



Manuel d'utilisation NS-WP

Table des matières

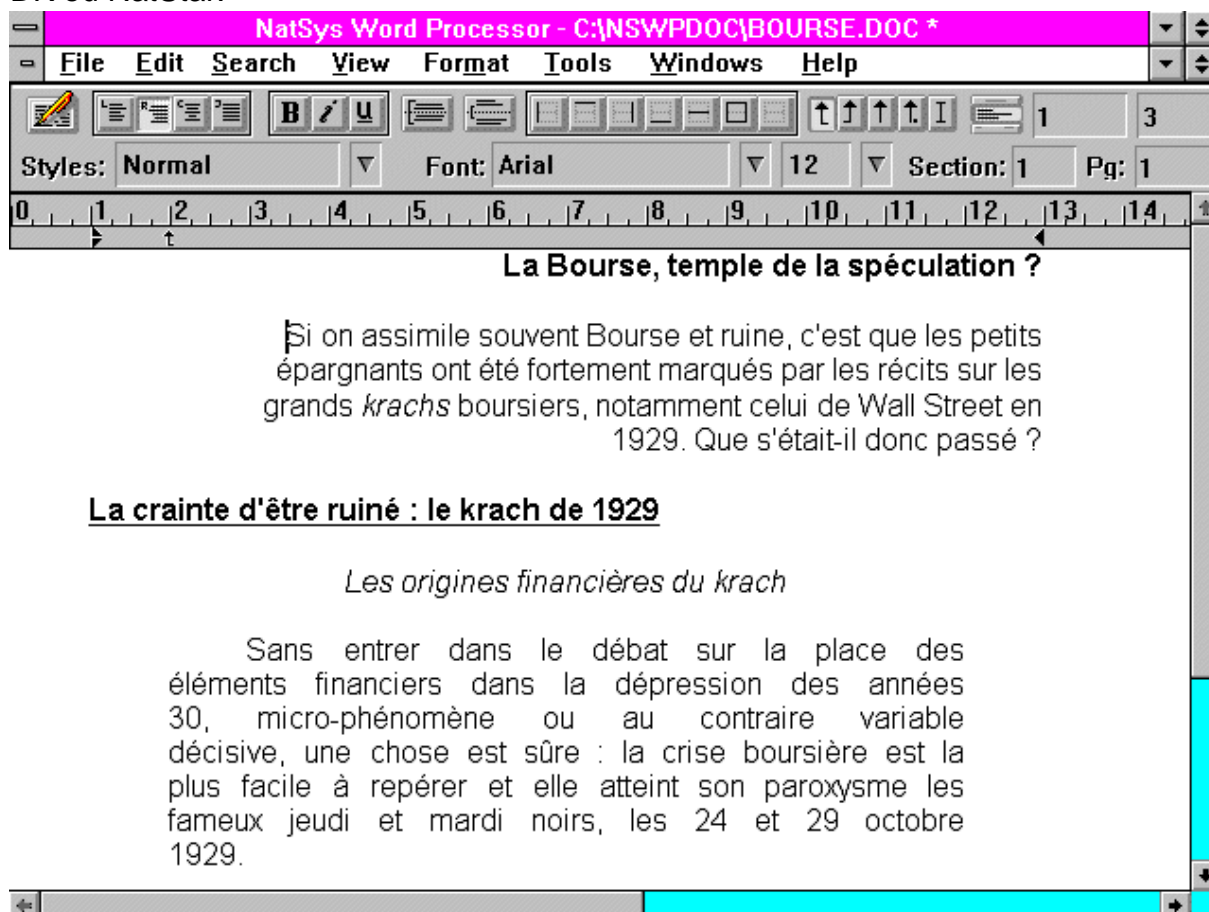
Introduction	7
Intégration dans NS-DK/NatStar	9
Installation.....	9
Intégration dans un projet.....	10
Génération	12
Livraison de l'application générée	13
Fonctionnalités de NS-WP	15
Introduction.....	15
Fonctionnalités de la librairie NS-WP	16
Affichage d'une barre d'outils standard et d'une règle graduée	17
Gestion du mode d'affichage du document	20
Mise en page du document.....	23
Formatage des paragraphes.....	28
Gestion des paragraphes	33
Formatage des caractères	34
Gestion des styles de caractères	35
Parcours du document	36
Insertion d'images bitmaps dans le texte.....	37
Insertion de bornes dans le texte.....	38
Gestion de dictionnaires et insertion de marqueurs	39
Gestion des modifications survenant dans la fenêtre NS WP	40
Choix d'un système de mesure	41
Chargement et sauvegarde d'un document	42
Création d'un document par fusion.....	43
Autres fonctionnalités	44
Présentation de l'API	45
Affichage des éléments de la fenêtre NS-WP	45
Gestion du mode d'affichage du document.....	46
Mise en page d'un document.....	47
Gestion des sections	48
Formatage d'un paragraphe (retraits, tabulations, alignements)	49
Association d'un entier à un paragraphe	51
Encadrement d'un paragraphe.....	52
Récupération du contenu d'un paragraphe	53
Recherche du handle d'un paragraphe.....	54
Formatage des caractères	55
Gestion des styles de caractères	57
Gestion du curseur	59
Gestion de la sélection	60

Insertion et suppression de texte	61
Gestion du presse-papiers	62
Gestion des coordonnées écran	63
Recherche et remplacement de texte	64
Parcours du document	65
Insertion d'images bitmaps.....	66
Insertion de bornes.....	67
Gestion d'un dictionnaire	68
Insertion de marqueurs.....	70
Gestion des modifications survenant dans la fenêtre NS-WP	71
Gestion du système de mesure	72
Chargement et sauvegarde d'un document.....	73
Impression.....	74
Fusion de documents.....	75
Gestion des erreurs	76
Gestion de la mémoire.....	77
Référence de l'API	79
Avertissements	79
Avertissement 1 : Au sujet du handle de la fenêtre NS-WP.....	79
Avertissement 2 : Au sujet des paramètres !V!.....	80
Événements	81
Événement CHANGED	81
Événement SELECTED.....	81
Événement TIMER.....	81
Événement USER0.....	81
Événement USER1.....	81
Événements USER0...USER15	81
Fonctions et Instructions NCL	83
Instructions LOAD et SAVE	83
Instructions UPDATE et NOUPDATE	83
Instructions LOCK et UNLOCK, fonction ISLOCKED%.....	83
Les APIs	84
Constantes	85
Fonctions	109
Instructions	212
Segments	329
Exemple d'un traitement de texte réalisé avec NS-WP : NSEDIT	347
Présentation de NSEDIT	347
Lancement de NSEDIT	349
Mise en oeuvre des fonctionnalités de NSEDIT	350
Fonctionnement de la barre d'outils	351

Gestion des styles de paragraphe	353
Génération automatique de la table des matières	356
Numérotation automatique	359
Composants de NSEDIT	362
Fenêtres.....	362
Librairies	366
Autres ressources	369
Utilisation de la librairie NS-WP au sein de NSEDIT.....	374
Limitations de NS-WP	393
Format de fichier NS-WP.....	395
Introduction.....	395
Les commandes simples	396
Les commandes groupes	397
Structure du document	398
Format de fichier en fusion	409
Introduction.....	409
Exemple de fusion	413
Raccourcis clavier.....	415
Glossaire	417
Index	421

INTRODUCTION

L'interface de programmation NS-WP (Nat System Word Processor) fournit les outils pour réaliser une fenêtre de traitement de texte intégrable dans une application NS-DK ou NatStar.



Une fenêtre de traitement de texte réalisée avec NS-WP

INTEGRATION DANS NS-DK/NATSTAR

Installation

L'interface de programmation NS WP complète les produits NS DK et NatStar.

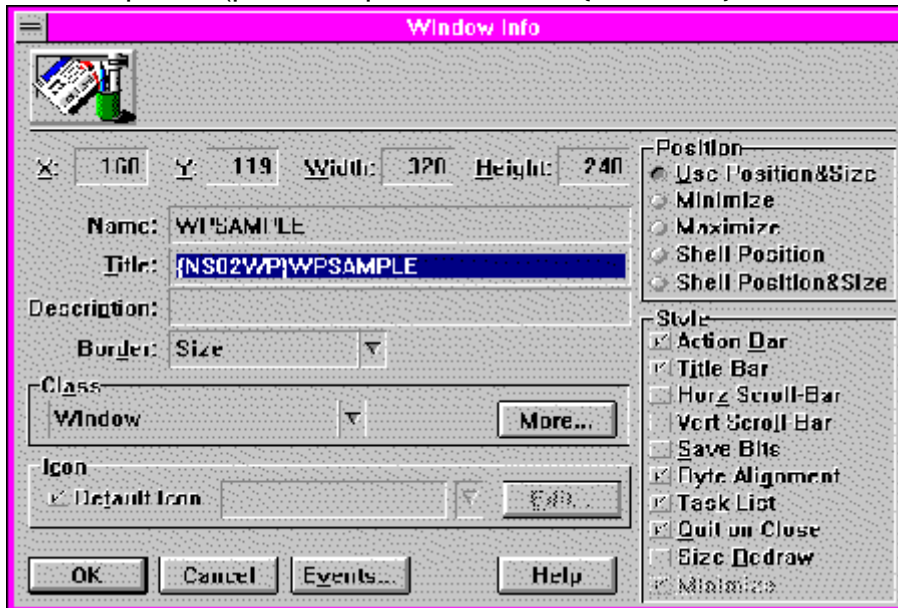
Le programme d'installation de NS-WP recherche le répertoire de NS-DK et, s'il existe, propose ce répertoire par défaut afin d'y copier les fichiers de NS WP.

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

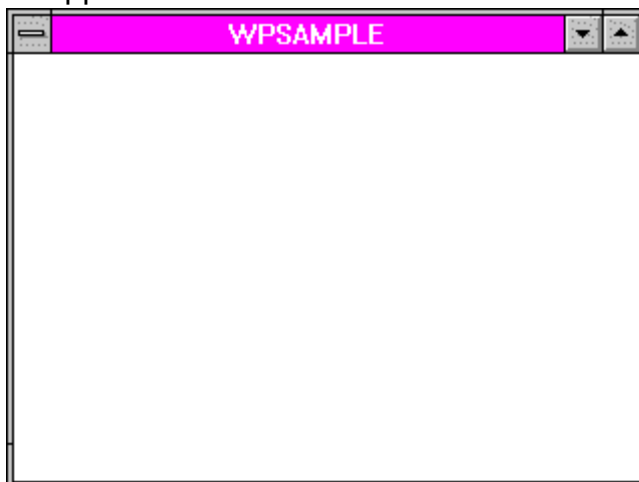
Pour créer une fenêtre NS-WP dans un projet NS-DK ou NatStar, il faut, depuis NS Design :

1. rajouter NS_WP dans Project Resources Libraries,
2. créer avec File/New une fenêtre de classe Window (par exemple WPSAMPLE),
3. ouvrir la boîte Window Info de cette fenêtre, et rajouter {NS02WP} en tête du champ Title (par exemple Title valant {NS02WP}WPSAMPLE).



Boîte info d'une fenêtre NS-WP

4. Testons maintenant cette fenêtre en faisant Run Start. La fenêtre suivante apparaît.



Une fenêtre NS-WP très simple

Cette fenêtre initiale comprend un titre (il apparaît sans les caractères {NS02WP} ajoutés dans le champ Title) et un plan de travail.

Elle peut déjà recevoir du texte dans le plan de travail.

Elle gère la souris :

Il est possible de se déplacer dans le plan de travail avec la souris et de sélectionner des zones de texte (en cliquant sur une limite du texte choisi et en "glissant" jusqu'à la limite opposée du texte). Un double-clic souris permet de sélectionner le mot situé sous le pointeur souris.

Elle gère des raccourcis clavier (voir annexe D).

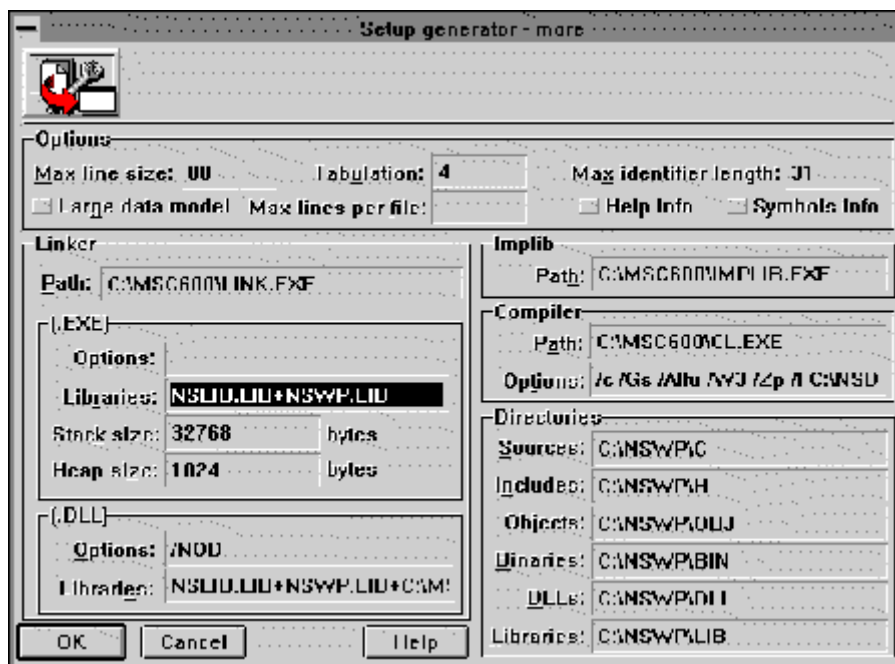
A vous maintenant de rajouter par programmation une barre de menus, une barre d'outils et une règle adaptées à vos besoins.

Génération

Pour générer une application NS-DK ou NatStar comportant une (ou plusieurs) fenêtre(s) NS WP, il est nécessaire de rajouter NSWP.LIB dans les librairies à utiliser lors de l'édition de liens. Pour cela :

1. Ouvrez la boîte Setup Generator More.
2. Ajoutez dans les champs Libraries des groupes [.EXE] et [.DLL] du groupe Linker :

NSWP.LIB



Ajout de NSWP.LIB dans les librairies pour le Link

Livraison de l'application générée

En plus de l'exécutable généré, les DLLs NS02WP.DLL et NS02WPTB.DLL doivent être livrées, avec les autres DLLs utilisées par l'application NS-DK ou NatStar (NS02LIB.DLL et autres NS*.DLL).

FONCTIONNALITES DE NS-WP

Introduction

La librairie NS-WP comprend un ensemble de fonctions et instructions spécialement conçues pour programmer une fenêtre NS-WP.

Ce chapitre décrit les fonctionnalités de cette librairie. Il ne décrit pas un traitement de texte, mais les outils qui permettent de construire un traitement de texte.

Quand il est écrit dans ce chapitre que vous pouvez formater les paragraphes avec NS-WP, cela signifie qu'il existe dans la librairie NS-WP une ou plusieurs fonctions agissant sur le formatage des paragraphes. A vous de créer la boîte de dialogue qui permettra à l'utilisateur de saisir les paramètres de formatage et qui fera elle-même appel à ces fonctions.

Vous trouverez les définitions des notions de traitement de texte évoquées dans ce chapitre dans le glossaire situé à la fin de ce manuel.

Les écrans visualisés montrent une application de ces fonctions, ils proviennent de l'exemple NSEDIT livré avec NS-WP.

Fonctionnalités de la librairie NS-WP

Affichage d'une barre d'outils standard et d'une règle graduée





Barre d'outils standard

La barre d'outils standard se présente sous la forme :






Elle comprend les boutons et les Combo box Entry field suivants :

- Boutons d'alignement

	Aligne à gauche
	Aligne à droite
	Centre
	Justifie


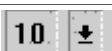
L'alignement s'applique au paragraphe courant ou aux paragraphes incluant une sélection de texte préalablement effectuée.

- Boutons appliquant des attributs d'enrichissement

	Met en gras
	Met en italique
	Souligne




Ces attributs s'appliquent aux caractères qui sont saisis par la suite ou à une sélection de texte préalablement effectuée.

- Combo box Entry field (CBE) de sélection de polices

	Liste les noms des polices de caractères disponibles
	Liste les tailles de caractères pour la police courante


La police de caractères s'applique aux caractères qui sont saisis par la suite ou à une sélection préalablement effectuée. Pour chacune de ces CBE, le choix peut se faire dans la liste associée, mais peut aussi être entré dans le champ de saisie.

- Bouton de changement/visualisation d'élément de page

	Affiche l'en-tête (haut du bouton)
	Affiche le corps de document (milieu du bouton)
	Affiche le pied de page (bas du bouton)

L'activation du bouton provoque l'affichage de l'élément de page associé dans la fenêtre NS-WP.

- Champ contenant la position du curseur

	Numéro de colonne et numéro de ligne
---	--------------------------------------

La position du curseur est constamment affichée dans la barre d'outils.

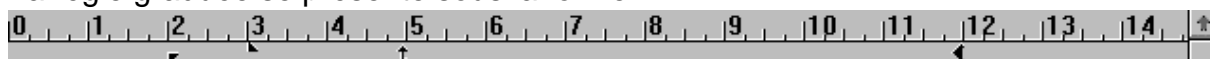
Un bouton "enfoncé" indique l'activation de la fonctionnalité associée. Les boutons peuvent être de deux types :

1. mutuellement exclusifs : l'activation de l'un provoque la désactivation du bouton courant, un des boutons de ce type reste donc toujours actif. Cela concerne les boutons d'alignement et de visualisation de page.
2. cumulatifs : chaque bouton du groupe auquel il appartient peut être activé indépendamment des autres.

La désactivation d'un bouton est provoquée par un nouvel appui. Toutes les options ainsi activées sont cumulées. Cela concerne les boutons d'attributs de caractères.

Règle

La règle graduée se présente sous la forme :



Lorsque le système de mesure adopté est le système métrique, la règle est graduée en centimètres.

Lorsque le système de mesure adopté est le système anglo-saxon, la règle est graduée en inch.

Lorsque le document est géré en mode "caractère", la règle est graduée en caractères.

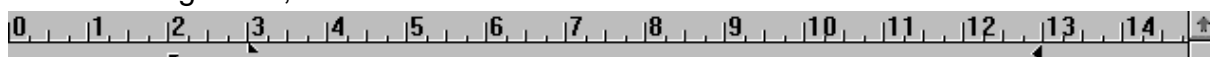
Les retraits de ligne sont représentés sur la règle :

1. retrait de première ligne,



Retrait de première ligne à 3 cm (système métrique)

2. retrait gauche,



Retrait gauche à 2 cm (système métrique)

3. retrait droit.

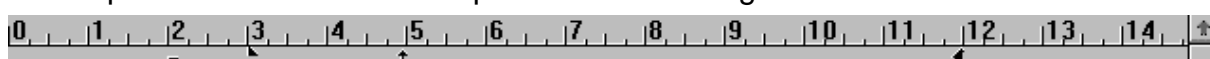


Retrait droit à 12 cm (système métrique)

Les retraits de ligne se déplacent sur la règle à l'aide de la souris.

Les indicateurs de retrait de la première ligne et de retrait à gauche peuvent être déplacés ensemble, ou séparément.

Les taquets de tabulation sont représentés sur la règle :



Tabulation à 5 cm (système métrique)

Les taquets de tabulation peuvent être déplacés à l'aide de la souris.

Un double-clic de la souris sur la règle permet d'insérer le taquet d'une tabulation ou d'une tabulation verticale.

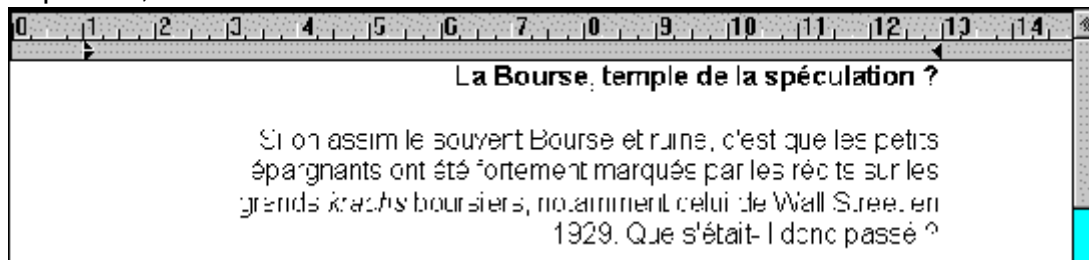
Gestion du mode d'affichage du document

Modes d'affichage

Vous disposez de plusieurs modes d'affichage pour le document :

- "standard",

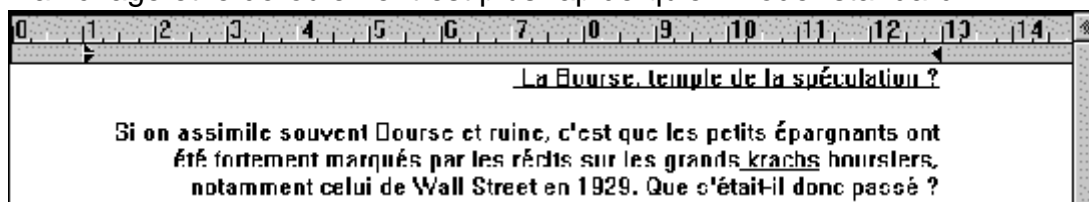
Les caractères gras, italique et souligné sont affichés. Les sauts de page, forcés et implicites, sont affichés.



- "brouillon",

La police utilisée à l'affichage est la police système. Les caractères gras et italiques sont affichés en souligné. La présentation à l'écran ne sera pas celle obtenue sur l'imprimante. Seuls les sauts de page forcés sont affichés.

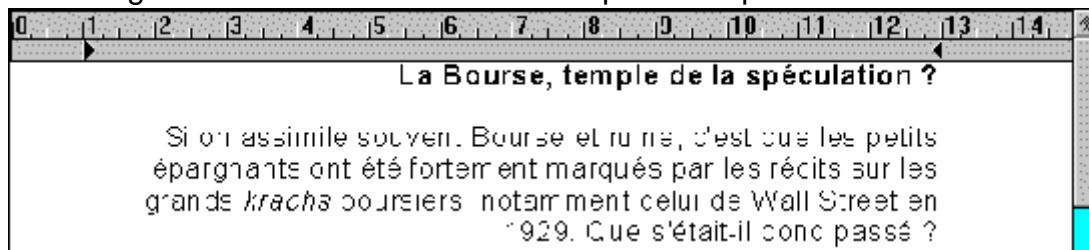
L'affichage et le déroulement est plus rapide qu'en mode "standard".



- "page",

La présentation et les polices utilisées à l'écran sont proches de ce qui est obtenu sur l'imprimante. Les caractères gras, italique et souligné sont affichés. Les sauts de page, forcés et implicites, sont affichés.

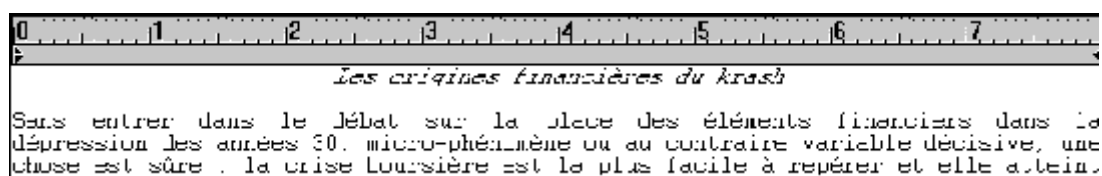
L'affichage et le déroulement du texte est plus lent qu'en mode "standard".



- "caractère".

Ce mode d'affichage, de type éditeur, ne peut être utilisé qu'avec une police non proportionnelle.

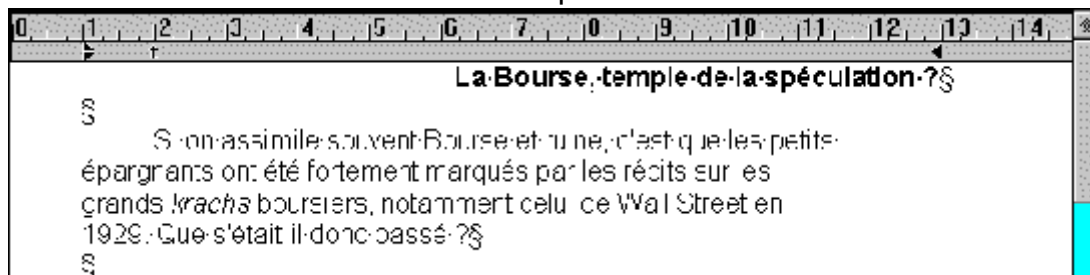
Dans ce mode, chaque ligne fait au maximum un nombre fixé de caractères. Les objets graphiques (tabulations verticales, encadrements, images bitmaps...), les en-têtes et les pieds de page ne sont pas reconnus.



Caractères affichés

Vous pouvez paramétrer les caractères spéciaux affichés à l'écran :

- marques de fin de paragraphe,
- marques de tabulations,
- espaces,
- textes des marqueurs,
- bornes,
- commandes d'insertion de bitmaps.



Affichage avec marques de fin de paragraphe, marques de tabulation et espaces visibles

Mise en page du document

En-têtes et pieds de page

Un document pourra comprendre des en-têtes et des pieds de page. Le plan de travail contiendra au choix :

- l'en-tête de la page courante,
- le corps de la page courante,
- le pied de page de la page courante.

Il y a un en-tête et un pied de page spécifiques pour la première page. Les en-têtes et les pieds de page des autres pages du document sont identiques. Vous pouvez cependant faire varier les en-têtes et les pieds de page d'un document en découpant le document en sections. Chaque section a ses propres en-têtes et pieds de page.

Un en-tête et un pied de page peuvent comprendre trois éléments spécifiques : le numéro de page, la date courante et l'heure courante.

L'intégration de ces trois éléments spécifiques ne nécessite pas de programmation. L'utilisateur de la fenêtre NS WP n'a qu'à insérer dans le texte de l'en-tête ou du pied de page un ou plusieurs des mots-clés suivants :

@PAGE

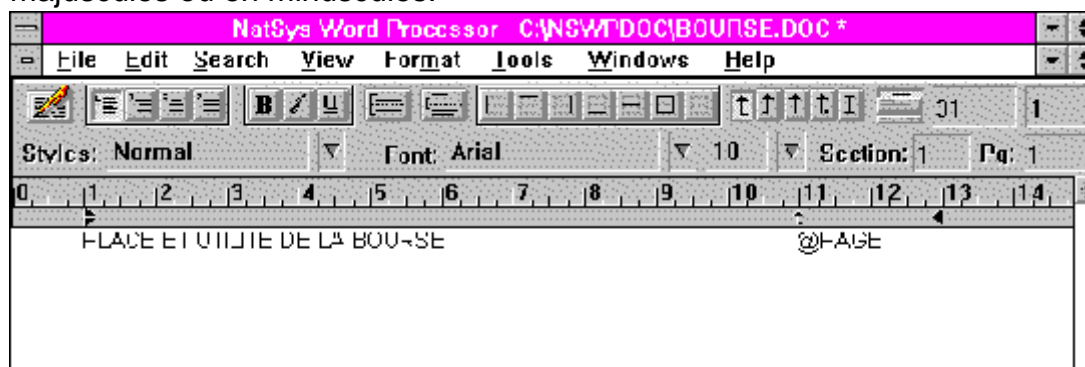
@DATE

@TIME

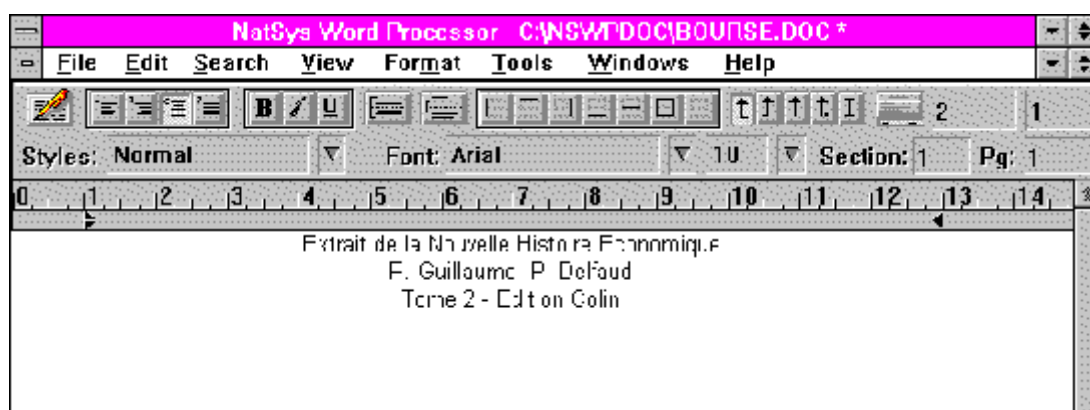
A l'impression, @PAGE est automatiquement remplacé par le numéro de page, @DATE par la date courante, @TIME par l'heure courante.

La date affichée a le même format qu'une date qui serait obtenue par la fonction DATE\$(CURRENTDATE%) de la librairie NSDATE. Ce format peut être spécifié au moyen des instructions SETDATEFORMAT/SETDATESEPARATOR de cette même librairie. Egalement, l'heure affichée a le même format qu'une heure qui serait obtenue avec TIME\$(CURRENTTIME%). Voir à ce sujet le Manuel de Programmation Librairies de NS-DK ou NatStar.

Les mots @PAGE, @DATE et @TIME peuvent être écrits indifféremment en majuscules ou en minuscules.



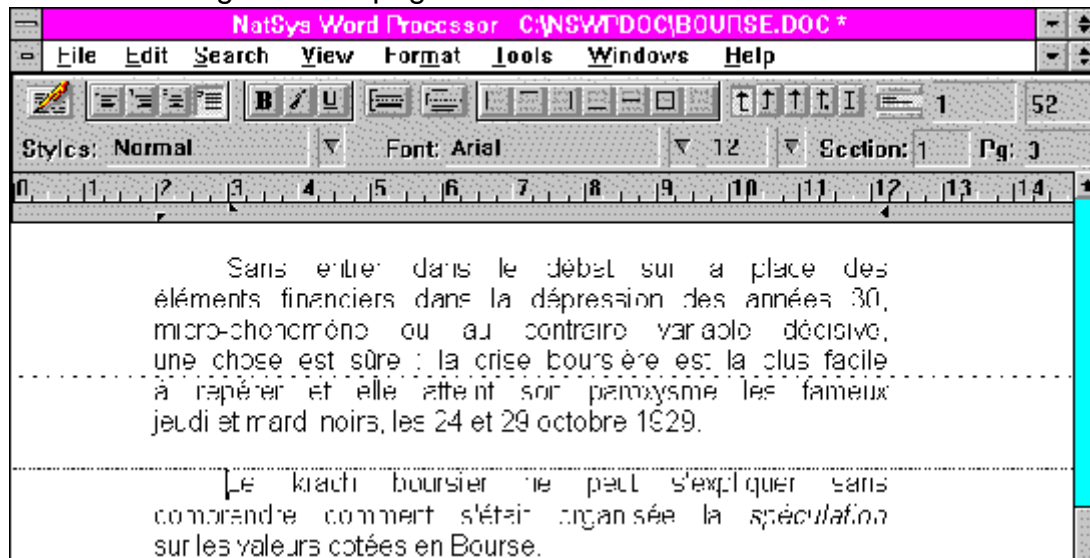
Un en-tête



Un pied de page

Sauts de page

Vous pouvez forcer des sauts de page. Les sauts de pages forcés se rajoutent aux sauts de pages placés implicitement dans le document chaque fois que le nombre maximum de lignes de la page est atteint.



Saut de page implicite, suivi d'un saut de page forcé

Paragraphe et lignes solidaires

Vous pouvez lier des paragraphes entre eux ou des lignes entre elles de façon à ce qu'à l'impression ils (elles) soient sur la même page.

Alignement vertical

Vous pouvez centrer, aligner sur le haut, aligner sur le bas ou justifier le texte d'une page. Vous pouvez faire varier les alignements de page d'un document en découpant le document en sections. Chaque section a son propre alignement vertical.

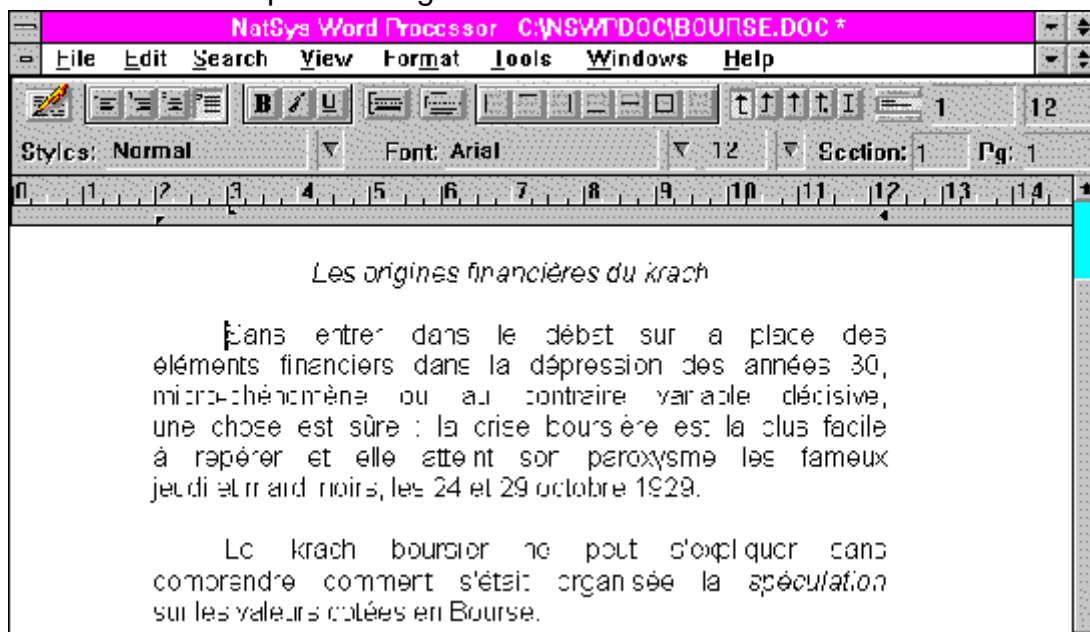
Formatage des paragraphes

Retraits

Vous pouvez modifier les limites d'un paragraphe, déterminées par les marges gauche et droite de la page, en créant des retraits dans le paragraphe.

Trois types de retraits sont reconnus :

- le retrait gauche,
- le retrait droit,
- le retrait de première ligne.



Retrait de première ligne : 3 cm
Retrait gauche : 2 cm
Retrait droit : 12 cm

Vous pouvez faire varier les retraits dans un paragraphe à l'aide de tabulations.

Les taquets de tabulation sont visualisés sur la règle. Un taquet peut être inséré directement sur la règle par un double-clic souris ou défini dans une boîte de dialogue.

Lorsque des taquets de tabulation sont positionnés dans un paragraphe, l'utilisateur peut insérer des marques de tabulation à la position d'insertion avec la touche [TAB].

A chacune de ces tabulations, vous pouvez associer des points de suite :

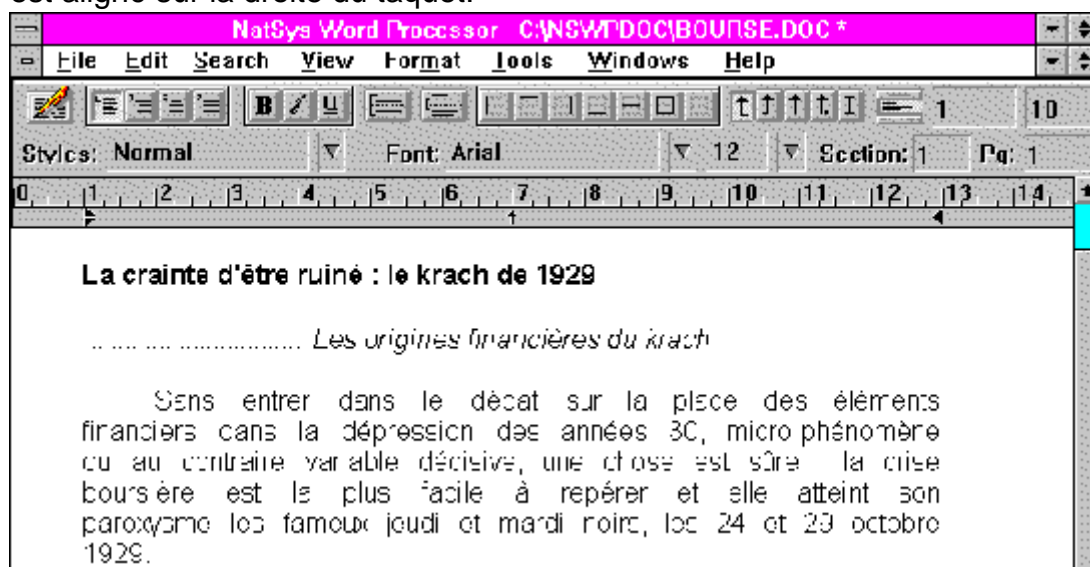
- pointillé,
- tirets,
- souligné.

De plus, le texte décalé au moyen d'une tabulation peut être positionné de différentes façons par rapport au taquet de tabulation :

- aligné sur la gauche du taquet de tabulation,
- centré sur le taquet de tabulation,

- aligné sur la droite du taquet de tabulation.

Lorsque le texte décalé au moyen d'une tabulation est un nombre, il est possible d'aligner le point décimal sur le taquet de tabulation. Un nombre avec zéro décimale est aligné sur la droite du taquet.



"Les origines..." est centré sur le taquet de tabulation situé à 7 cm ; le style de points de suite est un pointillé
 "Sans entrer..." est tabulé à 2 cm sans style de point de suite

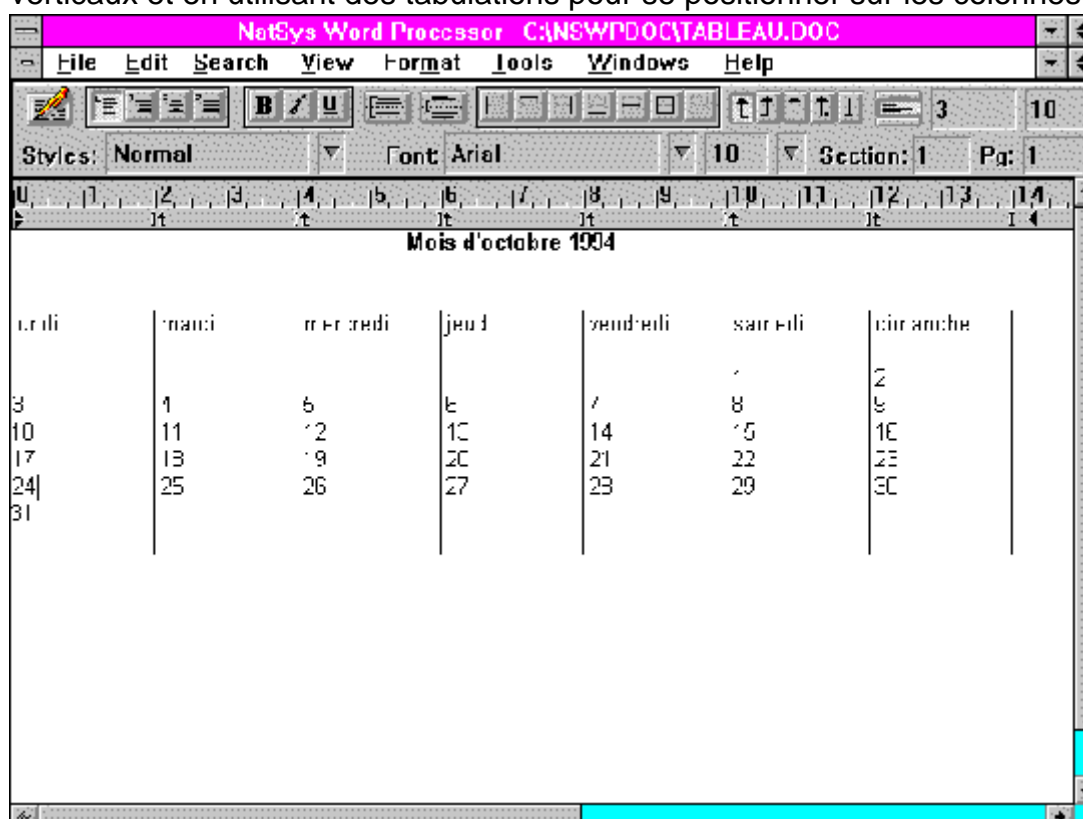
Tabulations verticales

Vous pouvez tracer des traits verticaux dans un paragraphe.

Il suffit de placer sur la règle des taquets spécifiques (taquets de tabulations verticales) aux positions des traits. Les traits sont alors tracés par dessus le texte du paragraphe.

Ces taquets de tabulations verticales peuvent être insérés sur la règle par un double-clic souris ou définis dans une boîte de dialogue.

Un utilisateur peut ainsi créer des tableaux en délimitant les colonnes par des traits verticaux et en utilisant des tabulations pour se positionner sur les colonnes.



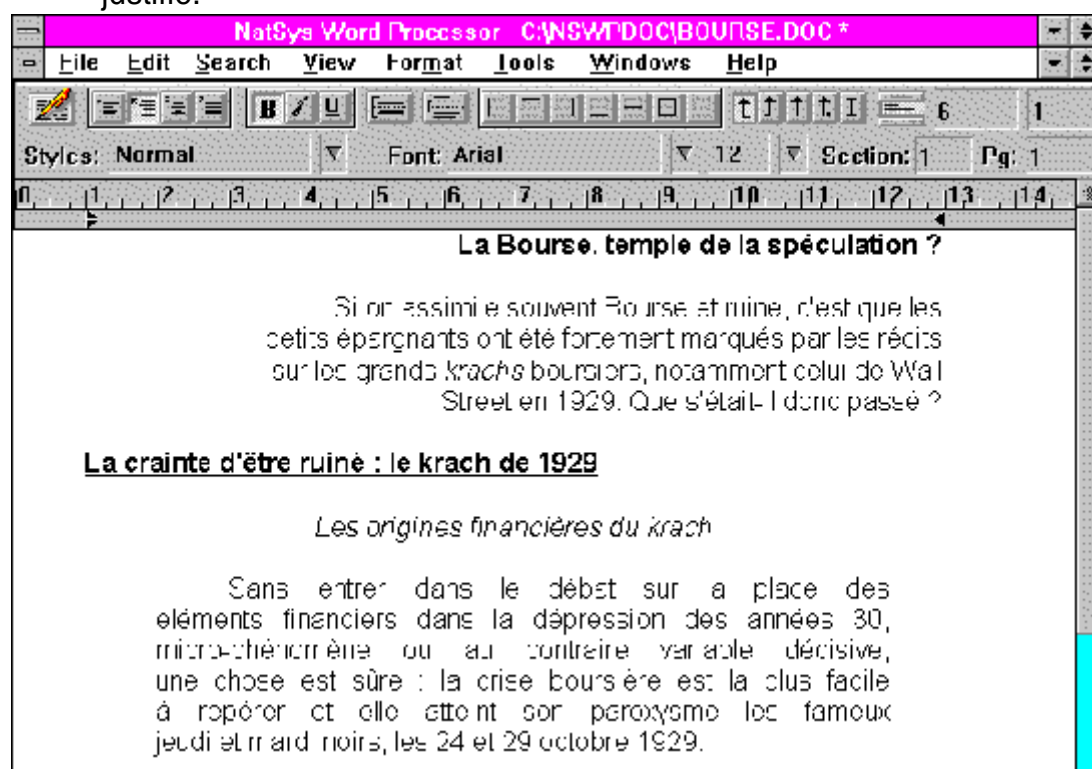
Tabulations verticales à 2, 4, 6, 8, 10, 12, 14 cm

Tabulations alignées à gauche à 2.1, 4.1, 6.1, 8.1, 10.1, 12.1, 14.1 cm

Alignements

Vous disposez de quatre alignements possibles pour un paragraphe :

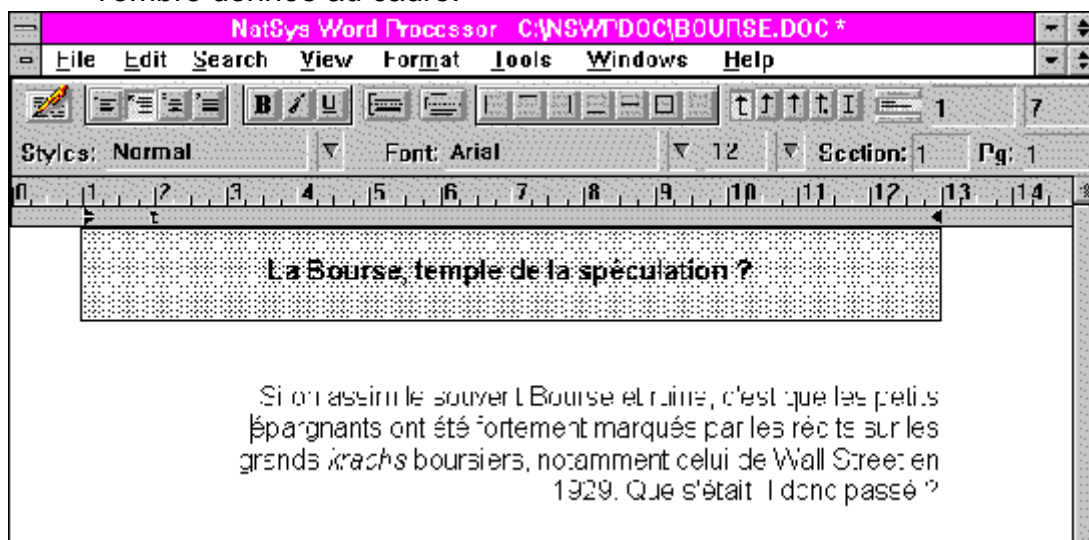
- gauche,
- droit,
- centré,
- justifié.



Encadrement

Vous pouvez encadrer un paragraphe. Un encadrement est complètement défini par :

- le tracé du trait, simple ou double,
- l'espace laissé entre le cadre et le texte,
- la couleur du trait, la couleur et le motif du fond du cadre,
- l'ombre donnée au cadre.



Encadrement avec motif de fond de cadre

Tout nouveau paragraphe hérite à sa création des retraits et des tabulations du paragraphe dont il est issu.

Gestion des paragraphes

La plupart des traitements de texte sont capables d'appliquer un style à un paragraphe et de reconnaître les paragraphes qui sont des titres.

Pour programmer une telle gestion de paragraphe, vous pouvez utiliser une facilité de NS WP qui consiste à associer deux entiers à chaque paragraphe.

Le premier de ces nombres est dit global : tout nouveau paragraphe hérite du nombre entier global associé au paragraphe dont il est issu.

Le deuxième de ces nombres est dit local : un nouveau paragraphe n'hérite pas du nombre entier local associé au paragraphe dont il est issu.

Mettez dans l'entier global d'un paragraphe l'index de son style. Vous liez ainsi le paragraphe et le style. Pour appliquer le style au paragraphe, il suffit que vous recherchiez les caractéristiques du style indexé et que vous les appliquiez au paragraphe. De plus, parce que l'entier est global, chaque paragraphe fils a le même index de style.

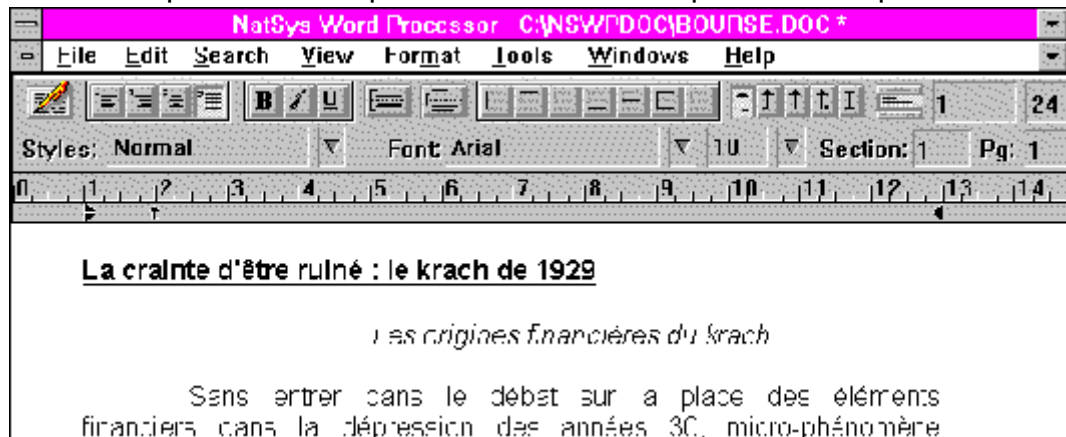
Mettez dans l'entier local d'un paragraphe titre son niveau de titre. De la sorte, vous serez capable au moment de la génération de la table des matières de déterminer quels paragraphes doivent y figurer et à quelle profondeur.

Formatage des caractères

Vous pouvez définir la police, le corps, les couleurs et les attributs d'enrichissement des caractères.

Les attributs d'enrichissement gérés par NS WP sont le gras, l'italique, le souligné et le barré.

Seules les polices des imprimantes connectées peuvent être prises en compte.



Gestion des styles de caractères

Vous pouvez regrouper sous l'appellation "style de caractères" les caractéristiques suivantes :

- une police,
- un corps de police,
- des attributs d'enrichissement,
- les couleurs de premier plan, d'arrière-plan, du trait de soulignement,
- un entier caractérisant le style.

Exemple d'utilisation de l'entier : cet entier sert à générer des glossaires.

Créez un style de caractères que vous n'appliquez qu'aux mots que vous voulez voir figurer dans le glossaire. Définissez pour ce style un entier particulier. Vous êtes désormais capable de retrouver tous les mots ayant ce style en parcourant le document et de les placer dans un glossaire.

Parcours du document

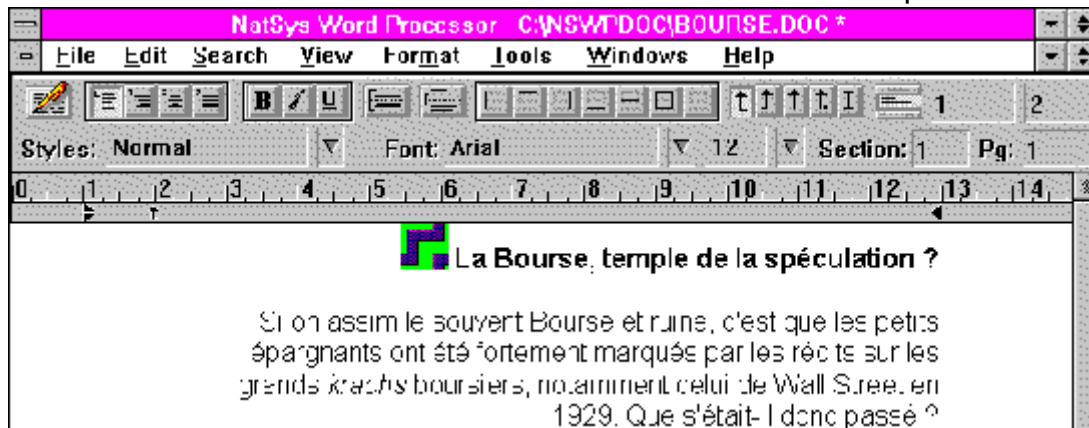
Vous disposez de fonctions qui vous permettent de parcourir le texte élément par élément et qui retournent les caractéristiques de chaque élément.

Les éléments identifiés sont les suites de caractères, les espaces, les marques de fin de paragraphe et de tabulation, les nouveaux styles de caractères, les commandes d'insertion d'une image bitmap, d'un marqueur ou d'une borne.

Le comptage des mots d'un texte, la génération de glossaires nécessitent de parcourir le document.

Insertion d'images bitmaps dans le texte

Un document peut comprendre des images bitmaps au format .BMP. Elles ne sont pas stockées dans le document, mais sont chargées à chaque fois que le document est ouvert. Le fichier d'extension .BMP doit donc être conservé parallèlement.



Une image bitmap

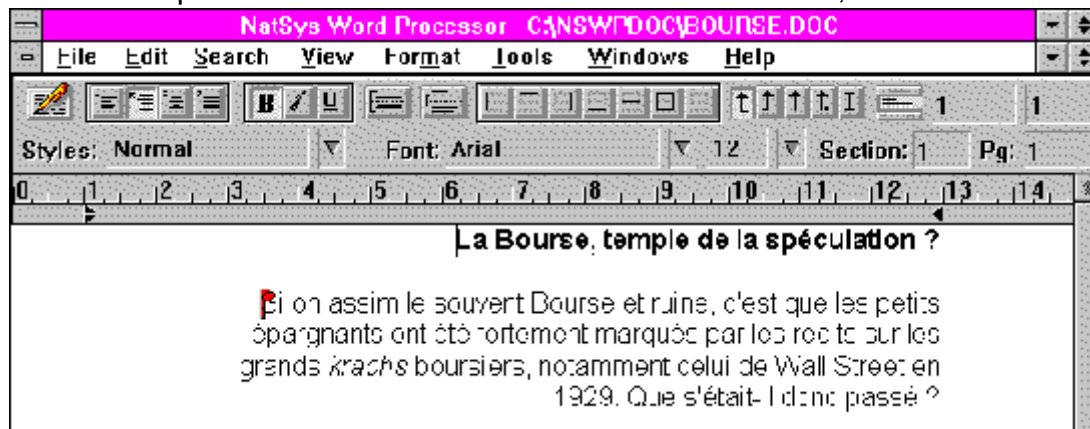
Insertion de bornes dans le texte

Vous pouvez gérer des bornes. Une borne est un repère placé dans le texte. Un utilisateur qui veut repérer un endroit d'un texte y place une borne. Il peut ensuite se repositionner à l'emplacement repéré au moyen des touches suivantes :

[Ctrl]+[Tab] (passage à la borne suivante)

[Ctrl]+[Shift]+[Tab] (passage à la borne précédente)

Les bornes peuvent être visualisées ou cachées. Cachées, elles restent actives.



Une borne en début de paragraphe

Gestion de dictionnaires et insertion de marqueurs

Un dictionnaire, au sens NS WP, est un ensemble de données associées à un document et sauvegardées dans le même fichier que le document.

Vous disposez dans la librairie NS WP de fonctions et instructions permettant de manipuler ces dictionnaires de données.

- Exemple d'utilisation : un dictionnaire peut servir à stocker le nom de la feuille de styles associée au document.

En général, il existe plusieurs feuilles de styles qui contiennent des styles adaptés à des documents de différente nature. Chaque feuille de styles est stockée dans un fichier spécifique. Lorsqu'un utilisateur crée un document, il indique quelle feuille de styles il va utiliser dans ce document. En stockant dans un dictionnaire associé au document le nom de cette feuille de styles, vous faites le lien entre le document et ses styles et vous êtes ainsi capable de récupérer les styles d'un document que vous chargez.

Dans ce cas, le dictionnaire ne comprend qu'une donnée.

Un marqueur est une commande particulière qui peut être insérée dans le texte d'un document.

Chaque marqueur apparaît à l'écran sous forme d'un texte. Vous disposez des fonctions et instructions qui permettent de retrouver les marqueurs et de les traiter (suppression, modification).

- Exemple d'utilisation : vous pouvez utiliser les marqueurs pour programmer un mécanisme de numérotation automatique.

L'utilisateur a une option de menu qui lui permet d'insérer un numéro dans le texte à l'emplacement de son choix. Le texte du marqueur inséré est le nombre immédiatement supérieur au nombre affiché dans le marqueur précédemment inséré. Si un marqueur est supprimé, vous êtes capable de renuméroter les paragraphes par un traitement sur les marqueurs placés dans le texte.

Vous pouvez aussi lier marqueurs et dictionnaires de données. En effet, un marqueur peut faire référence à une donnée d'un dictionnaire.

- Exemple d'utilisation : vous pouvez valoriser des zones de texte .

L'utilisateur en cliquant sur un marqueur fait apparaître une fenêtre qui liste le contenu d'un fichier texte dont le nom était la donnée associée au marqueur dans le dictionnaire. L'utilisateur choisit une des valeurs de cette liste, cette valeur remplace le texte du marqueur.

Gestion des modifications survenant dans la fenêtre NS WP

Vous disposez d'un mécanisme qui avertit une fenêtre NS DK ou NatStar des modifications ayant lieu dans le plan de travail de la fenêtre NS WP.

Les échanges d'informations se font par l'intermédiaire d'événements qui sont déclenchés lorsqu'une modification survient dans le plan de travail.

- Exemple d'utilisation : grâce à ce mécanisme, vous pouvez créer votre propre barre d'outils.

Vous créez une fenêtre NS DK ou NatStar dans laquelle vous disposez les boutons et les champs que vous voulez voir figurer dans la barre d'outils. Quand vous ouvrez la fenêtre NS WP, vous ouvrez également cette fenêtre barre d'outils et vous la placez au dessus du plan de travail.

La barre d'outils comprend des informations concernant le plan de travail, telles que la position du curseur, les attributs, la police et le corps du caractère courant, etc. Si vous déplacez le curseur ou modifiez un attribut, vous devez répercuter votre action au niveau des champs de la barre d'outils. Vous allez donc mettre en place un mécanisme d'échange d'informations entre la fenêtre NS WP et la fenêtre barre d'outils, qui vous permettra de toujours garder à jour l'affichage de la fenêtre barre d'outils.

