN)
----------------	-------------------------------------

Seules les cellules contenant des valeurs numériques sont prises en compte.

Exemple :

```
; Soient les cellules :  
; R1C1 = 10  
; R1C2 = 20  
; R2C1 = "TOTAL"  
; R2C2 = 30  
=@MIN(R1C1:R1C2) ; affichera 10
```

Fonctions logiques

Fonction @IF

Teste la condition Condition. Si celle ci est vraie, IfExpr est exécuté, sinon ElseExpr est exécuté.

Syntaxe	@IF (<i>Condition; IfExpr; ElseExpr</i>)
----------------	---

Exemple :

```
=@IF(R3C4; (10/R3C4); (10*2)) ; si R3C4 est différent de 0, affichera  
10/R3C4, sinon affichera 20
```

Fonctions date & heure

Fonction @CURRENTDATE

Retourne la date courante, en nombre de jours écoulés depuis le 1er janvier 1900.

Syntaxe	@CURRENTDATE
----------------	---------------------

Exemple :

```
=@STRING (@CURRENTDATE; "d/m/yy")
```

Fonction @CURRENTDAY

Retourne le jour courant.

Syntaxe	@CURRENTDAY
----------------	--------------------

Exemple :

```
=@STRING (@CURRENTDAY; "dd")
```


Fonction @CURRENTTIME

Retourne l'heure courante, en nombre de secondes écoulées depuis minuit (0h0mn0s).

Syntaxe**@CURRENTTIME**

Exemple :

```
=@STRING (@CURRENTTIME; "hh:mm")
```

Fonctions sur les chaînes

Fonction @COPY

Retourne une chaîne extraite de la chaîne ChaîneSource à partir du caractère Position et de longueur spécifiée par Longueur.

Syntaxe	@COPY (<i>ChaîneSource</i> ; <i>Position</i> ; <i>Longueur</i>)
----------------	--

Exemple :

```
=@COPY(R2C4; 1; 3) ; Affichera "TOT" si R2C4 contient "TOTAL"
```

Fonction @DELETE

Retourne la chaîne ChaîneSource dans laquelle ont été supprimés, pour la longueur précisée par Longueur, les caractères à partir du caractère Position.

Syntaxe	@DELETE (ChaîneSource; Position; Longueur)
----------------	--

Exemple :

```
=@DELETE(R2C4; 1; 3) ; Affichera "AL" si R2C4 contient "TOTAL"
```

Fonction @INSERT

Retourne la chaîne ChaîneDest dans laquelle a été insérée la chaîne ChaîneSource à la position spécifiée par Position.

Syntaxe	@INSERT (<i>ChaîneSource</i> ; <i>ChaîneDest</i> ; <i>Position</i>)
----------------	--

Exemple :

```
=@INSERT(R2C4; R3C4; 4) ; Affichera "TOTXXXAL" si R2C4 contient "XXX" et si R3C4 contient "TOTAL"
```

Fonction @POS

Retourne la position du premier caractère de ChaîneCherchée dans la chaîne ChaîneSource. Retourne 0 si la chaîne n'est pas trouvée.

Syntaxe	@POS (<i>ChaîneCherchée</i> ; <i>ChaîneSource</i>)
----------------	---

La recherche est 'case sensitive', autrement dit, la différence minuscules/majuscules est prise en compte.

Exemple :

```
=@POS("A"; R2C4) ; Affichera 4 si R2C4 contient "TOTAL"
```

Fonctions de formatage

Fonction @STRING

Retourne une chaîne correspondant au nombre Valeur passé en premier paramètre selon le format spécifié dans ChaîneFormat.

Syntaxe	@STRING (<i>Valeur; ChaîneFormat</i>)
----------------	--

Les formats acceptés sont tous les formats acceptés par NS Calc, soit les formats EXCEL (décrits dans le Manuel d'Utilisation de NS Design).

Exemple :

```
=@STRING(R3C6; "# ##0.00") ; Affichera 1 234,57 si R3C6 contient 1234.567
```

Fonctions sur événement

Fonction @SEND

Envoie à la fenêtre NS Calc l'événement USER précisé par Num, avec les paramètres Param1 et Param2.

Syntaxe	@SEND (<i>Num</i> ; <i>Param1</i> ; <i>Param2</i>)
----------------	---

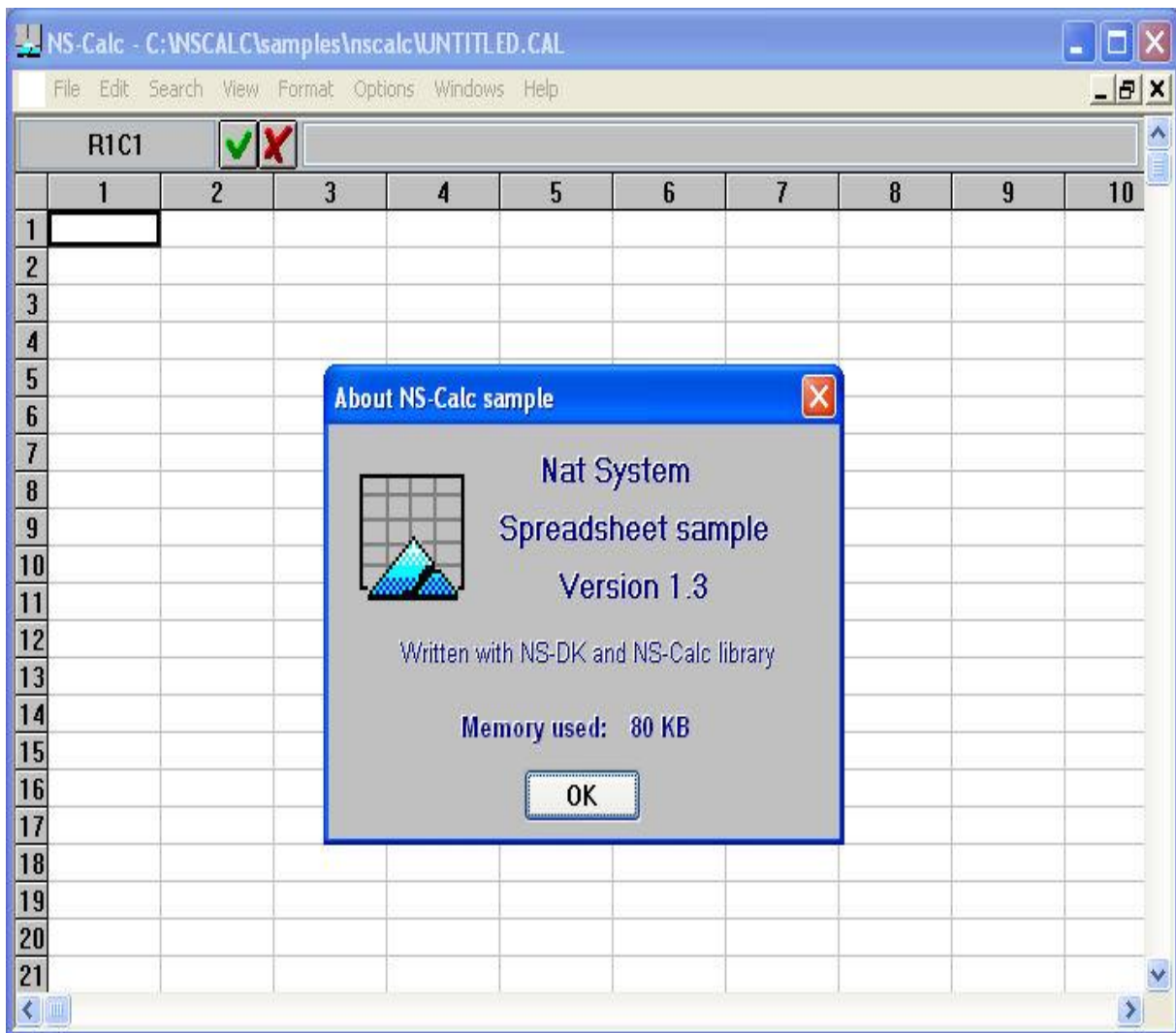
Num est un entier allant de 0 (USER0) à 15 (USER15).

L'EXEMPLE NS-CALC

Introduction

Cet exemple, nommé NS Calc, est livré sur la disquette du produit NS Calc. Il permet de montrer les différentes possibilités de NS Calc, et la façon de programmer avec cette librairie.

Les sources complets de NS Calc sont fournis, ainsi que l'application générée obtenue : NSCALC.EXE.



Composantes du projet NS-Calc

Fenêtres

Ce projet comporte une fenêtre de classe MDI Window, une fenêtre NS Calc (classe Window) et un ensemble de fenêtres de classe Dialog permettant, entre autres, de modifier les options, format ou affichage des cellules et des feuilles de calcul :

- NS_CALC1

Fenêtre (classe MDI Window), permettant d'ouvrir une fenêtre NS_CALC2 par feuille de calcul, grâce aux menus File.New ou File.Open.

- NS_CALC2

Fenêtre NS Calc (classe Window), ayant {NS02CALC} dans son titre et ayant son menu Windows nommé MDIWINDOW. Il s'agit de la seule fenêtre NS Calc de l'exemple. Depuis cette fenêtre sont ouvertes, lors de la sélection de ses menus, la majeure partie des autres fenêtres du projet, y compris de nouvelles instanciations d'elle même lors d'un File.New ou File.Open.

- NS_CALC3

Fenêtre (classe Dialog) d'à propos, ouverte depuis le menu Help.About de NS_CALC1 et NS_CALC2.

- NS_CALC4

Fenêtre (classe Dialog) de suppression de cellules, ouverte depuis le menu Edit.Delete de NS_CALC2.

- NS_CALC5

Fenêtre (classe Dialog) d'insertion de cellules, ouverte depuis le menu Edit.Insert de NS_CALC2.

- NS_CALC6

Fenêtre (classe Dialog) d'accès à une cellule particulière, ouverte depuis le menu Search.Goto de NS_CALC2.

- NS_CALC7

Fenêtre (classe Dialog) de sélection des options d'affichage, ouverte depuis le menu View.Display de NS_CALC2.

- NS_CALC8

Fenêtre (classe Dialog) de sélection des critères de tri, ouverte depuis le menu View.Sort de NS_CALC2.

- NS_CALC9

Fenêtre (classe Dialog) de définition de la hauteur des lignes, ouverte depuis le menu Format.Row height de NS_CALC2.

- NS_CALC10

Fenêtre (classe Dialog) de définition de la largeur des colonnes, ouverte depuis le menu Format.Column width de NS_CALC2.

- NS_CALC11

Fenêtre (classe Dialog) de définition du format d'affichage dans les cellules, ouverte depuis le menu Format.Number de NS_CALC2.

- NS_CALC12

Fenêtre (classe Dialog) de définition de nom de cellules, ouverte depuis le menu Search.Names de NS_CALC2.

- NS_CAL13

Fenêtre (classe Dialog) de définition du cadre des cellules, ouverte depuis le menu Format.Border de NS_CALC2.

- NS_CAL14

Fenêtre (classe Dialog) de définition des couleurs des cellules, ouverte depuis le menu Format.Color de NS_CALC2.

- NS_CAL15

Fenêtre (classe Dialog) de définition de l'alignement horizontal et vertical des cellules, ouverte depuis le menu Format.Alignment de NS_CALC2.

- NS_CAL16

Fenêtre (classe Dialog) de définition des caractéristiques de l'impression, ouverte depuis le menu File.Print de NS_CALC2.

- NS_CAL17

Fenêtre (classe Dialog) de sélection de la fonte utilisée dans les cellules, ouverte depuis le menu Format.Font de NS_CALC2.

- NS_CAL18

Fenêtre (classe Dialog) de définition des couleurs par défaut, ouverte depuis le menu Options.Default colors de NS_CALC2.

- NS_CAL19

Fenêtre (classe Dialog) de sélection des options à appliquer pour un collage spécial, ouverte depuis le menu Edit.Paste special de NS_CALC2.

- NS_CAL20

Fenêtre (classe Dialog) de sélection de l'option à appliquer pour une suppression, ouverte depuis le menu Edit.Clear special de NS_CALC2.

- NS_CAL21

Fenêtre (classe Dialog) de sélection des options pour l'insertion d'un lien 'avancé', ouverte depuis le menu View.Add enhanced link de NS_CALC2.

- NS_CAL22

Fenêtre (classe Dialog) de définition de l'ordre de chaînage des cellules, ouverte depuis le menu Options.Set threads de NS_CALC2.

- NS_CAL23

Fenêtre (classe Dialog) de spécification des cellules à bloquer, ouverte depuis le menu Options.Lock rows/columns de NS_CALC2.

- NS_CAL24

Fenêtre (classe Dialog) de spécification du "Presentation Space", ouverte depuis le menu View.PS de NS_CALC2.

- NS_CAL25

Fenêtre (classe Window) de définition du "Presentation Space", ouverte après validation de la fenêtre NS_CAL24.

- NS_CAL26

Fenêtre (classe Dialog) de sélection de la liste à charger dans la combo box de la ligne de statut, ouverte depuis le menu Format.Edit control de NS_CALC2.

- NS_CAL27

Fenêtre (classe Dialog) de sélection de l'imprimante, ouverte depuis le menu File.Printer setup de NS_CALC1 et NS_CALC2.

- NS_CAL28

Fenêtre (classe Dialog) de définition du format des cellules (numérique, texte, ...), ouverte depuis le menu Format.Cell de NS_CALC2.

- NS_CAL29

Fenêtre (classe Dialog) de sélection du format du fichier de sauvegarde et/ou de son nom, ouverte depuis le menu File.Save as de NS_CALC2.

Librairies

Cinq librairies font partie du projet NS Calc.

- NSCALC

Librairie nécessaire pour toute application NS Calc !

- NSCALCL

Librairie spécifique à l'exemple, comportant l'initialisation des variables globales ainsi que la définition d'instructions et fonctions telles que ERROR, INTERNAL, FIND_NAME%, NEW_NAME%, etc.

- NSGRAPH

Constantes GFS_*% utilisées pour les attributs de la fonte utilisée pour l'impression (fenêtre Print, NS_CAL16) ou sélectionnée dans la fenêtre Font Selection (NS_CAL17).

- NSMISC

Fonction STRING\$ utilisée lors de l'initialisation de la fenêtre NS_CALC3 et fonction FEXPAND\$ utilisée par la librairie NSCALCL.

- NSPATH

Fonction GETPATHNAME\$ utilisée par le menu File.Open des fenêtres NS_CALC1 et NS_CALC2, ainsi que par la boîte NS_CAL29.

- NSPRN

Instructions REP_* utilisées lors de l'initialisation et de la fermeture de la fenêtre NS_CALC1, ainsi que la constante PU_LOENGLISH%.

Segments et variables

Deux segments et deux ressources variables font partie du projet NS Calc.

- Segment NS_CALC1

alloué pour chaque fenêtre NS_CALC2 ouverte (voir la fonction NEW_NAME% de la librairie NSCALCL, appelée lors d'un File.New ou File.Open de NS_CALC1 et NS_CALC2) et dont le handle est mémorisé grâce à un SETDATA.

Segment NS_CALC2

regroupe toutes les informations manipulées par l'instruction CALC_DRAW.

- Segment EDITINFO

regroupe toutes les informations manipulées par la boîte NS_CALC17 et l'instruction CALC_SETBLOCKFONT et CALC_GETCELLFONT.

Ressources Variables NS_CALC1 et NS_CALC2

regroupent les déclarations des variables globales utilisées par le projet NS Calc.

Icônes et bitmaps

Deux icônes et plusieurs bitmaps font partie du projet NS Calc :

- Icône NS_CALCI1 et NS_CALCI2

utilisées respectivement comme icône de la fenêtre NS_CALC1 et de la fenêtre NS_CALC2.

- Bitmap NS_CALCB

affichée au sein de la fenêtre NS_CALC3.

- Bitmap FONTSEL

affichée au sein de la fenêtre NS_CAL17.

- Bitmaps *R et *P (AlternR et AlternP, SolidR et SolidP, etc.)

utilisées dans la fenêtre NS_CAL13 pour représenter les différents styles de bordure possibles.

Autres ressources

Il n'y a ni template, ni constante, ni pointeur dans le projet NS Calc.

Néanmoins, il reste à décrire les fichiers texte (.TXT) du projet : COLORS.TXT, DAYS.TXT, MONTHS.TXT et YESNO.TXT. Chacun de ces fichiers est une liste de choix proposés dans la ligne de statut, après sélection de la liste dans la fenêtre NS_CAL26.

Utilisation de la librairie NS Calc au sein de l'exemple

Voici la liste des mots clés de la librairie NS Calc utilisés dans l'exemple, avec l'emplacement de leur utilisation.

Mot clé	Ress.	Objet	Evénement
BORDERSTYLE	NS_CAL13	-	INIT
		OK	EXECUTED
CALC_ADDENHANCEDLINK	NS_CAL21	OK	EXECUTED
CALC_ADDLINK			
CALC_CLEAR	NS_CALC2	CLEAR1	EXECUTED
		DELETE1	EXECUTED
CALC_CLEARSPICAL	NS_CAL20	OK	EXECUTED
CALC_COLUMN%	NS_CAL10	-	INIT
	NS_CAL11	-	INIT
	NS_CAL12	-	INIT
	NS_CAL13	-	INIT
	NS_CAL14	-	INIT
	NS_CAL15	-	INIT
	NS_CAL16	-	INIT
	NS_CAL17	-	INIT
CALC_COPY	NS_CALC2	CUT1	EXECUTED
		COPY1	EXECUTED
CALC_CREATENAME	NS_CAL12	INS	EXECUTED
CALC_DELETENAME	NS_CAL12	DEL	EXECUTED
CALC_DISPLAYFORMULA	NS_CALC2	DISPLAYFORMULA	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_DISPLAYZERO	NS_CALC2	DISPLAYZERO	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_DRAW	NS_CAL25	-	PAINT
CALC_FILLDOWN	NS_CALC2	FILLD	EXECUTED
CALC_FILLRIGHT	NS_CALC2	FILLR	EXECUTED
CALC_FINDNAME\$			
CALC_FIRSTNAME%	NS_CAL12	NAMES	INIT
CALC_GET\$			

CALC_GETALLOCATEDMEM%	NS_CALC3	-	INIT
CALC_GETBACKCOLOR%	NS_CAL18	-	INIT
CALC_GETCELLBACKCOL%	NS_CAL14	-	INIT
CALC_GETCELLBORDER	NS_CAL13	-	INIT
CALC_GETCELLEDITCTRL			
CALC_GETCELLFILTER\$			
CALC_GETCELLFONT	NS_CAL17	-	INIT
CALC_GETCELLFORMAT\$	NS_CAL11	-	INIT
CALC_GETCELLGRID%			
CALC_GETCELLHORZJUSTIF%	NS_CAL15	-	INIT
CALC_GETCELLKIND%	NS_CAL28	-	INIT
CALC_GETCELLLENGTH%			
CALC_GETCELLTEXTCOL%	NS_CAL14	-	INIT
CALC_GETCELLVERTJUSTIF%	NS_CAL15	-	INIT
CALC_GETCELLWIDTH%			
CALC_GETDEFCOLOR%			
CALC_GETDEFHEIGHT%	NS_CALC9	HEIGHT	CHANGED
		DEFAULT	SELECTED
CALC_GETDEFWIDTH%	NS_CAL10	WIDTH	CHANGED
		DEFAULT	SELECTED
CALC_GETDRAWSIZE			
CALC_GETFILEFORMAT%	NS_CAL29	-	INIT
	NS_CALC2	SAVE1	EXECUTED
CALC_GETFORMAT			
CALC_GETHEIGHT%	NS_CALC9	-	INIT
CALC_GETLEFTCOLUMNS%	NS_CAL23	-	INIT
	NS_CALC7	-	INIT
	OK	EXECUTED	
CALC_GETNBCELLS%			
CALC_GETNBCOLUMNS%	NS_CAL16	-	INIT
	NS_CAL23	OK	EXECUTED
	NS_CALC2	DELETE1	EXECUTED
		INSERT1	EXECUTED
	NS_CALC6	OK	EXECUTED

	NS_CALC7	-	INIT
		OK	EXECUTED
CALC_GETNBROWS%	NS_CAL16	-	INIT
	NS_CAL23	OK	EXECUTED
	NS_CALC2	DELETE1	EXECUTED
		INSERT1	EXECUTED
	NS_CALC6	OK	EXECUTED
	NS_CALC7	-	INIT
		OK	EXECUTED
CALC_GETPRINTFILENBPAGES%			
CALC_GETPRINTNBPAGES%	NS_CAL16	-	INIT
		ZOOM	LOSEFOCUS
		OK	EXECUTED
CALC_GETTEXT\$	NS_CALC2	VALUE1	EXECUTED
CALC_GETTEXTCOLOR%	NS_CAL18	-	INIT
CALC_GETTHREAD\$	NS_CAL22	THREADS	INIT
CALC_GETTOPROWS%	NS_CAL23	-	INIT
	NS_CALC7	-	INIT
		OK	EXECUTED
CALC_GETVALUE\$	NS_CALC2	VALUE1	EXECUTED
CALC_GETWIDTH%	NS_CAL10	-	INIT
CALC_GOTO	NS_CAL12	GOTONAME	EXECUTED
	NS_CALC6	OK	EXECUTED
CALC_ISCELLEEMPTY%			
CALC_ISCELLLOCKED%			
CALC_ISCLIPBOARD%	NS_CALC2	EDIT1	INIT
CALC_ISFORMULADISPLAYED%	NS_CALC2	DISPLAYFORMULA	EXECUTED
		OPTIONS	INIT
		OPTIONS	SELECTED
	NS_CALC7	-	INIT
CALC_ISGRID%	NS_CALC2	MENUGRID	EXECUTED
		OPTIONS	SELECTED
	NS_CALC7	-	INIT
		OK	EXECUTED

CALC_ISHEADING%	NS_CALC2	MENUHEADING	EXECUTED
		OPTIONS	SELECTED
	NS_CALC7	-	INIT
		OK	EXECUTED
CALC_ISHEIGHTLOCKED%			
CALC_ISLINK%	NS_CAL21	REMOVE	EXECUTED
CALC_ISLOCKED%	NS_CALC2	EDIT1	INIT
CALC_ISSTATUSBAR%	NS_CALC2	STATUSBAR	EXECUTED
		OPTIONS	SELECTED
CALC_ISUNDO%	NS_CALC2	EDIT1	INIT
CALC_ISWIDTHLOCKED%			
CALC_ISZERODISPLAYED%	NS_CALC2	DISPLAYZERO	EXECUTED
		OPTIONS	INIT
	NS_CALC7	-	INIT
CALC_LOAD%	NS_CALC1	OPEN1	EXECUTED
CALC_LOCK	NS_CALC2	READONLY1	EXECUTED
CALC_LOCKBLOCK			
CALC_LOCKHEIGHT			
CALC_LOCKWIDTH			
CALC_MAX_COLUMNS%			
CALC_MAX_ROWS%			
CALC_MULTI_SELECT			
CALC_MULTI_SELECTION\$	NS_CAL11	OK	EXECUTED
	NS_CAL13	OK	EXECUTED
	NS_CAL14	OK	EXECUTED
	NS_CAL15	OK	EXECUTED
	NS_CAL17	OK	EXECUTED
	NS_CAL20	OK	EXECUTED
	NS_CAL28	OK	EXECUTED
CALC_NEXTNAME%	NS_CAL12	NAMES	INIT
CALC_PASTE	NS_CALC2	PASTE1	EXECUTED
CALC_PASTESPECIAL	NS_CAL19	OK	EXECUTED
CALC_PREPARE_PUT			
CALC_PRINT%	NS_CAL16	OK	EXECUTED

CALC_PRINTFILE%			
CALC_PUT			
CALC_RC2XY	NS_CAL12	GOTONAME	EXECUTED
	NS_CAL21	-	INIT
	NS_CALC6	OK	EXECUTED
	NSCALCL	Instruction BLOCK2XYXY	
CALC_REMLINK	NS_CAL21	REMOVE	EXECUTED
CALC_ROW%	NS_CAL11	-	INIT
	NS_CAL12	-	INIT
	NS_CAL13	-	INIT
	NS_CAL14		INIT
	NS_CAL15		INIT
	NS_CAL17		INIT
	NS_CALC9		INIT
CALC_SAVE%	NS_CAL29	OK	EXECUTED
	NS_CALC2	SAVE1	EXECUTED
CALC_SCROLL_DOWN	NS_CALC2	INSERT1	EXECUTED
CALC_SCROLL_LEFT	NS_CALC2	DELETE1	EXECUTED
CALC_SCROLL_RIGHT	NS_CALC2	INSERT1	EXECUTED
CALC_SCROLL_UP	NS_CALC2	DELETE1	EXECUTED
CALC_SELECT			
CALC_SELECTION\$	NS_CAL10	-	INIT
	NS_CAL11	OK	EXECUTED
	NS_CAL13	OK	EXECUTED
	NS_CAL14	OK	EXECUTED
	NS_CAL15	OK	EXECUTED
	NS_CAL16	-	INIT
	NS_CAL17	OK	EXECUTED
	NS_CAL20	OK	EXECUTED
	NS_CAL21	-	INIT
	NS_CAL24	-	INIT
	NS_CAL26	OK	EXECUTED
	NS_CALC2	CLEAR1	EXECUTED

		DELETEISNERT11	EXECUTED
		FILLR	EXECUTED
		FILLD	EXECUTED
		VALUE1	EXECUTED
		CELL1	EXECUTED
	NS_CALC4	-	INIT
	NS_CALC5	-	INIT
	NS_CALC8	-	INIT
	NS_CALC9	-	INIT
CALC_SETBACKCOLOR	NS_CAL18	OK	EXECUTED
CALC_SETBLOCKBACKCOL	NS_CAL14	OK	EXECUTED
CALC_SETBLOCKBORDER	NS_CAL13	OK	EXECUTED
CALC_SETBLOCKEDITCTRL	NS_CAL26	OK	EXECUTED
CALC_SETBLOCKFILTER			
CALC_SETBLOCKFONT	NS_CAL17	OK	EXECUTED
CALC_SETBLOCKFORMAT	NS_CAL11	OK	EXECUTED
CALC_SETBLOCKGRID			
CALC_SETBLOCKHORZJUSTIF	NS_CALC15	OK	EXECUTED
CALC_SETBLOCKKIND	NS_CAL28	OK	EXECUTED
CALC_SETBLOCKLENGTH			
CALC_SETBLOCKTEXTCOL	NS_CAL14	OK	EXECUTED
CALC_SETBLOCKVERTJUSTIF	NS_CAL15	OK	EXECUTED
CALC_SETDEFCOLOR			
CALC_SETFORMAT			
CALC_SETGRID	NS_CALC2	MENUGRID	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_SETHEADING	NS_CALC2	MENUHEADING	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_SETHEIGHT	NS_CALC9	OK	EXECUTED
CALC_SETLEFTCOLUMNS	NS_CAL23	OK	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_SETNBCELLS			
CALC_SETNBCOLUMNS	NS_CALC7	OK	EXECUTED
CALC_SETNBBROWS	NS_CALC7	OK	EXECUTED

CALC_SETSTATUSBAR	NS_CALC2	-	INIT
		STATUSBAR	EXECUTED
CALC_SETTEXTCOLOR	NS_CAL18	OK	EXECUTED
CALC_SETTHREAD	NS_CAL22	OK	EXECUTED
CALC_SETTOPROWS	NS_CAL23	OK	EXECUTED
	NS_CALC7	OK	EXECUTED
CALC_SETVARCELLWIDTHS			
CALC_SETWIDTH	NS_CAL10	OK	EXECUTED
CALC_SORT	NS_CALC8	OK	EXECUTED
CALC_UNDO	NS_CAL2	UNDO1	EXECUTED
CALC_UPDATE	NS_CALC2	CALCULATENOW	EXECUTED
	NS_CALC7	OK	EXECUTED
CK_DEFAULT%			
CK_FORMULA%			
CK_NUMBER%			
CK_TEXT%			
CLP_CORNER%			
CLP_HORIZONTAL%			
CLP_VERTICAL%			
CLS_BROKEN%			
CLS_STRAIGHT%			
CTRLDATA_ARRAY%			
CTRLDATA_FILE%	NS_CAL26	OK	EXECUTED
CTRLDATA_NONE%	NS_CAL26	OK	EXECUTED
EDTCTRL_CBE%	NS_CAL26	OK	EXECUTED
EDTCTRL_EF%	NS_CAL26	OK	EXECUTED
FF_NSCALC%	NS_CAL29	-	INIT
		OK	EXECUTED
	NS_CALC2	SAVE1	EXECUTED

FF_NSCALC_ANSI%	NS_CAL29	OK	EXECUTED
FF_NSCALC_MAC%	NS_CAL29	OK	EXECUTED
FF_NSCALC_PC%	NS_CAL29	OK	EXECUTED
FF_TEXT%	NS_CAL29	OK	EXECUTED
FF_TEXT_ANSI%			
FF_TEXT_MAC%			
FF_TEXT_PC%			
GK_ALTERNATE%			
GK_BOTTOM%			
GK_LEFT%			
GK_RIGHT%			
GK_TOP%			
HJ_CENTERED%			
HJ_DEFAULT%			
HJ_LEFT%			
HJ_RIGHT%			
JF_CENTERED%			
JF_DEFAULT%			
JF_LEFT%			
JF_RIGHT%			
LINKSTYLE	NS_CAL21	OK	EXECUTED
PK_BOTTOM%	NS_CAL16	-	INIT
		ZOOM	LOSEFOCUS
		OK	EXECUTED
PK_GRID%	NS_CAL16	OK	EXECUTED
PK_HCENTER%	NS_CAL16	OK	EXECUTED
PK_LEFT%	NS_CAL16	-	INIT
		ZOOM	LOSEFOCUS
		OK	EXECUTED

PK_PAGENUMBER%	NS_CAL16	OK	EXECUTED
PK_RIGHT%	NS_CAL16	OK	EXECUTED
PK_TITLE%	NS_CAL16	-	INIT
		ZOOM	LOSEFOCUS
		OK	EXECUTED
PK_TOP%	NS_CAL16	OK	EXECUTED
PK_VCENTER%	NS_CAL16	OK	EXECUTED
PT_ADD%	NS_CAL19	OK	EXECUTED
PT_ALL%	NS_CAL19	OK	EXECUTED
	NS_CAL20	OK	EXECUTED
PT_DIV%	NS_CAL19		
PT_FORMAT%	NS_CAL19		
	NS_CAL20		
PT_FORMULA%	NS_CAL19		
	NS_CAL20		
PT_MULT%	NS_CAL19		
PT_SUB%	NS_CAL19		
PT_VALUE%	NS_CAL19		
	NS_CAL20		
VJ_BOTTOM%			
VJ_CENTERED%			
VJ_DEFAULT%			
VJ_TOP%			
VJ_WORDWRAP%			

THEMES DE L'API

Affichage de la fenêtre NS-Calc

Fonctions	Description
CALC_ISGRID%	indique si la grille est affichée.
CALC_ISHEADING%	indique si les numéros de lignes et colonnes sont affichés.
CALC_ISSTATUSBAR%	indique si la ligne de statut est affichée.

Instructions	Description
CALC_SETGRID	affiche ou cache la grille du tableur.
CALC_SETHEADING	affiche ou cache les numéros de lignes et colonnes.
CALC_SETSTATUSBAR	affiche ou cache la ligne de statut.

Dimension de la feuille de calcul

Fonctions	Description
CALC_GETNBCOLUMNS%	retournent respectivement le nombre de colonnes et le nombre de lignes définis.
CALC_GETNBROWS%	

Instructions	Description
CALC_SETNBCOLUMNS	permettent de fixer le nombre de colonnes et de lignes
CALC_SETNBROWS	

Constantes	Description
CALC_MAX_COLUMNS%	indiquent le nombre maximum de colonnes et de lignes.
CALC_MAX_ROWS%	

Feuille à cellules variables

Fonctions	Description
CALC_GETNBCELLS%	retourne le nombre de colonnes fixé pour une ligne.
CALC_SETVARCELLWIDTHS	autorise ou empêche un nombre variable de cellules ainsi qu'une largeur variable pour ces cellules.

Instructions	Description
CALC_SETNBCELLS	permet de fixer le nombre de colonnes pour une ligne donnée.

Dimension des cellules

Fonctions	Description
CALC_GETCELLWIDTH%	retourne la largeur d'une cellule particulière.
CALC_GETDEFHEIGHT%	retourne la hauteur par défaut des cellules.
CALC_GETDEFWIDTH%	retourne la largeur par défaut des cellules.
CALC_ISHEIGHTLOCKED%	indique si la hauteur des lignes est bloquée (non modifiable avec la souris).
CALC_ISWIDTHLOCKED%	indique si la largeur des colonnes est bloquée.

Instructions	Description
CALC_LOCKHEIGHT	bloque la hauteur des cellules.
CALC_LOCKWIDTH	bloque la largeur des cellules.
CALC_SETHEIGHT	permet de fixer la hauteur des lignes.
CALC_SETWIDTH	permet de fixer la largeur des colonnes.

Blocage des cellules

Fonctions	Description
CALC_GETLEFTCOLUMNS%	retourne le nombre de colonnes bloquées.
CALC_GETTOPROWS%	retourne le nombre de lignes bloquées.
CALC_ISCELLLOCKED%	indique si une cellule particulière est bloquée en saisie.
CALC_ISLOCKED%	indique si toutes les cellules de la feuille sont bloquées en saisie.

Instructions	Description
CALC_LOCK	modifie le blocage de CALC_ISLOCKED%.
CALC_LOCKBLOCK	modifie l'état de blocage d'un bloc de cellules.
CALC_SETLEFTCOLUMNS	permet de bloquer le défilement des colonnes de gauche.
CALC_SETTOPROWS	permet de bloquer le défilement des lignes du haut.