Choix d'un système de mesure

Vous pouvez opter pour l'un des systèmes de mesure suivants :

- métrique (l'unité est le 1/10 de mm),
- anglo-saxon (l'unité est le 1/100 de pouce),
- caractère (l'unité est le caractère).

Chargement et sauvegarde d'un document

Vous pouvez stocker sur disque ou en mémoire le texte saisi dans la fenêtre de traitement de texte. Le document peut être sauvegardé dans les formats suivants :

- NS-WP : la totalité des informations de formatage sont sauvegardées,
- RTF (Rich Text Format) : la plupart des informations de formatage sont sauvegardées,
- RFT (Revisable Form Text) : la plupart des informations de formatage sont sauvegardées,
- ASCII sous OS/2, ANSI sous Windows : les informations de formatage et de tabulations sont perdues,
- Texte : il s'agit d'un format texte de type éditeur entièrement décrit par l'utilisateur.

Création d'un document par fusion

Vous pouvez créer un document au format NS-WP à partir d'un fichier de commandes.

Le fichier de commandes est un fichier au format NS-WP qui comporte du texte et des commandes spécifiques agissant sur ce texte.

Les commandes pouvant être insérées permettent notamment de poser une question et d'en récupérer la réponse, d'effectuer des boucles, d'inclure du texte provenant de fichiers NS-WP extérieurs.

Elles forment un véritable langage de manipulation de texte.

Une des applications de la création par fusion est l'envoi d'une lettre type à des clients.

Pour envoyer une lettre type à un client, il faut :

- définir le texte de la lettre commun à tous les clients,
- personnaliser ce texte avec des informations spécifiques au client (nom, adresse).

De plus, le texte peut subir quelques modifications : 'Monsieur' remplaçant 'Madame' si le client est un homme, etc.

Il suffit de créer la lettre personnalisée par fusion, en utilisant un fichier de commandes comprenant le texte standard de la lettre et son traitement inhérent.

Autres fonctionnalités

La librairie NS WP comprend aussi des fonctions qui permettent de :

- rechercher/remplacer des chaînes de caractères dans le texte,
- gérer un paragraphe (récupération de son contenu et de son handle),
- gérer le déplacement du curseur et les coordonnées écran,
- gérer les sélections de texte,
- insérer et supprimer du texte,
- gérer le presse-papiers,
- imprimer un document,
- gérer les erreurs,
- gérer la mémoire nécessaire à l'application.

PRESENTATION DE L'API

Affichage des éléments de la fenêtre NS-WP

Fonctions	Description
DOC_GETWINDOW%	retourne la partie du document affichée.
DOC_ISWINDOW%	indique si le document a un en-tête ou un pied de page.
DOC_ISRULERBAR%	permet de savoir si la barre d'outils standard et la règle sont affichées.
DOC_ISRULERBARSHOWN%	permet de savoir si la barre d'outils standard est affichée.
DOC_ISTOOLBOXSHOWN%	permet de savoir si la règle est affichée.

Instructions	Description
DOC_SETWINDOW	détermine la partie du document affichée dans le plan de travail (en-tête, corps du document, pied de page).
DOC_RULERBAR	affiche la barre d'outils standard et la règle.
DOC_SHOWRULERBAR	affiche uniquement la barre d'outils standard.
DOC_SHOWTOOLBOX	affiche uniquement la règle.

Constantes
DOC_WIN_*

Gestion du mode d'affichage du document

Fonctions	Description
DOC_GETMODE%	indique le mode d'affichage choisi avec DOC_SETMODE.
DOC_GETCONTROLCHARSVISIBILITY%	indique les caractères visualisés à l'écran (espaces, tabulations, marques de fin de paragraphe...), indépendamment du mode choisi.
DOC_ISLIKEPRINTER%	indique si la taille des caractères est ajustée.

Instructions	Description
DOC_SETMODE	permet de choisir, parmi les modes "brouillon", "standard" et "page", le mode d'affichage du document.
DOC_SETCONTROLCHARSVISIBILITY	détermine les caractères visualisés à l'écran (espaces, tabulations, marques de fin de paragraphe...), indépendamment du mode choisi.
DOC_SETLIKEPRINTER	ajuste la taille des caractères à la façon de l'imprimante.
DOC_UPDATE	bloque la mise à jour à l'écran. Les modifications effectuées ne sont pas répercutées à l'écran. Permet d'accélérer l'exécution en allégeant les traitements.
DOC_LAYOUT	bloque la mise à jour à l'écran et le formatage du texte. Permet d'accélérer l'exécution en allégeant les traitements.

Constantes
DOC_MODE_*
DOC_SHOW_*, DOC_HIDE_PICTURES%

Mise en page d'un document

Fonctions	Description
DOC_ISPAGEBREAK%	indique s'il y a un saut de page forcé au début du paragraphe courant.
DOC_GETPAGE%	retourne le numéro de la page courante.
DOC_GETNBPAGES%	donne le nombre de pages du document.
DOC_GETWIDOWCONTROL%	indique si le contrôle des lignes veuves et orphelines est activé pour le document.
DOC_GETPARAGRAPHKEEPTOGETHER%	indique si des lignes sont solidaires.
DOC_GETPARAGRAPHKEEPWITHNEXT%	indique si des paragraphes sont solidaires.

Instructions	Description
DOC_PAGEBREAK	insère un saut de page.
DOC_SETMARGINS	modifie les marges d'une page (gauche, droite, haute, basse).
DOC_GETMARGINS	donne les marges d'une page.
DOC_DEFAULTWIDOWCONTROL	positionnent le contrôle des lignes veuves et orphelines.
DOC_SETWIDOWCONTROL	
DOC_SETPARAGRAPHKEEPTOGETHER	déclare solidaires les lignes d'un paragraphe.
DOC_SETPARAGRAPHKEEPWITHNEXT	déclare solidaires deux paragraphes.
DOC_GETPAGEDIM	donne les dimensions de page imprimables, selon les caractéristiques de l'imprimante utilisée.

Gestion des sections

Fonctions	Description
DOC_ISSECTION%	indique s'il y a un saut de section au début du paragraphe courant.
DOC_GETNBSECTIONS%	donne le nombre de sections du document.
DOC_GETSECTION%	retourne le numéro de la section courante.
DOC_GETSECTIONJUSTIF%	indique quel type de justification verticale est adopté pour une section.

Instructions	Description
DOC_SETSECTION	insère un saut de section.
DOC_SETSECTIONJUSTIF	justifie verticalement le texte des pages d'une section.

Constantes
DOC_JUS_ *%

Formatage d'un paragraphe (retraits, tabulations, alignements)

Fonctions	Description
DOC_GETDEFTAB%	retourne la distance séparant les taquets de tabulation positionnés.
DOC_GETJUSTIF%	retourne l'alignement d'un paragraphe.
DOC_GETCURRENTRULEROBJECT%	indique le type des taquets qui seront positionnés sur la règle avec la souris : taquets de tabulation ou taquets de tabulation verticale.

Instructions	Description
DOC_DEFAULTPARAGRAPHINDENT	définit des retraits de paragraphe à prendre en compte par défaut.
DOC_DEFAULTTAB	
DOC_SETDEFTAB	positionne à intervalle constant des taquets de tabulation pour tout le document.
DOC_SETJUSTIF	modifie l'alignement d'un paragraphe.
DOC_GETTABS	conservée par souci de compatibilité ascendante
DOC_SETTABS	conservée par souci de compatibilité ascendante
DOC_GETXTABS	retourne les informations sur les tabulations et les retraits d'un paragraphe.
DOC_SETXTABS	remplace les taquets de tabulation et les retraits du paragraphe courant. Il est possible d'associer un style de points de suite à chaque taquet de tabulation et de décider de son alignement.
DOC_GETPARAGRAPHINFO	retourne les informations suivantes concernant un paragraphe : taille, numéro de la première ligne, nombre de lignes, alignement, retraits et taquets de tabulation.
DOC_SETPARAGRAPHSINFO	modifie l'alignement, les taquets de tabulation et les retraits d'un ou plusieurs paragraphes spécifiés par leurs handles.
DOC_GETCURRENTTAB	permet de connaître les caractéristiques des taquets de tabulation positionnés.

DOC_SETCURRENTTAB	décrit les taquets qui seront positionnés sur la règle avec la souris : taquets de tabulation.
DOC_GETCURRENTTABLINE	permet de connaître les caractéristiques des taquets de tabulation verticale positionnés.
DOC_SETCURRENTTABLINE	décrit les taquets qui seront positionnés sur la règle avec la souris : taquets de tabulation verticale.
DOC_GETTABLINES	renvoie les valeurs des tabulations verticales définies.
DOC_SETTABLINES	définit les tabulations verticales du paragraphe courant.
DOC_SPLITPARAGRAPH	insère une marque de fin de paragraphe.

Constantes
DOC_JUS_*
DOC_MAX_TABULATIONS%
DOC_LEADTAB_*
DOC_TYPTAB_*
TAB_RULER_OBJECT%
TABLINE_RULER_OBJECT%

La constante DOC_MAX_TABULATIONS% détermine le nombre maximum de tabulations d'un paragraphe.

Segments
DOC_TABS
DOC_XTABS
DOC_TABLINE
DOC_TABLINES

Association d'un entier à un paragraphe

Fonctions	Description
DOC_GETPARAGRAPHGLOBALUSERINFO%	récupère l'entier associé au paragraphe global.
DOC_GETPARAGRAPHLOCALSUSERINFO%	récupère l'entier associé au paragraphe local.

Instructions	Description
DOC_SETPARAGRAPHGLOBALUSERINFO	associe un entier à un paragraphe. Les paragraphes fils hériteront de cet entier.
DOC_SETPARAGRAPHLOCALSUSERINFO	associe aussi un entier à un paragraphe, mais cet entier ne se propagera pas aux paragraphes fils.

Encadrement d'un paragraphe

Instructions	Description
DOC_GETBORDER	donne les caractéristiques d'un encadrement.
DOC_SETBORDER	encadre un ou plusieurs paragraphes. Le cadre peut être complet ou partiel, ombré, coloré, plus ou moins épais.

Constantes
DOC_LINE_ *%

Segments
LINE_BORDER

Récupération du contenu d'un paragraphe

Fonctions	Description
DOC_GETTEXTSIZE%	retourne le nombre de caractères contenus dans un paragraphe.

Instructions	Description
DOC_GETTEXT	récupère le texte complet ou partiel d'un paragraphe.
DOC_GETLINE	décrit une ligne d'un paragraphe : nombre de caractères, texte de la ligne, polices, présence ou non d'une rupture de page.
DOC_GETLINEINFO	donne la position d'une ligne dans un paragraphe et son nombre de caractères

Segments
DOC_LINEDATA

Recherche du handle d'un paragraphe

Fonctions	Description
DOC_FIRSTPARAGRAPH%	retourne le handle du premier paragraphe
DOC_LASTPARAGRAPH%	retourne le handle du dernier paragraphe
DOC_NEXTPARAGRAPH%	retourne le handle du paragraphe suivant
DOC_PREVPARAGRAPH%	retourne le handle du paragraphe précédent
DOC_CURRENTPARAGRAPH%	retourne le handle du paragraphe courant

Ces cinq fonctions permettent de récupérer les handles de tous les paragraphes d'un document.

Formatage des caractères

Fonctions	Description
DOC_GETFONT\$	retourne la police et le corps d'un caractère au point d'insertion.
DOC_GETFONTAT\$	retourne la police et le corps d'un caractère à une position donnée.
DOC_GETFONTBACKCOLOR%	retourne les couleurs d'arrière-plan au point d'insertion.
DOC_GETFONTFORECOLOR%	retourne les couleurs d'avant-plan au point d'insertion.
DOC_GETFONTLINECOLOR%	retourne les couleurs de souligné au point d'insertion.
DOC_GETFORMAT%	retourne les attributs du texte au point d'insertion.
DOC_GETFORMATAT%	retourne les attributs du texte à une position donnée.
DOC_GETUSERFORMAT%	retourne l'entier associé à la police de caractères au point d'insertion.
DOC_GETUSERFORMATAT%	retourne l'entier associé à la police de caractères utilisée à une position donnée.

Instructions	Description
DOC_SETFONT	change la police et le corps des caractères.
DOC_GETFONTCOLORS	retourne la couleur des caractères au point d'insertion.
DOC_GETFONTCOLORSAT	retourne la couleur des caractères à une position donnée.
DOC_SETFONTCOLORS	change la couleur des caractères (couleur d'arrière-plan, couleur des lettres et couleur du souligné).
DOC_SETFONTBACKCOLOR	change la couleur d'arrière-plan.
DOC_SETFONTFORECOLOR	change la couleur des lettres.
DOC_SETFONTLINECOLOR	change la couleur du souligné.
DOC_MODIFYFORMAT	modifie un ou plusieurs des attributs d'une partie du texte (normal, gras, italique, souligné, barré, accessible en écriture).

DOC_SETFORMAT	donne de nouveaux attributs à une partie du texte (normal, gras, italique, souligné, barré, accessible en écriture)
DOC_GETFONTNAMECONTROL	permet de récupérer les noms des polices disponibles.
DOC_GETFONTSIZECONTROL	permet de récupérer les tailles des polices disponibles.
DOC_SETUSERFORMAT	associe un entier à la police de caractères courante.

Constantes

DOC_FMT_*

Segments

DOC_TEXTCOLORS

Gestion des styles de caractères

Fonctions	Description
DOC_GETFONTINDEXAT%	indique quel style de caractères est utilisé à une position donnée.
DOC_GETFONTINDEXFORMAT%	renvoie les attributs d'enrichissement (normal, gras, italique, souligné, barré, accessible en écriture).
DOC_GETFONTINDEXNAME\$	renvoie la police.
DOC_GETFONTINDEXSIZE%	renvoie les spécificités du corps des caractères.
DOC_GETFONTINDEXCOLORS%	renvoie les couleurs d'arrière-plan, d'avant-plan et de souligné.
DOC_GETFONTINDEXUSERFORMAT%	renvoie l'entier associé au style.

Instructions	Description
DOC_DEFAULTFONT	permet de créer un style de caractères qui sera utilisé par défaut.
DOC_LOCKFONTS	interdit dans le document courant l'utilisation de tout autre style que celui défini par défaut.
DOC_DEFAULTLOCKFONTS	interdit dans les nouveaux documents l'utilisation de tout autre style que celui défini par défaut.
DOC_SETFONTINDEXCOLORS	spécifie les couleurs d'arrière-plan, d'avant-plan et de souligné.
DOC_SETFONTINDEXFORMAT	spécifie les attributs d'enrichissement (normal, gras, italique, souligné, barré, accessible en écriture).
DOC_SETFONTINDEXNAME	spécifie la police.
DOC_SETFONTINDEXSIZE	spécifie le corps des caractères.
DOC_SETFONTINDEXUSERFORMAT	associe un entier au style.

Constantes
DOC_FMT_*

Segments
DOC_FONT
DOC_TEXTCOLORS

Cinq instructions suffisent pour créer un style de caractères :

- DOC_SETFONTINDEXNAME
- DOC_SETFONTINDEXSIZE
- DOC_SETFONTINDEXCOLORS
- DOC_SETFONTINDEXFORMAT
- DOC_SETFONTINDEXUSERFORMAT

Si l'une de ces instructions n'est pas employée, des valeurs par défaut sont associées au style (attribut normal, couleur d'arrière-plan blanc, d'avant-plan et de souligné noir, entier associé 0).

Gestion du curseur

Instructions	Description
DOC_GETPOS	donne la position du curseur.
DOC_SETPOS	positionne le curseur à un emplacement précis.

Gestion de la sélection

Fonctions	Description
DOC_ISSELECTION%	indique s'il y a une sélection en cours.
DOC_GETSELECTIONTEXT\$	retourne le texte sélectionné.

Instructions	Description
DOC_GETSELECTION	retourne les coordonnées de début et de fin du texte sélectionné.
DOC_SETSELECTION	sélectionne une partie de texte

Insertion et suppression de texte

Fonctions	Description
DOC_ISREADONLY%	indique si les modifications sont interdites.
DOC_ISUNDO%	indique si la dernière action est de celles qui peuvent être annulées.
DOC_ISREDO%	indique si une annulation d'annulation est possible.
DOC_GET_INSERT_MODE%	retourne le mode actif (insertion ou surimpression).

Instructions	Description
DOC_READONLY	empêche toute modification du texte affiché.
DOC_UNDO	annule la dernière action venant d'être accomplie, à condition que ce soit une action concernant le presse-papiers, la frappe clavier ou le formatage.
DOC_REDO	annule la dernière annulation effectuée avec DOC_UNDO.
DOC_DELETE	efface le texte sélectionné.
DOC_NEW	efface la totalité du texte.
DOC_DELETECHARS	efface une partie du document.
DOC_DISPOSEWINDOW	efface le texte d'une des parties du document (en-tête, corps, pied de page).
DOC_INSERTCHARS	insère du texte stocké en mémoire dans un paragraphe.
DOC_SET_INSERT_MODE	permet de passer du mode d'insertion au mode surimpression et inversement.

Gestion du presse-papiers

Fonctions	Description
DOC_ISCLIPBOARD%	indique s'il y a du texte dans le presse-papiers.

Instructions	Description
DOC_CUT	supprime le texte sélectionné et le place dans le presse-papiers.
DOC_COPY	copie le texte sélectionné dans le presse-papiers.
DOC_PASTE	insère le contenu du presse-papiers à gauche du curseur.

Gestion des coordonnées écran

Fonctions	Description
DOC_POS2XYVIEW%	donne les coordonnées en pixels (abscisse et ordonnée) d'un caractère du texte.
DOC_XYVIEW2POS%	donne l'emplacement dans le texte du caractère dont sont connues les coordonnées exprimées en pixels.
DOC_XPEL2UNITS%	convertit une abscisse exprimée en pixels dans l'unité courante (système métrique ou anglo-saxon).
DOC_YPEL2UNITS%	convertit une ordonnée exprimée en pixels dans l'unité courante (système métrique ou anglo-saxon).
DOC_UNITS2XPEL%	convertit une abscisse exprimée dans l'unité courante (système métrique ou anglo-saxon) en pixels.
DOC_UNITS2YPEL%	convertit une ordonnée exprimée dans l'unité courante (système métrique ou anglo-saxon) en pixels.

Pour chacune de ces fonctions, l'origine des axes pris comme référence est le coin inférieur gauche.

Constantes
DOC_POS_*

Recherche et remplacement de texte

Fonctions	Description
DOC_FIND%	recherche une chaîne de caractères dans le document.
DOC_CHANGE%	remplace une chaîne de caractères par une autre dans tout le document. Certaines options peuvent être spécifiées (demande de confirmation...).

Constantes
DOC_FIND_*%

Parcours du document

Fonctions	Description
DOC_ALLOCSCANCONTEXT%	alloue un contexte, dans lequel se déroulera le parcours.
DOC_FREESCANCONTEXT%	libère ce contexte une fois le parcours terminé.
DOC_SCANFIRST%	initialise le parcours, en indiquant notamment la position de début.
DOC_SCANNEXT%	permet de passer d'élément en élément.

Ces quatre fonctions permettent de parcourir le document en passant d'élément en élément. Les caractéristiques d'un élément sont à chaque fois récupérées, afin de pouvoir être éventuellement traitées.

Les éléments reconnus dans le document sont : les suites de caractères, les espaces, les marques de tabulation, les marques de fin de paragraphe, les changements de styles de caractères, les bornes, les marqueurs et les bitmaps.

Constantes
DOC_SCN_*

Segments
DOC_SCAN_CTXT

Insertion d'images bitmaps

Fonctions	Description
DOC_ADDPICTDFN%	Pour être inséré dans le texte, un fichier bitmap doit avant tout être chargé en mémoire avec DOC_ADDPICTDFN%. Au chargement, DOC_ADDPICTDFN% attribue un identificateur à la bitmap.
DOC_DELPICDFN%	libère la mémoire occupée par un fichier bitmap.
DOC_GETPICTDFNID%	donne l'identificateur associé à un fichier bitmap.
DOC_GETPICTDFNNAME%	donne le chemin et le nom du fichier bitmap connaissant son identificateur.
DOC_GETPICTSIZE%	donne les dimensions d'une image bitmap chargée en mémoire.
DOC_INSERTBITMAPCHAR%	insère dans le texte un fichier bitmap précédemment chargé en mémoire. Sa largeur, sa hauteur, sa position par rapport à la ligne doivent être précisées.
DOC_GETBITMAPCHAR%	donne les caractéristiques (identificateur, largeur, hauteur et position par rapport à la ligne) d'une image bitmap placée dans le texte.
DOC_MODIFYBITMAPCHAR%	modifie ces caractéristiques.
DOC_DELETEBITMAPCHAR%	supprime une image bitmap placée dans le texte.

Insertion de bornes

Fonctions	Description
DOC_ISTAG%	indique si une borne est présente à une position donnée.

Instructions	Description
DOC_SETTAG	insère une borne dans le texte.
DOC_CLEAR_TAG	efface la borne située à une position donnée.

Gestion d'un dictionnaire

Fonctions	Description
DOC_M_NEWMARKERCLASS%	crée un dictionnaire de données et retourne son handle.
DOC_M_CREATEMARKER%	ajoute dans un dictionnaire une entrée de données et retourne le handle de l'entrée créée. Un numéro d'insertion est automatiquement attribué à l'entrée.
DOC_M_CREATEMARKERWITHHANDLE%	ajoute dans un dictionnaire l'entrée de données dont le numéro est passé en paramètre, et retourne le handle de l'entrée créée.
DOC_M_GETCLASS%	retourne le handle d'un dictionnaire connaissant son nom.
DOC_M_GETCLASSNAME\$	retourne le nom d'un dictionnaire connaissant son handle.
DOC_M_GETMARKERCLASS%	retourne le handle du dictionnaire qui contient l'entrée dont le handle est passé en paramètre.
DOC_M_GETMARKERDATASIZE%	retourne la taille mémoire en octets occupée par les données stockées dans une entrée d'un dictionnaire.

Instructions	Description
DOC_M_REGISTERMARKERCLASS	attache un dictionnaire de données au document. Le dictionnaire est alors conjointement sauvegardé avec le document.
DOC_M_DEREGISTERMARKERCLASS	supprime le lien entre un dictionnaire et le document.
DOC_M_DISPOSEMARKERCLASS	détruit un dictionnaire.
DOC_M_DISPOSEMARKER	supprime une entrée d'un dictionnaire, connaissant son handle.
DOC_M_GETMARKERDATA	récupère les données stockées dans une entrée d'un dictionnaire.
DOC_M_SETMARKERDATA	insère des données dans un dictionnaire.

Constantes
DOC_M_KIND_DEFAULT%

Insertion de marqueurs

Fonctions	Description
DOC_M_FINDNEXTMARKER%	recherche dans le document le marqueur suivant.
DOC_M_FINDPREVMARKER%	recherche dans le document le marqueur précédent.
DOC_M_GETMARKERHANDLE%	retourne le handle d'un marqueur.
DOC_M_GETTEXT\$	retourne le texte d'un marqueur.
DOC_M_ISMARKER%	indique si un marqueur est présent à une position donnée.

Instructions	Description
DOC_M_INSERTMARKER	insère un marqueur, et son texte, dans le document. Ce marqueur peut être lié à des données d'un dictionnaire, il suffit pour cela de donner comme handle de marqueur le handle d'une entrée de données d'un dictionnaire.
DOC_M_SETTEXT	remplace le texte d'un marqueur.

Segments
DOC_M_MARKER

Gestion des modifications survenant dans la fenêtre NS-WP

Instructions	Description
DOC_MAPUSERMSG	informe une fenêtre réceptrice des modifications effectuées dans la fenêtre NS-WP. Pour cela, elle provoque l'activation d'un événement utilisateur dans la fenêtre réceptrice à chaque fois qu'une modification survient dans la fenêtre NS-WP. La modification peut alors être traitée dans la fenêtre réceptrice, ses caractéristiques lui étant passées.

Constantes
DOC_*_EVENT
DOC_USER_MESSAGE_*
DOC_RCHG_*

Segments
DOC_RULEINFO
DOC_M_MARKER

Gestion du système de mesure

Fonctions	Description
DOC_ISMETRICS%	indique quel est le système de mesure utilisé.

Instructions	Description
DOC_METRICS	<p>permet de choisir le système de mesure pris en compte pour le document : métrique, anglo-saxon, caractère.</p> <p>Le mode "caractère" est un mode de fonctionnement simplifié qui s'utilise avec une police non proportionnelle, la fenêtre NS-WP s'apparentant alors à une fenêtre d'édition de texte.</p>
DOC_DEFAULTMETRICS	permet de spécifier un système de mesure à utiliser par défaut.
DOC_DEFAULTMAXPARAWIDTH	définit la longueur maximum d'une ligne en mode "caractère".

Constantes
DOC_LO*%
DOC_CHARS%

Chargement et sauvegarde d'un document

Fonctions	Description
DOC_FILE_FORMAT%	indique si un fichier est au format NS-WP, ASCII (ANSI), RTF ou RFT.
ESCLOAD%	charge dans une fenêtre NS-WP gérée en mode "caractère" un fichier possédant un de ces formats texte spécifiques. Permet d'utiliser NS-WP en tant qu'éditeur de texte.
ESCPRIINT%	sauvegarde sur disque un document saisi en mode "caractère" dans un format texte entièrement décrit par l'utilisateur. Dans un tel format texte, les marques de fin de paragraphe, les changements de police... sont transformés en chaînes de caractères significatives, chaînes définies par l'utilisateur. Permet d'utiliser NS-WP en tant qu'éditeur de texte.

Instructions	Description
DOC_OPEN	charge dans la fenêtre NS-WP un document stocké sur disque ou en mémoire possédant un de ces formats.
DOC_SAVE	sauvegarde sur disque ou en mémoire un document dans un des formats suivants : format NS-WP, format ASCII sous OS/2 ou ANSI sous Windows, format RTF (Rich Text Format), format RFT (Revisable Form Text).

Constantes
DOC_FILE_ *%
TE*%

Segments
DOC_MEMFILE
TXTESCTABLE

Impression

Fonctions	Description
DOC_PRINT%	imprime plusieurs pages ou la totalité du document édité.
DOC_PRINTFILE%	imprime plusieurs pages ou la totalité d'un fichier ouvrable par NS-WP.
DOC_EXTPRINTSTART%	permet de créer un job d'impression qui pourra regrouper plusieurs lancements d'impression effectués avec DOC_EXTPRINT% (impression du document édité) ou DOC_EXTPRINTFILE% (impression d'un fichier).
DOC_EXTPRINT%	permettent d'assurer la continuité de la numérotation lors de l'impression de plusieurs documents.
DOC_EXTPRINTFILE%	

Instructions	Description
DOC_ASSOCIATEPRINTER	associe une imprimante à un document. Cela permet de récupérer pour le document les polices imprimables et les caractéristiques de mises en page liées à l'imprimante.
DOC_EXTPRINTSTOP	ferme le job d'impression et envoie l'impression.

Fusion de documents

Instructions	Description
DOC_MERGE	crée un nouveau fichier à partir d'un fichier de fusion constitué de commandes, et éventuellement à partir de données.
DOC_GETLASTMERGEERROR	décrit plus précisément une erreur provoquée par DOC_MERGE.

Gestion des erreurs

Fonctions	Description
DOC_ERROR%	retourne une des constantes DOC_ERR*% selon la dernière erreur rencontrée. Elle peut être appelée après l'utilisation de certaines fonctions de l'A.P.I. (ouverture de fichier, lecture/écriture sur disque, insertion de bitmap...).

Constantes
DOC_ERR*%

Gestion de la mémoire

Fonctions	Description
DOC_ALLOCATEDMEM%	retourne la place mémoire allouée à NS-WP.

REFERENCE DE L'API

Avertissements

Avertissement 1 : Au sujet du handle de la fenêtre NS-WP

Dans ce manuel de référence, le handle de la fenêtre NS-WP est noté :

- "handle-WP" dans la syntaxe des fonctions et instructions
- "HWP%" dans les exemples

Ce classique handle de fenêtre NS-DK ou NatStar désigne une fenêtre de classe Window ayant {NS02WP} dans son titre.

Avertissement 2 : Au sujet des paramètres !V!

Les paramètres dont la description commence par !V! sont passés par adresse.
Un tel 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.

Événements

Certains événements ont une signification particulière pour les fenêtres NS-WP :

Événement CHANGED

Reçu lorsque le texte contenu dans la fenêtre change. PARAM12% et PARAM34% valent -1 dans ce cas.

Événement SELECTED

Reçu lorsque le curseur change de place. PARAM12% contient le numéro de ligne et PARAM34% le numéro de colonne de la nouvelle position.

Événement TIMER

Reçu lorsqu'une sélection est faite avec la souris, uniquement pour les besoins internes de NS-WP. PARAM1% vaut 100. Afin d'éviter tout conflit, nous vous conseillons de ne pas employer de TIMER identifié par le numéro 100 (voir l'instruction STARTTIMER du langage NCL).

Événement USER0

Reçu lors de l'emploi de la fonction DOC_CHANGE% avec la constante DOC_FIND_CONFIRM%.

Événement USER1

Reçu lors de l'emploi de l'instruction DOC_MERGE, si le fichier de commande de fusion contient la commande /ASK.

Événements USER0...USER15

Pouvant être reçus lors de modifications dans la fenêtre NS-WP, suite à l'emploi de l'instruction DOC_MAPUSERMSG.

PARAM1% indique le type de la modification et vaut l'une des constantes DOC_*_EVENT.

Lorsqu'il y a action sur des marqueurs, PARAM34% contient l'adresse de la variable segment DOC_M_MARKER comportant la description du marqueur en cause.

Lorsqu'il y a modification de l'état du document, PARAM34% contient l'adresse de la variable segment DOC_RULEINFO comportant la description de la modification d'état.

Bien sûr, les événements habituellement reçus dans une fenêtre de classe Window sont reçus dans une fenêtre NS-WP. Citons notamment :

- Événements BUTTONDOWN et BUTTONUP pour les clics souris
- Événement MOUSEMOVE pour les déplacements souris
- Événement CHARACTER pour les appuis clavier
- Événements HSCROLL et VSCROLL pour les déroulements de texte
- Événement PAINT lorsque la fenêtre doit être redessinée

Fonctions et Instructions NCL

Trois familles de fonctions et instructions classiques NCL ont une signification particulière pour les fenêtres NS-WP :

Instructions **LOAD** et **SAVE**

Equivalentes à DOC_OPEN (avec DOC_FILE_NSWP%) et à DOC_SAVE (avec le format renvoyé par DOC_FILE_FORMAT%).

Instructions **UPDATE** et **NOUPDATE**

Equivalentes à DOC_UPDATE TRUE% ou FALSE%.

Instructions **LOCK** et **UNLOCK**, fonction **ISLOCKED%**

Equivalentes à DOC_READONLY TRUE%, DOC_READONLY FALSE%, et à DOC_ISREADONLY%.

Par contre, les fonctions et instructions NCL telles que INSERT, DELETE, SELECT, UNSELECT, SELECTION%, ISSELECTED% n'ont aucun effet sur les fenêtres NS WP.

Les APIs

Constantes

Constante DOC_M_KIND_DEFAULT%

Type du dictionnaire.

Valeur	Déclaration interne
DOC_M_KIND_DEFAULT%	0

Voir aussi DOC_M_NEWMARKERCLASS%

Constante **DOC_MAX_TABULATIONS%**

Nombre maximum de taquets de tabulation qu'il est possible de placer dans un paragraphe.

Valeur	Déclaration interne
DOC_MAX_TABULATIONS%	64

Voir aussi Segments *DOC TABS*, *DOC XTABS*, *DOC GETTABS*, *DOC SETTABS*, *DOC GETXTABS*, *DOC SETXTABS*

Constantes DOC_*_EVENT

Types de modifications pouvant affecter la fenêtre NS-WP.

Syntaxe	Déclaration interne	Description
DOC_SETFOCUS_EVENT	1	appui sur les touches Ctrl + T
DOC_RULER_EVENT	2	modification de l'état du texte (position du curseur, numéro de la page, police...)
DOC_FONTNAME_EVENT	3	changement de la liste des polices de caractères (généralement après la sélection d'une imprimante)
DOC_FONTSIZE_EVENT	4	changement de la liste des tailles de caractères d'une police (en fonction de la police de caractères sélectionnée)
DOC_M_DELETEMARKER_EVENT	5	suppression d'un marqueur
DOC_M_PASTEMARKER_EVENT	6	copie du presse-papiers dans un document, pour chaque marqueur contenu dans le presse-papiers

Voir aussi Constantes DOC_USER_MESSAGE *, DOC_MAPUSERMSG

Constantes DOC_ERR*%

Codes d'erreurs retournés par DOC_ERROR%.

Valeurs	Déclaration interne	Description
DOC_ERRBADPARAM%	17	mauvais paramètres dans une fonction ou une instruction
DOC_ERRBADPICTMETRICS%	21	dimensions de la bitmap incorrectes
DOC_ERRCANNOTCLOSEFILE%	16	fermeture de fichier impossible
DOC_ERRCANNOTCREATEFILE%	4	création de fichier impossible
DOC_ERRCANNOTOPENFILE%	3	ouverture de fichier impossible
DOC_ERRCANNOTREADFILE%	5	lecture de fichier impossible
DOC_ERRCANNOTWRITEONFILE%	6	écriture sur le fichier impossible
DOC_ERRDUPLPICTNAME%	22	bitmap déjà chargée en mémoire
DOC_ERRFILENOTFOUND%	9	fichier d'entrée non trouvé
DOC_ERRINTERNAL%	15	erreur interne NS-WP
DOC_ERRINVALIDFILE%	7	fichier inconnu
DOC_ERRINVALIDFILEFORMAT%	8	incohérence entre le format du fichier et son suffixe
DOC_ERRINVALIDPICT%	24	fichier bitmap invalide (n'a pas pu être lu)
DOC_ERRINVALIDVARIABLE%	14	/DEFINE erroné (variable ne se terminant pas par un des signes %, #, \$, ou £...)
DOC_ERRLOCKED%	25	insertion de caractères non autorisée (mode"Readonly")

DOC_ERRNO%	0	aucune erreur n'a été trouvée
DOC_ERRNOPICTATCURSPOS%	20	le curseur n'est pas sur une bitmap
DOC_ERRNOTAPICTID%	18	identificateur de la bitmap inconnu
DOC_ERRNOTAPICTNAME%	19	nom de la bitmap inconnu
DOC_ERRNOTENOUGHMEMORY%	1	mémoire insuffisante
DOC_ERRNOTENOUGHSYSRESOURCES%	23	Windows : ressources insuffisantes (nombre de fenêtres sous Windows limité)
DOC_ERRPARAGRAPHOVERFLOW%	2	paragraphe trop important
DOC_ERRREPDEFAULT%	101	erreur de type REP_DEFAULT%
DOC_ERRREPOPEN%	100	erreur de type REP_OPEN%
DOC_ERRREPSTART%	102	erreur de type REP_START%
DOC_ERRSTACKOVERFLOW%	13	overflow : trop d'imbrications de boucles (WHILE, REPEAT ou LOOP)
DOC_ERRSYNTAXERROR%	10	erreur de syntaxe
DOC_ERRTOOMANYINCLUDES%	11	trop de fichiers inclus (plus de 8)
DOC_ERRUNKNOWNVARIABLE%	12	variable non déclarée par /DEFINE

Les fonctions de l'A.P.I. DOC_MERGE, DOC_PRINT%, DOC_PRINTFILE%, DOC_EXTPRINT%, DOC_EXTPRINTFILE% et DOC_EXTPRINTSTART% renvoient également certains de ces codes.

Voir aussi DOC ERROR%, DOC MERGE, DOC PRINT%, DOC PRINTFILE%, DOC EXTPRINT%, DOC EXTPRINTFILE%, DOC EXTPRINTSTART%

Constantes DOC_FILE_ *%

Formats de sauvegarde des documents.

Valeurs	Déclaration interne	Description
DOC_FILE_NSWP%	0	format naturel de NSWP, la totalité des informations de formatage sont sauvegardées.
DOC_FILE_TEXT%	1	format ASCII sous OS/2, format ANSI sous Windows, les informations de formatage et de tabulations sont perdues.
DOC_FILE_RTF%	2	Rich Text Format, format supporté par de nombreux outils (Microsoft Word par exemple), la plupart des informations de formatage sont sauvegardées.
DOC_FILE_RFT%	3	Revisable Form Text, format supporté par les outils IBM, la plupart des informations de formatage sont sauvegardées.

Le format n'est pas à préciser lors d'un chargement par DOC_OPEN, NSWP reconnaissant automatiquement le format du fichier à ouvrir.

Voir aussi DOC_SAVE, Fonction DOC_FILE FORMAT, Annexe "Format de fichier NS-WP".

Constantes DOC_FIND_**%

Modes de recherche pris en compte dans les recherches et remplacements de chaînes de caractère.

Valeurs	Déclaration interne	Description
DOC_FIND_LOW2UPCASE%	1	pas de distinction entre les caractères majuscules et les caractères minuscules dans la chaîne recherchée.
DOC_FIND_WHOLEWORD%	2	seules les chaînes identiques à la chaîne recherchée et constituant un mot entier sont prises en compte.
DOC_FIND_DEFAULT%	0	mode de recherche et de remplacement par défaut : <ul style="list-style-type: none"> ▪ Distingue les caractères minuscules et majuscules dans la chaîne recherchée. ▪ Ne se limite pas au mot entier. ▪ Ne demande pas confirmation à chaque remplacement.
DOC_FIND_CONFIRM%	4	demande de confirmation avant chaque remplacement de chaîne.

Ces constantes sont des puissances de 2 : elles peuvent donc être combinées. Ainsi, DOC_FIND_LOW2UPCASE% + DOC_FIND_WHOLEWORD% signifie une recherche ne portant que sur les mots entiers et devant s'effectuer sans distinction de majuscules/minuscules.

Voir aussi DOC_FIND%, DOC_CHANGE%

Constantes DOC_FMT_*

Attributs de caractère.

Valeurs	Déclaration interne	Description
DOC_FMT_DEFAULT%	0	aucun attribut
DOC_FMT_STANDARD%	1	aucun attribut sauf l'attribut DOC_FMT_UNLOCKED% si celui-ci était appliqué précédemment
DOC_FMT_BOLD%	2	gras
DOC_FMT_ITALIC%	4	italique
DOC_FMT_UNDERLINE%	8	souligné
DOC_FMT_STRIKEOUT%	16	barré
DOC_FMT_UNLOCKED%	128	accessible en écriture, même si un DOC_READONLY est effectué ultérieurement sur le texte

Ces constantes sont des puissances de 2 : elles peuvent donc être combinées. Ainsi, DOC_FMT_ITALIC% + DOC_FMT_UNDERLINE% permet de définir le style italique souligné.

Emploi de DOC_FMT_UNLOCKED%. Une fois le texte mis en mode "lecture seulement" au moyen de l'instruction DOC_READONLY, il n'est plus possible de changer un attribut de caractère, et donc de positionner l'attribut DOC_FMT_UNLOCKED%. Pour être valide, cet attribut doit donc être positionné avant le passage du texte en mode "lecture seulement".

Voir aussi DOC_SETFORMAT, DOC_GETFORMAT%, DOC_MODIFYFORMAT, DOC_SETFONT, DOC_GETFONT\$, DOC_READONLY

Constantes DOC_JUS_**%

Types d'alignements applicables à un paragraphe ou à une page.

Valeurs	Déclaration interne	Description
DOC_JUS_LEFT%	0	alignement à gauche d'un paragraphe
DOC_JUS_RIGHT%	1	alignement à droite d'un paragraphe
DOC_JUS_CENTERED%	2	centrage d'un paragraphe (par rapport aux bords gauche et droit), centrage du texte constituant une page (par rapport aux bords haut et bas)
DOC_JUS_FULL%	3	justification d'un paragraphe (par rapport aux bords gauche et droit), justification d'une page (par rapport aux bords haut et bas)
DOC_JUS_LOWER%	1	alignement du texte constituant une page sur le bas de page
DOC_JUS_UPPER%	0	alignement du texte constituant une page sur le haut de page

Voir aussi DOC_GETJUSTIF%, DOC_SETJUSTIF, DOC_GETSECTIONJUSTIF%, DOC_SETSECTIONJUSTIF

Constantes DOC_LEADTAB_*

Styles de point de suite pouvant être associés aux tabulations.

Valeurs	Déclaration interne	Description
DOC_LEADTAB_NONE%	0	aucun point de suite
DOC_LEADTAB_DOT%	1
DOC_LEADTAB_HYPHEN%	2	-----
DOC_LEADTAB_UNDERSCORE%	3	_____

Voir aussi Segment DOC_XTABS, DOC_GETCURRENTTAB, DOC_SETCURRENTTAB

Constantes DOC_LINE_*

Eléments d'un encadrement.

Valeurs	Déclaration interne	Description
DOC_LINE_LEFT%	0	bord gauche
DOC_LINE_RIGHT%	1	bord droit
DOC_LINE_ABOVE%	2	bord haut
DOC_LINE_BELOW%	3	bord bas
DOC_LINE_BETWEEN%	4	ligne séparant les paragraphes faisant partie d'un même cadre
DOC_LINE_SHADOW%	5	ombre associée au cadre

Voir aussi [Segment DOC BORDER](#)

Constantes DOC_LINEKIND_**%

Types de traits d'un encadrement.

Valeur	Déclaration interne	Description
DOC_LINEKIND_SIMPLE%	0	trait simple
DOC_LINEKIND_DOUBLE%	1	trait double

Voir aussi Segments LINE BORDER, DOC TABLINE

Constantes DOC_LO*%, DOC_CHARS%

Systèmes de mesure.

Valeur	Déclaration interne	Description
DOC_LOENGLISH%	0	anglo-saxon L'unité est le 1/100 de pouce (inch).
DOC_LOMETRIC%	1	métrique L'unité est le 1/10 de mm.
DOC_CHARS%	2	caractère L'unité est le caractère.

Le mode de fonctionnement "caractère" s'utilise exclusivement avec une police non proportionnelle.

Cette police doit être spécifiée au moyen de l'instruction DOC_DEFAULTFONT.

Chaque ligne fait au maximum un certain nombre de caractères. Ce maximum est précisé au moyen des instructions DOC_DEFAULTMAXPARAWIDTH ou DOC_SETMAXPARAWIDTH. La taille des caractères est forcée à la taille de la police n°1.

Le chargement et l'enregistrement d'un document saisi en mode "caractère" se font avec les fonctions ESCLOAD% et ESCPRINT%. Les objets graphiques (tabulations verticales, encadrements, images bitmaps...), les en-têtes et les pieds de page ne sont pas reconnus en mode "caractère".

Le fonctionnement de NS-WP en mode "caractère" est donc proche de celui d'un éditeur de texte. Ses applications en sont multiples, à chaque fois que le besoin d'un outil de type éditeur existe (saisie des messages dans une messagerie par exemple).

Voir aussi DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%,
DOC_DEFAULTMAXPARAWIDTH, ESCLOAD%, ESCPRIINT%, DOC_DEFAULTFONT,
DOC_DEFAULTLOCKFONTS

Constantes DOC_MODE_*

Modes d'affichage.

Valeur	Déclaration interne	Description
DOC_MODE_DEFAULT%	0	<p>affichage du document en mode "standard".</p> <p>Les caractères gras, italique et souligné sont affichés. Les sauts de page, forcés et implicites, sont affichés.</p>
DOC_MODE_DRAFT%	1	<p>affichage du document en mode "brouillon".</p> <p>La police utilisée à l'affichage est la police système. Les caractères gras et italiques sont affichés en souligné. La présentation à l'écran ne sera pas celle obtenue sur l'imprimante. Seuls les sauts de page forcés sont affichés.</p> <p>L'affichage et le déroulement sont plus rapides qu'en mode "standard".</p>
DOC_MODE_PAGE%	2	<p>affichage du document en mode "page".</p> <p>La présentation et les polices utilisées à l'écran sont proches de ce qui est obtenu sur l'imprimante. Les caractères gras, italique et souligné sont affichés. Les sauts de page, forcés et implicites, sont affichés.</p> <p>L'affichage et le déroulement du texte sont plus lents qu'en mode "standard".</p>

Voir aussi DOC_SETMODE, DOC_GETMODE%, DOC_ASSOCIATEPRINTER, DOC_SETLIKEPRINTER, DOC_ISLIKEPRINTER%

Constantes DOC_PARA_*

Caractéristiques du paragraphe.

Valeur	Déclaration interne	Description
DOC_PARA_PAGEBREAK%	1	contient un saut de page
DOC_PARA_KEEPTOGETHER%	2	lignes solidaires
DOC_PARA_KEEPPWITHNEXT%	4	paragraphe solidaire avec le paragraphe suivant
DOC_PARA_SECTION%	8	contient un saut de section

Voir aussi Segment DOC RULEINFO

Constantes DOC_POS_*

Localisations dans la fenêtre des coordonnées obtenues avec DOC_POS2XYVIEW% ou DOC_XYVIEW2POS%.

Valeur	Déclaration interne	Description
DOC_POS_INSIDE%	0	à l'intérieur de l'aire client de la fenêtre
DOC_POS_LEFT%	1	à gauche de l'aire client
DOC_POS_RIGHT%	2	à droite de l'aire client (par exemple sur la barre de défilement verticale)
DOC_POS_TOP%	4	au dessus de l'aire client
DOC_POS_BOTTOM%	8	au dessous de l'aire client (par exemple sur la barre de défilement horizontale)

Voir aussi DOC_POS2XYVIEW%: DOC_XYVIEW2POS%

Constantes DOC_RCHG_*

Eléments de la fenêtre NS WP.

Valeur	Déclaration interne	Description
DOC_RCHG_COLUMN	1	abscisse
DOC_RCHG_LINE	2	ordonnée
DOC_RCHG_PAGE	4	page courante
DOC_RCHG_FACENAME	8	nom de la police
DOC_RCHG_FACESIZE	16	taille de la police
DOC_RCHG_FORMAT	32	format : gras, italique...
DOC_RCHG_COLORS	64	couleurs
DOC_RCHG_JUSTIF	128	alignement
DOC_RCHG_INDENTS	256	marges
DOC_RCHG_TABS	512	tabulations
DOC_RCHG_CONTEXT	1024	partie du document affichée
DOC_RCHG_USERFORMAT	2048	entier réservé à l'utilisateur
DOC_RCHG_USERPARAINFO	4096	information utilisateur locale ou globale
DOC_RCHG_PARAINFO	8192	attributs
DOC_RCHG_SECTION	16384	section courante

Voir aussi Constantes DOC_USER_MESSAGE_*, Segment DOC_RULEINFO, DOC_MAPUSERMSG

Constantes DOC_SCN_*

Valeurs retournées par les fonctions DOC_SCANFIRST% et DOC_SCANNEXT%.
Hormis DOC_SCN_ERR%, ces valeurs représentent les différents éléments du document.

Valeurs	Déclaration interne	Description
DOC_SCN_ENDPARA%	0	marque de fin de paragraphe
DOC_SCN_TEXT%	1	suite de caractères
DOC_SCN_SPACE%	2	espace
DOC_SCN_TAB%	3	marque de tabulation
DOC_SCN_FONT%	4	nouvelle police de caractères
DOC_SCN_TAG%	5	borne
DOC_SCN_BEGMARK%	6	début d'un marqueur
DOC_SCN_ENDMARK%	7	fin de marqueur
DOC_SCN_BEGBMP%	8	début de commande bitmap
DOC_SCN_ENDBMP%	9	fin de commande bitmap
DOC_SCN_ERR%	-1	erreur d'utilisation de DOC_FREESCANCONTEXT%, DOC_SCANFIRST% ou de DOC_SCANNEXT%

DOC_SCN_ERR% est également retournée par DOC_FREESCANCONTEXT%.

Voir aussi Segment DOC_SCAN_CTXT, DOC_FREESCANCONTEXT%, DOC_SCANFIRST%, DOC_SCANNEXT%

Constantes DOC_SHOW_*, DOC_HIDE_PICTURES%

Caractères spécifiques pouvant être visualisés au moyen de DOC_SETCONTROLCHARSVISIBILITY.

Valeur	Déclaration interne	Description
DOC_SHOW_SPACES%	1	espaces
DOC_SHOW_TABS%	2	tabulations
DOC_SHOW_PARAGRAPHS%	4	marques de fin de paragraphe
DOC_SHOW_COMMANDS%	8	commandes (bitmaps, marqueurs)
DOC_HIDE_PICTURES%	16	cadres grisés à la place des images bitmaps
DOC_SHOW_TAGS%	32	bornes

Ces constantes sont des puissances de 2 : elles peuvent donc être combinées. Ainsi, DOC_SHOW_SPACES% + DOC_SHOW_PARAGRAPHS% permet de visualiser les espaces et les marques de fin de paragraphe.

Voir aussi DOC_GETCONTROLCHARSVISIBILITY%, DOC_SETCONTROLCHARSVISIBILITY

Constantes DOC_TYPTAB_**%

Types d'alignement des tabulations.

Valeur	Déclaration interne	Description
DOC_TYPTAB_LEFT%	1	alignement à gauche
DOC_TYPTAB_CENTER%	2	alignement centré
DOC_TYPTAB_RIGHT%	3	alignement à droite
DOC_TYPTAB_DECIMAL%	4	alignement décimal

Voir aussi Segment DOC_XTABS, DOC_GETCURRENTTAB, DOC_SETCURRENTTAB

Constantes DOC_USER_MESSAGE_*

Numéros des événements utilisateur USER0, ..., USER15 employés avec l'instruction DOC_MAPUSERMSG.

Valeur	Déclaration interne
DOC_USER_MESSAGE_0	0
DOC_USER_MESSAGE_1	1
DOC_USER_MESSAGE_2	2
DOC_USER_MESSAGE_3	3
DOC_USER_MESSAGE_4	4
DOC_USER_MESSAGE_5	5
DOC_USER_MESSAGE_6	6
DOC_USER_MESSAGE_7	7
DOC_USER_MESSAGE_8	8
DOC_USER_MESSAGE_9	9
DOC_USER_MESSAGE_10	10
DOC_USER_MESSAGE_11	11
DOC_USER_MESSAGE_12	12
DOC_USER_MESSAGE_13	13
DOC_USER_MESSAGE_14	14
DOC_USER_MESSAGE_15	15

Voir aussi DOC_MAPUSERMSG

Constantes DOC_WIN_*

Parties du document.

Valeur	Déclaration interne	Description
DOC_WIN_BODY%	0	corps du document
DOC_WIN_HEADER%	1	en-tête de page (pour la section courante)
DOC_WIN_FOOTER%	2	pied de page (pour la section courante)
DOC_WIN_FIRSTHEADER%	3	en-tête de page (première page de la section)
DOC_WIN_FIRSTFOOTER%	4	pied de page (première page de la section)

Voir aussi DOC SETWINDOW, DOC GETWINDOW%, DOC ISWINDOW%, DOC DISPOSEWINDOW

Constantes TAB_RULER_OBJECT%, TABLINE_RULER_OBJECT%

Syntaxe	Déclaration interne	Description
TAB_RULER_OBJECT%	1	taquets de tabulation pouvant être positionnés sur la règle par un double-clic souris.
TABLINE_RULER_OBJECT%	2	taquets de tabulation verticale pouvant être positionnés sur la règle par un double-clic souris.

Voir aussi DOC GETCURRENTRULEROBJECT%

Constantes TE*%

Types de commandes figurant dans un document sauvegardé au format texte.

Valeur	Déclaration interne	Description
TESTARTDOC%	0	début de document
TESTOPDOC%	1	fin de document
TEFORMFEED%	2	formfeed
TELINEFEED%	3	linefeed
TEFONTNORMAL%	4	police par défaut
TEFONT001%	5	police d'index 1
TEFONT002%	6	police d'index 2
TEFONT003%	7	police d'index 3
TEFONT004%	8	police d'index 4
TEFONT005%	9	police d'index 5
TEFONT006%	10	police d'index 6
TEFONT007%	11	police d'index 7
TEFONT008%	12	police d'index 8

Voir aussi Segment TXTESCTABLE, ESCLOAD%, ESCPRIINT%

Fonctions

Fonction DOC_ADDPICTDFN%

Charge en mémoire l'image bitmap spécifiée dans nom-bitmap et lui attribue un identificateur.

Syntaxe	DOC_ADDPICTDFN% (<i>handle-WP, nom-bitmap, id-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-bitmap	CSTRING	!V! chemin et nom du fichier contenant l'image bitmap
	id-bitmap	INT(2)	!V! identificateur attribué à l'image bitmap
Valeur retournée	INT(2) TRUE% : l'image bitmap est chargée en mémoire. FALSE% : l'image bitmap n'a pas pu être chargée.		

1. Les formats reconnus sont les formats bitmap d'extension .BMP.
2. A la suite de DOC_ADDPICTDFN%, DOC_ERROR% peut retourner les codes d'erreur DOC_ERRDUPLPCTNAME% et DOC_ERRINVALIDPCT%.

Exemple :

```

LOCAL CSTRING NAME$
LOCAL ID%(2)

MOVE "C:\NSDK\BMP\RABBIT.BMP" TO NAME$
IF DOC_ADDPICTDFN%(HWP%, NAME$, ID%) = FALSE%
IF DOC_ERROR% (HWP%) = DOC_ERRDUPLPCTNAME%
MESSAGE "Avertissement", "Image bitmap déjà chargée en mémoire"
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Image bitmap non chargée"
...
ENDIF
ENDIF

```

Voir aussi Constantes [DOC_ERR*%](#), [DOC_ERROR%](#), [DOC_DELPCTDFN%](#), [DOC_GETPCTDFNID%](#), [DOC_GETPCTDFNNAME%](#)

Fonction **DOC_ALLOCATEDMEM%**

Retourne la taille de la mémoire allouée à NS-WP.

Syntaxe	DOC_ALLOCATEDMEM%
Valeur retournée	INT(4) Taille mémoire en octets.

Exemple :

```
MESSAGE "Taille allouée à NS-WP", DOC_ALLOCATEDMEM%
```

Fonction DOC_ALLOCSCANCONTEXT%

Alloue un contexte de recherche. Un segment DOC_SCAN_CTXT est attribué à ce contexte. Il contiendra les résultats du parcours du document effectué dans ce contexte.

Syntaxe	DOC_ALLOCSCANCONTEXT%
Valeur retournée	INT(4) Handle sur le contexte de recherche alloué.

Exemple :

```
LOCAL HSC%  
MOVE DOC_ALLOCSCANCONTEXT% TO HSC%
```

Voir aussi Segment DOC_SCAN_CTXT, DOC_FREESCANCONTEXT%, DOC_SCANFIRST%, DOC_SCANNEXT%

Fonction DOC_CHANGE%

Recherche dans l'ensemble du document la chaîne de caractères chaîne-recherche et remplace chacune de ces occurrences par la chaîne de caractères chaîne-remplace.

Le paramètre mode-remplace précise comment doit s'effectuer la recherche : prise en compte ou non des minuscules/majuscules, limitation ou non au mot entier, confirmation ou non de remplacement. Il est précisé par les constantes DOC_FIND_*.

Syntaxe	DOC_CHANGE% (<i>handle-WP, chaîne-recherche, chaîne-remplace, mode remplace</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	chaîne-recherche	CSTRING	chaîne de caractères à rechercher
	chaîne-remplace	CSTRING	chaîne de caractères devant remplacer la chaîne recherchée
	mode-remplace	INT(1)	mode de recherche/remplacement
Valeur retournée	INT(4) Nombre d'occurrences effectivement remplacées.		

Si une confirmation a été demandée (mode de remplacement DOC_FIND_CONFIRM%), un événement USER0 est envoyé à la fenêtre NS-WP à chaque occurrence trouvée. Il permet de demander à l'utilisateur s'il désire ou non effectuer le remplacement.

Pour gérer cette demande, il faut :

- Ajouter l'événement utilisateur USER0 à la fenêtre NS-WP.
- Coder le traitement permettant de poser une question à l'utilisateur et de fournir la réponse à DOC_CHANGE%. Cela peut être réalisé à l'aide des fonctions ASK2% ou ASK3% du NCL ou par l'intermédiaire d'une boîte de dialogue. La valeur retournée par l'événement USER0 constitue la réponse donnée à DOC_CHANGE% :
- RETURN YES% : confirme le remplacement et continue
- RETURN NO% : empêche le remplacement et continue
- RETURN CANCEL% : empêche le remplacement et arrête la recherche

Si l'événement USER0 n'est pas traité ou ne retourne rien, le remplacement est effectué.

Exemple :

```
LOCAL NBOCC%  
; Remplacement de toutes les occurrences de la chaîne "Monsieur" par la
```



```
chaîne "Madame"
MOVE DOC_CHANGE%(HWP%, "Monsieur", "Madame" , DOC_FIND_WHOLEWORD%) TO
NBOCC%

; Remplacement de la chaîne "premier" par "DERNIER" sans tenir compte des
majuscules/minuscules et avec demande de confirmation sur chaque occurrence
trouvée
MOVE DOC_CHANGE%(HWP%, "premier",
"DERNIER",DOC_FIND_LOW2UPCASE% + DOC_FIND_CONFIRM%) TO NBOCC%
; Evénement USER0 pour traiter la demande de confirmation :
RETURN ASK3%("Attention","Confirmez-vous le remplacement ?")
```

Voir aussi Constantes DOC_FIND *%, DOC_FIND%, Constantes YES%, NO%, CANCEL% (Manuel de Programmation NCL)

Fonction **DOC_CURRENTPARAGRAPH%**

Retourne le handle du paragraphe courant.