Cellule active, sélections, coordonnées

Fonctions	Description
CALC_COLUMN%	retourne le numéro de colonne de la cellule active.
CALC_MULTI_SELECTION\$	retourne les coordonnées d'un des blocs sélectionnés.
CALC_ROW%	retourne le numéro de ligne de la cellule active.
CALC_SELECTION\$	retourne les coordonnées du bloc sélectionné.

Instructions	Description
CALC_MULTI_SELECT	ajoute un nouveau bloc à la sélection en cours.
CALC_SELECT	sélectionne un bloc.
CALC_GOTO	permet d'activer une cellule (déplacer la cellule active).
CALC_RC2XY	retourne le numéro de colonne et le numéro de ligne, à partir des coordonnées RnCm d'une cellule.

Noms logiques

Fonctions	Description
CALC_FINDNAME\$	retourne les coordonnées de la cellule associée à un nom logique.
CALC_FIRSTNAME%	retourne les coordonnées et le nom logique de la première cellule associée à un nom.
CALC_NEXTNAME%	retourne les coordonnées et le nom logique des cellules suivantes.

Instructions	Description
CALC_CREATENAME	associe un nom logique à une cellule.
CALC_DELETENAME	supprime un nom logique.

Contenu des cellules

Fonctions	Description
CALC_GET\$	retourne le contenu complet des cellules d'un bloc.
CALC_GETTEXT\$	retourne les valeurs stockées des cellules d'un bloc.
CALC_GETVALUE\$	retourne les valeurs affichées des cellules d'un bloc.
CALC_ISCELLEEMPTY%	indique si une cellule est vide ou non.

Format des cellules et de la feuille de calcul

Fonctions	Description
CALC_GETBACKCOLOR%	retourne la couleur de fond des cellules.
CALC_GETCELLBACKCOL%	Les fonctions CALC_GETCELL*% retournent chacune une partie différente du format associé à une cellule particulière (couleurs, fonte, justification, type, cadre...).
CALC_GETCELLFILTER\$	
CALC_GETCELLFORMAT\$	
CALC_GETCELLGRID%	
CALC_GETCELLHORZJUSTIF%	
CALC_GETCELLKIND%	
CALC_GETCELLLENGTH%	
CALC_GETCELLTEXTCOL%	
CALC_GETCELLVERTJUSTIF%	
CALC_GETDEFCOLOR%	retourne la couleur du fond de la feuille de calcul.
CALC_GETTEXTCOLOR%	retourne la couleur des caractères des cellules.
CALC_ISFORMULADISPLAYED%	retourne 1 si la cellule contient une formule
CALC_ISZERODISPLAYED%	retourne 1 si la cellule est vide.
CALC_SETBLOCKHORZJUSTIF	les fonctions CALC_SETBLOCK* modifient distinctement les informations spécifiant ce format et ce pour un bloc de cellules.
CALC_SETBLOCKKIND	
CALC_SETBLOCKLENGTH	
CALC_SETBLOCKTEXTCOL	
CALC_SETBLOCKVERTJUSTIF	
CALC_SETDEFCOLOR	modifie la couleur du fond de la feuille de calcul.
CALC_SETFORMAT	modifie le format complet pour plusieurs cellules.
CALC_SETTEXTCOLOR	modifie la couleur des caractères des cellules.

Instructions	Description
CALC_DISPLAYFORMULA	

CALC_DISPLAYZERO	
CALC_GETCELLBORDER	les instructions CALC_GETCELL* retournent chacune une partie différente du format associé à une cellule particulière (couleurs, fonte, justification, type, cadre...).
CALC_GETCELLEDITCTRL	
CALC_GETCELLFONT	
CALC_GETFORMAT	retourne le format complet associé à une cellule.
CALC_SETBACKCOLOR	modifie la couleur de fond des cellules.
CALC_SETBLOCKBACKCOL	les instructions CALC_SETBLOCK* modifient distinctement les informations spécifiant ce format et ce pour un bloc de cellules.
CALC_SETBLOCKBORDER	
CALC_SETBLOCKEDITCTRL	
CALC_SETBLOCKFILTER	
CALC_SETBLOCKFONT	
CALC_SETBLOCKFORMAT	
CALC_SETBLOCKGRID	

Constantes	Description
CK_*	représentent le type (texte, numérique, formule, ...) des cellules.
CTRLDATA_*	indiquent le format sous lequel sont passées les données.
EDTCTRL_*	représentent le contrôle associé à une cellule, utilisé pour la saisie dans la ligne de statut.

GK_ *%	représentent le cadre associé aux cellules.
HJ_ *%	représentent la justification (gauche, droite, centré) des cellules.
JF_ *%	
VJ_ *%	représentent la justification verticale dans les cellules.
Segment BORDERSTYLE	contient les différentes caractéristiques des cadres pouvant entourer les cellules.

Liens entre les cellules

Fonctions	Description
CALC_GETTHREAD\$	retourne la chaîne des liens établie par CALC_SETTHREAD.

Instructions	Description
CALC_SETTHREAD	établie un lien entre les cellules (passage d'une cellule à l'autre par appui de la touche [Tab])

Présentation

Fonctions	Description
CALC_ISLINK%	indique s'il existe un trait entre deux cellules.
CALC_ADDLINK	trace un trait entre deux cellules.
CALC_REMLINK	supprime un trait.

Instructions	Description
CALC_ADDENHANCEDLINK	trace un trait dont il est possible d'indiquer les spécificités.

Constantes	Description
CLS_ *%	précisent la forme du lien
Segment LINKSTYLE	précise toutes les caractéristiques du lien (couleur, points d'encrage, taille, aspect...).
CLP_ *%	précisent le point de départ et le point d'arrivée du lien sur le bord des cellules.

Modification des cellules et des blocs

Fonctions	Description
CALC_ISUNDO%	indique si une action peut être annulée.
CALC_CLEAR SPECIAL	permet de limiter la suppression, limitation précisée par les constantes PT_ *%.
CALC_FILLDOWN	recopie le contenu de la première ligne d'un bloc dans les autres lignes.
CALC_FILLRIGHT	recopie le contenu de la première colonne d'un bloc dans les autres colonnes.
CALC_PUT	remplie les cellules d'un bloc avec la chaîne spécifiée.
CALC_SORT	trie les cellules d'un bloc.
CALC_UNDO	annule la dernière action effectuée.
CALC_UPDATE	bloque ou non la mise à jour des cellules.

Instructions	Description
CALC_CLEAR	efface tout le contenu des cellules d'un bloc.

Constantes
PT_ *%

Déplacement de blocs

Instructions	Description
CALC_SCROLLDOWN	déplace un bloc vers le bas.
CALC_SCROLLLEFT	déplace un bloc vers la gauche.
CALC_SCROLLRIGHT	déplace un bloc vers la droite.
CALC_SCROLLUP	déplace un bloc vers le haut.

Chargement, sauvegarde et impression

Fonctions	Description
CALC_GETFILEFORMAT%	retourne le format d'un fichier.
CALC_GETPRINTFILENBPAGES%	retourne le nombre de pages nécessaires pour imprimer un fichier, en tenant compte de certaines options parmi lesquelles les options d'impression spécifiées par une combinaison des constantes PK_ *%.
CALC_GETPRINTNBPAGES%	retourne le nombre de pages nécessaires pour imprimer une feuille de calcul, en tenant compte de certaines options parmi lesquelles les options d'impression spécifiées par une combinaison des constantes PK_ *%.
CALC_LOAD%	charge un fichier dans une fenêtre NS Calc.
CALC_PRINT%	imprime une feuille de calcul ouverte.
CALC_PRINTFILE%	imprime un fichier.
CALC_SAVE%	sauvegarde une feuille de calcul avec le format spécifié par une des constantes FF_ *%.

Constantes
FF_ *%
PK_ *%

Gestion du presse-papiers

Fonctions	Description
CALC_ISCLIPBOARD%	indique si des données se trouvent dans le presse papiers.
CALC_PASTE	copie le contenu du presse papiers dans la feuille de calcul.
CALC_PASTESPECIAL	permet de définir les actions à effectuer lors du collage, actions précisées par les constantes PT_ *%.

Instructions	Description
CALC_COPY	copie la sélection dans le presse papiers.

Constantes
PT_ *%

Gestion de la mémoire

Fonctions	Description
CALC_GETALLOCATEDMEM%	retourne la taille mémoire occupée par une feuille de calcul.

Instructions	Description
CALC_PREPARE_PUT	effectue une allocation mémoire pour utiliser l'instruction CALC_PUT (voir modification des cellules) sur un bloc important.

Presentation Space

Instructions	Description
CALC_DRAW	permet d'afficher un bloc d'une feuille de calcul dans un "Presentation Space".
CALC_GETDRAWSIZE	retourne la taille d'un tableau comprenant un affichage dans un "Presentation Space".

REFERENCE DE L'API

Avertissement 1 Au sujet de HCALC%, le handle Calc

La majorité des fonctions et instructions de NS Calc nécessitent le handle de la fenêtre NS Calc en premier paramètre. C'est sur cette fenêtre que les fonctions et instructions agissent. Ce handle est noté :

- "handle Calc" dans la syntaxe des fonctions et instructions,
- "HCALC%" dans les exemples.

Ce handle doit impérativement correspondre à une fenêtre de classe Window ayant {NS02CALC} dans son titre.

Avertissement 2 Au sujet des paramètres !V!

Certaines descriptions de paramètres commencent par un !V!. Cela signifie que le paramètre doit impérativement être une variable du type spécifié (cela ne peut être une constante), cette variable étant modifiée par la fonction ou l'instruction. Voici une utilisation typique de tels paramètres, les deuxième et troisième paramètres de CALC_FIRSTNAME% étant notés !V! :

Appel de CALC_FIRSTNAME%, qui permet de savoir si une cellule de la feuille de calcul porte un nom, les variables Nom_assoc\$ et Coord_cellule\$ contenant alors le nom associé à la première cellule nommée et les coordonnées de la cellule

```
; Variables qui seront modifiées par CALC_FIRSTNAME%
LOCAL CSTRING Nom_assoc$, CSTRING Coord_cellule$
IF CALC_FIRSTNAME%(HCALC%,Nom_assoc$,Coord_cellule$) <> 0
MESSAGE "Premier nom associé à une cellule", Nom_assoc$
MESSAGE "Coordonnées de cette cellule", Coord_cellule$
ELSE
MESSAGE "Erreur", "Aucune cellule n'est liée à un nom"
ENDIF
```


Rappel Paramètre coord bloc

La majorité des fonctions et instructions s'appliquent sur un ou plusieurs blocs de cellules. Dans ce cas, le paramètre coord bloc passé à l'appel d'une telle fonction ou instruction est soit :

- une cellule, définie sous la forme RnCm

Ex : "R3C5"

- une ligne, définie sous la forme Rn

Ex : "R5"

- une colonne, définie sous la forme Cm

Ex : "C12"

- un bloc, défini par deux coordonnées de cellules séparées par le caractère deux points ":", dans l'ordre angle 'haut gauche' du bloc / angle 'bas droite'

Ex : "R2C2:R4C4"

- un ensemble de blocs, cellules, lignes et/ou colonnes, défini par des coordonnées de blocs, de cellules, de lignes et/ou de colonnes, chaque coordonnée étant séparée de la précédente par le caractère point virgule ";"

Ex : "R2C2:R3C4;R6C10:R6C12;C14:C15;R18C20"

Dans tous les cas, ces coordonnées sont toujours passées sous forme de chaînes de caractères.

Événements Classiques

Quatre événements classiques ont une signification particulière pour les fenêtres NS Calc :

- Événement CHANGED

Reçu lorsque le texte contenu dans une cellule change. PARAM12% et PARAM34% valent 1 dans ce cas.

- Événement SELECTED

Reçu lorsque le curseur change de place. PARAM1% contient le numéro de colonne et PARAM2% le numéro de ligne de la nouvelle position.

- Événement EXECUTED

Reçu lors d'un double clic sur une cellule. PARAM1% contient le numéro de colonne et PARAM2% le numéro de ligne de cette cellule.

- Événement TIMER

Reçu lorsqu'une sélection est faite avec la souris, uniquement pour les besoins internes de NS Calc. PARAM1% vaut 100. Donc, pour éviter les conflits, ne pas employer de TIMER avec un identifieur 100.

Bien sûr, les autres événements reçus par une fenêtre de classe Window continuent à être reçus. Citons notamment :

- Événements BUTTONDOWN et BUTTONUP pour les clics souris.
- Événement MOUSEMOVE pour les déplacements souris.
- Événement CHARACTER pour les appuis clavier.
- Événement PAINT pour les rafraîchissements d'écran, particulièrement avec la barre d'outils.

Fonctions et Instructions Classiques

Deux familles de fonctions et instructions classiques NCL ont une signification particulière pour les fenêtres NS Calc :

- Instructions LOAD et SAVE

Equivalentes à CALC_LOAD% et CALC_SAVE% (avec le format FF_NSCALC%), sans indication sur le chargement ou la sauvegarde.

- Instructions UPDATE et NOUPDATE

Equivalentes à CALC_UPDATE TRUE% ou CALC_UPDATE FALSE%.

Des fonctions / instructions NCL telles que INSERT, DELETE, SELECT, UNSELECT, SELECTION%, ISSELECTED% n'ont aucune action sur les fenêtres NS Calc.

Fonctions Ns-Calc

Fonction **CALC_CHECKFORMULA**

Valide la syntaxe d'une formule.

Syntaxe	CALC_CHECKFORMULA <i>formula</i> \$		
Paramètre	formula\$	CSTRING	Formule à vérifier
Valeur retournée	INT(1)		

Voir aussi **CALC_CONVERTFORMULA**\$

Fonction **CALC_COLUMN%**

Retourne le numéro de colonne de la cellule active ou de la cellule initiale du bloc sélectionné.

Syntaxe	CALC_COLUMN% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

La cellule initiale d'un bloc est la première cellule sélectionnée, dans un angle du bloc, et n'est pas représentée en vidéo inverse.

Exemple :

```
; Activer la cellule située deux colonnes à droite de la cellule active, en
restant sur la même ligne
CALC_GOTO HCALC%, CALC_ROW%(HCALC%), CALC_COLUMN%(HCALC%)+2
```

Voir aussi **CALC_ROW%**, **CALC_RC2XY**

Fonction CALC_COMMAND%

Permet de définir des actions à effectuer sur la feuille de calcul.

Syntaxe	CALC_COMMAND% (<i>handle-Calc</i> , <i>id%</i> , <i>param%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	id%	INT(4)	identifiant de la feuille de calcul
	param%	INT(4)	constantes ID_*
Valeur retournée	INT(1)		

1. La signification des constantes ID_* est la suivante :
 - a) ID_SETFOCUS_STATUSBAR positionne le focus sur la barre d'état.
 - b) ID_INSERTRC permet d'insérer un retour à la ligne pour écrire sur plusieurs lignes.
2. La description interne des constantes ID_* est la suivante :
 - a) CONST ID_SETFOCUS_STATUSBAR 1
 - b) CONST ID_INSERTRC 2

Fonction **CALC_CONVERTFORMULA\$**

Convertit le format externe d'une formule au format interne.

Syntaxe	CALC_CONVERTFORMULA\$ (<i>internal%</i> , <i>formula\$</i>)		
Paramètres	internal%	INT(1)	format interne
	formula\$	CSTRING	formule à convertir
Valeur retournée	CSTRING		

Voir aussi [CALC_CHECKFORMULA](#)

Fonction **CALC_CREATEFORMATID%**

Permet de créer un nouveau format.

Syntaxe	CALC_CREATEFORMATID% (<i>handle-Calc</i> , <i>baseid%</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	baseid%	INT(2)	identifiant du format créé
Valeur retournée	INT(2)		

*Voir aussi CALC_GETFORMATIDATTR%, CALC_SETFORMATIDATTR%, CALC_SETBLOCKFORMATID%, CALC_GETCELLFORMATID%, CALC_GETDEFFORMATID%, CCALC_FMT *%*

Fonction CALC_FINDNAME\$

Retourne les coordonnées de la cellule qui est associée au nom logique "nom assoc".

Syntaxe	CALC_FINDNAME\$ (<i>handle Calc, nom assoc</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nom assoc	CSTRING(255)	nom logique associé
Valeur retournée	CSTRING(255)		

Si aucune cellule n'a été associée à nom assoc, CALC_FINDNAME\$ retourne une chaîne vide.

Exemple :

```
; Recherche des coordonnées de la cellule associée à "TOTAL"
LOCAL CSTRING Cell$ (255)

MOVE CALC_FINDNAME$ (HCALC%, "TOTAL") TO Cell$
IF Cell$ = ""
MESSAGE "Erreur !", "Aucune cellule n'est associée à TOTAL"
ELSE
MESSAGE "Nom Logique", "La cellule" &&Cell$&& "est associée à TOTAL"
ENDIF
```

Voir aussi [CALC_CREATENAME](#), [CALC_DELETENAME](#), [CALC_FIRSTNAME%](#), [CALC_NEXTNAME%](#)

Fonction CALC_FIRSTNAME%

Retourne l'index correspondant à la première cellule nommée trouvée.

Le nom logique de cette cellule est affecté à la variable nom assoc, ses coordonnées à la variable coord cellule.

En cas de recherche infructueuse, CALC_FIRSTNAME% retourne zéro.

Syntaxe	CALC_FIRSTNAME% (<i>handle Calc, nom assoc, coord cellule</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nom assoc	CSTRING (255)	!V! nom logique associé
	coord cellule	CSTRING (255)	!V! coordonnées de la cellule
Valeur retournée	INT(4)	Index de la première cellule nommée, utilisé pour la recherche suivante, ou 0 si la recherche n'a pas abouti.	

Utiliser ensuite CALC_NEXTNAME%, initialisée avec l'index retourné par CALC_FIRSTNAME%, pour rechercher les autres cellules nommées.

Exemple :

```
; Affichage dans la list box LB_NAMES de tous les noms logiques associés
aux coordonnées des cellules
LOCAL CSTRING Name$(255), CSTRING Cell$(255), i%

MOVE CALC_FIRSTNAME%(HCALC%, Name$, Cell$) TO i%
WHILE i% <> 0
  INSERT ASCENDING Name$ && Cell$ TO LB_NAMES
  MOVE CALC_NEXTNAME%(HCALC%, i%, Name$, Cell$) TO i%
ENDWHILE
```

Voir aussi CALC_NEXTNAME%, CALC_FINDNAME\$, CALC_CREATENAME, CALC_DELETENAME

Fonction CALC_GET\$

Retourne le contenu saisi (textes, valeurs numériques et/ou formules de calcul) des cellules du bloc défini par coord bloc.

Syntaxe	CALC_GET\$ (<i>handle Calc, coord bloc</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
Valeur retournée	CSTRING(255)		

1. Pour obtenir les valeurs des cellules (textes, valeurs et/ou résultats), utiliser la fonction CALC_GETVALUE\$.
2. Pour obtenir les chaînes affichées dans les cellules, en tenant compte des formats d'affichage, utiliser la fonction CALC_GETTEXT\$.
3. Dans la chaîne retournée les données contenues dans les cellules adjacentes appartenant à une même ligne sont séparées entre elles par CHR\$(9) (caractère Tabulation). Les données contenues dans des cellules appartenant à des lignes différentes sont séparées entre elles par CHR\$(13) (CR Carriage Return ou retour à la ligne).

Attention : la longueur maximum de la chaîne retournée est de 255 caractères, caractères séparateurs compris.

4. Si le contenu des cellules définies dans coord bloc est supérieur à cette valeur, CALC_GET\$ retournera une chaîne tronquée.

Exemple :

```
LOCAL CSTRING Contenu$(255)
; Quel que soit le format numérique défini, si les cellules contiennent
; respectivement
; R2C2 = "123.333"
; R2C3 = "321"
; R3C2 = "TOTAL"
; R3C3 = "=R2C2+R2C3"

MOVE CALC_GET$ (HCalc%, "R2C3:R3C3") TO Contenu$
; Contenu$ vaudra "123.333(tab)321(cr)TOTAL(tab)=R2C2+R2C3" où
; (tab)=CHR$(9) et (cr)=CHR$(13)
```

Voir aussi CALC_GETTEXT\$, CALC_GETVALUE\$, CALC_PUT

Fonction CALC_GETALLOCATEDMEM%

Retourne la taille mémoire, en octets, occupée par la feuille de calcul.

Syntaxe	CALC_GETALLOCATEDMEM% (<i>handle Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(4)		

Exemple :

```
LOCAL INT Mem% (4)

MOVE CALC_GETALLOCATEDMEM% (HCALC%) TO Mem%
MESSAGE "Mémoire allouée", STRING$(Mem%/1024, '#', ##0')&&"Ko"
```

Fonction **CALC_GETBACKCOLOR%**

Retourne la couleur par défaut du fond des cellules.

Syntaxe	CALC_GETBACKCOLOR% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(1)	Une des constantes COL_ *% (langage NCL).	

Exemple :

```
IF CALC_GETBACKCOLOR% (HCALC%) = COL_BLUE%  
MESSAGE "Couleur", "Le fond des cellules est en bleu"  
ENDIF
```

Voir aussi [CALC_GETCELLBACKCOL%](#), [CALC_GETDEFCOLOR%](#), [CALC_GETTEXTCOLOR%](#),
[CALC_SETBLOCKBACKCOL](#), [CALC_SETBACKCOLOR](#), [CALC_SETDEFCOLOR](#), [CALC_SETTEXTCOLOR](#)

Fonction **CALC_GETBLOCKHEIGHT%**

Retourne la hauteur, exprimée en pixels, d'un groupe de cellules en évitant les erreurs d'arrondi.

Syntaxe	CALC_GETBLOCKHEIGHT% (<i>handle Calc, block\$</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING	bloc de cellules
Valeur retournée	INT		

Vous pouvez utiliser le nom d'une cellule dans le paramètre block\$.

Voir aussi CALC_GETCELLWIDTH%, CALC_GETBLOCKWIDTH%

Fonction **CALC_GETBLOCKWIDTH%**

Retourne la largeur, exprimée en pixels, d'un groupe de cellules en évitant les erreurs d'arrondi.

Syntaxe	CALC_GETBLOCKWIDTH% (<i>handle Calc, block\$</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING	bloc de cellules
Valeur retournée	INT		

Vous pouvez utiliser le nom d'une cellule dans le paramètre block\$.

Voir aussi CALC_GETCELLWIDTH%, CALC_GETBLOCKHEIGHT%

Fonction **CALC_GETCELLBACKCOL%**

Retourne la couleur de fond de la cellule de coordonnées num ligne, num colonne.
Par défaut, la couleur de fond appliquée aux cellules est COL_BACKGROUND%.

Syntaxe	CALC_GETCELLBACKCOL% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Une des constantes COL_*% (langage NCL).	

Exemple :

```
; Modification de la couleur de fond de la cellule R2C2
IF CALC_GETCELLBACKCOL% (HCALC%, 3, 2) = COL_BLACK%
CALC_SETBLOCKBACKCOL HCALC%, "R2C3", COL_WHITE%
ENDIF
```

Voir **aussi** [CALC_GETBACKCOLOR%](#), [CALC_GETDEFCOLOR%](#),
[CALC_GETTEXTCOLOR%](#), [CALC_SETBLOCKBACKCOL](#), [CALC_SETBACKCOLOR](#), [CALC_SETDEFCOLOR](#),
[CALC_SETTEXTCOLOR](#)

Fonction **CALC_GETCELLDWSPACE**

Précise un espace entre le bord de la cellule et le texte contenu.

Syntaxe	CALC_GETCELLDWSPACE <i>Calc,num_colonne%,num_ligne%</i> (handle		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(2)		

Voir aussi CALC_SETDWSPACE, CALC_SETBLOCKDWSPACE, CALC_GETDWSPACE

Fonction CALC_GETCELLFILTER\$

Retourne le filtre de saisie affecté à la cellule de coordonnées num ligne, num colonne.

Ce filtre de saisie est une chaîne reprenant les attributs de la cellule auxquels s'ajoute la liste des caractères autorisés en saisie.

Chaque attribut, caractère ou ensemble de caractères est séparé du précédent par le caractère #1.

Par défaut, le filtre est 'AUTOSCROLL#1LEFT#1BELOW' (tous les caractères sont acceptés en saisie).

Syntaxe	CALC_GETCELLFILTER\$ (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	CSTRING (255)		

Les attributs de cellules possibles sont une combinaison des suivants :

- **UPCASE** : Saisie en majuscules, quelque soit l'état du clavier. La conversion de minuscule en majuscule est faite en cours de saisie.
- **SKIPBLANKS** : Suppression des éventuels espaces saisis en début de champ et en fin de champ.
- **NOBLANKS** : Impossibilité de saisir des espaces.
- **HIDETEXT** : Le texte saisi n'est pas affiché, mais les caractères sont remplacés par des étoiles.
- **FULLTEXT** : Oblige à saisir le nombre maximum de caractères spécifiés.
- **INTEGER** ou **DATE** ou **TIME** : Pour être acceptée, la valeur saisie doit être respectivement un entier, une date ou une heure.
- **OVER** ou **BELOW** : Le relief donne un aspect bombé (**OVER**) ou creux (**BELOW**) au champ de saisie.
- **AUTOSCROLL** : Un défilement automatique du contenu du champ de saisie est effectué lorsque le nombre de caractères saisis dépasse la taille du champ.
- **CENTER** ou **LEFT** ou **RIGHT** : Centre, justifie à gauche ou justifie à droite la valeur saisie dans le champ de saisie (indépendamment de la justification dans la cellule).
- **MARGIN** : Encadre le champ de saisie.
- "Caractères" : Caractères acceptés lors de la saisie. Ceux ci doivent être encadrés par des apostrophes et séparés par des virgules. Une suite de caractères (dans l'ordre de leur code ASCII), est indiquée par les deux bornes séparées par deux points (..).

Exemple :

```
LOCAL Filtre$  
; N'autoriser que les caractères majuscules de 'A' à 'P', le tiret ' ' et  
les chiffres de '0' à '5' pour la saisie de la cellule R2C3  
MOVE CALC_GETCELLFILTER$ (HCALC%, 3, 2) TO Filtre$  
  
MOVE Filtre$ & #1 & "'A'..'P',' ','0'..'5'" TO Filtre$  
CALC_SETBLOCKFILTER HCALC%, "R2C3", Filtre$
```

Voir aussi CALC_SETBLOCKFILTER

Fonction **CALC_GETCELLFORMAT\$**

Retourne le format d'affichage appliqué à la cellule de coordonnées num ligne, num colonne.

Par défaut, aucun format d'affichage n'est appliqué.

Syntaxe	CALC_GETCELLFORMAT\$ (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	CSTRING (255)	S'il n'y a pas de format appliqué, retourne une chaîne vide.	

Les formats acceptés par NS Calc sont les formats EXCEL (décrits dans le Manuel d'Utilisation de NS Design).

Exemple :

```
; Modification du format d'affichage
IF CALC_GETCELLFORMAT$ (HCALC%, 3, 2) = "#,##0"
CALC_SETBLOCKFORMAT HCALC%, "R2C3", "#,##0.00"
ENDIF
```

Voir aussi CALC_SETBLOCKFORMAT, CALC_GETFORMAT

Fonction **CALC_GETCELLFORMATID%**

Permet de récupérer le format d'une cellule.

Syntaxe	CALC_GETCELLFORMATID% (<i>handle-Calc</i> , <i>X%</i> , <i>Y%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	X%	INT(2)	numéro de la colonne
	Y%	INT(2)	numéro de la ligne
Valeur retournée	INT(2)	Une des constantes CCALC_FMT_*% -1 correspond à une erreur	

Voir aussi CALC_GETFORMATIDATTR%, CALC_SETFORMATIDATTR%, CALC_CREATEFORMATID%, CALC_SETBLOCKFORMATID%, CALC_GETCELLFORMATID%, CALC_GETDEFFORMATID%, CCALC_FMT_*%

Fonction **CALC_GETCELLGRID%**

Retourne le type de cadre appliqué à la cellule de coordonnées num ligne, num colonne.

Par défaut, aucun cadre n'est appliqué.

Syntaxe	CALC_GETCELLGRID% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Combinaison des constantes GK_*% ou zéro s'il n'y a pas de cadre.	

Exemple :

```
LOCAL Cadre%  
; Suppression, s'il y a lieu, des traits sur les bords haut et bas de la  
cellule R2C3  
MOVE CALC_GETCELLGRID%(HCALC%, 3, 2) TO Cadre%  
IF (Cadre% BAND (GK_TOP% + GK_BOTTOM%))  
MOVE Cadre% BXOR (GK_TOP% + GK_BOTTOM%) TO Cadre%  
CALC_SETBLOCKGRID HCALC%, "R2C3", Cadre%  
ENDIF
```

Voir aussi Constantes GK_*%, CALC_GETCELLBORDER, CALC_SETBLOCKBORDER,
CALC_SETBLOCKGRID, CALC_GETFORMAT

Fonction **CALC_GETCELLHORZJUSTIF%**

Retourne la justification horizontale appliquée à la cellule de coordonnées num ligne, num colonne.

Par défaut, la justification horizontale est HJ_DEFAULT%.

Syntaxe	CALC_GETCELLHORZJUSTIF% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Une des constantes <u>HJ_*</u> %.	

Exemple :

```
LOCAL Justif%(1)
; Inversion de la justification horizontale
MOVE CALC_GETCELLHORZJUSTIF%(HCALC%, 3, 2) TO Justif%
IF Justif% = HJ_LEFT%
  CALC_SETBLOCKHORZJUSTIF HCALC, "R2C3", HJ_RIGHT%
ELSEIF Justif% = HJ_RIGHT%
  CALC_SETBLOCKHORZJUSTIF HCALC, "R2C3", HJ_LEFT%
ENDIF
```

Voir aussi Constantes HJ_*%, CALC_SETBLOCKHORZJUSTIF, CALC_GETFORMAT

Fonction **CALC_GETCELLKIND%**

Retourne le type (numérique, alphanumérique, etc.) associé à la cellule de coordonnées num ligne, num colonne.

Par défaut, le type des cellules est CK_DEFAULT%.

Syntaxe	CALC_GETCELLKIND% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Une des constantes <u>CK_*</u> %.	

Exemple :

```
; Appliquer, s'il y a lieu, le type numérique
IF CALC_GETCELLKIND% (HCALC%, 3, 2) <> CK_NUMBER%
CALC_SETBLOCKKIND HCALC%, "R2C3", CK_NUMBER%
ENDIF
```

Voir aussi Constantes CK_*%, CALC_SETBLOCKKIND, CALC_GETFORMAT

Fonction **CALC_GETCELLLENGTH%**

Retourne le nombre maximum de caractères acceptés en saisie dans la cellule de coordonnées num ligne, num colonne.

Par défaut, le maximum est de 255 caractères.

Syntaxe	CALC_GETCELLLENGTH% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(2)		

Exemple :

```
; Appliquer, s'il y a lieu, un nombre maximum de 10 caractères pour la
cellule R2C3
IF CALC_GETCELLLENGTH% (HCALC%, 3, 2) > 10
  CALC_SETBLOCKLENGTH HCALC%, "R2C3", 10
ENDIF
```

Voir aussi [CALC_SETBLOCKLENGTH](#), [CALC_GETFORMAT](#)

Fonction **CALC_GETCELLOVERRUN%**

Positionne la propriété de débordement sur un groupe de cellules.

Syntaxe	CALC_GETCELLOVERRUN% (<i>handle Calc, x%, y%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de ligne
	y%	INT(2)	numéro de colonne
Valeur retournée	INT(1)		

Voir aussi CALC_ISOVERRUN%, CALC_SETBLOCKOVERRUN, CALC_GETCELLOVERRUN%, CALC_SETOVERRUN, CALC_PRINTGETDISPLAYEDCELLSIZE, CALC_GETDISPLAYEDCELLSIZE

Fonction CALC_GETCELLPATTCOL%

Retourne la couleur du pattern utilisé dans la cellule.

Syntaxe	CALC_GETCELLPATTCOL% (<i>handle Calc, X%, Y%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de colonne
	y%	INT(2)	numéro de ligne
Valeur retournée	INT(1)		

Voir aussi CALC_SETBLOCKPATTERN, CALC_SETBLOCKPATTCOL, CALC_GETCELLPATTERN%

Fonction **CALC_GETCELLPATTERN%**

Retourne le pattern utilisé dans la cellule.

Syntaxe	CALC_GETCELLPATTERN% (<i>handle Calc, X%, Y%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de colonne
	y%	INT(2)	numéro de ligne
Valeur retournée	INT(1)		

Voir aussi CALC_SETBLOCKPATTERN, CALC_SETBLOCKPATTCOL, CALC_GETCELLPATTCOL%

Fonction **CALC_GETCELLTEXTCOL%**

Retourne la couleur de texte de la cellule de coordonnées num ligne, num colonne.
Par défaut, la couleur de texte des cellules est COL_NEUTRAL%.

Syntaxe	CALC_GETCELLTEXTCOL% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Une des constantes COL_*% (langage NCL).	

Exemple :

```
; Modification de la couleur de texte de la cellule R2C3
IF CALC_GETCELLTEXTCOL% (HCALC%, 3, 2) = COL_WHITE%
CALC_SETBLOCKTEXTCOL HCALC%, "R2C3", COL_BLACK%
ENDIF
```

Voir aussi [CALC_GETTEXTCOLOR%](#), [CALC_GETDEFCOLOR%](#), [CALC_SETBLOCKTEXTCOL](#),
[CALC_SETTEXTCOLOR](#), [CALC_SETDEFCOLOR](#)

Fonction CALC_GETCELLVERTJUSTIF%

Retourne la justification verticale appliquée à la cellule de coordonnées num ligne, num colonne.

Par défaut, la justification verticale est VJ_DEFAULT%.

Syntaxe	CALC_GETCELLVERTJUSTIF% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(1)	Une des constantes <u>VJ</u> *%.	

Exemple :

```
LOCAL Justif%(1)
; Inversion de la justification verticale
MOVE CALC_GETCELLVERTJUSTIF%(HCALC%, 3, 2) TO Justif%
IF Justif% = VJ_TOP%
  CALC_SETBLOCKVERTJUSTIF HCALC, "R2C3", VJ_BOTTOM%
ELSEIF Justif% = VJ_BOTTOM%
  CALC_SETBLOCKVERTJUSTIF HCALC, "R2C3", VJ_TOP%
ENDIF
```

Voir aussi Constantes VJ *%, CALC_SETBLOCKVERTJUSTIF, CALC_GETFORMAT

Fonction **CALC_GETCELLWIDTH%**

Retourne la largeur, exprimée en pixels, de la cellule de coordonnées num colonne, num ligne.

Syntaxe	CALC_GETCELLWIDTH% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
Valeur retournée	INT(2)		

Exemple :

```
; Appliquer la largeur de la cellule R1C2 aux colonnes 3, 4 et 5
CALC_SETWIDTH HCALC%, "C3:C5", CALC_GETCELLWIDTH% (HCALC%, 2, 1)
```

Voir aussi [CALC_GETDEFHEIGHT%](#), [CALC_GETHEIGHT%](#), [CALC_SETHEIGHT](#), [CALC_GETDEFWIDTH%](#), [CALC_GETWIDTH%](#), [CALC_SETWIDTH](#)

Fonction **CALC_GETDEFCOLOR%**

Retourne la couleur de fond (sous les cellules) de la fenêtre NS Calc.

Par défaut, la couleur de fond est COL_RED%.

Syntaxe	CALC_GETDEFCOLOR% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(1)	Une des constantes COL_*% (langage NCL).	

Exemple :

```
IF CALC_GETDEFCOLOR% (HCALC%) = COL_RED%  
MESSAGE "Couleur", "La fenêtre est en fond rouge"  
ENDIF
```

Voir aussi [CALC_GETBACKCOLOR%](#), [CALC_GETCELLBACKCOL%](#), [CALC_GETTEXTCOLOR%](#),
[CALC_SETBACKCOLOR](#), [CALC_SETDEFCOLOR](#), [CALC_SETTEXTCOLOR](#)

Fonction **CALC_GETDEFCOLWIDTH%**

Retourne la hauteur par défaut, exprimée en unités courantes, des colonnes de la feuille de calcul.

Syntaxe	CALC_GETDEFCOLWIDTH% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT		

Voir aussi CALC_GETDEFWIDTH%, CALC_GETDEFHEIGHT%, CALC_GETHEIGHT%, CALC_SETHEIGHT, CALC_GETWIDTH%, CALC_GETCELLWIDTH%, CALC_SETWIDTH

Fonction **CALC_GETDEFFORMATID%**

Permet de récupérer le format de la feuille de calcul.

Syntaxe	CALC_GETDEFFORMATID% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)	Une des constantes CCALC_FMT_*% -1 correspond à une erreur	

Voir aussi CALC_GETFORMATIDATTR%, CALC_SETFORMATIDATTR%, CALC_CREATEFORMATID%, CALC_SETBLOCKFORMATID%, CALC_GETCELLFORMATID%, CCALC_FMT_%*

Fonction **CALC_GETDEFHEIGHT%**

Retourne la hauteur par défaut, exprimée en pixels, des lignes de la feuille de calcul.

Syntaxe	CALC_GETDEFHEIGHT%
Valeur retournée	INT(2)

Exemple :

```
; Modification de la hauteur de la ligne 2  
CALC_SETHEIGHT HCALC%, "R2", CALC_GETDEFHEIGHT% + 20
```

Voir aussi [CALC_GETDEFROWHEIGHT%](#), [CALC_GETHEIGHT%](#), [CALC_SETHEIGHT](#),
[CALC_GETDEFWIDTH%](#), [CALC_GETWIDTH%](#), [CALC_GETCELLWIDTH%](#), [CALC_SETWIDTH](#)

Fonction **CALC_GETDEFROWHEIGHT%**

Retourne la hauteur par défaut, exprimée en unités courantes, des lignes de la feuille de calcul.

Syntaxe	CALC_GETDEFROWHEIGHT% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT		

Voir aussi CALC_GETDEFHEIGHT%, CALC_GETHEIGHT%, CALC_SETHEIGHT, CALC_GETDEFWIDTH%, CALC_GETWIDTH%, CALC_GETCELLWIDTH%, CALC_SETWIDTH

Fonction **CALC_GETDEFWIDTH%**

Retourne la largeur par défaut, exprimée en pixels, des colonnes de la feuille de calcul.

Syntaxe	CALC_GETDEFWIDTH%
Valeur retournée	INT(2)

Exemple :

```
; Modification de la largeur des colonnes 3 et 4  
CALC_SETWIDTH HCALC%, "C3;C4", CALC_GETDEFWIDTH% + 30
```

Voir **aussi** CALC_GETDEFCOLWIDTH%, CALC_GETWIDTH%, CALC_GETCELLWIDTH%,
CALC_SETWIDTH, CALC_GETDEFHEIGHT%, CALC_GETHEIGHT%, CALC_SETHEIGHT

Fonction **CALC_GETDIMENSION**

Retourne le nombre de colonnes et de lignes effectivement utilisées dans une feuille.

Syntaxe	CALC_GETDIMENSION (<i>handle Calc, cols%, rows%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	cols%	INT	nombre de colonnes utilisées
	rows%	INT	nombre de lignes utilisées

Fonction **CALC_GETDWSPACE**

Permet de récupérer l'espace entre le bord d'une cellule et le texte.

Syntaxe	CALC_GETDWSPACE <i>handle Calc</i>		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

Voir aussi CALC_SETDWSPACE, CALC_SETBLOCKDWSPACE, CALC_GETCELLDWSPACE

Fonction CALC_GETFILEFORMAT%

Retourne le format du fichier dont les nom et chemin d'accès sont spécifiés dans nom fichier.

Deux formats sont reconnus par NS Calc : le format NS Calc et le format texte.

Syntaxe	CALC_GETFILEFORMAT% (<i>nom fichier</i>)		
Paramètre	nom fichier	CSTRING(255)	nom du fichier à étudier
Valeur retournée	INT(1)	La constante FF_NSCALC% ou la constante FF_TEXT% s'il s'agit d'un format reconnu par NS Calc 1 pour les autres formats ou en cas d'erreur.	

Exemple :

```
IF CALC_GETFILEFORMAT% ('c:\calc\work\ess1.cal') = 1
MESSAGE "Erreur", "Fichier non lisible"
ELSE
IF NOT CALC_LOAD% (HCALC%, 'c:\calc\work\ess1.cal')
MESSAGE "Erreur", "Chargement impossible"
EXIT
ENDIF
ENDIF
```

Voir aussi Constantes FF_*, CALC_LOAD%, CALC_SAVE%

Fonction **CALC_GETFORMATIDATTR%**

Permet de récupérer un format et ses attributs.

Syntaxe	CALC_GETFORMATIDATTR% (<i>handle-Calc, id%, idattr%, pvalue%</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	id%	INT(2)	identifiant de la feuille de calcul
	idattr%	POINTER	une des constantes CCALC_FMT_*
	pvalue%	POINTER	un pointeur sur une variable dont le type est indiqué dans NSCALC.NCL en commentaire à droite de la constante CCALC_FMT_*
Valeur retournée	INTEGER		

Voir aussi CALC_SETFORMATIDATTR%, CALC_CREATEFORMATID%, CALC_SETBLOCKFORMATID%, CALC_GETCELLFORMATID%, CALC_GETDEFFORMATID%, CCALC_FMT_*

Fonction **CALC_GETHEIGHT%**

Retourne la hauteur, exprimée en pixels, de la ligne de numéro num ligne.

Syntaxe	CALC_GETHEIGHT% (<i>handle Calc, num ligne</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num ligne	INT(2)	numéro de ligne
Valeur retournée	INT(2)		

Exemple :

```
; Doubler la hauteur de la ligne 2  
CALC_SETHEIGHT HCALC%, "R2", CALC_GETHEIGHT%(HCALC%, 2) * 2
```

Voir aussi [CALC_GETDEFHEIGHT%](#), [CALC_SETHEIGHT](#), [CALC_GETDEFWIDTH%](#), [CALC_GETWIDTH%](#), [CALC_GETCELLWIDTH%](#)

Fonction **CALC_GETLEFTCOLUMNS%**

Retourne le nombre de colonnes précédemment bloquées par l'instruction **CALC_SETLEFTCOLUMNS**.

Syntaxe	CALC_GETLEFTCOLUMNS% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

Exemple :

```
; Connaître le nombre de colonnes bloquées
LOCAL INT NBR% (2)
MOVE CALC_GETLEFTCOLUMNS% (HCALC%) to NBR%
IF NBR%
MESSAGE "Blocage", NBR% && "colonnes sont bloquées"
ELSE
MESSAGE "Erreur", "Aucune colonne n'est bloquée"
ENDIF
```

Voir aussi **CALC_SETLEFTCOLUMNS**, **CALC_GETTOPROWS%**, **CALC_SETTOPROWS**

Fonction **CALC_GETNBCELLS%**

Retourne le nombre de cellules définies pour une ligne donnée dans la feuille de calcul.

Syntaxe	CALC_GETNBCELLS% (<i>handle Calc, num ligne</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num ligne	INT(2)	numéro de ligne
Valeur retournée	INT(2)		

Par défaut cette fonction est équivalente à **CALC_GETNBCOLUMNS%**. Si le tableau est à cellules de taille variable (cf **CALC_SETVARCELLWIDTHS**), cette fonction peut retourner des valeurs différentes suivant la ligne.

Exemple :

```
; Récupérer le nombre de cellules de la troisième ligne  
MESSAGE "La feuille de calcul comporte à la 3e ligne", CALC_GETNBCELLS%  
(HCALC%, 3) && "cellules."
```

Voir aussi **CALC_SETNBCELLS**, **CALC_GETNBCOLUMNS%**, **CALC_SETVARCELLWIDTHS**

Fonction **CALC_GETNBCOLUMNS%**

Retourne le nombre de colonnes définies pour la feuille de calcul.

Syntaxe	CALC_GETNBCOLUMNS% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

Si l'instruction **CALC_SETNBCOLUMNS** initialisant le nombre de colonnes de la feuille de calcul n'a pas été utilisée, **CALC_GETNBCOLUMNS%** retourne la taille par défaut, soit 256 colonnes.

Exemple :

```
; Fixer, si nécessaire, le nombre de colonnes de la feuille de calcul au
maximum
IF CALC_GETNBCOLUMNS%(HCALC%) < CALC_MAX_COLUMNS%
  CALC_SETNBCOLUMNS HCALC%, CALC_MAX_COLUMNS%
ENDIF
```

Voir aussi **CALC_GETNBROWS%**, **CALC_GETNBCELLS%**, **CALC_SETNBCOLUMNS**, **CALC_SETNBROWS**

Fonction **CALC_GETNBROWS%**

Retourne le nombre de lignes définies pour la feuille de calcul.

Syntaxe	CALC_GETNBROWS% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

Si l'instruction **CALC_SETNBROWS** initialisant le nombre de lignes de la feuille de calcul n'a pas été utilisée, **CALC_GETNBROWS%** retourne la taille par défaut (et maximale), soit **CALC_MAX_ROWS%**.

Exemple :

```
; Fixer, si nécessaire, le nombre de lignes de la feuille de calcul au
maximum
IF CALC_GETNBROWS%(HCALC%) < CALC_MAX_ROWS%
  CALC_SETNBROWS HCALC%, CALC_MAX_ROWS%
ENDIF
```

Voir aussi **CALC_GETNBCOLUMNS%**, **CALC_SETNBROWS**, **CALC_SETNBCOLUMNS**

Fonction **CALC_GETONERROR%**

Permet de récupérer les messages d'erreur.

Syntaxe	CALC_GETONERROR% (<i>handle Calc</i> , <i>errtype%</i> , <i>RPLTYPE%</i> , <i>RPLMSG\$</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	errtype%	INT(4)	constantes CE_*
	RPLTYPE%	INT(4)	-1 : message par défaut 0 : message donné par CALC_SETONERROR%
	RPLMSG\$	INT(4)	message d'erreur
Valeur retournée	INT		

Voir aussi Constantes CE_*, CALC_SETONERROR%

Fonction CALC_GETPRINTFILENBPAGES%

Retourne le nombre de pages nécessaires pour l'impression du bloc de coordonnées coord bloc, extrait du fichier nom fichier.

Syntaxe	CALC_GETPRINTFILENBPAGES% (<i>handle prn, nom fichier, coord bloc, param imp, zoom, titre, fonte titre, style fonte</i>)		
Paramètres	handle prn	INT(4)	handle d'imprimante sur laquelle se fera l'impression.
	nom fichier	CSTRING(255)	nom et chemin d'accès du fichier à prendre en compte
	coord bloc	CSTRING(255)	coordonnées du bloc
	param imp	INT(2)	paramètres d'impression (justification, impression ou non du titre, impression ou non des numéros de pages, etc.) ; ces paramètres sont une combinaison des constantes PK_*.
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression 'taille réelle')
	titre	CSTRING(255)	titre de l'impression
	fonte titre	CSTRING(255)	fonte utilisée pour le titre (nom et taille)
	style fonte	INT(1)	attributs de la fonte pour le titre (gras, italique...) sous forme d'une combinaison des constantes GFS_* (librairie NSGRAPH)
Valeur retournée	INT(2)	En cas d'erreur ou si le format du fichier n'est pas reconnu, CALC_GETPRINTFILENBPAGES% retourne 0.	

1. handle prn est le handle retourné par la fonction REP_OPEN% (librairie NSPRN). Zéro (0) correspond à l'imprimante par défaut installée sur le système.

2. titre, fonte titre et style fonte ne sont utiles (et utilisés) que si l'impression du titre est demandée, c'est à dire si la constante PK_TITLE% se trouve dans param imp.

Exemple :

```
LOCAL Nb_Pages%(2)

MOVE CALC_GETPRINTFILENBPAGES%(0, "ess1.cal", "R1C1:R10C10", PK_LEFT% BOR
PK_TITLE%, \
100, "ESSAI", "Times New Roman, 10", GFS_BOLD%) TO Nb_Pages%
```



```
MESSAGE "Impression", Nb_Pages% && "pages sont nécessaires" && "pour  
imprimer le bloc R1C1 / R10C10"
```

Voir aussi Constantes PK_*, CALC_GETPRINTNBPAGES, CALC_PRINT, CALC_PRINTFILE

Fonction CALC_GETPRINTNBPAGES%

Retourne le nombre de pages nécessaires pour l'impression du bloc de coordonnées coord bloc, extrait de la fenêtre NS Calc spécifiée par handle Calc.

Syntaxe	CALC_GETPRINTNBPAGES% (<i>handle prn, handle Calc, coord bloc, param imp, zoom, titre, fonte titre, style fonte</i>)		
Paramètres	handle prn	INT(4)	handle d'imprimante sur laquelle se fera l'impression.
	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	param imp	INT(2)	paramètres d'impression (justification, impression ou non du titre, impression ou non des numéros de pages, etc.) ; ces paramètres sont une combinaison des constantes PK_*.
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression 'taille réelle')
	titre	CSTRING(255)	titre de l'impression
	fonte titre	CSTRING(255)	fonte utilisée pour le titre (nom et taille)
	style fonte	INT(1)	attributs de la fonte pour le titre (gras, italique...) sous forme d'une combinaison des constantes GFS_*% (librairie NSGRAPH)
Valeur retournée	INT(2)	En cas d'erreur, CALC_GETPRINTFILENBPAGES% retourne 0.	

1. handle prn est le handle retourné par la fonction REP_OPEN% (librairie NSPRN). Zéro (0) correspond à l'imprimante par défaut installée sur le système.

2. titre, fonte titre et style fonte ne sont utiles (et utilisés) que si l'impression du titre est demandée, c'est à dire si la constante PK_TITLE% se trouve dans param imp.

Exemple :

```
MOVE CALC_GETPRINTNBPAGES%(0, HCALC%, "R1C1:R10C10", PK_LEFT% BOR
PK_TITLE%, \
100, "ESSAI", "Times New Roman, 10", GFS_BOLD%) TO Nb_Pages%
MESSAGE "Impression", Nb_Pages% && "pages sont nécessaires" && "pour
imprimer le bloc R1C1 / R10C10"
```

Voir aussi Constantes PK_*, CALC_GETPRINTFILENBPAGES, CALC_PRINT, CALC_PRINTFILE

Fonction CALC_GETTEXT\$

Retourne la concaténation des chaînes affichées dans les cellules du bloc passé en paramètre (textes, résultats des formules de calcul, valeurs numériques), en tenant compte du ou des formats d'affichage des valeurs numériques.

Syntaxe	CALC_GETTEXT\$ (<i>handle Calc, coord bloc</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
Valeur retournée	CSTRING(255)		

1. Pour obtenir le contenu saisi (textes, valeurs numériques et/ou formules de calcul), utiliser la fonction CALC_GET\$.
2. Pour obtenir les valeurs des cellules (textes, valeurs et/ou résultats), utiliser la fonction CALC_GETVALUE\$.
3. Dans la chaîne retournée, les valeurs et textes affichés dans les cellules adjacentes appartenant à une même ligne sont séparés entre eux par CHR\$(9) (caractère Tabulation). Les valeurs et textes affichés dans des cellules appartenant à des lignes différentes sont séparés entre eux par CHR\$(13) (CR Carriage Return retour à la ligne).

La longueur maximum de la chaîne retournée est de 255 caractères, caractères séparateurs compris.

4. Si le contenu des cellules définies dans coord bloc est supérieur à cette valeur, CALC_GETTEXT\$ retournera une chaîne tronquée.

Exemple :

```
LOCAL CSTRING Contenu$(255)

; En utilisant un format numérique ###.00 dans les cellules contenant
; R2C2 = "123.333"
; R2C3 = "321.0"
; R3C2 = "TOTAL"
; R3C3 = "=R2C2+R2C3"

MOVE CALC_GETTEXT$ (HCalc%, "R2C3:R3C3") TO Contenu$
; Contenu$ vaudra "123.33(tab)321.00(cr)TOTAL(tab)444.33" où (tab) =
CHR$(9) et (cr) = CHR$(13)
```

Voir aussi CALC_GET\$, CALC_GETVALUE\$, CALC_PUT

Fonction **CALC_GETTEXTCOLOR%**

Retourne la couleur par défaut du texte des cellules.

Syntaxe	CALC_GETTEXTCOLOR% (<i>handle Calc</i>)		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(1)	Une des constantes COL_*% (langage NCL).	

Exemple :

```
IF CALC_GETTEXTCOLOR% (HCALC%) = COL_BLACK%
MESSAGE "Couleur", "Le texte des cellules est noir"
ENDIF
```

Voir aussi [CALC_GETBACKCOLOR%](#), [CALC_GETCELLBACKCOL%](#), [CALC_GETDEFCOLOR%](#), [CALC_SETBACKCOLOR](#), [CALC_SETBLOCKBACKCOL](#), [CALC_SETDEFCOLOR](#), [CALC_SETTEXTCOLOR](#)

Fonction CALC_GETTHREAD\$

Retourne l'ordre de chaînage des cellules (ordre de passage automatique du focus) défini par CALC_SETTHREAD.

Syntaxe	CALC_GETTHREAD\$ (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	CSTRING(255)		

1. Lorsqu'un tel chaînage est défini sur une feuille, il est impossible, pour l'utilisateur final, de cliquer sur une cellule non incluse dans ce chaînage et donc de modifier les cellules autres que celles chaînées.
2. Dans la chaîne retournée, chaque cellule ou bloc est séparé(e) du(de la) précédent(e) par un point-virgule ";".
3. Si aucun chaînage n'avait été défini par l'instruction CALC_SETTHREAD, la chaîne retournée sera vide.

Exemple :

```
; Ajout du bloc R2C3:R3C4 et de la cellule R15C10 à la fin de la chaîne  
précédemment définie par CALC_SETTHREAD  
CALC_SETTHREAD HCALC%, CALC_GETTHREAD$(HCALC%) & ";R2C3:R3C4;R15C10"
```

Voir aussi [CALC_SETTHREAD](#)

Fonction **CALC_GETTOPROWS%**

Retourne le nombre de lignes précédemment bloquées par l'instruction **CALC_SETTOPROWS**.

Syntaxe	CALC_TOPROWS% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(2)		

Exemple :

```
; Connaître le nombre de lignes bloquées
LOCAL INT NBR% (2)

MOVE CALC_GETTOPROWS% (HCALC%) to NBR%
IF NBR%
MESSAGE "Blocage", NBR% && "lignes bloquées"
ELSE
MESSAGE "Erreur", "Aucune ligne n'est bloquée"
ENDIF
```

Voir aussi [CALC_SETTOPROWS](#), [CALC_GETLEFTCOLUMNS%](#), [CALC_SETLEFTCOLUMNS](#)

Fonction **CALC_GETUPDATE%**

Récupère le statut de la mise à jour graphique de la fenêtre NS-Calc.

Syntaxe	CALC_GETUPDATE% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INTEGER	TRUE%, toute modification est répercutée immédiatement à l'écran FALSE%, aucune mise à jour de la fenêtre NS-Calc n'est effectuée.	

Voir aussi **CALC_UPDATE**

Fonction CALC_GETVALUE\$

Retourne la concaténation des valeurs contenues dans les cellules du bloc passé en paramètre (textes, résultats des formules de calcul, valeurs numériques), sans tenir compte du ou des formats d'affichage des valeurs numériques.

Syntaxe	CALC_GETVALUE\$ (<i>handle Calc, coord bloc</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
Valeur retournée	CSTRING(255)		

1. Pour obtenir le contenu saisi (textes, valeurs numériques et/ou formules de calcul), utiliser la fonction CALC_GET\$.
2. Pour obtenir les chaînes affichées dans les cellules, en tenant compte des formats d'affichage, utiliser la fonction CALC_GETTEXT\$.
3. Dans la chaîne retournée, les valeurs et textes affichés dans les cellules adjacentes appartenant à une même ligne sont séparés entre eux par CHR\$(9) (caractère Tabulation). Les valeurs et textes affichés dans des cellules appartenant à des lignes différentes sont séparés entre eux par CHR\$(13) (CR Carriage Return retour à la ligne).

La longueur maximum de la chaîne retournée est de 255 caractères, caractères séparateurs compris.

4. Si le contenu des cellules définies dans coord bloc est supérieur à cette valeur, CALC_GETVALUE\$ retournera une chaîne tronquée.

Exemple :

```
LOCAL CSTRING Contenu$(255)
; En utilisant un format numérique ###.00 dans les cellules contenant
; R2C2 = "123.333"
; R2C3 = "321.0"
; R3C2 = "TOTAL"
; R3C3 = "=R2C2+R2C3"

MOVE CALC_GETVALUE$ (HCALC%, "R2C3:R3C3") TO Contenu$
; Contenu$ vaudra "123.333(tab)321(cr)TOTAL(tab)444.333" où (tab) = CHR$(9)
et (cr) = CHR$(13)
```

Voir aussi CALC_GET\$, CALC_GETTEXT\$, CALC_PUT

Fonction CALC_GETWIDTH%

Retourne la largeur, exprimée en pixels, de la colonne de numéro num colonne.

Syntaxe	CALC_GETWIDTH% (<i>handle Calc, num colonne</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num colonne	INT(2)	numéro de la colonne
Valeur retournée	INT (2)		

Exemple :

```
; Doubler la largeur de la colonne 3
CALC_SETWIDTH HCALC%, "3", CALC_GETWIDTH%(HCALC%, 3) * 2
```

Voir aussi [CALC_GETDEFWIDTH%](#), [CALC_GETCELLWIDTH%](#), [CALC_SETWIDTH](#), [CALC_GETDEFHEIGHT%](#), [CALC_GETHEIGHT%](#)

Fonction CALC_GETZOOM%

Retourne la largeur, exprimée en pixels, de la colonne de numéro num colonne.

Syntaxe	CALC_GETZOOM% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(2)		

Voir aussi [CALC_SETZOOM](#)

Fonction **CALC_ISCELLEEMPTY%**

Indique si la cellule de coordonnées num ligne, num colonne est vide.

Syntaxe	CALC_ISCELLEEMPTY% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num colonne	INT(2)	numéro de colonne
	num ligne	INT(2)	numéro de ligne
Valeur retournée	INT (1)	TRUE% si la cellule est vide. FALSE% si la cellule contient du texte, une valeur numérique ou une formule de calcul.	

Exemple :

```
; Affichage d'une étoile si la cellule est vide
IF CALC_ISCELLEEMPTY% (HCALC%, 3, 2)
  CALC_PUT HCALC%, "R2C3", "*"
ENDIF
```

Voir aussi **CALC_ISCELLLOCKED%**

Fonction **CALC_ISCELLLOCKED%**

Indique si la cellule de coordonnées num ligne, num colonne a été précédemment verrouillée par un appel à **CALC_LOCKBLOCK**. Si c'est le cas, aucune saisie ne pourra être effectuée dans cette cellule.

Syntaxe	CALC_ISCELLLOCKED% (<i>handle Calc, num colonne, num ligne</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num colonne	INT(2)	numéro de colonne
	num ligne	INT(2)	numéro de ligne
Valeur retournée	INT (1)	TRUE% si la cellule est verrouillée. FALSE% si elle peut être modifiée.	

Exemple :

```

LOCAL c%(2), l%(2)
LOCAL nb_col%(2), nb_ligne%(2)

MOVE CALC_GETNBCOLUMNS%(HCALC%) TO nb_col%
MOVE CALC_GETNBROWS%(HCALC%) TO nb_ligne%
MOVE 1 TO c%
MOVE 1 TO l%

; Remplacer la couleur de fond des cellules verrouillées
WHILE l% <= nb_ligne%
WHILE c% <= nb_col%
IF CALC_ISCELLLOCKED%(HCALC%, c%, l%)
CALC_SETBLOCKBACKCOL HCALC%, "R"&l% & "C"&c%, COL_LIGHTMAGENTA%
ENDIF
MOVE c%+1 TO c%
ENDWHILE
MOVE 1 TO c%
MOVE l%+1 TO l%
ENDWHILE

```

Voir aussi **CALC_ISLOCKED%**, **CALC_LOCK**, **CALC_LOCKBLOCK**

Fonction **CALC_ISCLIPBOARD%**

Retourne une indication sur le contenu du presse papiers.

Syntaxe	CALC_ISCLIPBOARD% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si le presse papiers contient des données. FALSE% si le presse papiers est vide.	

Exemple :

```
; Activation de l'option Paste si le presse papiers est non vide, sur  
l'événement INIT du menu correspondant  
IF CALC_ISCLIPBOARD% (HCALC%)  
ENABLE MENU_PASTE  
ELSE  
DISABLE MENU_PASTE  
ENDIF
```

Voir aussi CALC_COPY, CALC_PASTE, CALC_PASTESPECIAL

Fonction **CALC_ISFORMULADISPLAYED%**

Retourne une indication sur l'affichage des formules de calcul dans les cellules.

Syntaxe	CALC_ISFORMULADISPLAYED% (<i>handle Calc</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT (1)	TRUE% si les formules de calcul sont affichées. FALSE% si le résultat de ces formules est affiché.	

Exemple :

```
; Inversion de l'affichage formules/résultats sur le menu correspondant
IF CALC_ISFORMULADISPLAYED% (HCALC%)
  CALC_DISPLAYFORMULA HCALC%, FALSE%
ELSE
  CALC_DISPLAYFORMULA HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_DISPLAYFORMULA**

Fonction CALC_ISGRID%

Retourne une indication sur la présence de la grille dans la feuille de calcul.

Syntaxe	CALC_ISGRID% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si la grille est affichée. FALSE% si la grille est absente.	

Exemple :

```
; Inversion de l'affichage de la grille sur le menu correspondant
IF CALC_ISGRID% (HCALC%)
  CALC_SETGRID HCALC%, FALSE%
ELSE
  CALC_SETGRID HCALC%, TRUE%
ENDIF
```

Voir aussi [CALC_SETGRID](#)

Fonction **CALC_ISHEADING%**

Retourne une indication sur l'affichage des numéros de lignes et de colonnes dans la feuille de calcul.

Syntaxe	CALC_ISHEADING% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si les numéros de lignes et de colonnes sont affichés. FALSE% si les numéros de lignes et de colonnes sont absents.	

Exemple :

```
; Inversion de l'affichage des numéros sur le menu correspondant
IF CALC_ISHEADING% (HCALC%)
  CALC_SETHEADING HCALC%, FALSE%
ELSE
  CALC_SETHEADING HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_SETHEADING**

Fonction **CALC_ISHEIGHTLOCKED%**

Retourne une indication sur la possibilité de changer la hauteur des lignes avec la souris.

Syntaxe	CALC_ISHEIGHTLOCKED% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% s'il est possible de changer la hauteur des lignes. FALSE% s'il n'est pas possible de changer la hauteur des lignes.	

Exemple :

```
; Inversion du blocage de la hauteur des lignes sur le menu correspondant
IF CALC_ISHEIGHTLOCKED%(HCALC%)
  CALC_LOCKHEIGHT HCALC%, FALSE%
ELSE
  CALC_LOCKHEIGHT HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_LOCKHEIGHT**

Fonction **CALC_ISLINK%**

Retourne une indication sur l'existence d'un trait entre la cellule de coordonnées num ligne src, num colonne src et la cellule de coordonnées num ligne dest, num colonne dest.

Syntaxe	CALC_ISLINK% (<i>handle Calc, num ligne src, num colonne src, num ligne dest, num colonne dest</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	num ligne src	INT(2)	ligne source
	num colonne src	INT(2)	colonne source
	num ligne dest	INT(2)	ligne destination
	num colonne dest	INT(2)	colonne destination
Valeur retournée	INT (1)	TRUE% s'il existe un trait entre les deux cellules. FALSE% s'il n'existe pas de trait entre les deux cellules.	

Exemple :

```
; Suppression, s'il existe, du trait tracé entre R1C1 et R3C4
IF CALC_ISLINK%(HCALC%, 1, 1, 3, 4)
CALC_REMLINK HCALC%, 1, 1, 3, 4
ENDIF
```

Voir aussi [CALC_ADDLINK](#), [CALC_ADDENHANCEDLINK](#), [CALC_REMLINK](#)

Fonction CALC_ISLOCKED%

Indique si la feuille de calcul a été précédemment verrouillée par un appel à CALC_LOCK. Si c'est le cas, aucune saisie ne pourra être effectuée sur cette feuille.

Syntaxe	CALC_ISLOCKED% (handle-Calc)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si la feuille de calcul est verrouillée. FALSE% si les cellules peuvent être modifiées.	

Exemple :

```
; Inverser le verrouillage de la feuille sur le menu correspondant
IF CALC_ISLOCKED% (HCALC%)
  CALC_LOCK HCALC%, FALSE%
ELSE
  CALC_LOCK HCALC%, TRUE%
ENDIF
```

Voir aussi CALC_ISCELLLOCKED%, CALC_LOCK, CALC_LOCKBLOCK

Fonction CALC_ISOVERRUN%

Retourne une indication sur le débordement (ou non) du texte d'une cellule sur les cellules placées à droite.

Syntaxe	CALC_ISOVERRUN% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si il y a débordement du texte FALSE% si n'il y a pas de débordement du texte	

Fonction **CALC_ISPAGEBREAK%**

Permet de savoir à un endroit précis s'il y a ou non un saut de page.

Syntaxe	CALC_ISPAGEBREAK% <i>handle-Calc, roworcol%, coor%</i>		
Paramètres	handle- Calc	INT(4)	handle fenêtre NS-Calc
	roworcol%	INTEGER	saut de page à partir d'une colonne ou d'une ligne
	coor%	INT(2)	coordonnées de la colonne ou de la ligne
Valeur retournée	INTEGER		

Voir aussi **CALC_SETPAGEBREAK%**

Fonction **CALC_ISREDO%**

Retourne une indication sur la possibilité de refaire ou non la dernière action effectuée.

Syntaxe	CALC_ISREDO% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si l'annulation de l'action est possible. FALSE% si l'annulation de l'action n'est pas possible.	

1. Les différentes actions pouvant être annulées concernent les actions effectuées avec le presse papiers (couper, coller, etc.) et la saisie clavier.
2. Si l'annulation de l'action est possible, celle ci peut être effectuée par l'instruction **CALC_UNDO**.

Exemple :

```
; Inverser le verrouillage de la feuille sur le menu correspondant
IF CALC_ISLOCKED% (HCALC%)
  CALC_LOCK HCALC%, FALSE%
ELSE
  CALC_LOCK HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_REDO**, **CALC_UNDO**, **CALC_COPY**, **CALC_PASTE**, **CALC_PASTESPECIAL**

Fonction **CALC_ISSTATUSBAR%**

Retourne une indication sur la visibilité ou non de la ligne de statut.

Syntaxe	CALC_ISSTATUSBAR% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si la ligne de statut est visible. FALSE% si la ligne de statut n'est pas visible.	