Syntaxe	DOC_DEFAULTMETRICS syst-mesure		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(4) Handle du paragraphe courant.		

Exemple :

```
; Recherche de la position du paragraphe courant
LOCAL HPARA%(4), I%(4), H%(4)
MOVE 1 TO I%
MOVE DOC_CURRENTPARAGRAPH%(HWP%) TO HPARA%
MOVE DOC_FIRSTPARAGRAPH%(HWP%) TO H%
WHILE H% <> HPARA%
MOVE I%+1 TO I%
MOVE DOC_NEXTPARAGRAPH%(HWP%, H%) TO H%
ENDWHILE
MESSAGE "Recherche faite", "Le paragraphe courant est le" && I% & "ème du
document."
```

Voir aussi **DOC_FIRSTPARAGRAPH%**, **DOC_LASTPARAGRAPH%**, **DOC_PREVPARAGRAPH%**,
DOC_NEXTPARAGRAPH%, **DOC_GETPOS**

Fonction DOC_DELETEBITMAPCHAR%

Supprime l'image bitmap située à droite du curseur.

Syntaxe	DOC_DELETEBITMAPCHAR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) TRUE% si l'image bitmap a été supprimée, FALSE% sinon.		

1. A la suite de DOC_DELETEBITMAPCHAR%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOPICTATCURSPOS%.
2. Une image bitmap, qui est vue dans le texte comme un caractère, peut aussi être effacée au moyen des touches [Supp] ou [Backspace].

Exemple :

```
; Suppression de l'image bitmap située à droite du curseur
IF DOC_DELETEBITMAPCHAR% (HWP%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR% (HWP%), "La suppression a échoué"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_INSERTBITMAPCHAR%, DOC_GETBITMAPCHAR%, DOC_MODIFYBITMAPCHAR%

Fonction DOC_DELPICDFN%

Libère de la mémoire l'image bitmap précédemment chargée avec DOC_ADDPICDFN% et dont l'identificateur est id-bitmap.

Syntaxe	DOC_DELPICDFN% (<i>handle-WP, id-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	identificateur attribué à l'image bitmap
Valeur retournée	INT(2) TRUE% : la mémoire est libérée. FALSE% : la mémoire n'a pas pu être libérée.		

A la suite de DOC_DELPICDFN%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOTAPICTID%.

Exemple :

```
LOCAL ID%(2)
LOCAL CSTRING NAME$

IF DOC_DELPICDFN% (HWP%, NAME$, ID%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Mémoire non libérée"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_ADDPICDFN%, DOC_GETPICDFNID%, DOC_GETPICDFNNAME%

Fonction DOC_ERROR%

Retourne le code d'erreur de la dernière fonction / instruction employée de la librairie NSWP.

Syntaxe	DOC_ERROR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(4) Une des constantes DOC_ERR*%.		

Les fonctions / instructions concernées sont :

- DOC_OPEN (et LOAD)
- DOC_INSERT
- DOC_SAVE (et SAVE)
- DOC_MERGE
- DOC_EXTPRINT%
- DOC_EXTPRINTFILE%
- DOC_EXTPRINTSTART%
- DOC_PRINT%
- DOC_PRINTFILE%
- DOC_ADDPICTDFN%
- DOC_DELPICTDFN%
- DOC_GETPICTDFNID%
- DOC_GETPICTDFNNAME%
- DOC_DELETEBITMAPCHAR%
- DOC_GETBITMAPCHAR%
- DOC_INSERTBITMAPCHAR%
- DOC_MODIFYBITMAPCHAR%

Exemple :

```
DOC_OPEN HWP%, "FICHER.DOC"
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Fichier non chargé"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_OPEN, DOC_INSERT, DOC_SAVE, DOC_MERGE, DOC_EXTPRINT%, DOC_EXTPRINTFILE%, DOC_EXTPRINTSTART%, DOC_PRINT%, DOC_PRINTFILE%, DOC_ADDPICTDFN%, DOC_DELPICTDFN%, DOC_GETPICTDFNID%, DOC_GETPICTDFNNAME%, DOC_DELETEBITMAPCHAR%, DOC_GETBITMAPCHAR%, DOC_INSERTBITMAPCHAR%, DOC_MODIFYBITMAPCHAR%, DOC_GETLASTMERGEERROR

Fonction DOC_EXTPRINT%

Imprime une partie ou la totalité du document, au sein du job d'impression handle-job.

Cette impression ne porte que sur des pages entières. La partie à imprimer est donc bornée par deux numéros de pages.

Le job d'impression doit avoir été démarré auparavant avec DOC_EXTPRINTSTART%.

La chaîne descript-imp permet de préciser les pages à imprimer. Elle a la structure suivante :

```
"/STARTFROM:valeur /STOPAT:valeur /FIRST:valeur"
```

/STARTFROM est suivi du numéro de la première page à imprimer, /STOPAT du numéro de la dernière page à imprimer. Les pages sont numérotées à partir de 1.

/FIRST est suivi du numéro donné à la première page du document à l'impression. Ce numéro permet de faire suivre plusieurs documents, en faisant à chaque fois recommencer la numérotation là où s'était arrêtée l'impression précédente.

Syntaxe	DOC_EXTPRINT% (<i>handle-job, handle-WP, descript-imp, nb-pages</i>)		
Paramètres	handle-job	INT(4)	handle du job d'impression
	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-imp	CSTRING	descriptif de l'impression à effectuer
	nb-pages	INT(2)	!V! nombre de pages effectivement imprimées
Valeur retournée	INT(2) Code d'erreur : une des constantes DOC_ERR*%.		

1. Le code d'erreur peut être aussi retourné par la fonction DOC_ERROR%.
2. La fonction DOC_PRINT% permet également d'imprimer le document. Avec cette fonction, une session d'impression est démarrée à chaque impression.

Exemple :

```
; Impression de la page 10 du document en lui donnant le numéro 11
LOCAL CSTRING PARAM$
LOCAL NB%(2), STARTFROM%(2), STOPAT%(2), FIRST%(2)

IF DOC_EXTPRINTSTART%(0, HJOB%, "Impression NS-WP") = DOC_ERRNO%
MOVE 10 TO STARTFROM%
MOVE 10 TO STOPAT%
MOVE 2 TO FIRST%
MOVE "/STARTFROM:" & STARTFROM% &&
"/STOPAT:" & STOPAT% &&
"/FIRST:" & FIRST% TO PARAM$
IF DOC_EXTPRINT%(HJOB%, HWP%, PARAM$, NB%) = DOC_ERRNO%
MESSAGE "OK", "Impression effectuée"
```

```
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "L'impression a échoué"
ENDIF
...
DOC_EXTPRINTSTOP HJOB%
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Le démarrage du job a échoué"
ENDIF
```

Voir aussi Constantes DOC ERR*%, DOC ERROR%, DOC EXTPRINTSTART%, DOC EXTPRINTSTOP, DOC EXTPRINTFILE%, DOC PRINT%, Librairie NSPRN (Manuel de Programmation Librairies)

Fonction DOC_EXTPRINTFILE%

Imprime un fichier, au sein du job d'impression handle-job.

Cette impression ne porte que sur des pages entières. La partie à imprimer est donc bornée par deux numéros de pages.

Le job d'impression doit avoir été démarré auparavant avec DOC_EXTPRINTSTART%.

La chaîne descript-imp permet de préciser les pages à imprimer. Elle a la structure suivante :

```
"/STARTFROM:valeur /STOPAT:valeur /FIRST:valeur"
```

/STARTFROM est suivi du numéro de la première page à imprimer, /STOPAT du numéro de la dernière page à imprimer. Les pages sont numérotées à partir de 1.

/FIRST est suivi du numéro donné à la première page du document à l'impression. Ce numéro permet de faire suivre plusieurs documents, en faisant à chaque fois recommencer la numérotation là où s'était arrêtée l'impression précédente.

Syntaxe	DOC_EXTPRINTFILE% (<i>handle-job, nom-fichier, descript-imp, nb-pages</i>)		
Paramètres	handle-job	INT(4)	handle du job d'impression
	nom-fichier	CSTRING	chemin et nom du fichier à imprimer
	descript-imp	CSTRING	descriptif de l'impression à effectuer
	nb-pages	INT(2)	!V! nombre de pages effectivement imprimées
Valeur retournée	INT(2) Code d'erreur : une des constantes DOC_ERR*%.		

1. Il n'est pas nécessaire d'ouvrir préalablement le fichier. Il est même possible d'imprimer un document alors qu'un autre document est visualisé dans la fenêtre NS-WP.

2. Le format du fichier doit être celui d'un document NS-WP ou d'un document texte ASCII.

3. Le code d'erreur peut être aussi retourné par la fonction DOC_ERROR%.

4. La fonction DOC_PRINTFILE% permet également d'imprimer un fichier. Avec cette fonction, une session d'impression est démarrée à chaque impression.

Exemple :

```
; Impression de la page 10 du fichier "FICHIER.DOC" en lui donnant le
numéro 11
LOCAL CSTRING PARAM$
LOCAL NB%(2), STARTFROM%(2), STOPAT%(2), FIRST%(2)
LOCAL ERROR%(2)
```



```
MOVE DOC_EXTPRINTSTART%(0, HJOB%, "Impression NS-WP") TO ERROR%
IF ERROR% = DOC_ERRNO%
MOVE 10 TO STARTFROM%
MOVE 10 TO STOPAT%
MOVE 2 TO FIRST%
MOVE "/STARTFROM:" & STARTFROM% &&
"/STOPAT:" & STOPAT% &&
"/FIRST:" & FIRST% TO PARAM$
MOVE DOC_EXTPRINTFILE%(HJOB%, "FICHIER.DOC", PARAM$, NB%) TO ERROR%
IF ERROR% = DOC_ERRNO%
MESSAGE "OK", "Impression effectuée"
ELSE
MESSAGE "Erreur" && ERROR%, "L'impression a échoué"
ENDIF
...
DOC_EXTPRINTSTOP HJOB%
ELSE
MESSAGE "Erreur" && ERROR%, "Le démarrage du job a échoué"
ENDIF
```

Voir aussi Constantes DOC ERR*%, DOC ERROR%, DOC PRINT%, DOC_EXTPRINTSTART%, DOC_EXTPRINTSTOP, DOC_EXTPRINT%, Librairie NSPRN (Manuel de Programmation Librairies)

Fonction DOC_EXTPRINTSTART%

Démarre un job NS-WP dans le spooler d'impression.

handle-imprimante est obtenu avec les fonctions REP_OPEN% et REP_DEFAULT% de la librairie NSPRN. handle-imprimante peut également être initialisé à 0. Dans ce cas, l'imprimante prise en compte est l'imprimante associée à la fenêtre NS-WP, ou l'imprimante attachée par défaut à la machine si aucune association n'a été effectuée.

Syntaxe	DOC_EXTPRINTSTART% (<i>handle-imprimante, handle-job, titre-imp</i>)		
Paramètres	handle-imprimante	INT(4)	handle de l'imprimante
	handle-job	INT(4)	handle du job d'impression
	titre-imp	CSTRING	texte qui apparaît dans le spooler du système
Valeur retournée	INT(2) Code d'erreur : une des constantes DOC_ERR*%.		

1. Plusieurs impressions peuvent être groupées dans un même job d'impression. Cela permet d'imprimer diverses parties d'un ou plusieurs documents à la suite, sans que des feuilles blanches ou d'autres travaux viennent s'intercaler.
2. Le code d'erreur peut être aussi retourné par la fonction DOC_ERROR%.

Exemple :

```
LOCAL HJOB%
IF DOC_EXTPRINTSTART%(0, HJOB%, "Impression NS-WP") = DOC_ERRNO%
...
DOC_EXTPRINTSTOP HJOB%
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Le démarrage du job a échoué"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_EXTPRINT%, DOC_EXTPRINTFILE%, DOC_EXTPRINTSTOP, DOC_ASSOCIATEPRINTER, Librairie NSPRN (Manuel de Programmation Librairies)

Fonction **DOC_FILE_FORMAT%**

Retourne le format du document chargé dans la fenêtre NS-WP.

Syntaxe	DOC_FILE_FORMAT% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes DOC_FILE_* %.		

Exemple :

```
; Sauvegarde au format courant  
DOC_SAVE HWP%, "FICHIER.DOC", DOC_FILE_FORMAT%(HWP%)
```

Voir aussi Constantes **DOC_FILE_***%, **DOC_SAVE**

Fonction DOC_FIND%

Recherche la chaîne de caractères chaîne-recherche dans l'ensemble du document courant.

La recherche s'effectue de la position du curseur à la fin du document, puis du début du document jusqu'à la position du curseur.

Si la chaîne est trouvée, elle est automatiquement sélectionnée.

Le paramètre mode-recherche permet de spécifier le mode de recherche : prise en compte ou non des majuscules/minuscules, limitation ou non au mot entier. Il est précisé par les constantes DOC_FIND_*.

Syntaxe	DOC_FIND% (<i>handle-WP, chaîne-recherche, mode-recherche</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	chaîne-recherche	CSTRING	chaîne de caractères à rechercher
	mode-recherche	INT(1)	mode de recherche
Valeur retournée	INT(1) TRUE% : la chaîne a été trouvée. FALSE% : la chaîne n'a pas été trouvée.		

Exemple :

```
IF DOC_FIND%(HWP%, "Madame", DOC_FIND_WHOLEWORD%)
MESSAGE "OK", "Madame a été trouvé"
ELSE
MESSAGE "Non", "Pas de Madame dans le document"
ENDIF
```

Voir aussi Constantes DOC_FIND_*, DOC_CHANGE%

Fonction **DOC_FIRSTPARAGRAPH%**

Retourne le handle du premier paragraphe.

Syntaxe	DOC_FIRSTPARAGRAPH% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(4) Handle du premier paragraphe.		

DOC_FIRSTPARAGRAPH% retourne toujours un handle même dans le cas d'une fenêtre vide, une fenêtre vide étant constituée d'un paragraphe vide.

Exemple :

```
; Sélection du premier paragraphe
DOC_SETSELECTION HWP%, DOC_FIRSTPARAGRAPH%(HWP%), 0 ,
DOC_NEXTPARAGRAPH%(HWP%, DOC_FIRSTPARAGRAPH%(HWP%)), 0
```

Voir aussi [DOC_CURRENTPARAGRAPH%](#), [DOC_LASTPARAGRAPH%](#), [DOC_PREVPARAGRAPH%](#), [DOC_NEXTPARAGRAPH%](#)

Fonction DOC_FREESCANCONTEXT%

Libère un contexte de recherche précédemment alloué avec DOC_ALLOCSCANCONTEXT%.

Syntaxe	DOC_FREESCANCONTEXT% (<i>handle-contexte</i>)		
Paramètres	handle-contexte	INT(4)	handle du contexte de recherche
Valeur retournée	INT(2) DOC_SCN_ERR% si l'opération échoue, 0 sinon.		

Exemple :

```
IF DOC_FREESCANCONTEXT% (HSC%) = DOC_SCN_ERR%  
MESSAGE "Erreur", "Contexte déjà libéré"  
ENDIF
```

Voir aussi Constantes DOC_SCN_*, DOC_ALLOCSCANCONTEXT%, DOC_SCANFIRST%, DOC_SCANNEXT%

Fonction `DOC_GET_INSERT_MODE%`

Retourne le mode d'insertion des caractères.

Syntaxe	<code>DOC_GET_INSERT_MODE%</code> (<i>handle-WP</i>)		
Paramètres	<code>handle-WP</code>	<code>INT(4)</code>	handle de la fenêtre NS-WP
Valeur retournée	<code>INT(1)</code> <code>TRUE%</code> lorsque le mode "insertion" est activé, <code>FALSE%</code> sinon.		

Exemple :

```
; Passage en mode "insertion"
IF DOC_GET_INSERT_MODE% (HWP%) = FALSE%
DOC_SET_INSERT_MODE HWP%, TRUE%
ENDIF
```

Voir aussi `DOC SET INSERT MODE`

Fonction DOC_GETBITMAPCHAR%

Copie dans id-bitmap, larg-bitmap, haut-bitmap et pos-bitmap les caractéristiques de l'image bitmap située à gauche du curseur.

pos-bitmap représente le décalage de l'image bitmap par rapport à la base de la ligne : un nombre négatif situe l'image bitmap au dessus de la ligne, une valeur nulle sur la ligne précisément, un nombre positif au dessous de la ligne.

larg-bitmap, haut-bitmap et pos-bitmap sont exprimées en pixels.

Syntaxe	DOC_GETBITMAPCHAR% (<i>handle-WP, id-bitmap, larg-bitmap, haut-bitmap, pos bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	!V! identificateur attribué à l'image bitmap
	larg-bitmap	INT(4)	!V! largeur à donner à l'image bitmap
	haut-bitmap	INT(4)	!V! hauteur à donner à l'image bitmap
	pos-bitmap	INT(4)	!V! position de l'image bitmap sur la ligne
Valeur retournée	INT(2) TRUE% lorsque la copie est réussie, FALSE% sinon.		

A la suite de DOC_GETBITMAPCHAR%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOPICTATCURSPOS%.

Exemple :

```
; Obtention des caractéristiques de l'image bitmap située à gauche du
curseur
IF DOC_GETBITMAPCHAR% (HWP%, ID%, WIDTH%, HEIGHT%, OFFSET%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR% (HWP%)
"La recherche a échoué"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_ADDPICTDFN%,
DOC_INSERTBITMAPCHAR%, DOC_MODIFYBITMAPCHAR%, DOC_DELETEBITMAPCHAR%

Fonction **DOC_GETCONTROLCHARSVISIBILITY%**

Retourne un entier représentatif des codes visibles à l'écran. Ces codes ont été déterminés par l'instruction **DOC_SETCONTROLCHARSVISIBILITY**.

Syntaxe	DOC_GETCONTROLCHARSVISIBILITY% (handle-WP)		
Paramètre	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2). Constante DOC_SHOW_* ou combinaison de constantes DOC_SHOW_* .		

Exemple :

```
; Codes visibles
EVALUATE DOC_GETCONTROLCHARSVISIBILITY% (HWP%)
WHERE 1
MESSAGE "Sont visualisés :", "Les espaces"
ENDWHERE
WHERE 2
MESSAGE "Sont visualisés :", "Les tabulations"
ENDWHERE
WHERE 3
MESSAGE "Sont visualisés :", \
"Les espaces et les tabulations"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi Constantes **DOC_SHOW_***, **DOC_HIDE_PICTURES%**, **DOC_SETCONTROLCHARSVISIBILITY**

Fonction DOC_GETCURRENTRULEROBJECT%

Retourne le type des taquets qui seront positionnés sur la règle par un double-clic souris : taquets de tabulation ou taquets de tabulation verticale.

Syntaxe	DOC_GETCURRENTRULEROBJECT% (<i>handle-WP</i>)		
Paramètre	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	TAB_RULER_OBJECT% : taquet de tabulation TABLINE_RULER_OBJECT% : taquet de tabulation verticale		

Exemple :

```
LOCAL TYPE%(1), LEAD%(1)
IF DOC_GETCURRENTRULEROBJECT% (HWP%) = TAB_RULER_OBJECT%
DOC_GETCURRENTTAB HWP%, TYPE%, LEAD%
ELSE
DOC_GETCURRENTTABLINE HWP%, TYPE%, LEAD%
ENDIF
```

Voir aussi Constantes TAB_RULER_OBJECT%, TABLINE_RULER_OBJECT%, DOC_GETCURRENTTAB, DOC_SETCURRENTTAB, DOC_GETCURRENTTABLINE, DOC_SETCURRENTTABLINE

Fonction **DOC_GETDEFTAB%**

Retourne la distance séparant les taquets de tabulation positionnés par défaut.

Syntaxe	DOC_GETDEFTAB% (handle-WP)		
Paramètre	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Distance exprimée dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).		

Exemple :

```
LOCAL WIDTH%(2)
MOVE DOC_GETDEFTAB% (HWP%) TO WIDTH%
```

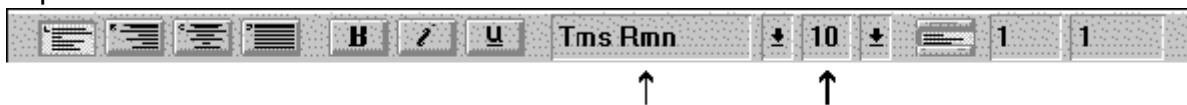
Voir aussi [DOC_SETDEFTAB](#), [DOC_DEFAULTTAB](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#)

Fonction DOC_GETFONT\$

Retourne le nom et la taille de la police au point d'insertion.

Syntaxe	DOC_GETFONT\$ (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	CSTRING. Nom et taille de la police courante. Ces deux informations sont séparées par une virgule.		

Equivalent : Barre d'outils standard



Exemple :

```
; Affichage, par exemple, de "Tms Rmn,10"  
MESSAGE "La police courante est", DOC_GETFONT$(HWP%)
```

Voir aussi [DOC_SETFONT](#), [DOC_GETFONTNAMECONTROL](#), [DOC_GETFONTSIZECONTROL](#)

Fonction DOC_GETFONTAT\$

Retourne le nom et la taille de la police de caractères utilisée à la position indiquée dans handle-parag et pos-caract.

pos-caract est exprimée en nombre de caractères précédant la position.

Syntaxe	DOC_GETFONTAT\$ (<i>handle-WP, handle-parag, pos-caract</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-caract	INT(2)	position dans le paragraphe
Valeur retournée	CSTRING Nom et taille de la police. Ces deux informations sont séparées par une virgule.		

Exemple :

```
; Affichage du nom et de la taille de la police courante (à noter que ce
résultat peut être obtenu directement avec DOC_GETFONT$)
DOC_GETPOS% HWP%, HPARA%, OFFSET%
MESSAGE "La police courante est", DOC_GETFONTAT$ (HWP%, HPARA%, OFFSET%)
```

Voir aussi DOC_GETFONT\$, DOC_SETFONT, DOC_GETFONTNAMECONTROL, DOC_GETFONTSIZECONTROL

Fonction **DOC_GETFONTBACKCOLOR%**

Retourne la couleur d'arrière-plan au point d'insertion.

Syntaxe	DOC_GETFONTBACKCOLOR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes COL_ *%.		

Exemple :

```
; Affichage de la couleur d'arrière-plan
EVALUATE DOC_GETFONTBACKCOLOR% (HWP%)
WHERE COL_BLACK%
MESSAGE "Couleur d'arrière-plan", "Noir"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi [DOC_SETFONTBACKCOLOR](#), [DOC_SETFONTCOLORS](#), *Constantes COL_ *%* (*Manuel de Programmation NCL*)

Fonction **DOC_GETFONTFORECOLOR%**

Retourne la couleur des caractères au point d'insertion.

Syntaxe	DOC_GETFONTFORECOLOR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes COL_ *%.		

Exemple :

```
; Affichage de la couleur des caractères
EVALUATE DOC_GETFONTFORECOLOR% (HWP%)
WHERE COL_BLACK%
MESSAGE "Couleur des caractères", "Noir"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi [DOC_SETFONTFORECOLOR](#), [DOC_SETFONTCOLORS](#), Constantes COL_ *% (*Manuel de Programmation NCL*)

Fonction **DOC_GETFONTINDEXAT%**

Retourne l'index du style de caractères utilisé à la position indiquée dans handle-parag et pos-caract.

pos-caract est exprimée en nombre de caractères précédant la position.

Syntaxe	DOC_GETFONTINDEXAT% (<i>handle-WP, handle-parag, pos-caract</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-caract	INT(2)	position dans le paragraphe
Valeur retournée	INT(2) Index du style de caractères.		

Exemple :

```
MESSAGE "Index du style de caractères", DOC_GETFONTINDEXAT% (HWP%, HPARA%,  
OFFSET%)
```

Voir aussi [DOC_SETFONTINDEXNAME](#), [DOC_SETFONTINDEXSIZE](#), [DOC_SETFONTINDEXFORMAT](#),
[DOC_SETFONTINDEXCOLORS](#), [DOC_SETFONTINDEXUSERFORMAT](#)

Fonction **DOC_GETFONTINDEXFORMAT%**

Retourne les attributs du style de caractères d'index index-style.

Syntaxe	DOC_GETFONTINDEXFORMAT% (<i>handle-WP</i> , <i>index-style</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	Index d'un style de caractères
Valeur retournée	INT(1) Constante DOC_FMT_* ou combinaison de constantes DOC_FMT_* si un style est défini pour l'index spécifié, 0 sinon.		

Exemple :

```
; Affichage des attributs du style d'index 1
EVALUATE DOC_GETFONTINDEXFORMAT% (HWP%, 1)
WHERE 0
MESSAGE "Attributs du style d'index 1", "Aucun"
ENDWHERE
WHERE 1
MESSAGE "Attributs du style d'index 1", "Standard"
ENDWHERE
WHERE 2
MESSAGE "Attributs du style d'index 1", "Gras"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi Constantes **DOC_FMT_***, **DOC_SETFONTINDEXFORMAT**

Fonction **DOC_GETFONTINDEXNAME\$**

Retourne la police du style de caractères d'index index-style.

Syntaxe	DOC_GETFONTINDEXNAME\$ (<i>handle-WP</i> , <i>index-style</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	Index d'un style de caractères
Valeur retournée	CSTRING. Police si un style est défini pour l'index spécifié, chaîne vide sinon.		

Exemple :

```
MESSAGE "Police du style d'index 1", DOC_GETFONTINDEXNAME$ (HWP%, 1)
```

Voir aussi [DOC_SETFONTINDEXNAME](#)

Fonction **DOC_GETFONTINDEXSIZE%**

Retourne la taille des caractères du style d'index index-style.

Syntaxe	DOC_GETFONTINDEXSIZE% (<i>handle-WP, index-style</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	Index d'un style de caractères
Valeur retournée	INT(2) Taille des caractères si un style est défini pour l'index spécifié, 0 sinon.		

Exemple :

```
MESSAGE "Taille des caractères du style d'index 1", DOC_GETFONTINDEXSIZE%  
(HWP%, 1)
```

Voir aussi [DOC_SETFONTINDEXSIZE](#)

Fonction **DOC_GETFONTINDEXUSERFORMAT%**

Retourne l'entier associé au style d'index index-style.

Syntaxe	DOC_GETFONTINDEXUSERFORMAT% (<i>handle-WP, index-style</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	Index d'un style de caractères
Valeur retournée	INT(4) Entier associé si un style est défini pour l'index spécifié, 0 sinon.		

Exemple :

```
MESSAGE "Entier associé au style d'index 1", DOC_GETFONTINDEXUSERFORMAT%  
(HWP%, 1)
```

Voir aussi [DOC_SETFONTINDEXUSERFORMAT,](#) [DOC_GETPARAGRAPHGLOBALUSERINFO%,](#)
[DOC_SETPARAGRAPHGLOBALUSERINFO,](#) [DOC_GETPARAGRAPHLOCALUSERINFO%,](#)
[DOC_SETPARAGRAPHLOCALUSERINFO](#)

Fonction **DOC_GETFONTLINECOLOR%**

Retourne la couleur utilisée pour souligner ou barrer le texte sélectionné.

Syntaxe	DOC_GETFONTLINECOLOR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes COL_*.		

Exemple :

```
EVALUATE DOC_GETFONTLINECOLOR% (HWP%)  
WHERE COL_BLACK%  
MESSAGE "Couleur du souligné ou du barré", "Noir"  
ENDWHERE  
...  
ENDEVALUATE
```

Voir aussi [DOC_SETFONTLINECOLOR](#), Constantes COL_*% (*Manuel de Programmation NCL*)

Fonction DOC_GETFORMAT%

Retourne les attributs de caractère au point d'insertion.

Syntaxe	DOC_GETFORMAT% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Constante DOC_FMT_+% ou combinaison de constantes DOC_FMT_+%. 		

Equivalent : Barre d'outils standard



Exemple :

```
; Cochage des menus selon le format sur événement INIT du menu déroulant
correspondant
LOCAL FMT%(1)
MOVE DOC_GETFORMAT%(HWP%) TO FMT%
IF FMT% BAND DOC_FMT_BOLD%
MOVE CHECKED% TO MENU_GRAS
ELSE
MOVE UNCHECKED% TO MENU_GRAS
ENDIF
IF FMT% BAND DOC_FMT_ITALIC%
MOVE CHECKED% TO MENU_ITALIQUE
ELSE
MOVE UNCHECKED% TO MENU_ITALIQUE
ENDIF
IF FMT% BAND DOC_FMT_UNDERLINE%
MOVE CHECKED% TO MENU_SOULIGNE
ELSE
MOVE UNCHECKED% TO MENU_SOULIGNE
ENDIF
```

Voir aussi Constantes DOC_FMT_+%, DOC_SETFORMAT, DOC_GETFONT\$

Fonction DOC_GETFORMATAT%

Retourne les attributs du caractère spécifié dans handle-parag et pos-caract.
pos-caract est le nombre de caractères précédant le caractère dans le paragraphe.

Syntaxe	DOC_GETFORMATAT% (<i>handle-WP, handle-parag, pos-caract</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle d'un paragraphe
	pos-caract	INT(2)	position du caractère dans le paragraphe
Valeur retournée	INT(1) Constante DOC_FMT_% ou combinaison de constantes DOC_FMT_%.		

Equivalent : Barre d'outils standard



Exemple :

```
; Cochage des menus selon le format sur événement INIT du menu déroulant
correspondant
LOCAL FMT%(1)
GETPOS HWP%, HPARA%, OFFSET%
MOVE DOC_GETFORMATAT%(HWP%, HPARA%, OFFSET%) TO FMT%
IF FMT% BAND DOC_FMT_BOLD%
    MOVE CHECKED% TO MENU_GRAS
ELSE
    MOVE UNCHECKED% TO MENU_GRAS
ENDIF
IF FMT% BAND DOC_FMT_ITALIC%
    MOVE CHECKED% TO MENU_ITALIQUE
ELSE
    MOVE UNCHECKED% TO MENU_ITALIQUE
ENDIF
IF FMT% BAND DOC_FMT_UNDERLINE%
    MOVE CHECKED% TO MENU_SOULIGNE
ELSE
    MOVE UNCHECKED% TO MENU_SOULIGNE
ENDIF
```

Voir aussi Constantes DOC_FMT_*, DOC_SETFORMAT, DOC_GETFONT\$

Fonction DOC_GETJUSTIF%

Retourne le type d'alignement du paragraphe courant.

Syntaxe	DOC_GETJUSTIF% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes DOC_JUS_*.		

1. Le type d'alignement peut également être obtenu par DOC_GETPARAGRAPHINFO.
2. Equivalent : Barre d'outils standard



Exemple :

```
; Cochage des menus selon l'alignement sur événement INIT du menu déroulant
correspondant
MOVE UNCHECKED% TO MENU_GAUCHE
MOVE UNCHECKED% TO MENU_DROIT
MOVE UNCHECKED% TO MENU_CENTRE
MOVE UNCHECKED% TO MENU_JUSTIFIE
EVALUATE DOC_GETJUSTIF%(HWP%)
  WHERE DOC_JUS_LEFT%
    MOVE CHECKED% TO MENU_GAUCHE
  ENDWHERE
  WHERE DOC_JUS_RIGHT%
    MOVE CHECKED% TO MENU_DROIT
  ENDWHERE
  WHERE DOC_JUS_CENTERED%
    MOVE CHECKED% TO MENU_CENTRE
  ENDWHERE
  WHERE DOC_JUS_FULL%
    MOVE CHECKED% TO MENU_JUSTIFIE
  ENDWHERE
ENDEVALUATE
```

Voir aussi Constantes DOC_JUS_*, DOC_SETJUSTIF, DOC_GETPARAGRAPHINFO

Fonction **DOC_GETMAXPARAWIDTH%**

Retourne le nombre maximum de caractères d'une ligne de texte lorsque la fenêtre est gérée en mode "caractère".

Syntaxe	DOC_GETMAXPARAWIDTH% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Nombre maximum de caractères d'une ligne, 0 s'il n'y a pas de limite.		

Exemple :

```
MESSAGE "Longueur de la ligne", DOC_GETMAXPARAWIDTH% (HWP%)
```

Voir aussi Constantes **DOC LO*%, DOC CHARS%, DOC DEFAULTMAXPARAWIDTH, DOC SETMAXPARAWIDTH, DOC DEFAULTMETRICS, DOC METRICS, DOC ISMETRICS%**

Fonction DOC_GETMODE%

Retourne le mode d'affichage du document (standard, brouillon ou page)..

Syntaxe	DOC_GETMODE% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes DOC_MODE_ *%.		

Exemple :

```
; Cochage des menus selon le mode d'affichage sur événement INIT du menu
déroulant correspondant
MOVE UNCHECKED% TO MENU_STANDARD
MOVE UNCHECKED% TO MENU_BROUILLON
MOVE UNCHECKED% TO MENU_PAGE
  EVALUATE DOC_GETMODE% (HWP%)
    WHERE DOC_MODE_DEFAULT%
      MOVE CHECKED% TO MENU_STANDARD
    ENDWHERE
    WHERE DOC_MODE_DRAFT%
      MOVE CHECKED% TO MENU_BROUILLON
    ENDWHERE
    WHERE DOC_MODE_PAGE%
      MOVE CHECKED% TO MENU_PAGE
    ENDWHERE
  ENDEVALUATE
```

Voir aussi Constantes DOC_MODE_ *%, DOC_SETMODE

Fonction DOC_GETNBPAGES%

Retourne le nombre de pages du document.

Syntaxe	DOC_GETNBPAGES% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Nombre de pages.		

Les sauts de page implicites ne sont comptabilisés que si les caractéristiques de mise en page ont été récupérées au moyen de l'instruction DOC_ASSOCIATEPRINTER.

Exemple :

```
MESSAGE "Nombre de pages", DOC_GETNBPAGES%(HWP%)
```

Voir aussi [DOC_ASSOCIATEPRINTER](#), [DOC_PAGEBREAK](#)

Fonction DOC_GETNBSECTIONS%

Retourne le nombre de sections du document.

Syntaxe	DOC_GETNBSECTIONS% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Nombre de sections.		

Exemple :

```
MESSAGE "Nombre de sections", DOC_GETNBSECTIONS% (HWP%)
```

Voir aussi DOC_SECTION, DOC_ISSECTION%, DOC_GETSECTION%, DOC_SETSECTION, DOC_GETSECTIONJUSTIF%, DOC_SETSECTIONJUSTIF

Fonction DOC_GETPAGE%

Retourne le numéro de la page courante.

Syntaxe	DOC_GETPAGE% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Numéro de page.		

Exemple :

```
; Positionnement au début de la page courante  
DOC_SETPAGE HWP%, DOC_GETPAGE% (HWP%)
```

Voir aussi [DOC_GETNBPAGES%](#)

Fonction DOC_GETPARAGRAPHGLOBALUSERINFO%

Retourne l'entier utilisateur de type global associé au paragraphe.

Syntaxe	DOC_GETPARAGRAPHGLOBALUSERINFO% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Entier associé au paragraphe.		

Exemple :

```
MESSAGE "Entier associé", DOC_GETPARAGRAPHGLOBALUSERINFO%
```

Voir aussi DOC_SETPARAGRAPHGLOBALUSERINFO, DOC_GETPARAGRAPHLOCALUSERINFO%,
DOC_SETPARAGRAPHLOCALUSERINFO

Fonction **DOC_GETPARAGRAPHKEEPTOGETHER%**

Indique si les lignes du paragraphe courant sont solidaires.

Syntaxe	DOC_GETPARAGRAPHKEEPTOGETHER% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si les lignes sont solidaires, FALSE% sinon.		

Exemple :

```
; Lie les lignes du paragraphe courant si elles ne sont pas déjà solidaires
IF DOC_GETPARAGRAPHKEEPTOGETHER% (HWP%) = FALSE%
DOC_SETPARAGRAPHKEEPTOGETHER HWP%, TRUE%
ENDIF
```

Voir aussi DOC_SETPARAGRAPHKEEPTOGETHER, DOC_GETPARAGRAPHKEEPWITHNEXT%,
DOC_SETPARAGRAPHKEEPWITHNEXT

Fonction **DOC_GETPARAGRAPHKEEPWITHNEXT%**

Indique si le paragraphe courant est solidaire du suivant.

Syntaxe	DOC_GETPARAGRAPHKEEPWITHNEXT% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le paragraphe est solidaire du suivant, FALSE% sinon.		

Exemple :

```
; Lie le paragraphe courant avec son suivant s'ils ne sont pas déjà  
solidaires  
IF DOC_GETPARAGRAPHKEEPWITHNEXT% (HWP%) = FALSE%  
DOC_SETPARAGRAPHKEEPWITHNEXT HWP%, TRUE%  
ENDIF
```

Voir aussi **DOC_SETPARAGRAPHKEEPWITHNEXT**, **DOC_GETPARAGRAPHKEEPTOGETHER%**,
DOC_SETPARAGRAPHKEEPTOGETHER

Fonction DOC_GETPARAGRAPHLOCALUSERINFO%

Retourne l'entier utilisateur de type local associé au paragraphe.

Syntaxe	DOC_GETPARAGRAPHLOCALUSERINFO% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Entier associé au paragraphe.		

Exemple :

```
MESSAGE "Entier associé", DOC_GETPARAGRAPHLOCALUSERINFO%
```

Voir aussi DOC_SETPARAGRAPHLOCALUSERINFO, DOC_GETPARAGRAPHGLOBALUSERINFO%,
DOC_SETPARAGRAPHGLOBALUSERINFO

Fonction **DOC_GETPARALINEOFFSET%**

Retourne le nombre de caractères précédant la ligne dans le paragraphe.

La ligne est spécifiée dans handle-parag et num-ligne. La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETPARALINEOFFSET% (<i>handle-WP, handle-parag, num-ligne</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle d'un paragraphe contenant la ligne
	num-ligne	INT(2)	numéro de la ligne dans le paragraphe
Valeur retournée	INT(2). Nombre de caractères précédant la ligne.		

Si handle-parag vaut 0, la ligne prise en compte est la ligne contenant le curseur.

Exemple :

```
MESSAGE "Position de la ligne", DOC_GETPARALINEOFFSET% (HWP%, 0, 0)
```

Voir aussi [DOC_GETLINE](#), [DOC_GETLINEINFO](#), [DOC_GETNEXTPARALINE](#), [DOC_GETPREVPARALINE](#)

Fonction DOC_GETPICTDFNID%

Donne l'identificateur associé à l'image bitmap stockée dans le fichier de nom nom-bitmap.

L'image bitmap doit être chargée en mémoire. Son identificateur est copié dans id-bitmap.

Syntaxe	DOC_GETPICTDFNID% (<i>handle-WP, nom-bitmap, id-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-bitmap	CSTRING	!V! chemin et nom du fichier contenant l'image bitmap
	id-bitmap	INT(2)	!V! identificateur attribué à l'image bitmap
Valeur retournée	INT(2) TRUE% : l'identificateur est trouvé. FALSE% : l'identificateur n'est pas trouvé.		

A la suite de DOC_GETPICTDFNID%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOTAPICTNAME%.

Exemple :

```

LOCAL CSTRING NAME$
LOCAL ID%(2)

MOVE "C:\NSDK\BIN\RABBIT.BMP" TO NAME$
IF DOC_GETPICTDFNID% (HWP%, NAME$, ID%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Identificateur non trouvé"
ELSE
MESSAGE "Image bitmap" && NAME$, "Son identificateur est" && ID%
ENDIF

```

Voir aussi Constantes [DOC_ERR*%](#), [DOC_ERROR%](#), [DOC_ADDPICTDFN%](#), [DOC_DELPICDFN%](#), [DOC_GETPICTDFNNAME%](#)

Fonction **DOC_GETPICTDFNNAME%**

Copie dans nom-bitmap le chemin et le nom du fichier bitmap comprenant l'image bitmap chargée en mémoire d'identificateur id-bitmap.

Syntaxe	DOC_GETPICTDFNNAME% (<i>handle-WP, id-bitmap, nom-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	!V! identificateur attribué à l'image bitmap
	nom-bitmap	CSTRING	!V! chemin et nom du fichier contenant l'image bitmap
Valeur retournée	INT(2) TRUE% lorsque le nom est trouvé, FALSE% sinon.		

A la suite de **DOC_GETPICTDFNNAME%**, **DOC_ERROR%** peut retourner le code d'erreur **DOC_ERRNOTAPICTID%**.

Exemple :

```
LOCAL CSTRING NAME$
LOCAL ID%(2)

MOVE "C:\NSDK\BIN\RABBIT.BMP" TO NAME$
IF DOC_GETPICTDFNNAME% (HWP%, ID%, NAME$) = FALSE%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Nom non trouvé"
ELSE
MESSAGE "Image bitmap d'identificateur" && ID%, "Son nom est" && NAME$
ENDIF
```

Voir aussi Constantes **DOC_ERR*%**, **DOC_ERROR%**, **DOC_ADDPICTDFN%**, **DOC_DELPICDFN%**, **DOC_GETPICTDFNID%**, **DOC_GETPICTSIZE%**

Fonction DOC_GETPICTSIZE%

Copie dans larg-bitmap et haut-bitmap la hauteur et la largeur de l'image bitmap chargée en mémoire d'identificateur id-bitmap.

larg-bitmap et haut-bitmap sont exprimées dans l'unité courante (système métrique ou anglo-saxon).

Syntaxe	DOC_GETPICTSIZE% (<i>handle-WP, id-bitmap, larg-bitmap, haut-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	identificateur attribué à l'image bitmap
	larg-bitmap	INT(2)	!V! largeur de l'image bitmap
	haut-bitmap	INT(2)	!V! hauteur de l'image bitmap
Valeur retournée	INT(2) TRUE% lorsque la copie est réussie, FALSE% sinon.		

A la suite de DOC_GETPICTSIZE%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOTAPICTID%.

Exemple :

```

LOCAL ID%(2), WIDTH%(2), HEIGHT%(2)

IF DOC_GETPICTDSIZE (HWP%, ID%, WIDTH%, HEIGHT%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Dimensions non trouvées"
ELSE
MESSAGE "Image bitmap d'identificateur" && ID%, "Largeur" && WIDTH% &&
"Hauteur" && HEIGHT%
ENDIF

```

Voir aussi Constantes [DOC_ERR*%](#), [DOC_ERROR%](#), [DOC_ADDPICTDFN%](#), [DOC_DELPICDFN%](#), [DOC_GETPICTDFNID%](#), [DOC_GETPICTDFNNAME%](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#)

Fonction DOC_GETSECTION%

Retourne le numéro de la section courante.

Syntaxe	DOC_GETSECTION% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(2) Numéro de la section.		

Les sections sont numérotées à partir de 1.

Exemple :

```
; Positionnement au début de la section courante  
DOC_SETSECTION HWP%, DOC_GETSECTION% (HWP%)
```

Voir aussi DOC_SECTION, DOC_ISSECTION%, DOC_GETNBSECTIONS%, DOC_SETSECTION, DOC_GETSECTIONJUSTIF%, DOC_SETSECTIONJUSTIF

Fonction DOC_GETSECTIONJUSTIF%

Retourne le type d'alignement vertical appliqué au texte de la section courante.

Syntaxe	DOC_GETSECTIONJUSTIF% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	Type INT(1). Une des constantes DOC_JUS_UPPER%, DOC_JUS_LOWER%, DOC_JUS_CENTERED%, DOC_JUS_FULL%.		

Exemple :

```

EVALUATE DOC_GETSECTIONJUSTIF% (HWP%)
WHERE DOC_JUS_UPPER%
MESSAGE "Alignement vertical", "Haut de page"
ENDWHERE
WHERE DOC_JUS_LOWER%
MESSAGE "Alignement vertical", "Bas de page"
ENDWHERE
WHERE DOC_JUS_CENTERED%
MESSAGE "Alignement vertical", "Centré"
ENDWHERE
WHERE DOC_JUS_FULL%
MESSAGE "Alignement vertical", "Justifié"
ENDWHERE
ENDEVALUATE

```

Voir aussi Constantes DOC JUS *%, DOC SECTION, DOC ISSECTION%, DOC_GETNBSECTIONS%, DOC_GETSECTION%, DOC_SETSECTION, DOC_SETSECTIONJUSTIF

Fonction DOC_GETSELECTIONTEXT\$

Retourne le texte sélectionné.

Syntaxe	DOC_GETSELECTIONTEXT\$ (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	Type CSTRING. Chaîne de caractères d'au maximum 256 caractères contenant le texte sélectionné, chaîne vide si aucune sélection n'est en cours.		

1. Aucun code de style de caractère (police ou attribut par exemple) ne figure dans la chaîne retournée.
2. Le texte retourné est tronqué s'il excède 256 caractères.

Exemple :

```
IF DOC_ISSELECTION%(HWP%)
MESSAGE "Texte sélectionné", DOC_GETSELECTIONTEXT$(HWP%)
ELSE
MESSAGE "Attention", "Aucune sélection"
ENDIF
```

Voir aussi DOC_GETSELECTION, DOC_SETSELECTION, DOC_ISSELECTION%, DOC_GETTEXT

Fonction DOC_GETTEXTSIZE%

Retourne la taille du paragraphe de handle handle-parag.

Cette taille est exprimée en nombre de caractères contenus dans le paragraphe. Les caractères de formatage sont comptés.

Syntaxe	DOC_GETTEXTSIZE% (<i>handle-WP</i> , <i>handle-parag</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
Valeur retournée	INT(2) Nombre de caractères contenus dans le paragraphe.		

Le nombre de caractères peut également être obtenu par DOC_GETPARAGRAPHINFO.

Exemple :

```
MESSAGE "Taille du paragraphe courant", DOC_GETTEXTSIZE%(HWP%,  
DOC_CURRENTPARAGRAPH%(HWP%))
```

Voir aussi [DOC_GETPARAGRAPHINFO](#), [DOC_GETTEXT](#)

Fonction DOC_GETUSERFORMAT%

Retourne l'entier associé à la police de caractères courante.

Syntaxe	DOC_GETUSERFORMAT% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(4) Entier associé à la police de caractères.		

Exemple :

```
MESSAGE "Entier associé", DOC_GETUSERFORMAT% (HWP%)
```

Voir *aussi* **DOC_GETUSERFORMATAT%**, **DOC_SETUSERFORMAT,**
DOC_GETPARAGRAPHGLOBALUSERINFO%, **DOC_SETPARAGRAPHGLOBALUSERINFO,**
DOC_GETPARAGRAPHLOCALUSERINFO%, **DOC_SETPARAGRAPHLOCALUSERINFO**

Fonction **DOC_GETUSERFORMATAT%**

Retourne l'entier associé à la police de caractères utilisée à la position handle-parag, pos-caract.

pos-caract est exprimée en nombre de caractères précédant la position.

Syntaxe	DOC_GETUSERFORMATAT% (<i>handle-WP, handle-parag, pos-caract</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle d'un paragraphe
	pos-caract	INT(2)	position d'un caractère dans le paragraphe
Valeur retournée	INT(1) Entier associé à la police de caractères.		

Exemple :

```
MESSAGE "Entier associé", DOC_GETUSERFORMATAT% (HWP%, HPARA%, OFFSET%)
```

Voir **aussi** **DOC_GETUSERFORMAT%**, **DOC_SETUSERFORMAT**,
DOC_GETPARAGRAPHGLOBALUSERINFO%, **DOC_SETPARAGRAPHGLOBALUSERINFO**,
DOC_GETPARAGRAPHLOCALUSERINFO%, **DOC_SETPARAGRAPHLOCALUSERINFO**

Fonction **DOC_GETWIDOWCONTROL%**

Indique si le contrôle des lignes veuves et orphelines est activé pour le document.

Syntaxe	DOC_GETWIDOWCONTROL% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le contrôle des lignes veuves et orphelines est activé, FALSE% sinon.		

Exemple :

```
IF DOC_GETWIDOWCONTROL% (HWP%) = FALSE%  
DOC_SETWIDOWCONTROL HWP%, TRUE%  
ENDIF
```

Voir aussi [DOC_DEFAULTWIDOWCONTROL](#), [DOC_SETWIDOWCONTROL](#)

Fonction DOC_GETWINDOW%

Indique quelle est la partie du document affichée.

La partie du document affichée peut être :

- L'en-tête de la première page.
- L'en-tête des pages suivantes du document.
- Le corps du document.
- Le pied de page de la première page.
- Le pied de page des pages suivantes du document.

Syntaxe	DOC_GETWINDOW% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes DOC_WIN_*.%		

Equivalent : Barre d'outils standard



Exemple :

```
; Cochage des menus selon la partie affichée sur événement INIT du menu
déroutant correspondant
MOVE UNCHECKED% TO MENU_ENTETE
MOVE UNCHECKED% TO MENU_CORPS
MOVE UNCHECKED% TO MENU_PIEDDEPAGE

EVALUATE DOC_GETWINDOW%(HWP%)
WHERE DOC_WIN_HEADER%
MOVE CHECKED% TO MENU_ENTETE
ENDWHERE
WHERE DOC_WIN_BODY%
MOVE CHECKED% TO MENU_CORPS
ENDWHERE
WHERE DOC_WIN_FOOTER%
MOVE CHECKED% TO MENU_PIEDDEPAGE
ENDWHERE
ENDEVALUATE
```

Voir aussi Constantes DOC_WIN_*.%, DOC_SETWINDOW, DOC_ISWINDOW%,
DOC_DISPOSEWINDOW

Fonction DOC_INSERTBITMAPCHAR%

Insère à gauche du curseur l'image bitmap d'identificateur id-bitmap, image bitmap précédemment chargée avec DOC_ADDPICTDFN%.

pos-bitmap représente le décalage de l'image bitmap par rapport à la base de la ligne : un nombre négatif situe l'image bitmap au dessus de la ligne, une valeur nulle sur la ligne précisément, un nombre positif au dessous de la ligne.

larg-bitmap, haut-bitmap et pos-bitmap sont exprimées en pixels.

Syntaxe	DOC_INSERTBITMAPCHAR% (<i>handle-WP</i> , <i>id-bitmap</i> , <i>larg-bitmap</i> , <i>haut-bitmap</i> , <i>pos-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	identificateur attribué à l'image bitmap
	larg-bitmap	INT(4)	largeur à donner à la bitmap
	haut-bitmap	INT(4)	hauteur à donner à l'image bitmap
	pos-bitmap	INT(4)	position de l'image bitmap sur la ligne
Valeur retournée	INT(2) TRUE% lorsque l'insertion est réussie, FALSE% sinon.		

1. A la suite de DOC_INSERTBITMAPCHAR%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRNOPICTID%.
2. Une image bitmap doit être insérée dans une ligne vide.

Exemple :

```
; Insertion de l'image bitmap d'identificateur ID% de taille 64*64 pixels
IF DOC_INSERTBITMAPCHAR% (HWP%, ID%, 64, 64, 0) = FALSE%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Problème lors de l'insertion"
ENDIF
```

Voir aussi Constantes [DOC_ERR*%](#), [DOC_ERROR%](#), [DOC_ADDPICTDFN%](#), [DOC_GETBITMAPCHAR%](#), [DOC_MODIFYBITMAPCHAR%](#), [DOC_DELETEBITMAPCHAR%](#)

Fonction DOC_ISCLIPBOARD%

Indique l'état du presse-papiers.

Syntaxe	DOC_ISCLIPBOARD% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le presse-papiers n'est pas vide, FALSE% sinon.		

Exemple :

```
; Menu Paste grisé si presse-papiers vide sur événement INIT du menu
déroulant correspondant
IF DOC_ISCLIPBOARD%(HWP%)
ENABLE MENU_PASTE
ELSE
DISABLE MENU_PASTE
ENDIF
```

Voir aussi DOC CUT, DOC COPY, DOC PASTE

Fonction DOC_ISFONTLOCKED%

Indique si l'utilisation de polices autres que celle spécifiée par DOC_DEFAULTFONT est interdite.

Syntaxe	DOC_ISFONTLOCKED% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si une seule police est autorisée, FALSE% sinon.		

Exemple :

```
IF DOC_ISFONTLOCKED% (HWP%) = FALSE%  
DOC_LOCKFONTS HWP%  
ENDIF
```

Voir aussi DOC_DEFAULTLOCKFONTS, DOC_LOCKFONTS

Fonction DOC_ISLIKEPRINTER%

Indique si le mode "comme à l'imprimante" est activé.

Syntaxe	DOC_ISLIKEPRINTER% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le mode est activé, FALSE% sinon.		

Exemple :

```
IF NOT DOC_ISLIKEPRINTER%  
DOC_SETLIKEPRINTER TRUE%  
ENDIF
```

Voir aussi [DOC_SETLIKEPRINTER](#)

Fonction DOC_ISMETRICS%

Retourne le système de mesure utilisé (métrique, anglo-saxon ou caractère).

Syntaxe	DOC_ISMETRICS% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) Une des constantes DOC_LO*%, DOC_CHARS%.		

Graduations de Règle différentes selon le système

Exemple :

```
IF DOC_ISMETRICS%(HWP%) = DOC_LOENGLISH%  
MESSAGE "Système utilisé", "Anglo-saxon"  
ELSEIF DOC_ISMETRICS%(HWP%) = DOC_LOMETRIC%  
MESSAGE "Système utilisé", "Métrique"  
ELSEIF DOC_ISMETRICS%(HWP%) = DOC_CHARS%  
MESSAGE "Système utilisé", "Caractère"  
ENDIF
```

Voir aussi Constantes DOC_LO*%, DOC_CHARS%, DOC_METRICS, DOC_DEFAULTMETRICS

Fonction **DOC_ISPAGEBREAK%**

Indique la présence d'un saut de page au début du paragraphe courant.

Syntaxe	DOC_ISPAGEBREAK% (handle-WP)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le paragraphe débute par un saut de page, FALSE% sinon.		

Exemple :

```
; Positionnement d'un saut de page au début du paragraphe courant
IF NOT DOC_ISPAGEBREAK%(HWP%)
DOC_PAGEBREAK HWP%
ENDIF
```

Voir aussi [DOC_PAGEBREAK](#), [DOC_GETPOS](#)

Fonction DOC_ISREADONLY%

Indique le mode d'accessibilité du document (lecture/écriture ou lecture seulement).

Syntaxe	DOC_ISREADONLY% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le document est en mode "lecture seulement". FALSE% si le document est en mode "lecture/écriture".		

DOC_ISREADONLY% est rigoureusement équivalent à ISLOCKED%.

Exemple :

```
; Inversion de l'état de blocage, sur événement EXECUTED de l'élément de
menu correspondant
IF DOC_ISREADONLY%(HWP%)
DOC_READONLY HWP%, FALSE%
MOVE UNCHECKED% TO MENU_READONLY
ELSE
DOC_READONLY HWP%, TRUE%
MOVE CHECKED% TO MENU_READONLY
ENDIF
```

Voir aussi Constantes DOC_FMT *%, DOC_READONLY, ISLOCKED% (*Manuel de Programmation NCL*)

Fonction DOC_ISREDO%

Indique s'il est possible de rejouer la dernière action annulée par un DOC_UNDO.

Syntaxe	DOC_ISREDO% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si l'annulation de l'annulation est possible, FALSE% sinon.		

Si l'annulation de l'annulation est possible, celle-ci se fera avec l'instruction DOC_REDO.

Exemple :

```
IF DOC_ISREDO% (HWP%)  
  DOC_REDO HWP%  
ENDIF
```

Voir aussi DOC_REDO, DOC_UNDO

Fonction **DOC_ISRULERBAR%**

Indique si la barre d'outils standard et/ou la règle sont affichées.

Syntaxe	DOC_ISRULERBAR% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si la barre d'outils standard est affichée ou si la règle est affichée, FALSE% sinon.		

Exemple :

```
; Inversion de l'état de la barre d'outils et de la règle  
DOC_RULERBAR HWP%, NOT DOC_ISRULERBAR% (HWP%)
```

Voir aussi **DOC_ISRULERBARSHOWN%**, **DOC_ISTOOLBOXSHOWN%**, **DOC_RULERBAR**,
DOC_SHOWRULERBAR, **DOC_SHOWTOOLBOX**

Fonction DOC_ISRULERBARSHOWN%

Indique si la règle est affichée.

Syntaxe	DOC_ISRULERBARSHOWN% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si la règle est affichée, FALSE% sinon.		

Exemple :

```
; Inversion de l'état de la règle
DOC_SHOWRULERBAR HWP%, NOT DOC_ISRULERBARSHOWN% (HWP%)
$$Line$$
```

Voir aussi DOC_ISRULERBAR%, DOC_SHOWRULERBAR, DOC_RULERBAR

Fonction **DOC_ISSECTION%**

Indique si un saut de section est présent au début du paragraphe courant.

Syntaxe	DOC_ISSECTION% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si le paragraphe débute une nouvelle section, FALSE% sinon.		

Exemple :