Exemple :

```
; Affichage, si nécessaire, de la ligne de statut
IF NOT CALC_ISSTATUSBAR%(HCALC%)
CALC_SETSTATUSBAR HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_SETSTATUSBAR**

Fonction **CALC_ISUNDO%**

Retourne une indication sur la possibilité d'annuler ou non la dernière action effectuée.

Syntaxe	CALC_ISUNDO% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si l'annulation de l'action est possible. FALSE% si l'annulation de l'action n'est pas possible.	

1. Les différentes actions pouvant être annulées concernent les actions effectuées avec le presse papiers (couper, coller, etc.) et la saisie clavier.
2. Si l'annulation de l'action est possible, celle ci peut être effectuée par l'instruction **CALC_UNDO**.

Exemple :

```
; Menu activé si la dernière action peut être annulée
IF CALC_ISUNDO%(HCALC%)
ENABLE MENU_UNDO
ELSE
DISABLE MENU_UNDO
ENDIF
```

Voir aussi **CALC_UNDO**, **CALC_COPY**, **CALC_PASTE**, **CALC_PASTESPECIAL**

Fonction **CALC_ISUNLOCKEDUNDOREDO%**

Autorise les actions "refaire" et "annuler" de la dernière action effectuée.

Syntaxe	CALC_ISUNLOCKEDUNDOREDO% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)		

Voir aussi **CALC_UNLOCKUNDOREDO**

Fonction CALC_ISVARCELLWIDTHS%

Permet de savoir si la feuille est spécifiée avec des cellules de taille variable.

Syntaxe	CALC_ISVARCELLWIDTHS% (<i>handle-Calc</i>)		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)		

Fonction **CALC_ISWIDTHLOCKED%**

Retourne une indication sur la possibilité de changer la largeur des colonnes avec la souris.

Syntaxe	CALC_ISWIDTHLOCKED% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% s'il est possible de changer la largeur des colonnes. FALSE% s'il n'est pas possible de changer la largeur des colonnes.	

Exemple :

```
; Inversion du blocage de la largeur des colonnes sur le menu correspondant
IF CALC_ISWIDTHLOCKED%(HCALC%)
  CALC_LOCKWIDTH HCALC%, FALSE%
ELSE
  CALC_LOCKWIDTH HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_LOCKWIDTH**

Fonction CALC_ISZERODISPLAYED%

Retourne une indication sur l'affichage dans les cellules des valeurs numériques égales à 0 (zéro).

Syntaxe	CALC_ISZERODISPLAYED% (<i>handle-Calc</i>)		
Paramètre	handle- Calc	INT(4)	handle fenêtre NS-Calc
Valeur retournée	INT(1)	TRUE% si les valeurs nulles sont affichées, FALSE% sinon.	

Exemple :

```
; Inversion de l'affichage des valeurs nulles sur le menu correspondant
IF CALC_ISZERODISPLAYED%(HCALC%)
  CALC_DISPLAYZERO HCALC%, FALSE%
ELSE
  CALC_DISPLAYZERO HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_DISPLAYZERO**

Fonction **CALC_LOAD%**

Charge dans la feuille de calcul le fichier (format NS Calc ou format texte) dont le nom est passé en paramètre.

Syntaxe	CALC_LOAD% (<i>handle Calc, nom fichier</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	nom fichier	CSTRING(255)	nom du fichier à charger
Valeur retournée	INT(1)	TRUE% si le fichier a été correctement chargé. FALSE% en cas de problème lors du chargement.	

"CALC_LOAD%(HCALC%, F\$)" est équivalent à "LOAD F\$ TO HCALC%", sans indication sur le chargement.

Exemple :

```
; Charger le fichier RESU_92 depuis la disquette contenue dans le lecteur A
LOCAL CSTRING Name$ (255)

MOVE "A:\RESU_92" to Name$
IF NOT CALC_LOAD% (HCALC%, Name$)
MESSAGE "Erreur", "Fichier" && Name$ && "introuvable"
ELSE
MESSAGE "OK", "Fichier" && Name$ && "chargé"
ENDIF
```

Voir aussi CALC_SAVE%, LOAD (Manuel de Programmation NCL)

Fonction CALC_MULTI_SELECTION\$

Retourne les coordonnées du bloc d'index index bloc dans le cas d'une sélection multiple.

index bloc vaut 0 pour le dernier bloc sélectionné ; dans ce cas, CALC_MULTI_SELECTION\$ est équivalent à CALC_SELECTION\$. Pour les autres blocs, le premier bloc sélectionné a pour index 1, le deuxième 2, etc.

Syntaxe	CALC_MULTI_SELECTION\$ (<i>handle Calc, index bloc</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	index bloc	INT(2)	index du bloc
Valeur retournée	CSTRING (255)		

1. CALC_MULTI_SELECTION\$ retourne une chaîne vide si l'index passé est supérieur au nombre de blocs sélectionnés.
2. Les coordonnées du bloc sont retournées dans une chaîne de caractères sous la forme "RnCm:RzCt" : RnCm sont les coordonnées de la première cellule du bloc, RzCt sont les coordonnées de la dernière.
3. Si la sélection porte sur une seule cellule, les coordonnées retournées sont de la forme "RnCm:RnCm".
4. Si toute la colonne m de la feuille de calcul comportant n lignes est sélectionnée, CALC_MULTI_SELECTION\$ retourne R1Cm:RnCm.
5. Si toute la ligne n de la feuille de calcul comportant m colonnes est sélectionnée, CALC_MULTI_SELECTION\$ retourne RnC1:RnCm.

Exemple :

```
LOCAL i%, s$
; Insérer dans la list box BLOCS l'ensemble des blocs sélectionnés

MOVE 0 TO i%
REPEAT
MOVE CALC_MULTI_SELECTION$(HCALC%, i%) TO s$
IF s$ <> ""
INSERT AT END s$ TO BLOCS
MOVE i%+1 TO i%
ENDIF
UNTIL s$ = ""
```

Voir aussi CALC_SELECTION\$, CALC_SELECT, CALC_MULTI_SELECT

Fonction CALC_NEXTNAME%

Retourne l'index correspondant à la cellule nommée suivante. La recherche est effectuée en fonction de l'index de recherche index, retourné par un appel précédent à CALC_FIRSTNAME% ou CALC_NEXTNAME%.

Syntaxe	CALC_NEXTNAME% (<i>handle Calc, index, nom assoc, coord cellule</i>)		
Paramètres	handle- Calc	INT(4)	handle fenêtre NS-Calc
	index	INT(4)	index de recherche
	nom assoc	CSTRING(255)	!V! nom logique associé
	coord cellule	CSTRING(255)	!V! coordonnées de la cellule
Valeur retournée	INT(4)	Index de la cellule nommée trouvée, utilisé pour la recherche suivante, ou 0 si la recherche est terminée.	

1. L'index retourné sera utilisé avec CALC_NEXTNAME% pour la recherche de la cellule suivante.
2. Si la recherche a abouti, le nom logique de la cellule trouvée est affecté à la variable nom assoc, ses coordonnées à la variable coord cellule.
3. Si la recherche est achevée, CALC_NEXTNAME% retourne 0 (zéro).

Exemple :

```
LOCAL CSTRING Name$(255), CSTRING Cell$(255)
LOCAL INT Index%(4)
; Rechercher le nom associé à la cellule R4C12

MOVE CALC_FIRSTNAME%(HCALC%, Name$, Cell$) TO Index%
WHILE (Index% <> 0) AND (Cell$ <> "R4C12")
MOVE CALC_NEXTNAME%(HCALC%, Index%, Name$, Cell$) TO Index%
ENDWHILE
IF Cell$ = "R4C12"
MESSAGE "Cellule R4C12", "Le nom de la cellule R4C12 est" && Name$
ELSE
MESSAGE "Erreur", "Aucun nom n'a été donné à la cellule R4C12"
ENDIF
```

Voir aussi CALC_FIRSTNAME%, CALC_FINDNAME\$, CALC_CREATENAME, CALC_DELETENAME

Fonction CALC_PRINT%

Imprime le bloc de coordonnées coord bloc, extrait de la fenêtre NS Calc spécifiée par handle Calc.

Syntaxe	CALC_PRINT% (<i>handle prn, handle Calc, coord bloc, param imp, première page, dernière page, zoom, titre, fonte, style fonte</i>)		
Paramètres	handle prn	INT(4)	handle de l'imprimante sur laquelle se fait l'impression
	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	param imp	INT(2)	paramètres d'impression (justification, impression ou non du titre, impression ou non des numéros de pages, etc.) ; ces paramètres sont une combinaison des constantes PK_ *%.
	première page	INT(2)	numéro de la première page à imprimer
	dernière page	INT(2)	numéro de la dernière page à imprimer
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression 'taille réelle').
	titre	CSTRING(255)	titre de l'impression
	fonte	CSTRING(255)	fonte utilisée (nom et taille)
	style fonte	INT(1)	attributs de la fonte (gras, italique, etc.) sous forme d'une combinaison des constantes GFS_ *%

		(librairie NSGRAPH).
Valeur retournée	INT (1)	En cas d'erreur, CALC_PRINT% retourne une valeur différente de 0.

1. handle prn est le handle retourné par la fonction REP_OPEN% (librairie NSPRN). Zéro (0) correspond à l'imprimante par défaut installée sur le système.
2. titre, fonte titre et style fonte ne sont utiles (et utilisés) que si l'impression du titre est demandée, c'est à dire si la constante PK_TITLE% se trouve dans param imp.
3. Pour imprimer la totalité des pages, spécifier zéro pour première page et dernière page.

Exemple :

```
; Imprimer les 5 premières pages du bloc sélectionné ou la totalité si ce
bloc peut être imprimé en moins de 5 pages
IF CALC_GETPRINTNBPAGES%(0, HCALC%, CALC_SELECTION$(HCALC%), PK_LEFT% BOR
PK_TITLE%, \
  100, "Impression de...", "Times New Roman, 10", GFS_BOLD%) <= 5
IF CALC_PRINT%(0, HCALC%, CALC_SELECTION$(HCALC%), PK_LEFT% BOR PK_TITLE%,
\
  0,0, 100, "Impression de...", "Times New Roman, 10", GFS_BOLD%)
MESSAGE "Erreur", "Problème lors de l'impression"
ENDIF
ELSE
IF CALC_PRINT%(0, HCALC%, CALC_SELECTION$(HCALC%), PK_LEFT% BOR PK_TITLE%,
\
  1,5, 100, "Impression de...", "Times New Roman, 10", GFS_BOLD%)
MESSAGE "Erreur", "Problème lors de l'impression"
ENDIF
ENDIF
```

Voir aussi Constantes PK *%, CALC_PRINTFILE%, CALC_GETPRINTNBPAGES%,
CALC_GETPRINTFILENBPAGES%

Fonction CALC_PRINTFILE%

Imprime le bloc de coordonnées coord bloc, extrait du fichier nom fichier.

Syntaxe	CALC_PRINTFILE% (<i>handle prn, nom fichier, coord bloc, param imp, première page, dernière page, zoom, titre, fonte, style fonte</i>)		
Paramètres	handle prn	INT(4)	handle de l'imprimante sur laquelle se fait l'impression
	nom fichier	CSTRING(255)	nom du fichier à imprimer
	coord bloc	CSTRING(255)	coordonnées du bloc
	param imp	INT(2)	paramètres d'impression (justification, impression ou non du titre, impression ou non des numéros de pages, etc.) ; ces paramètres sont une combinaison des constantes PK_ *%.
	première page	INT(2)	numéro de la première page à imprimer
	dernière page	INT(2)	numéro de la dernière page à imprimer
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression 'taille réelle')
	titre	CSTRING(255)	titre de l'impression
	fonte	CSTRING(255)	fonte utilisée (taille et nom)
	style fonte	INT(1)	attributs de la fonte (gras, italique, etc.) sous forme d'une combinaison des constantes GFS_ *% (librairie NSGRAPH).

Valeur retournée	INT (1)	En cas d'erreur ou si le format du fichier n'est pas reconnu, CALC_PRINTFILE% retourne une valeur différente de 0.
-------------------------	---------	--

1. handle prn est le handle retourné par la fonction REP_OPEN% (bibliothèque NSPRN). Zéro (0) correspond à l'imprimante par défaut installée sur le système.
2. titre, fonte titre et style fonte ne sont utiles (et utilisés) que si l'impression du titre est demandée, c'est à dire si la constante PK_TITLE% se trouve dans param imp.
3. Pour imprimer la totalité des pages, spécifier zéro pour première page et dernière page.

Exemple :

```
; Imprimer le bloc "R1C1:R10C10" du fichier ess1.cal
IF CALC_PRINTFILE%(0, "ess1.cal", "R1C1:R10C10", PK_LEFT% BOR PK_TITLE%, \
0,0, 100, "ESSAI", "Times New Roman, 10", GFS_BOLD%
MESSAGE "Erreur", "Problème d'impression"
ENDIF
```

Voir aussi Constantes PK *%, CALC_PRINT%, CALC_GETPRINTNBPAGES%,
CALC_GETPRINTFILENBPAGES%

Fonction **CALC_PRINTFINDPAGE%**

Impression de la page recherchée.

Syntaxe	CALC_PRINTFINDPAGE% (<i>hJob%</i> , <i>COL%</i> , <i>ROW%</i> , <i>NBX%</i> , <i>NBY%</i>)		
Paramètres	hJob%	INT(4)	handle du job d'impression
	COL%	INT(2)	numéro de la colonne
	ROW%	INT(2)	numéro de la ligne
	NBX%	INT(2)	nombre de page en largeur
	NBY%	INT(2)	nombre de page en hauteur
Valeur retournée	INT (1)		

Voir aussi CALC_PRINTSTARTJOB%, CALC_PRINTSTOPJOB%, CALC_PRINTPAGE%

Fonction **CALC_PRINTGETPAGECOUNT%**

Retourne le nombre total de pages imprimées.

Syntaxe	CALC_PRINTGETPAGECOUNT% (<i>hJob%</i> , <i>NBX%</i> , <i>NBY%</i>)		
Paramètres	hJob%	INT(4)	handle du job d'impression
	NBX%	INT(2)	nombre de page en largeur
	NBY%	INT(2)	nombre de page en hauteur
Valeur retournée	INT(2)	nombre de pages total	

La feuille de calcul est découpée en largeur et en hauteur en plusieurs morceaux pour être imprimée.

Voir aussi CALC_PRINTSTARTJOB%, CALC_PRINTSTOPJOB%, CALC_PRINTREPAGINATE%,
CALC_PRINTGETPAGEINFO%, CALC_PRINTFINDPAGE%, CALC_PRINTPAGE%,
CALC_PRINTNEWFRAME%, CALC_PRINTGETDISPLAYEDCELLSIZE

Fonction **CALC_PRINTGETPAGEINFO%**

Récupère des informations sur la page.

Syntaxe	CALC_PRINTGETPAGEINFO% (<i>hJob%</i> , <i>NBX%</i> , <i>NBY%</i> , <i>CALCPAGEINFO</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>NBX%</i>	INT(2)	nombre de page en largeur
	<i>NBY%</i>	INT(2)	nombre de page en hauteur
	<i>CALCPAGEINFO</i>	SEGMENT	contient des informations sur la page
Valeur retournée	INT(2)	0, si tout s'est bien déroulé.	

Voir aussi *CALCPAGEINFO*, *CALC_PRINTSTARTJOB%*, *CALC_PRINTSTOPJOB%*, *CALC_PRINTREPAGINATE%*, *CALC_PRINTGETPAGECOUNT%*, *CALC_PRINTFINDPAGE%*, *CALC_PRINTPAGE%*, *CALC_PRINTNEWFRAME%*, *CALC_PRINTGETDISPLAYEDCELLSIZE*

Fonction CALC_PRINTNEWFRAME%

Ejection d'une page.

Syntaxe	CALC_PRINTNEWFRAME% (<i>hJob%</i>)		
Paramètre	<i>hJob%</i>	INT(4)	handle du job d'impression
Valeur retournée	INTEGER		

Voir aussi CALC_PRINTSTARTJOB%, CALC_PRINTSTOPJOB%, CALC_PRINTREPAGINATE%, CALC_PRINTGETPAGECOUNT%, CALC_PRINTGETPAGEINFO%, CALC_PRINTFINDPAGE%, CALC_PRINTPAGE%, CALC_PRINTGETDISPLAYEDCELLSIZE

Fonction CALC_PRINTPAGE%

Impression d'une page.

Syntaxe	CALC_PRINTPAGE% (<i>hJob%</i> , <i>NBX%</i> , <i>NBY%</i> , <i>fOPTS%</i> , <i>szPARAMS\$</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>NBX%</i>	INT(2)	nombre de page en largeur
	<i>NBY%</i>	INT(2)	nombre de page en hauteur
	<i>fOPTS%</i>	INT(4)	combinaison de constantes PK_*%
	<i>szPARAMS\$</i>	CSTRING	Paramètres optionnels d'impression. Peut être mis à 0.
Valeur retournée	INTEGER		

Le paramètre *szPARAMS\$* peut prendre les valeurs suivantes :

Valeur	Définition	Exemple
BLOCK	bloc de la feuille à imprimer	" /BLOCK:R2C2:R10C12 "
ZOOM	facteur d'agrandissement/réduction (par défaut 100)	ZOOM:100
PAGENB	numéro de la page (pour remplacer la variable @PAGE)	" /PAGENB:10 "
HDRTEXT	Texte d'en-tête	
HDRFONT	Police de l'en-tête	" /HDRFONT :Courier,10"
HDRSELS	attributs de la police (valeur entière)	" /HDRSELS:8" pour mettre en gras
Une combinaison de constantes GFS_*% de la librairie NsGraph		

Voir aussi Constantes *PK_*%*, *CALC_PRINTSTARTJOB%*, *CALC_PRINTSTOPJOB%*, *CALC_PRINTREPAGINATE%*, *CALC_PRINTGETPAGECOUNT%*, *CALC_PRINTGETPAGEINFO%*, *CALC_PRINTFINDPAGE%*, *CALC_PRINTNEWFRAME%*, *CALC_PRINTGETDISPLAYEDCELLSIZE*

Fonction CALC_PRINTREPAGINATE%

Recalcule des pages avec de nouvelles options ou après changement du contenu, des marges...

Syntaxe	CALC_PRINTREPAGINATE% (<i>hJob%</i> , <i>fOPTS%</i> , <i>szPARAMS\$</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>fOPTS%</i>	INT(4)	combinaison de constantes PK_*
	<i>szPARAMS\$</i>	CSTRING	Paramètres optionnels d'impression. Peut être mis à 0.
Valeur retournée	INT(2)		

Le paramètre *szPARAMS\$* peut prendre les valeurs suivantes :

Valeur	Définition	Exemple
BLOCK	bloc de la feuille à imprimer	" /BLOCK:R2C2:R10C12 "
ZOOM	facteur d'agrandissement/réduction (par défaut 100)	ZOOM:100
PAGENB	numéro de la page (pour remplacer la variable @PAGE)	" /PAGENB:10 "
HDRTEXT	Texte d'en-tête	
HDRFONT	Police de l'en-tête	" /HDRFONT :Courier,10"
HDRSELS	attributs de la police (valeur entière)	" /HDRSELS:8" pour mettre en gras
Une combinaison de constantes GFS_*% de la librairie NsGraph		

Voir aussi les constantes PK *, CALC_PRINTSTOPJOB%, CALC_PRINTSTARTJOB%, CALC_PRINTGETPAGECOUNT%, CALC_PRINTGETPAGEINFO%, CALC_PRINTFINDPAGE%, CALC_PRINTPAGE%, CALC_PRINTNEWFRAME%, CALC_PRINTGETDISPLAYEDCELLSIZE

Fonction CALC_PRINTSTARTJOB%

Démarre le job d'impression.

Syntaxe	CALC_PRINTSTARTJOB% (<i>hJob%</i> , <i>handle%</i> , <i>prn%</i> , <i>szTITLE\$</i> , <i>fOPTS%</i> , <i>szPARAMS\$</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>handle%</i>	INT(4)	handle fenêtre NS-Calc
	<i>prn%</i>	POINTER	handle de l'imprimante
	<i>szTITLE\$</i>	CSTRING	titre du job d'impression affiché par le Gestionnaire d'Impression.
	<i>fOPTS%</i>	INT(4)	combinaison de constantes PK_*%
	<i>szPARAMS\$</i>	CSTRING	Paramètres optionnels d'impression. Peut être mis à 0.
Valeur retournée	INT(2)		

Le paramètre *szPARAMS\$* peut prendre les valeurs suivantes :

Valeur	Définition	Exemple
BLOCK	bloc de la feuille à imprimer	" /BLOCK:R2C2:R10C12 "
ZOOM	facteur d'agrandissement/réduction (par défaut 100)	ZOOM:100
PAGENB	numéro de la page (pour remplacer la variable @PAGE)	" /PAGENB:10 "
HDRTEXT	Texte d'en-tête	
HDRFONT	Police de l'en-tête	" /HDRFONT :Courier,10"
HDRSELS	attributs de la police (valeur entière)	" /HDRSELS:8" pour mettre en gras
Une combinaison de constantes GFS_*% de la librairie NsGraph		

Voir aussi les constantes PK_*%, CALC_PRINTSTOPJOB%, CALC_PRINTREPAGINATE%, CALC_PRINTGETPAGECOUNT%, CALC_PRINTGETPAGEINFO%, CALC_PRINTFINDPAGE%, CALC_PRINTPAGE%, CALC_PRINTNEWFRAME%, CALC_PRINTGETDISPLAYEDCELLSIZE

Fonction CALC_PRINTSTOPJOB%

Désactive l'impression.

Syntaxe	CALC_PRINTSTOPJOB% (<i>hJob%</i>)		
Paramètre	<i>hJob%</i>	INT(4)	handle du job d'impression
Valeur retournée	INT(2)		

Voir aussi CALC_PRINTSTARTJOB%, CALC_PRINTREPAGINATE%, CALC_PRINTGETPAGECOUNT%,
CALC_PRINTGETPAGEINFO%, CALC_PRINTFINDPAGE%, CALC_PRINTPAGE%,
CALC_PRINTNEWFRAME%, CALC_PRINTGETDISPLAYEDCELLSIZE

Fonction CALC_PRINTXPRN2SCR%

Permet de convertir des valeurs de l'unité utilisée pour l'impression en unité écran.

Syntaxe	CALC_PRINTXPRN2SCR% (<i>hJob%</i> , <i>handle%</i> , <i>X%</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>X%</i>	INT(4)	une valeur
Valeur retournée	INT(1)		

Voir aussi CALC_PRINTXSCR2PRN%, CALC_PRINTYSCR2PRN%, CALC_PRINTYPRN2SCR%, **constantes** CCALC_*

Fonction **CALC_PRINTXSCR2PRN%**

Permet de convertir des valeurs en unité écran dans l'unité utilisée pour l'impression.

Syntaxe	CALC_PRINTXSCR2PRN% (<i>hJob%</i> , <i>handle%</i> , <i>X%</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>X%</i>	INT(4)	une valeur
Valeur retournée	INT(1)		

Voir aussi CALC_PRINTYSCR2PRN%, CALC_PRINTXPRN2SCR%, CALC_PRINTYPRN2SCR%, **constantes**
CCALC *%

Fonction CALC_PRINTYPRN2SCR%

Permet de convertir des valeurs de l'unité utilisée pour l'impression en unité écran.

Syntaxe	CALC_PRINTYPRN2SCR% (<i>hJob%</i> , <i>handle%</i> , <i>X%</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>X%</i>	INT(4)	une valeur
Valeur retournée	INT(1)		

Voir aussi CALC_PRINTXSCR2PRN%, CALC_PRINTYSCR2PRN%, CALC_PRINTXPRN2SCR%, **constantes**
CCALC_*

Fonction **CALC_PRINTYSCR2PRN%**

Permet de convertir des valeurs en unité écran dans l'unité utilisée pour l'impression.

Syntaxe	CALC_PRINTYSCR2PRN% (<i>hJob%</i> , <i>handle%</i> , <i>X%</i>)		
Paramètres	<i>hJob%</i>	INT(4)	handle du job d'impression
	<i>X%</i>	INT(4)	une valeur
Valeur retournée	INT(1)		

Voir aussi CALC_PRINTXSCR2PRN%, CALC_PRINTXPRN2SCR%, CALC_PRINTYPRN2SCR%, **constantes** **CCALC_***%

Fonction **CALC_ROW%**

Retourne le numéro de ligne de la cellule active ou de la cellule initiale du bloc sélectionné (la cellule initiale d'un bloc est celle qui n'est pas en vidéo inverse dans un angle du bloc).

Syntaxe	CALC_ROW% (<i>handle Calc</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	INT(2)		

Exemple :

```
; Déplacer la cellule active de trois lignes vers le bas, en restant dans  
la même colonne  
CALC_GOTO HCALC%, CALC_ROW%(HCALC%) + 3, CALC_COLUMN%(HCALC%)
```

Voir aussi [CALC_COLUMN%](#), [CALC_SELECTION\\$](#), [CALC_RC2XY](#)

Fonction CALC_SAVE%

Sauvegarde le contenu de la feuille de calcul dans le fichier dont le nom est passé dans nom fichier.

Syntaxe	CALC_SAVE% (<i>handle Calc, nom fichier, format</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nom fichier	CSTRING(255)	nom du fichier à sauvegarder
	format	INT(1)	format du fichier
Valeur retournée	INT(1)	TRUE% si la sauvegarde s'est bien déroulée. FALSE% en cas de problème lors de la sauvegarde.	

- format spécifie le format dans lequel doit s'effectuer la sauvegarde :
 - format NS Calc (constantes FF_NSCALC*) : forme la plus complète de sauvegarde, intégrant les largeurs/hauteurs de cellules, les formats, etc.,
 - format texte (constantes FF_TEXT*) : forme compacte, ne retenant que la valeur des cellules : le caractère de tabulation CHR\$(9) sert de séparateur pour les cellules d'une même ligne, la séquence fin de ligne CHR\$(13) sert de séparateur pour les lignes.
- Pour chacun des deux formats précédents, il est possible de spécifier le jeu de caractères utilisé :
 - ASCII (PC) : jeu de caractères sous OS/2 et MS DOS,
 - ANSI : jeu de caractères sous Windows,
 - MAC : jeu de caractères sous Macintosh.
- "CALC_SAVE%(HCALC%, F\$, FF_NSCALC%)" est équivalent à "SAVE HCALC% TO F\$" sans indication sur la sauvegarde, le fichier F\$ étant donc sauvegardé au format NS Calc.

Exemple :

```
LOCAL CSTRING Nom$ (255)
; Sauvegarde de la feuille de calcul dans le fichier RESU_92 sur la
disquette A

MOVE "A:\RESU_92" TO Nom$
IF NOT CALC_SAVE% (HCALC%, Nom$, FF_NSCALC_ANSI%)
MESSAGE "Erreur !", "Fichier" && Nom$ && "non sauvegardé"
ELSE
MESSAGE "OK !", Nom$ && "sauvegardé"
ENDIF
```

Voir aussi CALC_LOAD%, Constantes FF *%, SAVE (Manuel de Programmation NCL)

Fonction CALC_SELECTION\$

Retourne les coordonnées du bloc sélectionné.

Syntaxe	CALC_SELECTION\$ (<i>handle Calc</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
Valeur retournée	CSTRING (255)		

1. Les coordonnées du bloc sont retournées dans une chaîne de caractères sous la forme "RnCm:RzCt"
2. RnCm sont les coordonnées de la première cellule du bloc, RzCt sont les coordonnées de la dernière.
3. Si la sélection porte sur une seule cellule, les coordonnées retournées seront de la forme "RnCm:RnCm".
4. Si toute la colonne m de la feuille de calcul comportant n lignes est sélectionnée, CALC_SELECTION\$ retourne : R1Cm:RnCm.
5. Si toute la ligne n de la feuille de calcul comportant m colonnes est sélectionnée, CALC_SELECTION\$ retourne : RnC1:RnCm.

Exemple :

```
; Effacer le bloc sélectionné
CALC_CLEAR HCALC%, CALC_SELECTION$(HCALC%)
```

Voir aussi CALC_COLUMN%, CALC_ROW%

Fonction CALC_SETBLOCKFORMATID%

Permet de définir le format d'un bloc de cellules.

Syntaxe	CALC_SETBLOCKFORMATID% (<i>handle-Calc, block\$, id%</i>)		
Paramètres	handle- Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING(255)	coordonnées du bloc
	id%	INT(2)	une des constantes CCALC_FMT_*
Valeur retournée	INTEGER		

Voir aussi CALC_GETFORMATIDATTR%, CALC_SETFORMATIDATTR%, CALC_CREATEFORMATID%, CALC_GETCELLFORMATID%, CALC_GETDEFFORMATID%, CCALC_FMT_*

Fonction **CALC_SETFORMATIDATTR%**

Permet de définir un format.

Syntaxe	CALC_SETFORMATIDATTR% (<i>handle-Calc, id%, idattr%, pvalue%</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	id%	INT(2)	identifiant de la feuille de calcul
	idattr%	POINTER	une des constantes CCALC_FMT_ *%
	pvalue%	POINTER	un pointeur sur une variable dont le type est indiqué dans NSCALC.NCL en commentaire à droite de la constante CCALC_FMT_ *%
Valeur retournée	INTEGER		

Voir aussi CALC_GETFORMATIDATTR%, CALC_CREATEFORMATID%, CALC_SETBLOCKFORMATID%, CALC_GETCELLFORMATID%, CALC_GETDEFFORMATID%, CCALC_FMT_ *%

Fonction **CALC_SETONERROR%**

Permet de personnaliser le message d'erreur affiché dans une cellule.

Syntaxe	CALC_SETONERROR% (<i>handle-Calc</i> , <i>errtype%</i> , <i>RPLTYPE%</i> , <i>RPLMSG\$</i>)		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	errtype%	INT(4)	constantes CE_*%
	RPLTYPE%	INT(4)	-1 : utiliser le message par défaut 0 : spécifier le message donné par RPLMSG\$
	RPLMSG\$	INT(4)	message d'erreur
Valeur retournée	INTEGER		

Voir aussi Constantes CE_*%, CALC_GETONERROR%

Instructions Ns-Calc

Instruction **CALC_ADDENHANCEDLINK**

Trace un trait entre la cellule de coordonnées num ligne src, num colonne src et la cellule de coordonnées num ligne dest, num colonne dest, en appliquant à ce trait les différentes caractéristiques spécifiées par style lien, segment de type LINKSTYLE.

Syntaxe	CALC_ADDENHANCEDLINK <i>handle Calc, num ligne src, num colonne src, num ligne dest, num colonne dest, style lien</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	num ligne src	INT(2)	ligne source
	num colonne src	INT(2)	colonne source
	num ligne dest	INT(2)	ligne destination
	num colonne dest	INT(2)	colonne destination
	style lien	LINKSTYLE	!V! caractéristiques du lien

1. Pour que ce trait soit visible, la fenêtre doit être re-dessinée (par exécution de l'événement PAINT, qui peut être provoqué par l'instruction INVALIDATE).
2. Cette instruction est très similaire à l'instruction CALC_ADDLINK, excepté que cette dernière ne permet pas de spécifier l'aspect, la couleur et l'emplacement du trait.

Exemple :

```
LOCAL LINKSTYLE Style_Lien

MOVE CLP_VERTICAL% TO Style_Lien.STARTKIND
MOVE CLP_VERTICAL% TO Style_Lien.ENDKIND
MOVE GLK_SOLID% TO Style_Lien.LINEKIND
MOVE COL_CYAN% TO Style_Lien.LINECOLOR
MOVE CLS_BROKEN% TO Style_Lien.LINKKIND
MOVE 2 TO Style_Lien.LINEWIDTH

CALC_ADDENHANCEDLINK HCALC%, 2, 2, 5, 4, Style_Lien
```

Voir aussi Segment [LINKSTYLE](#), [CALC_ADDLINK](#), [CALC_REMLINK](#)

Instruction **CALC_ADDLINK**

Trace un trait entre la cellule de coordonnées num ligne src, num colonne src et la cellule de coordonnées num ligne dest, num colonne dest, en appliquant à ce trait le type et la largeur spécifiés par type et largeur.

Syntaxe	CALC_ADDLINK <i>handle Calc, num ligne src, num colonne src, num ligne dest, num colonne dest, type, largeur</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	num ligne src	INT(2)	ligne source
	num colonne src	INT(2)	colonne source
	num ligne dest	INT(2)	ligne destination
	num colonne dest	INT(2)	colonne destination
	type	INT(1)	type du trait : constantes GLK_ *% (librairie NSGRAPH)
	largeur	INT(1)	largeur du trait (compris entre 1 et 3)

Pour que ce trait soit visible, la fenêtre doit être re-dessinée (par exécution de l'événement PAINT, qui peut être provoqué par l'instruction INVALIDATE).

Exemple :

```
; Tracer un trait entre les cellules R1C1 et R3C4
CALC_ADDLINK HCALC%, 1, 1, 3, 4, GLK_DOT%, 3
```

Voir aussi CALC_REMLINK, CALC_ISLINK%, CALC_ADDENHANCEDLINK

Instruction **CALC_CLEAR**

Efface le contenu des cellules contenues dans le(s) bloc(s) coord bloc.

Syntaxe	CALC_CLEAR <i>handle Calc, coord bloc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées des blocs

Si le bloc n'est pas correctement défini, par exemple "R3C3:R2C2", l'instruction CALC_CLEAR n'a aucun effet.

Exemple :

```
; Effacer simultanément le bloc R1C2:R3C3, le bloc R5C1:R6C4, la cellule R8C6
CALC_CLEAR HCALC%, "R1C2:R3C3; R5C1:R6C4; R8C6"

; Effacer simultanément la rangée de cellules R2et la colonne de cellules C3
CALC_CLEAR HCALC%, "R2; C3"
```

Voir aussi CALC_CLEARSPECIAL, CALC_PUT, CALC_COPY

Instruction **CALC_CLEAR****SPECIAL**

Similaire à **CALC_CLEAR**, **CALC_CLEAR****SPECIAL** efface une partie du contenu des cellules indiquées par coord bloc.

Syntaxe	CALC_CLEAR SPECIAL <i>handle Calc, coord bloc, type</i>		
Paramètres	handle- Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées des blocs
	type	INT(2)	type de suppression (format, valeur, formule...). Il est défini par une combinaison des constantes PT_* .

Si le bloc n'est pas correctement défini, par exemple "R3C3:R2C2", l'instruction **CALC_CLEAR****SPECIAL** n'a aucun effet.

Exemple :

```
; Effacer les formules de la rangée de cellules R2 et de la colonne de  
cellules C3  
CALC_CLEARSPECIAL HCALC_*, "R2; C3", PT_FORMULA_*
```

Voir aussi Constantes **PT_***, **CALC_CLEAR**, **CALC_COPY**, **CALC_PASTE**, **CALC_PUT**

Instruction **CALC_COPY**

Copie le contenu de la cellule active ou de la sélection courante dans le presse papiers (clipboard).

Syntaxe	CALC_COPY <i>handle Calc</i>		
Paramètre	handle-Calc	INT(4)	handle fenêtre NS Calc

1. Combinée avec CALC_PASTE, cette instruction permet de dupliquer le contenu d'une feuille de calcul dans un document ou dans une autre fenêtre NS Calc.
2. Le format de la sélection est également copié.
3. La fonction CALC_ISCLIPBOARD% retournera TRUE% après cette instruction, même si une seule cellule vide a été copiée dans le presse papiers.

Exemple :

```
; Placer dans le presse papiers le contenu des cellules sélectionnées  
CALC_COPY HCALC%
```

Voir aussi CALC_ISCLIPBOARD%, CALC_PASTE, CALC_PASTESPECIAL

Instruction **CALC_CREATENAME**

Associe le nom logique nom assoc à la cellule ayant pour coordonnées coord cellule.

Syntaxe	CALC_CREATENAME <i>handle Calc, nom assoc, coord cellule</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	nom assoc	CSTRING(255)	nom logique associé
	coord cellule	CSTRING(255)	coordonnée de la cellule

1. L'affectation d'un nom logique à une cellule permet à l'utilisateur final de simplifier l'accès ultérieur à cette cellule (passage du paramètre nom assoc au lieu des coordonnées).
2. Les noms logiques des cellules sont 'case sensitive', la différence minuscules/majuscules est prise en compte. Pour éviter des incohérences, il est donc préférable de toujours définir ces noms en majuscules.
3. Le paramètre coord cellule doit être de la forme RnCm. Si celui ci est erroné (par exemple définition d'un bloc R1C1:R2C2), le nom logique sera affecté à la cellule R0C0.

Exemple :

```
; Affectation du nom logique SOMME à la cellule R5C3
CALC_CREATENAME HCALC%, "SOMME", "R5C3"
...
; Déplacement de la cellule active vers la cellule nommée
LOCAL R%(2), C%(2), CELL$

MOVE CALC_FINDNAME$ (HCALC%, "SOMME") TO CELL$
CALC_RC2XY HCALC%, CELL$, C%, R%
CALC_GOTO HCALC%, R%, C%
```

Voir aussi CALC_DELETENAME, CALC_FINDNAME\$, CALC_FIRSTNAME%, CALC_NEXTNAME%

Instruction **CALC_DELETENAME**

Supprime le nom logique nom assoc précédemment lié à une cellule par l'instruction CALC_CREATENAME.

Syntaxe	CALC_DELETENAME <i>handle Calc, nom assoc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	nom assoc	CSTRING(255)	nom logique associé

Cette instruction n'a aucun effet si nom assoc ne correspond à aucun nom logique associé à la feuille de calcul.

Exemple :

```
; Suppression du nom SOMME précédemment affecté à une cellule  
CALC_DELETENAME HCALC%, "SOMME"
```

Voir aussi CALC_CREATENAME, CALC_FINDNAME\$, CALC_FIRSTNAME%, CALC_NEXTNAME%

Instruction **CALC_DISPLAYFORMULA**

Active ou inhibe l'affichage des formules de calcul dans les cellules :

- Si indic affichage vaut FALSE%, le résultat des formules est affiché dans les cellules.
- Si indic affichage vaut TRUE%, les formules sont affichées.

Syntaxe	CALC_DISPLAYFORMULA <i>handle Calc, indic affichage</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	indic affichage	INT(1)	indicateur d'affichage

Exemple :

```
; Inversion de l'affichage formules/résultats sur le menu correspondant
IF CALC_ISFORMULADISPLAYED% (HCALC%)
CALC_DISPLAYFORMULA HCALC%, FALSE%
ELSE
CALC_DISPLAYFORMULA HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISFORMULADISPLAYED%**

Instruction **CALC_DISPLAYZERO**

Active ou inhibe l'affichage dans les cellules des valeurs numériques égales à zéro :

- Si indic affichage vaut FALSE%, les valeurs nulles ne sont pas affichées.
- Si indic affichage vaut TRUE%, les valeurs nulles sont affichées.

Syntaxe	CALC_DISPLAYZERO <i>handle Calc, indic affichage</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	indic affichage	INT(1)	indicateur d'affichage

Exemple :

```
; Inversion de l'affichage des valeurs nulles sur le menu correspondant
IF CALC_ISZERODISPLAYED% (HCALC%)
CALC_DISPLAYZERO HCALC%, FALSE%
ELSE
CALC_DISPLAYZERO HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISZERODISPLAYED%**

Instruction **CALC_DRAW**

Affiche un bloc de la feuille de calcul dans un "Presentation Space" (fenêtre ou imprimante).

Syntaxe	CALC_DRAW <i>handle Calc, handle ps, position horiz, position vert, num prem ligne, num prem colonne, num dern ligne, num dern colonne, imprimante, cadre, zoom</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	handle ps	INT(4)	handle d'un Presentation Space
	position horiz	INT(2)	position horizontale de départ dans l'espace d'affichage
	position vert	INT(2)	position verticale de départ dans l'espace d'affichage
	num prem ligne	INT(2)	première ligne du bloc à afficher
	num prem colonne	INT(2)	première colonne du bloc à afficher
	num dern ligne	INT(2)	dernière ligne du bloc à afficher
	num dern colonne	INT(2)	dernière colonne du bloc à afficher
	imprimante	INT(1)	indicateur imprimante (TRUE%) / écran (FALSE%)
	grille	INT(1)	indicateur d'affichage de la grille (présence TRUE% / absence FALSE%)
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression ou un affichage 'taille réelle').

Exemple :

```

LOCAL Largeur%(2), Hauteur%(2)

CALC_GETDRAWSIZE HCalc%, H_PS%,
1, 1, 10, 10,
FALSE%, 100, Largeur%, Hauteur%

; Modifier la taille du tableau en faisant évoluer le paramètre Zoom
IF (Largeur% > 100) OR (Hauteur% > 100)
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 80
ELSEIF (Largeur% < 50) OR (Hauteur% < 50)
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 150
ELSE
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 100
ENDIF

```

Voir aussi **CALC_GETDRAWSIZE**

Instruction CALC_EXTDRAW

Affiche sur un PS (écran ou imprimante) un groupe de cellule.

Syntaxe	CALC_EXTDRAW <i>handle Calc, PS%, x%, y% ,r1%, c1%, r2%, c2%, kinds%, grid%, zoom%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	PS%	INT	Presentation Space (imprimante ou écran)
	x%	INT(2)	destination coordonnée en X coin inférieur gauche
	y%	INT(2)	destination coordonnée en Y coin inférieur gauche
	r1%	INT(2)	numéro de la première ligne
	c1%	INT(2)	numéro de la première colonne
	r2%	INT(2)	numéro de la seconde ligne
	c2%	INT(2)	numéro de la seconde colonne
	kinds%	INT(2)	0 : utilise les polices
	grid%	INT(1)	TRUE% affiche la grille
	zoom%	INT(2)	Facteur de zoom

1. Toutes les fonctions et instructions graphiques peuvent s'effectuer aussi bien sur la fenêtre courante que sur une autre fenêtre, ou même sur une imprimante. Pour indiquer sur lequel de ces supports on veut effectuer une opération graphique, il faut préciser le "Presentation Space" permettant d'identifier de façon unique le support. Ce Presentation Space (ou PS en abrégé) est un entier. En plus du support, le Presentation Space mémorise les attributs graphiques :

- a) Couleurs
- b) Mix Modes

- c) Pattern
- d) Type de Ligne
- e) Marker
- f) Font

ainsi que la position courante du pointeur graphique.

2. Pour créer un PS, utiliser la fonction GPI_OPENRES% de la librairie NSGraph.

Voir aussi CALC_GETDRAWSIZE, CALC_DRAW

Instruction **CALC_FILLDOWN**

Duplique le contenu des cellules de la première ligne du bloc dans les lignes suivantes du bloc.

Syntaxe	CALC_FILLDOWN <i>handle Calc, coord bloc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc

Si les cellules de la première ligne du bloc sont vides, CALC_FILLDOWN n'a aucun effet.

Exemple :

```
; Copier la cellule R3C3 dans les cellules R4C3 et R5C3,
; R3C4 R4C4 et R5C4
; ...
; R3C8 R4C8 et R5C8
CALC_FILLDOWN HCALC%, "R3C3:R5C8"
```

Voir aussi [CALC_FILLRIGHT](#), [CALC_FILLUP](#), [CALC_FILLEFT](#)

Instruction **CALC_FILLEFT**

Duplique le contenu des cellules de la colonne de gauche du bloc dans les autres colonnes du bloc.

Syntaxe	CALC_FILLEFT <i>handle Calc, coord bloc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc

Voir aussi CALC_FILLRIGHT, CALC_FILLDOWN, CALC_FILLUP

Instruction **CALC_FILLRIGHT**

Duplique le contenu des cellules de la colonne de droite du bloc dans les autres colonnes du bloc.

Syntaxe	CALC_FILLRIGHT <i>handle Calc, coord bloc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc

Si les cellules de la première ligne du bloc sont vides, CALC_RIGHT n'a aucun effet.

Exemple :

```
; Copier la cellule R3C3 dans les cellules R3C4 à R3C8
; R4C3 R4C4 à R4C8
; R5C3 R5C4 à R5C8
CALC_FILLRIGHT HCALC%, "R3C3:R5C8"
```

Voir aussi [CALC_FILLDOWN](#), [CALC_FILLUP](#), [CALC_FILLDOWN](#)

Instruction **CALC_FILLUP**

Duplique le contenu des cellules de la dernière ligne du bloc dans les lignes précédentes du bloc.

Syntaxe	CALC_FILLUP <i>handle Calc, coord bloc</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc

Si les cellules de la dernière ligne du bloc sont vides, CALC_FILLUP n'a aucun effet.

Voir aussi CALC_FILLEFT, CALC_FILLRIGHT, CALC_FILLDOWN

Instruction **CALC_GETCELLBORDER**

Retourne dans style bordure, segment de type BORDERSTYLE, les caractéristiques de la bordure de la cellule num ligne, num colonne.

Ces caractéristiques concernent, pour chaque bord, le type de trait, sa taille et sa couleur.

Par défaut, le type de trait est GLK_INVISIBLE%, la taille est 1 et la couleur est COL_NEUTRAL%.

Syntaxe	CALC_GETCELLBORDER <i>handle Calc, num colonne, num ligne, style bordure</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
	style bordure	BORDERSTYLE	!V! caractéristiques de la bordure

Exemple :

```

LOCAL BORDERSTYLE Style_Bordure
; Ne modifier que le style de trait des bords de la cellule R2C3
CALC_GETCELLBORDER HCALC%, 3, 2, Style_Bordure

MOVE GLK_DOUBLEDOT% TO Style_Bordure.LeftLineKind
MOVE GLK_DOUBLEDOT% TO Style_Bordure.RightLineKind
MOVE GLK_DOT% TO Style_Bordure.TopLineKind
MOVE GLK_DOT% TO Style_Bordure.BottomLineKind

CALC_SETBLOCKBORDER HCALC%, "R2C3", Style_Bordure

```

Voir aussi Segment **BORDERSTYLE**, **CALC_SETBLOCKBORDER**, **CALC_GETFORMAT**

Instruction **CALC_GETCELLCOMMENT**

Permet de lire des données (binaire, pas obligatoirement du texte) ou "commentaire" associé à une cellule et enregistré par **CALC_SETBLOCKCOMMENT**.

Syntaxe	CALC_GETCELLCOMMENT <i>handle Calc, x%, y%, size%, buff%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de la colonne
	y%	INT(2)	numéro de la ligne
	size%	INT(4)	taille des données retournées
	buff%	INT(4)	pointer sur un buffer où seront retourné les données.

1. Si le paramètre buff%=0, alors la taille des données seule est retournée dans size%. Cela permet de rappeler cette instruction avec une taille de buffer suffisante pour recevoir les octets de données.

2. Notez que NS-Calc alloue dynamiquement la zone mémoire nécessaire pour les données fournies.

Voir aussi **CALC_SETBLOCKCOMMENT**

Instruction CALC_GETCELLEDITCTRL

Retourne le type de contrôle utilisé dans la ligne de statut pour la saisie de la cellule de coordonnées num colonne, num ligne.

type contrôle est l'une des constantes EDTCTRL_*% : EDTCTRL_EF% si la saisie s'effectue dans un champ de saisie normal, ou EDTCTRL_CBE% si la saisie s'effectue à l'aide d'une combo box avec champ de saisie.

Dans le cas d'une combo box, données un tableau de 4 entiers, qui est passé par adresse à l'instruction, et dont la taille est spécifiée par taille données. Les informations concernant la liste affectée au contrôle sont retournées dans les éléments de ce tableau :

- le premier élément du tableau données est le type des données : CTRLDATA_ARRAY%,
- le deuxième élément est le nombre de lignes (chaînes) composant la liste de la combo box,
- le troisième élément est la taille maximale + 1 des chaînes de cette liste,
- le quatrième élément pointe sur la zone mémoire contenant les chaînes.

Si les données n'ont pas besoin d'être récupérées, passer 0 à chacun des deux derniers paramètres.

Syntaxe	CALC_GETCELLEDITCTRL <i>handle Calc, num colonne, num ligne, type contrôle, données, taille données</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
	type contrôle	INT(2)	!V! type du contrôle de saisie
	données	INT(4)	!V! données affectées au contrôle
	taille données	INT(4)	!V! taille du buffer à affecter

Exemple :

```

LOCAL Type%(2)
LOCAL Data%[4]
LOCAL Fichier$
; Associer une CBE pour la saisie de la cellule R2C3
CALC_GETCELLEDITCTRL HCALC%, 3, 2, Type%, 0,0

IF Type% <> EDTCTRL_CBE%
MOVE "COLORS.TXT" TO Fichier$

MOVE CTRLDATA_FILE% TO Data%[0]
MOVE @Fichier$ TO Data%[1]
```

```
CALC_SETBLOCKEDITCTRL HCALC%, "R2C3", EDTCTRL_CBE%, @Data%  
ENDIF
```

Voir aussi Constantes CTRLDATA *%, EDTCTRL *%, CALC SETBLOCKEDITCTRL

Instruction **CALC_GETCELLEXTBORDER**

Permet de créer un encadrement de cellules en spécifiant la couleur en mode RGB.

Syntaxe	CALC_GETCELLEXTBORDER <i>handle Calc, x%, y%, EXTBORDERSTYLE</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de la colonne
	y%	INT(2)	numéro de la ligne
	EXTBORDERSTYLE	SEGMENT	style de bordure des cellules

1. Le segment EXTBORDERSTYLE permet notamment de sélectionner le côté de la ligne (à gauche, à droite, en haut, ou en bas) qui contiendra le trait de bordure. Il utilise également le segment BORDERLINE. La déclaration interne du segment EXTBORDERSTYLE est :

```
SEGMENT EXTBORDERSTYLE
  BORDERLINE LINE[4] ; 0: Left, 1: Right, 2: Top, 3: Bottom
ENDSEGMENT
```

2. Le segment BORDERLINE permet de définir les options de style, de largeur et de couleur de bordure souhaitées. La déclaration interne du segment BORDERLINE est :

```
SEGMENT BORDERLINE
  INT KIND(2) ; Type de ligne.
  INT WIDTH(2) ; Epaisseur.
  INT RED(2) ; Couleur en mode RGB.
  INT GREEN(2)
  INT BLUE(2)
ENDSEGMENT
```

Voir aussi **CALC_SETBLOCKEXTBORDER**

Instruction **CALC_GETCELLFONT**

Retourne dans nom fonte et style fonte les caractéristiques de la fonte utilisée dans la cellule de coordonnées num ligne, num colonne.

nom fonte retourne le nom et la taille de la fonte, sous la forme d'une chaîne de caractères, les deux informations étant séparées par une virgule.

style fonte est une combinaison des constantes GFS_*(librairie NSGRAPH) indiquant les attributs de caractères (gras, italique, etc.).

Syntaxe	CALC_GETCELLFONT <i>handle Calc, num colonne, num ligne, nom fonte, style fonte</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
	nom fonte	CSTRING(255)	!V! nom et taille de la fonte
	style fonte	INT(1)	!V! style de la fonte

Exemple :

```
LOCAL Nom_fonte$, Style_fonte%(1)
; Inversion de l'attribut 'italique' pour les caractères de la cellule R2C3

CALC_GETCELLFONT HCALC%, 3, 2, Nom_fonte$, Style_fonte%
IF GFS_ITALIC% BAND Style_fonte%
MOVE Style_fonte% GFS_ITALIC% TO Style_fonte%
ELSE
MOVE Style_fonte% + GFS_ITALIC% TO Style_fonte%
ENDIF
CALC_SETBLOCKFONT HCALC%, "R2C3", Nom_fonte$, Style_fonte%
```

Voir aussi [CALC_SETBLOCKFONT](#), [CALC_GETFORMAT](#)

Instruction **CALC_GETCELLRGBCOLOR**

Retourne les niveaux de couleur RGB de la cellule de coordonnées x%, y%.

Syntaxe	CALC_GETCELLRGBCOLOR <i>handle Calc, x%, y%, r%, g%, b%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	x%	INT(2)	numéro de la colonne
	y%	INT(2)	numéro de la ligne
	r%	INT(2)	niveau de la couleur rouge comprise entre 0 et 255.
	g%	INT(2)	niveau de la couleur verte comprise entre 0 et 255.
	b%	INT(2)	niveau de la couleur bleue comprise entre 0 et 255.

Voir aussi **CALC_SETBLOCKRGBCOLOR**

Instruction **CALC_GETDEFAULTHEADMODE**

Retourne le mode d'affichage et la chaîne de caractères.

Syntaxe	CALC_GETDEFAULTHEADMODE		
Paramètres	mode%	INT(4)	mode d'affichage
	head\$	CSTRING	chaîne de caractères

Commentaires

1. Deux modes d'affichage sont disponibles : numérique et alphanumérique avec les constantes suivantes :
 - a) HMODE_RC% 0 exemple: R1C1
 - b) HMODE_ALPHA% 1 exemple: A1 pour R1C1
2. Si le mode numérique est utilisé, la chaîne de caractère head\$ contient les deux caractères utilisés, par exemple « RC » (mode par défaut).

Voir aussi CALC_SETDEFAULTHEADMODE

Instruction **CALC_GETDISPLAYEDCELLSIZE**

Permet de récupérer la taille du texte affiché dans une cellule.

Syntaxe	CALC_GETDISPLAYEDCELLSIZE <i>handle%, row%, TY%, W%, H%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	rows%	INT	nombre de lignes utilisées
	TY%	INT(2)	une constante CTEXT*%
	W%	INT(2)	longueur du texte
	H%	INT(2)	hauteur du texte

1. En connaissant la taille de la cellule, ça permet de vérifier que le texte rentre bien dans la cellule.
2. Le paramètre TY% permet de sélectionner le type de contenu affiché :

```
CTEXTDSPTYPE% 0 Texte
CTEXTOVERRUNDSPTYPE% 1 Texte + overrun
CNUMDSPTYPE% 2 Numérique
CBMPDSPTYPE% 3 Bitmap
```

Voir aussi **CALC_PRINTGETDISPLAYEDCELLSIZE**

Instruction CALC_GETDRAWSIZE

Détermine la taille du tableau contenant l'affichage d'un bloc de la feuille de calcul dans un "Presentation Space" (fenêtre ou imprimante). La largeur et la hauteur de ce tableau sont affectées dans les variables largeur et hauteur.

Syntaxe	CALC_GETDRAWSIZE <i>handle Calc, handle ps, num prem ligne, num prem colonne, num dern ligne, num dern colonne, imprimante, zoom, largeur, hauteur</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	handle ps	INT(4)	handle d'un presentation space
	num prem ligne	INT(2)	première ligne du bloc à afficher
	num prem colonne	INT(2)	première colonne du bloc à afficher
	num dern ligne	INT(2)	dernière ligne du bloc à afficher
	num dern colonne	INT(2)	dernière colonne du bloc à afficher
	imprimante	INT(1)	indicateur imprimante (TRUE%) / écran (FALSE%)
	zoom	INT(2)	échelle de reproduction (indiquer 100 pour une impression ou un affichage 'taille réelle')
	largeur	INT(2)	!V! largeur du tableau
	hauteur	INT(2)	!V! hauteur du tableau

Exemple :

```
LOCAL Largeur%(2), Hauteur%(2)
CALC_GETDRAWSIZE HCalc%, H_PS%, 1, 1, 10, 10, FALSE%, 100, Largeur%,
Hauteur%

; Modifier la taille du tableau en faisant évoluer le paramètre Zoom
IF (Largeur% > 100) OR (Hauteur% > 100)
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 80
ELSEIF (Largeur% < 50) OR (Hauteur% < 50)
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 150
ELSE
CALC_DRAW HCalc%, H_PS%, 1, 1, 1, 1, 10, 10, FALSE%, TRUE%, 100
ENDIF
```


Voir aussi CALC_DRAW

Instruction CALC_GETFORMAT

Retourne les caractéristiques affectées à la cellule définie par les coordonnées num ligne, num colonne.

Syntaxe	CALC_GETFORMAT <i>handle Calc, num colonne, num ligne, col texte, col fond, cadre, fonte, justif, verrou, format, filtre, type, long sais</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonne	INT(2)	numéro de la colonne
	num ligne	INT(2)	numéro de la ligne
	col texte	INT(1)	!V! couleur du texte : vaut une des constantes COL_*% (langage NCL).
	col fond	INT(1)	!V! couleur de fond de cellule : vaut une des constantes COL_*% (langage NCL).
	cadre	INT(1)	!V! option cadre entourant la cellule : vaut l'une des constantes GK_*% ou la combinaison de plusieurs d'entre elles.
	fonte	CSTRING(255)	!V! fonte utilisée (nom et taille)
	justif	INT(1)	!V! type de justification (gauche, droit, centré, etc.) appliqué à la cellule et vaut l'une des constantes JF_*%.
	verrou	INT(1)	!V! autorisation d'écriture (saisie interdite TRUE% et autorisée FALSE%)

	format	CSTRING(255)	!V! format d'affichage appliqué à la cellule
	filtre	CSTRING(255)	!V! chaîne des caractères autorisés à la saisie
	type	INT(1)	!V! type de la cellule (numérique, formule, ...) et vaut l'une des constantes CK_*.
	long sais	INT(2)	!V! nombre maximum de caractères acceptés en saisie

Si au moins une des coordonnées est nulle, les valeurs retournées par CALC_GETFORMAT sont les valeurs attribuées par défaut à l'initialisation de la feuille de calcul.

Exemple :

```

LOCAL INT Col_Texte%(1), INT Col_Fond%(1)
LOCAL INT Cadre%(1), CSTRING Fonte$(255)
LOCAL INT Justif%(1), INT Verrou%(1)
LOCAL CSTRING Format$(255), CSTRING Filtre$(255)
LOCAL INT Type%(1), INT Long_Sais%(2)

CALC_GETFORMAT HCalc%, 3, 2, Col_texte%, Col_fond%, Cadre%, Fonte$,
Justif%, Verrou%, Format$, Filtre$, Type%, Long_Sais%

IF Type% = CK_NUMBER%
MESSAGE "Attention", "La cellule R2C3 est de type numérique"
ENDIF

```

Voir aussi Constantes CK *, GK * et JF *, CALC SETFORMAT

Instruction **CALC_GETMARGINS**

Permet de récupérer les marges du document à imprimer.

Syntaxe	CALC_GETMARGINS (<i>handle%</i> , <i>left%</i> , <i>top%</i> , <i>right%</i> , <i>bottom%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	left%	INT(4)	distance à respecter entre le bord gauche de la page et l'extrémité gauche des lignes alignées à gauche sans retrait
	top%	INT(4)	distance à respecter entre le bord supérieur de la page et le haut de la première ligne
	right%	INT(4)	distance à respecter entre le bord droit de la page et l'extrémité droite des lignes alignées à droite sans retrait
	bottom%	INT(4)	distance à respecter entre le bord inférieur de la page et le bas de la dernière ligne

Voir aussi **CALC_SETMARGINS**

Instruction CALC_GOTO

Positionne le focus sur la cellule dont les coordonnées sont spécifiées par num ligne et num colonne. Cette cellule devient active.

Syntaxe	CALC_GOTO <i>handle Calc, num ligne, num colonne</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num ligne	INT(2)	numéro de la ligne
	num colonne	INT(2)	numéro de la colonne

1. Si nécessaire, la feuille de calcul est automatiquement décalée pour rendre visible la cellule activée.
2. num ligne doit être compris entre 1 et le nombre de lignes de la feuille de calcul, spécifié par CALC_SETNBROWS (CALC_MAX_ROWS% par défaut). Dans le cas contraire, la nouvelle cellule active sera sur la dernière ligne (dont le numéro est retourné par CALC_GETNBROWS%) de la feuille de calcul.
3. num colonne doit être compris entre 1 et le nombre de colonnes de la feuille de calcul, spécifié par CALC_SETNBCOLUMNS (256 par défaut). Dans le cas contraire, la nouvelle cellule active sera sur la dernière colonne (dont le numéro est retourné par CALC_GETNBCOLUMNS%) de la feuille de calcul.

Exemple :

```
; Activation de la cellule située sur la 10ème ligne de la même colonne
CALC_GOTO HCALC%, 10, CALC_COLUMN%(HCALC%)
```

Voir aussi CALC_COLUMN%, CALC_ROW%, CALC_SETTHREAD, CALC_GETTHREAD\$

Instruction **CALC_LOCK**

Interdit (indic verrou à TRUE%) ou autorise (indic verrou à FALSE%) la modification par saisie de la feuille de calcul.

Syntaxe	CALC_LOCK <i>handle Calc, indic verrou</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic verrou	INT(1)	indicateur de verrouillage

Les cellules bloquées individuellement avec CALC_LOCKBLOCK ne sont pas concernées si indic verrou est à FALSE%.

Exemple :

```
; Inverser le verrouillage de la feuille sur le menu correspondant
IF CALC_ISLOCKED% (HCALC%)
CALC_LOCK HCALC%, FALSE%
ELSE
CALC_LOCK HCALC%, TRUE%
ENDIF
```

Voir aussi CALC_ISLOCKED%, CALC_ISCELLLOCKED%, CALC_LOCKBLOCK

Instruction **CALC_LOCKBLOCK**

Interdit (indic verrou à TRUE%) ou autorise (indic verrou à FALSE%) la modification par saisie des cellules du bloc défini par coord bloc.

Syntaxe	CALC_LOCKBLOCK <i>handle Calc, coord bloc, indic verrou</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	indic verrou	INT(1)	indicateur de verrouillage

Exemple :

```
; Inverser le verrouillage de la cellule R2C3
IF CALC_ISCELLLOCKED% (HCALC%, 3, 2)
CALC_LOCKBLOCK HCALC%, "R2C3", FALSE%
ELSE
CALC_LOCKBLOCK HCALC%, "R2C3", TRUE%
ENDIF
```

Voir aussi CALC_ISLOCKED%, CALC_ISCELLLOCKED%, CALC_LOCK

Instruction **CALC_LOCKHEIGHT**

Autorise (indic verrou à TRUE%) ou interdit (indic verrou à FALSE%) la modification avec la souris de la hauteur des lignes de la feuille de calcul.

Syntaxe	CALC_LOCKHEIGHT <i>handle Calc, indic verrou</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic verrou	INT(1)	indicateur de verrouillage

Exemple :

```
; Inversion du blocage de la hauteur des lignes sur le menu correspondant
IF CALC_ISHEIGHTLOCKED% (HCALC%)
CALC_LOCKHEIGHT HCALC%, FALSE%
ELSE
CALC_LOCKHEIGHT HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISHEIGHTLOCKED%**

Instruction **CALC_LOCKWIDTH**

Autorise (indic verrou à TRUE%) ou interdit (indic verrou à FALSE%) la modification avec la souris de la largeur des colonnes de la feuille de calcul.

Syntaxe	CALC_LOCKWIDTH <i>handle Calc, indic verrou</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic verrou	INT(1)	indicateur de verrouillage

Exemple :

```
; Inversion du blocage de la hauteur des lignes sur le menu correspondant
IF CALC_ISWIDTHLOCKED% (HCALC%)
CALC_LOCKWIDTH HCALC%, FALSE%
ELSE
CALC_LOCKWIDTH HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISWIDTHLOCKED%**

Instruction **CALC_MAPUSERMSG**

Au moment de l'impression d'un document NS-CALC, cette instruction permet de rediriger les notifications nsCalc sur une fenêtre destination et un événement utilisateur USERn pour chacune des pages imprimées.

Syntaxe	CALC_MAPUSERMSG <i>handle-Calc, hDest%, iCalcMsg%, iUserMsg%</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS Calc
	hDest%	INT(4)	handle de la fenêtre de destination
	iCalcMsg%	INT(2)	constantes CCALC_MSG_DRAW_*
	iUserMsg%	INT(2)	numéro de l'événement dans lequel est reçu le CALLBACK

1. Les paramètres de la fonction de rappel sont les suivants :

a) Param12% : CCALC_MSG_DRAW_*

b) Param34% : pointeur sur une structure de type MSGDRAWMARGIN

2. Au moment de la réception de la fonction de rappel, l'utilisateur peut alors, à l'aide des informations contenues dans le segment MSGDRAWMARGIN (handle de la feuille calc, le Presentation-Space, ...), dessiner et/ou écrire ce qu'il veut sur la zone en question ou toute autre partie de la page en cours d'impression.

Voir aussi CALC_MSG_DRAW*, MSGDRAWMARGIN

Instruction **CALC_MULTI_SELECT**

Positionne le focus sur la cellule dont les coordonnées sont spécifiées par num ligne et num colonne, sans modifier la sélection courante.

CALC_MULTI_SELECT doit être utilisée pour obtenir une sélection multiple, c'est à dire pour sélectionner plusieurs blocs.

Syntaxe	CALC_MULTI_SELECT <i>handle Calc, num ligne, num colonne</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num ligne	INT(2)	numéro de la ligne
	num colonne	INT(2)	numéro de la colonne

1. Si nécessaire, la feuille de calcul est automatiquement décalée pour rendre visible la cellule activée.
2. num ligne doit être compris entre 1 et le nombre de lignes de la feuille de calcul, spécifié par CALC_SETNBROWS (CALC_MAX_ROWS% par défaut). Dans le cas contraire, la nouvelle cellule active sera sur la dernière ligne (dont le numéro est retourné par CALC_GETNBROWS%) de la feuille de calcul.
3. num colonne doit être compris entre 1 et le nombre de colonnes de la feuille de calcul, spécifié par CALC_SETNBCOLUMNS (256 par défaut). Dans le cas contraire, la nouvelle cellule active sera sur la dernière colonne (dont le numéro est retourné par CALC_GETNBCOLUMNS%) de la feuille de calcul.

Exemple :

```
; Sélectionner le bloc R1C1/R3C5
CALC_GOTO HCALC%, 1, 1
CALC_SELECT HCALC%, 3, 5

; Déplacer le focus sur la cellule R7C2 et ajouter le bloc R7C2/R9C10 à la
sélection
CALC_MULTI_SELECT HCALC%, 7, 2
CALC_SELECT HCALC%, 9, 10
```

Voir aussi CALC_GOTO, CALC_MULTI_SELECTION\$, CALC_SELECT

Instruction CALC_PASTE

Copie dans la feuille de calcul le contenu du presse papiers à partir de la cellule active.

Syntaxe	CALC_PASTE <i>handle Calc,</i>		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc

1. Le presse papiers doit avoir été préalablement rempli par l'instruction CALC_COPY. Dans le cas contraire, CALC_PASTE est sans effet.
2. Si une seule cellule vide a été copiée dans le presse papiers, CALC_PASTE n'effacera pas le contenu de la cellule active. Par contre, si le presse papiers contient plusieurs cellules dont certaines sont vides, CALC_PASTE effacera le contenu des cellules correspondantes.

Exemple :

```
; Placer le contenu du presse-papiers dans la feuille de calcul à partir de  
la cellule active  
IF CALC_ISCLIPBOARD%  
CALC_PASTE HCALC%  
ENDIF
```

Voir aussi CALC_PASTESPECIAL, CALC_ISCLIPBOARD%, CALC_COPY, CALC_CLEAR

Instruction **CALC_PASTESPECIAL**

Similaire à **CALC_PASTE**, **CALC_PASTESPECIAL** colle une partie du presse papiers à partir de la cellule active, mais sans obligatoirement remplacer le contenu des cellules concernées.

Syntaxe	CALC_PASTESPECIAL <i>handle Calc, type</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	type	INT(2)	type de collage

1. type permet de spécifier les informations contenues dans le presse papiers à prendre en compte (format, valeur, formule, etc.) ainsi que le type de collage à effectuer (ajouter le contenu des cellules du presse papiers aux cellules concernées par le collage, le soustraire, etc.) ; il est une combinaison des constantes **PT_***.

2. Le presse papiers doit avoir été préalablement rempli par l'instruction **CALC_COPY**. Dans le cas contraire, **CALC_PASTESPECIAL** est sans effet.

Exemple :