```
; Insertion d'une section
IF NOT DOC_ISSECTION% (HWP%)
DOC_SECTION HWP%
ENDIF
```

Voir aussi [DOC_SECTION](#), [DOC_GETNBSECTIONS%](#), [DOC_GETSECTION%](#), [DOC_SETSECTION](#), [DOC_GETSECTIONJUSTIF%](#), [DOC_SETSECTIONJUSTIF](#)

Fonction DOC_ISSELECTION%

Indique la présence d'une sélection.

Syntaxe	DOC_ISSELECTION% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si une sélection existe, FALSE% sinon.		

Exemple :

```
; Menus Cut et Copy grisés si aucune sélection sur événement INIT du menu
déroulant correspondant
IF DOC_ISSELECTION%(HWP%)
ENABLE MENU_CUT
ENABLE MENU_COPY
ELSE
DISABLE MENU_CUT
DISABLE MENU_COPY
ENDIF
```

Voir aussi [DOC_SETSELECTION](#), [DOC_GETSELECTION](#), [DOC_GETSELECTIONTEXT\\$](#),
[DOC_COPYSELECTION](#), [DOC_MOVESELECTION](#)

Fonction DOC_ISTAG%

Indique si une borne est présente à la position pos-tag du paragraphe handle-parag.
pos-tag est le nombre de caractères précédant la borne dans le paragraphe.

Syntaxe	DOC_ISTAG% (<i>handle-WP, handle-parag, pos-tag</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-tag	INT(2)	position de la borne dans le paragraphe
Valeur retournée	INT(1) TRUE% si une borne est présente, FALSE% sinon.		

Exemple :

```
; Suppression de la borne située à droite du curseur
DOC_GETPOS HWP%, HPARA%, OFFSET%
IF DOC_ISTAG% (HWP%, HPARA%, OFFSET%)
DOC_CLEAR_TAG HWP%, HPARA%, OFFSET%
ENDIF
```

Voir aussi DOC_CLEAR_TAG, DOC_SET_TAG

Fonction **DOC_ISTOOLBOXSHOWN%**

Indique si la barre d'outils standard est affichée.

Syntaxe	DOC_ISTOOLBOXSHOWN% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si la barre d'outils standard est affichée, FALSE% sinon.		

Exemple :

```
; Inversion de l'état de la barre d'outils standard  
DOC_SHOWTOOLBOX HWP%, NOT DOC_ISTOOLBOXSHOWN% (HWP%)
```

Voir aussi [DOC_ISRULERBAR%](#), [DOC_SHOWTOOLBOX](#), [DOC_RULERBAR](#)

Fonction **DOC_ISUNDO%**

Indique la possibilité d'annuler la dernière action effectuée.

Syntaxe	DOC_ISUNDO% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(1) TRUE% si l'annulation de l'action est possible, FALSE% sinon.		

Si l'annulation est possible, celle-ci se fera avec l'instruction **DOC_UNDO**.

Exemple :

```
IF DOC_ISUNDO% (HWP%)  
DOC_UNDO HWP%  
ENDIF
```

Voir aussi **DOC_UNDO**

Fonction DOC_ISWINDOW%

Indique si la partie du document spécifiée dans partie-document existe.
partie-document est définie par l'une des constantes DOC_WIN_ *%.

Syntaxe	DOC_ISWINDOW% (<i>handle-WP, partie-document</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	partie-document	INT(1)	partie du document (en-tête de page, corps, pied de page...)
Valeur retournée	INT(1) TRUE% si la partie spécifiée existe, FALSE% sinon.		

Exemple :

```
; Affichage du pied de page
IF (DOC_ISWINDOW% (HWP%, DOC_WIN_FOOTER%) = TRUE%) AND (DOC_GETWINDOW%
(HWP%) <> DOC_WIN_FOOTER%)
DOC_SETWINDOW HWP%, DOC_WIN_FOOTER%
ENDIF
```

Voir aussi Constantes DOC_WIN_ *%, DOC_DISPOSEWINDOW, DOC_GETWINDOW%,
DOC_SETWINDOW

Fonction **DOC_LASTPARAGRAPH%**

Retourne le handle du dernier paragraphe.

Syntaxe	DOC_LASTPARAGRAPH% (<i>handle-WP</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
Valeur retournée	INT(4) Handle du dernier paragraphe.		

DOC_LASTPARAGRAPH% retourne toujours un handle. En effet même une fenêtre vide comporte un paragraphe, sans caractères.

Exemple :

```
; Sélection du dernier paragraphe
DOC_SETSELECTION HWP%, DOC_PREVPARAGRAPH%(HWP%, DOC_LASTPARAGRAPH%(HWP%)),
0, DOC_LASTPARAGRAPH%(HWP%), 0
```

Voir aussi DOC_CURRENTPARAGRAPH, DOC_FIRSTPARAGRAPH, DOC_PREVPARAGRAPH%, DOC_NEXTPARAGRAPH%

Fonction DOC_M_CREATEMARKER%

Crée une nouvelle entrée dans un dictionnaire de données.

Syntaxe	DOC_M_CREATEMARKER% (<i>handle-dictionnaire</i>)		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
Valeur retournée	INT (4) Handle de l'entrée créée.		

DOC_M_CREATEMARKER% crée une entrée vide dans le dictionnaire. Elle insère un nouveau numéro, lui affecte un handle, mais aucune donnée ne lui est associée. L'insertion des données se fait avec l'instruction DOC_M_SETMARKERDATA.

Exemple :

```
; Création d'une entrée dans le dictionnaire de handle ACLASS%  
MOVE DOC_M_CREATEMARKER%(AClass%) TO CSTYLE_SHEET%
```

Voir aussi DOC_M_CREATEMARKERWITHHANDLE%, DOC_M_SETMARKERDATA

Fonction DOC_M_CREATEMARKERWITHHANDLE%

Crée dans le dictionnaire de données dont le handle est handle-dictionnaire la donnée de numéro numéro-donnée.

Syntaxe	DOC_M_CREATEMARKERWITHHANDLE% (<i>handle-dictionnaire, numéro-donnée</i>)		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
	numéro-donnée	INT(2)	numéro de la donnée à créer
Valeur retournée	INT (4) Handle de l'entrée créée.		

DOC_M_CREATEMARKERWITHHANDLE% crée une entrée vide dans le dictionnaire. Elle insère le numéro, lui affecte un handle, mais aucune donnée ne lui est associée. L'insertion des données se fait avec l'instruction DOC_M_SETMARKERDATA.

Exemple :

```
; Création de l'entrée n°1 dans le dictionnaire de handle ACLASS%  
MOVE DOC_M_CREATEMARKERWITHHANDLE%(AClass%, 1) TO CStyle_Sheet%
```

Voir aussi DOC M_CREATEMARKER%, DOC M_SETMARKERDATA

Fonction **DOC_M_FINDNEXTMARKER%**

Recherche dans le document le marqueur suivant.

La recherche s'effectue à partir de la position indiquée dans handle-parag et pos-marqueur vers la fin du document. Lorsque handle-parag est initialisé à 0, la recherche commence à la position du curseur.

Si un marqueur est trouvé, handle-parag, pos-marqueur et pos-fin-marqueur sont mis à jour avec les coordonnées du marqueur.

pos-marqueur et pos-fin-marqueur sont exprimées en nombre de caractères précédant la position.

Syntaxe	DOC_M_FINDNEXTMARKER% (<i>handle-WP, handle-parag, pos-marqueur, pos fin marqueur</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	!V! handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	!V! position du marqueur dans le paragraphe
	pos-fin-marqueur	INT(2)	!V! position de fin du marqueur dans le paragraphe (juste après le texte du marqueur)
Valeur retournée	INT(1) TRUE% si le marqueur existe, FALSE% sinon.		

DOC_M_FINDNEXTMARKER% ne modifie pas la position du curseur dans le document.

Exemple :

```
; Recherche des marqueurs situés après le curseur Il est important de
modifier la valeur de OFFSETF%, sinon DOC_M_FINDNEXTMARKER% trouvera
éternellement le même marqueur !
WHILE DOC_M_FINDNEXTMARKER%(HWP%, HPARA%, OFFSETF%, OFFSETL%)
MOVE DOC_M_GETMARKERHANDLE%(HWP%, HPARA%, OFFSETF%) TO HMARKER%
...
MOVE OFFSETL% TO OFFSETF%
ENDWHILE
```

Voir aussi **DOC_M_FINDPREVMARKER%**

Fonction DOC_M_FINDPREVMARKER%

Recherche dans le document le marqueur précédent.

La recherche s'effectue à partir de la position indiquée dans handle-parag et pos-marqueur vers le début du document. Lorsque handle-parag est initialisé à 0, la recherche commence à la position du curseur.

Si un marqueur est trouvé, handle-parag, pos-marqueur et pos-fin-marqueur sont mis à jour avec les coordonnées du marqueur.

pos-marqueur et pos-fin-marqueur sont exprimées en nombre de caractères précédant la position.

Syntaxe	DOC_M_FINDPREVMARKER% (<i>handle-WP, handle-parag, pos-marqueur, pos fin marqueur</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	!V! handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	!V! position du marqueur dans le paragraphe
	pos-fin-marqueur	INT(2)	!V! position de fin du marqueur dans le paragraphe (juste après le texte du marqueur)
Valeur retournée	INT(1) TRUE% si le marqueur existe, FALSE% sinon.		

DOC_M_FINDPREVMARKER% ne modifie pas la position du curseur dans le document.

Exemple :

```
; Recherche des marqueurs situés avant le curseur
WHILE DOC_M_PREVNEXTMARKER%(HWP%, HPARA%, OFFSETF%, OFFSETL%)
MOVE DOC_M_GETMARKERHANDLE%(HWP%, HPARA%, OFFSETF%) TO HMARKER%
...
ENDWHILE
```

Voir aussi DOC_M_FINDNEXTMARKER%

Fonction **DOC_M_GETCLASS%**

Retourne le handle du dictionnaire de données lié au document NS-WP et dont le nom est nom-dictionnaire.

Syntaxe	DOC_M_GETCLASS% (<i>handle-WP, nom-dictionnaire</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-dictionnaire	CSTRING	nom du dictionnaire
Valeur retournée	INT(4) Le handle du dictionnaire si nom-dictionnaire correspond à un des dictionnaires du document, 0 sinon.		

Si deux dictionnaires ont le même nom, le handle retourné est celui du premier créé.

Exemple :

```
; Obtention du handle du dictionnaire de nom CLASSNAME$  
MOVE DOC_M_GETCLASS% (HWP%, CLASSNAME$) TO ACLASS%  
IF ACLASS% = 0  
MESSAGE "Erreur", CLASSNAME$ && "n'est pas un nom de dictionnaire"  
ELSE  
MESSAGE "", "Le dictionnaire" && CLASSNAME$ && "a pour handle" && ACLASS%  
ENDIF
```

Voir aussi [DOC_M_GETCLASSNAME\\$](#), [DOC_M_REGISTERMARKERCLASS](#)

Fonction DOC_M_GETCLASSNAME\$

Retourne le nom du dictionnaire dont le handle est handle-dictionnaire.

Syntaxe	DOC_M_GETCLASSNAME\$ (<i>handle-dictionnaire</i>)		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
Valeur retournée	CSTRING Nom du dictionnaire.		

Exemple :

```
; Obtention du nom du dictionnaire de handle ACLASS%  
MOVE DOC_M_GETCLASSNAME$ (AClass%) TO CLASSNAME$  
MESSAGE "", "Le dictionnaire de handle" && ACLASS% && "a pour nom" &&  
CLASSNAME$
```

Voir aussi DOC M_GETCLASS%

Fonction DOC_M_GETMARKERCLASS%

Retourne le handle du dictionnaire de données lié au document NS-WP et comprenant les données dont le handle est handle-données.

Syntaxe	DOC_M_GETMARKERCLASS% (<i>handle-WP, handle-données</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-données	INT(4)	handle de l'entrée de données
Valeur retournée	INT(4) Le handle du dictionnaire si handle-données correspond à un des dictionnaires du document, 0 sinon.		

Exemple :

```
MOVE DOC_M_GETMARKERCLASS% (HWP%, CSTYLE_SHEET%) TO ACLASS%
MESSAGE "", "Les données de handle" && CSTYLE_SHEET% && "sont stockées dans
le dictionnaire de handle" && ACLASS%
```

Voir *aussi* [DOC M INSERTMARKER,](#) [DOC M CREATEMARKER%,](#)
[DOC M CREATEMARKERWITHHANDLE%,](#) [DOC M CREATEMARKERWITHHANDLE%](#)

Fonction **DOC_M_GETMARKERDATASIZE%**

Retourne la taille mémoire des données associées à l'entrée de handle handle-données du dictionnaire.

Syntaxe	DOC_M_GETMARKERDATASIZE% (<i>handle-dictionnaire, handle-données</i>)		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
	handle-données	INT(4)	handle de l'entrée de données
Valeur retournée	INT(2) Taille en octets des données.		

Exemple :

```
; Test sur la taille mémoire
LOCAL CSTRING DATA$

IF DOC_M_GETMARKERDATASIZE% (AClass%, HMarker%) = 0
MESSAGE "Erreur", "Cette entrée de données est vide"
ELSE
DOC_M_GETMARKERDATA AClass%, HMarker%, @DATA$, SIZEOF DATA$
ENDIF
```

Voir aussi [DOC_M_GETMARKERDATA](#), [DOC_M_SETMARKERDATA](#)

Fonction **DOC_M_GETMARKERHANDLE%**

Retourne le handle du marqueur se trouvant à la position désignée par handle-parag et pos-marqueur

pos-marqueur est le nombre de caractères précédant le marqueur dans le paragraphe.

Syntaxe	DOC_M_GETMARKERHANDLE% (<i>handle-WP</i> , <i>handle-parag</i> , <i>pos-marqueur</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	position du marqueur dans le paragraphe
Valeur retournée	INT(4) Handle du marqueur.		

Exemple :

```
; Recherche du handle du marqueur se trouvant à la position OFFSET% du
paragraphe de handle HPARA%
IF DOC_M_ISMARKER% (HWP%, HPARA%, OFFSET%) = FALSE%
MESSAGE "Erreur", "Cette position ne correspond pas à un marqueur"
ELSE
MOVE DOC_M_GETMARKERHANDLE% (HWP%, HPARA%, OFFSET%) TO HMARKER%
ENDIF
```

Voir aussi DOC_M_INSERTMARKER, DOC_M_ISMARKER%

Fonction DOC_M_GETTEXT\$

Retourne le texte du marqueur se trouvant à la position désignée par handle-parag et pos-marqueur.

pos-marqueur est le nombre de caractères précédant le marqueur dans le paragraphe.

Syntaxe	DOC_M_GETTEXT\$ (<i>handle-WP, handle-parag, pos-marqueur</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	position du marqueur dans le paragraphe
Valeur retournée	CSTRING. Texte du marqueur.		

Exemple :

```
; Recherche du texte du marqueur se trouvant à la position OFFSET% du
paragraphe de handle HPARA%
LOCAL CSTRING TEXT$

IF DOC_M_ISMARKER% (HWP%, HPARA%, OFFSET%) = FALSE%
MESSAGE "Erreur", "Cette position ne correspond pas à un marqueur"
ELSE
MOVE DOC_M_GETTEXT$ (HWP%, HPARA%, OFFSET%) TO TEXT$
ENDIF
```

Voir aussi DOC M INSERTMARKER, DOC M ISMARKER%, DOC M SETTEXT

Fonction **DOC_M_ISMARKER%**

Indique si la position spécifiée correspond à un marqueur.

pos-marqueur est le nombre de caractères précédant le marqueur dans le paragraphe.

Syntaxe	DOC_M_ISMARKER% (<i>handle-WP, handle-parag, pos-marqueur</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	position du marqueur dans le paragraphe
Valeur retournée	INT(1) TRUE% si la position correspond à un marqueur, FALSE% sinon.		

Exemple :

```
; Recherche du handle du marqueur se trouvant à la position OFFSET% du
paragraphe de handle HPARA%
IF DOC_M_ISMARKER% (HWP%, HPARA%, OFFSET%) = FALSE%
MESSAGE "Erreur", "Cette position ne correspond pas à un marqueur"
ELSE
MOVE DOC_M_GETMARKERHANDLE% (HWP%, HPARA%, OFFSET%) TO HMARKER%
ENDIF
```

Voir aussi [DOC M_INSERTMARKER](#)

Fonction DOC_M_NEWMARKERCLASS%

Crée le dictionnaire de données de nom nom-dictionnaire et de type type-dictionnaire, et retourne le handle de ce nouveau dictionnaire.

type-dictionnaire vaut l'une des constantes DOC_M_KIND_ *%.

Syntaxe	DOC_M_NEWMARKERCLASS% (<i>nom-dictionnaire</i> , <i>type-dictionnaire</i>)		
Paramètres	nom-dictionnaire	CSTRING	nom du dictionnaire
	type-dictionnaire	INT(4)	type du dictionnaire
Valeur retournée	INT(4) Handle du dictionnaire.		

Exemple :

```
; Création du dictionnaire de handle ACLASS%  
MOVE DOC_M_NEWMARKERCLASS% (CLASSNAME$, DOC_M_KIND_DEFAULT%) TO ACLASS%
```

Voir aussi Constantes DOC_M_KIND_ *%, DOC_M_DISPOSEMARKERCLASS

Fonction DOC_MODIFYBITMAPCHAR%

Modifie les caractéristiques de l'image bitmap située à gauche du curseur.

Toutes les caractéristiques de l'image bitmap peuvent être modifiées : largeur, hauteur, positionnement et même identificateur.

Attention cependant, un tel changement ne modifie pas l'identificateur de l'image chargée en mémoire, mais remplace l'image affichée par l'image correspondant au nouvel identificateur. Changer l'identificateur revient donc à afficher une nouvelle image bitmap, ou un cadre vide si le nombre spécifié n'est pas un identificateur d'image bitmap.

pos-bitmap représente le décalage de l'image bitmap par rapport à la base de la ligne : un nombre négatif situe l'image bitmap au dessus de la ligne, une valeur nulle sur la ligne précisément, un nombre positif au dessous de la ligne.

larg-bitmap, haut-bitmap et pos-bitmap sont exprimées en pixels.

Syntaxe	DOC_MODIFYBITMAPCHAR% (<i>handle-WP, id-bitmap, larg-bitmap, haut-bitmap, pos-bitmap</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	id-bitmap	INT(2)	identificateur attribué à l'image bitmap
	larg-bitmap	INT(4)	largeur à donner à l'image bitmap
	haut-bitmap	INT(4)	hauteur à donner à l'image bitmap
	pos-bitmap	INT(4)	positionnement de l'image bitmap sur la ligne
Valeur retournée	INT(2) TRUE% lorsque la modification est réussie, FALSE% sinon.		

A la suite de DOC_MODIFYBITMAPCHAR%, DOC_ERROR% peut retourner le code d'erreur DOC_ERRPARAGRAPHOVERFLOW%.

Exemple :

```
; Dimensionnement de l'image bitmap située à gauche du curseur à 32*32
pixels
IF DOC_GETBITMAPCHAR% (HWP%, ID%, WIDTH%, HEIGHT%, OFFSET%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR% (HWP%)
"La recherche a échoué"
ELSE
IF DOC_MODIFYBITMAPCHAR% (HWP%, ID%, 32, 32, OFFSET%) = FALSE%
MESSAGE "Erreur" && DOC_ERROR% (HWP%)
"La modification a échoué"
ENDIF
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_ADDPICTDFN%,
DOC_INSERTBITMAPCHAR%, DOC_GETBITMAPCHAR%, DOC_DELETEBITMAPCHAR%

Fonction DOC_NEXTPARAGRAPH%

Retourne le handle du paragraphe qui suit le paragraphe de handle handle-parag.

Syntaxe	DOC_NEXTPARAGRAPH% (handle-WP, handle-parag)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle de paragraphe
Valeur retournée	INT(4) Handle du paragraphe suivant, 0 si handle-parag est le dernier paragraphe.		

Si handle-parag vaut 0, le handle retourné est celui du paragraphe qui suit le paragraphe courant.

Exemple :

```
MESSAGE "Handle du deuxième paragraphe", DOC_NEXTPARAGRAPH%(HWP%,  
DOC_FIRSTPARAGRAPH%(HWP%))
```

Voir aussi DOC_CURRENTPARAGRAPH%, DOC_FIRSTPARAGRAPH%, DOC_LASTPARAGRAPH%, DOC_PREVPARAGRAPH%

Fonction **DOC_POS2XYVIEW%**

Donne les coordonnées écran du caractère défini dans *handle-parag* et *pos-caract*.
L'abscisse et l'ordonnée du caractère sont copiées dans *absc-caract* et *ord-caract*.
Elles sont exprimées en pixels, l'origine des axes se situant en bas à gauche.

Syntaxe	DOC_POS2XYVIEW% (<i>handle-WP</i> , <i>handle-parag</i> , <i>pos-caract</i> , <i>absc-caract</i> , <i>ord-caract</i>)		
Paramètres	<i>handle-WP</i>	INT(4)	handle de la fenêtre NS-WP
	<i>handle-parag</i>	INT(4)	handle du paragraphe
	<i>pos-caract</i>	INT(2)	position dans le paragraphe
	<i>absc-caract</i>	INT(4)	!V! abscisse correspondante
	<i>ord-caract</i>	INT(4)	!V! ordonnée correspondante
Valeur retournée	INT(2) Une des constantes DOC_POS_* %. Elle localise les coordonnées (à l'intérieur ou à l'extérieur de l'aire client de la fenêtre).		

Exemple :

```
; Coordonnées écran du curseur
DOC_GETPOS HWP%, HPARA%, OFFSET%
IF DOC_POS2XYVIEW% (HWP%, HPARA%, OFFSET%, X%, Y%) <>
DOC_POS_INSIDE%
MESSAGE "Erreur", "Curseur non positionné sur l'aire client"
ENDIF
```

Voir aussi Constantes DOC_POS_*, DOC_XYVIEW2POS%, DOC_GETPOS

Fonction DOC_PREVPARAGRAPH%

Retourne le handle du paragraphe qui précède le paragraphe de handle handle-parag.

Syntaxe	DOC_PREVPARAGRAPH% (<i>handle-WP, handle-parag</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
Valeur retournée	INT(4) Handle du paragraphe précédent, 0 si handle-parag est le premier paragraphe.		

Si handle-parag vaut 0, le handle retourné est celui du paragraphe qui précède le paragraphe courant.

Exemple :

```
MESSAGE "Handle de l'avant-dernier paragraphe", DOC_PREVPARAGRAPH%(HWP%,
DOC_LASTPARAGRAPH%(HWP%))
```

Voir aussi DOC_CURRENTPARAGRAPH%, DOC_FIRSTPARAGRAPH%, DOC_LASTPARAGRAPH%, DOC_NEXTPARAGRAPH%

Fonction DOC_PRINT%

Imprime une partie ou la totalité du document.

Cette impression ne porte que sur des pages entières. La partie à imprimer est donc bornée par deux numéros de pages.

Syntaxe	DOC_PRINT% (<i>handle-WP, num-page-deb, num-page-fin</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	num-page-deb	INT(2)	numéro de la première page à imprimer
	num-page-fin	INT(2)	numéro de la dernière page à imprimer
Valeur retournée	INT(2) Code d'erreur : une des constantes DOC_ERR*%.		

1. L'impression est envoyée sur l'imprimante associée à la fenêtre NS-WP ou sur l'imprimante attachée par défaut à la machine si aucune association n'a été effectuée.
2. Les pages sont numérotées à partir de 1.
3. Pour imprimer la totalité du document, employer la constante DEFRET% en dernier paramètre.
4. Le code d'erreur peut être aussi retourné par la fonction DOC_ERROR%.
5. La fonction DOC_EXTPRINT% permet également d'imprimer le document. Elle offre deux nouvelles fonctionnalités : la possibilité d'imprimer plusieurs fois de suite dans la même session et la possibilité de donner le numéro de son choix à la première page du document.

Exemple :

```
IF DOC_PRINT%(HWP%, 1, DEFRET%) = DOC_ERRNO%
MESSAGE "OK", "Document imprimé"
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "L'impression a échoué"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_PRINTFILE%, DOC_EXTPRINT%, DOC_ASSOCIATEPRINTER

Fonction DOC_PRINTFILE%

Imprime un fichier.

Cette impression ne porte que sur des pages entières. La partie à imprimer est donc bornée par deux numéros de pages.

Syntaxe	DOC_PRINTFILE% (<i>nom-fichier, num-page-deb, num-page-fin</i>)		
Paramètres	nom-fichier	CSTRING	chemin et nom du fichier à imprimer
	num-page-deb	INT(2)	numéro de la première page à imprimer
	num-page-fin	INT(2)	numéro de la dernière page à imprimer
Valeur retournée	INT(2) Code d'erreur : une des constantes DOC_ERR*%.		

1. L'impression est envoyée sur l'imprimante associée à la fenêtre NS-WP ou sur l'imprimante attachée par défaut à la machine si aucune association n'a été effectuée.
2. Il n'est pas nécessaire d'ouvrir préalablement le fichier. Il est même possible d'imprimer un document alors qu'un autre document est visualisé dans la fenêtre NS-WP.
3. Le format du fichier doit être celui d'un document NS-WP ou d'un document texte ASCII.
4. Les pages sont numérotées à partir de 1.
5. Pour imprimer la totalité du document, employer la constante DEFRET% en dernier paramètre.
6. Le code d'erreur peut être aussi retourné par la fonction DOC_ERROR%.
7. La fonction DOC_EXTPRINTFILE% permet également d'imprimer un fichier. Elle offre deux nouvelles fonctionnalités : la possibilité d'imprimer plusieurs fois de suite dans la même session et la possibilité de donner le numéro de son choix à la première page du document.

Exemple :

```
LOCAL ERROR%(2)

MOVE DOC_PRINTFILE%("FICHIER.DOC", 1, DEFRET%) TO ERROR%
IF ERROR% = DOC_ERRNO%
MESSAGE "OK", "La totalité du document a été imprimée"
ELSE
MESSAGE "Erreur" && ERROR%, "L'impression a échoué"
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_PRINT%, DOC_EXTPRINTFILE%, DOC_ASSOCIATEPRINTER

Fonction DOC_SCANFIRST%

Initialise, dans le contexte handle-contexte, la recherche des éléments du texte, et la démarre.

Initialisation :

- La recherche s'applique au paragraphe de handle handle-parag. Elle débute à la position pos-caract de ce paragraphe.
- Un buffer est déclaré. Il sert à stocker le résultat des recherches initialisées avec DOC_SCANFIRST%. Il est défini par son adresse adresse-buffer et sa taille maximale taille-max-buffer.

Démarrage :

- DOC_SCANFIRST% trouve le premier élément et le stocke dans le segment DOC_SCAN_CTXT et/ou dans le buffer déclaré.

Syntaxe	DOC_SCANFIRST% (<i>handle-contexte, handle-parag, pos-caract, adresse-buffer, taille-max-buffer</i>)		
Paramètres	handle-contexte	INT(4)	handle du contexte de recherche
	handle-parag	INT(4)	handle du paragraphe
	pos-caract	INT(2)	position dans le paragraphe
	adresse-buffer	INT(4)	adresse du buffer
	taille-max-buffer	INT(2)	taille maximum du buffer
Valeur retournée	INT(2) Une des constantes DOC_SCN_*. Elle indique le type de l'élément trouvé.		

1. La recherche est liée à un contexte de recherche. Il doit obligatoirement avoir été alloué précédemment.
2. Plusieurs recherches dans des contextes différents peuvent être menées parallèlement. De la sorte, il est possible de reprendre une recherche interrompue par une autre à l'endroit précis où elle avait été stoppée.

Exemple :

```
; Recherche des marqueurs placés dans un document
; Le buffer, dans ce cas précis, n'est pas utilisé
LOCAL CSTRING BUFF$

MOVE DOC_FIRSTPARAGRAPH% (HWP%) TO HPARA%
MOVE DOC_ALLOCSCANCONTEXT% TO HSC%

; Tant que ce n'est pas la fin du texte...
WHILE (HPARA% <> 0) AND (TYPE% <> DOC_SCN_ERR%)
MOVE DOC_SCANFIRST%(HSC%, HPARA%, 0, @BUFF$, 200) TO TYPE%
```

```
; Traitement d'un paragraphe
WHILE (TYPE% <> DOC_SCN_ENDPARA%) AND (TYPE% <> DOC_SCN_ERR%)
IF TYPE% = DOC_SCN_BEGMARK%
MESSAGE "Numéro dictionnaire", DOC_SCAN_CTXT(HSC%).sPARAM1
MESSAGE "Numéro entrée de données", DOC_SCAN_CTXT(HSC%).sPARAM2
ENDIF
MOVE DOC_SCANNEXT%(HSC%) TO TYPE%
ENDWHILE

; Fin de traitement du paragraphe
MOVE DOC_NEXTPARAGRAPH% (HWP%, HPARA%) TO HPARA%
ENDWHILE

IF DOC_FREESCANCONTEXT%(HSC%) = DOC_SCN_ERR%
MESSAGE "Erreur", "Libération de contexte non effectuée"
ENDIF
```

Voir aussi Constantes DOC SCN %, Segment DOC SCAN_CTXT, DOC ALLOCSCANCONTEXT%, DOC FREESCANCONTEXT%, DOC SCAN

Fonction DOC_SCANNEXT%

Recherche, dans le contexte de recherche handle-contexte, l'élément suivant du paragraphe (marque de fin de paragraphe, espace, marque de tabulation, borne...). L'élément trouvé est stocké dans le segment DOC_SCAN_CTXT et/ou dans le buffer déclaré à l'initialisation de la recherche.

Syntaxe	DOC_SCANNEXT% (<i>handle-contexte</i>)		
Paramètres	handle-contexte	INT(4)	handle du contexte de recherche
Valeur retournée	INT(2) Une des constantes DOC_SCN_*. Elle indique le type de l'élément trouvé.		

1. La recherche doit avoir été initialisée avec DOC_SCANFIRST%. Elle se limite au paragraphe indiqué.
2. Pour parcourir le document entier, il est nécessaire de passer de paragraphe en paragraphe. Cela implique de réinitialiser la recherche par un DOC_SCANFIRST% à chaque changement de paragraphe.

Exemple :

```
; Recherche des marqueurs placés dans un document
; Le buffer, dans ce cas précis, n'est pas utilisé
LOCAL CSTRING BUFF$

MOVE DOC_FIRSTPARAGRAPH% (HWP%) TO HPARA%
MOVE DOC_ALLOCSCANCONTEXT% TO HSC%

; Tant que ce n'est pas la fin du texte...
WHILE (HPARA% <> 0) AND (TYPE% <> DOC_SCN_ERR%)
MOVE DOC_SCANFIRST%(HSC%, HPARA%, 0, @BUFF$, 200) TO TYPE%

; Traitement d'un paragraphe
WHILE (TYPE% <> DOC_SCN_ENDPARA%) AND (TYPE% <> DOC_SCN_ERR%)
IF TYPE% = DOC_SCN_BEGMARK%
MESSAGE "Numéro dictionnaire", DOC_SCAN_CTXT(HSC%).sPARAM1
MESSAGE "Numéro entrée de données", DOC_SCAN_CTXT(HSC%).sPARAM2
ENDIF
MOVE DOC_SCANNEXT%(HSC%) TO TYPE%
ENDWHILE

; Fin de traitement du paragraphe
MOVE DOC_NEXTPARAGRAPH% (HWP%, HPARA%) TO HPARA%
ENDWHILE

IF DOC_FREESCANCONTEXT%(HSC%) = DOC_SCN_ERR%
MESSAGE "Erreur", "Libération de contexte non effectuée"
ENDIF
```

Voir aussi Constantes DOC_SCN_*, DOC_ALLOCSCANCONTEXT%, DOC_FREESCANCONTEXT%, DOC_SCANFIRST%

Fonction DOC_UNITS2XPEL%

Convertit une abscisse exprimée dans l'unité courante (métrique, anglo-saxonne ou caractère) en pixels.

L'origine des axes est le coin inférieur gauche de l'écran.

Syntaxe	DOC_UNITS2XPEL% (<i>handle-WP</i> , <i>val-abscisse</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	val-abscisse	INT(4)	valeur d'une abscisse exprimée dans l'unité courante
Valeur retournée	INT(4) Abscisse exprimée en pixels.		

Exemple :

```
MESSAGE "Abscisse en pixels", DOC_UNITS2XPEL% (HWP%, X%)
```

Voir aussi DOC_XPEL2UNITS%, DOC_YPEL2UNITS%, DOC_UNITS2YPEL%

Fonction **DOC_UNITS2YPEL%**

Convertit une ordonnée exprimée dans l'unité courante (métrique, anglo-saxonne ou caractère) en pixels.

L'origine des axes est le coin inférieur gauche de l'écran.

Syntaxe	DOC_UNITS2YPEL% (<i>handle-WP, val-ordonnée</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	val-ordonnée	INT(4)	valeur d'une ordonnée exprimée dans l'unité courante
Valeur retournée	INT(4) Ordonnée exprimée en pixels.		

Exemple :

```
MESSAGE "Ordonnée en pixels", DOC_UNITS2YPEL% (HWP%, Y%)
```

Voir aussi [DOC_XPEL2UNITS%](#), [DOC_YPEL2UNITS%](#), [DOC_UNITS2XPEL%](#)

Fonction **DOC_XPEL2UNITS%**

Convertit une abscisse exprimée en pixels dans l'unité courante (métrique, anglo-saxonne ou caractère).

L'origine des axes se situe en bas à gauche.

Syntaxe	DOC_XPEL2UNITS% (<i>handle-WP</i> , <i>abscisse-car</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	abscisse-car	INT(4)	abscisse d'un caractère
Valeur retournée	INT(4) Abscisse exprimée dans l'unité courante.		

Exemple :

```
DOC_GETPOS HWP%, HPARA%, OFFSET%
IF DOC_POS2XYVIEW% (HWP%, HPARA%, OFFSET%, X%, Y%) =
DOC_POS_INSIDE%
MESSAGE "Abscisse dans l'unité courante du curseur", DOC_XPEL2UNITS% (HWP%,
X%)
...
ENDIF
```

Voir aussi [DOC_YPEL2UNITS%](#), [DOC_UNITS2YPEL%](#), [DOC_UNITS2XPEL%](#)

Fonction **DOC_XYVIEW2POS%**

Donne l'emplacement dans le texte des coordonnées absc-caract et ord-caract.
L'emplacement trouvé est celui qui coïncide le plus avec les coordonnées spécifiées.
L'abscisse et l'ordonnée sont exprimées en pixels, l'origine des axes se situant en bas à gauche.

Syntaxe	DOC_XYVIEW2POS% (<i>handle-WP</i> , <i>absc-caract</i> , <i>ord-caract</i> , <i>handle-parag</i> , <i>pos caract</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	absc-caract	INT(4)	abscisse d'un caractère
	ord-caract	INT(4)	ordonnée d'un caractère
	handle-parag	INT(4)	!V! handle du paragraphe correspondant
	pos-caract	INT(2)	!V! position dans le paragraphe
Valeur retournée	INT(2) Une des constantes DOC_POS_* %. Elle localise l'emplacement (à l'intérieur ou à l'extérieur de l'aire client de la fenêtre).		

Exemple :

```
IF DOC_XYVIEW2POS% (HWP%, X%, Y%, HPARA%, OFFSET%) <>
DOC_POS_INSIDE%
MESSAGE "Erreur", "Coordonnées n'appartenant pas à l'aire client"
ENDIF
```

Voir aussi Constantes **DOC_POS_***, **DOC_POS2XYVIEW%**

Fonction **DOC_YPEL2UNITS%**

Convertit une ordonnée exprimée en pixels dans l'unité courante (métrique, anglo-saxonne ou caractère).

L'origine des axes est en bas à gauche.

Syntaxe	DOC_YPEL2UNITS% (<i>handle-WP</i> , <i>ordonnée-car</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	ordonnée-car	INT(4)	ordonnée d'un caractère
Valeur retournée	INT(4) Ordonnée exprimée dans l'unité courante.		

Exemple :

```
DOC_GETPOS HWP%, HPARA%, OFFSET%
IF DOC_POS2XYVIEW% (HWP%, HPARA%, OFFSET%, X%, Y%) =
DOC_POS_INSIDE%
MESSAGE "Ordonnée dans l'unité courante du curseur",
DOC_YPEL2UNITS% (HWP%, Y%)
...
ENDIF
```

Voir aussi [DOC_XPEL2UNITS%](#), [DOC_UNITS2YPEL%](#), [DOC_UNITS2XPEL%](#)

Fonction ESCLOAD%

Charge un fichier précédemment sauvegardé dans un format texte.

Le format texte utilisé est décrit dans le segment format-texte. Le nom du fichier et son emplacement sur disque sont également indiqués dans format-texte.

Ce fichier remplace le fichier éventuellement déjà chargé dans la fenêtre NS-WP, sauf si :

- Le caractère '&' figure au début de la chaîne contenant le nom de fichier

Dans ce cas, il est concaténé au fichier éventuellement déjà chargé dans la fenêtre NS WP.

- Le caractère '@' figure au début de la chaîne contenant le nom de fichier

Dans ce cas, il s'insère dans le fichier éventuellement déjà chargé dans la fenêtre NS WP à la position du curseur.

Syntaxe	ESCLOAD% (<i>handle-WP, format-texte</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	format-texte	TXTESCTABLE	!V! description du format texte utilisé par la sauvegarde
Valeur retournée	INT(2) Une des constantes DOC_ERR*%.		

Pour être récupéré correctement, un fichier chargé par ESCLOAD% doit être lu en mode "caractère".

Exemple :

```
local txtesctable table

move "STARTD" to table.table [TESTARTDOC%]
move "STOPD" to table.table [TESTOPDOC%]
move "FF" to table.table [TEFORMFEED%]
move "LF" to table.table [TELINEFEED%]
move "FN" to table.table [TEFONTNORMAL%]
move "c:\nsdk\txt\name.txt" to table.filename
move 30 to table.nblines
move CHS_ANSI% to table.charset

if escLOAD%(hwp%, table) = doc_errno%
message "", "fichier chargé"
endif
```

Voir aussi Segment [TXTESCTABLE](#), [DOC_DEFAULTMETRICS](#), [DOC METRICS](#), [DOC ISMETRICS](#), [ESCPRINT](#)

Fonction ESCPRINT%

Sauvegarde un document dans un format texte spécifique.

Le format texte est totalement décrit dans le segment format-texte. Le nom de fichier donné au document et son emplacement sur disque sont également indiqués dans le segment format-texte.

Syntaxe	ESCPRINT% (<i>handle-WP, format-texte</i>)		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	format-texte	TXTESCTABLE	!V! description du format texte utilisé par la sauvegarde
Valeur retournée	INT(2) Une des constantes DOC_ERR*%.		

Ce type de sauvegarde n'est destiné qu'à des documents écrits en mode "caractère".

Exemple :

```
local txtesctable table

move "STARTD" to table.table [TESTARTDOC%]
move "STOPD" to table.table [TESTOPDOC%]
move "FF" to table.table [TEFORMFEED%]
move "LF" to table.table [TELINEFEED%]
move "FN" to table.table [TEFONTNORMAL%]
move "c:\nsdk\txt\name.txt" to table.filename
move 30 to table.nblines
move CHS_ANSI% to table.charset

if escprint%(hwp%, table) = doc_errno%
message "", "sauvegarde effectuée"
endif
```

Voir aussi Segment [TXTESCTABLE](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#), [ESCLOAD%](#)

Instructions

Instruction DOC_ASSOCIATEPRINTER

Associe à la fenêtre NS-WP l'imprimante définie par le handle handle-imprimante. Les polices et les caractéristiques de mise en page associées à l'imprimante sont récupérées.

Les noms et tailles des polices récupérées sont stockés dans des Combo box with Entry field cachées à usage interne.

Ils sont visualisés dans la barre d'outils standard. Ils peuvent être également obtenus avec les instructions DOC_GETFONTNAMECONTROL et DOC_GETFONTSIZECONTROL.

Les caractéristiques de mise en page sont exploitées lors des impressions pour calculer la zone de texte effectivement éditée dans une page. Certaines de ces caractéristiques peuvent être connues avec l'instruction DOC_GETPAGEDIM.

Ces caractéristiques de mise en page permettent de générer les sauts de page implicites.

Syntaxe	DOC_ASSOCIATEPRINTER <i>handle-WP, handle-imprimante</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-imprimante	INT(4)	handle de l'imprimante

1. Le handle de l'imprimante est obtenu par appel aux fonctions REP_OPEN% et REP_DEFAULT% de la librairie NSPRN.
2. Si handle-imprimante = 0, les polices récupérées sont les polices système. Aucune caractéristique de mise en page n'est récupérée.
3. S'il n'est pas effectué de DOC_ASSOCIATEPRINTER :
 - a) Aucune liste de polices n'est connue.
 - b) Le mode "page" n'est pas fondé.
 - c) Les documents imprimés ne correspondent pas à l'imprimante sur laquelle ils sortent.

Exemple :

```
GLOBAL HSESSION%, HPRINTER%
; Démarrage d'une session d'impression
MOVE REP_OPEN% TO HSESSION%

; Choix de l'imprimante par défaut
MOVE REP_DEFAULT% (HSESSION%) TO HPRINTER%

; Association de l'imprimante
DOC_ASSOCIATEPRINTER HWP%, HPRINTER%
```

Voir aussi Constantes DOC MODE %, DOC_EXTPRINTSTART%, DOC_EXTPRINTSTOP%,
DOC_EXTPRINT%, DOC_EXTPRINTFILE%, DOC_PRINT%, DOC_PRINTFILE%,

**DOC_GETFONTNAMECONTROL, DOC_GETPAGEDIM, Librairie NSPRN (Manuel de Programmation
Librairies)**

Instruction **DOC_CLEAR**TAG

Supprime la borne située à la position pos-tag dans le paragraphe handle-parag.
pos-tag est le nombre de caractères qui précèdent la borne dans le paragraphe.

Syntaxe	DOC_CLEAR TAG <i>handle-WP, handle-parag, pos-tag</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-tag	INT(2)	position de la borne dans le paragraphe

Exemple :

```
; Suppression de la borne située à droite du curseur
DOC_GETPOS HWP%, HPARA%, OFFSET%
IF DOC_ISTAG% (HWP%, HPARA%, OFFSET%)
DOC_CLEAR TAG HWP%, HPARA%, OFFSET%
ENDIF
```

Voir aussi DOC_SETTAG, DOC_ISTAG%

Instruction DOC_COPY

Copie le texte sélectionné dans le presse-papiers.

Syntaxe	DOC_COPY <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

1. Combinée avec DOC_PASTE, cette instruction permet de dupliquer un texte dans un document, de le copier dans une autre fenêtre NS-WP ou dans une autre application de l'environnement graphique (Microsoft Word sous Windows par exemple).
2. Le style des caractères de la sélection (polices, attributs) est également copié.
3. Equivalent Clavier [Ctrl]+[Ins]

Exemple :

```
; Copie du texte sélectionné au début du dernier paragraphe
IF DOC_ISSELECTION%(HWP%)
DOC_COPY HWP%
DOC_SETPOS HWP%, DOC_LASTPARAGRAPH%(HWP%), 0
DOC_PASTE HWP%
ENDIF
```

Voir aussi DOC_CUT, DOC_PASTE, DOC_DELETE, DOC_UNDO, DOC_ISCLIPBOARD%, DOC_ISSELECTION%

Instruction **DOC_COPYSELECTION**

Copie le texte sélectionné dans le document source à la position courante du document de destination.

Syntaxe	DOC_COPYSELECTION <i>handle-WP-src, handle-WP-dest</i>		
Paramètres	handle-WP-src	INT(4)	handle de la fenêtre NS-WP contenant le document source
	handle-WP-dest	INT(4)	handle de la fenêtre NS-WP contenant le document de destination

Cette instruction permet de copier du texte entre fenêtres NS-WP sans passer par le presse-papiers, d'où un gain en rapidité.

Exemple :

```
IF DOC_ISSELECTION% (HWP1%)  
DOC_COPYSELECTION HWP1%, HWP2%  
ENDIF
```

Voir aussi DOC_COPY, DOC_PASTE, DOC_MOVESELECTION, DOC_ISSELECTION%

Instruction DOC_CUT

Supprime le texte sélectionné et le copie dans le presse-papiers.

Syntaxe	DOC_CUT <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

1. Combinée avec DOC_PASTE, cette instruction permet de déplacer un texte dans un document, ou de le déplacer dans une autre fenêtre NS-WP ou dans une autre application de l'environnement graphique (Microsoft Word sous Windows par exemple).
2. Le style des caractères de la sélection (polices, attributs) est également copié.
3. Equivalent Clavier [Shift]+[Del]

Exemple :

```
IF DOC_ISSELECTION% (HWP%)
DOC_CUT HWP%
ENDIF
```

Voir aussi DOC_COPY, DOC_PASTE, DOC_DELETE, DOC_UNDO, DOC_ISCLIPBOARD%, DOC_ISSELECTION%, DOC_DELETECHARS, DOC_NEW

Instruction **DOC_DEFAULTFONT**

Décrit le style de caractères qui sera utilisé par défaut dans les nouveaux documents.

Le style d'index *index-style* doit être entièrement décrit dans le segment *descript-style*.

La taille mémoire occupée par *descript-style* doit également être précisée. Elle peut se calculer avec l'instruction `NCL SIZEOF`.

Syntaxe	DOC_DEFAULTFONT <i>index-style, descript-style, taille-mémoire</i>		
Paramètres	<i>index-style</i>	INT(2)	index du style de caractères
	<i>descript-style</i>	DOC_FONT	!V! description du style de caractères
	<i>taille-mémoire</i>	INT(2)	taille en octets de <i>descript-style</i>

Si l'index indiqué se rapporte à un style déjà défini au moyen des instructions `DOC_SETFONTINDEX*`, il convient malgré tout de repréciser les caractéristiques du style dans *descript-style*.

Exemple :

```
; Choix du style d'index 1 comme style par défaut
LOCAL DOC_FONT FONTS
DOC_GETFONTINDEXCOLORS HWP%, 1, FONTS.COLORS
MOVE DOC_GETFONTINDEXFORMAT% (HWP%, 1) TO FONTS.FORMAT
MOVE DOC_GETFONTINDEXNAME$ (HWP%, 1) TO FONTS.NAME
MOVE DOC_GETFONTINDEXSIZE% (HWP%, 1) TO FONTS.SIZE
MOVE DOC_GETFONTINDEXUSERFORMAT% (HWP%, 1) TO FONTS.USERFORMAT%
DOC_DEFAULTFONT 1, FONTS, SIZEOF FONTS
```

Voir aussi Segment **DOC_FONT**, **DOC_DEFAULTLOCKFONTS**, **DOC_LOCKFONTS**,
DOC_GETFONTINDEXCOLORS, **DOC_GETFONTINDEXFORMAT%**, **DOC_GETFONTINDEXNAME\$**,
DOC_GETFONTINDEXSIZE%, **DOC_GETFONTINDEXUSERFORMAT%**, **DOC_SETFONTINDEXCOLORS**,
DOC_SETFONTINDEXFORMAT, **DOC_SETFONTINDEXNAME**, **DOC_SETFONTINDEXSIZE**,
DOC_SETFONTINDEXUSERFORMAT, **SIZEOF** (*Manuel de Programmation NCL*)

Instruction **DOC_DEFAULTLOCKFONTS**

Interdit dans les nouveaux documents l'utilisation de styles autres que celui spécifié par DOC_DEFAULTFONT.

Si indic-style vaut TRUE%, seul le style par défaut est pris en compte.

Si indic-style vaut FALSE%, d'autres styles peuvent être pris en compte.

Syntaxe	DOC_DEFAULTLOCKFONTS <i>indic-style</i>		
Paramètres	indic-style	INT(1)	indicateur interdisant l'utilisation de styles autres que le style défini par défaut

Exemple :

```
DOC_DEFAULTLOCKFONTS TRUE%
```

Voir aussi DOC_DEFAULTFONT, DOC_LOCKFONTS, DOC_ISFONTLOCKED%

Instruction **DOC_DEFAULTMAXPARAWIDTH**

Définit le nombre maximum de caractères d'une ligne en mode "caractère".

Si nb-caract = 0, il n'y a pas de limite.

Syntaxe	DOC_DEFAULTMAXPARAWIDTH <i>nb-caract</i>		
Paramètres	nb-caract	INT(2)	nombre maximum de caractères d'une ligne

Exemple :

```
DOC_DEFAULTMAXPARAWIDTH 80
```

Voir aussi Constantes DOC LO*%, DOC_CHARS%, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%, ESCLOAD%, ESCPRINT%

Instruction DOC_DEFAULTMETRICS

Spécifie le système de mesure qui sera utilisé par défaut dans les nouveaux documents.

syst-mesure vaut l'une des constantes DOC_LO*%, DOC_CHARS%.

Syntaxe	DOC_DEFAULTMETRICS syst-mesure		
Paramètres	syst-mesure	INT(2)	système de mesure par défaut

1. L'aspect de la règle graduée change selon le système sélectionné.
2. Il est possible de changer le système de mesure d'un document avec DOC_METRICS.
3. NS-WP fonctionne par défaut dans le système métrique.

Exemple :

```
; Système de mesure anglo-saxon
DOC_DEFAULTMETRICS DOC_LOENGLISH%
```

Voir aussi Constantes [DOC LO*%](#), [DOC_CHARS%](#), [DOC METRICS](#), [DOC ISMETRICS%](#), [DOC DEFAULTMAXPARAWIDTH](#), Segments [DOC BORDER](#), [DOC TABS](#), [DOC XTABS](#), Segments [LINE BORDER](#), [DOC TABLINE](#), [DOC DEFAULTPARAGRAPHINDENT](#), [DOC DEFAULTTAB](#), [DOC GETCURRENTTABLINE](#), [DOC GETDEFTAB%](#), [DOC GETMARGINS](#), [DOC GETPAGEDIM](#), [DOC SETCURRENTTABLINE](#), [DOC SETDEFTAB](#), [DOC SETMARGINS](#), [DOC GETPICTSIZE%](#)

Instruction **DOC_DEFAULTPARAGRAPHINDENT**

Définit les retraits de paragraphe qui seront utilisés par défaut dans les nouveaux documents.

Les valeurs des retraits sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_DEFAULTPARAGRAPHINDENT <i>retrait-prem, retrait-gauche, retrait-droit</i>		
Paramètres	retrait-prem	INT(2)	retrait gauche de la première ligne du paragraphe
	retrait-gauche	INT(2)	retrait gauche du paragraphe
	retrait-droit	INT(2)	retrait droit du paragraphe

Exemple :

```
DOC_DEFAULTPARAGRAPHINDENT 20, 0, 130
```

Voir aussi [DOC_GETTABS](#), [DOC_SETTABS](#), [DOC_GETXTABS](#), [DOC_SETXTABS](#),
[DOC_GETPARAGRAPHINFO](#), [DOC_SETPARAGRAPHSINFO](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#),
[DOC_ISMETRICS%](#)

Instruction **DOC_DEFAULTTAB**

Définit les taquets de tabulation qui seront positionnés dans les nouveaux documents.

Ces taquets de tabulation seront disposés selon un intervalle régulier égal à val-tab. val-tab est exprimée dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_DEFAULTTAB <i>val-tab</i>		
Paramètres	val-tab	INT(2)	intervalle entre deux taquets de tabulation

Exemple :

```
; Définit des taquets de tabulation tous les 5 mm (l'unité de mesure
courante est supposée être DOC_LOMETRICS% : 1/10 mm)
DOC_DEFAULTTAB 50
```

Voir aussi DOC_GETDEFTAB%, DOC_SETDEFTAB, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%

Instruction **DOC_DEFAULTWIDOWCONTROL**

Applique le contrôle des lignes veuves et orphelines aux nouveaux documents.

Si indic-vo vaut TRUE%, le contrôle des lignes veuves et orphelines est activé.

Si indic-vo vaut FALSE%, le contrôle des lignes veuves et orphelines est désactivé.

Syntaxe	DOC_DEFAULTWIDOWCONTROL <i>indic-vo</i>		
Paramètres	indic-vo	INT(1)	indicateur de contrôle des lignes veuves et orphelines

Exemple :

```
DOC_DEFAULTWIDOWCONTROL TRUE%
```

Voir aussi [DOC_GETWIDOWCONTROL%](#), [DOC_SETWIDOWCONTROL](#)

Instruction **DOC_DELETE**

Supprime le texte sélectionné.

Syntaxe	DOC_DELETE <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

Aucune sauvegarde de ce qui a été supprimé n'est effectuée dans le presse-papiers.
Equivalent Clavier [Del]

Exemple :

```
IF DOC_ISSELECTION% (HWP%)  
DOC_DELETE HWP%  
ENDIF
```

Voir aussi [DOC_COPY](#), [DOC_CUT](#), [DOC_PASTE](#), [DOC_UNDO](#), [DOC_DELETECHARS](#), [DOC_NEW](#)

Instruction DOC_DELETECHARS

Efface une partie du texte.

pos-deb et pos-fin sont des positions relatives aux paragraphes concernés. Elles sont exprimées en nombre de caractères, le premier caractère du paragraphe ayant la position 0, le deuxième la position 1, et ainsi de suite.

Syntaxe	DOC_DELETECHARS <i>handle-WP, handle-parag-deb, pos-deb, handle-parag-fin, pos fin</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag-deb	INT(4)	handle du paragraphe où débute le texte à supprimer
	pos-deb	INT(2)	position du caractère sur lequel débute le texte à supprimer
	handle-parag-fin	INT(4)	handle du paragraphe où finit le texte à supprimer
	pos-fin	INT(2)	position du caractère sur lequel finit le texte à supprimer

Equivalent Clavier : [Del] sur sélection

Exemple :

```
; Effacement de tout le texte sauf 1er et dernier paragraphes
DOC_DELETECHARS HWP%, DOC_NEXTPARAGRAPH%(HWP%, DOC_FIRSTPARAGRAPH%(HWP%)),
0, DOC_LASTPARAGRAPH%(HWP%), 0
```

Voir aussi DOC_INSERTCHARS, DOC_DELETE, DOC_CUT, DOC_NEW

Instruction **DOC_DISPOSEWINDOW**

Détruit la partie du document spécifiée dans partie-document.
partie-document est définie par l'une des constantes DOC_WIN_ *%.

Syntaxe	DOC_DISPOSEWINDOW <i>handle-WP, partie-document</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	partie-document	INT(1)	partie du document (en-tête de page, corps, pied de page...)

Exemple :

```
; Destruction du pied de page
IF DOC_ISWINDOW% (HWP%, DOC_WIN_FOOTER%) = TRUE%
DOC_DISPOSEWINDOW HWP%, DOC_WIN_FOOTER%
ENDIF
```

Voir aussi Constantes DOC_WIN_ *%, DOC_ISWINDOW%, DOC_GETWINDOW%, DOC_SETWINDOW

Instruction **DOC_EXTPRINTSTOP**

Clôt un job d'impression NS-WP.

Syntaxe	DOC_EXTPRINTSTOP <i>handle-job</i>		
Paramètres	handle-job	INT(4)	handle du job d'impression

Exemple :

```
LOCAL HJOB%
IF DOC_EXTPRINTSTART%(0, HJOB%, "Impression NS-WP") = DOC_ERRNO%
...
DOC_EXTPRINTSTOP HJOB%
ELSE
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Le démarrage du job a échoué"
ENDIF
```

Voir aussi [DOC_EXTPRINT%](#), [DOC_EXTPRINTFILE%](#), [DOC_EXTPRINTSTART%](#)

Instruction DOC_GETBORDER

Copie dans le segment descript-cadre les caractéristiques de l'encadrement associé au paragraphe courant.

Syntaxe	DOC_GETBORDER <i>handle-WP, descript-cadre, taille-cadre</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-cadre	DOC_BORDER	!V! description du cadre
	taille-mémoire	INT(2)	taille en octets de descript-cadre

La taille mémoire occupée par descript-cadre doit être précisée. Elle se calcule avec l'instruction NCL SIZEOF.

Exemple :

```
LOCAL DOC_BORDER BORDER
; Caractéristiques de l'encadrement du paragraphe courant
doc_Getborder hwp%, border, sizeof border
MESSAGE "Espacement entre le texte et le cadre", border.dwspace
MESSAGE "Motif utilisé en fond de cadre", border.PATTERN
MESSAGE "Couleur d'avant-plan du motif", border.FORECOLOR
```

Voir aussi Segment DOC_BORDER, DOC_SETBORDER

Instruction **DOC_GETCURRENTTAB**

Copie dans alignement et point-de-suite le type d'alignement et le style de points de suite des tabulations insérées par un double-clic souris sur la règle.

Le type d'alignement et le style de point de suite sont décrits respectivement par les constantes **DOC_TYPTAB_*** et **DOC_LEADTAB_***.

Syntaxe	DOC_GETCURRENTTAB <i>handle-WP, alignement, point-de-suite</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	alignement	INT(1)	!V! type d'alignement de la tabulation
	point-de-suite	INT(1)	!V! style de points de suite de la tabulation

DOC_GETCURRENTRULEROBJECT% permet de vérifier que le taquet inséré est un taquet de tabulation.