```
; Ajouter aux cellules touchées par le collage les formules des cellules
contenues dans le presse papiers
IF CALC_ISCLIPBOARD%(HCALC%)
CALC_PASTESPECIAL HCALC%, PT_FORMULA% + PT_ADD%
ENDIF
```

Voir aussi Constantes PT_*, CALC_PASTE, CALC_ISCLIPBOARD%, CALC_COPY, CALC_CLEAR

Instruction **CALC_PREPARE_PUT**

Cette instruction effectue une allocation mémoire pour préparer l'instruction **CALC_PUT**.

Syntaxe	CALC_PREPARE_PUT <i>handle Calc, coord bloc</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées des blocs

CALC_PREPARE_PUT n'est pas indispensable si le **CALC_PUT** qui la suit n'est pas trop volumineux. Par contre, il est conseillé d'utiliser le **CALC_PREPARE_PUT** si le **CALC_PUT** affecte plusieurs centaines de cellules, l'allocation mémoire accélérera le traitement.

Exemple :

```
LOCAL CSTRING Chaine$ (255)

MOVE "****" TO Chaine$
CALC_PREPARE_PUT HCALC%, "R10C2:R2010C200"
CALC_PUT HCALC%, "R10C2:R2010C200", Chaine$
```

Voir aussi **CALC_PUT**

Instruction **CALC_PRINTGETDISPLAYEDCELLSIZE**

Permet de récupérer la taille du texte tel qu'il sera imprimé dans un job d'impression.

Syntaxe	CALC_PRINTGETDISPLAYEDCELLSIZE <i>handle%, row%, TY%, W%, H%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	rows%	INT	nombre de lignes utilisées
	TY%	INT(2)	une constante CTEXT*%
	W%	INT(2)	largeur du texte
	H%	INT(2)	hauteur du texte

Le paramètre TY% permet de sélectionner le type de contenu affiché :

- CTEXTDSPTYPE% 0 Texte
- CTEXTOVERRUNDSPTYPE% 1 Texte + overrun
- CNUMDSPTYPE% 2 Numérique
- CBMPDSPTYPE% 3 Bitmap

Voir aussi **CALC_GETDISPLAYEDCELLSIZE**

Instruction CALC_PUT

Place le contenu valeur (numérique, chaîne ou formule) dans toutes les cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_PUT <i>handle Calc, coord bloc, valeur</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées des blocs
	valeur	CSTRING(255)	valeur à insérer

Si le bloc n'est pas correctement défini, par exemple "R3C3:R2C2", l'instruction CALC_PUT n'a aucun effet.

Exemple :

```
; Placer dans les cellules du bloc courant la formule "=R2C2*R3C3"
CALC_PUT HCALC%, CALC_SELECTION$(HCALC%), "=R2C2*R3C3"
; Toutes les cellules afficheront 6 si R2C2 contient 2 et R3C3 contient 3
```

Voir aussi CALC PREPARE PUT, CALC COPY, CALC PASTE, CALC CLEAR

Instruction **CALC_RC2XY**

Transforme les coordonnées passées sous forme d'une chaîne "RnCm" dans coord cellule en coordonnées num colonne et num ligne.

Syntaxe	CALC_RC2XY <i>handle Calc, coord cellule, num colonne, num ligne</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord cellule	CSTRING(255)	coordonnées de la cellule
	num colonne	INT(2)	!V! numéro de colonne
	num ligne	INT(2)	!V! numéro de ligne

Si la chaîne représente les coordonnées d'un bloc (par exemple RnCm:RzCt), CALC_RC2XY retournera 0 dans num colonne et num ligne.

Exemple :

```
; Affecter aux variables entières Colonne% et Ligne% le numéro de colonne
et de ligne de la cellule R21C45
LOCAL INT Colonne% (2), INT Ligne% (2)

CALC_RC2XY HCalc%, "R21C45", Colonne%, Ligne%
; Retournera Colonne% = 45 et Ligne% = 21
```

Voir aussi CALC_COLUMN%, CALC_ROW%, CALC_SELECTION\$

Instruction **CALC_REDO**

Refait l'action la plus récente, lorsque cela est possible.

Les actions pouvant être refaites sont celles qui concernent le presse papiers (couper, coller, etc.) et la saisie clavier.

Syntaxe	CALC_REDO <i>handle Calc</i>		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc

Exemple :

```
IF CALC_ISREDO% (HCALC%)  
CALC_REDO HCALC%  
ENDIF
```

Voir aussi **CALC_ISUNDO%**, **CALC_ISREDO%**, **CALC_UNDO**, **CALC_COPY**, **CALC_PASTE**, **CALC_PASTESPECIAL**

Instruction **CALC_REMLINK**

Supprime le trait entre la cellule de coordonnées num ligne src, num colonne src et la cellule de coordonnées num ligne dest, num colonne dest.

Syntaxe	CALC_REMLINK <i>handle Calc, num ligne src, num colonne src, num ligne dest, num colonne dest</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num ligne src	INT(2)	ligne source
	num colonne src	INT(2)	colonne source
	num ligne dest	INT(2)	ligne destination
	num colonne dest	INT(2)	colonne destination

Exemple :

```
; Effacer le trait entre les cellules R1C1 et R3C4
CALC_REMLINK HCalc%, 1, 1, 3, 4
```

Voir aussi **CALC_ADDLINK**, **CALC_ADDENHANCEDLINK**, **CALC_ISLINK%**

Instruction **CALC_RESET**

Efface la totalité d'une feuille.

Syntaxe	CALC_RESET <i>handle Calc, flags%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	flags%	INT(2)	indicateur de la feuille

Instruction **CALC_SCROLL_DOWN**

Décale de nb lignes vers le bas le contenu des cellules définies dans coord bloc, et ce, à l'intérieur du bloc.

Syntaxe	CALC_SCROLL_DOWN <i>handle Calc, coord bloc, nb lignes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nb lignes	INT(2)	nombre de lignes de décalage

1. Le contenu du bloc écrasera le contenu des cellules cibles si elles ne sont pas vides.
2. Les cellules libérées sont vidées. Aucune cellule en dehors du bloc ne sera modifiée.

Exemple :

```
; Placer le contenu de la cellule R4C3 en R6C3
; R4C4 en R6C4
; R5C3 en R7C3
; ...
; R8C5 en R10C5
```

```
CALC_SCROLL_DOWN HCALC%, "R4C3:R10C5", 2
```

Voir aussi [CALC_SCROLL_LEFT](#), [CALC_SCROLL_RIGHT](#), [CALC_SCROLL_UP](#)

Instruction **CALC_SCROLL_LEFT**

Décale de nb colonnes colonnes vers la gauche le contenu des cellules définies dans coord bloc, et ce, à l'intérieur du bloc.

Syntaxe	CALC_SCROLL_LEFT <i>handle Calc, coord bloc, nb colonnes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nb colonnes	INT(2)	nombre de colonnes de décalage

1. Le contenu du bloc écrasera le contenu des cellules cibles si elles ne sont pas vides.
2. Les cellules libérées sont vidées. Aucune cellule en dehors du bloc ne sera modifiée.

Exemple :

```
; Placer le contenu de la cellule R4C3 en R4C1
; R4C4 en R4C2
; R5C3 en R5C1
; ...
; R8C5 en R8C3
```

```
CALC_SCROLL_LEFT HCALC%, "R4C1:R8C5", 2
```

Voir aussi [CALC_SCROLL_DOWN](#), [CALC_SCROLL_RIGHT](#), [CALC_SCROLL_UP](#)

Instruction **CALC_SCROLL_RIGHT**

Décale de nb colonnes vers la droite le contenu des cellules définies dans coord bloc, et ce, à l'intérieur du bloc.

Syntaxe	CALC_SCROLL_RIGHT <i>handle Calc, coord bloc, nb colonnes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nb colonnes	INT(2)	nombre de colonnes de décalage

1. Le contenu du bloc écrasera le contenu des cellules cibles si elles ne sont pas vides.
2. Les cellules libérées sont vidées. Aucune cellule en dehors du bloc ne sera modifiée.

Exemple :

```
; Placer le contenu de la cellule R4C3 en R4C5
; R4C4 en R4C6
; R5C3 en R5C5
; ...
; R8C5 en R8C7
```

```
CALC_SCROLL_RIGHT HCALC%, "R4C3:R8C7", 2
```

Voir aussi [CALC_SCROLL_DOWN](#), [CALC_SCROLL_LEFT](#), [CALC_SCROLL_UP](#)

Instruction **CALC_SCROLL_UP**

Décale de nb lignes vers le haut le contenu des cellules définies dans coord bloc, et ce, à l'intérieur du bloc.

Syntaxe	CALC_SCROLL_UP <i>handle Calc, coord bloc, nb lignes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nb lignes	INT(2)	nombre de lignes de décalage

1. Le contenu du bloc écrasera le contenu des cellules cibles si elles ne sont pas vides.
2. Les cellules libérées sont vidées. Aucune cellule en dehors du bloc ne sera modifiée.

Exemple :

```
; Placer le contenu de la cellule R4C3 en R2C3
; R4C4 en R2C4
; R5C3 en R3C3
; ...
; R8C5 en R6C5
```

```
CALC_SCROLL_UP HCALC%, "R2C3:R8C5", 2
```

Voir aussi [CALC_SCROLL_DOWN](#), [CALC_SCROLL_LEFT](#), [CALC_SCROLL_RIGHT](#)

Instruction **CALC_SELECT**

Sélectionne un bloc de la position du curseur (cellule active) à la cellule dont les coordonnées sont passées dans les paramètres num ligne, num colonne.

Syntaxe	CALC_SELECT <i>handle Calc, num lignes, num colonne</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num ligne	INT(2)	numéro de la ligne
	num colonne	INT(2)	numéro de la colonne

Exemple :

```
; Sélectionner le bloc R1C1 à R3C5
CALC_GOTO HCALC%, 1, 1
CALC_SELECT HCALC%, 3, 5
```

Voir aussi [CALC_SELECTION\\$](#), [CALC_MULTI_SELECT](#), [CALC_MULTI_SELECTION\\$](#)

Instruction **CALC_SETBACKCOLOR**

Modifie la couleur par défaut du fond des cellules avec la couleur col fond.

Syntaxe	CALC_SETBACKCOLOR <i>handle Calc, col fond</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	col fond	INT(1)	code couleur

1. Les cellules modifiées individuellement avec **CALC_SETBLOCKBACKCOL** ne sont pas concernées par cette modification.
2. Les codes couleur correspondent aux constantes **COL_*** (langage NCL).

Ne pas confondre avec **CALC_SETDEFCOLOR** qui spécifie le fond de la page en dehors des cellules (visible si l'on affiche par exemple 10 lignes x 10 colonnes).

Exemple :

```
; Colorer le fond des cellules en rouge  
CALC_SETBACKCOLOR HCALC%, COL_RED%
```

Voir aussi **CALC_GETBACKCOLOR%**, **CALC_GETCELLBACKCOL%**, **CALC_SETBLOCKBACKCOL**, **CALC_GETTEXTCOLOR%**, **CALC_SETTEXTCOLOR**, **CALC_GETDEFCOLOR%**, **CALC_SETDEFCOLOR**

Instruction **CALC_SETBLOCKBACKCOL**

Modifie la couleur de fond des cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_SETBLOCKBACKCOL <i>handle Calc, coord bloc, col fond</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	col fond	INT(1)	code couleur

Les codes couleur correspondent aux constantes COL_*(langage NCL).

Exemple :

```
; Modification de la couleur de fond de la cellule R2C3
IF CALC_GETCELLBACKCOL% (HCALC%, 3, 2) = COL_BLACK%
CALC_SETBLOCKBACKCOL HCALC%, "R2C3", COL_WHITE%
ENDIF
```

Voir aussi [CALC_GETCELLBACKCOL%](#), [CALC_GETBACKCOLOR%](#), [CALC_SETBACKCOLOR](#), [CALC_GETTEXTCOLOR%](#), [CALC_SETTEXTCOLOR](#), [CALC_GETDEFCOLOR%](#), [CALC_SETDEFCOLOR](#)

Instruction **CALC_SETBLOCKBORDER**

Modifie les caractéristiques de la bordure des cellules contenues dans le bloc coord bloc.

Ces caractéristiques concernent, pour chaque bord, le type de trait, sa taille et sa couleur. Elles sont spécifiées dans les champs du segment style bordure de type BORDERSTYLE.

Syntaxe	CALC_SETBLOCKBORDER <i>handle Calc, coord bloc, style bordure</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	style bordure	BORDERSTYLE	!V! caractéristiques de la bordure

Exemple :

```
LOCAL BORDERSTYLE Style_Bordure

; Appliquer une bordure bleue aux cellules du bloc sélectionné
CALC_GETCELLBORDER HCALC%, CALC_COLUMN%(HCALC%), CALC_ROW(HCALC%),
Style_Bordure
MOVE COL_BLUE% TO Style_Bordure.LeftLineColor
MOVE COL_BLUE% TO Style_Bordure.RightLineColor
MOVE COL_BLUE% TO Style_Bordure.TopLineColor
MOVE COL_BLUE% TO Style_Bordure.BottomLineColor

CALC_SETBLOCKBORDER HCALC%, CALC_SELECTION$(HCALC%), Style_Bordure
```

Voir aussi Segment **BORDERSTYLE**, **CALC_GETCELLBORDER**, **CALC_SETFORMAT**

Instruction **CALC_SETBLOCKCOMMENT**

Permet d'associer une chaîne de caractères ou "commentaire" à des cellules.

Syntaxe	CALC_SETBLOCKCOMMENT <i>handle Calc, block\$, size%, buff%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING(255)	coordonnées du bloc
	size%	INT(4)	nombre d'octets pointé par buff%
	buff%	INT(4)	buffer contenant les données

Vous pouvez utiliser le nom d'une cellule dans le paramètre block\$.

Voir aussi [CALC_GETCELLCOMMENT](#)

Instruction **CALC_SETBLOCKDWSPACE**

Permet de spécifier l'espace entre un bloc de cellules et le texte.

Syntaxe	CALC_SETBLOCKDWSPACE <i>handle Calc, block\$, w%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING(255)	bloc de cellules
	w%	INT(2)	espace entre le bord gauche du bloc de cellules et le texte contenu

Vous pouvez utiliser le nom d'une cellule dans le paramètre block\$.

Voir aussi CALC_GETDWSPACE, CALC_SETBLOCKDWSPACE, CALC_GETCELLDWSPACE

Instruction **CALC_SETBLOCKEDITCTRL**

Définit le type de contrôle utilisé dans la ligne de statut pour la saisie des cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_SETBLOCKEDITCTRL <i>handle Calc, coord bloc, type contrôle, données</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	type contrôle	INT(2)	type du contrôle de saisie. L'une des constantes EDTCTRL_*% : EDTCTRL_EF% (utilisé par défaut) pour que la saisie s'effectue dans un champ de saisie normal, EDTCTRL_CBE% pour que la saisie s'effectue à l'aide d'une combo box avec champ de saisie.
	données	INT(4) [4]	données à affecter au contrôle dans le cas d'une combo box. Ces données peuvent être contenues soit dans un fichier, soit dans un tableau. S'il n'y a pas de donnée à affecter, ce paramètre peut être nul.

données est un tableau de 4 entiers, qui est passé par adresse à l'instruction. Le premier élément de ce tableau contient le type des données : CTRLDATA_NONE% s'il n'y a pas de donnée à affecter au contrôle, CTRLDATA_FILE% si les données sont contenues dans un fichier, CTRLDATA_ARRAY% si les données sont contenues dans un tableau. Les autres éléments du tableau données dépendent du type de données :

- CTRLDATA_NONE% pas de donnée, donc aucune information supplémentaire n'est nécessaire.
- CTRLDATA_FILE% le deuxième élément du tableau données est l'adresse d'une chaîne contenant le nom du fichier texte à charger.
- CTRLDATA_ARRAY% le deuxième élément du tableau données est le nombre de lignes (chaînes) composant le tableau à charger,
- le troisième élément du tableau données est la taille maximale + 1 de ces chaînes de caractères,
- le quatrième élément du tableau données est l'adresse du tableau de chaînes à charger.

Exemple 1 : pas de données

```
CALC_SETBLOCKEDITCTRL HCALC%, "R2C3", EDTCTRL_EF%, 0
```

Exemple 1 bis : pas de données

```
LOCAL Data%[4]

MOVE CTRLDATA_NONE% TO Data%[0]

CALC_SETBLOCKEDITCTRL HCALC%, "R2C3", EDTCTRL_EF%, @Data%
```

Exemple 2 : données dans un fichier

```
LOCAL Data%[4]
LOCAL Fichier$

MOVE "COLORS.TXT" TO Fichier$

MOVE CTRLDATA_FILE% TO Data%[0]
MOVE @Fichier$ TO Data%[1]

CALC_SETBLOCKEDITCTRL HCALC%, "R2C3", EDTCTRL_CBE%, @Data%
```

Exemple 3 : données dans un tableau

```
LOCAL Data%[4]
LOCAL Chaînes$(16) [5]

MOVE "Première chaîne" TO Chaînes$[0]
MOVE "Deuxième chaîne" TO Chaînes$[1]
MOVE "Troisième chaîne" TO Chaînes$[2]
MOVE "Quatrième chaîne" TO Chaînes$[3]
MOVE "Cinquième chaîne" TO Chaînes$[4]

MOVE CTRLDATA_ARRAY% TO Data%[0]
MOVE 5 TO Data%[1] ; 5 = nombre de chaînes
MOVE 17 TO Data%[2] ; 17 = taille des chaînes (16) + 1
MOVE @Chaînes$ TO Data%[3]

CALC_SETBLOCKEDITCTRL HCALC%, "R2C3", EDTCTRL_CBE%, @Data%
```

Voir aussi Constantes CTRLDATA_*, EDTCTRL_*, CALC_GETCELLEDITCTRL

Instruction **CALC_SETBLOCKEXTBORDER**

Permet de récupérer le style de bordure d'un bloc de cellules.

Syntaxe	CALC_SETBLOCKEXTBORDER <i>handle Calc, block\$, EXT BORDERSTYLE</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING(255)	coordonnées du bloc
	EXT BORDERSTYLE	BORDERSTYLE	style de bordure des cellules

1. Le segment **EXT BORDERSTYLE** permet notamment de sélectionner le côté de la ligne (à gauche, à droite, en haut, ou en bas) qui contiendra le trait de bordure. Il utilise également le segment **BORDERLINE**. La déclaration interne du segment **EXT BORDERSTYLE** est :

```
SEGMENT EXT BORDERSTYLE
  BORDERLINE LINE[4] ; 0: Left, 1: Right, 2: Top, 3: Bottom
ENDSEGMENT
```

2. Le segment **BORDERLINE** permet de définir les options de style, de largeur et de couleur de bordure souhaitées. La déclaration interne du segment **BORDERLINE** est :

```
SEGMENT BORDERLINE
  INT KIND(2) ; Type de ligne.
  INT WIDTH(2) ; Epaisseur.
  INT RED(2) ; Couleurs en mode RGB.
  INT GREEN(2); Couleurs en mode RGB.
  INT BLUE(2); Couleurs en mode RGB.
ENDSEGMENT
```

Voir aussi **CALC_GETCELLEXTBORDER**

Instruction CALC_SETBLOCKFILTER

Modifie le filtre de saisie des caractères des cellules contenues dans le bloc coord bloc.

Ce filtre de saisie est une chaîne reprenant les attributs de la cellule auxquels s'ajoute la liste des caractères autorisés en saisie.

Chaque attribut, caractère ou ensemble de caractères est séparé du précédent par le caractère #1.

Par défaut, le filtre est 'AUTOSCROLL#1LEFT#1BELOW' (tous les caractères sont acceptés en saisie).

Syntaxe	CALC_SETBLOCKFILTER <i>handle Calc, coord bloc, filtre</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	filtre	CSTRING(255)	filtre de saisie

Les attributs de cellules possibles sont une combinaison des suivants :

- **UPCASE** : Saisie en majuscules, quelque soit l'état du clavier. La conversion de minuscule en majuscule est faite en cours de saisie.
- **SKIPBLANKS** : Suppression des éventuels espaces saisis en début de champ et en fin de champ.
- **NOBLANKS** : Impossibilité de saisir des espaces.
- **HIDETEXT** : Le texte saisi n'est pas affiché, mais les caractères sont remplacés par des étoiles.
- **FULLTEXT** : Oblige à saisir le nombre maximum de caractères spécifiés.
- **INTEGER** ou **DATE** ou **TIME** : Pour être acceptée, la valeur saisie doit être respectivement un entier, une date ou une heure.
- **OVER** ou **BELOW** : Le relief donne un aspect bombé (**OVER**) ou creux (**BELOW**) au champ de saisie.
- **AUTOSCROLL** : Un défilement automatique du contenu du champ de saisie est effectué lorsque le nombre de caractères saisis dépasse la taille du champ.
- **CENTER** ou **LEFT** ou **RIGHT** : Centre, justifie à gauche ou justifie à droite la valeur saisie dans le champ de saisie (indépendamment de la justification dans la cellule).
- **MARGIN** : Encadre le champ de saisie.
- **Caractères** : Caractères acceptés lors de la saisie. Ceux ci doivent être encadrés par des apostrophes et séparés par des virgules. Une suite de caractères (dans l'ordre de leur code ASCII), est indiquée par les deux bornes séparées par deux points (..).

Exemple :

```
LOCAL Filtre$  
; N'autoriser que les caractères majuscules de 'A' à 'P', le tiret ' ' et  
les chiffres de '0' à '5' pour la saisie de la cellule R2C3
```

```
MOVE CALC_GETCELLFILTER$ (HCALC%, 3, 2) TO Filtre$  
MOVE Filtre$ & #1 & "'A'..'P',' ','0'..'5'" TO Filtre$  
CALC_SETBLOCKFILTER HCALC%, "R2C3", Filtre$
```

Voir aussi CALC_GETCELLFILTER\$, CALC_SETFORMAT

Instruction **CALC_SETBLOCKFONT**

Modifie les caractéristiques de la fonte utilisée dans les cellules du bloc coord bloc en leur affectant les nouvelles caractéristiques nom fonte et style fonte.

Syntaxe	CALC_SETBLOCKFONT <i>handle Calc, coord bloc, nom fonte, style fonte</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nom fonte	CSTRING(255)	nom et taille de la fonte (sous la forme d'une chaîne de caractères, les deux informations étant séparées par une virgule).
	style fonte	INT(1)	style de la fonte (gras, italique, etc.) : combinaison des constantes GFS_*% (bibliothèque NSGRAPH)

Exemple :

```
LOCAL Nom_fonte$, Style_fonte%(1)
; Appliquer l'attribut 'souligné' à la cellule courante

CALC_GETCELLFONT HCALC%, CALC_COLUMN%(HCALC%), CALC_ROW%(HCALC%),
Nom_fonte$, Style_fonte%

CALC_SETBLOCKFONT HCALC%, "R"&CALC_ROW%(HCALC%) & "C"&CALC_COLUMN%(HCALC%),
Nom_fonte$, \
Style_fonte% BOR GFS_UNDERSCORE%
```

Voir aussi [CALC_GETCELLFONT](#), [CALC_SETFORMAT](#)

Instruction **CALC_SETBLOCKFORMAT**

Modifie le format d'affichage des cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_SETBLOCKFORMAT <i>handle Calc, coord bloc, format</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	format	CSTRING(255)	format d'affichage

Les formats acceptés par NS Calc sont les formats EXCEL (décrits dans le Manuel d'Utilisation de NS Design).

Exemple :

```
; Modification du format d'affichage
IF CALC_GETCELLFORMAT$ (HCALC%, 3, 2) = "#,##0"
CALC_SETBLOCKFORMAT HCALC%, "R2C3", "#,##0.00"
ENDIF
```

Voir aussi **CALC_GETCELLFORMAT\$**, **CALC_SETFORMAT**

Instruction **CALC_SETBLOCKGRID**

Modifie le cadre des cellules contenues dans le bloc coord bloc.

Les caractéristiques du cadre appliqué (haut, bas, etc.), sont une combinaison des constantes GK_ *%.

Syntaxe	CALC_SETBLOCKGRID <i>handle Calc, coord bloc, cadre</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	cadre	INT(1)	caractéristiques du cadre

Exemple :

```
LOCAL Cadre%  
; Suppression, s'il y a lieu, des traits sur les bords haut et bas de la  
cellule R2C3  
  
MOVE CALC_GETCELLGRID%(HCALC%, 3, 2) TO Cadre%  
IF Cadre% BAND (GK_TOP% + GK_BOTTOM%)  
MOVE Cadre% BXOR (GK_TOP% + GK_BOTTOM%) TO Cadre%  
CALC_SETBLOCKGRID HCALC%, "R2C3", Cadre%  
ENDIF
```

Voir aussi Constantes GK_ *%, CALC_GETCELLBORDER, CALC_SETBLOCKBORDER,
CALC_GETCELLGRID%, CALC_SETFORMAT

Instruction **CALC_SETBLOCKHORZJUSTIF**

Modifie la justification horizontale des cellules du bloc coord bloc.

justif est l'une des constantes HJ_ *%.

Syntaxe	CALC_SETBLOCKHORZJUSTIF <i>handle Calc, coord bloc, justif</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	justif	INT(1)	justification horizontale

Exemple :

```

LOCAL Justif%(1)
; Inversion de la justification horizontale

MOVE CALC_GETCELLHORZJUSTIF%(HCALC%, 3, 2) TO Justif%
IF Justif% = HJ_LEFT%
CALC_SETBLOCKHORZJUSTIF HCALC, "R2C3", HJ_RIGHT%
ELSEIF Justif% = HJ_RIGHT%
CALC_SETBLOCKHORZJUSTIF HCALC, "R2C3", HJ_LEFT%
ENDIF

```

Voir aussi Constantes HJ_ *%, CALC_GETCELLHORZJUSTIF%, CALC_SETFORMAT

Instruction **CALC_SETBLOCKKIND**

Modifie le type (numérique, alphanumérique, etc.) des cellules contenues dans le bloc coord bloc. type est l'une des constantes CK_ *%.

Par défaut, le type des cellules est CK_DEFAULT%.

Syntaxe	CALC_SETBLOCKKIND <i>handle Calc, coord bloc, type</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	type	INT(1)	type de données

Exemple :

```
; Appliquer, s'il y a lieu, le type numérique
IF CALC_GETCELLKIND% (HCALC%, 3, 2) <> CK_NUMBER%
CALC_SETBLOCKKIND HCALC%, "R2C3", CK_NUMBER%
ENDIF
```

Voir aussi Constantes CK_ *%, CALC_GETCELLKIND%, CALC_SETFORMAT

Instruction **CALC_SETBLOCKLENGTH**

Modifie le nombre maximum de caractères acceptés en saisie dans les cellules du bloc coord bloc.

Le maximum possible est de 255 caractères. C'est aussi le maximum par défaut.

Syntaxe	CALC_SETBLOCKLENGTH <i>handle Calc, coord bloc, nb caract</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	nb caract	INT(2)	nombre maximum de caractères

Exemple :

```
; Appliquer, s'il y a lieu, un nombre maximum de 10 caractères pour la
cellule R2C3
IF CALC_GETCELLLENGTH% (HCALC%, 3, 2) > 10
CALC_SETBLOCKLENGTH HCALC%, "R2C3", 10
ENDIF
```

Voir aussi **CALC_GETCELLLENGTH%**, **CALC_SETFORMAT**

Instruction **CALC_SETBLOCKOVERRUN**

Positionne la propriété de débordement sur un groupe de cellules.

Syntaxe	CALC_SETBLOCKOVERRUN <i>handle Calc, overrun%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	block\$	CSTRING	bloc de cellules
	overrun%	INT(1)	débordement du texte

1. Vous pouvez utiliser le nom d'une cellule dans le paramètre block\$.
2. Le paramètre overrun% peut prendre les valeurs suivantes :

Constantes	Valeurs	Description
OV_INACTIVE%	0	Pas de débordement.
OV_ACTIVE%	1	Débordement autorisé.
OV_INDETERMINATE%	2	Déterminé par la propriété de débordement de la feuille.

Voir aussi CALC ISOVERRUN%, CALC SETOVERRUN, CALC GETCELLOVERRUN%, CALC PRINTGETDISPLAYEDCELLSIZE, CALC GETDISPLAYEDCELLSIZE

Instruction **CALC_SETBLOCKPATTCOL**

Permet de choisir la couleur de pattern du bloc de cellules.

Syntaxe	CALC_SETBLOCKPATTCOL <i>handle Calc, coord bloc, pattern%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	pattcolor%	INT(1)	couleur du pattern

Les valeurs possibles du champ pattern% sont les constantes PATSYM_*% de la librairie NSGraph.

Voir aussi CALC_GETCELLPATTERN%, CALC_SETBLOCKPATTERN, CALC_GETCELLPATTCOL%

Instruction **CALC_SETBLOCKPATTERN**

Permet de choisir le type de pattern du bloc de cellules.

Syntaxe	CALC_SETBLOCKPATTERN <i>handle Calc, coord bloc, pattern%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	pattern%	INT(1)	type de pattern

Les valeurs possibles du champ pattern% sont les constantes PATSYM_%% de la librairie NSGraph.

Voir aussi CALC_GETCELLPATTERN%, CALC_GETCELLPATTCOL%, CALC_SETBLOCKPATTCOL

Instruction **CALC_SETBLOCKRGBCOLOR**

Modifie la couleur (du texte, du fond ou du pattern) des cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_SETBLOCKRGBCOLOR <i>handle-Calc, coord bloc, item%, r%, g%, b%</i>		
Paramètres	handle-Calc	INT(4)	handle fenêtre NS-Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	item%	CSTRING	une des constantes IT_CELL%
	r%	INT(2)	niveau de couleur rouge comprise entre 0 et 255
	g%	INT(2)	niveau de couleur verte comprise entre 0 et 255
	b%	INT(2)	niveau de couleur bleue comprise entre 0 et 255

1. Le paramètre item% peut prendre l'une des valeurs suivantes :

Constantes	Valeurs	Description
IT_CELLTEXT%	1	Couleur du texte.
IT_CELLBACKGROUND%	2	Couleur de fond.
IT_CELLFOREGROUND%	3	Couleur du pattern.

2. Les paramètres r%, g% et b% correspondent à des niveaux de couleurs. Par exemple, 255,255,255 correspond à la couleur blanche.

Voir aussi **CALC_GETCELLRGBCOLOR**

Instruction **CALC_SETBLOCKTEXTCOL**

Modifie la couleur de texte des cellules contenues dans le bloc coord bloc.

Syntaxe	CALC_SETBLOCKTEXTCOL <i>handle Calc, coord bloc, col texte</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	col texte	INT(1)	couleur de texte

Les codes couleur correspondent aux constantes COL_ *% (langage NCL).

Exemple :

; Modification de la couleur de texte de la cellule R2C3

IF CALC_GETCELLTEXTCOL% (HCALC%, 3, 2) = COL_WHITE%

CALC_SETBLOCKTEXTCOL HCALC%, "R2C3", COL_BLACK%

ENDIF

Voir aussi CALC_GETCELLTEXTCOL%, CALC_GETCELLBACKCOL%, CALC_SETBLOCKBACKCOL, CALC_GETTEXTCOLOR%, CALC_SETTEXTCOLOR, CALC_GETDEFCOLOR%, CALC_SETDEFCOLOR

Instruction **CALC_SETBLOCKVERTJUSTIF**

Modifie la justification verticale des cellules du bloc coord bloc.

Syntaxe	CALC_SETBLOCKVERTJUSTIF <i>handle Calc, coord bloc, justif</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	justif	INT(1)	justification verticale. L'une des constantes VJ_*.

Exemple :

```

LOCAL Justif%(1)
; Inversion de la justification verticale
MOVE CALC_GETCELLVERTJUSTIF%(HCALC%, 3, 2) TO Justif%
IF Justif% = VJ_TOP%
CALC_SETBLOCKVERTJUSTIF HCALC, "R2C3", VJ_BOTTOM%
ELSEIF Justif% = VJ_BOTTOM%
CALC_SETBLOCKVERTJUSTIF HCALC, "R2C3", VJ_TOP%
ENDIF

```

Voir aussi Constantes VJ_*, CALC_GETCELLVERTJUSTIF%, CALC_SETFORMAT

Instruction **CALC_SETDEFAULTHEADMODE**

Permet de spécifier le mode d'affichage par défaut.

Syntaxe	CALC_SETDEFAULTHEADMODECOLOR <i>mode%</i> , <i>head\$</i>		
Paramètres	<i>mode%</i>	INT(4)	mode d'affichage (constantes HMODE_*%)
	<i>head\$</i>	CSTRING	chaîne de caractères

1. Deux modes d'affichage sont disponibles : numérique et alphanumérique avec les constantes suivantes :

a) HMODE_RC% 0 -> exemple : R1C1

b) HMODE_ALPHA% 1 -> exemple : A1 pour R1C1

2. Si le mode numérique est utilisé, la chaîne de caractère *head\$* contient les deux caractères utilisés, par exemple « RC » (mode par défaut).

Voir aussi **CALC_GETDEFAULTHEADMODE**

Instruction **CALC_SETDEFCOLOR**

Modifie la couleur du fond de la fenêtre (en arrière de la feuille de calcul) avec la couleur col fen.

Syntaxe	CALC_SETDEFCOLOR <i>handle Calc, col fen</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	col fen	INT(1)	code couleur

1. Les codes couleur correspondent aux constantes COL_ *% (langage NCL).
2. Attention à ne pas confondre avec CALC_SETBACKCOLOR qui permet de spécifier le couleur de fond des cellules. L'instruction CALC_SETDEFCOLOR définit la couleur de fond de la page en dehors des cellules.

Exemple :

```
; Colorer le fond de la fenêtre NS Calc en cyan
CALC_SETDEFCOLOR HCALC%, COL_CYAN%
```

Voir aussi CALC_GETDEFCOLOR%, CALC_SETBACKCOLOR

Instruction **CALC_SETDWSPACE**

Permet de spécifier l'espace entre le bord d'une cellule et le texte.

Syntaxe	CALC_SETDWSPACE <i>handle Calc, W%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	W%	INT(2)	espace entre le bord gauche de la cellule et le texte contenu

Voir aussi CALC_GETDWSPACE, CALC_SETBLOCKDWSPACE, CALC_GETCELLDWSPACE

Instruction **CALC_SETENTERDIRECTION**

Permet de spécifier l'action à exécuter lors de l'appui sur la touche [Entrée].

Syntaxe	CALC_SETENTERDIRECTION <i>handle Calc, direct%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	direct%	INT	une des constantes CALC_ENTER_ *%

Voir aussi CCALC_ENTER_ *%

Instruction CALC_SETFORMAT

Modifie les caractéristiques des cellules du bloc coord bloc.

Syntaxe	CALC_SETFORMAT <i>handle Calc, coord bloc, col texte, col fond, cadre, fonte, justif, verrou, format, filtre, type, nb caract</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	coord bloc	CSTRING(255)	coordonnées du bloc
	col texte	INT(1)	couleur du texte (constantes COL_*)
	col fond	INT(1)	couleur de fond (constantes COL_*)
	cadre	INT(1)	cadre devant entourer les cellules du bloc qui vaut l'une des constantes GK_*, ou la combinaison de plusieurs d'entre elles.
	fonte	CSTRING(255)	nom et taille de la fonte utilisée, séparés par une virgule (ex. : "Times New Roman, 10").
	justif	INT(1)	type de justification (gauche, droit, centré, etc.) appliqué aux cellules qui vaut l'une des constantes JF_*
	verrou	INT(1)	autorise (FALSE%) ou interdit (TRUE%) la saisie dans les cellules.
	format	CSTRING(255)	chaîne (format EXCEL, décrits dans le Manuel d'Utilisation de NS Design) précisant le format d'affichage devant être appliqué aux

			cellules du bloc.
	filtre	CSTRING(255)	chaîne de caractères précisant la liste des caractères acceptés en saisie.
	type	INT(1)	format des cellules (numérique, formule, ...) qui vaut l'une des constantes CK_ *%.
	nb caract	INT(2)	nombre maximum de caractères acceptés en saisie.

Exemple :

```
; Modification du format des cellules R2C2 à R3C5
; Coordonnées du bloc : "R2C2:R3C5"
; Couleur texte : bleu
; Couleur fond : cyan
; Encadrement (const. GK_*) : cadre sur 4 côtés
; Fonte caractères : "Courier, 10"
; Justification (const. JF_*) : centré
; Verrouillage : saisie autorisée
; Format : avec 2 décimales(#.##)
; Filtre : pas de filtre
; Type caractère (const. CK_*) : numérique
; Nombre caractères par cellule : 10 caractères maximum
CALC_SETFORMAT HCALC%, "R2C2:R3C5", \
COL_BLUE%, COL_CYAN%, \
GK_LEFT%+GK_RIGHT%+GK_TOP%+GK_BOTTOM%, \
"Courier, 10", JF_CENTERED%, FALSE%, \
"0.00", "", CK_NUMBER%, 10
```

Voir aussi CALC_GETFORMAT, Constantes CK_*, GK_* et JF_*

Instruction **CALC_SETGRID**

Affiche ou cache la grille sur la feuille de calcul en fonction de la valeur (TRUE% ou FALSE%) de indic affichage.

Syntaxe	CALC_SETGRID <i>handle Calc, indic affichage</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic affichage	INT(1)	indicateur d'affichage

Exemple :

```
; Inversion de l'affichage de la grille sur le menu correspondant
IF CALC_ISGRID%(HCALC%)
CALC_SETGRID HCALC%, FALSE%
ELSE
CALC_SETGRID HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISGRID%**

Instruction **CALC_SETHEADING**

Affiche ou cache les numéros de lignes et de colonnes sur la feuille de calcul en fonction de la valeur (TRUE% ou FALSE%) de indic affichage.

Syntaxe	CALC_SETHEADING <i>handle Calc, indic affichage</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic affichage	INT(1)	indicateur d'affichage

Les numéros de lignes et les numéros de colonnes sont indissociables.

Exemple :

```
; Inversion de l'affichage des numéros de lignes et de colonnes sur le menu
correspondant
IF CALC_ISHEADING%(HCALC%)
CALC_SETHEADING HCALC%, FALSE%
ELSE
CALC_SETHEADING HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISHEADING%**

Instruction **CALC_SETHEIGHT**

Fixe la hauteur des cellules des lignes num lignes avec la valeur hauteur cell, exprimée en pixels.

Syntaxe	CALC_SETHEIGHT <i>handle Calc, num lignes, hauteur cell</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num lignes	CSTRING(255)	numéro des lignes
	hauteur cell	INT(2)	hauteur des cellules

1. Un ensemble de rangées adjacentes sera défini par "Rn:Rm".
2. Si les rangées ne sont pas côte à côte, le séparateur utilisé dans num lignes est le point virgule ";".

Exemple :

```
; Donner aux lignes 2, 3, 4, et 6 une hauteur de 50 pixels  
CALC_SETHEIGHT HCalc%, "R2:R4; R6", 50
```

Voir aussi [CALC_GETHEIGHT%](#), [CALC_GETDEFHEIGHT%](#), [CALC_SETWIDTH](#)

Instruction **CALC_SETLEFTCOLUMNS**

Bloque nb col colonnes à gauche de la feuille de calcul.

Syntaxe	CALC_SETLEFTCOLUMNS <i>handle Calc, nb col</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nb col	INT(2)	nombre de colonnes

Cette instruction sera utilisée pour garder visible les premières colonnes de la feuille de calcul (contenant les titres par exemple) tout en travaillant dans les autres cellules.

Veiller à ne pas bloquer un nombre de colonnes plus grand que n'en peut contenir la fenêtre, sinon la feuille de calcul sera figée horizontalement.

Exemple :

```
; Rendre les colonnes 1 et 2 visibles en permanence
CALC_SETLEFTCOLUMNS HCALC%, 2
```

Voir aussi **CALC_GETLEFTCOLUMNS%**, **CALC_SETTOPROWS**, **CALC_GETTOPROWS%**

Instruction **CALC_SETMARGINS**

Permet de personnaliser la présentation du document à imprimer, en définissant les marges.

Syntaxe	CALC_SETMARGINS (<i>handle%</i> , <i>left%</i> , <i>top%</i> , <i>right%</i> , <i>bottom%</i>)		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	left%	INT(4)	distance à respecter entre le bord gauche de la page et l'extrémité gauche des lignes alignées à gauche sans retrait
	top%	INT(4)	distance à respecter entre le bord supérieur de la page et le haut de la première ligne
	right%	INT(4)	distance à respecter entre le bord droit de la page et l'extrémité droite des lignes alignées à droite sans retrait
	bottom%	INT(4)	distance à respecter entre le bord inférieur de la page et le bas de la dernière ligne

Voir aussi **CALC_GETMARGINS**

Instruction **CALC_SETNBCELLS**

Définit le nombre de colonnes de la ligne num ligne.

Syntaxe	CALC_SETNBCELLS <i>handle Calc, num ligne, nb colonnes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num ligne	INT(2)	numéro de ligne
	nb colonnes	INT(2)	nombre de colonnes

Cette instruction ne peut être utilisée que pour les tableaux qui acceptent des cellules de taille variable ; pour cela il est nécessaire d'appeler au préalable l'instruction **CALC_SETVARCELLWIDTHS**.

Exemple :

```
; Limiter la feuille de calcul à 2 lignes,
; avec 1 colonne dans la première ligne,
; 3 colonnes dans la deuxième ligne
CALC_SETVARCELLWIDTHS HCALC%, TRUE%
CALC_SETNBROWS HCALC%, 2
CALC_SETNBCELLS HCALC%, 1, 1
CALC_SETNBCELLS HCALC%, 2, 3
```

Voir aussi **CALC_GETNBCELLS%**, **CALC_SETVARCELLWIDTHS**

Instruction **CALC_SETNBCOLUMNS**

Définit le nombre de colonnes pour la feuille de calcul.

Syntaxe	CALC_SETNBCOLUMNS <i>handle Calc, nb colonnes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nb colonnes	INT(2)	nombre de colonnes

1. Le paramètre nb colonnes est accepté s'il est compris entre 1 et CALC_MAX_COLUMNS% qui est le nombre maximum de colonnes admissibles.
2. Cette instruction sera de préférence effectuée lors de l'initialisation de la feuille de calcul.
3. Si cette instruction est passée en cours d'utilisation :
 - a) si le nombre de colonnes est augmenté : le contenu des nouvelles cellules sera nul, mais le contenu des colonnes non affectées sera conservé,
 - b) si le nombre de colonnes est diminué : le contenu des colonnes non affectées sera conservé, et les colonnes en surplus seront supprimées.

Exemple :

```
; Limiter la feuille de calcul à 12 colonnes  
CALC_SETNBCOLUMNS HCALC%, 12
```

Voir aussi CALC_GETNBCOLUMNS%, CALC_SETNBROWS, CALC_GETNBROWS%

Instruction **CALC_SETNBROWS**

Définit le nombre de lignes pour la feuille de calcul.

Syntaxe	CALC_SETNBROWS <i>handle Calc, nb lignes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nb lignes	INT(2)	nombre de lignes

1. Le paramètre nb lignes est accepté s'il est compris entre 1 et CALC_MAX_ROWS% qui est le nombre maximum de lignes admissibles.
2. Cette instruction sera de préférence effectuée lors de l'initialisation de la feuille de calcul.
3. Si cette instruction est passée en cours d'utilisation :
 - a) si le nombre de lignes est augmenté : le contenu des nouvelles cellules sera nul, mais le contenu des lignes non affectées sera conservé,
 - b) si le nombre de lignes est diminué : le contenu des lignes non affectées sera conservé, et les lignes en surplus seront supprimées.

Exemple :

```
; Limiter la feuille de calcul à 20 lignes
CALC_SETNBROWS HCALC%, 20
```

Voir aussi CALC_GETNBROWS%, CALC_SETNBCOLUMNS, CALC_GETNBCOLUMNS%

Instruction **CALC_SETOVERRUN**

Positionne la propriété de débordement sur l'ensemble de la fenêtre.

Syntaxe	CALC_SETOVERRUN <i>handle Calc, overrun%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	overrun%	INT(1)	débordement du texte

Le paramètre overrun% peut prendre les valeurs suivantes :

Constantes	Valeurs	Description
OV_INACTIVE%	0	Pas de débordement.
OV_ACTIVE%	1	Débordement autorisé.
OV_INDETERMINATE%	2	Déterminé par la propriété de débordement de la feuille.

Voir aussi CALC_ISOVERRUN%, CALC_SETBLOCKOVERRUN, CALC_GETCELLOVERRUN%,
CALC_PRINTGETDISPLAYEDCELLSIZE, CALC_GETDISPLAYEDCELLSIZE

Instruction **CALC_SETPAGEBREAK**

Permet de définir un saut de page.

Syntaxe	CALC_SETPAGEBREAK <i>handle Calc, roworcol%, coor%, yorn%</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS-Calc
	roworcol%	INTEGER	saut de page à partir d'une colonne ou d'une ligne
	coor%	INT(2)	coordonnées de la colonne ou de la ligne
	yorn%	INTEGER	saut de page ou non

Voir aussi **CALC_ISPAGEBREAK%**

Instruction **CALC_SETSTATUSBAR**

Permet d'afficher (indic affichage à TRUE%) ou de cacher (indic affichage à FALSE%) la ligne de statut pour la feuille de calcul.

Syntaxe	CALC_SETSTATUSBAR <i>handle Calc, indic affichage</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic affichage	INT(1)	indicateur d'affichage

L'indicateur d'affichage contient les valeurs suivantes :

- 0 Barre d'état cachée
- 1 Barre d'état affichée
- 2 Barre d'état cachée avec saisie directe dans la feuille à l'aide d'un champ de saisie

Exemple :

```
; Inversion de l'affichage de la ligne de statut sur le menu correspondant
IF CALC_ISSTATUSBAR%(HCALC%)
CALC_SETSTATUSBAR HCALC%, FALSE%
ELSE
CALC_SETSTATUSBAR HCALC%, TRUE%
ENDIF
```

Voir aussi **CALC_ISSTATUSBAR%**

Instruction **CALC_SETTEXTCOLOR**

Modifie la couleur par défaut des caractères dans les cellules de la feuille de calcul.

Syntaxe	CALC_SETTEXTCOLOR <i>handle Calc, col texte</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	col texte	INT(1)	code couleur

1. Les cellules modifiées individuellement avec **CALC_SETBLOCKTEXTCOLOR** ne sont pas concernées par cette modification.
2. Les codes couleur correspondent aux constantes COL_ *%.
3. Attention à ne pas confondre avec [CALC_SETBACKCOLOR](#) qui permet de spécifier la couleur par défaut du fond des cellules et [CALC_SETDEFCOLOR](#) qui spécifie le fond de la page en dehors des cellules.

Exemple :

```
; Afficher le texte en bleu
CALC_SETTEXTCOLOR HCALC%, COL_BLUE%
```

Voir aussi [CALC_GETTEXTCOLOR%](#), [CALC_GETCELLTEXTCOL%](#), [CALC_SETBLOCKTEXTCOL](#), [CALC_GETBACKCOLOR%](#), [CALC_SETBACKCOLOR](#), [CALC_GETDEFCOLOR%](#), [CALC_SETDEFCOLOR](#)

Instruction **CALC_SETTHREAD**

Définit l'ordre de chaînage des cellules, c'est à dire l'ordre de passage automatique du focus par la touche [Tab] (les cellules sont définies dans liste).

Syntaxe	CALC_SETTHREAD <i>handle Calc, liste</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	liste	CSTRING(255)	liste des cellules et blocs

1. liste est un ensemble de blocs et de cellules, défini par des coordonnées de blocs et de cellules, chaque coordonnée étant séparée de la précédente par le caractère point virgule ";".
2. Cette instruction permet de faciliter et de rendre beaucoup plus efficace la saisie dans la feuille de calcul.
3. Lorsqu'un tel chaînage est défini sur une feuille, il est impossible, pour l'utilisateur final, de cliquer sur une cellule non incluse dans ce chaînage et donc de modifier les cellules autres que celles chaînées.
4. L'ordre du déplacement à l'intérieur d'un bloc est le "Z order", ordre de lecture occidentale (de gauche à droite et de haut en bas).

Exemple :

```
; Forcer le passage automatique du focus de la cellule R3C3 à la cellule  
R6C5  
; puis au bloc R6C9:R9C10, puis à la cellule R4C2 par appui de la touche  
[Tab]  
CALC_SETTHREAD HCALC%, "R3C3; R6C5; R6C9:R9C10; R4C2"
```

Voir aussi **CALC_GETTHREAD\$**

Instruction **CALC_SETTOPROWS**

Bloque les nb lignes du haut de la feuille de calcul.

Syntaxe	CALC_SETTOPROWS <i>handle Calc, nb lignes</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	nb lignes	INT(2)	nombre de lignes

1. Cette instruction sera utilisée pour garder visible les premières lignes de la feuille de calcul (contenant les titres par exemple) tout en travaillant dans les autres cellules.
2. Veiller à ne pas bloquer un nombre de lignes plus grand que n'en peut contenir la fenêtre, sinon la feuille de calcul sera bloquée verticalement.

Exemple :

```
; Garder visibles en permanence les lignes 1 et 2
CALC_SETTOPROWS HCALC%, 2
```

Voir aussi **CALC_GETTOPROWS%**, **CALC_SETLEFTCOLUMNS%**, **CALC_GETLEFTCOLUMNS%**

Instruction **CALC_SETVARCELLWIDTHS**

Autorise (var à TRUE%) ou supprime (var à FALSE%) la possibilité de fixer avec CALC_SETWIDTH la largeur d'une cellule individuelle. Cette instruction est à appeler dans l'évènement INIT de la fenêtre NS Calc avant toute autre instruction. Elle nécessite de fixer ensuite le nombre de lignes (avec CALC_SETROWS) et le nombre de cellules pour chaque ligne (avec CALC_SETNBCELLS).

Syntaxe	CALC_SETVARCELLWIDTHS <i>handle Calc, var</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	var	INT(1)	indicateur de largeur variable

Il n'est pas possible d'afficher une numérotation des lignes ou colonnes dans le cas d'un tableau de cellules à largeur variable.

Exemple :

```
; Séquence INIT
CALC_SETVARCELLWIDTHS HCALC%, TRUE%
CALC_SETNBROWS HCALC%, 3
CALC_SETNBCELLS HCALC%, 1, 3
CALC_SETNBCELLS HCALC%, 2, 5
CALC_SETNBCELLS HCALC%, 3, 2
CALC_SETWIDTH HCALC%, 'R2C1', 100
```

Voir aussi [CALC_SETNBCELLS](#)

Instruction **CALC_SETWIDTH**

Fixe la largeur des cellules des colonnes num colonnes avec la valeur largeur cell, exprimée en pixels.

Syntaxe	CALC_SETWIDTH <i>handle Calc, num colonnes, largeur cell</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	num colonnes	CSTRING(255)	numéro des colonnes
	largeur cell	INT(2)	largeur des colonnes

1. Un ensemble de colonnes adjacentes sera défini par : "Cn:Cm".
2. Si les colonnes à modifier ne sont pas côte à côte, le séparateur utilisé dans num colonnes est le point virgule ";".
3. Dans le cas d'un tableau de cellules à taille variable (cf CALC_SETVARCELLWIDTHS), CALC_SETWIDTH permet de changer la largeur d'une cellule individuelle, en spécifiant dans num colonnes les coordonnées RnCm de cette cellule.

Exemple :

```
; Fixer la largeur des colonnes 2, 3, 4, 5 et 7 à 125 pixels
CALC_SETWIDTH HCALC%, "C2:C5; C7", 125
```

Voir aussi CALC_GETWIDTH%, CALC_GETDEFWIDTH%, CALC_SETHEIGHT

Instruction **CALC_SETZOOM**

Affiche une feuille avec un facteur de réduction ou d'agrandissement.

Syntaxe	CALC_SETZOOM <i>handle%, zoom</i>		
Paramètres	handle%	INT(4)	handle fenêtre NS Calc
	zoom	INT(2)	facteur de réduction ou d'agrandissement

Voir aussi **CALC_GETZOOM%**

Instruction **CALC_SORT**

Effectue le tri alphabétique (ordre ASCII des caractères) du bloc de coordonnées coord bloc, ordonné sur le contenu des lignes ou des colonnes, en fonction du paramètre critères.

Le paramètre critères contient les numéros de colonnes (Cm) ou de lignes (Rn) ; les critères sont séparés par un point virgule ";".

Syntaxe	CALC_SORT <i>handle Calc, critères, coord bloc</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	critères	CSTRING(255)	critères de tri
	coord bloc	CSTRING(255)	coordonnées du bloc

Pour effectuer un tri dans l'ordre décroissant, il suffit de faire précéder le numéro de ligne ou de colonne par le signe négation (" ").

Exemple :

```
; Tri alphabétique du bloc sélectionné sur les deux premières colonnes
CALC_SORT HCALC%, "C1; C2", CALC_SELECTION$ (HCALC%)
```

Instruction **CALC_UNDO**

Annule l'effet de l'action la plus récente, lorsque cela est possible.

Les actions pouvant être annulées sont celles qui concernent le presse papiers (couper, coller, etc.) et la saisie clavier.

Syntaxe	CALC_UNDO <i>handle Calc</i>		
Paramètre	handle Calc	INT(4)	handle fenêtre NS Calc

Exemple :

```
IF CALC_ISUNDO% (HCALC%)  
CALC_UNDO HCALC%  
ENDIF
```

Voir aussi [CALC_ISUNDO%](#), [CALC_COPY](#), [CALC_PASTE](#), [CALC_PASTESPECIAL](#)

Instruction **CALC_UNLOCKUNDOREDO**

Verrouille la possibilité des fonctionnalités "refaire" et "annuler" sur la dernière action effectuée.

Syntaxe	CALC_UNLOCKUNDOREDO <i>handle Calc, lock%</i>		
Paramètres	handle Calc	INT (4)	handle fenêtre NS Calc
	lock%	INT (4)	verrouillage des actions "refaire" et "annuler"

Voir aussi CALC_ISUNDO%, CALC_ISREDO%, CALC_REDO, CALC_ISUNLOCKEDUNDOREDO%, CALC_COPY, CALC_PASTE, CALC_PASTESPECIAL

Instruction **CALC_UPDATE**

Autorise ou interdit la mise à jour graphique de la fenêtre NS Calc en fonction de la valeur indic action passée en paramètre.

- Si indic action vaut TRUE%, toute modification dans les cellules de la fenêtre NS Calc est immédiatement répercutée à l'écran.
- Si indic action vaut FALSE%, aucune mise à jour de la fenêtre NS Calc n'est effectuée.

Syntaxe	CALC_UPDATE <i>handle Calc, indic action</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	indic action	INT(1)	indicateur d'action à effectuer

1. L'interdiction de mise à jour entre chaque entrée de donnée pourra être utilisée pour éviter de freiner la saisie si les temps de calcul sont importants.
2. A l'initialisation de la feuille de calcul, la mise à jour est activée.

Exemple :

```
; Activation de la mise à jour des cellules  
CALC_UPDATE HCALC%, TRUE%
```

Voir aussi **CALC_GETUPDATE%**, **UPDATE**, **NOUPDATE** (*Manuel de Programmation NCL*)

Instruction CALC_XY2RC

Renvoie la cellule (ligne / colonne) à partir de ses coordonnées.

Syntaxe	CALC_XY2RC <i>handle Calc, X, Y, Ligne, Colonne</i>		
Paramètres	handle Calc	INT(4)	handle fenêtre NS Calc
	X	INT	
	Y	INT	
	Ligne	INT(2)	ligne dont la cellule est à renvoyer
	Colonne	INT(2)	colonne dont la cellule est à renvoyer

Exemple :

```
;Evenement MouseMove
local x%, y%, R%(2), C%(2)
x% = param1%
y% = param2%
CALC_XY2RC self%, x%, y%, R%, C%
```

Segments Ns-Calc

Segment BORDERSTYLE

Utilisé en paramètre des instructions CALC_GETCELLBORDER et CALC_SETBLOCKBORDER, ce segment définit (ou retourne) la bordure de la cellule ou du bloc concerné(e) en spécifiant pour chaque bord, les type, largeur et couleur du trait.

- LeftLineKind, RightLineKind, TopLineKind et BottomLineKind indiquent respectivement le type des bords gauche, droit, supérieur et inférieur. Ce type est spécifié par une des constantes GLK_ *% (bibliothèque NSGRAPH).
- LeftLineWidth, RightLineWidth, TopLineWidth et BottomLineWidth indiquent respectivement la largeur du trait des bords gauche, droit, supérieur et inférieur. Cette largeur est comprise entre 1 et 3.
- LeftLineColor, RightLineColor, TopLineColor et BottomLineColor indiquent respectivement la couleur des bords gauche, droit, supérieur et inférieur. Cette couleur est spécifiée par une des constantes COL_ *% (langage NCL).

Syntaxe	Segment BORDERSTYLE
	LeftLineKind
	LeftLineWidth
	LeftLineColor
	RightLineKind
	RightLineWidth
	RightLineColor
	TopLineKind
	TopLineWidth
	TopLineColor
	BottomLineKind
	BottomLineWidth
	BottomLineColor
	EndSegment

Détail :

Champ	Type	Description
Bord gauche du cadre		
LeftLineKind	INT(1)	Type de trait
LeftLineWidth	INT(1)	Largeur du trait
LeftLineColor	INT(1)	Couleur du trait
Bord droit du cadre		
RightLineKind	INT(1)	Type de trait
RightLineWidth	INT(1)	Largeur du trait

RightLineColor	INT(1)	Couleur du trait
Bord supérieur du cadre		
TopLineKind	INT(1)	Type de trait
TopLineWidth	INT(1)	Largeur du trait
TopLineColor	INT(1)	Couleur du trait
Bord inférieur du cadre		
BottomLineKind		Type de trait
BottomLineWidth		Largeur du trait
BottomLineColor		Couleur du trait

Exemple :

```

LOCAL BORDERSTYLE Style_Bordure

; Bordure en pointillé
MOVE GLK_ALTERNATE% TO Style_Bordure.LeftLineKind
MOVE GLK_ALTERNATE% TO Style_Bordure.RightLineKind
MOVE GLK_ALTERNATE% TO Style_Bordure.TopLineKind
MOVE GLK_ALTERNATE% TO Style_Bordure.BottomLineKind

; Largeur du trait : 1 pour les bords gauche et droit, 2 pour les bords
inf. et sup.
MOVE 1 TO Style_Bordure.LeftLineWidth
MOVE 1 TO Style_Bordure.RightLineWidth
MOVE 2 TO Style_Bordure.TopLineWidth
MOVE 2 TO Style_Bordure.BottomLineWidth

; Couleur : Cyan pour les bords gauche et inférieur, Magenta pour les bords
droit et supérieur
MOVE COL_CYAN% TO Style_Bordure.LeftLineKind
MOVE COL_MAGENTA% TO Style_Bordure.RightLineKind
MOVE COL_MAGENTA% TO Style_Bordure.TopLineKind
MOVE COL_CYAN% TO Style_Bordure.BottomLineKind

CALC_SETBLOCKBORDER HCALC%, "R2C2:R4C5", Style_Bordure

```

Voir aussi [CALC_GETCELLBORDER](#), [CALC_SETBLOCKBORDER](#)

Segment CALCPAGEINFO

Ce segment donne des informations sur la page. Il est utilisé avec la fonction CALC_PRINTGETPAGEINFO%.

Syntaxe	SEGMENT CALCPAGEINFO INT CLEFT INT RTOP INT CRIGHT INT RBOTTOM INT USRWIDTH INT USRHEIGHT INT PRNWIDTH INT PRNHEIGHT ENDSEGMENT
----------------	--

Voir aussi CALC_PRINTGETPAGEINFO%.

Segment CTRLSTYLE

Définit le type de contrôle utilisé dans la ligne de statut pour la saisie des cellules.

Syntaxe	Segment CTRLSTYLE Ctype Cdata EndSegment
----------------	---

Détail :

Champ	Type	Description
Ctype	INT	type du contrôle de saisie
Cdata	INT	données affectées au contrôle

Voir aussi Constantes CCALC_FMT_*, CALC_SETBLOCKEDITCTRL

Segment FONTSTYLE

Utilisé en paramètre des instructions de formatage de la feuille de calcul, ce segment définit (ou retourne) les caractéristiques de la police.

Syntaxe	Segment FONTSTYLE Name Size Sels EndSegment
----------------	---

Détail :

Champ	Type	Description
Name	CSTRING	nom de la police
Size	INT	taille de la police
Sels	INT	style de la police (gras, italique, ...)

Voir aussi Constantes CCALC_FMT_*%, CALC_SETBLOCKFONT

Segment LINKSTYLE

Syntaxe	Segment LINKSTYLE StartKind EndKind LineKind LineColor LinkKind LineWidth EndSegment
----------------	---

Détail :

Champ	Type	Description
StartKind	INT(2)	position du début du lien
EndKind	INT(2)	position de la fin du lien
LineKind	INT(2)	aspect du trait
LineColor	INT(2)	couleur du trait
LinkKind	INT(2)	type du lien
LineWidth	INT(2)	largeur du lien

1. Utilisé en paramètre de l'instruction CALC_ADDENHANCEDLINK, ce segment définit (ou retourne) les caractéristiques d'un lien entre deux cellules.
2. StartKind et EndKind indiquent respectivement la position du début du trait et la position de la fin du trait sur les cellules. Ces positions sont spécifiées par une des constantes CLP_*.
3. LineKind indique le type ou aspect du trait représentant le lien. Ce type est spécifié par une des constantes GLK_*. (bibliothèque NSGRAPH).
4. LineColor indique la couleur du trait représentant le lien. Cette couleur est spécifiée par une des constantes COL_*. (langage NCL).
5. LinkKind indique le type du lien (droit, cassé, etc.). Ce type est une des constantes CLS_*.
6. LinkWidth indique la largeur du trait représentant le lien. Cette largeur est comprise entre 1 et 16. Si CLP_CORNER% est positionné, LinkWidth est égal à 1.

Exemple :

```

LOCAL LINKSTYLE Style_Lien

; Début du lien sur le bord droit de la cellule "R2C2"
MOVE CLP_VERTICAL% TO Style_Lien.StartKind
; Fin sur le coin supérieur gauche de la cellule "R5C4"
MOVE CLP_CORNER% TO Style_Lien.EndKind
; Trait en pointillé
MOVE GLK_ALTERNATE% TO Style_Lien.LineKind
; Trait Cyan
MOVE COL_CYAN% TO Style_Lien.LineColor

```

```
; Trait 'cassé'  
MOVE CLS_BROKEN% TO Style_Lien.LinkKind  
; Trait d'une largeur de 2 pixels  
MOVE 2 TO Style_Lien.LineWidth  
  