Exemple :

```
LOCAL TYPE%(1), LEAD%(1)
IF DOC_GETCURRENTRULEROBJECT% (HWP%) = TAB_RULER_OBJECT%
DOC_GETCURRENTTAB HWP%, TYPE%, LEAD%
ENDIF
```

Voir aussi Constantes **DOC_TYPTAB_***, **DOC_LEADTAB_***, **TAB_RULER_OBJECT%**, **DOC_GETCURRENTRULEROBJECT%**, **DOC_SETCURRENTTAB**, **DOC_GETCURRENTTABLINE**

Instruction **DOC_GETCURRENTTABLINE**

Copie dans largeur, type et couleur les caractéristiques des tabulations verticales insérées par un double-clic souris sur la règle.

La largeur du trait est exprimée dans l'unité de mesure courante (système métrique ou anglo-saxon). Le type du trait vaut 0 pour un trait simple, 1 pour un trait double. La couleur est décrite par les constantes COL_ *% du langage NCL.

Syntaxe	DOC_GETCURRENTTABLINE <i>handle-WP, largeur, type, couleur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	largeur	INT(1)	!V! largeur du trait vertical
	type	INT(2)	!V! type du trait vertical (simple ou double)
	couleur	INT(1)	!V! couleur du trait vertical

DOC_GETCURRENTRULEROBJECT% permet de vérifier que le taquet inséré est un taquet de tabulation verticale.

Exemple :

```
LOCAL WIDTH%(1), TYPE%(2), COLOR%(1)
IF DOC_GETCURRENTRULEROBJECT% (HWP%) = TABLINE_RULER_OBJECT%
DOC_GETCURRENTTABLINE HWP%, WIDTH%, TYPE%, COLOR%
ENDIF
```

Voir aussi Constante **TABLINE RULER OBJECT%**, DOC_GETCURRENTRULEROBJECT%, DOC_SETCURRENTTABLINE, DOC_GETCURRENTTAB, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%, **Constantes COL_ *% (Manuel de Programmation NCL)**

Instruction **DOC_GETFMTLINE**

Copie dans `descript-ligne` les caractéristiques d'une ligne de texte.

La ligne est spécifiée dans `handle-parag` et `num-ligne`. La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETFMTLINE <i>handle-WP, handle-parag, num-ligne, pos-ligne, descript ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	num-ligne	INT(2)	numéro de la ligne dans le paragraphe
	pos-ligne	INT(2)	nombre de caractères précédant la ligne dans le paragraphe
	descript-ligne	DOC_FMTLINE	!V! description de la ligne

1. Cette instruction ne doit être utilisée qu'en mode "caractère".
2. Si `handle-parag` vaut 0, la ligne prise en compte est la ligne contenant le curseur.

Exemple :

```
LOCAL DOC_FMTLINE FMTLINE
DOC_GETFMTLINE HWP%, 0, 0, FMTLINE
MESSAGE "Nb de caractères de la ligne", FMTLINE.COUNT
```

Voir aussi [Segment DOC_FMTLINE](#), [DOC_GETLINE](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#)

Instruction **DOC_GETFONTCOLORS**

Copie dans le segment couleurs les couleurs au point d'insertion.

Syntaxe	DOC_GETFONTCOLORS <i>handle-WP, couleurs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	couleurs	DOC_TEXTCOLORS	!V! couleurs du texte

Exemple :

```
; Affichage des couleurs
LOCAL DOC_TEXTCOLORS COLORS
DOC_GETFONTCOLORS HWP%, COLORS
EVALUATE COLORS.FORECOLOR
WHERE COL_BLACK%
MESSAGE "Couleur des caractères", "Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.BACKCOLOR
WHERE COL_BLACK%
MESSAGE "Couleur d'arrière-plan", "Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.LINECOLOR
WHERE COL_BLACK%
MESSAGE "Couleur du souligné et du barré", "Noir"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi Segment **DOC_TEXTCOLORS**, **DOC_SETFONTCOLORS**, **DOC_GETFONTBACKCOLOR%**,
DOC_SETFONTBACKCOLOR, **DOC_GETFONTFORECOLOR%**, **DOC_SETFONTFORECOLOR**,
DOC_GETFONTLINECOLOR%, **DOC_SETFONTLINECOLOR**

Instruction **DOC_GETFONTCOLORSAT**

Copie dans le segment couleurs les couleurs utilisées à la position indiquée dans handle-parag et pos-caract.

pos-caract est exprimée en nombre de caractères précédant la position.

Syntaxe	DOC_GETFONTCOLORSAT <i>handle-WP, handle-parag, pos-caract, couleurs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-caract	INT(2)	position dans le paragraphe
	couleurs	DOC_TEXTCOLORS	!V! couleurs du texte

Exemple :

```
; Affichage des couleurs du texte courant
LOCAL DOC_TEXTCOLORS COLORS
DOC_GETPOS% HWP%, HPARA%, OFFSET%
DOC_GETFONTCOLORSAT HWP%, HPARA%, OFFSET%, COLORS
EVALUATE COLORS.FORECOLOR
WHERE COL_BLACK%
MESSAGE "Couleur des caractères", "Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.BACKCOLOR
WHERE COL_Black%
MESSAGE "Couleur d'arrière-plan", "Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.LINECOLOR
WHERE COL_Black%
MESSAGE "Couleur du souligné et du barré", "Noir"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi Segment DOC_TEXTCOLORS, DOC_SETFONTCOLORS, DOC_GETFONTBACKCOLOR%, DOC_SETFONTBACKCOLOR, DOC_GETFONTFORECOLOR%, DOC_SETFONTFORECOLOR, DOC_GETFONTLINECOLOR%, DOC_SETFONTLINECOLOR

Instruction **DOC_GETFONTINDEXCOLORS**

Copie dans le segment couleurs-style les couleurs du style de caractères d'index index style.

Syntaxe	DOC_GETFONTINDEXCOLORS <i>handle-WP, index-style, couleurs-style</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	index d'un style de caractères
	couleurs-style	DOC_TEXTCOLORS	!V! couleurs d'un style de caractères

Exemple :

```
; Affichage des couleurs du style d'index 1
LOCAL DOC_TEXTCOLORS COLORS
DOC_GETFONTINDEXCOLORS HWP%, 1, COLORS
EVALUATE COLORS.FORECOLOR
WHERE COL_BLACK%
MESSAGE "Couleur des caractères","Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.BACKCOLOR
WHERE COL_BLACK%
MESSAGE "Couleur d'arrière-plan","Noir"
ENDWHERE
...
ENDEVALUATE

EVALUATE COLORS.LINECOLOR
WHERE COL_BLACK%
MESSAGE "Couleur du souligné et du barré","Noir"
ENDWHERE
...
ENDEVALUATE
```

Voir aussi [Segment DOC_TEXTCOLORS](#), [DOC_SETFONTINDEXCOLORS](#)

Instruction **DOC_GETFONTNAMECONTROL**

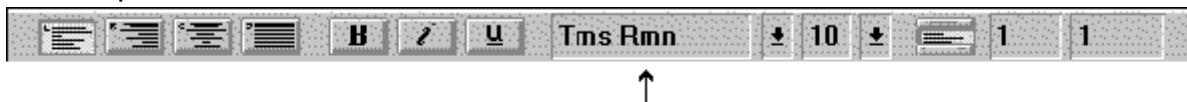
Associe une variable à la Combo box with Entry field (CBE) qui contient les noms des polices de caractères disponibles. Cette CBE a un usage interne et n'est jamais affichée. Elle est initialisée par **DOC_ASSOCIATEPRINTER**.

Cette variable permet de récupérer les noms des polices disponibles.

Syntaxe	DOC_GETFONTNAMECONTROL <i>handle-WP, combo-police</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	combo-police	CONTROL	!V! variable associée à la CBE contenant les noms des polices disponibles

1. La CBE contenant les noms des polices, affichée dans la barre d'outils standard, utilise également les informations de cette CBE.

2. Equivalent : Barre d'outils standard



Exemple :

```
; Remplissage de la List box LISTBOX avec les polices disponibles
LOCAL CONTROL CBE_FONTNAMES
LOCAL I%
DOC_GETFONTNAMECONTROL HWP%, CBE_FONTNAMES
MOVE LINECOUNT%(CBE_FONTNAMES) TO I%
MESSAGE "FontNames", "Il y a" && I% && "polices disponibles."
NOUPDATE LISTBOX
WHILE I%>0
MOVE I%-1 TO I%
INSERT AT 0 CBE_FONTNAMES[I%] TO LISTBOX
ENDWHILE
UPDATE LISTBOX
```

Voir aussi **DOC_GETFONTSIZECONTROL**, **DOC_GETFONT\$**, **DOC_SETFONT**, **DOC_ASSOCIATEPRINTER**

Instruction DOC_GETFONTSIZECONTROL

Associe une variable à la Combo box with Entry field (CBE) qui contient les tailles de caractères disponibles dans la police courante. Cette CBE a un usage interne et n'est jamais affichée. Elle est initialisée par DOC_ASSOCIATEPRINTER.

Cette variable permet de récupérer les tailles disponibles pour une police.

Syntaxe	DOC_GETFONTSIZECONTROL <i>handle-WP, combo-police</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	combo-police	CONTROL	!V! variable associée à la CBE contenant les tailles de caractères disponibles dans la police courante

1. La CBE contenant les tailles de caractère de la police courante, affichée dans la barre d'outils standard, utilise également les informations de cette CBE.
2. Equivalent : Barre d'outils standard



Exemple :

```
; Remplissage d'une list-box avec les tailles disponibles
LOCAL CONTROL CBE_FONTSTIZES
LOCAL I%
DOC_GETFONTSIZECONTROL HWP%, CBE_FONTSTIZES
MOVE LINECOUNT%(CBE_FONTSTIZES) TO I%
MESSAGE "FontSizes", "Il y a" && I% && "tailles disponibles."
NOUPDATE LISTBOX
WHILE I%>0
MOVE I%-1 TO I%
INSERT AT 0 CBE_FONTSTIZES[I%] TO LISTBOX
ENDWHILE
UPDATE LISTBOX
```

Voir aussi DOC_GETFONTNAMECONTROL, DOC_GETFONT\$, DOC_SETFONT, DOC_ASSOCIATEPRINTER

Instruction **DOC_GETLASTMERGEERROR**

Précise dans quel fichier a eu lieu la dernière erreur renvoyée par DOC_MERGE, et la décrit.

Dans la chaîne descripteur-erreur sont indiqués la ligne, l'opérateur, ... qui posent problème.

Syntaxe	DOC_GETLASTMERGEERROR <i>handle-WP, nom-fichier, descripteur-erreur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-fichier	CSTRING	chemin et nom du fichier où a eu lieu l'erreur
	descripteur-erreur	CSTRING	description de l'erreur

1. Si le code d'erreur retourné est **DOC_ERRCANNOTWRITEONFILE%**, descripteur-erreur contient la ligne qui n'a pas été écrite.
2. Si le code d'erreur retourné est **DOC_ERRSYNTAXERROR%**, descripteur-erreur indique quelle erreur de syntaxe a été détectée, ce qui peut être selon le cas :

```
Bad expression (...) : 'TO expected'
Bad expression (...) : 'ENDWHILE expected'
Bad expression (...) : 'WHILE expected'
Bad expression (...) : 'LOOP expected'
Bad expression (...) : 'REPEAT expected'
Bad expression (...) : 'WHILE, REPEAT or LOOP expected'
Bad expression (...) : 'ELSE, ENDIF or ELSEIF expected'
Bad expression (...) : 'IF or ELSE expected'
Bad expression (...) : 'IF expected'
Bad expression (...) : 'ENDIF expected'
Bad expression (...) : 'Label (...) expected'
Bad expression (...) : 'GOTO error. LABEL (...) expected'
Bad expression (...) : 'CALL error. LABEL (...) expected'
Bad expression (...) : 'LABEL (...) expected'
```

3. Si le code d'erreur retourné est **DOC_ERRTOOMANYINCLUDES%**, descripteur-erreur contient :

```
Maximum include = 8
```

4. Si le code d'erreur retourné est **DOC_ERRUNKNOWNVARIABLE%**, descripteur-erreur contient :

```
Unknown variable (...)
```

5. Si le code d'erreur retourné est **DOC_ERRINVALIDVARIABLE%**, descripteur-erreur contient :

```
Invalid variable (...)
```

Exemple :

```
LOCAL ERRCODE%(2), ERRLINE%(2)
LOCAL LASTNAME$, LASTERROR$
DOC_MERGE HWP%, ERRCODE%, ERRLINE%, "COMMANDS.DOC", "RESULT.DOC"
IF _ERRCODE% <> DOC_ERRNO%
DOC_GETLASTMERGEERROR HWP%, LASTNAME$, LASTERROR$
MESSAGE "Erreur" && ERRCODE% && LASTERROR$, "La fusion a échoué ligne" &&
ERRLINE% && "dans le fichier" && LASTNAME$
ENDIF
```

Voir aussi Constantes DOC_ERR*%, DOC_ERROR%, DOC_MERGE

Instruction DOC_GETLINE

Copie dans `descript-ligne` les caractéristiques d'une ligne de texte (décrites par les différents champs du segment `DOC_LINEDATA`).

La ligne est spécifiée dans `handle-parag` et `num-ligne`. La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETLINE <i>handle-WP, handle-parag, num-ligne, descript-ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	num-ligne	INT(2)	numéro de la ligne
	descript-ligne	DOC_LINEDATA	!V! description de la ligne

1. Cette instruction ne doit être utilisée qu'avec un système métrique ou anglo-saxon.
2. Si `handle-parag` vaut 0, la ligne prise en compte est la ligne contenant le curseur.

Exemple :

```
LOCAL DOC_LINEDATA LINEDATA
DOC_GETLINE HWP%, 0, 0, LINEDATA
MESSAGE "Nb de caractères de la ligne", LINEDATA.NB%
```

Voir aussi [Segment DOC_LINEDATA](#), [DOC_GETLINEINFO](#), [DOC_GETPARALINEOFFSET%](#), [DOC_GETNEXTPARALINE](#), [DOC_GETPREVPARALINE](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#)

Instruction DOC_GETLINEINFO

Copie dans pos-ligne et long-ligne la position et la longueur d'une ligne de texte. La ligne est spécifiée dans handle-parag et num-ligne. La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETLINEINFO <i>handle-WP, handle-parag, num-ligne, pos-ligne, long-ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	num-ligne	INT(2)	numéro de la ligne
	pos-ligne	INT(2)	!V! nombre de caractères précédant la ligne dans le paragraphe
	ong-ligne	INT(2)	!V! longueur de la ligne

1. DOC_GETLINEINFO met la valeur -1 dans pos-ligne et dans long-ligne lorsque num-ligne est supérieur au nombre de lignes du paragraphe désigné.
2. Si handle-parag vaut 0, la ligne prise en compte est la ligne contenant le curseur.
3. pos-ligne vaut toujours 0 lorsque la ligne est la première du paragraphe.

Exemple :

```
; Longueur de la ligne courante
DOC_GETLINEINFO HWP%, 0, 0, OFFSET%, LINELENGHT%
```

Voir aussi DOC_GETLINE, DOC_GETPARALINEOFFSET%, DOC_GETNEXTPARALINE, DOC_GETPREVPARALINE

Instruction **DOC_GETMARGINS**

Copie dans marge-gauche, marge-droite, marge-haute et marge-basse les valeurs des marges de la page courante.

Ces valeurs sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_GETMARGINS <i>handle-WP, marge-gauche, marge-droite, marge-haute, marge-basse</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	marge-gauche	INT(2)	!V! marge gauche de page
	marge-droite	INT(2)	!V! marge droite de page
	marge-haute	INT(2)	!V! marge haute de page
	marge-basse	INT(2)	!V! marge basse de page

Exemple :

```
LOCAL MARGEG%(2), MARGED%(2), MARGEH%(2), MARGEB%(2)

DOC_GETMARGINS HWP%, MARGEG%, MARGED%, MARGEH%, MARGEB%
MESSAGE "Marge gauche", MARGEG%
MESSAGE "Marge droite", MARGED%
MESSAGE "Marge haute", MARGEH%
MESSAGE "Marge basse", MARGEB%
```

Voir aussi [DOC_SETMARGINS](#), [DOC_GETPAGEDIM](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#)

Instruction **DOC_GETNEXTPARALINE**

Copie dans handle-parag et num-ligne la position de la ligne suivant la ligne courante dans le texte.

La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETNEXTPARALINE <i>handle-WP, handle-parag, num-ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	!V! handle du paragraphe contenant la ligne suivante
	num-ligne	INT(2)	!V! numéro de la ligne dans le paragraphe

Exemple :

```
LOCAL HPARA%, NUMBER%(2)
DOC_GETNEXTPARALINE HWP%, HPARA%, NUMBER%
```

Voir aussi [DOC_GETLINE](#), [DOC_GETLINEINFO](#), [DOC_GETPARALINEOFFSET](#),
[DOC_GETPREVPARALINE](#)

Instruction **DOC_GETPAGEDIM**

Copie dans largeur-page et hauteur-page la hauteur et la largeur de page imprimables.

Ces valeurs sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_GETPAGEDIM <i>handle-WP, largeur-page, hauteur-page</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	largeur-page	INT(2)	largeur de page du document
	hauteur-page	INT(2)	!V! hauteur de page du document

Les informations obtenues proviennent de l'imprimante associée à la fenêtre NS WP au moyen de l'instruction **DOC_ASSOCIATEPRINTER**.

Exemple :

```
LOCAL PAGEL%(2), PAGEH%(2)

DOC_GETPAGEDIM HWP%, PAGEL%, PAGEH%
MESSAGE "Largeur de page", PAGEL%
MESSAGE "Hauteur de page", PAGEH%
```

Voir aussi **DOC_GETMARGINS**, **DOC_SETMARGINS**, **DOC_DEFAULTMETRICS**, **DOC_METRICS**, **DOC_ISMETRICS%**, **DOC_ASSOCIATEPRINTER**

Instruction DOC_GETPARAGRAPHINFO

Récupère les caractéristiques du paragraphe de handle handle-parag.

La taille du paragraphe est exprimée en nombre de caractères. Les caractères de formatage sont comptés.

Le numéro de la première ligne du paragraphe est relatif au début du document, la première ligne du document étant numérotée 0.

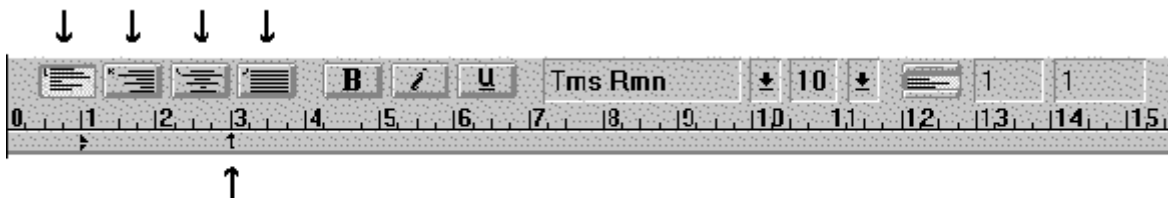
Le nombre de lignes constituant le paragraphe est de 1 au minimum.

Le type d'alignement du paragraphe vaut l'une des constantes DOC_JUS_*.%

Syntaxe	DOC_GETPARAGRAPHINFO <i>handle-WP, handle-parag, taille-parag, num ligne, nb-lignes, type-align, descript-tab</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	taille-parag	INT(2)	!V! taille du paragraphe
	num-ligne	INT(2)	!V! numéro de la première ligne du paragraphe
	nb-lignes	INT(2)	!V! nombre de lignes constituant le paragraphe
	type-align	INT(1)	!V! alignement du paragraphe
	descript-tab	DOC_TABS	!V! descriptif des tabulations du paragraphe

1. taille-parag peut également être obtenu par DOC_GETTEXTSIZE%, type-align par DOC_GETJUSTIF%, les informations de descript-tabs par DOC_GETTABS ou par DOC_GETXTABS.

2. Equivalent : Barre d'outils standard et règle (alignement et tabulations)



Exemple :

```
LOCAL SIZE%(2), FIRSTLINE%(2), NBLINES%(2), JUSTIF%(1)
LOCAL DOC_TABS MYTABS
```

```
DOC_GETPARAGRAPHINFO HWP%, DOC_CURRENTPARAGRAPH%(HWP%), SIZE%, FIRSTLINE%,
NBLINES%, JUSTIF%, MYTABS
```

```
MESSAGE "Le paragraphe courant contient", SIZE% && "caractères et" &&  
NBLINES% && "lignes."
```

***Voir aussi Constantes DOC JUS *%, Segment DOC TABS, DOC SETPARAGRAPHSINFO,
DOC GETTEXT, DOC SETTABS, DOC GETTABS, DOC GETJUSTIF%, DOC GETTEXTSIZE%,
DOC GETXTABS***

Instruction DOC_GETPOS

Donne la position du curseur.

Cette position est complètement identifiée par :

- Le handle du paragraphe où se situe le curseur.
- L'emplacement du curseur dans le paragraphe.

Syntaxe	DOC_GETPOS <i>handle-WP, handle-parag, pos-curseur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	!V! handle du paragraphe dans lequel est situé le curseur
	pos-curseur	INT(2)	!V! nombre de caractères précédant le curseur dans le paragraphe

1. DOC_POS2XYVIEW% permet de convertir cette position en coordonnées écran.

2. La position en ligne/colonne, telle qu'elle est affichée dans la barre d'outils, s'obtient grâce aux paramètres de l'événement SELECTED.

Exemple :

```
LOCAL HPARA%(4), OFFSET%(2), I%, P%

; Lecture de la position
DOC_GETPOS HWP%, HPARA%, OFFSET%

; Recherche du paragraphe de handle HPARA%
MOVE 1 TO I%
MOVE DOC_FIRSTPARAGRAPH%(HWP%) TO P%
WHILE P% <> HPARA%
  MOVE I%+1 TO I%
  MOVE DOC_NEXTPARAGRAPH%(HWP%, P%) TO P%
ENDWHILE
MESSAGE "Position du curseur", "Paragraphe numéro" && I% && "Caractère
numéro" && OFFSET%
```

Voir aussi DOC_SETPOS, DOC_CURRENTPARAGRAPH%, DOC_POS2XYVIEW%, Evénement SELECTED

Instruction **DOC_GETPREVPARALINE**

Copie dans handle-parag et num-ligne la position de la ligne précédant la ligne courante dans le texte.

La première ligne d'un paragraphe porte le numéro 0.

Syntaxe	DOC_GETPREVPARALINE <i>handle-WP, handle-parag, num-ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	!V! handle du paragraphe contenant la ligne précédente
	num-ligne	INT(2)	!V! numéro de la ligne dans le paragraphe

Exemple :

```
LOCAL HPARA%, NUMBER%(2)
DOC_GETPREVPARALINE HWP%, HPARA%, NUMBER%
```

Voir aussi [DOC_GETLINE](#), [DOC_GETLINEINFO](#), [DOC_GETPARALINEOFFSET%](#), [DOC_GETNEXTPARALINE](#)

Instruction DOC_GETSELECTION

Donne les coordonnées du texte sélectionné.

pos-deb et pos-fin sont des positions relatives au début du paragraphe. Elles sont exprimées en nombre de caractères, le premier caractère ayant la position 0, le deuxième la position 1 et ainsi de suite.

Syntaxe	DOC_GETSELECTION <i>handle-WP, handle-parag-deb, pos-deb, handle-parag-fin, pos-fin</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag-deb	INT(4)	!V! handle du paragraphe où débute la sélection
	pos-deb	INT(2)	!V! position du caractère sur lequel débute la sélection
	handle-parag-fin	INT(4)	!V! handle du paragraphe où finit la sélection
	pos-fin	INT(2)	!V! position du caractère sur lequel finit la sélection

Si aucune sélection n'est en cours, les coordonnées retournées sont celles du curseur. Dans ce cas, handle-deb et handle-fin sont identiques ainsi que pos-deb et pos-fin.

Exemple :

```

LOCAL HPARA1% (4), CARA1%(2)
LOCAL HPARA2% (4), CARA2%(2)

; Obtention des coordonnées de la sélection
DOC_GETSELECTION HWP%, HPARA1%, CARA1%, HPARA2%, CARA2%

; Affichage du résultat
MESSAGE "Début sélection :" && HPARA1% && CARA1%, "Fin sélection :" &&
HPARA2% && CARA2%
```

Voir aussi DOC SETSELECTION, DOC ISSELECTION%, DOC GETSELECTIONTEXT\$

Instruction **DOC_GETTABLINES**

Copie dans le segment `descript-tabs-vert` les valeurs des tabulations verticales du paragraphe courant.

Syntaxe	DOC_GETTABLINES <i>handle-WP, descript-tabs-vert</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-tabs-vert	DOC_TABLINES	!V! descriptif des tabulations verticales

Exemple :

```
LOCAL DOC_TABLINES MYTABLINES
DOC_GETTABLINES HWP%, MYTABLINES
```

Voir aussi [Segment DOC TABLINES](#), [DOC SETTABLINES](#)

Instruction DOC_GETTABS

Copie dans le segment descript-tabs les caractéristiques des retraits et des taquets de tabulation du paragraphe courant.

Syntaxe	DOC_GETTABS <i>handle-WP, descript-tabs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-tabs	DOC_TABS	!V! descriptif des retraits et des taquets de tabulation

1. Basée sur le segment DOC_TABS, DOC_GETTABS gère moins d'informations que DOC_GETXTABS. Elle est conservée dans l'A.P.I. par souci de compatibilité ascendante.
2. DOC_GETPARAGRAPHINFO renseigne aussi sur les retraits et les tabulations.
3. Equivalent : Règle -> Affichage des retraits et des taquets de tabulation

Exemple :

```

LOCAL DOC_TABS MYTABS
LOCAL I%(1)

; Lecture des tabulations et retraits du paragraphe courant
DOC_GETTABS HWP%, MYTABS

; Affichage des valeurs des retraits
MESSAGE "Retrait gauche", MYTABS.LEFTINDENT
MESSAGE "Retrait gauche 1ère ligne", MYTABS.FIRSTLINE
MESSAGE "Retrait droit", MYTABS.RIGHTINDENT

; Affichage des tabulations
MOVE MYTABS.NBTABS TO I%
MESSAGE "Nb de tabulations", I% ; Maximum 64
WHILE I% > 0
MESSAGE "Position tabulation" && I%, MYTABS.TABS[I%]
MOVE I%-1 TO I%
ENDWHILE

```

Voir aussi Constante **DOC MAX TABULATIONS%**, Segment **DOC TABS**, **DOC SETTABS**, **DOC GETXTABS**, **DOC GETPARAGRAPHINFO**

Instruction DOC_GETTEXT

Récupère dans une zone mémoire un extrait du paragraphe de handle *handle-parag*. L'extrait commence à la position spécifiée dans *pos-premier*. Il finit à la fin du paragraphe si la fin du paragraphe est atteinte avant la limite de taille fixée dans *taille-max*. Sinon, il finit quand il occupe une taille équivalente à *taille-max*.

taille-max est exprimée en octets. Elle peut être calculée par l'intermédiaire de l'instruction NCL SIZEOF.

Syntaxe	DOC_GETTEXT <i>handle-WP, handle-parag, pos-premier, adresse-mem, taille-max</i>		
Paramètres	<i>handle-WP</i>	INT(4)	handle de la fenêtre NS-WP
	<i>handle-parag</i>	INT(4)	handle du paragraphe
	<i>pos-premier</i>	INT(2)	nombre de caractères précédant dans le paragraphe le premier caractère du texte à récupérer
	<i>adresse-mem</i>	INT(4)	adresse de la zone mémoire
	<i>taille-max</i>	INT(2)	taille maximale disponible dans la zone mémoire

Le texte obtenu est formaté. Les codes des attributs (gras, italique, ...) des caractères y figurant comptent dans la taille occupée (voir l'annexe "Format de Fichier NS-WP").

Exemple :

```
SEGMENT MEM
CHAR TEXT$(32000)
ENDSEGMENT

LOCAL HMEM%

; Allocation mémoire
NEW MEM, HMEM%

; Lecture de 32000 caractères maxi du paragraphe courant
DOC_GETTEXT HWP%, DOC_CURRENTPARAGRAPH%(HWP%), 0, HMEM%, SIZEOF MEM

.... ; Traitement de MEM(HMEM%).TEXT$

; Libération mémoire
DISPOSE HMEM%
```

Voir aussi [DOC_GETTEXTSIZE%](#), [DOC_INSERTCHARS](#), [DOC_GETPARAGRAPHINFO](#), [DOC_GETSELECTION](#), [DOC_GETSELECTIONTEXT\\$](#)

Instruction DOC_GETXTABS

Copie dans le segment `descript-tabs` les caractéristiques des retraits et des taquets de tabulation du paragraphe courant.

Syntaxe	DOC_GETXTABS <i>handle-WP, descript-tabs</i>		
Paramètres	<code>handle-WP</code>	<code>INT(4)</code>	handle de la fenêtre NS-WP
	<code>descript-tabs</code>	<code>DOC_XTABS</code>	!V! descriptif des retraits et des taquets de tabulation

1. Basée sur le segment `DOC_XTABS`, `DOC_GETXTABS` gère plus d'informations que `DOC_GETTABS`, qui n'est conservée dans l'A.P.I. que par souci de compatibilité ascendante.
2. `DOC_GETPARAGRAPHINFO` renseigne aussi sur les retraits et les tabulations.
3. Equivalent : Règle -> Affichage des retraits et des taquets de tabulation

Exemple :

```

LOCAL DOC_XTABS XTABS
LOCAL I%(2)

; Lecture des tabulations et retraits du paragraphe courant
DOC_GETXTABS HWP%, XTABS

; Affichage des valeurs des retraits
MESSAGE "Retrait gauche", XTABS.LEFTINDENT
MESSAGE "Retrait gauche 1ère ligne", XTABS.FIRSTLINE
MESSAGE "Retrait droit", XTABS.RIGHTINDENT

; Affichage des tabulations
MOVE XTABS.NBTABS TO I%
MESSAGE "Nb de tabulations", I% ; Maximum 64
WHILE I% > 0
MESSAGE "Position tabulation" && I%, XTABS.TABS[I%]
MESSAGE "Alignement tabulation" && I%, XTABS.TABTYPE[I%]
MESSAGE "Point de suite tabulation" && I%, XTABS.TABLEAD[I%]
MOVE I%-1 TO I%
ENDWHILE

```

Voir aussi Constante `DOC_MAX_TABULATIONS%`, Segment `DOC_XTABS`, `DOC_SETXTABS`, `DOC_GETTABS`, `DOC_GETPARAGRAPHINFO`

Instruction DOC_INSERT

Insère un fichier à la position du curseur.

Syntaxe	DOC_INSERT <i>handle-WP, nom-fichier</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-fichier	CSTRING	chemin et nom du fichier à insérer

1. DOC_INSERT accepte de charger des fichiers de trois formats différents : ASCII, RTF et NSWP.
2. DOC_OPEN permet également d'insérer un document à la fin du document courant, lorsque le nom du fichier est précédé de "&".
3. A la suite de DOC_INSERT, DOC_ERROR% peut être utilisé pour retourner les codes d'erreur.

Exemple :

```
; Insertion d'un document au début du document courant
DOC_SETPOS HWP%, DOC_FIRSTPARAGRAPH%(HWP%), 0
DOC_INSERT HWP%, "FICHIER.DOC"
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Fichier non inséré"
ENDIF
```

Voir aussi [DOC_ERROR%](#), [DOC_OPEN](#), [DOC_NEW](#), [DOC_INSERTCHARS](#), [DOC_PASTE](#)

Instruction **DOC_INSERTCHARS**

Insère dans le paragraphe de handle *handle-parag* du texte provenant d'une zone mémoire.

Le texte est inséré juste après la position *pos-insert*.

pos-insert est une position relative au début du paragraphe et est exprimée en nombre de caractères. Le premier caractère a pour position 0.

Syntaxe	DOC_INSERTCHARS <i>handle-WP, handle-parag, pos-insert, adresse-texte, taille texte</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-insert	INT(2)	position d'insertion dans le paragraphe
	adresse-texte	INT(4)	adresse de la zone mémoire contenant le texte à insérer
	taille-texte	INT(2)	nombre de caractères du texte à insérer

1. Le texte est inséré avec la police du point d'insertion.
2. Si *handle-parag* vaut 0, l'insertion se fait à la position du curseur. Le curseur est ensuite positionné après le texte inséré.

Exemple :

```
; Insertion de "Monsieur" au début du paragraphe courant
LOCAL S$(80)

MOVE "Monsieur" TO S$
DOC_INSERTCHARS HWP%, DOC_CURRENTPARAGRAPH%(HWP%), 0, @S$, LENGTH S$
```

Voir aussi [DOC_DELETECHARS](#), [DOC_GETTEXT](#), [DOC_PASTE](#), [DOC_INSERT](#)

Instruction **DOC_LAYOUT**

Valide ou invalide le formatage du texte et la mise à jour à l'écran.

- Si indic-action vaut FALSE%, le texte n'est plus formaté et l'écran n'est plus mis à jour.
- Si indic-action vaut TRUE%, toute modification dans le texte est immédiatement répercutée à l'écran et entraîne un reformatage lié aux marges, aux tabulations...

Syntaxe	DOC_LAYOUT <i>handle-WP, indic-action</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-action	INT(1)	indicateur d'action à effectuer

1. Invalider le formatage du texte et la mise à jour de l'écran permet de gagner en rapidité d'exécution. Ce gain de temps est particulièrement significatif si le texte fait l'objet de manipulations importantes et répétées. Il est à recommander également s'il permet d'éviter des formatages redondants.
2. L'instruction **DOC_UPDATE** permet aussi de ne pas mettre à jour l'écran.

Exemple :

```
; Ouverture d'une fenêtre cachée
OPENH FEN, HWP%, W%

; Formatage et mise à jour écran invalidés
DOC_LAYOUT W%, FALSE%

; Appel de diverses fonctions
DOC_SETXTABS W%, XTABS
DOC_SETFONT W%, 'Tms Rmn 10'
DOC_MODIFYFORMAT W%, DEFRET%, 0

; Fermeture de la fenêtre cachée
CLOSE W%
```

Voir aussi DOC_UPDATE

Instruction **DOC_LOCKFONTS**

Interdit dans le document l'utilisation de styles autres que celui spécifié par **DOC_DEFAULTFONT**.

Syntaxe	DOC_LOCKFONTS <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

Exemple :

```
IF DOC_ISFONTLOCKED%(HWP%) = FALSE%  
DOC_LOCKFONTS HWP%  
ENDIF
```

Voir aussi **DOC_DEFAULTFONT**, **DOC_DEFAULTLOCKFONTS**, **DOC_ISFONTLOCKED%**

Instruction **DOC_M_DEREGISTERMARKERCLASS**

Supprime le lien entre le dictionnaire de données dont le handle est handle-dictionnaire et le document NS-WP.

Syntaxe	DOC_M_DEREGISTERMARKERCLASS handle-WP, handle-dictionnaire		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-dictionnaire	INT(4)	handle du dictionnaire

Exemple :

```
; Suppression du lien entre le dictionnaire de handle ACLASS% et la fenêtre  
NS-WP  
DOC_M_DEREGISTERMARKERCLASS HWP%, ACLASS%
```

Voir aussi **DOC M REGISTERMARKERCLASS**

Instruction **DOC_M_DISPOSEMARKER**

Supprime une entrée de données dans un dictionnaire.

L'entrée de données est identifiée par son handle.

Syntaxe	DOC_M_DISPOSEMARKER <i>handle-dictionnaire, handle-données</i>		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
	handle-données	INT(4)	handle de l'entrée de données

Exemple :

```
; Suppression de l'entrée de handle CSTYLE_SHEET% dans le dictionnaire de  
handle ACLASS%  
DOC_M_DISPOSEMARKER ACLASS%, CSTYLE_SHEET%
```

Voir aussi [DOC M_CREATEMARKER%](#), [DOC M_CREATEMARKERWITHHANDLE%](#)

Instruction **DOC_M_DISPOSEMARKERCLASS**

Détruit un dictionnaire de données.

Syntaxe	DOC_M_DISPOSEMARKERCLASS <i>handle-dictionnaire</i>		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire

Exemple :

```
; Suppression du dictionnaire de handle ACLASS%  
DOC_M_DISPOSEMARKERCLASS ACLASS%
```

Voir aussi **DOC M_NEWMARKERCLASS%**

Instruction **DOC_M_GETMARKERDATA**

Récupère les données du dictionnaire stockées dans l'entrée de handle handle-données.

Les données sont décrites par leur adresse et leur taille.

Syntaxe	DOC_M_GETMARKERDATA <i>handle-dictionnaire, handle-données, adresse données, taille-mémoire</i>		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
	handle-données	INT(4)	handle de l'entrée de données
	adresse-données	INT(4)	adresse mémoire des données
	taille-mémoire	INT(2)	taille en octets des données

Exemple :

```
; Contenu d'une entrée du dictionnaire
LOCAL CSTRING DATA$

DOC M GETMARKERDATA ACLASS%, HMARKER%, @DATA$, SIZEOF DATA$
MESSAGE "", "Les données de handle" && HMARKER% && "contenues dans le
dictionnaire sont" && DATA$
```

Voir aussi [DOC M_GETMARKERDATASIZE%](#), [DOC M_SETMARKERDATA](#)

Instruction DOC_M_INSERTMARKER

Insère dans le document un marqueur, et son texte, à la position désignée par handle-parag et pos-marqueur.

Le marqueur est décrit par son handle et son texte.

Lorsque handle-parag et pos-marqueur sont positionnés à 0, le marqueur est inséré à gauche du curseur.

pos-marqueur est le nombre de caractères précédant le marqueur dans le paragraphe.

Syntaxe	DOC_M_INSERTMARKER <i>handle-WP, handle-parag, pos-marqueur, texte-marqueur, handle-marqueur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	position du marqueur dans le paragraphe
	texte-marqueur	CSTRING	texte du marqueur
	handle-marqueur	INT(4)	handle du marqueur

1. Le marqueur placé dans le document peut être lié à des données stockées dans un dictionnaire. Il suffit d'indiquer dans handle-marqueur un handle de données créé précédemment à l'aide de DOC_M_CREATEMARKER% ou de DOC_M_CREATEMARKERWITHHANDLE%.

2. Le texte d'un marqueur est vu comme un seul caractère.

3. Un marqueur s'efface comme un caractère quelconque, par exemple au moyen des touches [Suppr] ou [Backspace]. Cependant l'effacement d'un marqueur peut être régi grâce à l'instruction DOC_MAPUSERMSG et au segment DOC_M_MARKER. Il est ainsi possible d'empêcher toute suppression de marqueurs.

Exemple 1 :

```
; Insertion d'un flag à gauche du curseur
DOC_M_INSERTMARKER HWP%, 0, 0, "Flag", HMARKER%
MOVE HMARKER% + 1 TO HMARKER%
```

Exemple 2 :

```
; Association d'un marqueur et de données
LOCAL CSTRING DATA$

MOVE "Drapeau placé dans le texte" TO DATA$
MOVE DOC_M_CREATEMARKER% (AClass%) TO HMARKER%
DOC_M_SETMARKERDATA AClass%, HMARKER%, @DATA$, SIZEOF DATA$
```

```
DOC_M_INSERTMARKER HWP%, 0, 0, "Flag", HMARKER%
```

Voir aussi Segment DOC M MARKER, DOC MAPUSERMSG, DOC M CREATEMARKER%, DOC M CREATEMARKERWITHHANDLE%, DOC M SETMARKERDATA, DOC M ISMARKER%, DOC M GETTEXT\$, DOC M SETTEXT

Instruction **DOC_M_REGISTERMARKERCLASS**

Attache un dictionnaire de données au document NS-WP. De la sorte, le dictionnaire sera sauvegardé avec le document.

Syntaxe	DOC_M_REGISTERMARKERCLASS <i>handle-WP, handle-dictionnaire</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-dictionnaire	INT(4)	handle du dictionnaire

Plusieurs dictionnaires de données peuvent être attachés à un même document.

Exemple :

```
; Création d'un lien entre le dictionnaire de handle ACLASS% et la fenêtre NS-WP
DOC_M_REGISTERMARKERCLASS HWP%, ACLASS%
```

Voir aussi **DOC_M_DEREGISTERMARKERCLASS**

Instruction **DOC_M_SETMARKERDATA**

Insère des données à l'entrée du handle handle-données du dictionnaire.

Les données sont identifiées par leur adresse et leur taille mémoire.

Syntaxe	DOC_M_SETMARKERDATA <i>handle-dictionnaire, handle-données, adresse données, taille-mémoire</i>		
Paramètres	handle-dictionnaire	INT(4)	handle du dictionnaire
	handle-données	INT(4)	handle de l'entrée
	adresse-données	INT(4)	adresse mémoire des données
	taille-mémoire	INT(2)	taille en octets des données

1. L'entrée doit déjà exister. Il n'est donc possible d'effectuer un **DOC_M_SETMARKERDATA** qu'après un **DOC_M_CREATEMARKER%** ou un **DOC_M_CREATEMARKERWITHHANDLE%**.
2. La taille mémoire occupée par les données doit être précisée. Elle se calcule avec l'instruction **NCL SIZEOF**.

Exemple :

```
; Insertion de données dans le dictionnaire de handle ACLASS%
MOVE "Drapeau placé dans le texte" TO DATA$
MOVE DOC_M_CREATEMARKER% (AClass%) TO HMarker%
DOC_M_SETMARKERDATA ACLASS%, HMarker%, @DATA$, SIZEOF DATA$
```

Voir aussi **DOC M_CREATEMARKER%**, **DOC M_CREATEMARKERWITHHANDLE%**

Instruction DOC_M_SETTEXT

Remplace le texte du marqueur se trouvant à la position spécifiée par le texte écrit dans texte-marqueur.

pos-marqueur est le nombre de caractères précédant le marqueur dans le paragraphe.

Syntaxe	DOC_M_SETTEXT <i>handle-WP, handle-parag, pos-marqueur, texte-marqueur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe dans lequel est situé le marqueur
	pos-marqueur	INT(2)	position du marqueur dans le paragraphe
	texte-marqueur	CSTRING	nouveau texte pour le marqueur

Exemple :

```
; Remplacement du texte du marqueur se trouvant à la position OFFSET% du
paragraphe de handle HPARA%
IF DOC_M_ISMARKER% (HWP%, HPARA%, OFFSET%) = FALSE%
MESSAGE "Erreur", "Cette position ne correspond pas à un marqueur"
ELSE
DOC_M_SETTEXT HWP%, HPARA%, OFFSET%, "Flag"
ENDIF
```

Voir aussi DOC M INSERTMARKER, DOC M ISMARKER%, DOC M GETTEXT\$

Instruction DOC_MAPUSERMSG

Provoque l'envoi d'un événement utilisateur à la fenêtre de handle handle-fen-récept à chaque fois qu'une modification de type type-modif survient dans la fenêtre NS-WP.

Plusieurs types de modifications sont reconnus. Ils sont référencés par les constantes DOC_*_EVENT.

Le nom de l'événement utilisateur est le résultat de la concaténation de USER et de num-événement.

num-événement est une des constantes DOC_USER_MESSAGE_*.

Lorsque DOC_MAPUSERMSG a été activée pour les actions de type DOC_RULER_EVENT (modification de l'état du texte) :

- Chaque modification d'état survenant dans le document est enregistrée dans le segment DOC_RULEINFO.

Lorsque DOC_MAPUSERMSG a été activée pour les actions de type DOC_M_DELETEMARKER_EVENT (suppression d'un marqueur) :

- Le segment DOC_M_MARKER est mis à jour à chaque tentative de suppression de marqueur.
- La valeur retournée par l'événement utilisateur détermine le traitement :
 - RETURN TRUE% : le marqueur est effacé
 - RETURN FALSE% : la suppression n'a pas lieu

Lorsque DOC_MAPUSERMSG a été activée pour les actions de type DOC_M_PASTEMARKER_EVENT (copie du presse-papiers) :

- Le segment DOC_M_MARKER est mis à jour à chaque copie de marqueur.
- La valeur retournée par l'événement utilisateur détermine le traitement :
 - RETURN TRUE% : le marqueur est copié
 - RETURN FALSE% : la copie n'a pas lieu

Syntaxe	DOC_MAPUSERMSG <i>handle-WP, handle-fen-récept, type-modif, num-événement</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-fen-récept	INT(4)	handle de la fenêtre réceptrice
	type-modif	INT(2)	type de la modification affectant la fenêtre NS-WP
	num-événement	INT(2)	numéro de l'événement activé

1. Cette instruction est très utile pour gérer une barre d'outils. En effet, le développeur qui désire gérer sa propre barre d'outils doit être capable de répercuter au niveau des champs et des icônes de la barre les manipulations effectuées dans le plan de travail. Il le peut grâce à l'usage combiné de cette instruction et des segments DOC_RULEINFO et DOC_M_MARKER. La barre d'outils constitue la fenêtre réceptrice.

2. Grâce à DOC_MAPUSERMSG, la barre d'outils est avertie chaque fois qu'une manipulation est effectuée dans le plan de travail. Du code stocké dans des événements utilisateur se déroule alors automatiquement. Ce code peut tenir compte de la manipulation effectuée par l'intermédiaire des segments DOC_RULEINFO et DOC_M_MARKER, mis à jour dès qu'il y a action sur le plan de travail.

Exemples :

```
; Soit la barre d'outils de handle TOOLBOX%
; L'événement USER0 de la barre d'outils sera activé si une modification de
type DOC_SETFOCUS_EVENT survient dans la fenêtre NS-WP
DOC_MAPUSERMSG HWP%, TOOLBOX%, DOC_SETFOCUS_EVENT, DOC_USER_MESSAGE_0

; L'événement USER1 de la barre d'outils sera activé si une modification de
type DOC_RULER_EVENT survient dans la fenêtre NS-WP
DOC_MAPUSERMSG HWP%, TOOLBOX%, DOC_RULER_EVENT, DOC_USER_MESSAGE_1

; L'événement USER2 de la barre d'outils sera activé si une modification de
type DOC_FONTNAME_EVENT survient dans la fenêtre NS-WP
DOC_MAPUSERMSG HWP%, TOOLBOX%, DOC_FONTNAME_EVENT, DOC_USER_MESSAGE_2

; L'événement USER3 de la barre d'outils sera activé si une modification de
type DOC_FONTSIZE_EVENT survient dans la fenêtre NS-WP
DOC_MAPUSERMSG HWP%, TOOLBOX%, DOC_FONTSIZE_EVENT, DOC_USER_MESSAGE_3
```

Voir aussi Constantes DOC * EVENT, DOC USER MESSAGE *, DOC RULEINFO, DOC M MARKER

Instruction DOC_MERGE

Crée, par fusion, un fichier au format NS-WP.

Le nom du fichier créé est précisé dans nom-fichier-résult. Il est obtenu automatiquement à partir des commandes figurant dans le fichier nom-fichier-cmde et, éventuellement, à partir des données associées à la fenêtre NS-WP.

L'ensemble des commandes utilisables est décrit dans l'annexe "Format de Fichier de Fusion".

En cas d'erreur dans le fichier de commande ou lors de la création du fichier NS WP, le code de l'erreur est copié dans la variable code-erreur. Le numéro de la ligne posant problème est indiqué dans ligne-erreur. code-erreur vaut l'une des constantes DOC_ERR*%.

Syntaxe	DOC_MERGE <i>handle-WP, code-erreur, ligne-erreur, nom-fichier-cmde, nom fichier-résult</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	code-erreur	INT(2)	!V! code d'erreur
	ligne-erreur	INT(2)	!V! numéro de ligne erronée
	nom-fichier-cmde	CSTRING	chemin et nom du fichier de commandes de fusion
	nom-fichier-résult	CSTRING	chemin et nom du fichier résultat de la fusion

1. Le code d'erreur peut également être obtenu par appel à la fonction DOC_ERROR%.

Les codes d'erreur possibles sont :

```
DOC_ERRFILENOTFOUND%
DOC_ERRSYNTAXERROR%
DOC_ERRCANNOTWRITEONFILE%
DOC_ERRTOOMANYINCLUDES%
DOC_ERRUNKNOWNVARIABLE%
DOC_ERRSTACKOVERFLOW%
DOC_ERRINVALIDVARIABLE%.
```

2. En cas d'erreur, l'instruction DOC_GETLASTMERGEERROR permet d'obtenir plus de précisions sur l'erreur détectée.

Exemple :

```
LOCAL ERRCODE%(2), ERRLINE%(2)
LOCAL LASTNAME$, LASTERROR$

DOC_MERGE HWP%, ERRCODE%, ERRLINE%, "COMMANDS.DOC", "RESULT.DOC"
IF ERRCODE% <> DOC_ERRNO%
DOC_GETLASTMERGEERROR HWP%, LASTNAME$, LASTERROR$
MESSAGE "Erreur" && ERRCODE% && LASTERROR$, "La fusion a échoué ligne" &&
```

```
ERRLINE% && "dans le fichier" && LASTNAME$  
ENDIF
```

Voir aussi Constantes DOC ERR*%, DOC ERROR%, DOC GETLASTMERGEERROR, DOC OPEN, DOC SAVE, Annexe "Format de Fichier de Fusion"

Instruction DOC_METRICS

Modifie le système de mesure utilisé dans la fenêtre NS-WP.

syst-mesure vaut l'une des constantes DOC_LO*%, DOC_CHARS%.

Syntaxe	DOC_METRICS <i>handle-WP, syst-mesure</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	syst-mesure	INT(2)	système de mesure

1. L'aspect de la règle graduée change selon le système sélectionné.
2. NS-WP fonctionne par défaut dans le système métrique, sauf si une autre mesure par défaut a été définie avec DOC_DEFAULTMETRICS.

Exemple :

```
; Fonctionnement en système anglo-saxon
DOC_METRICS HWP%, DOC_LOENGLISH%
```

Voir aussi Constantes DOC LO*%, DOC CHARS%, Segments DOC BORDER, DOC TABS,
DOC XTABS, LINE BORDER, DOC TABLINE,
DOC DEFAULTMETRICS, DOC ISMETRICS%, DOC DEFAULTMAXPARAWIDTH,
DOC DEFAULTPARAGRAPHINDENT, DOC DEFAULTTAB,
DOC GETCURRENTTABLINE, DOC GETDEFTAB%, DOC GETMARGINS, DOC GETPAGEDIM,
DOC SETCURRENTTABLINE,
DOC SETDEFTAB, DOC SETMARGINS, DOC GETPICTSIZE%

Instruction DOC_MODIFYFORMAT

Modifie les attributs de caractère du texte sélectionné ou du point d'insertion si aucune sélection n'est en cours. Certains attributs sont retirés et d'autres appliqués. attributs-sup et attributs-ajout sont des constantes DOC_FMT_% ou des combinaisons de constantes DOC_FMT_%.

Syntaxe	DOC_MODIFYFORMAT <i>handle-WP, attributs-sup, attributs-ajout</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	attributs-sup	INT(1)	attributs à retirer (gras, italique...)
	attributs-ajout	INT(1)	attributs à ajouter (gras, italique...)

1. Pour supprimer la totalité des attributs, employer la constante DEFRET% en deuxième paramètre.

2. Equivalents :

a) Clavier [Ctrl]+[B] | [I] | [S] | [U] selon attribut

b) Barre d'outils standard



Exemple :

```
; Suppression du gras et mise en italique  
DOC_MODIFYFORMAT HWP%, DOC_FMT_BOLD%, DOC_FMT_ITALIC%
```

Voir aussi Constantes DOC_FMT %, DOC_GETFORMAT%, DOC_SETFORMAT, DOC_SETFONT

Instruction **DOC_MOVESELECTION**

Transfère le texte sélectionné dans le document source à la position courante du document de destination.

Syntaxe	DOC_MOVESELECTION <i>handle-WP-src, handle-WP-dest</i>		
Paramètres	handle-WP-src	INT(4)	handle de la fenêtre NS-WP contenant le document source
	handle-WP-dest	INT(4)	handle de la fenêtre NS-WP contenant le document de destination

Cette instruction permet de transférer du texte entre fenêtres NS-WP sans passer par le presse-papiers, d'où un gain en rapidité.

Exemple :

```
IF DOC_ISSELECTION% (HWP1%)  
DOC_MOVESELECTION HWP1%, HWP2%  
ENDIF
```

Voir aussi [DOC_COPYSELECTION](#), [DOC_ISSELECTION%](#)

Instruction **DOC_NEW**

Efface complètement le contenu de la fenêtre NS-WP.

Syntaxe	DOC_NEW <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

Equivalent : Clavier [Del] si texte entier sélectionné

Exemple :

```
IF ASK2%("Attention", "Voulez-vous tout effacer ?") = YES%  
DOC_NEW HWP%  
ENDIF
```

Voir aussi DOC_OPEN, DOC_CUT, DOC_DELETE, DOC_DELETECHARS

Instruction DOC_OPEN

Charge dans la fenêtre NS-WP un document stocké sur disque ou dans un buffer mémoire.

Le document chargé remplace le document éventuellement déjà affiché dans la fenêtre NS WP, sauf si :

- le caractère '&' figure au début de la chaîne nom

Dans ce cas, il est concaténé au document éventuellement déjà chargé dans la fenêtre NS WP.

- le caractère '@' figure au début de la chaîne nom

Dans ce cas, il s'insère dans le document éventuellement déjà chargé dans la fenêtre NS-WP à la position du curseur.

Le segment DOC_MEMFILE sert à décrire le buffer mémoire.

Syntaxe	DOC_OPEN <i>handle-WP</i> , <i>nom</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom	CSTRING	chemin et nom du fichier à ouvrir ou '#' concaténé à l'adresse du segment décrivant le buffer

- DOC_OPEN accepte de charger des documents de trois formats différents : NSWP, ASCII et RTF. DOC_OPEN accepte également de charger des fichiers disque au format RFT.
- "DOC_OPEN HWP%, F\$" est rigoureusement équivalent à "LOAD F\$ TO HWP%".
- A la suite de DOC_OPEN, DOC_ERROR% peut être utilisé pour retourner les codes d'erreur.

Exemple 1 :

```
; Ouverture de "fichier1.doc"
DOC_OPEN HWP%, "FICHIER1.DOC"
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Document non chargé"
ENDIF
```

Exemple 2 :

```
; Ajout de "fichier2.doc"
DOC_OPEN HWP%, "&FICHIER2.DOC"
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Document non chargé"
ENDIF
```

Exemple 3 :

```
; Ouverture du document contenu dans le buffer d'adresse BUFF%
LOCAL DOC_MEMFILE MEMFILE
```

```
MOVE 0 TO MEMFILE.TYPE
MOVE 64000 TO MEMFILE.MAXSIZE
MOVE BUFF% TO MEMFILE.BUFF
MOVE '#' & @MEMFILE TO S$

DOC_OPEN HWP%, S$
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Document non chargé"
ENDIF
```

Voir aussi Segment DOC MEMFILE, DOC ERROR%, DOC INSERT, DOC SAVE, DOC NEW, LOAD (Manuel de Programmation NCL)

Instruction **DOC_PAGEBREAK**

Insère un saut de page forcé au début du paragraphe courant.

Syntaxe	DOC_PAGEBREAK <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

1. Insérer un saut de page à deux reprises au même endroit revient à l'effacer.
2. Un saut de page forcé se matérialise à l'écran par un pointillé serré.
3. Equivalent : Clavier [Ctrl]+[P]

Exemple :

```
; Insertion d'un saut de page
IF NOT DOC_ISPAGEBREAK%(HWP%)
DOC_PAGEBREAK HWP%
ENDIF
```

Voir aussi [DOC_ISPAGEBREAK%](#), [DOC_GETNBPAGES%](#)

Instruction **DOC_PASTE**

Insère le contenu du presse-papiers à gauche du curseur.

Syntaxe	DOC_PASTE <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

1. Le presse-papiers peut avoir été préalablement rempli à l'aide des instructions **DOC_COPY** ou **DOC_CUT**.
2. Equivalent : Clavier [Shift]+[Ins]

Exemple :

```
IF DOC_ISCLIPBOARD% (HWP%)  
DOC_PASTE HWP%  
ENDIF
```

Voir aussi **DOC_COPY**, **DOC_CUT**, **DOC_DELETE**, **DOC_UNDO**, **DOC_ISCLIPBOARD%**,
DOC_INSERTCHARS, **DOC_INSERT**

Instruction DOC_READONLY

Modifie le mode d'accessibilité du document.

- Si indic-accès vaut TRUE%, la fenêtre NS-WP n'est accessible qu'en mode "lecture". Aucune modification de texte n'est alors possible.
- Si indic-accès vaut FALSE%, la fenêtre NS-WP est accessible en mode "lecture/écriture".

Syntaxe	DOC_READONLY <i>handle-WP, indic-accès</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-accès	INT(1)	indicateur de passage en mode "lecture seulement"

1. A moins d'être expressément mis en mode "lecture seulement", un document est considéré comme étant en mode "lecture/écriture".
2. Des parties d'un texte en mode "lecture seulement" peuvent avoir été positionnées en "lecture écriture", grâce à l'attribut de caractère DOC_FMT_UNLOCKED%.
3. "DOC_READONLY HWP%, TRUE%" est rigoureusement équivalent à "LOCK HWP%".
4. "DOC_READONLY HWP%, FALSE%" est rigoureusement équivalent à "UNLOCK HWP%".