CALC_ADDENHANCEDLINK HCALC%, 2, 2, 5, 4, Style_Lien
```

Voir aussi Constantes CLP_*, CLS_*, CALC_ADDENHANCEDLINK

Segment MSGDRAWMARGIN

Ce segment est utilisé avec l'instruction CALC_MAPUSERMSG. Ce segment apporte des informations sur l'impression (zone à imprimer, zoom, marge, numéro de page, ...).

Syntaxe	SEGMENT MSGDRAWMARGIN HCALC HPS LSCRORPRN LMETRICS XRATIO YRATIO XPAGE YPAGE LEFTCOL TOPROW RIGHTCOL BOTTOMROW LX LY LW LH NUMPAGE ENDSEGMENT
---------	---

Détail :

Champ	Type	Description
HCALC	INT	handle de la fenêtre Calc
HPS	INT	Presentation Space
LSCRORPRN	INT	1: imprimante (c'est toujours le cas)
LMETRICS	INT	métrique utilisée par l'imprimante (toujours en pixel=8) CCALC_UM_PELS%
XRATIO	INT	zoom en X
YRATIO	INT	zoom en Y. Les deux zooms sont identiques aux valeurs données dans l'instruction CALC_PRINT*% au paramètre zoom%
XPAGE	INT	référence du bloc imprimé en X et Y (le document est découpé en bloc de cellules pour être imprimé)

YPAGE	INT	référence du bloc imprimé en X et Y (le document est découpé en bloc de cellules pour être imprimé)
LEFTCOL	INT	Coordonnées de la colonne de gauche du bloc de cellules imprimé
TOPROW	INT	Coordonnées de la ligne du haut du bloc de cellules imprimé
RIGHTCOL	INT	Coordonnées de la colonne de droite du bloc de cellules imprimé
BOTTOMROW	INT	Coordonnées de la ligne du bas du bloc de cellules imprimé
LX	INT	Coordonnées X de la marge
LY	INT	Coordonnées Y de la marge
LW	INT	Coordonnées de la largeur de la marge
LH	INT	Coordonnées de la hauteur de la marge
NUMPAGE	INT	Numéro de page

Voir aussi CALC MAPUSERMSG, constantes CALC MSG DRAW *%

Constantes Ns-Calc

Constantes CALC_MAX_*

Nombre maximum de lignes et de colonnes pour une feuille de calcul.

Syntaxe	Déclaration interne
CALC_MAX_COLUMNS%	16249
CALC_MAX_ROWS%	16249

Voir aussi CALC_GETNBCOLUMNS%, CALC_GETNBROWS%, CALC_SETNBCOLUMNS, CALC_SETNBROWS

Constantes CALC_MSG_DRAW_*

Ces constantes sont utilisées avec l'instruction CALC_MAPUSERMSG. Elles permettent de définir l'impression des hauts et bas de page, ainsi que les marges gauche et droite de chaque page.

Syntaxe	Déclaration interne	Description
CCALC_MSG_DRAW_TOPMARGIN%	11	Impression de l'en-tête de la page
CCALC_MSG_DRAW_LEFTMARGIN%	12	Impression de la marge gauche
CCALC_MSG_DRAW_RIGHTMARGIN%	13	Impression de la marge droite
CCALC_MSG_DRAW_BOTTOMMARGIN%	14	Impression du bas de page

Ce qui change pour ces 4 événements, ce sont les coordonnées de la zone en question du segment MSGDRAWMARGIN, soit respectivement :

- INT LX
- INT LY
- INT LW
- INT LH

Ces coordonnées sont mises à jour en fonction de la zone en question.

Par exemple, pour CCALC_MSG_DRAW_TOPMARGIN% $X=0$; $Y=\text{hauteur de la page} - \text{hauteur de la marge haute} + 1$ (+1 car les coordonnées $X*Y$ commencent à 0); $W = \text{largeur de la page}$, $H = \text{hauteur de la marge haute}$, alors la hauteur de la page entière = $Y + H - 1$

Voir aussi CALC_MAPUSERMSG, segment MSGDRAWMARGIN

Constantes CCALC_ENTER_*

Constantes utilisées avec l'instruction CALC_SETENTERDIRECTION. Elles indiquent à NS-CALC l'action à exécuter lors de l'appui sur la touche clavier [Entrée].

Syntaxe	Déclaration interne	Description
CCALC_ENTER_NONE%	0	Valide la saisie mais ne change pas la place de la cellule active
CCALC_ENTER_VERT%	1	[Entrée] Déplacement vers le bas – [Maj]+[Entrée] déplacement vers le haut (idem flèche haut/bas)
CCALC_ENTER_HORZ%	2	[Entrée] Déplacement vers la droite – [Maj]+[Entrée] déplacement vers la gauche (idem flèche droite/gauche)
CCALC_ENTER_TAB%	3	[Entrée] / [Maj]+[Entrée] permet de se déplacer dans l'ordre de chaînage des cellules défini par CALC_SETTHREAD, idem tab/back-tab
CCALC_ENTER_AUTO%	4	se déplace dans le même sens que le précédent déplacement, ainsi si on active la touche <flèche-haut> avant, la touche [Entrée] reproduira le même comportement que <flèche-haut>

La constante définie par défaut est CALC_ENTER_VERT%.

Voir aussi CALC_SETENTERDIRECTION

Constantes CCALC_FMT_*

Ces constantes sont utilisées avec les fonctions de formatage de la feuille de calcul.

Syntaxe	Déclaration interne	Description
CCALC_FMT_FORECOL%	0 ;@int	couleur d'avant-plan
CCALC_FMT_BACKCOL%	1 ;@int	couleur d'arrière-plan
CCALC_FMT_GRID%	2 ;@int	affichage de la grille
CCALC_FMT_BORDER%	3 ;@borderstyle	segment BORDERSTYLE permettant de paramétrer les bordures des feuilles de calcul
CCALC_FMT_FONT%	4 ;@fontstyle	segment FONTSTYLE permettant de paramétrer la police des feuilles de calcul
CCALC_FMT_HORZJUSTIF%	5 ;@int	justification horizontale du texte dans les cellules de la feuille de calcul
CCALC_FMT_VERTJUSTIF%	6 ;@int	justification verticale du texte dans les cellules de la feuille de calcul
CCALC_FMT_LOCKED%	7 ;@int	verrouillage de la feuille de calcul
CCALC_FMT_FORMAT%	8 ;@cstring	format de la feuille de calcul
CCALC_FMT_FILTER%	9 ;@cstring	filtrage de la feuille de calcul
CCALC_FMT_KIND%	10 ;@int	type de la feuille de calcul
CCALC_FMT_LENGTH%	11 ;@int	longueur de la feuille de calcul
CCALC_FMT_CTRLEDIT%	12 ;@ctrlstyle	segment CTRLSTYLE permettant de définir le type de contrôle utilisé pour la saisie des cellules
CCALC_FMT_OVERRUN%	13 ;@int	débordement du texte dans les cellules de la feuille de calcul
CCALC_FMT_PATTCOL%	14 ;@int	couleur de motif de la feuille de calcul
CCALC_FMT_PATTERN%	15 ;@int	motif de la feuille de calcul
CCALC_FMT_DWSPACE%	16 ;@int	espace par défaut entre la cellule et le texte

Voir aussi CALC GETFORMATIDATTR%, CALC SETFORMATIDATTR%, CALC CREATEFORMATID%, CALC SETBLOCKFORMATID%, CALC GETCELLFORMATID%, CALC GETDEFFORMATID%, CCALC_FMT_*, BORDERSTYLE, FONTSTYLE, CTRLSTYLE

Constantes CCALC_UM_*

Ces constantes sont utilisées dans les fonctions CALC_PRINTXSCR2PRN%, CALC_PRINTYSCR2PRN%, CALC_PRINTXPRN2SCR%, CALC_PRINTYPRN2SCR%, et précisent le type d'unités pixel et écran.

Syntaxe	Déclaration interne	Description
CCALC_UM_VIEWPELS%	32	résolution de l'écran
CCALC_UM_PELS%	8	résolution des fontes
CCALC_UM_LOMETRICS%	12	1/100 cm
CCALC_UM_HIMETRICS%	16	1/1000 cm
CCALC_UM_LOENGLISH%	20	1/100 inch
CCALC_UM_HIENGLISH%	24	1/1000 inch
CCALC_UM_TWIPS%	28	1/1440 inch

1. Les unités CCALC_UM_PELS% et CCALC_UM_VIEWPELS% sont des unités pixels.
2. CCALC_UM_VIEWPELS% se base sur la résolution de l'écran.
3. CCALC_UM_PELS% est la résolution de référence de "NS-CALC". Elle se base sur la résolution des fontes. Elle est légèrement plus grande que la résolution définie par la constante "CCALC_UM_VIEWPELS%" afin de permettre une présentation optimum des polices de caractères.

Voir aussi CALC_PRINTXSCR2PRN%, CALC_PRINTYSCR2PRN%, CALC_PRINTXPRN2SCR%, CALC_PRINTYPRN2SCR%

Constantes CE_ERR%

Ces constantes sont utilisées avec les fonctions `CALC_GETONERROR%` et `CALC_SETONERROR%`. Elles contiennent un texte d'au plus sept caractères.

Syntaxe	Déclaration interne
<code>CE_ERRERR%</code>	1 !ERR
<code>CE_ERRFORMAT%</code>	2 !FMT
<code>CE_ERRNUM%</code>	3 !NUM
<code>CE_ERRREF%</code>	4 !REF
<code>CE_ERRDIV0%</code>	5 !DIV/0
<code>CE_ERRRANGE%</code>	6 !RANGE
<code>CE_ERRNAME%</code>	7 !NAME?
<code>CE_ERRVALUE%</code>	8 !VALUE
<code>CE_ERRFLOAT%</code>	9 !FLOAT
<code>CE_ERRSTACK%</code>	10 !STACK
<code>CE_ERRFORMULA%</code>	11 !ERROR
<code>CE_ERRDSPNUM%</code>	12 ; display number : number too large

Voir aussi `CALC_GETONERROR%`, `CALC_SETONERROR%`

Constantes CK_*


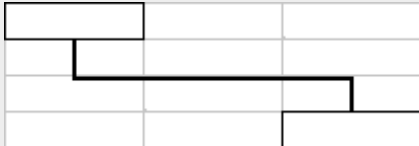
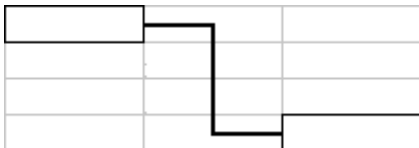
Ces constantes sont utilisées dans le paramètre type des instructions `CALC_SETBLOCKKIND`, `CALC_GETFORMAT` et `CALC_SETFORMAT`, ou sont retournées par la fonction `CALC_GETCELLKIND%`, et précisent les formats de données des cellules.

Syntaxe	Déclaration interne	Description
<code>CK_DEFAULT%</code>	0	accepte n'importe quelle donnée en saisie.
<code>CK_FORMULA%</code>	3	n'accepte que la saisie de formules (commençant par le signe égal "=").
<code>CK_NUMBER%</code>	1	n'accepte que la saisie de caractères numériques.
<code>CK_TEXT%</code>	2	n'accepte que la saisie de caractères alphanumériques (pas de caractères de formules).
<code>CK_DATE%</code>	4	n'accepte que la saisie de date.
<code>CK_TIME%</code>	5	n'accepte que la saisie d'heures.

Voir aussi `CALC_GETCELLKIND%`, `CALC_SETBLOCKKIND`, `CALC_GETFORMAT`, `CALC_SETFORMAT`

Constantes CLP_**%

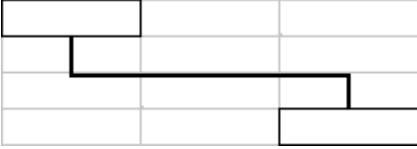

Ces constantes sont utilisées dans les champs StartKind et EndKind du segment style lien de l'instruction CALC_ADDENHANCEDLINK, et précisent l'emplacement du point de départ ou d'arrivée du lien sur le bord des cellules.

Syntaxe	Déclaration interne	Description
CLP_CORNER%	2	le lien part (ou arrive) d'un coin de la cellule. 
CLP_HORIZONTAL%	1	le lien part (ou arrive) d'un bord horizontal de la cellule, soit le bord supérieur ou le bord inférieur 
CLP_VERTICAL%	0	le lien part (ou arrive) d'un bord vertical de la cellule, soit le bord gauche ou le bord droit 

Voir aussi [CALC_ADDENHANCEDLINK](#)

Constantes CLS_*

Ces constantes sont utilisées dans le champ LinkKind du segment style lien de l'instruction CALC_ADDENHANCEDLINK, et précisent la forme du lien.

Syntaxe	Déclaration interne	Description
CLS_BROKEN%	1	<p>le lien est représenté par des lignes brisées perpendiculaires.</p> 
CLS_STRAIGHT%	0	<p>le lien est représenté sous la forme d'une droite.</p> 

Voir aussi CALC_ADDENHANCEDLINK

Constantes CTRLDATA_*

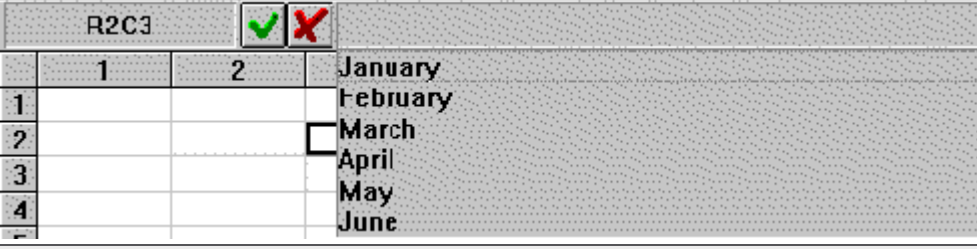
Ces constantes sont utilisées dans le tableau données des instructions CALC_GETCELLEDITCTRL et CALC_SETBLOCKEDITCTRL, et précisent le type de données qui sont affectées au contrôle utilisé pour la saisie dans la ligne de statut.

Syntaxe	Déclaration interne	Description
CTRLDATA_ARRAY%	2	données contenues dans un tableau (pour une combo box avec champ de saisie)
CTRLDATA_FILE%	1	données contenues dans un fichier (pour une combo box avec champ de saisie)
CTRLDATA_NONE%	0	pas de donnée (cas du champ de saisie)

Voir aussi CALC_GETCELLEDITCTRL, CALC_SETBLOCKEDITCTRL

Constantes EDTCTRL_*

Ces constantes sont utilisées dans le paramètre type contrôle des instructions CALC_GETCELLEDITCTRL et CALC_SETBLOCKEDITCTRL, et précisent le contrôle utilisé pour la saisie dans la ligne de statut.

Syntaxe	Déclaration interne	Description
EDTCTRL_CBE%	2	<p>pour une combo box avec champ de saisie</p> 
EDTCTRL_EF%	0	<p>pour un champ de saisie</p> 
EDTCTRL_CB%	1	pour une combo box classique

Voir aussi [CALC_GETCELLEDITCTRL](#), [CALC_SETBLOCKEDITCTRL](#)

Constantes FF_*

Ces constantes sont utilisées avec la fonction CALC_SAVE% pour spécifier le format du fichier sauvegardé.

Syntaxe	Déclaration interne	Description
FF_NSCALC%	0	sauvegarde le fichier dans sa forme la plus complète (en intégrant les largeurs/hauteurs de cellules, les formats, etc.), avec le jeu de caractères par défaut.
FF_NSCALC_ANSI%	32	précise le jeu de caractère utilisé : jeu de caractères ANSI utilisé sous Windows
FF_NSCALC_MAC%	48	précise le jeu de caractère utilisé : jeu de caractères utilisé sous Macintosh
FF_NSCALC_PC%	16	précise le jeu de caractère utilisé : jeu de caractères ASCII utilisé sous OS/2 et MS DOS
FF_TEXT%	1	sauvegarde le fichier sous forme compacte en ne retenant que la valeur des cellules : le caractère de tabulation sert de séparateur pour les cellules d'une même ligne, la séquence fin de ligne sert de séparateur pour les lignes, avec le jeu de caractères par défaut.
FF_TEXT_ANSI%	33	précise le jeu de caractère utilisé : jeu de caractères ANSI utilisé sous Windows
FF_TEXT_MAC%	49	précise le jeu de caractère utilisé : jeu de caractères utilisé sous Macintosh
FF_TEXT_PC%	17	précise le jeu de caractère utilisé : jeu de caractères ASCII utilisé sous OS/2 et MS DOS

Voir aussi CALC_SAVE%, CALC_GETFILEFORMAT%

Constantes GK_*

Ces constantes sont utilisées dans le paramètre cadre des instructions CALC_SETBLOCKGRID, CALC_GETFORMAT et CALC_SETFORMAT, ou sont retournées par la fonction CALC_GETCELLGRID%, et précisent le ou les bords des cellules sur lesquels est tracé le trait du cadre.

Syntaxe	Déclaration interne	Description
GK_ALTERNATE%	16	trace le bord correspondant en pointillés
GK_BOTTOM%	8	trace le bord inférieur des cellules
GK_LEFT%	1	trace le bord gauche des cellules
GK_RIGHT%	2	trace le bord droit des cellules
GK_TOP%	4	trace le bord supérieur des cellules

Il est possible de combiner plusieurs constantes GK_*.

Exemple : si cadre vaut 19, (bits 0, 1 et 4 positionnés à 1), il affectera les constantes GK_LEFT%, GK_RIGHT% et GK_ALTERNATE%.

Voir aussi CALC_GETCELLGRID%, CALC_SETBLOCKGRID, CALC_GETFORMAT, CALC_SETFORMAT

Constantes HJ_**%

Ces constantes sont utilisées dans le paramètre justif de l'instruction `CALC_SETBLOCKHORZJUSTIF` ou sont retournées par la fonction `CALC_GETCELLHORZJUSTIF%`, et précisent la justification horizontale dans les cellules.

Syntaxe	Déclaration interne	Description
<code>HJ_CENTERED%</code>	3	centre les données au milieu des cellules
<code>HJ_DEFAULT%</code>	0	justifie les données dans les cellules en fonction de leur type : à droite pour les valeurs numériques, à gauche pour l'alphanumérique
<code>HJ_LEFT%</code>	1	justifie à gauche les données dans les cellules
<code>HJ_RIGHT%</code>	2	justifie à droite les données dans les cellules
<code>HJ_MCENTERED%</code>	4	justifie horizontalement et centralement sur plusieurs cellules

Voir aussi `CALC_GETCELLHORZJUSTIF%`, `CALC_SETBLOCKHORZJUSTIF`

Constantes JF_*

Ces constantes sont utilisées dans le paramètre justif des instructions CALC_GETFORMAT et CALC_SETFORMAT.

Syntaxe	Déclaration interne	Description
JF_CENTERED%	3	centre les données au milieu des cellules
JF_DEFAULT%	0	justifie les données d'une cellule en fonction de leur type : à droite pour les valeurs numériques, à gauche pour l'alphanumérique
JF_LEFT%	1	justifie à gauche les données dans les cellules
JF_RIGHT%	2	justifie à droite les données dans les cellules

Voir aussi CALC_GETFORMAT, CALC_SETFORMAT

Constantes PK_*

Ces constantes sont utilisées dans le paramètre param imp des fonctions CALC_PRINT%, CALC_PRINTFILE%, CALC_GETPRINTFILENBPAGES% et CALC_GETPRINTNBPAGES%, et précisent différentes caractéristiques de l'impression (justification, impression ou non du titre, impression ou non des numéros de pages, etc.).

Syntaxe	Déclaration interne	Description
PK_BOTTOM%	16	la feuille de calcul (donc le tableau correspondant) sera respectivement imprimée en bas, à gauche, à droite, en haut de la page, ou centré horizontalement ou verticalement sur la page
PK_LEFT%	1	
PK_RIGHT%	2	
PK_TOP%	8	
PK_HCENTER%	4	
PK_VCENTER%	32	
PK_TITLE%	128	le titre (spécifié dans le paramètre titre de ces mêmes fonctions)
PK_GRID%	256	la grille est imprimée sur le tableau
PK_PAGENUMBER%	64	les numéros de page sont imprimés ; ils sont placés en bas de chaque page et centré horizontalement
PK_PRINTFILE%	512	impression d'une feuille à partir d'un document sur un disque
PK_INFOJOB%	1024	mise à zéro du titre du job d'impression affiché par le Gestionnaire d'impression
PK_GENERICFONTS%	16384	Utilisation des fontes génériques (fontes fonctionnant aussi bien sur l'imprimante que l'écran). OS/2 PM uniquement
PK_ALLFONTS%	32768	Utilisation de toutes les fontes disponibles. OS/2 PM uniquement

Il est possible de combiner l'une des 6 premières constantes décrites ci dessus avec les trois autres constantes PK_*.

Exemple : si le paramètre param imp vaut 193, (bits 0, 6 et 7 positionnés à 1), il affectera les constantes PK_LEFT%, PK_PAGENUMBER% et PK_TITLE%.

Voir aussi [CALC_GETPRINTFILENBPAGES%](#), [CALC_GETPRINTNBPAGES%](#), [CALC_PRINT%](#), [CALC_PRINTFILE%](#)

Constantes PT_*

Ces constantes sont utilisées dans le paramètre type des instructions CALC_CLEARSPCIAL et CALC_PASTESPECIAL, et précisent quels éléments de la cellule doivent être supprimés ou quelles actions doivent être effectuées lors du collage.

Syntaxe	Déclaration interne	Description
PT_ADD%	16	Pour CALC_PASTESPECIAL : ajoute algébriquement le contenu d'une cellule du presse papiers dans la cellule correspondante.
PT_ALL%	1	Pour CALC_CLEARSPCIAL : toute les cellules sont effacées. Pour CALC_PASTESPECIAL : copie la totalité d'une cellule contenue dans le presse papiers dans la cellule correspondante.
PT_DIV%	128	Pour CALC_PASTESPECIAL : divise algébriquement le contenu d'une cellule de la feuille par la cellule correspondante dans le presse papiers (à condition que les deux cellules soient numériques).
PT_FORMAT%	4	Pour CALC_CLEARSPCIAL : le format des cellules est effacé, c'est à dire que ces cellules retrouvent le format par défaut. Pour CALC_PASTESPECIAL : affecte le format d'une cellule contenue dans le presse papiers à la cellule correspondante.
PT_FORMULA%	8	Pour CALC_CLEARSPCIAL : seules les cellules contenant des formules sont effacées. Pour CALC_PASTESPECIAL : copie la formule d'une cellule contenue dans le presse papiers dans la cellule correspondante.
PT_MULT%	64	Pour CALC_PASTESPECIAL : multiplie algébriquement le contenu d'une cellule du presse papiers par la cellule correspondante (à condition que les deux cellules soient numériques).
PT_SUB%	32	Pour CALC_PASTESPECIAL : soustrait algébriquement le contenu d'une cellule du presse papiers de la cellule correspondante.

PT_VALUE%	2	<p>Pour CALC_CLEARSPPECIAL : seules les cellules contenant des valeurs (texte ou nombre) sont effacées.</p> <p>Pour CALC_PASTESPECIAL : copie la valeur (texte ou nombre) d'une cellule contenue dans le presse papiers dans la cellule correspondante.</p>
PT_COMMENTS%	4096	Pour CALC_CLEARSPPECIAL : permet de supprimer les commentaires
PT_TEXT%	256	Pour CALC_PASTESPECIAL : copie le texte d'une cellule contenue dans le presse papiers dans la cellule correspondante.
PT_ZERONUM%	512	Pour CALC_CLEARSPPECIAL : permet de supprimer le numéro zéro
PT_BLANKNUM%	1024	Pour CALC_CLEARSPPECIAL : permet de supprimer les blancs

Voir aussi CALC_CLEARSPPECIAL, CALC_PASTESPECIAL

Constantes VJ_*

Ces constantes sont utilisées dans le paramètre justif de l'instruction `CALC_SETBLOCKVERTJUSTIF` ou sont retournées par la fonction `CALC_GETCELLVERTJUSTIF%`, et précisent la justification verticale dans les cellules.

Syntaxe	Déclaration interne	Description
VJ_DEFAULT%	0	applique la justification verticale par défaut (bas des cellules)
VJ_BOTTOM%	1	justifie en bas les données dans les cellules
VJ_TOP%	2	justifie en haut les données dans les cellules
VJ_CENTERED%	3	centre verticalement les données dans les cellules
VJ_WORDWRAP%	4	si le contenu de la cellule (plus particulièrement du texte) est trop grand, celui ci est 'coupé' de manière à apparaître sur plusieurs lignes

Voir aussi `CALC_GETCELLVERTJUSTIF%`, `CALC_SETBLOCKVERTJUSTIF`

INDEX

@

@ATAN 58
 @AVG 62
 @AVG1 63
 @COPY 72
 @COS 59
 @COUNT 64
 @COUNT1 65
 @CURRENTDATE 69
 @CURRENTDAY 70
 @CURRENTTIME 71
 @DELETE 73
 @EXP 48
 @FRAC 49
 @IF 68
 @INSERT 74
 @LN 50
 @MAX 66
 @MIN 67
 @PI 60
 @POS 75
 @POWER 51
 @RANDOM 52
 @ROUND 53
 @SEND 77
 @SQR 54
 @SQRT 55
 @STRING 76
 @SUM 56
 @TRUNC 57

A

A partir du clavier 35

Avec la souris 34

B

BORDERLINE 279

BORDERSTYLE 322

C

CALC_ADDENHANCEDLINK 221

CALC_ADDLINK 222

CALC_CHECKFORMULA 122

CALC_CLEAR 223

CALC_CLEARSPECIAL 224

CALC_COLUMN% 123

CALC_COMMAND% 124

CALC_CONVERTFORMULA\$ 125

CALC_COPY 225

CALC_CREATEFORMATID% 126

CALC_CREATENAME 226

CALC_DELETENAME 227

CALC_DISPLAYFORMULA 228

CALC_DISPLAYZERO 229

CALC_DRAW 230

CALC_EXTDRAW 231

CALC_FILLDOWN 233

CALC_FILLEFT 234

CALC_FILLRIGHT 235

CALC_FILLUP 236

CALC_FINDNAME\$ 127

CALC_FIRSTNAME% 128

CALC_GET\$ 129

CALC_GETALLOCATEDMEM% 130

CALC_GETBACKCOLOR% 131

CALC_GETBLOCKHEIGHT% 132

CALC_GETBLOCKWIDTH% 133

CALC_GETCELLBACKCOL% 134

CALC_GETCELLBORDER 237

CALC_GETCELLCOMMENT 238

CALC_GETCELLDWSPACE 135

CALC_GETCELLEDITCTRL 239

CALC_GETCELLEXTBORDER 241

CALC_GETCELLFILTER\$ 136

CALC_GETCELLFONT 242

CALC_GETCELLFORMAT\$ 138

CALC_GETCELLFORMATID% 139

CALC_GETCELLGRID% 140

CALC_GETCELLHORZJUSTIF% 141

CALC_GETCELLKIND% 142

CALC_GETCELLLENGTH% 143

CALC_GETCELLOVERRUN% 144

CALC_GETCELLPATTCOL% 145

CALC_GETCELLPATTERN% 146
CALC_GETCELLRGBCOLOR 243
CALC_GETCELLTEXTCOL% 147
CALC_GETCELLVERTJUSTIF% 148
CALC_GETCELLWIDTH% 149
CALC_GETDEFAULTHEADMODE
244
CALC_GETDEFCOLOR% 150
CALC_GETDEFCOLWIDTH% 151
CALC_GETDEFFORMATID% 152
CALC_GETDEFHEIGHT% 153
CALC_GETDEFROWHEIGHT% 154
CALC_GETDEFWIDTH% 155
CALC_GETDIMENSION 156
CALC_GETDISPLAYEDCELLSIZE
245
CALC_GETDRAWSIZE 246
CALC_GETDWSPACE 157
CALC_GETFILEFORMAT% 158
CALC_GETFORMAT 248
CALC_GETFORMATIDATTR% 159
CALC_GETHEIGHT% 160
CALC_GETLEFTCOLUMNS% 161
CALC_GETMARGINS 250
CALC_GETNBCELLS% 162
CALC_GETNBCOLUMNS% 163
CALC_GETNBROWS% 164
CALC_GETONERROR% 165
CALC_GETPRINTFILENBPAGES%
166
CALC_GETPRINTNBPAGES% 168
CALC_GETTEXT\$ 170
CALC_GETTEXTCOLOR% 171
CALC_GETTHREAD\$ 172
CALC_GETTOPROWS% 173
CALC_GETUPDATE% 174
CALC_GETVALUE\$ 175
CALC_GETWIDTH% 176
CALC_GETZOOM% 177
CALC_GOTO 251
CALC_ISCELLEEMPTY% 178
CALC_ISCELLLOCKED% 179
CALC_ISCLIPBOARD% 180
CALC_ISFORMULADISPLAYED%
181
CALC_ISGRID% 182
CALC_ISHEADING% 183
CALC_ISHEIGHTLOCKED% 184
CALC_ISLINK% 185
CALC_ISLOCKED% 186
CALC_ISOVERRUN% 187
CALC_ISPAGEBREAK% 188
CALC_ISREDO% 189
CALC_ISSTATUSBAR% 190
CALC_ISUNDO% 191
CALC_ISUNLOCKEDUNDOREDO%
192
CALC_ISVARCELLWIDTHS% 193
CALC_ISWIDTHLOCKED% 194
CALC_ISZERODISPLAYED% 195
CALC_LOAD% 196
CALC_LOCK 252
CALC_LOCKBLOCK 253
CALC_LOCKHEIGHT 254
CALC_LOCKWIDTH 255
CALC_MAPUSERMSG 256
CALC_MAX_COLUMNS% 331
CALC_MAX_ROWS% 331
CALC_MULTI_SELECT 257
CALC_MULTI_SELECTION\$ 197
CALC_NEXTNAME% 198
CALC_PASTE 258
CALC_PASTESPECIAL 259
CALC_PREPARE_PUT 260
CALC_PRINT% 199
CALC_PRINTFILE% 201
CALC_PRINTFINDPAGE% 203
CALC_PRINTGETDISPLAYEDCELLSI
ZE 261
CALC_PRINTGETPAGECOUNT%
204
CALC_PRINTGETPAGEINFO% 205
CALC_PRINTNEWFRAME% 206
CALC_PRINTPAGE% 207
CALC_PRINTREPAGINATE% 208
CALC_PRINTSTARTJOB% 209
CALC_PRINTSTOPJOB% 210

CALC_PRINTXPRN2SCR% 211
CALC_PRINTXSCR2PRN% 212
CALC_PRINTYPRN2SCR% 213
CALC_PRINTYSCR2PRN% 214
CALC_PUT 262
CALC_RC2XY 263
CALC_REDO 264
CALC_REMLINK 265
CALC_RESET 266
CALC_ROW% 215
CALC_SAVE% 216
CALC_SCROLL_DOWN 267
CALC_SCROLL_LEFT 268
CALC_SCROLL_RIGHT 269
CALC_SCROLL_UP 270
CALC_SELECT 271
CALC_SELECTION\$ 217
CALC_SETBACKCOLOR 272
CALC_SETBLOCKBACKCOL 273
CALC_SETBLOCKBORDER 274
CALC_SETBLOCKCOMMENT 275
CALC_SETBLOCKDWSPACE 276
CALC_SETBLOCKEDITCTRL 277
CALC_SETBLOCKEXTBORDER 279
CALC_SETBLOCKFILTER 280
CALC_SETBLOCKFONT 282
CALC_SETBLOCKFORMAT 283
CALC_SETBLOCKFORMATID% 218
CALC_SETBLOCKGRID 284
CALC_SETBLOCKHORZJUSTIF 285
CALC_SETBLOCKKIND 286
CALC_SETBLOCKLENGTH 287
CALC_SETBLOCKOVERRUN 288
CALC_SETBLOCKPATTCOL 289
CALC_SETBLOCKPATTERN 290
CALC_SETBLOCKRGBCOLOR 291
CALC_SETBLOCKTEXTCOL 292
CALC_SETBLOCKVERTJUSTIF 293
CALC_SETDEFAULTHEADMODE
294
CALC_SETDEFBCOLOR 295
CALC_SETDWSPACE 296
CALC_SETENTERDIRECTION 297
CALC_SETFORMAT 298
CALC_SETFORMATIDATTR% 219
CALC_SETGRID 300
CALC_SETHEADING 301
CALC_SETHEIGHT 302
CALC_SETLEFTCOLUMNS 303
CALC_SETMARGINS 304
CALC_SETNBCELLS 305
CALC_SETNBCOLUMNS 306
CALC_SETNBROWS 307
CALC_SETONERROR% 220
CALC_SETOVERRUN 308
CALC_SETPAGEBREAK 309
CALC_SETSTATUSBAR 310
CALC_SETTEXTCOLOR 311
CALC_SETTHREAD 312
CALC_SETTOPROWS 313
CALC_SETVARCELLWIDTHS 314
CALC_SETWIDTH 315
CALC_SETZOOM 316
CALC_SORT 317
CALC_UNDO 318
CALC_UNLOCKUNDOREDO 319
CALC_UPDATE 320
CALC_XY2RC 321
CALCPAGEINFO 324
CBMPDSPTYPE% 245
CCALC_ENTER_AUTO% 333
CCALC_ENTER_HORZ% 333
CCALC_ENTER_NONE% 333
CCALC_ENTER_TAB% 333
CCALC_ENTER_VERT% 333
CCALC_FMT_BACKCOL% 334
CCALC_FMT_BORDER% 334
CCALC_FMT_CTRLEDIT% 334
CCALC_FMT_DWSPACE% 334
CCALC_FMT_FILTER% 334
CCALC_FMT_FONT% 334
CCALC_FMT_FORECOL% 334
CCALC_FMT_FORMAT% 334
CCALC_FMT_GRID% 334
CCALC_FMT_HORZJUSTIF% 334
CCALC_FMT_KIND% 334

- CCALC_FMT_LENGTH% 334
- CCALC_FMT_LOCKED% 334
- CCALC_FMT_OVERRUN% 334
- CCALC_FMT_PATTCOL% 334
- CCALC_FMT_PATTERN% 334
- CCALC_FMT_VERTJUSTIF% 334
- CCALC_MSG_DRAW_BOTTOMMARGIN% 332
- CCALC_MSG_DRAW_LEFTMARGIN% 332
- CCALC_MSG_DRAW_RIGHTMARGIN% 332
- CCALC_MSG_DRAW_TOPMARGIN% 332
- CCALC_UM_HIENGLISH% 335
- CCALC_UM_HIMETRICS% 335
- CCALC_UM_LOENGLISH% 335
- CCALC_UM_LOMETRICS% 335
- CCALC_UM_PELS% 335
- CCALC_UM_TWIPS% 335
- CCALC_UM_VIEWPELS% 335
- CE_ERRDIV0% 336
- CE_ERRERR% 336
- CE_ERRFLOAT% 336
- CE_ERRFORMAT% 336
- CE_ERRNAME% 336
- CE_ERRNUM% 336
- CE_ERRRANGE% 336
- CE_ERRREF% 336
- CE_ERRSTACK% 336
- CE_ERRVALUE% 336
- CK_DATE% 337
- CK_DEFAULT% 337
- CK_FORMULA% 337
- CK_NUMBER% 337
- CK_TEXT% 337
- CK_TIME% 337
- CLP_CORNER% 338
- CLP_HORIZONTAL% 338
- CLP_VERTICAL% 338
- CLS_BROKEN% 339
- CLS_STRAIGHT% 339
- CNUMDSPTYPE% 245
- CTEXTDSPTYPE% 245
- CTEXTTOVERRUNDSPTYPE% 245
- CTRLDATA_ARRAY% 340
- CTRLDATA_FILE% 340
- CTRLDATA_NONE% 340
- CTRLSTYLE 325
- E
- EDTCTRL_CB% 341
- EDTCTRL_CBE% 341
- EDTCTRL_EF% 341
- Evénements Classiques 120
- EXTBORDERSTYLE 279
- F
- FF_NSCALC% 342
- FF_NSCALC_ANSI% 342
- FF_NSCALC_MAC% 342
- FF_NSCALC_PC% 342
- FF_TEXT% 342
- FF_TEXT_ANSI% 342
- FF_TEXT_MAC% 342
- FF_TEXT_PC% 342
- Fonctions et Instructions Classiques 121
- FONTSTYLE 326
- Formules de calcul 39
- G
- GK_ALTERNATE% 343
- GK_BOTTOM% 343
- GK_LEFT% 343
- GK_RIGHT% 343
- GK_TOP% 343
- H
- Hauteur 32
- HJ_CENTERED% 344
- HJ_DEFAULT% 344
- HJ_LEFT% 344
- HJ_MCENTERED% 344
- HJ_RIGHT% 344
- HMODE_ALPHA% 294
- HMODE_RC% 294
- I
- ID_INSERTRC% 124
- ID_SETFOCUS_STATUSBAR% 124
- IT_CELLBACKGROUND% 291

IT_CELLFOREGROUND% 291

IT_CELLTEXT% 291

J

JF_CENTERED% 345

JF_DEFAULT% 345

JF_LEFT% 345

JF_RIGHT% 345

L

Largeur 33

LINKSTYLE 327

M

MSGDRAWMARGIN 329

O

OV_ACTIVE% 308

OV_INACTIVE% 308

OV_INDETERMINATE% 308

P

PK_ *% 346

PK_ALLFONTS% 346

PK_BOTTOM% 346

PK_GENERICFONTS% 346

PK_GRID% 346

PK_HCENTER% 346

PK_INFOJOB% 346

PK_LEFT% 346

PK_PAGENUMBER% 346

PK_PRINTFILE% 346

PK_RIGHT% 346

PK_TITLE% 346

PK_TOP% 346

PK_VCENTER% 346

PT_ADD% 347

PT_ALL% 347

PT_BLANKNUM% 347

PT_COMMENTS% 347

PT_DIV% 347

PT_FORMAT% 347

PT_FORMULA% 347

PT_MULT% 347

PT_SUB% 347

PT_TEXT% 347

PT_VALUE% 347

PT_ZERONUM% 347

S

Sélectionner plusieurs blocs 36

U

Utilisation du Tableur 19

V

VJ_BOTTOM% 349

VJ_CENTERED% 349

VJ_DEFAULT% 349

VJ_TOP% 349

VJ_WORDWRAP% 349