Exemple :

```
; Inversion de l'état de blocage, sur événement EXECUTED de l'élément de
menu correspondant
IF DOC_ISREADONLY%(HWP%)
DOC_READONLY HWP%, FALSE%
MOVE UNCHECKED% TO MENU_READONLY
ELSE
DOC_READONLY HWP%, TRUE%
MOVE CHECKED% TO MENU_READONLY
ENDIF
```

Voir aussi Constantes DOC_FMT *%, DOC_ISREADONLY%, LOCK, UNLOCK (Manuel de Programmation NCL)

Instruction **DOC_REDO**

Rejoue la dernière action annulée par un **DOC_UNDO**.

Syntaxe	DOC_REDO <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

Exemple :

```
IF DOC_ISREDO% (HWP%)  
DOC_REDO HWP%  
ENDIF
```

Voir aussi **DOC_ISREDO%**, **DOC_UNDO**

Instruction DOC_RULERBAR

Gère l'affichage de la barre d'outils standard et de la règle dans la fenêtre NS WP.

- Si indic-affichage vaut TRUE%, la barre d'outils standard et la règle sont affichées.
- Si indic-affichage vaut FALSE%, la barre d'outils standard et la règle ne sont plus affichées.

Syntaxe	DOC_RULERBAR <i>handle-WP, indic-affichage</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-affichage	INT(1)	indicateur d'affichage de la barre d'outils standard et de la règle

Equivalent : à l'appel successif des deux instructions :

DOC_SHOWRULERBAR

DOC_SHOWTOOLBOX

Exemple :

```
; Inversion de l'état de la barre d'outils et de la règle
DOC_RULERBAR HWP%, NOT DOC_ISRULERBAR% (HWP%)
```

Voir aussi DOC_ISRULERBAR%, DOC_ISRULERBARSHOWN%, DOC_SHOWRULERBAR, DOC_ISTOOLBOXSHOWN%, DOC_SHOWTOOLBOX

Instruction DOC_SAVE

Sauvegarde dans un fichier ou dans un buffer mémoire le document affiché dans la fenêtre NS-WP.

format est une des constantes DOC_FILE_ *%.

Le segment DOC_MEMFILE sert à décrire le buffer mémoire.

Syntaxe	DOC_SAVE handle-WP, nom, format		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom	CSTRING	chemin et nom du fichier ou '#' concaténé à l'adresse du segment décrivant le buffer
	format	INT(1)	format utilisé (NS-WP, ASCII, etc.)

1. "SAVE HWP% TO F\$" équivaut à sauvegarder le document dans le format qui lui est associé, renvoyé par DOC_FILE_FORMAT%.
2. A la suite de DOC_SAVE, DOC_ERROR% peut être utilisé pour retourner les codes d'erreur.
3. La sauvegarde dans un buffer mémoire ne fonctionne pas avec le format RFT.

Exemple 1 :

```
; Sauvegarde dans un fichier
DOC_SAVE HWP%, "FICHIER.DOC", DOC_FILE_NSWP%
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Document non sauvegardé"
ENDIF
```

Exemple 2 :

```
; Sauvegarde dans le buffer d'adresse BUFF%
LOCAL DOC_MEMFILE MEMFILE

MOVE 0 TO MEMFILE.TYPE
MOVE 64000 TO MEMFILE.MAXSIZE
MOVE BUFF% TO MEMFILE.BUFF
MOVE '#' & @MEMFILE TO S$

DOC_SAVE HWP%, S$, DOC_FILE_NSWP%
IF DOC_ERROR%(HWP%) <> DOC_ERRNO%
MESSAGE "Erreur" && DOC_ERROR%(HWP%), "Document non sauvegardé"
ENDIF
```

Voir aussi Constantes DOC FILE *%, Segment DOC MEMFILE, DOC ERROR%, DOC OPEN, DOC FILE FORMAT%, SAVE (Manuel de Programmation NCL), Annexe "Format de fichier NS-WP".

Instruction **DOC_SECTION**

Insère un saut de section au début du paragraphe courant.

Syntaxe	DOC_SECTION <i>handle-WP</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP

1. Insérer un saut de section à deux reprises au même endroit revient à l'effacer.
2. Le texte d'une section supprimée s'intègre à la section précédente et hérite de ses caractéristiques.
3. Un saut de section se matérialise à l'écran par une ligne continue.

Exemple :

```
; Positionnement d'un saut de section
IF DOC_ISSECTION% (HWP%) = FALSE%
DOC_SECTION HWP%
ENDIF
```

Voir aussi [DOC_ISSECTION%](#), [DOC_GETNBSECTIONS%](#), [DOC_GETSECTION%](#), [DOC_SETSECTION](#), [DOC_GETSECTIONJUSTIF%](#), [DOC_SETSECTIONJUSTIF](#)

Instruction **DOC_SET_INSERT_MODE**

Modifie le mode d'insertion des caractères.

- Si indic-insert vaut TRUE%, le mode "insertion" est activé. Les caractères tapés s'insèrent dans le texte.
- Si indic-insert vaut FALSE%, le mode "surimpression" est activé. Les caractères tapés remplacent les caractères existants.

Syntaxe	DOC_SET_INSERT_MODE <i>handle-WP, indic-insert</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-insert	INT(1)	indicateur de mode d'insertion

Equivalent : Clavier [Ins]

Exemple :

```
; Passage en mode "insertion"
IF DOC_GET_INSERT_MODE% (HWP%) = FALSE%
DOC_SET_INSERT_MODE HWP%, TRUE%
ENDIF
```

Voir aussi **DOC_GET_INSERT_MODE%**

Instruction DOC_SETBORDER

Met un cadre (ou supprime un cadre) autour du paragraphe courant ou des paragraphes sélectionnés.

- Encadrement :

L'encadrement doit être complètement décrit dans le segment descript-cadre. La taille mémoire occupée par descript-cadre doit être précisée dans taille-mémoire.

- Suppression d'un encadrement :

Il suffit de positionner le paramètre taille-mémoire à 0 pour supprimer l'encadrement du paragraphe courant ou des paragraphes sélectionnés. descript-cadre ne sert pas pour une suppression.

Syntaxe	DOC_SETBORDER <i>handle-WP, descript-cadre, taille-cadre</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-cadre	DOC_BORDER	description de l'encadrement
	taille-mémoire	INT(2)	taille en octets de descript-cadre

La taille mémoire de descript-cadre peut se calculer avec l'instruction NCL SIZEOF.

Exemple 1 (encadrement) :

```

LOCAL DOC_BORDER BORDER
; Spécifications de l'encadrement
; Espacement entre le texte et les bords, motif utilisé en fond, couleur
d'avant-plan du motif, couleur d'arrière-plan du motif
move 10 to border.dwspace
move patsym_diag4% to border.pattern
move col_green% to border.forecolor
move col_white% to border.backcolor

; Bord gauche de l'encadrement : ici trait simple rouge d'épaisseur 5/10 mm
move 5 to border.line[doc_line_left%].width
move 0 to border.line[doc_line_left%].kind
move col_red% to border.line[doc_line_left%].color

; Bord droit
move 5 to border.line[doc_line_right%].width
move 0 to border.line[doc_line_right%].kind
move col_red% to border.line[doc_line_right%].color

; Bord haut
move 5 to border.line[doc_line_above%].width
move 0 to border.line[doc_line_above%].kind
move col_red% to border.line[doc_line_above%].color

; Bord bas
move 5 to border.line[doc_line_below%].width
move 0 to border.line[doc_line_below%].kind
move col_red% to border.line[doc_line_below%].color

; Trait séparant les paragraphes
move 5 to border.line[doc_line_between%].width

```

```

move 0 to border.line[doc_line_between%].kind
move col_RED% to border.line[doc_line_between%].color

; Ombrage :ici trait simple rouge d'épaisseur 5/10 mm
move 10 to border.line[doc_line_shadow%].width
move patsym_HALFTONE% to border.line[doc_line_shadow%].kind
move col_RED% to border.line[doc_line_shadow%].color

; Encadrement
doc_setborder hwp%, border, sizeof border

```

Exemple 2 : suppression du cadre entourant le paragraphe courant

```

doc_setborder hwp%, border, 0

```

Voir aussi Constantes DOC_LINE *%, Segments DOC BORDER, LINE BORDER, DOC GETBORDER, Constantes PATSYM *% (Librairie graphique NSGRAPH), Constantes COL *%, instruction SIZEOF (Manuel de Programmation NCL)

Instruction **DOC_SETCONTROLCHARSVISIBILITY**

Détermine la visibilité des caractères non imprimables et de certaines commandes :

- Les espaces entre les mots.
- Les marques de tabulation.
- Les marques de fin de paragraphe.
- Les commandes (bitmaps, marqueurs).
- Des cadres grisés à la place des images bitmaps.
- Les bornes.

codes-visibles est une combinaison des constantes `DOC_SHOW_*` et `DOC_HIDE_PICTURES%`.

Une image bitmap peut donc être remplacée à l'affichage par :

- La commande d'insertion de la bitmap, c'est-à-dire par exemple `<BMP:C:\NSDK\BMP\RABBITS.BMP>`.

ou

- Un cadre grisé.

Syntaxe	DOC_SETCONTROLCHARSVISIBILITY <i>handle-WP</i> , <i>codes-visibles</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	codes-visibles	INT(2)	codes visibles

Exemple 1 :

```
; Visualisation des espaces et des tabulations
DOC_SETCONTROLCHARSVISIBILITY HWP%, DOC_SHOW_SPACES% BOR \
DOC_SHOW_TABS%
```

Exemple 2 :

```
; Affichage d'un cadre gris à la place des images bitmaps
DOC_SETCONTROLCHARSVISIBILITY HWP%, DOC_HIDE_PICTURES%
```

Voir **aussi** **Constantes** **DOC_SHOW *%,** **DOC_HIDE_PICTURES%,**
DOC_GETCONTROLCHARSVISIBILITY%

Instruction **DOC_SETCURRENTTAB**

Indique que les taquets insérés par un double-clic souris sur la règle sont des taquets de tabulation.

Le type d'alignement et le style de point de suite suffisent à définir la tabulation. Ils sont décrits respectivement par les constantes `DOC_TYPTAB_*` et `DOC_LEADTAB_*`.

Syntaxe	DOC_SETCURRENTTAB <i>handle-WP, alignement, point-de-suite</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	alignement	INT(2)	type d'alignement de la tabulation
	point-de-suite	INT(2)	style de points de suite de la tabulation

Exemple :

```
; Tabulation : alignée à gauche + associée au point de suite -----  
DOC_SETCURRENTTAB HWP%, DOC_TYPTAB_LEFT%, DOC_LEADTAB_HYPHEN%
```

Voir aussi Constantes `DOC_TYPTAB_*`, `DOC_LEADTAB_*`, `DOC_GETCURRENTTAB`

Instruction **DOC_SETCURRENTTABLINE**

Indique que les taquets insérés par un double-clic souris sur la règle sont des taquets de tabulation verticale.

La largeur du trait est exprimée dans l'unité de mesure courante (système métrique ou anglo-saxon). La valeur d'un trait simple est 0, d'un trait double 1. La couleur est décrite par les constantes COL_ *% du langage NCL.

Syntaxe	DOC_SETCURRENTTABLINE <i>handle-WP, largeur, type, couleur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	largeur	INT(1)	largeur du trait vertical
	type	INT(2)	type du trait (simple ou double)
	couleur	INT(1)	couleur du trait

Exemple :

```
; Tabulation verticale formée d'un trait simple de largeur 5/10 mm (si le
système courant est le système métrique) et de couleur rouge
DOC_SETCURRENTTABLINE HWP%, 5, 0, COL_RED%
```

Voir aussi [DOC_GETCURRENTTABLINE](#), [DOC_DEFAULTMETRICS](#), [DOC_METRICS](#), [DOC_ISMETRICS%](#),
Constantes COL_ *% (Manuel de Programmation NCL)

Instruction **DOC_SETDEFTAB**

Positionne des taquets de tabulation dans la fenêtre NS WP.

Ces taquets de tabulation seront disposés selon un intervalle régulier égal à val-tab.

val-tab est exprimée dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_SETDEFTAB <i>handle-WP, val-tab</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	val-tab	INT(2)	distance entre les taquets de tabulation par défaut

Exemple :

; Positionne des taquets de tabulation tous les 5 mm dans la fenêtre de handle HWP% (système de mesure métrique)

```
DOC_SETDEFTAB HWP%, 50
```

Voir aussi DOC_GETDEFTAB%, DOC_DEFAULTTAB, DOC_DEFAULTMETRICS, DOC_ISMETRICS%

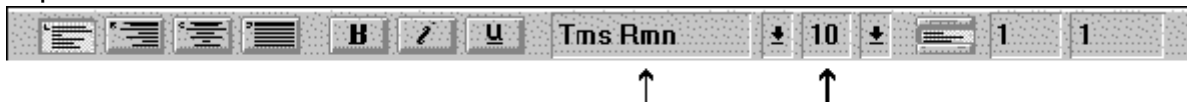
Instruction DOC_SETFONT

Change la police du texte sélectionné ou du point d'insertion si aucune sélection n'est en cours.

Le paramètre nom-taille-police est une chaîne de caractère contenant le nom et la taille de la nouvelle police séparés par une virgule.

Syntaxe	DOC_SETFONT <i>handle-WP, nom-taille-police</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nom-taille-police	CSTRING	nom et taille de la nouvelle police

Equivalent : Barre d'outils standard



Exemple :

```
DOC_SETFONT HWP%, "Tms Rmn,10"
```

Voir aussi [DOC_GETFONT\\$](#), [DOC_GETFONTNAMECONTROL](#), [DOC_GETFONTSIZECONTROL](#)

Instruction **DOC_SETFONTBACKCOLOR**

Définit la couleur d'arrière-plan (du texte sélectionné ou du point d'insertion si aucune sélection n'est en cours.

La couleur d'arrière-plan est le couleur de fond sur laquelle s'inscrira le texte.

couleur-arrière-plan vaut l'une des constantes COL_*.

Syntaxe	DOC_SETFONTBACKCOLOR <i>handle-WP, couleur-arrière-plan</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	couleur-arrière-plan	INT(1)	couleur d'arrière-plan

Exemple :

```
DOC_SETFONTBACKCOLOR HWP%, COL_BLUE%
```

Voir aussi [DOC_GETFONTBACKCOLOR%](#), *Constantes COL_** (*Manuel de Programmation NCL*)

Instruction **DOC_SETFONTCOLORS**

Définit les couleurs du texte sélectionné ou du point d'insertion si aucune sélection n'est en cours.

Les couleurs sont décrites dans le segment couleurs.

Syntaxe	DOC_SETFONTCOLORS <i>handle-WP, couleurs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	couleurs	DOC_TEXTCOLORS	couleurs du texte

Equivalent : Equivaut à l'appel successif des trois instructions :

- **DOC_SETFONTBACKCOLOR**
- **DOC_SETFONTFORECOLOR**
- **DOC_SETFONTLINECOLOR**

Exemple :

```
; Modification de la couleur du texte sélectionné
LOCAL DOC_TEXTCOLORS COLORS

MOVE COL_WHITE% TO COLORS.FORECOLOR
MOVE COL_BLACK% TO COLORS.BACKCOLOR
MOVE COL_WHITE% TO COLORS.LINECOLOR

DOC_SETFONTCOLORS HWP%, COLORS
```

Voir aussi Segment **DOC_TEXTCOLORS**, **DOC_GETFONTCOLORS**, **DOC_SETFONTBACKCOLOR**, **DOC_SETFONTFORECOLOR**, **DOC_SETFONTLINECOLOR**

Instruction **DOC_SETFONTFORECOLOR**

Définit la couleur d'avant-plan du texte sélectionné.

La couleur d'avant-plan est la couleur des lettres composant le texte.

couleur-avant-plan vaut l'une des constantes COL_ *%.

Syntaxe	DOC_SETFONTFORECOLOR <i>handle-WP, couleur-avant-plan</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	couleur-avant-plan	INT(1)	couleur d'avant-plan

Exemple :

DOC_SETFONTFORECOLOR HWP%, COL_BLUE%

*Voir aussi DOC_GETFONTFORECOLOR, Constantes COL_ *% (Manuel de Programmation NCL)*

Instruction **DOC_SETFONTINDEXCOLORS**

Affecte les couleurs du style de caractères défini par l'index index-style.
Les couleurs sont décrites dans le segment couleurs-style.

Syntaxe	DOC_SETFONTINDEXCOLORS <i>handle-WP, index-style, couleurs-style</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	index d'un style de caractères
	couleurs-style	DOC_TEXTCOLORS	couleurs d'un style de caractères

Exemple :

```
; Définition des couleurs du style d'index 1
LOCAL DOC_TEXTCOLORS COLORS

MOVE COL_WHITE% TO COLORS.FORECOLOR
MOVE COL_BLACK% TO COLORS.BACKCOLOR
MOVE COL_WHITE% TO COLORS.LINECOLOR

DOC_SETFONTINDEXCOLORS HWP%, 1, COLORS
```

Voir aussi Segment DOC_TEXTCOLORS, DOC_GETFONTINDEXCOLORS

Instruction **DOC_SETFONTINDEXFORMAT**

Affecte les attributs du style de caractères défini par l'index index-style.

attributs-style est une des constantes `DOC_FMT_*` ou une combinaison de constantes `DOC_FMT_*`.

Syntaxe	DOC_SETFONTINDEXFORMAT <i>handle-WP, index-style, attributs-style</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	index d'un style de caractères
	attributs-style	INT(1)	attributs du style de caractères (gras, italique...)

Exemple :

```
; Aucun attribut pour le style d'index 1
DOC_SETFONTINDEXFORMAT HWP%, 1, DOC_FMT_DEFAULT%
```

Voir aussi Constantes `DOC_FMT_*`, `DOC_GETFONTINDEXFORMAT%`

Instruction **DOC_SETFONTINDEXNAME**

Affecte la police du style de caractères défini par l'index index-style.

Syntaxe	DOC_SETFONTINDEXNAME <i>handle-WP, index-style, nom-police</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	index d'un style de caractères
	nom-police	CSTRING	police de caractères

Exemple :

```
; Détermination de la police du style d'index 1
DOC_SETFONTINDEXNAME HWP%, 1, "Tms Rmn"
```

Voir aussi [DOC_GETFONTINDEXNAME\\$](#)

Instruction **DOC_SETFONTINDEXSIZE**

Affecte la taille de la fonte du style de caractères défini par l'index `index-style`.

Syntaxe	DOC_SETFONTINDEXSIZE <i>handle-WP, index-style, taille-police</i>		
Paramètres	<code>handle-WP</code>	INT(4)	handle de la fenêtre NS-WP
	<code>index-style</code>	INT(2)	index d'un style de caractères
	<code>taille-police</code>	CSTRING	taille de la police de caractères

Exemple :

```
; Définition de la taille des caractères du style d'index 1
DOC_SETFONTINDEXSIZE HWP%, 1, 10
```

Voir aussi [DOC_GETFONTINDEXSIZE%](#)

Instruction **DOC_SETFONTINDEXUSERFORMAT**

Associe un entier au style de caractères défini par l'index index-style.

Syntaxe	DOC_SETFONTINDEXUSERFORMAT <i>handle-WP, index-style, user-format</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	index-style	INT(2)	index d'un style de caractères
	user-format	INT(2)	entier associé au style de caractères

Exemple :

```
; Association de l'entier 1 au style d'index 1
DOC_SETFONTINDEXUSERFORMAT HWP%, 1, 1
```

Voir aussi [DOC_GETFONTINDEXUSERFORMAT%](#), [DOC_GETPARAGRAPHGLOBALUSERINFO%](#),
[DOC_SETPARAGRAPHGLOBALUSERINFO](#), [DOC_GETPARAGRAPHLOCALUSERINFO%](#),
[DOC_SETPARAGRAPHLOCALUSERINFO](#)

Instruction **DOC_SETFONTLINECOLOR**

Définit la couleur utilisée, au point d'insertion si aucune sélection n'est en cours ou dans le texte sélectionné, pour souligner ou barrer.

couleur-ligne vaut l'une des constantes COL_ *%.

Syntaxe	DOC_SETFONTLINECOLOR <i>handle-WP, couleur-ligne</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	couleur-ligne	INT(1)	couleur du souligné ou du barré

Exemple :

```
DOC_SETFONTLINECOLOR HWP%, COL_BLUE%
```

Voir aussi [DOC_GETFONTLINECOLOR%](#), **Constantes COL_ *% (Manuel de Programmation NCL)**

Instruction DOC_SETFORMAT

Définit les attributs de caractère du texte sélectionné ou du point d'insertion si aucune sélection n'est en cours.

attributs-caract est une constante DOC_FMT_ *% ou une combinaison de constantes DOC_FMT_ *%.

Syntaxe	DOC_SETFORMAT <i>handle-WP, attributs-caract</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	attributs-caract	INT(1)	attributs (gras, italique...)

1. Les attributs existants du texte sont supprimés ; donner avec DOC_SETFORMAT l'unique attribut gras à un texte déjà en italique supprime l'italique.

2. Equivalents :

a) Clavier [Ctrl]+[B] | [I] | [S] | [U] selon attribut

b) Barre d'outils standard



Exemple 1 :

```
; Un unique attribut : gras
DOC_SETFORMAT HWP%, DOC_FMT_BOLD%
```

Exemple 2 :

```
; Inversion "gras"/"non gras" et "non gras"/"gras"
; Utilisation avec BXOR pour tenir compte des attributs précédents
DOC_SETFORMAT HWP%, DOC_FMT_BOLD% BXOR DOC_GETFORMAT%(HWP%)

; Attention : DOC_SETFORMAT se base sur le dernier caractère de la
sélection lorsqu'une sélection est en cours :
; si le dernier caractère est gras et les autres non, alors l'ensemble de
la sélection deviendra non gras
```

Voir aussi Constantes DOC_FMT_ *%, DOC_GETFORMAT%, DOC_MODIFYFORMAT, DOC_SETFONT

Instruction DOC_SETJUSTIF

Modifie l'alignement du paragraphe courant.

Le paramètre type-align est une constante DOC_JUS_*.%

Syntaxe	DOC_SETJUSTIF <i>handle-WP, type-align</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	type_align	INT(1)	alignement à appliquer

Equivalents :

1. Clavier [Ctrl]+[L] | [R] | [C] | [J] selon alignement
2. Barre d'outils standard



Exemple :

```
; Centrage du paragraphe
DOC_SETJUSTIF HWP%, DOC_JUS_CENTERED%
```

Voir aussi Constantes DOC_JUS_*.%, DOC_GETJUSTIF%

Instruction **DOC_SETLIKEPRINTER**

Active ou désactive le mode "comme à l'imprimante". Le mode "comme à l'imprimante" complète et affine le mode "page".

En mode "page", la taille des caractères est arrondie à l'écran, parce que la définition de l'écran est moins importante que celle de l'imprimante. Cet arrondi est cause d'écarts entre l'affichage et l'impression. Lorsque le mode "comme à l'imprimante" est activé, ces écarts sont corrigés.

- Si `indic-affichage` vaut `TRUE%`, le mode "comme à l'imprimante" est activé.
- Si `indic-affichage` vaut `FALSE%`, le mode "comme à l'imprimante" est désactivé.

Syntaxe	DOC_SETLIKEPRINTER <i>handle-WP, indic-affichage</i>		
Paramètres	<code>handle-WP</code>	<code>INT(4)</code>	handle de la fenêtre NS-WP
	<code>indic-affichage</code>	<code>INT(1)</code>	indicateur de passage en mode "comme à l'imprimante"

Le mode "comme à l'imprimante" ne s'applique pas aux modes "standard" et "brouillon".

Exemple :

```
IF NOT DOC_ISLIKEPRINTER%
DOC_SETLIKEPRINTER TRUE%
ENDIF
```

Voir aussi [DOC_ISLIKEPRINTER%](#), [DOC_SETMODE](#)

Instruction **DOC_SETMARGINS**

Modifie les marges de la page courante.

Les valeurs des marges sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Syntaxe	DOC_SETMARGINS <i>handle-WP, marge-gauche, marge-droite, marge haute, marge-basse</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	marge-gauche	INT(2)	marge gauche de page
	marge-droite	INT(2)	marge droite de page
	marge-haut	INT(2)	marge haute de page
	marge-bas	INT(2)	marge basse de page

Exemple :

```
LOCAL MARGEG%(2), MARGED%(2), MARGEH%(2), MARGEB%(2)
; Lecture des marges
DOC_GETMARGINS HWP%, MARGEG%, MARGED%, MARGEH%, MARGEB%

; Elargissement double des marges
DOC_SETMARGINS HWP%, 2*MARGEG%, 2*MARGED%, 2*MARGEH%, 2*MARGEB%
```

Voir aussi [DOC_GETMARGINS](#), [DOC_GETPAGEDIM](#), [DOC_METRICS](#), [DOC_DEFAULTMETRICS](#), [DOC_ISMETRICS%](#)

Instruction **DOC_SETMAXPARAWIDTH**

Définit le nombre maximum de caractères d'une ligne en mode "caractère" dans le document courant.

Si nb-caract = 0, il n'y a pas de limite.

Syntaxe	DOC_SETMAXPARAWIDTH <i>handle-WP, nb-caract</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	nb-caract	INT(2)	nombre maximum de caractères d'une ligne

Exemple :

```
IF DOC_ISMETRICS% (HWP%) = DOC_CHARS%
DOC_SETMAXPARAWIDTH HWP%, 80
ENDIF
```

Voir aussi Constantes DOC LO*%, DOC CHARS%, DOC DEFAULTMAXPARAWIDTH,
DOC GETMAXPARAWIDTH%, DOC DEFAULTMETRICS, DOC METRICS, DOC ISMETRICS%,
ESCLOAD%, ESCPRINT%

Instruction **DOC_SETMODE**

Détermine le mode d'affichage du document (standard, brouillon, page).
mode-affichage est une des constantes **DOC_MODE_***.

Syntaxe	DOC_SETMODE <i>handle-WP, mode-affichage</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	mode-affichage	INT(1)	mode d'affichage de la fenêtre NS-WP

Exemple :

```
; Affichage en mode brouillon  
DOC_SETMODE HWP%, DOC_MODE_DRAFT%
```

Voir aussi Constantes **DOC_MODE_***, **DOC_GETMODE%**, **DOC_UPDATE**

Instruction **DOC_SETPAGE**

Positionne le curseur au début de la page de numéro num-page.

Syntaxe	DOC_SETPAGE <i>handle-WP, num-page</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	num-page	INT(2)	numéro de page

Exemple :

```
; Positionnement au début de la page courante  
DOC_SETPAGE HWP%, DOC_GETPAGE% (HWP%)
```

Voir aussi [DOC_GETPAGE%](#)

Instruction **DOC_SETPARAGRAPHGLOBALUSERINFO**

Associe un entier au paragraphe. Cet entier est de type global.

Il se propage aux paragraphes issus de celui-ci, que ce soit par un retour-chariot ou au moyen de l'instruction **DOC_SPLITPARAGRAPH**.

Syntaxe	DOC_SETPARAGRAPHGLOBALUSERINFO <i>handle-WP, entier-global</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	entier-global	INT(2)	entier associé au paragraphe

Cette instruction permet par exemple d'associer à un paragraphe un style identifié par un numéro. Ce style sera transmis aux paragraphes fils.

Exemple :

```
DOC_SETPARAGRAPHGLOBALUSERINFO HWP%, INDEX_STYLE%
```

Voir aussi **DOC_GETPARAGRAPHGLOBALUSERINFO%**, **DOC_GETPARAGRAPHLOCALUSERINFO%**, **DOC_SETPARAGRAPHLOCALUSERINFO**, **DOC_SPLITPARAGRAPH**

Instruction **DOC_SETPARAGRAPHKEEPTOGETHER**

Lie les lignes du paragraphe courant si lignes-solidaires vaut TRUE%, supprime le lien si lignes-solidaires vaut FALSE%.

Aucun saut de page n'est possible entre des lignes solidaires.

Syntaxe	DOC_SETPARAGRAPHKEEPTOGETHER <i>handle-WP, lignes-solidaires</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	lignes-solidaires	INT(1)	indicateur de lien entre les lignes

Exemple :

```
; Lie les lignes du paragraphe courant si elles ne sont pas déjà solidaires
IF DOC_GETPARAGRAPHKEEPTOGETHER% (HWP%) = FALSE%
DOC_SETPARAGRAPHKEEPTOGETHER HWP%, TRUE%
ENDIF
```

Voir aussi **DOC_GETPARAGRAPHKEEPWITHNEXT%**

Instruction **DOC_SETPARAGRAPHKEEPWITHNEXT**

Lie le paragraphe courant avec son suivant si paragraphes-solidaires vaut TRUE%,
supprime le lien si paragraphes-solidaires vaut FALSE%.

Aucun saut de page n'est possible entre deux paragraphes solidaires.

Syntaxe	DOC_SETPARAGRAPHKEEPWITHNEXT <i>handle-WP, paragraphes-solidaires</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	paragraphes-solidaires	INT(1)	indicateur de lien entre les paragraphes

Exemple :

```
; Lie le paragraphe courant avec son suivant s'ils ne sont pas déjà  
solidaires  
IF DOC_GETPARAGRAPHKEEPWITHNEXT% (HWP%) = FALSE%  
DOC_SETPARAGRAPHKEEPWITHNEXT HWP%, TRUE%  
ENDIF
```

Voir aussi **DOC_GETPARAGRAPHKEEPWITHNEXT%**

Instruction **DOC_SETPARAGRAPHLOCALUSERINFO**

Associe un entier au paragraphe. Cet entier est de type local.

Il ne se propage pas aux paragraphes issus de celui-ci.

Syntaxe	DOC_SETPARAGRAPHLOCALUSERINFO <i>handle-WP, entier-local</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	entier-local	INT(2)	entier associé au paragraphe

Cette instruction permet par exemple de définir dans le texte des entrées de table des matières identifiées par un numéro.

Exemple :

```
DOC_SETPARAGRAPHLOCALUSERINFO HWP%, INDEX%
```

Voir aussi [DOC_GETPARAGRAPHGLOBALUSERINFO%](#), [DOC_GETPARAGRAPHLOCALUSERINFO%](#), [DOC_SETPARAGRAPHLOCALUSERINFO](#)

Instruction **DOC_SETPARAGRAPHSINFO**

Modifie les caractéristiques d'un ou de plusieurs paragraphes.

Cette modification porte sur plusieurs paragraphes dont les handles sont passés dans `handle-prem-parag` et `handle-dern-parag`, ou sur un seul paragraphe en répétant son handle dans les deux variables.

`type-align` vaut l'une des constantes `DOC_JUS_*`.

Syntaxe	DOC_SETPARAGRAPHSINFO <i>handle-WP, handle-prem-parag, handle-dern-parag, type-align, descript-tabs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-prem-parag	INT(4)	handle du premier paragraphe
	handle-dern-parag	INT(4)	handle du dernier paragraphe
	type-align	INT(1)	alignement à appliquer aux paragraphes
	descript-tabs	DOC_TABS	descriptif des tabulations des paragraphes

1. Contrairement à `DOC_GETPARAGRAPHINFO`, noter le "S" de la syntaxe de `DOC_SETPARAGRAPHSINFO`.
2. `type-align` peut également être modifié par `DOC_SETJUSTIF`, les informations de `descript-tabs` peuvent également être modifiées par `DOC_SETTABS` ou par `DOC_SETXTABS`.

Exemple :

```
LOCAL SIZE%(2), FIRSTLINE%(2), NBLINES%(2), JUSTIF%(1)
LOCAL DOC_TABS MYTABS

; Lecture des infos du paragraphe courant
DOC_GETPARAGRAPHINFO HWP%, DOC_CURRENTPARAGRAPH%(HWP%), SIZE%, FIRSTLINE%,
NBLINES%, JUSTIF%, MYTABS

; Affectation de l'alignement et des tabulations du paragraphe courant à la
totalité du document
DOC_SETPARAGRAPHSINFO HWP%, DOC_FIRSTPARAGRAPH%(HWP%),
DOC_LASTPARAGRAPH%(HWP%), JUSTIF%, MYTABS
```

Voir aussi Constantes `DOC_JUS_*`, **Segment** `DOC_TABS`, `DOC_GETPARAGRAPHINFO`, `DOC_GETTABS`, `DOC_SETTABS`, `DOC_SETXTABS`, `DOC_SETJUSTIF`

Instruction **DOC_SETPOS**

Positionne le curseur dans un paragraphe.

Syntaxe	DOC_SETPOS <i>handle-WP, handle-parag, pos-curseur</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe devant contenir le curseur
	pos-curseur	INT(2)	nombre de caractères précédant le curseur dans ce paragraphe

Equivalent : Clavier -> En déplaçant le curseur

Exemple :

```
; Positionnement en début de document
DOC_SETPOS HWP%, DOC_FIRSTPARAGRAPH%(HWP%), 0
```

Voir aussi [DOC_GETPOS](#), [DOC_SETSELECTION](#)

Instruction **DOC_SETSECTION**

Positionne le curseur au début de la section de numéro num-section.

Syntaxe	DOC_SETSECTION <i>handle-WP, num-section</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	num-section	INT(2)	numéro de section

Exemple :

```
; Positionnement au début de la section courante  
DOC_SETSECTION HWP%, DOC_GETSECTION% (HWP%)
```

Voir aussi [DOC_SECTION](#), [DOC_ISSECTION%](#), [DOC_GETNBSECTIONS%](#), [DOC_GETSECTION%](#),
[DOC_GETSECTIONJUSTIF%](#), [DOC_SETSECTIONJUSTIF](#)

Instruction **DOC_SETSECTIONJUSTIF**

Aligne le texte verticalement pour toutes les pages de la section courante. Cet alignement n'est visible qu'à l'impression du document.

type-align vaut l'une des constantes suivantes :

- **DOC_JUS_UPPER%** le texte est aligné sur le haut de la page
- **DOC_JUS_LOWER%** le texte est aligné sur le bas de la page
- **DOC_JUS_CENTERED%** le texte est centré entre le haut et le bas de la page
- **DOC_JUS_FULL%** le texte est justifié entre le haut et le bas de la page

Syntaxe	DOC_SETSECTIONJUSTIF <i>handle-WP, type-align</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	type-align	INT(1)	type d'alignement

1. Ces alignements ne sont significatifs que pour les pages dont la rupture est forcée, les variations de mise en page verticale y étant possibles.
2. Par défaut, le texte est aligné sur le haut de la page.

Exemple :

```
DOC_SETSECTIONJUSTIF HWP%, DOC_JUS_FULL%
```

Voir aussi Constantes **DOC_JUS_***, **DOC_SECTION**, **DOC_ISSECTION%**, **DOC_GETNBSECTIONS%**, **DOC_GETSECTION%**, **DOC_SETSECTION**, **DOC_GETSECTIONJUSTIF%**

Instruction **DOC_SETSELECTION**

Sélectionne une partie du document.

pos-deb et pos-fin sont des positions relatives au paragraphe. Ces positions sont exprimées en nombre de caractères, le premier caractère ayant la position 0, le deuxième la position 1, et ainsi de suite.

Syntaxe	DOC_SETSELECTION <i>handle-WP, handle-parag-deb, pos-deb, handle-parag-fin, pos-fin</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag-deb	INT(4)	handle du paragraphe où débute la sélection
	pos-deb	INT(2)	position du caractère sur lequel débute la sélection
	handle-parag-fin	INT(4)	handle du paragraphe où finit la sélection
	pos-fin	INT(2)	position du caractère sur lequel finit la sélection

Equivalent : Clavier [Shift]+déplacement curseur

Exemple :

```
; Sélection du premier paragraphe
DOC_SETSELECTION HWP%, DOC_FIRSTPARAGRAPH%(HWP%), 0 ,
DOC_NEXTPARAGRAPH%(HWP%, DOC_FIRSTPARAGRAPH%(HWP%)), 0
```

Voir aussi [DOC_GETSELECTION](#), [DOC_GETSELECTIONTEXT\\$](#)

Instruction **DOC_SETTABLINES**

Remplace les tabulations verticales du paragraphe courant ou des paragraphes sélectionnés par de nouvelles tabulations verticales.

Les nouvelles tabulations verticales sont décrites dans le segment `descript-tabs-vert`.

Syntaxe	DOC_SETTABLINES <i>handle-WP, descript-tabs-vert</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-tabs-vert	DOC_TABLINES	descriptif des tabulations verticales

Exemple :

```
; Association d'une tabulation verticale au paragraphe courant
LOCAL DOC_TABLINE LINE
LOCAL DOC_TABLINES MYTABLINES

move 400 to line.xpos
move 5 to line.width
move 0 to line.kind
move col_red% to line.color

move 1 to MYtablines.nbtablines
move line to MYtablines.tabs[0]

doc_settablines hwp%, MYtablines
```

Voir aussi Segment [DOC_TABLINES](#), [DOC_GETTABLINES](#)

Instruction DOC_SETTABS

Remplace les retraits et les taquets de tabulation du paragraphe courant ou des paragraphes sélectionnés par de nouveaux retraits et de nouvelles tabulations. Les nouveaux retraits et les nouveaux taquets de tabulations sont décrits dans `descript-tabs`.

Syntaxe	DOC_SETTABS <i>handle-WP, descript-tabs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-tabs	DOC_TABS	descriptif des tabulations du paragraphe

1. Pour changer les retraits et les taquets de tabulation d'un paragraphe, il est conseillé de procéder en deux étapes :
 - a) Récupération dans `descript-tabs` des retraits et des taquets de tabulation actuels au moyen de l'instruction `DOC_GETTABS`.
 - b) Mise à jour de `descript-tabs`, puis lancement de `DOC_SETTABS`.
2. `DOC_SETTABS` est inopérante si les retraits ne sont pas spécifiés dans `descript-tabs`.
3. Basée sur le segment `DOC_TABS`, `DOC_SETTABS` gère moins d'informations que `DOC_SETXTABS`. Elle est conservée dans l'A.P.I. par souci de compatibilité ascendante.
4. Equivalent : Règle -> En positionnant les retraits et les taquets de tabulation avec la souris

Exemple :

```
LOCAL DOC_TABS MYTABS

; Lecture des tabulations et retraits du paragraphe courant
DOC_GETTABS HWP%, MYTABS

; Mise à zéro du retrait gauche du paragraphe courant
MOVE 0 TO MYTABS.LEFTINDENT
DOC_SETTABS HWP%, MYTABS
```

Voir aussi Constante `DOC_MAX_TABULATIONS%`, Segment `DOC_TABS`, `DOC_GETTABS`, `DOC_SETXTABS`, `DOC_SETPARAGRAPHSINFO`

Instruction **DOC_SETTAG**

Insère une borne à l'emplacement indiqué dans handle-parag et pos-caract.

Syntaxe	DOC_SETTAG <i>handle-WP, handle-parag, pos-caract</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle du paragraphe
	pos-caract	INT(2)	position dans le paragraphe

Exemple :

```
; Insertion d'une borne à gauche du curseur  
DOC_GETPOS HWP%, HPARA%, OFFSET%  
DOC_SETTAG HWP%, HPARA%, OFFSET%
```

Voir aussi [DOC_CLEAR](#)[TAG](#), [DOC_ISTAG](#)[%](#)

Instruction **DOC_SETUSERFORMAT**

Associe un entier à la police de caractères courante.

Syntaxe	DOC_SETUSERFORMAT <i>handle-WP, entier-associé</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	entier-associé	INT(4)	entier à associer à la police courante

Exemple :

```
DOC_SETUSERFORMAT HWP%, 1
```

Voir ***aussi*** **DOC_GETUSERFORMAT%**, **DOC_GETUSERFORMATAT%**,
DOC_GETPARAGRAPHGLOBALUSERINFO%, **DOC_SETPARAGRAPHGLOBALUSERINFO%**,
DOC_GETPARAGRAPHLOCALUSERINFO%, **DOC_SETPARAGRAPHLOCALUSERINFO%**

Instruction **DOC_SETWIDOWCONTROL**

Gère le contrôle des lignes veuves et orphelines.

- Si `indic-vo` vaut `TRUE%`, le contrôle des lignes veuves et orphelines est activé pour tout le document.
- Si `indic-vo` vaut `FALSE%`, le contrôle des lignes veuves et orphelines est désactivé pour tout le document.

Syntaxe	DOC_SETWIDOWCONTROL <i>handle-WP, indic-vo</i>		
Paramètres	<code>handle-WP</code>	INT(4)	handle de la fenêtre NS-WP
	<code>indic-vo</code>	INT(1)	indicateur de contrôle des lignes veuves et orphelines

Exemple :

```
IF DOC_GETWIDOWCONTROL% (HWP%) = FALSE%  
DOC_SETWIDOWCONTROL HWP%, TRUE%  
ENDIF
```

Voir aussi [DOC_DEFAULTWIDOWCONTROL](#), [DOC_GETWIDOWCONTROL%](#)

Instruction DOC_SETWINDOW

Précise la partie du document à afficher à l'écran.

Ce peut être le corps du document, l'en-tête ou le pied de page associés à la première page d'une section, l'en-tête ou le pied de page associés aux autres pages. partie-document est une constante DOC_WIN_ *%.

Syntaxe	DOC_SETWINDOW <i>handle-WP, partie-document</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	partie-document	INT(1)	partie du document à afficher

1. Le comportement de NS-WP dépend de la partie visualisée.
2. Par défaut, c'est le corps du document qui est affiché.
3. Si la partie du document n'existe pas, elle est créée.
4. Equivalents :
 - a) Clavier [Ctrl]+[F] | [H] | [K] selon la partie
 - b) Barre d'outils standard



Exemple :

```
; Affichage de l'en-tête de la première page du document
DOC_SETWINDOW HWP%, DOC_WIN_FIRSTHEADER%
```

Voir aussi Constantes DOC_WIN_ *%, DOC_GETWINDOW%, DOC_ISWINDOW%,
DOC_DISPOSEWINDOW

Instruction DOC_SETXTABS

Remplace les retraits et les taquets de tabulation du paragraphe courant ou des paragraphes sélectionnés par de nouveaux retraits et de nouvelles tabulations. Les nouveaux retraits et taquets de tabulations sont décrits dans `descript-tabs`.

Syntaxe	DOC_SETXTABS <i>handle-WP, descript-tabs</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	descript-tabs	DOC_XTABS	descriptif des retraits et des taquets de tabulation

1. Pour changer les retraits et les taquets de tabulation d'un paragraphe, il est conseillé de procéder en deux étapes :
 - a) Récupération dans `descript-tabs` des retraits et des taquets de tabulation actuels au moyen de l'instruction `DOC_GETXTABS`.
 - b) Mise à jour de `descript-tabs`, puis lancement de `DOC_SETXTABS`.
2. `DOC_SETXTABS` est inopérante si les retraits ne sont pas spécifiés dans `descript-tabs`.
3. Basée sur le segment `DOC_XTABS`, `DOC_SETXTABS` gère plus d'informations que `DOC_SETTABS`, qui n'est conservée dans l'A.P.I. que par souci de compatibilité ascendante.
4. Equivalent : Règle -> En positionnant les retraits et les taquets de tabulation avec la souris

Exemple :

```
LOCAL DOC_XTABS MYXTABS

; Lecture des tabulations et retraits du paragraphe courant
DOC_GETXTABS HWP%, MYXTABS

; Mise à zéro du retrait gauche du paragraphe courant
MOVE 0 TO MYXTABS.LEFTINDENT

; Association du style de point de suite ----- au premier taquet de
; tabulation
MOVE DOC_LEADTAB_HYPHEN% TO MYXTABS.TABLEAD[0]

; Modification
DOC_SETXTABS HWP%, MYXTABS
```

Voir aussi Constante `DOC_MAX_TABULATIONS%`, Segment `DOC_XTABS`, `DOC_GETXTABS`, `DOC_SETTABS`, `DOC_SETPARAGRAPHSINFO`

Instruction **DOC_SHOWRULERBAR**

Gère l'affichage de la règle dans la fenêtre NS-WP.

- Si indic-affichage vaut TRUE%, la règle est affichée juste au dessus de l'aire client.
- Si indic-affichage vaut FALSE%, la règle n'est plus affichée.

Syntaxe	DOC_SHOWRULERBAR <i>handle-WP, indic-affichage</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-affichage	INT(1)	indicateur d'affichage de la règle

DOC_RULERBAR affiche également la règle dans la fenêtre NS-WP.

Exemple :

```
; Inversion de l'état de la règle  
DOC_SHOWRULERBAR HWP%, NOT DOC_ISRULERBARSHOWN% (HWP%)
```

Voir aussi DOC_RULERBAR, DOC_ISRULERBARSHOWN%

Instruction **DOC_SHOWTOOLBOX**

Gère l'affichage de la barre d'outils standard dans la fenêtre NS-WP.

- Si indic-affichage vaut TRUE%, la barre d'outils standard est affichée.
- Si indic-affichage vaut FALSE%, la barre d'outils standard n'est plus affichée.

Syntaxe	DOC_SHOWTOOLBOX <i>handle-WP, indic-affichage</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	indic-affichage	INT(1)	indicateur d'affichage de la barre d'outils

DOC_RULERBAR affiche également la barre d'outils standard dans la fenêtre NS-WP.

Exemple :

```
; Inversion de l'état de la barre d'outils standard  
DOC_SHOWTOOLBOX HWP%, NOT DOC_ISTOOLBOXSHOWN% (HWP%)
```

Voir aussi DOC_RULERBAR, DOC_ISTOOLBOXSHOWN%, DOC_ISRULERBAR%

Instruction DOC_SPLITPARAGRAPH

Insère une marque de fin de paragraphe à la position spécifiée.

pos-caract est exprimée en nombre de caractères précédant la position.

Syntaxe	DOC_SPLITPARAGRAPH <i>handle-WP, handle-parag, pos-caract</i>		
Paramètres	handle-WP	INT(4)	handle de la fenêtre NS-WP
	handle-parag	INT(4)	handle d'un paragraphe
	pos-caract	INT(2)	position dans le paragraphe

1. Le nouveau paragraphe hérite du format du paragraphe initial, à l'exception de l'entier de type local associé au paragraphe par DOC_SETPARAGRAPHLOCALUSERINFO et de l'attribut "saut de page" ou "saut de section".
2. Si handle-parag vaut 0, la marque de fin de paragraphe est insérée à la position du curseur. Le curseur est ensuite positionné sur le nouveau paragraphe.
3. Equivalent : Clavier -> retour chariot

Exemple :

```
; Coupe le paragraphe en deux à gauche du curseur
DOC_SPLITPARAGRAPH HWP%, 0, 0
```

Instruction DOC_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, la frappe clavier et le formatage.

Syntaxe	DOC_UNDO <i>handle-WP</i>		
Paramètres	<i>handle-WP</i>	INT(4)	<i>handle de la fenêtre NS-WP</i>

1. Une action annulée par DOC_UNDO peut être rejouée avec DOC_REDO.
2. L'ensemble des actions effectuées sur un écran bloqué par DOC_LAYOUT ou par DOC_UPDATE est vu comme une seule action. Un appel à DOC_UNDO, après avoir débloqué l'écran, annule donc l'intégralité de ces actions.

Exemple :

```
IF DOC_ISUNDO% (HWP%)  
DOC_UNDO HWP%  
ENDIF
```

Voir aussi DOC_ISUNDO%, DOC_REDO, DOC_CUT, DOC_COPY, DOC_PASTE, DOC_DELETE

Instruction **DOC_UPDATE**

Bloque ou débloquent la mise à jour à l'écran.

- Si `indic-action` vaut `FALSE%`, l'écran n'est plus mis à jour.
- Si `indic-action` vaut `TRUE%`, toute modification dans le texte est immédiatement répercutée à l'écran.

Syntaxe	DOC_UPDATE <i>handle-WP, indic-action</i>		
Paramètres	<code>handle-WP</code>	<code>INT(4)</code>	handle de la fenêtre NS-WP
	<code>indic-action</code>	<code>INT(1)</code>	indicateur d'action à effectuer

1. Afin d'éviter de pénibles effets de clignotement à l'écran, le blocage de la mise à jour de la fenêtre NS-WP doit être utilisé chaque fois que des modifications multiples sont effectuées de façon automatique. Le déblocage de la mise à jour n'est effectué qu'une fois les modifications terminées.
2. "`DOC_UPDATE HWP%, TRUE%`" est rigoureusement équivalent à "`UPDATE HWP%`".
3. "`DOC_UPDATE HWP%, FALSE%`" est rigoureusement équivalent à "`NOUPDATE HWP%`".
4. L'instruction `DOC_LAYOUT` permet aussi de ne pas mettre à jour l'écran.

Exemple :

```
; Arrêt de la mise à jour de la fenêtre NS-WP
DOC_UPDATE HWP%, FALSE%

; Modifications du texte (restent invisibles)
...

; Reprise de la mise à jour de la fenêtre NS-WP rendant visibles les
modifications faites
DOC_UPDATE HWP%, TRUE%
```

Voir aussi `DOC_LAYOUT`, `DOC_SETMODE`, `UPDATE`, `NOUPDATE` (*Manuel de Programmation NCL*)

Segments

Segment DOC_BORDER

Description d'un encadrement.

L'espace laissé entre le cadre et le texte encadré est exprimé dans l'unité de mesure courante (système métrique ou anglo-saxon).

Le motif du fond du cadre est défini au moyen des constantes PATSYM_%% de la librairie NSGRAPH. Les couleurs du motif choisi sont définies avec les constantes COL_%% du langage NCL.

Les six éléments suivants sont décrits dans les segments de type LINE_BORDER :

- Les bords gauche, droit, haut et bas du cadre.
- La ligne utilisée pour séparer des paragraphes disposés dans un même cadre.
- L'ombre associée au cadre.

Ces éléments sont identifiés par les constantes DOC_LINE_%%.

Champs	Type	Description
DWSPACE	INT(2)	espace laissé entre les bords du cadre et le texte encadré
PATTERN	INT(2)	motif du fond du cadre
FORECOLOR	INT(1)	couleur d'avant-plan du fond du cadre
BACKCOLOR	INT(1)	couleur d'arrière-plan du fond du cadre
RESERVED	INT(2)	réservé (usage interne)
LINE	LINE_BORDER [6]	caractéristiques des éléments constituant le cadre

Voir aussi Constantes DOC_LINE_%%, Segment LINE_BORDER, DOC_GETBORDER, DOC_SETBORDER, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%, Constantes PATSYM_%% (Librairie graphique NSGRAPH), Constantes COL_%% (Manuel de Programmation NCL)

Segment DOC_FMTLINE

Description d'une ligne de texte en mode "caractère".

La ligne est formatée à l'aide d'espaces. Les tabulations sont remplacées par des espaces.

Champs	Type	Description
COUNT	INT(2)	nombre de caractères
RESERVED	INT(2)	réservé (usage interne)
CH	CHAR [256]	caractères de la ligne (plus un zéro de fin)
INDEX	INT (2) [256]	index des polices associées

Voir aussi DOC_GETFMTLINE

Segment DOC_FONT

Description du style pris en compte par défaut.

Le champ FORMAT est une des constantes DOC_FMT_ *% ou une combinaison de constantes DOC_FMT_ *%.

Champs	Type	Description
NAME	STRING(31)	police
SIZE	INT(2)	taille des caractères
FORMAT	INT(2)	attributs
COLOR	DOC_TEXTCOLOR	couleurs
USERFORMAT	INT(4)	entier associé au style

Voir aussi Constantes DOC_FMT_ *%, Segment DOC_TEXTCOLORS, DOC_DEFAULTFONT

Segment DOC_LINEDATA

Description d'une ligne de texte.

Champs	Type	Description
NB	INT(2)	nombre de caractères
RESERVED1	INT(2)	réservé (usage interne)
CHARS	CHAR [256]	caractères de la ligne
FONTINDEXS	INT (2) [256]	index des polices associées
PAGEBREAK	INT (1)	TRUE% si la ligne comporte une rupture de page, FALSE% sinon
BRESERVED	INT(1)	réservé (usage interne)
RESERVED2	INT(2)	réservé (usage interne)

Voir aussi DOC_GETLINE

Segment DOC_M_MARKER

Description d'un marqueur.

Ce segment permet de traiter les actions de type DOC_M_DELETEMARKER_EVENT et DOC_M_PASTEMARKER_EVENT.

Il est mis à jour :

- Après l'appel à DOC_MAPUSERMSG.
- Lors de la tentative de suppression d'un marqueur (action de type DOC_M_DELETEMARKER).
- Lors de la copie du presse-papiers dans un document, pour chaque marqueur contenu dans le presse-papiers (action de type DOC_M_PASTEMARKER).

Champs	Type	Description
ACCLASS	INT(4)	handle du dictionnaire associé au marqueur
MARKEr_HANDLE	INT(4)	handle du marqueur
TEXT	CSTRING	texte du marqueur
RESERVED	INT(2)	réservé (usage interne)
SIZE	INT(2)	taille en octets des données associées au marqueur
DATA	INT(4)	adresse mémoire des données associées au marqueur

Exemple :

```
; Événement INIT de la fenêtre NS-WP
; Activation de l'événement USER10 en cas de tentative de suppression d'un
marqueur
DOC_MAPUSERMSG HWP%, HWP%, DOC_M_DELETEMARKER_EVENT, DOC_USER_MESSAGE_10

; Événement USER10
; Confirmation de la suppression d'un marqueur
LOCAL CLASS$, DATA$
IF PARAM1% = DOC_M_DELETEMARKER_EVENT

; Lecture du nom du dictionnaire et des données
MOVE DOC_M_GETCLASSNAME$ (DOC_M_MARKER(PARAM34%).ACCLASS TO CLASS$
DOC_M_GETMARKERDATA DOC_M_MARKER(PARAM34%).ACCLASS,
DOC_M_MARKER(PARAM34%).MARKEr_HANDLE, @DATA$, SIZEOF DATA$ - 1

; Demande d'autorisation de suppression
; Lorsque l'événement retourne TRUE%, la suppression a lieu
; Lorsque l'événement retourne FALSE%, le marqueur n'est pas supprimé
IF ASK2%("Suppression marqueur, dictionnaire :" && CLASS$ && "Données :" &&
DATA$, '(' & DOC_M_MARKER(PARAM34%).MARKEr_HANDLE & ')') &&
DOC_M_MARKER(PARAM34%).TEXT) = YES%
RETURN TRUE%
ELSE
RETURN FALSE%
```

```
ENDIF  
ENDIF
```

Voir aussi Constantes DOC * EVENT, DOC USER MESSAGE, DOC MAPUSERMSG

Segment DOC_MEMFILE

Description du buffer de sauvegarde.

Champs	Type	Description
TYPE	INT(4)	réservé à un usage futur
MAXSIZE	INT(4)	taille du buffer
SIZE	INT(4)	taille réellement utilisée lors de la sauvegarde
BUFF	INT(4)	adresse du buffer

1. TYPE doit être initialisé à 0.
2. S'il n'y a pas d'erreur lors de la sauvegarde, DOC_SAVE met dans SIZE la taille réellement utilisée. Cette taille ne comprend pas le dernier caractère (caractère zéro).
3. La zone du buffer non utilisée est remplie avec des zéros.
4. Les tailles sont exprimées en octets.

Voir aussi DOC_OPEN, DOC_SAVE

Segment DOC_RULEINFO

Description d'une modification d'état du document.

Ce segment permet de traiter les modifications de type DOC_RULER_EVENT. Il est mis à jour :

- Après l'appel à DOC_MAPUSERMSG.
- A la suite de chaque modification d'état du document.

Champs	Type	Description
RULECHANGES	INT(4)	identifie l'élément de la fenêtre NS-WP qui a été modifié. Vaut l'une des constantes DOC_RCHG_*. La modification est contenue dans le champ désigné par RULECHANGES.
CURSORCOLUMN	INT(2)	nouveau numéro de colonne
CURSORLINE	INT(2)	nouveau numéro de ligne
CURRENTPAGE	INT(2)	nouveau numéro de page
RESERVED1	INT(2)	réservé (usage interne)
FONTNAME	STRING(31)	nouveau nom de police
FONTSIZE	INT(2)	nouvelle taille d'une police
FORMAT	INT(2)	nouveau format d'une police (gras, italique...). Vaut l'une des constantes DOC_FMT_ *%.
COLORS	DOC_TEXTCOLORS	nouvelles couleurs d'une police
USERFORMAT	INT(4)	nouveau format-utilisateur d'une police
PARAINFO	INT(4)	nouvel attribut de paragraphe. Vaut l'une des constantes DOC_PARA_ *%.
LOCALUSERPARAINFO	INT(2)	nouvel attribut de paragraphe utilisateur, de type local
GLOBALUSERPARAINFO	INT(2)	nouvel attribut de paragraphe utilisateur, de type global

JUSTIFICATION	INT(2)	nouvel alignement de paragraphe. Vaut l'une des constantes <u>DOC_JUS_*</u> %.
RESERVED2	INT(2)	réservé (usage interne)
TABULATIONS	DOC_XTABS	nouveaux retraits ou nouveaux taquets de tabulation
CONTEXT	INT(2)	nouvelle partie du document affichée (en-tête, corps du texte ou bas de page). Vaut l'une des constantes <u>DOC_WIN_*</u> %.
CURRENTSECTION	INT(2)	ouveau numéro de section

Voir aussi Constantes DOC_FMT_* %, DOC_JUS_* %, DOC_WIN_* %, DOC_PARA_* %, **Constantes** DOC * EVENT, DOC_RCHG_*, **Segments** DOC_TEXTCOLORS, DOC_XTABS, DOC_MAPUSERMSG

Segment DOC_SCAN_CTXT

Description d'un élément du document (paragraphe, tabulations, index de la police...).

Ce segment est mis à jour à la sortie des fonctions DOC_SCANFIRST% et DOC_SCANNEXT%.

La constante DOC_SCN_*% retournée par DOC_SCANFIRST% ou par DOC_SCANNEXT% est copiée dans ITYPE.

Les autres champs du segment sont remplis en fonction de la valeur de ITYPE.

- Si ITYPE vaut DOC_SCN_TEXT% alors :

Le buffer utilisateur contient la suite de caractères et SSIZE contient le nombre de caractères.

- Si ITYPE vaut DOC_SCN_SPACE% alors :

Le buffer utilisateur contient un espace et SSIZE vaut 1.

- Si ITYPE vaut DOC_SCN_TAB% alors :

Le buffer utilisateur contient la marque de tabulation et SSIZE vaut 1.

- Si ITYPE vaut DOC_SCN_FONT% alors :

Le champ SPARAM1 contient l'index de la police.

- Si ITYPE vaut DOC_SCN_BEGMARK% ou DOC_SCN_ENDMARK% alors :

Le champ SPARAM1 indique le dictionnaire attaché au document qui contient l'entrée de données associée au marqueur. Cette indication est fournie sous la forme d'un numéro qui reflète l'ordre de liaison du dictionnaire et du document (par exemple 1 pour premier dictionnaire attaché au document).

Si le marqueur n'est pas associé à un dictionnaire ou si ce dictionnaire n'est pas attaché au document, le champ SPARAM1 contient 0.

Le champ SPARAM2 contient le numéro de l'entrée de données associée au marqueur si elle existe, 0 sinon.

Champs	Type	Description
RSRVD0	INT(2)	réservé (usage interne)
ITYPE	INT(2)	type d'élément trouvé
HPARA	INT(4)	handle du paragraphe dans lequel est situé l'élément
OFSBEG	INT(2)	position dans le paragraphe du début de l'élément
OFSEND	INT(2)	position dans le paragraphe du début de l'élément suivant
SSIZE	INT(2)	taille du texte stocké dans le buffer
SMAXSIZE	INT(2)	taille du buffer fourni par l'utilisateur
PBUFF	INT(4)	adresse du buffer

SPARAM1	INT(2)	si l'élément est une police, index de la police ou si l'élément est un marqueur associé à une entrée de données d'un dictionnaire, numéro du dictionnaire
SPARAM2	INT(2)	si l'élément est un marqueur associé à une entrée de données d'un dictionnaire, numéro de l'entrée de données
RSRVD1	INT (2) [20]	réservé (usage interne)

Voir aussi Constantes `DOC SCN *%`, `DOC SCANFIRST%`, `DOC SCANNEXT%`

Segment DOC_TABLINE

Description d'une tabulation verticale.

L'abscisse et la largeur du trait sont exprimées dans l'unité de mesure courante (système métrique ou anglo-saxon), le type du trait est défini avec les constantes DOC_LINEKIND_*, la couleur du trait est définie avec les constantes COL_* du langage NCL.

Champs	Type	Description
XPOS	INT(2)	abscisse du trait
WIDTH	INT(1)	largeur du trait
COLOR	INT(1)	couleur du trait
KIND	INT(1)	type du trait
RESERVED	INT(2)	réservé (usage interne)

Voir aussi Constantes DOC_LINEKIND_*, Segment DOC_TABLINES, DOC_GETTABLINES, DOC_SETTABLINES, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%, Constantes COL_* (*Manuel de Programmation NCL*)

Segment DOC_TABLINES

Description des tabulations verticales d'un paragraphe.

NBTABLINES indique le nombre d'éléments significatifs du tableau TABS, c'est-à-dire le nombre de tabulations verticales du paragraphe. La première tabulation verticale a pour index 0, la deuxième a pour index 1, et ainsi de suite.

Champs	Type	Description
NBTABLINES	INT(2)	nombre de tabulations verticales dans le paragraphe
RESERVED	INT(2)	réservé (usage interne)
TABS	DOC_TABLINE[64]	caractéristiques de chaque tabulation verticale

Voir aussi Segment DOC_TABLINE, DOC_GETTABLINES, DOC_SETTABLINES

Segment DOC_TABS

Description des retraits et des taquets de tabulation d'un paragraphe.

NBTABS indique le nombre d'éléments significatifs du tableau TABS, c'est-à-dire le nombre de taquets de tabulation du paragraphe. Le premier taquet de tabulation a pour index 0, le deuxième a pour index 1, et ainsi de suite.

Les positions des retraits et des taquets de tabulation sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Champs	Type	Description
LEFTINDENT	INT(2)	retrait gauche du paragraphe
FIRSTLINE	INT(2)	retrait gauche de la première ligne du paragraphe
RIGHTINDENT	INT(2)	retrait droite du paragraphe
NBTABS	INT(2)	nombre de taquets de tabulation dans le paragraphe
TABS	INT(2)[64]	position de chaque taquet de tabulation

Voir aussi DOC SETTABS, DOC GETTABS, DOC DEFAULTMETRICS, DOC METRICS, DOC ISMETRICS%

Segment DOC_TEXTCOLORS

Description des couleurs.

Champs	Type	Description
FORECOLOR	INT(1)	couleur du texte
BACKCOLOR	INT(1)	couleur d'arrière-plan
LINECOLOR	INT(1)	couleur du souligné et du barré
BRESERVED	INT(1)	réservé (usage interne)

Voir aussi DOC_GETFONTINDEXCOLORS, DOC_SETFONTINDEXCOLORS, DOC_GETFONTCOLORS, DOC_SETFONTCOLORS, DOC_GETFONTCOLORSAT, *Constantes COL_** (Manuel de Programmation NCL)

Segment DOC_XTABS

Description des retraits et des taquets de tabulation d'un paragraphe.

NBTABS indique le nombre d'éléments significatifs des tableaux TABS, TABTYPE et TABLEAD, c'est-à-dire le nombre de taquets de tabulation du paragraphe. Le premier taquet de tabulation a pour index 0, le deuxième a pour index 1, et ainsi de suite.

Les positions des retraits et des taquets de tabulation sont exprimées dans l'unité de mesure courante (métrique, anglo-saxonne ou caractère).

Le type d'alignement de chaque taquet de tabulation est défini avec les constantes DOC_TYPTAB_*, le style de point de suite avec les constantes DOC_LEADTAB_*.

Champs	Type	Description
LEFTINDENT	INT(2)	retrait gauche du paragraphe
FIRSTLINE	INT(2)	retrait gauche de la première ligne du paragraphe
RIGHTINDENT	INT(2)	retrait droite du paragraphe
NBTABS	INT(2)	nombre de taquets de tabulation dans le paragraphe
TABS	INT(2)[64]	position de chaque taquet de tabulation
TABTYPE	INT(1)[64]	type d'alignement de chaque taquet de tabulation
TABLEAD	INT(1)[64]	style de point de suite associé à chaque taquet de tabulation

Le segment DOC_TABS décrit également les retraits et les tabulations d'un paragraphe. Utilisé dans une version précédente du produit, moins complet que DOC_XTABS, il est conservé dans l'A.P.I. par souci de compatibilité ascendante.

Voir aussi Constantes DOC TYPTAB *, DOC LEADTAB *, **Segment** DOC TABS, DOC GETXTABS, DOC SETXTABS, DOC MAX TABULATIONS, DOC DEFAULTMETRICS, DOC METRICS, DOC ISMETRICS

Segment **LINE_BORDER**

Description d'un élément d'un encadrement.

Cet élément peut être au choix :

- Un des quatre bords de l'encadrement : gauche, droit, haut et bas.
- La ligne utilisée pour séparer des paragraphes disposés dans un même cadre.
- L'ombre donnée à l'encadrement.

Si l'élément décrit est un des bords de l'encadrement ou la ligne séparatrice :

- Le champ **WIDTH** contient l'épaisseur du bord ou de la ligne. Celle-ci est exprimée dans l'unité de mesure courante (système métrique ou anglo-saxon).
- **KIND** indique le type de trait. **KIND** vaut une des constantes **DOC_LINEKIND_***.
- Le champ **COLOR** contient la couleur du trait. Celle-ci est définie avec les constantes **COL_*** du langage NCL.

Si l'élément décrit est l'ombre de l'encadrement :

- Le champ **WIDTH** contient l'épaisseur de l'ombre, exprimée dans l'unité de mesure courante (système métrique ou anglo-saxon).
- **KIND** représente le type d'ombre. Les constantes **PATSYM_*** de la librairie **NSGRAPH** permettent de spécifier ce type d'ombre.
- Le champ **COLOR** contient la couleur de l'ombre. Celle-ci est définie avec les constantes **COL_*** du langage NCL.

Champs	Type	Description
WIDTH	INT(2)	épaisseur
KIND	INT(2)	type de trait, type d'ombre
COLOR	INT(1)	couleur
BRESERVED	INT(1)	réservé (usage interne)
RESERVED2	INT(2)	réservé (usage interne)

Voir aussi Constantes DOC_LINEKIND_*, Segment DOC_BORDER, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS, Constantes PATSYM_* (Librairie graphique **NSGRAPH), Constantes COL_* (Manuel de Programmation NCL)**

Segment TXTESCTABLE

Description du format texte utilisé par les fonctions ESCLOAD% et ESCPRINT%.

Le champ TABLE est constitué d'un ensemble de chaînes de caractères indexées par les constantes TE*%.

Le champ CHARSET vaut l'une des constantes NCL CHS_*%.

ESCPRI NT% transcrit dans un fichier le document saisi dans une fenêtre fonctionnant en mode "caractère" :

- En remplaçant les différentes commandes (indication de début de document, passage à la ligne, changement de police...) par les chaînes de caractères spécifiées dans le champ TABLE.
- En tenant compte des indications fournies dans les champs NBLINES, HEADER1, HEADERN et CHARSET.

ESCLOAD% effectue la démarche contraire.

Champs	Type	Description
FILENAME	STRING(63)	chemin et nom du fichier
NBLINES	INT(2)	nombre de lignes par page
HEADER1	INT(2)	nombre de lignes d'en-tête, première page
HEADERN	INT(2)	nombre de lignes d'en-tête, autres pages
CHARSET	INT(2)	jeu de caractères (ASCII, ANSI ou EBCDIC)
TABLE	STRING (15) [16]	description des différentes commandes

Voir aussi Constantes TE*%, DOC_DEFAULTMETRICS, DOC_METRICS, DOC_ISMETRICS%, ESCLOAD%, ESCPRINT%, Constantes CHS_*% (Manuel de Programmation NCL)

EXEMPLE D'UN TRAITEMENT DE TEXTE REALISE AVEC NS-WP : NSEDIT

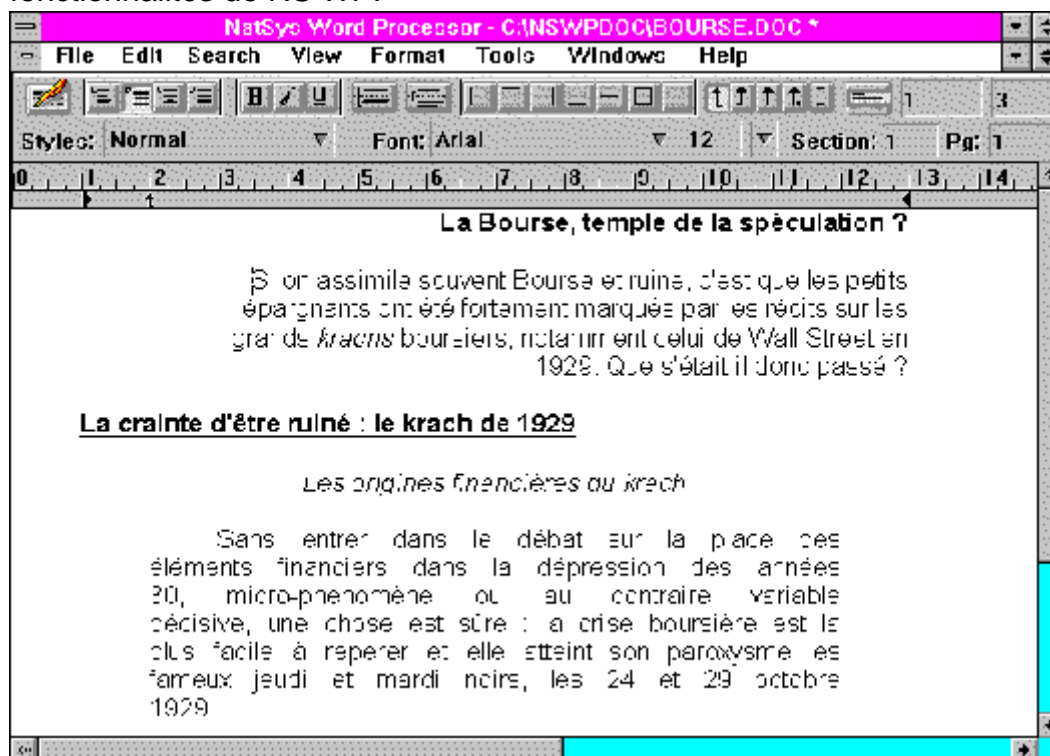
Présentation de NSEDIT

Le traitement de texte NSEDIT a été réalisé avec l'A.P.I. NS WP. Il fait appel à la plupart des verbes NS WP et offre donc les fonctionnalités classiques d'un traitement de texte.

NSEDIT fait partie du package NS WP. Ses sources complètes sont fournis, ainsi que l'application générée : NSEDIT.EXE.

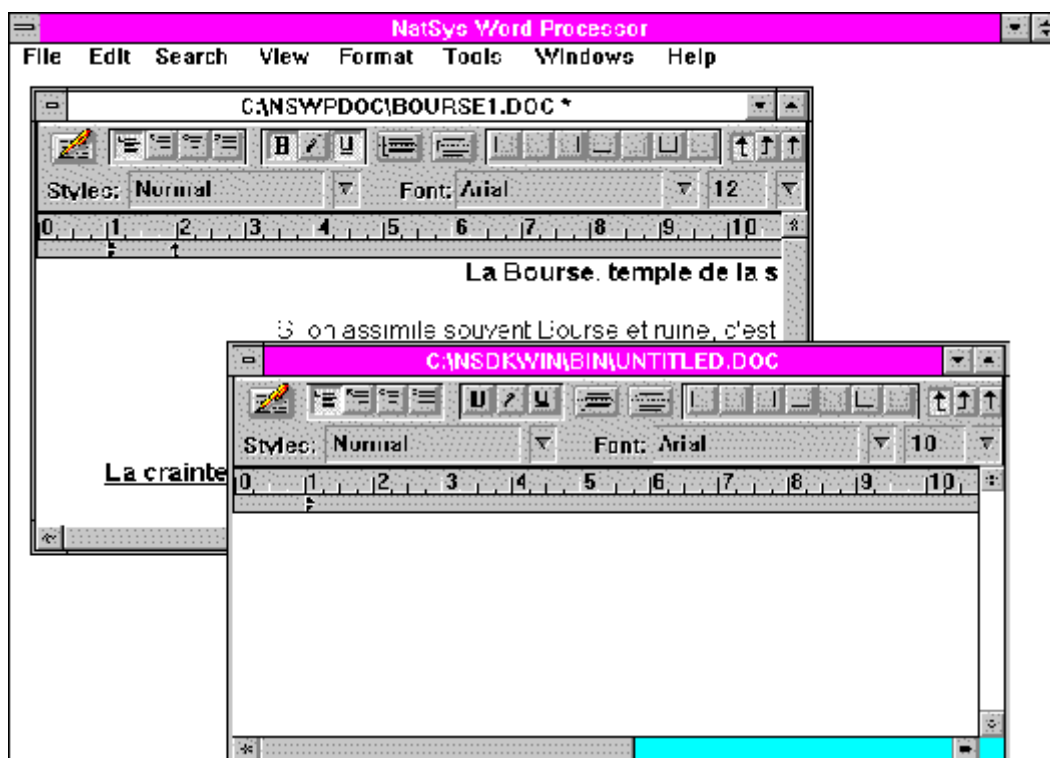
En l'exécutant, vous pourrez vous faire une idée précise des possibilités de NS WP.

En étudiant ses fichiers source, vous disposerez d'exemples de mise en œuvre des fonctionnalités de NS WP.



Traitement de texte NSEDIT

NSEDIT est une application M.D.I. (Multiple Document Interface), ce qui signifie qu'il est possible d'ouvrir plusieurs documents simultanément.



NSEDIT gère le chargement, la sauvegarde et l'impression d'un document. Il offre les facilités d'édition coutumières (presse-papiers, recherche et remplacement de chaînes, annulation de la dernière action).

Un document créé avec NSEDIT peut comprendre des pieds de page et des en-têtes. Sa disposition dépend des marges et de l'alignement vertical choisis. Les paragraphes peuvent être formatés et encadrés. L'aspect des caractères est déterminé par leur police, leur corps, leurs attributs d'enrichissement et leurs couleurs.

Il est possible de faire varier les en-têtes et les pieds de page d'un document ainsi que l'alignement vertical des pages en découpant le document en plusieurs sections. Des images bitmaps, la date et l'heure courantes peuvent être insérées dans un document. Des tableaux sont réalisables au moyen de tabulations verticales.

NSEDIT gère des styles de paragraphe.

NSEDIT gère plusieurs modes d'affichage (cela va du mode "brouillon" au mode "page comme à l'imprimante") et deux systèmes de mesure : le système métrique et le système anglo-saxon.

Il est possible de placer des bornes dans le texte qui faciliteront les déplacements.

NSEDIT est capable de générer une table des matières et de numérotter automatiquement des paragraphes.

Enfin, un document peut être créé au moyen d'un fichier de commandes (création par fusion).

Lancement de NSEDIT

NSEDIT est automatiquement mis sur votre disque lors de l'installation de NS WP (dans un sous-répertoire SAMPLES).

Pour lancer NSEDIT :

Windows

1. Activez le menu Fichier/Exécuter de Windows.
2. Dans le champ Ligne de commande, spécifiez le chemin d'accès à l'exécutable NSEDIT.EXE.

Par exemple :

```
C:\NSDKWIN\SAMPLES\EXE\NSEDIT.EXE
```

3. Validez.

OS/2

1. Dans une ligne de commande OS/2, spécifiez le chemin d'accès à l'exécutable NSEDIT.EXE.

Par exemple :

```
C:\NSDK\SAMPLES\EXE\NSEDIT.EXE
```

2. Validez.

Mise en oeuvre des fonctionnalités de NSEDIT

Le plus souvent, la mise en œuvre dans NSEDIT d'une fonctionnalité de NS WP est directe : une option de menu ouvre une boîte de dialogue dans laquelle sont saisies un certain nombre d'informations qui sont les paramètres nécessaires pour lancer une fonction ou une instruction de la librairie NS-WP.

Exemple de la sauvegarde d'un document :

Le menu File/Save ouvre la boîte Save Document où doivent être spécifiés le format de sauvegarde et le nom du fichier. La validation de ces informations lance l'instruction DOC_SAVE avec les paramètres adéquats.

Néanmoins, la mise en œuvre de certaines des options de NSEDIT, qui font appel à un ensemble de verbes NS WP, est plus complexe. Ce sont principalement :

- le fonctionnement de la barre d'outils,
- la gestion des styles de paragraphes,
- la génération automatique de la table des matières,
- la numérotation automatique.

Fonctionnement de la barre d'outils

La barre d'outils de NSEDIT a été créée spécifiquement. Elle a les fonctionnalités de la barre d'outils proposée en standard par NS WP, augmentées d'une gestion des styles, des sections, de l'encadrement des paragraphes, des lignes et des paragraphes solidaires, ainsi que des tabulations.

Mise en œuvre : fonctionnement de la barre d'outils

La barre d'outils est une fenêtre de classe Dialog dans laquelle sont insérés des boutons et des Combo box Entry field.

Un mécanisme d'échange d'informations entre la fenêtre barre d'outils et le monde extérieur (fenêtre NS WP...) est mis en place :

La fenêtre barre d'outils comporte quatre événements utilisateur USER0, USER1, USER2 et USER3 qui sont activés lorsque certaines actions ont lieu :

- modifications de l'état du texte (position du curseur, numéro de page, changement de police...),
- appui sur les touches [Ctrl] +[T],
- changement des polices de caractère (sélection d'une nouvelle imprimante).

Chaque événement utilisateur contient le traitement à effectuer lorsque l'action a lieu. De la sorte, par l'intermédiaire de ces événements :

- les modifications de l'état du texte sont automatiquement répercutées dans l'affichage de la barre d'outils,
- l'appui sur les touches [Ctrl] +[T] met le focus sur la barre d'outils,
- les listes des polices de caractères et des tailles de caractères d'une police sont automatiquement mises à jour s'il y a sélection d'une nouvelle imprimante.

C'est l'instruction DOC_MAPUSERMSG qui permet de dire quel événement utilisateur doit être activé lorsqu'une action a lieu.

Gestion des styles de paragraphe

Avec NSEDIT, vous pouvez créer des styles et les appliquer aux paragraphes composant le document.

Ces styles contiennent :

- le formatage des caractères qui composent le paragraphe (police, corps, couleurs, attributs d'enrichissement),
- le formatage du paragraphe (retraits, tabulations, alignements),
- les liens entre les lignes du paragraphe, le lien avec le paragraphe suivant,
- les caractéristiques de l'encadrement du paragraphe,
- les caractéristiques des traits verticaux placés dans le paragraphe.

Lorsque vous créez un style, il est ajouté dans la feuille de style _DEFAULT.STY au moment de la sauvegarde du document. NSEDIT ne permet pas de mettre à jour d'autres feuilles de styles que _DEFAULT.STY.

Mise en œuvre : gestion des styles de paragraphes

NSEDIT associe à chaque document un dictionnaire de données qui contient le nom de sa feuille de style. Actuellement, ce nom est dans tous les cas `_DEFAULT.STY`.

Le dictionnaire est nommé `CL_STYLE`.

La feuille de styles est un fichier binaire dont chaque enregistrement comprend les caractéristiques d'un style. Les styles sont identifiés par un numéro unique. Pour mettre à jour ce fichier, NSEDIT passe par l'intermédiaire d'un segment.

Pour appliquer un style à un paragraphe, NSEDIT :

- récupère, dans le dictionnaire, le nom de la feuille de styles associée au document,
- récupère les caractéristiques du style dans la feuille de styles,
- fait ensuite appel aux instructions `DOC_SETFONT*` et `DOC_SETFORMAT` pour modifier la présentation du paragraphe.

Le numéro du style est mis dans l'entier global associé au paragraphe, pour que les paragraphes fils héritent de la présentation de leur père.

Lancement

Pour ajouter un style, lancez Menu Tools/Style
Les styles existants sont affichés dans la barre d'outils.

Génération automatique de la table des matières

Avec NSEDIT, vous pouvez générer automatiquement la table des matières de votre document.

Pour ce faire NSEDIT distingue deux catégories de styles de paragraphe :

- les styles de paragraphes servant de titres dans le document,
- les autres styles, dits normaux.

Dix niveaux de titres sont gérés, un titre de niveau n étant plus important qu'un titre de niveau $n + 1$.

Dès le moment où vous appliquez un style de la catégorie titre à un paragraphe, il figurera dans la table des matières. Son retrait dépendra de son niveau.

La table des matières est insérée à la position courante.

Mise en œuvre : génération de la table des matières

Lorsque vous appliquez un style de la catégorie titre à un paragraphe, le niveau du titre est stocké dans l'entier local associé au paragraphe.

Lorsque NSEDIT génère la table des matières, il insère les paragraphes dont l'entier local associé est différent de zéro.

Lancement

Pour définir un style de titre, lancez Menu Tools/Style

Pour appliquer un style de titre au paragraphe courant, choisissez le style voulu dans le champ Styles de la barre d'outils.

Pour générer une table des matières, lancez Menu Tools/Table of Contents

Numérotation automatique

NSEDIT propose un mécanisme de numérotation automatique de paragraphes.

Vous pouvez numéroter le paragraphe de votre choix. Son numéro est le nombre immédiatement supérieur au numéro qui le précède dans le texte. Tous les numéros suivants dans le texte sont alors incrémentés de un.

Si vous supprimez la numérotation d'un paragraphe, tous les numéros suivants dans le texte sont alors décrémentés de un.

Mise en œuvre : numérotation automatique

NSEDIT utilise les marqueurs pour effectuer cette numérotation automatique.

Chaque numéro inséré est un marqueur. Le texte visualisé est le nombre.

A chaque fois que vous demandez à insérer un nouveau numéro, NSEDIT recherche les marqueurs situés avant dans le texte pour valoriser ce nouveau numéro.

Puis, NSEDIT renumérote les paragraphes marqués situés après dans le texte en modifiant les textes associés aux marqueurs.

Lorsqu'un numéro est supprimé, NSEDIT renumérote les paragraphes marqués situés après dans le texte en modifiant les textes associés aux marqueurs.

Ces opérations de recherche et de mise à jour se font avec les fonctions et instructions DOC_M_*. Elles sont lancées dans l'événement utilisateur USER10 de la fenêtre NS WP activé à chaque insertion ou suppression de marqueur. C'est l'emploi de l'instruction DOC_MAPUSERMSG qui permet le déclenchement de cet événement utilisateur.

Lancement

Pour numéroté le paragraphe courant, lancez Menu Tools/Insert Number

Pour supprimer la numérotation d'un paragraphe, effacez le numéro au moyen de la touche [Del] ou de la touche [Backspace].

Composants de NSEDIT

Fenêtres

NS_WP1 : Natsys word processor

Fenêtre de classe MDI Window permettant d'ouvrir une fenêtre NS-WP par document.

NS_WP2

Fenêtre NS-WP de classe Window ayant {NS02WP} dans son titre.

Depuis cette fenêtre sont ouvertes, lors des sélections 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_WP3 : About natsys word processor

A propos sur NSEDIT.

Fenêtre de classe Dialog appelée par Help/About.

NS_WP4 : Format paragraph

Spécification des alignements, des tabulations et des retraits d'un paragraphe.

Fenêtre de classe Dialog appelée par Format/Paragraph.

NS_WP4A : Verticals lines

Spécification des tabulations verticales.

Fenêtre de classe Dialog appelée par Format/Verticals lines.

NS_WP5 : Font selection

Choix du style courant.

Fenêtre de classe Dialog appelée par Format/Font.

NS_WP5A

Affichage de l'aspect du style spécifié.

Fenêtre NS-WP de classe Window ayant {NS02WP} dans son titre. Cette fenêtre est visualisée dans la fenêtre Font selection.

NS_WP6 : Select printer

Sélection de l'imprimante.

Fenêtre de classe Dialog appelée par File/Printer setup.

NS_WP7 : Print document

Lancement de l'impression.

Fenêtre de classe Dialog appelée par File/Print.

NS_WP8 : Search string

Recherche de texte.

Fenêtre de classe Dialog appelée par Search/Find, par Search/Last find ou par Search/Selected text.

NS_WP9 : Change string

Remplacement de texte.

Fenêtre de classe Dialog appelée par Search/Change.

NS_WP9A : Replace

Confirmation de remplacement.

Fenêtre de classe Dialog appelée depuis l'événement USER0 de NS_WP2.

NS_WP10 : Save document

Spécification des caractéristiques du fichier de sauvegarde.

Fenêtre de classe Dialog appelée par File/Save as ou par File/Save.

NS_WP11 : Go to page

Positionnement sur une page du texte.

Fenêtre de classe Dialog appelée par Search/Goto page.

NS_WP12 : Show Codes

Sélection des caractères spéciaux à visualiser.

Fenêtre de classe Dialog appelée par View/Show codes.

NS_WP13 : Picture

Spécification de la bitmap à insérer.

Fenêtre de classe Dialog appelée par Tools/Picture.

NS_WP14 : Text color

Choix des couleurs du texte à entrer.

Fenêtre de classe Dialog appelée par Format/Colors.

NS_WP17 : Format for document

Mise en page du document.

Fenêtre de classe Dialog appelée par Format/Document.

NS_WP18 : Styles

Mise à jour des styles de paragraphe.

Fenêtre de classe Dialog appelée par Tools/Styles.

NS_WP19 : Modify formats

Définition d'un style.

Fenêtre de classe Dialog appelée par bouton Modify de la boîte Styles.

NS_WP20

Barre d'outils.

Fenêtre de classe Dialog.

NS_WP21 : Section

Spécification de l'alignement vertical des pages d'une section.

Fenêtre de classe Dialog appelée par Format/Section

NS_WP22 : Table of content

Table des matières.

Fenêtre cachée où sont construites les tables des matières.

NS_WP23 : Border

Spécification des caractéristiques d'un encadrement.

Fenêtre de classe Dialog appelée par Format/Border.

NS_WP24

Affichage de l'aspect de l'encadrement spécifié.

Fenêtre NS-WP de classe Window ayant {NS02WP} dans son titre. Cette fenêtre est visualisée dans la fenêtre Border.

NS_WP25 : Border background

Spécification des caractéristiques du motif de fond d'écran d'un encadrement.

Fenêtre de classe Dialog appelée par bouton Background de la fenêtre Border.

NS_WP27 : Statistic

Statistiques sur le document.

Fenêtre de classe Dialog appelée par Help/Statistics.

NS_WP28

Création d'un en-tête ou d'un pied de page.

Fenêtre de classe Dialog appelée dans la barre d'outils.

NS_WAIT

Message d'attente.

Fenêtre de classe Dialog.

Librairies

Librairies NS DK/NatStar

- NSDATE
- NSGRAPH
- NSMISC
- NSPATH
- NSPRN
- NSWIN (sous Windows), NSPM (sous OS/2 PM)

Librairies spécifiques à NSEDIT

- NS_WPL

Déclaration de segments globaux et gestion du mécanisme M.D.I.

- NS_WPL2

Gestion de la barre d'outils.

- NS_WPL3

Gestion des styles de paragraphes et génération de la table des matières.

- NS_WPL4

Encadrement de paragraphes.

- NS_WPL5

Numérotation automatique.

- NS_WPLDRG

Manipulation des images bitmaps.

- NS_WPLU

Utilitaires.

Librairie NS_WP

Librairie nécessaire pour toute application NSWP.

Autres ressources

Segment NS_WPS

Segment alloué pour chaque fenêtre NS_WP2 ouverte.

Ressource Variables NS_WPV

Déclarations des variables globales et textes des commentaires utilisés dans le projet NSEDIT.

Icônes NS_WPI1 et NS_WPI2

Icônes de la fenêtre NS_WP1 et de la fenêtre NS_WP2.

Bitmaps BALL, BALLP, BBORDER, ..., WHEADD

Bitmaps affichées dans la barre d'outils.

Bitmap NS_WPB

Bitmap affichée au sein de la fenêtre NS_WP3.

Utilisation de la librairie NS-WP au sein de NSEDIT

Cette partie recense les verbes de l'A.P.I. NS WP qui sont utilisés dans NSEDIT, avec leurs localisations (dans quel événement de quel contrôle de quelle fenêtre, dans quelle librairie).

Fonction/Instruction Segment/Constante	Ressource	Contrôle	Evénement
DOC_ALLOCATEDMEM%	NS_WP3	-	INIT
DOC_ALLOCSCANCONTEXT%	NS_WP2	MI_CLEAR_TAG	EXECUTED
	NS_WP27	-	INIT
	NS_WPLU		
DOC_ADDPICTDFN%	NS_WP13	FFIND	EXECUTED
DOC_ASSOCIATEPRINTER	NS_WP5A	-	INIT
	NS_WP6	OK	EXECUTED
	NS_WPL		
DOC_BORDER	NS_WP2	MI_BORDER1	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	B_DOWN_N	EXECUTED
	NS_WP23	B_DOWN_Y	EXECUTED
	NS_WP23	B_LEFT_N	EXECUTED
	NS_WP23	B_LEFT_Y	EXECUTED
	NS_WP23	B_RIGHT_N	EXECUTED
	NS_WP23	B_RIGHT_Y	EXECUTED
	NS_WP23	B_UP_N	EXECUTED
	NS_WP23	B_UP_Y	EXECUTED
	NS_WP23	B_WEEN_N	EXECUTED
	NS_WP23	B_WEEN_Y	EXECUTED
	NS_WP23	CHK_BACK	SELECTED
	NS_WP23	DSHADOW	SELECTED
	NS_WP23	SPACE	SELECTED
	NS_WP25	-	INIT
	NS_WP25	OK	EXECUTED
	NS_WPL2		

	NS_WPL3		
	NS_WPL4		
DOC_CHANGE%	NS_WP2	MI_CHANGE1	EXECUTED
DOC_CHANGE_ALL%	NS_WP9A	ALL	EXECUTED
DOC_CHANGE_CANCEL%	NS_WP9A	PB_CANCEL	EXECUTED
DOC_CHANGE_NO%	NS_WP9A	PB_NO	EXECUTED
DOC_CHANGE_YES%	NS_WP9A	PB_YES	EXECUTED
DOC_CLEAR TAG	NS_WP2	MI_TOGGLETAG	EXECUTED
	NS_WP2	MI_CLEAR TAG	EXECUTED
DOC_COPY	NS_WP2	MI_COPY1	EXECUTED
DOC_COPYSELECTION	NS_WPL3		
DOC_CUT	NS_WP2	MI_CUT1	EXECUTED
DOC_DELETE	NS_WP2	MI_DELETE1	EXECUTED
	NS_WP2	MI_NUMBERED	EXECUTED
DOC_DELETECHARS	NS_WPL3		
DOC_DELPIC TDFN%	NS_WP13	FFIND	EXECUTED
DOC_DISPOSEWINDOW	NS_WP28	PB_DELETE	EXECUTED
DOC_ERRCANNOTREADFILE%	NS_WP13	FFIND	EXECUTED
DOC_ERRDUPLPICTNAME%	NS_WP13	FFIND	EXECUTED
DOC_ERRINVALIDPICT%	NS_WP13	FFIND	EXECUTED
DOC_ERRNO%	NS_WP7	PB_PRINT	EXECUTED
DOC_ERR*%	NS_WP1	MI_OPEN1	EXECUTED
	NS_WP2	MI_SAVE1	EXECUTED
	NS_WP10	OK	EXECUTED
DOC_ERROR%	NS_WP1	MI_OPEN1	EXECUTED
	NS_WP1	MI_MERGE1	EXECUTED
	NS_WP2	MI_SAVE1	EXECUTED
	NS_WP10	OK	EXECUTED
	NS_WP13	FFIND	EXECUTED
	NS_WPL		EXECUTED
DOC_FILE_NSWP%	NS_WP10	OK	EXECUTED
DOC_FILE_RTF%	NS_WP10	OK	EXECUTED
DOC_FILE_RFT%	NS_WP10	OK	EXECUTED
DOC_FILE_FORMAT%	NS_WP10	-	INIT

DOC_FIND_CONFIRM%	NS_WP2	MI_CHANGE1	EXECUTED
DOC_FIND_LOW2UPCASE%	NS_WP2	MI_LASTFIND1	EXECUTED
	NS_WP2	MI_SELECTED1	EXECUTED
	NS_WP2	MI_CHANGE1	EXECUTED
	NS_WP8	OK	EXECUTED
DOC_FIND_WHOLEWORD%	NS_WP2	MI_LASTFIND1	EXECUTED
	NS_WP2	MI_SELECTED1	EXECUTED
	NS_WP2	MI_CHANGE1	EXECUTED
	NS_WP8	OK	EXECUTED
DOC_FIND%	NS_WP2	MI_LASTFIND1	EXECUTED
	NS_WP2	MI_SELECTED1	EXECUTED
DOC_FIRSTPARAGRAPH%	NS_WP2	MI_LASTFIND1	EXECUTED
	NS_WP2	MI_SELECTED1	EXECUTED
	NS_WP2	MI_CHANGE1	EXECUTED
	NS_WP8	OK	EXECUTED
DOC_FMT_BOLD%	NS_WP2	MI_BOLD1	EXECUTED
	NS_WP2	M_FORMAT1	INIT
	NS_WP5	-	INIT
	NS_WP5	-	REFRESH V
	NS_WP5	OK	EXECUTED
	NS_WP20	BMP_BOLD	EXECUTED
	NS_WPL2		SELECTED
DOC_FMT_ITALIC%	NS_WP2	MI_ITALIC1	EXECUTED
	NS_WP2	M_FORMAT1	INIT
	NS_WP5	-	INIT
	NS_WP5	-	REFRESH V
	NS_WP5	OK	EXECUTED
	NS_WP20	BMP_ITALIC	SELECTED
	NS_WPL2		
DOC_FMT_STANDARD%	NS_WP2	MI_STANDARD1	EXECUTED
DOC_FMT_STRIKEOUT%	NS_WP2	M_FORMAT1	INIT
	NS_WP5	-	INIT
	NS_WP5	-	REFRESH V
	NS_WP5	OK	EXECUTED

DOC_FMT_UNDERLINE%	NS_WP2	MI_UNDERLINE1	EXECUTED
	NS_WP2	M_FORMAT1	INIT
	NS_WP5	-	INIT
	NS_WP5	-	REFRESH V
	NS_WP5	OK	EXECUTED
	NS_WP20	BMP_UNDERLINE	SELECTED
	NS_WPL2		
DOC_FMT_UNLOCKED%	NS_WP20	-	USER1
DOC_FONTNAME_EVENT	NS_WPL2		
DOC_FONTSIZE_EVENT	NS_WPL2		
DOC_FREESCANCONTEXT%	NS_WP2	MI_CLEAR_TAG	EXECUTED
	NS_WP27	-	INIT
	NS_WPLU		
DOC_GETBITMAPCHAR%	NS_WP2	-	BUTTONUP
	NS_WPDRG		
DOC_GETBORDER	NS_WP2	MI_BORDER1	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_GETCONTROLCHARSVISIBILITY%	NS_WP12	-	INIT
DOC_GETCURRENTRULEROBJECT%	NS_WPL2		
DOC_GETCURRENTTAB	NS_WPL2		
DOC_GETDEFTAB%	NS_WP17	-	INIT
	NS_WP17	OK	EXECUTED
DOC_GETFONT\$	NS_WP2	MI_FONT1	EXECUTED
	NS_WP5	-	INIT
	NS_WP5	OK	EXECUTED
	NS_WPL3		
DOC_GETFONTCOLORS	NS_WP14	-	INIT
DOC_GETFONTINDEXFORMAT%	NS_WP5	-	REFRESH V
DOC_GETFONTNAMECONTROL	NS_WP5	-	INIT
	NS_WP20	-	USER2
DOC_GETFONTSIZECONTROL	NS_WP5	-	INIT
	NS_WP5	-	REFRESH V
	NS_WP20	-	USER2

DOC_GETFORMAT%	NS_WP2	M_FORMAT1	INIT
	NS_WP2	MI_ITALIC1	EXECUTED
	NS_WP2	MI_UNDERLINE1	EXECUTED
	NS_WP2	MI_STRIKEOUT1	EXECUTED
	NS_WP2	MI_BOLD1	EXECUTED
	NS_WP2	MI_FONT1	EXECUTED
	NS_WP5	-	INIT
	NS_WP5	OK	EXECUTED
	NS_WP20	BMP_BOLD	SELECTED
	NS_WP20	BMP_ITALIC	SELECTED
	NS_WP20	BMP_UNDERLINE	SELECTED
	NS_WPL3		
DOC_GETJUSTIF%	NS_WP2		
	NS_WP2	M_FORMAT1	INIT
	NS_WPL3	MI_PARAGRAPH1	EXECUTED
DOC_GETLASTMERGEERROR	NS_WP1	MI_MERGE1	EXECUTED
DOC_GETLINE	NS_WPL3		
DOC_GETMARGINS	NS_WP7	-	INIT
	NS_WP17	-	INIT
	NS_WPL3		
DOC_GETMODE%	NS_WP2	M_VIEW1	INIT
DOC_GETNBPAGES%	NS_WP2	-	CHARACTE
	NS_WP11	-	INIT
DOC_GETNBSECTIONS%	NS_WP21	-	INIT
DOC_GETPAGE%	NS_WP2	-	CHARACTE
	NS_WP11	-	INIT
	NS_WPL3		
DOC_GETPAGEDIM	NS_WP7	-	INIT
	NS_WP17	-	INIT
	NS_WPL3		
DOC_GETPAGEZOOM	NS_WP16	OK	EXECUTED
DOC_GETPARAGRAPHGLOBALUSERINFO%	NS_WP18	-	INIT
	NS_WP18	BP_APPLY	EXECUTED
DOC_GETPARAGRAPHKEEPTOGETHER%	NS_WP2	MI_PARAGRAPH1	EXECUTED

	NS_WP20	BMP_KEEP	SELECTED
DOC_GETPARAGRAPHKEEPWITHNEXT%	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WP20	BMP_KEEPl	SELECTED
DOC_GETPARAGRAPHLOCALUSERINFO%	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL3		
DOC_GETPICTDFNNAME%	NS_WPDRG		
DOC_GETPICTSIZE%	NS_WP13	FFIND	EXECUTED
DOC_GETPOS	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL2		
	NS_WPL3		
	NS_WPDRG		
DOC_GETSECTION%	NS_WP21	-	INIT
	NS_WP28	-	INIT
DOC_GETSECTIONJUSTIF%	NS_WP21	-	INIT
DOC_GETSELECTION	NS_WP14	BACKLIST	SELECTED
	NS_WP14	FORELIST	SELECTED
	NS_WP14	LINELIST	SELECTED
	NS_WP14	-	INIT
	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL2		
	NS_WPL3		
	NS_WPLU		
DOC_GETSELECTIONTEXT\$	NS_WP2	MI_SELECTED1	EXECUTED
DOC_GETTABLINES	NS_WP2	MI_TABLINES1	EXECUTED
	NS_WPL3		
DOC_GETTABS	NS_WP4	-	INIT
DOC_GETXTABS	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WPL3		
DOC_GETWIDOWCONTROL%	NS_WP17	-	INIT
	NS_WP17	OK	EXECUTED
DOC_GETWINDOW%	NS_WP2	M_VIEW1	INIT
	NS_WP2	M_FORMAT1	INIT
	NS_WP20	-	USER1

	NS_WP20	-	WINDOW
	NS_WP28	-	INIT
DOC_HIDE_PICTURES%	NS_WP12	OK	EXECUTED
DOC_INSERTCHARS	NS_WP2	MI_DATE1	EXECUTED
	NS_WP2	MI_TIME1	EXECUTED
	NS_WP5A	-	INIT
	NS_WPL3		
DOC_ISCLIPBOARD%	NS_WP2	M_EDIT1	INIT
DOC_ISLIKEPRINTER%	NS_WP2	MI_LIKEPRINTER1	EXECUTED
	NS_WP2	M_VIEW1	INIT
DOC_ISMETRICS%	NS_WP4	-	INIT
	NS_WP4	OK	EXECUTED
	NS_WP17	-	INIT
	NS_WP17	OK	EXECUTED
DOC_ISPAGEBREAK%	NS_WP2	M_FORMAT1	INIT
	NS_WPL3		
DOC_ISREADONLY%	NS_WP2	M_EDIT1	INIT
	NS_WP2	MI_READONLY1	EXECUTED
	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WP2	MI_BORDER1	EXECUTED
	NS_WP2	MI_FONT1	EXECUTED
	NS_WP2	M_TOOLS1	INIT
	NS_WP2	MI_TABLINES1	EXECUTED
	NS_WP7	-	INIT
	NS_WP7	PB_PRINT	EXECUTED
	NS_WP19	PB_BORDER	EXECUTED
	NS_WP19	PB_VLINES	EXECUTED
	NS_WP19	BP_PARAGRAPH	EXECUTED
	NS_WP19	BP_FONT	EXECUTED
	NS_WP20	BMP_EDIT	SELECTED
	NS_WP20	-	USER1
DOC_ISREDO%	NS_WP2	M_EDIT1	INIT
DOC_ISRULERBARSHOWN%	NS_WP2	M_VIEW1	INIT
	NS_WP2	MI_RULERBAR1	EXECUTED

DOC_ISSECTION%	NS_WP2	M_FORMAT1	INIT
DOC_ISSELECTION%	NS_WP2	M_EDIT1	INIT
	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WP2	MI_FONT1	EXECUTED
	NS_WP14	-	INIT
	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL2		
	NS_WPL3		
	NS_WPLU		
DOC_ISTAG%	NS_WP2	MI_TOGGLETAG	EXECUTED
DOC_ISUNDO%	NS_WP2	M_EDIT1	INIT
DOC_JUS_LEFT%	NS_WP2	M_FORMAT1	INIT
	NS_WP2	MI_LEFT1	EXECUTED
	NS_WP20	BMP_LEFT_P	EXECUTED
	NS_WP20	BMP_LEFT_R	EXECUTED
	NS_WPL2		
DOC_JUS_RIGHT%	NS_WP2	M_FORMAT1	INIT
	NS_WP2	MI_RIGHT1	EXECUTED
	NS_WP20	BMP_RIGHT_P	EXECUTED
	NS_WP20	BMP_RIGHT_R	EXECUTED
	NS_WPL2		
DOC_JUS_CENTERED%	NS_WP2	M_FORMAT1	INIT
	NS_WP2	MI_CENTERED1	EXECUTED
	NS_WP20	BMP_CENTERED_P	EXECUTED
	NS_WP20	BMP_CENTERED_R	EXECUTED
	NS_WPL2		
DOC_JUS_FULL%	NS_WP2	M_FORMAT1	INIT
	NS_WP2	MI_FULL1	EXECUTED
	NS_WP20	BMP_FULL_P	EXECUTED
	NS_WP20	BMP_FULL_R	EXECUTED
	NS_WPL2		
DOC_LASTPARAGRAPH%	NS_WP2	-	CHARACTE
	NS_WP2	MI_SELECTALL1	EXECUTED
	NS_WP23	-	INIT

DOC_LAYOUT	NS_WP1	-	INIT
	NS_WP1	MI_OPEN1	EXECUTED
	NS_WPL		
	NS_WPI3		
DOC_LEADTAB_DOT%	NS_WPL3		
DOC_LINE_ABOVE%	NS_WP20	BP_BUP	EXECUTED
	NS_WP20	BP_BUR	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	B_UP_N	EXECUTED
	NS_WP23	B_UP_Y	EXECUTED
	NS_WPL2		
DOC_LINE_BELOW%	NS_WP20	BP_BDP	EXECUTED
	NS_WP20	BP_BDR	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	B_DOWN_N	EXECUTED
	NS_WP23	B_DOWN_Y	EXECUTED
	NS_WPL2		
DOC_LINE_BETWEEN%	NS_WP20	BP_BMP	EXECUTED
	NS_WP20	BP_BMR	EXECUTED
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	B_WEEN_N	EXECUTED
	NS_WP23	B_WEEN_Y	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_LINE_LEFT%	NS_WP20	BP_BLP	EXECUTED
	NS_WP20	BP_BLR	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED

	NS_WP23	B_LEFT_N	EXECUTED
	NS_WP23	B_LEFT_Y	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_LINE_RIGHT%	NS_WP20	BP_BRP	EXECUTED
	NS_WP20	BP_BRR	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	B_RIGHT_N	EXECUTED
	NS_WP23	B_RIGHT_Y	EXECUTED
	NS_WPL2		
DOC_LINE_SHADOW%	NS_WP23	-	INIT
	NS_WP23	-	UPDATEBO
	NS_WP23	B_ALL_N	EXECUTED
	NS_WP23	B_ALL_Y	EXECUTED
	NS_WP23	DSHADOW	SELECTED
DOC_LINEDATA	NS_WPL3		
DOC_M_CREATEMARKERWITHHANDLE%	NS_WPL3		
DOC_M_DELETEMARKER_EVENT	NS_WP2	-	USER10
	NS_WPL		
DOC_M_FINDNEXTMARKER%	NS_WPL5		
DOC_M_FINDPREVMARKER%	NS_WP2	MI_NUMBERED	EXECUTED
DOC_M_GETCLASS%	NS_WPL3		
DOC_M_GETCLASSNAME\$	NS_WP2	-	USER10
DOC_M_GETMARKERDATA	NS_WP2	-	USER10
	NS_WPL3		
DOC_M_GETMARKERHANDLE%	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WPL5		
DOC_M_GETTEXT\$	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WPL5		
DOC_M_INSERTMARKER	NS_WP2	MI_NUMBERED	EXECUTED
DOC_M_MARKER	NS_WP2	-	USER10
DOC_M_NEWMARKERCLASS%	NS_WPL3		

DOC_M_PASTEMARKER_EVENT	NS_WP2	-	USER10
	NS_WPL		
DOC_M_REGISTERMARKERCLASS	NS_WPL3		
DOC_M_SETMARKERDATA	NS_WPL3		
DOC_M_SETTEXT	NS_WPL5		
DOC_MAPUSERMSG	NS_WPL		
	NS_WPL2		
DOC_METRICS	NS_WP4	OK	EXECUTED
	NS_WP17	OK	EXECUTED
DOC_MODE_DEFAULT%	NS_WP2	MI_DEFAULT	EXECUTED
	NS_WP2	M_VIEW1	INIT
DOC_MODE_DRAFT%	NS_WP2	MI_DRAFT	EXECUTED
	NS_WP2	M_VIEW1	INIT
DOC_MODE_PAGE%	NS_WP2	MI_PAGE	EXECUTED
	NS_WP2	M_VIEW1	INIT
DOC_MODIFYBITMAPCHAR%	NS_WP2	-	BUTTONUP
DOC_MODIFYFORMAT	NS_WP2	MI_FONT1	EXECUTED
	NS_WPL3		
DOC_MOVESELECTION	NS_WPL3		
DOC_NEXTPARAGRAPH%	NS_WP2	MI_CLEAR_TAG	EXECUTED
	NS_WP18	BP_APPLY	EXECUTED
	NS_WP23	-	INIT
	NS_WPL2		
	NS_WPL3		
DOC_PAGEBREAK	NS_WP2	MI_PAGEBREAK1	EXECUTED
	NS_WPL3		
DOC_PASTE	NS_WP2	MI_PASTE1	EXECUTED
DOC_POS2XYVIEW%	NS_WPDRG		
DOC_PRINT%	NS_WP7	PB_PRINT	EXECUTED
DOC_PRINTFILE%	NS_WP7	PB_PRINT	EXECUTED
DOC_RCHG_COLUMN	NS_WP20	-	USER1
DOC_RCHG_LINE	NS_WP20	-	USER1
DOC_RCHG_PAGE	NS_WP20	-	USER1
DOC_RCHG_SECTION	NS_WP20	-	USER1

DOC_RCHG_CONTEXT	NS_WP20	-	USER1
DOC_RCHG_FACENAME	NS_WP20	-	USER1
DOC_RCHG_FACESIZE	NS_WP20	-	USER1
DOC_RCHG_PARAINFO	NS_WP20	-	USER1
DOC_RCHG_USERPARAINFO	NS_WP20	-	USER1
DOC_RCHG_BORDER	NS_WP20	-	USER1
DOC_READONLY	NS_WP2	MI_READONLY1	EXECUTED
	NS_WP20	BMP_EDIT	SELECTED
	NS_WP23	-	UPDATEBO
	NS_WP23	BP_BACKGROUND	EXECUTED
DOC_REDO	NS_WP2	MI_REDO1	EXECUTED
DOC_RULEINFO	NS_WP20	-	USER1
DOC_RULER_EVENT	NS_WPL2		
DOC_RULERBAR	NS_WP2	-	GETFOCUS
	NS_WP2	MI_RULERBAR	EXECUTED
DOC_SAVE	NS_WP10	OK	EXECUTED
DOC_SCAN_CTXT	NS_WP2	MI_CLEARTAG	EXECUTED
DOC_SCANFIRST%	NS_WP2	MI_CLEARTAG	EXECUTED
	NS_WP27	-	INIT
	NS_WPLU		
DOC_SCANNEXT%	NS_WP2	MI_CLEARTAG	EXECUTED
	NS_WP27	-	INIT
DOC_SCN_BEGBMP%	NS_WP2	MI_CLEARTAG	EXECUTED
DOC_SCN_BEGMARK%	NS_WP27	-	INIT
DOC_SCN_ENDPARA%	NS_WP2	MI_CLEARTAG	EXECUTED
	NS_WP27	-	INIT
DOC_SCN_ERR%	NS_WP2	MI_CLEARTAG	EXECUTED
	NS_WP27	-	INIT
	NS_WPLU		
DOC_SCN_FONT%	NS_WP27	-	INIT
DOC_SCN_SPACE%	NS_WP27	-	INIT
DOC_SCN_TAB%	NS_WP27	-	INIT
DOC_SCN_TAG%	NS_WP2	MI_CLEARTAG	EXECUTED
	NS_WP27	-	INIT

DOC_SECTION	NS_WP2	MI_DELSECTION1	EXECUTED
	NS_WP2	MI_ADDSECTION1	EXECUTED
DOC_SETBORDER	NS_WP2	MI_BORDER1	EXECUTED
	NS_WP23	-	UPDATEBO
	NS_WP23	BP_BACKGROUND	EXECUTED
	NS_WP23	CHK_BACK	SELECTED
	NS_WPL2		
	NS_WPL3		
DOC_SETCONTROLCHARSVISIBILITY	NS_WP12	OK	EXECUTED
DOC_SETCURRENTTAB	NS_WPL2		
DOC_SETCURRENTTABLINE	NS_WPL2		
DOC_SETDEFTAB	NS_WP17	OK	EXECUTED
DOC_SETFOCUS_EVENT	NS_WPL2		
DOC_SETFONT	NS_WP2	MI_FONT1	EXECUTED
	NS_WP20	CMB_FONTNAME	EXECUTED
	NS_WP20	CMB_FONTSIZE	EXECUTED
	NS_WP5	OK	EXECUTED
	NS_WP20	OK_FONTS	EXECUTED
	NS_WPL3		
DOC_SETFONTBACKCOLOR	NS_WP14	BACKLIST	SELECTED
	NS_WP14	BP_OK	EXECUTED
DOC_SETFONTCOLORS	NS_WPL3		
DOC_SETFONTFORECOLOR	NS_WP14	FORELIST	SELECTED
	NS_WP14	BP_OK	EXECUTED
DOC_SETFONTINDEXFORMAT	NS_WP5	-	REFRESH V
DOC_SETFONTINDEXNAME	NS_WP5	-	REFRESH V
DOC_SETFONTINDEXSIZE	NS_WP5	-	REFRESH V
DOC_SETFONTLINECOLOR	NS_WP14	LINELIST	SELECTED
	NS_WP14	BP_OK	EXECUTED
DOC_SETFORMAT	NS_WP2	MI_STANDARD1	EXECUTED
	NS_WP2	MI_BOLD1	EXECUTED
	NS_WP2	MI_ITALIC1	EXECUTED
	NS_WP2	MI_UNDERLINE1	EXECUTED
	NS_WP2	MI_STRIKEOUT1	EXECUTED

	NS_WP5	OK	EXECUTED
	NS_WP20	BMP_BOLD	SELECTED
	NS_WP20	BMP_ITALIC	SELECTED
	NS_WP20	BMP_UNDERLINE	SELECTED
	NS_WPL3		
DOC_SETJUSTIF	NS_WP2	MI_LEFT1	EXECUTED
	NS_WP2	MI_RIGHT1	EXECUTED
	NS_WP2	MI_CENTERED1	EXECUTED
	NS_WP2	MI_FULL1	EXECUTED
	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WP20	BMP_RIGHT_R	EXECUTED
	NS_WP20	BMP_CENTERED_R	EXECUTED
	NS_WP20	BMP_FULL_R	EXECUTED
	NS_WP20	BMP_LEFT_R	EXECUTED
	NS_WP20	BMP_RIGHT_P	EXECUTED
	NS_WP20	BMP_CENTERED_P	EXECUTED
	NS_WP20	BMP_FULL_P	EXECUTED
	NS_WP20	BMP_LEFT_P	EXECUTED
	NS_WPL3		
DOC_SETLIKEPRINTER	NS_WP2	MI_LIKEPRINTER1	EXECUTED
DOC_SETMARGINS	NS_WP7	-	CHECK_MA
	NS_WP17	OK	EXECUTED
DOC_SETMODE	NS_WP2	MI_DEFAULT	EXECUTED
	NS_WP2	MI_DRAFT	EXECUTED
	NS_WP2	MI_PAGE	EXECUTED
	NS_WP5A	-	INIT
	NS_WP16	OK	EXECUTED
DOC_SETPAGE	NS_WP2	-	CHARACTE
	NS_WP11	OK	EXECUTED
DOC_SETPAGEZOOM	NS_WP16	OK	EXECUTED
DOC_SETPARAGRAPHGLOBALUSERINFO	NS_WPL3		
DOC_SETPARAGRAPHKEEPTOGETHER	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WP20	BMP_KEEP	SELECTED
DOC_SETPARAGRAPHKEEPWITHNEXT	NS_WP2	MI_PARAGRAPH1	EXECUTED

	NS_WP20	BMP_KEEPN	SELECTED
DOC_SETPARAGRAPHLOCALUSERINFO	NS_WPL3		
DOC_SETPOS	NS_WP2	-	CHARACTE
	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WP5A	-	INIT
	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_SETSECTIONJUSTIF	NS_WP21	OK	EXECUTED
DOC_SETSELECTION	NS_WP2	MI_SELECTALL1	EXECUTED
	NS_WP14	BACKLIST	SELECTED
	NS_WP14	FORELIST	SELECTED
	NS_WP14	LINELIST	SELECTED
	NS_WP14	-	INIT
	NS_WP18	BP_APPLY	EXECUTED
	NS_WP23	-	INIT
	NS_WPL2		
	NS_WPL3		
DOC_SETTABLINES	NS_WP2	MI_TABLINES1	EXECUTED
	NS_WPL3		
DOC_SETTABS	NS_WP4	OK	EXECUTED
DOC_SETTAG	NS_WP2	MI_TOGGLETAG	EXECUTED
DOC_SETXTABS	NS_WP2	MI_PARAGRAPH1	EXECUTED
	NS_WPL3		
DOC_SETWIDOWCONTROL	NS_WP17	OK	EXECUTED
DOC_SETWINDOW	NS_WP2	MI_HEADER1	EXECUTED
	NS_WP2	MI_BODY1	EXECUTED
	NS_WP2	MI_FOOTER1	EXECUTED
	NS_WP20	-	WINDOW
	NS_WP28	PB_EDIT	EXECUTED
	NS_WP28	PB_NEW	EXECUTED
DOC_SHOW_SPACES%	NS_WP12	OK	EXECUTED
DOC_SHOW_TABS%	NS_WP12	OK	EXECUTED
DOC_SHOW_PARAGRAPHS%	NS_WP12	OK	EXECUTED

DOC_SHOW_COMMANDS%	NS_WP12	OK	EXECUTED
DOC_SHOW_TAGS%	NS_WP12	OK	EXECUTED
DOC_SHOWRULERBAR	NS_WP2	MI_RULERBAR1	EXECUTED
	NS_WPL		
DOC_SPLITPARAGRAPH	NS_WP23	-	INIT
	NS_WPL3		
DOC_TABLINES	NS_WP2	MI_TABLINES1	EXECUTED
	NS_WP4A	-	INIT
	NS_WP4A	LB_TABULATIONS	SELECTED
	NS_WP4A	PB_DELETE1	EXECUTED
	NS_WP4A	PB_DELETEALL1	EXECUTED
	NS_WP4A	PB_INSERT1	EXECUTED
	NS_WP4A	PB_MODIFY1	EXECUTED
	NS_WPL3		
DOC_TABS	NS_WP4	-	INIT
	NS_WP4	OK	EXECUTED
	NS_WP23	-	INIT
DOC_TEXTCOLORS	NS_WPL3		
DOC_TYPTAB_CENTER%	NS_WP20	TCENTER_R	EXECUTED
	NS_WPL2		
DOC_TYPTAB_DECIMAL%	NS_WP20	TDECIMAL_R	EXECUTED
	NS_WPL2		
DOC_TYPTAB_LEFT%	NS_WP20	TLEFT_R	EXECUTED
	NS_WPL2		
DOC_TYPTAB_RIGHT%	NS_WP20	TRIGHT_R	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_UNDO	NS_WP2	MI_UNDO1	EXECUTED
DOC_UNITS2XPEL%	NS_WPDRG		
DOC_UNITS2YPEL%	NS_WPDRG		
DOC_UPDATE	NS_WP2	MI_NUMBERED	EXECUTED
	NS_WP14	BACKLIST	SELECTED
	NS_WP14	FORELIST	SELECTED
	NS_WP14	LINELIST	SELECTED

	NS_WP18	BP_APPLY	EXECUTED
	NS_WPL2		
	NS_WPL3		
DOC_USER_MESSAGE_0	NS_WPL2		
DOC_USER_MESSAGE_1	NS_WPL2		
DOC_USER_MESSAGE_2	NS_WPL2		
DOC_USER_MESSAGE_3	NS_WPL2		
DOC_USER_MESSAGE_10	NS_WPL		
DOC_WIN_BODY%	NS_WP2	MI_BODY1	EXECUTED
	NS_WP2	M_VIEW1	INIT
	NS_WP20	BMP_BODY	SELECTED
	NS_WPL2		
DOC_WIN_FIRSTFOOTER%	NS_WP2	M_VIEW1	INIT
	NS_WP20	-	USER1
	NS_WP20	-	WINDOW
DOC_WIN_FIRSTHEADER%	NS_WP2	M_VIEW1	INIT
	NS_WP20	-	USER1
	NS_WP20	-	WINDOW
DOC_WIN_FOOTER%	NS_WP2	MI_FOOTER1	EXECUTED
	NS_WP2	M_VIEW1	INIT
	NS_WP20	BMP_FOOTER	SELECTED
	NS_WP20	-	USER1
	NS_WP20	-	WINDOW
	NS_WPL2		
DOC_WIN_HEADER%	NS_WP2	MI_HEADER1	EXECUTED
	NS_WP2	M_VIEW1	INIT
	NS_WP20	BMP_HEADER	SELECTED
	NS_WP20	-	USER1
	NS_WP20	-	WINDOW
	NS_WPL2		
DOC_XPEL2UNITS%	NS_WPDRG		
DOC_XTABS	NS_WPL3		
	NS_WPLU		
DOC_YPEL2UNITS%	NS_WPDRG		

LIMITATIONS DE NS-WP

Les caractères acceptés sont ceux supérieurs ou égaux à l'espace.

Le nombre maximum d'entrées dans la liste des polices est 254.

Le nombre maximum de tabulations et de traits verticaux pour un paragraphe est 64. 256 caractères peuvent être affichés sur une ligne, 32 images bitmaps peuvent être affichées sur une ligne. 128 changements de style de caractère sont possibles dans une même ligne.

Un paragraphe a au maximum 1024 lignes. Le nombre maximum de pages et le nombre maximum de paragraphes d'un document dépend de la puissance de la machine. Ces nombres sont compris en général entre 1000 et 10000.

Un document peut comprendre au maximum 512 sections.

Le presse-papiers n'est pas limité en taille, mais par contre les sauts de section ne sont pas dupliqués.

FORMAT DE FICHIER NS-WP

Introduction

Un document ayant le format NS-WP est un fichier texte constitué de commandes et de texte.

Il est donc possible de créer un tel document avec n'importe quel éditeur de texte et pas seulement avec un traitement de texte réalisé avec NS WP.

Comme tout fichier texte, un document NS WP est constitué d'un ensemble de lignes, chaque ligne étant séparée par la séquence CR (retour chariot ou caractère ASCII 13) et LF (saut de ligne ou caractère ASCII 10). Une ligne est constituée de caractères imprimables (caractères ASCII supérieurs à 31), du caractère ESPACE et du caractère TAB (tabulation horizontale ou caractère ASCII 9).

Il existe deux catégories de commandes : les commandes simples et les commandes groupes.

Les commandes simples

Une commande simple a un nombre figé de paramètres de natures différentes.
Sa syntaxe est la suivante :

```
/NOM-COMMANDE PARAMETRES
```

Il est possible d'écrire une commande simple sur plusieurs lignes en plaçant le caractère \ à la fin de chaque ligne qui se continue.

Exemple d'une commande simple qui tient sur une ligne :

```
/DOCUMENT EXEMPLE
```

Cette commande indique le début du document EXEMPLE.

Exemple d'une commande simple qui tient sur plusieurs lignes :

```
/PARAGRAPH=CENTERED,100,100,1300,TC:TLDOT:500,TLUNDE:1000, \  
TR:TLDOT:1500
```

Cette commande définit les caractéristiques d'un paragraphe.

Les commandes groupes

Une commande groupe a pour paramètres plusieurs données de même nature.
Sa syntaxe est la suivante :

```
{  
/ {NOM-COMMANDE  
DONNEE-1  
...  
DONNEE-N  
/  
}
```

Structure du document

La structure d'un document est la suivante :

Informations générales
Palette de couleurs
Polices
Jeu de caractères
Marges
Tabulations par défaut
Système de mesure
Contrôle veuve/orpheline
Stockage du dictionnaire
Chargement des images

Section 1	
En-tête première page	Paragraphe 1 ... Paragraphe N
En-tête	Paragraphe 1 ... Paragraphe N
Bas de page première page	Paragraphe 1 ... Paragraphe N
Bas de page	Paragraphe 1 ... Paragraphe N
Paragraphe 1 ... Paragraphe N	
Section N	
En-tête première page	Paragraphe 1 ... Paragraphe N

En-tête	Paragraphe 1 ... Paragraphe N
Bas de page première page	Paragraphe 1 ... Paragraphe N
Bas de page	Paragraphe 1 ... Paragraphe N
Paragraphe 1 ... Paragraphe N	

Voici les descriptions de ces différentes commandes :

```
{/BORDER=<attribut-cadre>:<val-attribut-cadre> [,<attribut-cadre>:<val-attribut-cadre>...]  
<id-ligne>=<attribut-ligne>:<val-attribut-ligne> [,<attribut-ligne>:<val-attribut-ligne>...]  
<id-ligne>=<attribut-ligne>:<val-attribut-ligne> [,<attribut-ligne>:<val-attribut-ligne>...]  
...  
/}
```

Définit l'encadrement d'un paragraphe.

Paramètres :

<attribut-cadre> : caractéristique du cadre. Il peut prendre quatre valeurs :

- BRSP : espace entre le texte et le cadre
- BRPAT : motif de fond de cadre
- CF : couleur d'avant-plan
- CB : couleur d'arrière-plan

<val-attribut-cadre> : valeur numérique de l'attribut

<id-ligne> : identificateur d'une ligne du cadre. <id-ligne> peut prendre six valeurs :

- BRDRL : ligne gauche
- BRDRR : ligne droite
- BRDRT : ligne du haut
- BRDRB : ligne du bas
- BRDRTW : ligne séparant les paragraphes
- BRDRS : ombre

<attribut-ligne> : caractéristique d'une ligne. <attribut-ligne> peut prendre trois valeurs :

- BRLWDTH : épaisseur

- BRLKIND : type de la ligne (simple ou double)
- BRPAT : couleur de la ligne

<val-attribut-ligne> : valeur numérique de l'attribut

Les épaisseurs, largeurs, hauteurs sont définies dans l'unité courante.

Exemple :

```
{BORDER=BRSP:0,CF:1,CB:1
/BRDRL=BRLWDTH:5,BRLKIND:0,CL:1
/BRDRR=BRLWDTH:5,BRLKIND:0,CL:1
/BRDRT=BRLWDTH:5,BRLKIND:0,CL:1
/BRDRB=BRLWDTH:5,BRLKIND:0,CL:1
/}
```

```
/CHARSET=<jeu-de-caractères>
```

Définit le jeu de caractères utilisé.

Paramètres :

<jeu-de-caractères> vaut une des valeurs : ANSI, PC ou MAC

Exemple :

```
/CHARSET=ANSI
```

```
{COLORTBL
<définition noir>
<définition bleu>
<définition vert>
<définition cyan>
<définition rouge>
<définition magenta>
<définition marron>
<définition gris clair>
<définition gris foncé>
<définition bleu clair>
<définition vert clair>
<définition cyan clair>
<définition rouge clair>
<définition magenta clair>
<définition jaune>
<définition blanc>
<définition couleur de fond>
<définition neutre>
/}
```

Décrit la palette de couleurs.

Paramètres :

<définition couleur> : définition de la couleur à partir des trois couleurs fondamentales : le rouge, le vert et le bleu.

- <rouge> : valeur entre 0 et 255 (0 : rouge foncé ; 128 : rouge ; 255 : rouge clair)
- <vert> : valeur entre 0 et 255
- <bleu> : valeur entre 0 et 255

```
/DEFTAB <valeur>
```

Spécifie l'espacement séparant les taquets de tabulation positionnés par défaut.

Paramètres :

<valeur> : espacement en pixels entre deux taquets de tabulation.

Si <valeur> = 0, aucune tabulation par défaut n'est définie.



```
/DOCUMENT <Nom-du-document>
```

Indique le début du document. Cette commande doit obligatoirement être placée en première ligne du document.

Paramètres :

<Nom-du-document> est le nom du fichier contenant le document.

Exemple :

```
/DOCUMENT EXEMPLE
```

```
/ENDDOCUMENT
```

Marque la fin du document. Aucune commande ou aucun texte ne peut suivre cette commande.

```
/ENDFIRSTFOOTER
```

Marque la fin du bas de page de la première page du document.

```
/ENDFOOTER
```

Marque la fin du bas de page du document.

```
/ENDHEADER
```

Marque la fin de l'en-tête du document.

/ENDPARAGRAPH

Marque la fin d'un paragraphe.

/ENDSECTION

Marque la fin d'une section.

/FIRSTHEADER

Indique le début de l'en-tête de la première page du document.

Un en-tête peut comprendre des paragraphes et des sauts de ligne.

/FIRSTFOOTER

Indique le début du bas de page de la première page du document.

Un bas de page peut comprendre des paragraphes et des sauts de ligne.

/FONT, <no-style>=<nom-fonte>, <taille-fonte> [,<attribut>, <attribut>...]

Définit un style de caractères.

Paramètres :

<no-style> : numéro unique associé au style (entre 1 et 254). Ce numéro servira dans la suite du document à désigner le style.

<nom-fonte> : nom de la police (par exemple Courier).

<taille-fonte> : corps de la police (par exemple 10).

<attribut> : attribut de représentation du style. <attribut> peut prendre une des valeurs :

- BOLD : attribut gras
- ITALIC : attribut italique
- CF:<index> : couleur des caractères
- CB:<index> : couleur de fond
- CL:<index> : couleur du souligné et du barré
- USER:<nb> : style utilisateur

<index> : index sur la palette des couleurs

Le style numéro 1 doit être obligatoirement présent car il représente le style utilisé par défaut. Les espaces sont significatifs dans le nom d'une fonte. Ainsi, Tms Rmn n'est pas équivalent à TmsRmn.

Exemple :

```
/FONT,001=Tms Rmn,10
/FONT,002=Tms Rmn,10,BOLD
/FONT,003=Tms Rmn,10,BOLD,ITALIC
/FONT,004=Courier,10
```

```
/FOOTER
```

Indique le début du bas de page du document.

Un bas de page peut comprendre des paragraphes et des sauts de ligne.



```
/HEADER
```

Indique le début de l'en-tête du document.

Un en-tête peut comprendre des paragraphes et des sauts de ligne.



```
/LINEBREAK
```

Passe à la ligne lors de l'impression.

```
{/LINETABS
/LTAB=<caract>:<val-caract> [, <caract>:<val-caract>...]
...
/}
```

Définit les tabulations verticales.

Paramètres :

<caract> : caractéristique d'une ligne verticale. <caract> peut prendre quatre valeurs :

- LPOS : position de la tabulation sur la règle
- BRLWDTH : largeur de la ligne
- BRLKIND : type de la ligne
- CL : couleur de la ligne

<val-caract> : valeur numérique de la caractéristique

Exemple :

```
{/LINETABS
/LTAB=LPOS:4000,BRLWDTH:10,BRLKIND:0,CL:0
/}
```

```
/LOMETRICS
```

Indique que le système de mesure pris en compte est le système métrique.

```
/LOENGLISH
```

Indique que le système de mesure pris en compte est le système anglo-saxon.

```
/MARGINS <gauche>, <droite>, <haute>, <basse>
```

Définit les marges du document (en plus des marges définies par le driver d'imprimante).

Paramètres :

<gauche> : marge par rapport au bord gauche de la feuille.

<droite> : marge par rapport au bord droit de la feuille.

<haute> : marge en haut de la feuille.

<basse> : marge en bas de la feuille.

```
{/MARKERCLASS:<id-dict>=<nom-dict>
/{DATA:<numéro>
<donnée>
/}
[/{DATA:<numéro>
<donnée>
/}
... ]
/}
```

Décrit un dictionnaire de données.

Paramètres :

<id-dict> : identificateur du dictionnaire

<nom-dict> : nom du dictionnaire

<numéro> : numéro d'une donnée

<donnée> : donnée

Exemple :

```
{/MARKERCLASS:001=CL_STYLE
/{DATA:001
DEFAULT
/}
```

```
/PAGEBREAK
```

Insère un saut de page lors de l'impression.

```

/PARAGRAPH=<alignement>
[, BREAK]
[, USERGLO:<nombre>]
[, USERLOC:<nombre>]
[, <marge-gauche-prem>, <marge-gauche>, <marge-droite>]
[, <tab-alignement><points-de-suite>]<val-tab>
[, <tab-alignement><points-de-suite>]<val-tab>
...

```

Indique le début d'un paragraphe.

Paramètres :

<alignement> : type d'alignement souhaité pour le paragraphe. Ce type a une des valeurs suivantes :

- LEFT (aligné à gauche),
- RIGHT (aligné à droite),
- CENTERED (centré),
- FULL (justifié).
- BREAK : force un saut de page avant l'impression de la première ligne du paragraphe.

<marge-gauche-prem> : marge gauche pour la première ligne.

<marge-gauche> : marge gauche pour les lignes suivantes.

<marge-droite> : marge droite.

<tab-alignement> : alignement d'une tabulation. <tab-alignement> peut prendre les valeurs :

- TR (alignée à droite),
- TC (centrée),
- TDEC (alignée sur le point décimal).

Lorsque <tab-alignement> n'est pas spécifié, la tabulation est alignée à gauche.

<points-de-suite> : type de points de suite associé à la tabulation. <points-de-suite> peut prendre les valeurs :

- TLNONE (pas de points de suite),
- TLDOT (caractère .),
- TLHYPH (caractère -),
- TLUND (caractère _).

<val-tab> : position de la tabulation.

Notes :

1. Les différentes marges ou tabulations sont définies dans l'unité courante.
2. Le texte d'un paragraphe (contenu entre une commande /PARAGRAPH et une commande /ENDPARAGRAPH) peut comporter plusieurs lignes. A l'impression, le saut de ligne entre deux lignes est automatiquement remplacé par un espace. Il n'est généré par le traitement de texte que pour éviter de stocker des lignes trop longues (par exemple, supérieures à 254 caractères, limite actuelle du produit).
3. Un texte peut comprendre des séquences :
 - a) /xxx

où xxx est un numéro de fonte. Cette fonte est utilisée pour imprimer le texte à partir de cette séquence.

b) /T

Il s'agit d'une borne placée dans le texte.

Lorsque le texte est affiché dans la fenêtre NS-WP, l'appui sur [Ctrl]+[Tab] permet de positionner le curseur sur le prochain caractère "marqué", et ainsi de suite de borne en borne jusqu'à la fin du texte. Employé par exemple sur un texte Read Only et avec une fonte ayant l'attribut UNLOCKED, cela permet de positionner rapidement le curseur sur le champ non bloqué suivant.

4. Le caractère / est doublé dans un texte (//) pour éviter toute confusion avec le préfixe d'une commande ou d'une séquence.

Exemple 1 :

```
/PARAGRAPH = LEFT,100,100,1600
Cher /002Monsieur/001,
/ENDPARAGRAPH
/PARAGRAPH = LEFT,100,100,1600
/ENDPARAGRAPH
/PARAGRAPH = LEFT,100,100,1600
Veuillez agréer l'expression de mes sentiments
distingués....
/ENDPARAGRAPH
```

Le texte suivant sera imprimé (en supposant que le style 2 est un style gras et souligné) :

Cher Monsieur,

Veuillez agréer l'expression de mes sentiments distingués....

Exemple 2 :

```
/PARAGRAPH=CENTERED,100,100,1300,TC:TLDOT:500,TLUNDE:1000, \
TR:TLDOT:1500
```

```
{/PICT:<id-bitmap>:2:0:0:0:0:0:<largeur>:<hauteur>:<pos-base>=<BMP:<nom-
image>>/}
```

Insère une image bitmap dans le texte. Cette image doit avoir été précédemment chargée en mémoire.

Paramètres :

<id-bitmap> : identificateur attribué à l'image bitmap lors de son chargement

<largeur> : largeur en pixels de l'image bitmap

<hauteur> : hauteur en pixels de l'image bitmap

<pos-base> : position de l'image bitmap par rapport à la ligne de base

<nom-image> : chemin, nom et extension du fichier contenant l'image bitmap.

<nom-image> est affiché dans le texte à la place de l'image lorsque l'utilisateur demande à visualiser les codes des commandes. <nom-image> n'a pas d'autre utilité.

Exemple :

```
{PICT:0:2:0:0:0:0:0:76:76:0=<BMP:C:\WINDOWS\ZIGZAG.BMP>}
```

```
{PICTDFN:<id-bitmap>=<nom-image>  
}
```

Charge en mémoire une image bitmap.

Paramètres :

<id-bitmap> : identificateur de l'image bitmap.

<nom-image> : chemin, nom et extension du fichier contenant l'image bitmap

Une seule image bitmap peut être actuellement chargée en mémoire avec PICTDFN.

Exemple :

```
{PICTDFN:0=C:\WINDOWS\ZIGZAG.BMP  
}
```

```
/SECTION=<description> [, <description>]
```

Indique le début d'une section.

Paramètres :

<description> : propriété de la section. <description> peut prendre une des valeurs :

- PORTRAIT : orientation du papier en mode portrait
- TOP : texte aligné verticalement sur le haut de page
- CENTERED : texte centré entre le haut et le bas de page
- FULL : texte justifié entre le haut et le bas de page
- BOTTOM : texte aligné verticalement sur le bas de page

Les quatre dernières propriétés s'excluent mutuellement.

```
/WIDOWCTRL
```

Positionne le contrôle des lignes veuves et orphelines.

FORMAT DE FICHIER EN FUSION

Introduction

Un fichier de fusion est un fichier texte, au format NS-WP, qui respecte une syntaxe précise et qui, à travers l'appel à l'instruction DOC_MERGE, permet la génération conditionnelle d'un nouveau fichier.

Rappelons la syntaxe de DOC_MERGE :

```
DOC_MERGE handle-WP, code-erreur, ligne-erreur,  
nom-fichier-cmde, nom-fichier-résult
```

où :

- handle-WP désigne le handle de la fenêtre NS-WP qui sera appelée pour renseigner les valeurs des variables (utilisé uniquement si la commande /ASK est utilisée),
- code-erreur est un code d'erreur en retour de l'appel,
- ligne-erreur est le numéro de ligne dans le fichier de commandes où l'erreur a été diagnostiquée,
- nom-fichier-cmde est le nom du fichier NS-WP de commandes, en entrée,
- nom-fichier-résult est le nom du fichier NS-WP résultat de la fusion, en sortie.

Cette annexe décrit la syntaxe que doit respecter le fichier de commandes, indiqué dans l'avant-dernier paramètre "nom-fichier-cmde" de DOC_MERGE.

Ce fichier est un fichier texte au format NS-WP, dont certaines lignes commencent par une commande, dont voici la liste :

```
/ASK /IF  
/BREAK /INCLUDE  
/CALL /LABEL  
/COMMENT /LOOP  
/CONTINUE /MOVE TO  
/DEFINE /PUT  
/ELSE (avec /IF) /REPEAT  
/ELSEIF (avec /IF) /RETURN  
/ENDIF (avec /IF) /STOP  
/ENDLOOP (avec /LOOP) /UNTIL (avec /REPEAT)  
/ENDWHILE (avec /WHILE) /WHILE  
/GOTO
```

Si une ligne ne commence pas par une de ces commandes (ou par une des commandes du format NS-WP, comme par exemple /FONT), la ligne est simplement insérée dans le fichier destination.

Certaines commandes nécessitent une expression (/ELSEIF, /IF, /MOVE, /PUT, /UNTIL, /WHILE). Une expression correspond aux expressions reconnues par NCL, pouvant comporter les opérateurs et fonctions intrinsèques standards du NCL (voir Manuel de Programmation NCL).

Opérateurs acceptés :

```
+, -, *, /, %, &, &&, <, <=, >, >=, <>, =, AND, NOT, OR, XOR, ABS, NUM,
INT, STRING, BAND, BNOT, BXOR, BOR, SKIP, LSKIP, RSKIP, UPCASE, LOWCASE,
LOW, HIW, LOB, HIB, LENGTH, ().
```

Fonctions acceptées :

```
INSERT$, COPY$, DELETE$, POS%, FILLER$, STRING$, ASC%, CHR$.
```

Pour créer un fichier de fusion, il est conseillé de le créer initialement à l'aide du traitement de texte NS-WP, afin qu'il respecte déjà le format NS-WP (/DOCUMENT, /FONT, ...), puis de le compléter à l'aide d'un éditeur ASCII avec des commandes de fusion.

```
/ASK <var-name>
```

Envoie à la fenêtre dont le handle est dans handle-WP la demande de valeur pour la variable <var-name>. Le message reçu est le message NS-DK/NatStar USER1 avec dans PARAM1\$ le nom de la variable. La variable PARAM2\$ est utilisée pour rendre la valeur associée à la variable. <var-name> doit avoir été définie par l'instruction /DEFINE.

Exemple :

```
; Dans le fichier de fusion :
/ASK TOTO$

; Dans l'événement USER1 de NS-DK/NatStar
IF PARAM1$ = 'TOTO$'
MOVE 'Dupont' TO PARAM2$
ENDIF
/BREAK
; Permet de sortir d'une boucle (/WHILE/ENDWHILE, /LOOP/ENDLOOP ou
/REPEAT/UNTIL).
/CALL <label>
; Provoque l'appel vers le label <label>. Il sera possible de revenir de la
ligne qui suit le /CALL grâce à /RETURN.
/COMMENT ...
; Indique un commentaire. Le reste de la ligne est ignoré.
/CONTINUE
; Permet de sauter à la boucle suivante.
/DEFINE <var-name>
; Définit une variable qui pourra être utilisée dans une expression. La
variable peut être entière (terminée par le caractère %), réelle (terminée
par le caractère f ou #) ou de type chaîne de caractères (terminée par le
caractère $).
/ELSE
; Voir /IF.
/ELSEIF <expression>
; Voir /IF.
/ENDIF
; Voir /IF.
/ENDLOOP
; Voir /LOOP.
/ENDWHILE
; Voir /WHILE.
```

```

/GOTO <label>
; Déroute l'exécution vers le label <label>.
/IF <expression-A>
... (A)
/ELSEIF <expression-B>
... (B)
/ELSE
... (C)
/ENDIF

```

Permet de sélectionner en fonction des valeurs retournées par <expression> la portion de texte retenue.

La partie (A) est retenue si l'expression <expression-A> retourne une valeur non nulle.

Sinon, la partie (B), si présente, est retenue si l'expression <expression-B> retourne une valeur non nulle.

Enfin, la partie (C), si présente, est retenue si aucune des expressions ne retourne une valeur non nulle.

```

/INCLUDE <nom-de-fichier>

```

Inclut le fichier <nom-de-fichier> à la ligne courante et continue l'analyse.

Exemple :

```

/INCLUDE C:\FICHER.DOC
/LABEL <label>
; Définit un label (entier entre 1 et 32767) qui pourra être utilisé par
l'instruction /GOTO ou /CALL. Un label ne peut être à l'intérieur d'une
boucle.
/LOOP
... (A)
/ENDLOOP
; Permet de répéter infiniment le texte (A) compris entre les commandes
/LOOP et /ENDLOOP. Pour sortir de cette boucle, il faut appeler soit
l'instruction /BREAK soit l'instruction /GOTO.
/MOVE <expression> TO <var-name>
; Stocke dans la variable <var-name> l'expression <expression>. <var-name>
doit avoir été définie par l'instruction /DEFINE.
/PUT <expression>
; Stocke dans le texte l'expression <expression>.
/REPEAT
... (A)
/UNTIL <expression>
; Permet de répéter le texte (A) compris entre les commandes /REPEAT et
/UNTIL jusqu'à ce que l'expression <expression> retourne une valeur non
nulle.
/RETURN
/RETURN <label>
; Provoque le retour à la ligne qui suit une instruction /CALL préalable ou
à une ligne précise. Le <label> doit se situer dans le même fichier que
celui contenant la commande /RETURN ou dans un fichier actuellement chargé.
/STOP
; Provoque le déroutement vers la fin du fichier.
/UNTIL

```

```
; Voir /REPEAT.  
/WHILE <expression>  
... (A)  
/ENDWHILE  
; Permet de répéter le texte (A) compris entre les commandes /WHILE et  
/ENDWHILE jusqu'à ce que l'expression <expression> retourne une valeur  
nulle.
```

Exemple de fusion

Fichier fusion "COMMANDS.DOC"

```

/DOCUMENT COMMANDS
/FONT,001=Helv,10
/FONT,002=Helv,10,BOLD
/DEFINE OK%
/DEFINE SEXE%
/DEFINE NOM$
/WHILE 1
/ASK OK%
/IF NOT OK%
/BREAK
/ENDIF
/PARAGRAPH = LEFT,100,100,1600
/ASK SEXE%
/ASK NOM$
/IF SEXE% = 0
Cher Monsieur
/ELSEIF SEXE% = 1
Chère Madame
/ELSE
Chère Mademoiselle
/ENDIF
/PUT NOM$
'
/ENDPARAGRAPH
/PARAGRAPH = LEFT,100,100,1600
....
/ENDPARAGRAPH
/PARAGRAPH = LEFT,100,100,1600
Veuillez agréer l'expression de mes sentiments distingués.
/ENDPARAGRAPH
/PARAGRAPH = LEFT,100,100,1600,800
La direction.
/ENDPARAGRAPH
/PAGEBREAK
/ENDWHILE
/ENDDOCUMENT
Script NCL de fusion

LOCAL ERRCODE%(2), ERRLINE%(2)

; Initialisation d'une recherche NS-DB
DB_SEARCH HDB%, PERSONNE WHERE NOM >= ''

; Fusion
DOC_MERGE HWP%, ERRCODE%, ERRLINE%, \
'COMMANDS.DOC', 'OUTPUT.DOC'
IF ERRCODE% <> DOC_ERRNO%
MESSAGE 'Erreur' && ERRCODE%,
'La fusion a échoué ligne' && ERRLINE%
ENDIF
Script NCL de l'événement USER1 de la fenêtre HWP% :
```

```

; Test du nom de variable demandé par /ASK dans PARAM1$
; et retour de sa valeur dans PARAM2$
IF PARAM1$ = 'OK%'
MOVE (DB_ERROR% (HDB%) = 0) TO PARAM2$ ; TRUE% ou FALSE%
ELSEIF PARAM1$ = 'SEXE%'
MOVE PERSONNE.SEXE TO PARAM2$
ELSEIF PARAM1$ = 'NOM%'
MOVE PERSONNE.NOM TO PARAM2$
ELSE
MESSAGE 'Erreur !','NOM Invalide :' && PARAM1$
ENDIF

; Recherche de l'enregistrement suivant
DB_NEXT HDB%, PERSONNE

```

RACCOURCIS CLAVIER

Dès qu'un utilisateur lance une fenêtre de traitement de texte de type NS WP, il bénéficie de raccourcis clavier préprogrammés :

[Del] Supprime le caractère courant

[Insert] Gère le mode d'insertion

[Shift]+[Del] Coupe le texte sélectionné

[Ctrl]+[Ins] Copie le texte sélectionné dans le presse-papiers

[Shift]+[Ins] Colle le texte du presse-papiers

[Backspace] Supprime le caractère précédent

[Ctrl]+[T] Donne le focus à la barre d'outils standard

- Déplace le curseur sur le (la) :

[Right] Caractère suivant

[Left] Caractère précédent

[Down] Ligne suivante

[Up] Ligne précédente

[Home] Début de la ligne

[End] Fin de la ligne

[PageDown] Ecran suivant

[PageUp] Ecran précédent

[Ctrl]+[Right] Mot suivant

[Ctrl]+[Left] Mot précédent

[Ctrl]+[Home] Début du document

[Ctrl]+[End] Fin du document

[Ctrl]+[Tab] Borne suivante

[Ctrl]+[Shift]+[Tab] Borne précédente

- Sélectionne le (la) :

[Shift]+[Right] Caractère suivant

[Shift]+[Left] Caractère précédent

[Shift]+[Down] Ligne suivante

[Shift]+[Up] Ligne précédente

[Shift]+[Home] Début de la ligne

[Shift]+[End] Fin de la ligne

[Shift]+[PageDown] Ecran suivant

[Shift]+[PageUp] Ecran précédent

[Shift]+[Ctrl]+[Right] Mot suivant

[Shift]+[Ctrl]+[Left] Mot précédent

[Shift]+[Ctrl]+[Home] Début du document

[Shift]+[Ctrl]+[End] Fin du document

- Insère un(e) :

[Tab] Tabulation

[Enter] Retour chariot

[Ctrl]+[P] Saut de page forcé

- Met en :

[Ctrl]+[B] Gras

[Ctrl]+[I] Italique

[Ctrl]+[U] Souligné

[Ctrl]+[S] Standard (non gras, non italique, non souligné)

- Aligne :

[Ctrl]+[L] A gauche

[Ctrl]+[R] A droite

[Ctrl]+[C] En centrant

[Ctrl]+[J] En justifiant

- Affiche :

[Ctrl]+[H] L'en-tête

[Ctrl]+[F] Le bas de page

[Ctrl]+[K] Le corps du document

GLOSSAIRE

A

Alignement: Positionnement horizontal d'un paragraphe. Un paragraphe peut être aligné à gauche, à droite, centré ou justifié.

Alignement vertical: Positionnement vertical des paragraphes sur une page. Le texte de la page peut être aligné sur le haut de la page, sur le bas de la page, centré ou justifié.

Attributs d'enrichissement: Attributs modifiant la présentation d'un caractère : gras, italique, souligné...

B

Barre d'outils: Barre graphique avec des boutons et des champs qui permettent d'accéder rapidement aux fonctionnalités les plus courantes du traitement de texte.

C

Centré (paragraphe): Les lignes sont centrées par rapport aux limites gauche et droite du paragraphe, sans les justifier ni à droite ni à gauche.

Centrée (page): Les paragraphes sont centrés par rapport aux limites haute et basse de la page, sans les justifier ni en haut ni en bas.

Coller: Action de reprendre le contenu du presse-papiers pour l'insérer dans le texte.

Copier: Action de mettre la sélection de texte dans le presse-papiers, sans effacer la sélection.

Corps: Texte même du document. Ce texte est organisé en paragraphes, un groupe plus ou moins important de paragraphes constituant une page.

Corps (d'un caractère): Valeur numérique définissant la hauteur et la largeur d'un caractère.

Couper: Action de mettre la sélection de texte dans le presse-papiers, puis d'effacer la sélection.

Curseur: Tiret clignotant vertical qui apparaît dans le texte édité et qui indique l'emplacement auquel sera écrit le prochain caractère.

E

Effacer: Détruit la sélection de texte, sans la mettre dans le presse-papiers.

En-tête: Texte se trouvant en haut de chaque page d'un document.

F

Format de fichier: Format selon lequel le texte est stocké dans un fichier. Ex : RTF (Rich Text Format).

Formatage de caractères: Mise en forme au moyen de polices, d'attributs, de couleurs.

Formatage de paragraphes: Mise en forme au moyen de retraits, de tabulations, d'alignements, de bordures.

Fusion: Combinaison d'un document principal et d'un fichier de données. Le document principal contient un texte standard (par exemple le texte d'une lettre type), le fichier de données contient un texte variable (par exemple un nom, une adresse). Le document final combine le texte standard et les données.

J

Justifié (paragraphe): Le texte est aligné par rapport aux limites du paragraphe en agissant sur l'espacement entre les caractères.

Justifiée (page): Les paragraphes sont alignés par rapport aux limites haute et basse de la page en agissant sur l'espacement entre les lignes.

L

Liaisons: Paragraphes ou lignes qui à l'impression seront obligatoirement sur la même page.

M

Modes d'affichage: Différents types de présentation à l'écran d'un document. Le mode "brouillon" utilise peu de graphisme mais est rapide. Le mode "page" donne un résultat proche de ce qui sera obtenu à l'impression, mais est plus lent. Le mode "standard" donne un résultat intermédiaire, aussi bien en qualité d'affichage qu'en performances.

P

Page: Page physique : dimension de la page telle qu'elle est spécifiée au niveau de la configuration de l'imprimante. Page d'un document : zone éditabile maximale (page physique diminuée des marges de l'imprimante si présentes). Cette zone peut être restreinte à un cadre défini par les marges haute, basse, gauche et droite du document.

Paragraphe: Tout texte dont la saisie s'est terminée par un retour chariot explicite.

Partie de document: En-tête, corps ou pied de page d'un document.

Pied de page: Texte se trouvant en bas de chaque page d'un document.

Point d'insertion: Tiret clignotant vertical qui apparaît dans le texte édité et qui indique l'emplacement auquel sera écrit le prochain caractère.

Points de suite: Caractères affichés à l'emplacement d'une tabulation. Ces caractères sont souvent des tirets ou des points.

Police: Forme d'un jeu de caractères. Ex : Tms Rmn, Helvetica, ...

Presse-papiers: Mémoire temporaire permettant d'échanger des informations entre applications ou au sein d'une même application, utilisée grâce aux Couper / Copier / Coller.

R

Règle: Barre graphique affichée dans la partie supérieure d'une fenêtre de traitement de texte. Elle permet de placer des taquets de tabulation, de définir des marges et des retraits de paragraphe avec la souris.

Retrait de première ligne: Distance entre la marge gauche de la page et le bord gauche de la première ligne du paragraphe.

Retrait droit: Distance entre la marge droite de la page et le bord droit du paragraphe.

Retrait gauche: Distance entre la marge gauche de la page et le bord gauche du paragraphe.

S

Saut de page: Séparateur placé entre deux pages du document.

Saut de section: Séparateur placé entre deux sections du document.

Section: Sous-division d'un document. Le découpage en sections permet de faire varier les en-têtes et les pieds de page d'un document, ainsi que son alignement vertical.

Solitaires (lignes): Lignes qui à l'impression seront obligatoirement sur la même page.

Solitaires (paragraphe): Paragraphes qui à l'impression seront obligatoirement sur la même page.

Style de caractère: Mise en forme d'un caractère : sa police, sa taille, ses attributs, ses couleurs. Appliquer un style à une zone de texte confère à tous ses caractères la présentation décrite dans le style.

Style de paragraphe: Caractéristiques de présentation d'un paragraphe : le style des caractères qui le composent et sa mise en forme (retraits, tabulations, alignements, bordures, liaisons). Appliquer un style à une zone de texte confère à tous ses paragraphes la présentation décrite dans le style.

T

Tabulation: Espace permettant d'aligner un paragraphe à un emplacement précis. Cet espace est placé au moyen de la touche [Tab]. L'emplacement est déterminé par un taquet de tabulation.

Taille: Valeur numérique définissant la hauteur et la largeur d'un caractère.

Taquet de tabulation: Position où sera aligné le paragraphe au moyen d'une tabulation.

U

Unité de mesure: Unité utilisée pour définir les marges, les tabulations et les retraits.

V

Veuve/Orpheline (contrôle): Empêche d'imprimer la dernière ligne d'un paragraphe seule en haut d'une page ou la première ligne d'un paragraphe seule en bas d'une page.

INDEX

@

@DATE 21

@PAGE 21

@TIME 21

{

{NS02WP} 8

A

Affichage en mode 44

brouillon 44

page 18, 44

standard 18, 44

Alignement 29, 47

Alignement décimal 26

Alignement vertical 25, 46

Attributs d'enrichissement 32, 33

B

Barre d'outils 15, 38, 348

Bloquage de la mise à jour 44

Boîte info 8

Bordures 30

Bornes 36

C

Caractères spéciaux 20, 44

Couleurs des caractères 32, 33

D

Date 21

Dictionnaires 37

DOC_ADDPICTDFN% 107

DOC_ALLOCATEDMEM% 108

DOC_ALLOCSCANCONTEXT% 109

DOC_ASSOCIATEPRINTER 210

DOC_BORDER 327

DOC_CHANGE% 110

DOC_CHARS% 95, 180

DOC_CLEAR_TAG 212

DOC_COPY 213

DOC_COPYSELECTION 214

DOC_CURRENTPARAGRAPH% 112

DOC_CUT 215

DOC_DEFAULTFONT 216

DOC_DEFAULTLOCKFONTS 217

DOC_DEFAULTMAXPARAWIDTH
218

DOC_DEFAULTMETRICS 219

DOC_DEFAULTPARAGRAPHINDENT
220

DOC_DEFAULTTTAB 221

DOC_DEFAULTWIDOWCONTROL
222

DOC_DELETE 223

DOC_DELETEBITMAPCHAR% 113

DOC_DELETECHARS 224

DOC_DELPICITDFN% 114

DOC_DISPOSEWINDOW 225

DOC_ERRBADPARAM% 86

DOC_ERRBADPICTMETRICS% 86

DOC_ERRCANNOTCLOSEFILE% 86

DOC_ERRCANNOTCREATEFILE%
86

DOC_ERRCANNOTOPENFILE% 86

DOC_ERRCANNOTREADFILE% 86

DOC_ERRCANNOTWRITEONFILE%
86

DOC_ERRDUPLPICTNAME% 86

DOC_ERRFILENOTFOUND% 86

DOC_ERRINTERNAL% 86

DOC_ERRINVALIDFILE% 86

DOC_ERRINVALIDFILEFORMAT% 86

DOC_ERRINVALIDPICT% 86

DOC_ERRINVALIDVARIABLE% 86

DOC_ERRLOCKED% 86

DOC_ERRNO% 86

DOC_ERRNOPICTATCURSPOS% 86

DOC_ERRNOTAPICTID% 86

DOC_ERRNOTAPICTNAME% 86

DOC_ERRNOTENOUGHMEMORY%
86

DOC_ERRNOTENOUGHSYSRESOU
RCES% 86

DOC_ERROR% 115

DOC_ERRPARAGRAPHOVERFLOW
% 86
 DOC_ERRREPDEFAULT% 86
 DOC_ERRREPOPEN% 86
 DOC_ERRREPSTART% 86
 DOC_ERRSTACKOVERFLOW% 86
 DOC_ERRSYNTAXERROR% 86
 DOC_ERRTOOMANYINCLUDES% 86
 DOC_ERRUNKNOWNVARIABLE% 86
 DOC_EXTPRINT% 116
 DOC_EXTPRINTFILE% 118
 DOC_EXTPRINTSTART% 120
 DOC_EXTPRINTSTOP 226
 DOC_FILE_FORMAT% 121
 DOC_FILE_NSWP% 88
 DOC_FILE_RFT% 88
 DOC_FILE_RTF% 88
 DOC_FILE_TEXT% 88
 DOC_FIND% 122
 DOC_FIND_CONFIRM% 89
 DOC_FIND_DEFAULT% 89
 DOC_FIND_LOW2UPCASE% 89
 DOC_FIND_WHOLEWORD% 89
 DOC_FIRSTPARAGRAPH% 123
 DOC_FMT_BOLD% 90
 DOC_FMT_DEFAULT% 90
 DOC_FMT_ITALIC% 90
 DOC_FMT_STANDARD% 90
 DOC_FMT_STRIKEOUT% 90
 DOC_FMT_UNDERLINE% 90
 DOC_FMT_UNLOCKED% 90
 DOC_FMTLINE 328
 DOC_FONT 329
 DOC_FONTNAME_EVENT 85
 DOC_FONTSIZE_EVENT 85
 DOC_FREESCANCONTEXT% 124
 DOC_GET_INSERT_MODE% 125
 DOC_GETBITMAPCHAR% 126
 DOC_GETBORDER 227
 DOC_GETCONTROLCHARSVISIBILITY% 127
 DOC_GETCURRENTRULEROBJECT
% 128
 DOC_GETCURRENTTAB 228
 DOC_GETCURRENTTABLINE 229
 DOC_GETDEFTAB% 129
 DOC_GETFMTLINE 230
 DOC_GETFONT\$ 130
 DOC_GETFONTAT\$ 131
 DOC_GETFONTBACKCOLOR% 132
 DOC_GETFONTCOLORS 231
 DOC_GETFONTCOLORSAT 232
 DOC_GETFONTFORECOLOR% 133
 DOC_GETFONTINDEXAT% 134
 DOC_GETFONTINDEXCOLORS 233
 DOC_GETFONTINDEXFORMAT%
135
 DOC_GETFONTINDEXNAME\$ 136
 DOC_GETFONTINDEXSIZE% 137
 DOC_GETFONTINDEXUSERFORMAT%
138
 DOC_GETFONTLINECOLOR% 139
 DOC_GETFONTNAMECONTROL 234
 DOC_GETFONTSIZECONTROL 235
 DOC_GETFORMAT% 140
 DOC_GETFORMATAT% 141
 DOC_GETJUSTIF% 142
 DOC_GETLASTMERGEERROR 236
 DOC_GETLINE 238
 DOC_GETLINEINFO 239
 DOC_GETMARGINS 240
 DOC_GETMAXPARAWIDTH% 143
 DOC_GETMODE% 144
 DOC_GETNB_PAGES% 145
 DOC_GETNBSECTIONS% 146
 DOC_GETNEXTPARALINE 241
 DOC_GETPAGE% 147
 DOC_GETPAGEDIM 242
 DOC_GETPARAGRAPHGLOBALUSER
RINFO% 148
 DOC_GETPARAGRAPHINFO 243
 DOC_GETPARAGRAPHKEEPTOGET
HER% 149
 DOC_GETPARAGRAPHKEEPWITHN
EXT% 150
 DOC_GETPARAGRAPHLOCALUSER
INFO% 151
 DOC_GETPARALINEOFFSET% 152

DOC_GETPICTDFNID% 153
 DOC_GETPICTDFNNAME% 154
 DOC_GETPICTSIZE% 155
 DOC_GETPOS 245
 DOC_GETPREVPARALINE 246
 DOC_GETSECTION% 156
 DOC_GETSECTIONJUSTIF% 157
 DOC_GETSELECTION 247
 DOC_GETSELECTIONTEXT\$ 158
 DOC_GETTABLINES 248
 DOC_GETTABS 249
 DOC_GETTEXT 250
 DOC_GETTEXTSIZE% 159
 DOC_GETUSERFORMAT% 160
 DOC_GETUSERFORMATAT% 161
 DOC_GETWIDOWCONTROL% 162
 DOC_GETWINDOW% 163
 DOC_GETXTABS 251
 DOC_HIDE_PICTURES% 101
 DOC_INSERT 252
 DOC_INSERTBITMAPCHAR% 164
 DOC_INSERTCHARS 253
 DOC_ISCLIPBOARD% 165
 DOC_ISFONTLOCKED% 166
 DOC_ISLIKEPRINTER% 167
 DOC_ISMETRICS% 168
 DOC_ISPAGEBREAK% 169
 DOC_ISREADONLY% 170
 DOC_ISREDO% 171
 DOC_ISRULERBAR% 172
 DOC_ISRULERBARSHOWN% 173
 DOC_ISSECTION% 174
 DOC_ISSELECTION% 175
 DOC_ISTAG% 176
 DOC_ISTOOLBOXSHOWN% 177
 DOC_ISUNDO% 178
 DOC_ISWINDOW% 179
 DOC_JUS_CENTERED% 91
 DOC_JUS_FULL% 91
 DOC_JUS_LEFT% 91
 DOC_JUS_LOWER% 91
 DOC_JUS_RIGHT% 91
 DOC_JUS_UPPER% 91
 DOC_LASTPARAGRAPH% 180
 DOC_LAYOUT 180, 254
 DOC_LEADTAB_*% 180
 DOC_LEADTAB_DOT% 92
 DOC_LEADTAB_HYPHEN% 92
 DOC_LEADTAB_NONE% 92
 DOC_LEADTAB_UNDERSCORE% 92
 DOC_LINE_*% 180
 DOC_LINE_ABOVE% 93
 DOC_LINE_BELOW% 93
 DOC_LINE_BETWEEN% 93
 DOC_LINE_LEFT% 93
 DOC_LINE_RIGHT% 93
 DOC_LINE_SHADOW% 93
 DOC_LINEDATA 180, 330
 DOC_LINEKIND_*% 180
 DOC_LINEKIND_DOUBLE% 94
 DOC_LINEKIND_SIMPLE% 94
 DOC_LOCKFONTS 180, 255
 DOC_LOENGLISH% 95
 DOC_LOMETRIC% 95
 DOC_M_CREATEMARKER% 180, 181
 DOC_M_CREATEMARKERWITHHANDLE% 180, 182
 DOC_M_DELETEMARKER_EVENT 85
 DOC_M_DEREGISTERMARKERCLASS 180, 256
 DOC_M_DISPOSEMARKER 180, 257
 DOC_M_DISPOSEMARKERCLASS 180, 258
 DOC_M_FINDNEXTMARKER% 180, 183
 DOC_M_FINDPREVMARKER% 184
 DOC_M_GETCLASS% 185
 DOC_M_GETCLASSNAME\$ 186
 DOC_M_GETMARKERCLASS% 187
 DOC_M_GETMARKERDATA 259
 DOC_M_GETMARKERDATASIZE% 188
 DOC_M_GETMARKERHANDLE% 189
 DOC_M_GETTEXT\$ 190
 DOC_M_INSERTMARKER 260

DOC_M_ISMARKER% 191
DOC_M_KIND_DEFAULT% 83
DOC_M_MARKER 331
DOC_M_NEWMARKERCLASS% 192
DOC_M_PASTEMARKER_EVENT 85
DOC_M_REGISTERMARKERCLASS
262
DOC_M_SETMARKERDATA 263
DOC_M_SETTEXT 264
DOC_MAPUSERMSG 265
DOC_MAX_TABULATIONS% 84
DOC_MEMFILE 333
DOC_MERGE 267
DOC_METRICS 269
DOC_MODE_DEFAULT% 96
DOC_MODE_DRAFT% 96
DOC_MODE_PAGE% 96
DOC_MODIFYBITMAPCHAR% 193
DOC_MODIFYFORMAT 270
DOC_MOVESELECTION 271
DOC_NEW 272
DOC_NEXTPARAGRAPH% 195
DOC_OPEN 273
DOC_PAGEBREAK 275
DOC_PARA_KEEPTOGETHER% 97
DOC_PARA_KEEPWITHNEXT% 97
DOC_PARA_PAGEBREAK% 97
DOC_PARA_SECTION% 97
DOC_PASTE 276
DOC_POS_BOTTOM% 98
DOC_POS_INSIDE% 98
DOC_POS_LEFT% 98
DOC_POS_RIGHT% 98
DOC_POS_TOP% 98
DOC_POS2XYVIEW% 196
DOC_PREVPARAGRAPH% 197
DOC_PRINT% 198
DOC_PRINTFILE% 199
DOC_RCHG_COLORS 99
DOC_RCHG_COLUMN 99
DOC_RCHG_CONTEXT 99
DOC_RCHG_FACENAME 99
DOC_RCHG_FACESIZE 99
DOC_RCHG_FORMAT 99
DOC_RCHG_INDENTS 99
DOC_RCHG_JUSTIF 99
DOC_RCHG_LINE 99
DOC_RCHG_PAGE 99
DOC_RCHG_PARAINFO 99
DOC_RCHG_SECTION 99
DOC_RCHG_TABS 99
DOC_RCHG_USERFORMAT 99
DOC_RCHG_USERPARAINFO 99
DOC_READONLY 277
DOC_REDO 278
DOC_RULEINFO 334
DOC_RULER_EVENT 85
DOC_RULERBAR 279
DOC_SAVE 280
DOC_SCAN_CTXT 336
DOC_SCANFIRST% 200
DOC_SCANNEXT% 202
DOC_SCN_BEGBMP% 100
DOC_SCN_BEGMARK% 100
DOC_SCN_ENDBMP% 100
DOC_SCN_ENDMARK% 100
DOC_SCN_ENDPARA% 100
DOC_SCN_ERR% 100
DOC_SCN_FONT% 100
DOC_SCN_SPACE% 100
DOC_SCN_TAB% 100
DOC_SCN_TAG% 100
DOC_SCN_TEXT% 100
DOC_SECTION 281
DOC_SET_INSERT_MODE 282
DOC_SETBORDER 283
DOC_SETCONTROLCHARSVISIBILI
Y 285
DOC_SETCURRENTTAB 286
DOC_SETCURRENTTABLINE 287
DOC_SETDEFTAB 288
DOC_SETFOCUS_EVENT 85
DOC_SETFONT 289
DOC_SETFONTBACKCOLOR 290
DOC_SETFONTCOLORS 291
DOC_SETFONTFORECOLOR 292

DOC_SETFONTINDEXCOLORS 293
 DOC_SETFONTINDEXFORMAT 294
 DOC_SETFONTINDEXNAME 295
 DOC_SETFONTINDEXSIZE 296
 DOC_SETFONTINDEXUSERFORMAT 297
 DOC_SETFONTLINECOLOR 298
 DOC_SETFORMAT 299
 DOC_SETJUSTIF 300
 DOC_SETLIKEPRINTER 301
 DOC_SETMARGINS 302
 DOC_SETMAXPARAWIDTH 303
 DOC_SETMODE 304
 DOC_SETPAGE 305
 DOC_SETPARAGRAPHGLOBALUSERINFO 306
 DOC_SETPARAGRAPHKEEPTOGETHER 307
 DOC_SETPARAGRAPHKEEPWITHNEXT 308
 DOC_SETPARAGRAPHLOCALUSERINFO 309
 DOC_SETPARAGRAPHSINFO 310
 DOC_SETPOS 311
 DOC_SETSECTION 312
 DOC_SETSECTIONJUSTIF 313
 DOC_SETSELECTION 314
 DOC_SETTABLINES 315
 DOC_SETTABS 316
 DOC_SETTAG 317
 DOC_SETUSERFORMAT 318
 DOC_SETWIDOWCONTROL 319
 DOC_SETWINDOW 320
 DOC_SETXTABS 321
 DOC_SHOW_COMMANDS% 101
 DOC_SHOW_PARAGRAPHS% 101
 DOC_SHOW_SPACES% 101
 DOC_SHOW_TABS% 101
 DOC_SHOW_TAGS% 101
 DOC_SHOWRULERBAR 322
 DOC_SHOWTOOLBOX 323
 DOC_SPLITPARAGRAPH 324
 DOC_TABLINE 338
 DOC_TABLINES 339
 DOC_TABS 340
 DOC_TEXTCOLORS 341
 DOC_TYPTAB_CENTER% 102
 DOC_TYPTAB_DECIMAL% 102
 DOC_TYPTAB_LEFT% 102
 DOC_TYPTAB_RIGHT% 102
 DOC_UNDO 325
 DOC_UNITS2XPEL% 203
 DOC_UNITS2YPEL% 204
 DOC_UPDATE 326
 DOC_USER_MESSAGE_0 103
 DOC_USER_MESSAGE_1 103
 DOC_USER_MESSAGE_10 103
 DOC_USER_MESSAGE_11 103
 DOC_USER_MESSAGE_12 103
 DOC_USER_MESSAGE_13 103
 DOC_USER_MESSAGE_14 103
 DOC_USER_MESSAGE_15 103
 DOC_USER_MESSAGE_2 103
 DOC_USER_MESSAGE_3 103
 DOC_USER_MESSAGE_4 103
 DOC_USER_MESSAGE_5 103
 DOC_USER_MESSAGE_6 103
 DOC_USER_MESSAGE_7 103
 DOC_USER_MESSAGE_8 103
 DOC_USER_MESSAGE_9 103
 DOC_WIN_BODY% 104
 DOC_WIN_FIRSTFOOTER% 104
 DOC_WIN_FIRSTHEADER% 104
 DOC_WIN_FOOTER% 104
 DOC_WIN_HEADER% 104
 DOC_XPEL2UNITS% 205
 DOC_XTABS 342
 DOC_XYVIEW2POS% 206
 DOC_YPEL2UNITS% 207
 E
 Editeur 18
 En 21
 Encadrement 30
 Entiers associés 31, 33, 49
 Envoi d'une lettre type 41
 ESCLOAD% 208
 ESCPRINT% 209

- Evénements 79
- F
- Fenêtre NS 8
- Feuille de styles 37
- Fichier de commandes 41
- Fonctions et Instructions NCL 81
- Format 40
 - ANSI 40
 - ASCII 40
 - NS 40
 - RFT 40
 - RTF 40
 - Texte 40
- Fusion 41, 73
- G
- Glossaires 33
- H
- Heure 21
- I
- Images bitmaps 35
- Introduction 5
- L
- LINE_BORDER 343
- M
- M.D.I. 345
- Marges 45
- Marqueurs 37
- Mode caractère 18, 39
- N
- NS_WP 8
- NS02WP.DLL 11
- NS02WPTB.DLL 11
- NSEDIT 13, 345
- Numéro de page 21, 45
- Numérotation automatique 37, 348
- O
- Ombre 30
- P
- Parcours du document 34
- Pied de page 21
- Points de suite 26, 47
- Polices de caractères 32, 33
- Project Resources Libraries 8
- R
- Recherche de texte 62
- Règle 17
- Remplacement de texte 62
- Retraits 26, 47
- S
- Saut de page 23, 45
- Saut de section 46
- Setup Generator 10
- Solidaires 45
 - lignes 24, 45
 - paragraphes 24, 45
- Styles de caractères 33
- Styles de paragraphes 31, 348
- Système 39
 - anglo 39
 - métrique 39
- T
- TAB_RULER_OBJECT% 105
- Table des matières 348
- Tableaux 28
- TABLINE_RULER_OBJECT% 105
- Tabulations 26, 47
- Tabulations verticales 28, 47
- TE*% 106
- Titres 31
- Traits verticaux 28, 47
- Trame 30
- TXTESCTABLE 344
- V
- Veuve 45