

# BULLETIN TECHNIQUE

## NS-DK Version 12.0.1

### Introduction

Ce document présente la liste des nouvelles fonctionnalités de la 12.0.1, disponible courant juin 2023.

Vous trouverez dans ce bulletin technique :

- Des éléments relatifs à la stratégie produit mise en œuvre par Nat System.
- Une description sommaire des améliorations proposées dans NS-DK 12.
- La liste des configurations et plates-formes supportées par NS-DK 12.
- La liste des anomalies corrigées.

Tous les efforts ont été mis en œuvre pour garantir la validité et la pertinence des informations proposées au sein de ce document. Toutefois, dans un souci constant de qualité et de fiabilité, et en fonction d'impératifs d'ordre technique, marketing ou calendaire, nous serons éventuellement amenés à en affiner le contenu dans les semaines à venir.

Ainsi, ce bulletin ne saurait engager la responsabilité de Nat System et ne constitue pas, de ce fait, un engagement contractuel.

### Définition et objectifs de la version 12

Pour accompagner ses clients et leur fournir des nouveaux contrôles, bibliothèques, ainsi qu'une documentation et des exemples complets, Nat System met à disposition la version 12.

#### *Support de Windows 11 et Windows Server 2022*

Le support des nouvelles versions de Windows et Windows Server est ajouté dans la version 12.

Windows 11, successeur de Windows 10, n'existe qu'en version 64 bits.

Tout comme Windows 10 64 bits, il dispose d'un sous-système 32 bits permettant l'exécution d'application 32 bits.

La version 12 de NS-DK supporte les deux modes d'exécution : 32 et 64 bits.

NS-DK 12 fonctionne également sur Windows 10.

Windows Server 2022 est le successeur de la version 2019. Windows Server est souvent utilisé comme base des serveurs Citrix. Nat System garantit le support de ces logiciels sous Windows Server.

Le bon fonctionnement sous Citrix est de la responsabilité de Citrix qui doit garantir qu'une application fonctionnant sur Windows Server fonctionnera sur Citrix.

NS-DK 12 est la première version NS-DK qui assure le support de Windows 11 et Windows Server 2022.

C'est la version minimale requise pour utiliser ces plateformes.

Attention ! la hauteur par défaut de la barre de tâche Windows augmente entre Windows 10 et Windows 11, passant de 40 pixels à 47 pixels, réduisant de 7 pixels l'espace vertical disponible (dans le cas d'une barre de tâche horizontale).

Si les écrans de vos applications utilisent tout l'espace vertical et ont une taille figée, une partie de l'écran pourrait être

rognée lors du passage en Windows 11.

Nat System est en cours de rédaction d'un guide méthodologique afin de vous accompagner dans la démarche de migration Windows 11 en proposant des solutions pour résoudre ce problème.

## *Suivre les évolutions technologiques*

L'environnement de vos applications évolue, il faut être capable de s'y insérer efficacement.

Avec la version 12, NS-DK ajoute le support :

- Red Hat 8.5 en remplacement de 8.0
- Aix 7.2 en remplacement de Aix 7.1
- Oracle 21c

La liste complète des configurations supportées est décrite à la fin de ce document dans le chapitre « Configurations Supportées ».

La version d'OpenSSL utilisée pour les modules NSHTTP et NSSOAP est passée en version 3.1 pour une meilleure sécurité de vos échanges.

La version du JDK livrée est passé de la 1.8 à la version 17 du JDK. Les outils de génération des clients de service web utilisent cette version en mode compliance JDK 11 qui devient le minimum requis pour les outils.

Pour rappel, le JDK est toujours en version 64 bits indépendamment de la version 32 ou 64 bits de NS-DK.

Suite à la réécriture du runtime en C++ pour la partie Windows en v11, Nat System a continué la modernisation de ces runtimes en portant ce runtime sous Linux/Aix de manière à avoir un runtime unifié entre Windows et Unix/Linux.

Cette opération va faciliter l'ajout de nouvelles fonctionnalités et favoriser la cohérence des comportements.

## *Support du 64 bits*

Le 64 bits est en train de devenir le standard des nouveaux composants de votre système d'information. Pour vous permettre de suivre la vague, Nat System met à disposition depuis la version 11, l'outil NS-DK et les bibliothèques de runtime associés en 64 bits.

Cette nouvelle version permet de générer des applications Windows 64 bits qui pourront s'interfacer avec les nouveaux services disponibles en 64 bits.

Pour vous aider dans la phase de migration, deux évolutions ont été réalisées :

- Support d'un nouvel entier INT(0) qui a la même taille qu'un pointeur. (4 octets en 32 bits, 8 octets en 64 bits)
- Nouveau mot clé #null dans le NCL. Permettant de tester proprement les pointeurs, en évitant les warnings.

## *Faciliter la mise au point de vos applications.*

Nat System profite dans cette version de la réécriture du runtime pour ajouter quelques fonctionnalités pour vous faciliter la mise au point de vos applications.

La plus importante est l'introduction d'un mécanisme de log unifié, adapté à la production qui va vous permettre d'avoir une bonne compréhension des dysfonctionnements de vos applications.

Les autres évolutions sont :

- Sauvegarde de la pile d'un crash dans le fichier de log
- Amélioration de la vitesse d'affichage dans la fenêtre de LOG.
- Mise en avant des warnings C4700 (variables utilisées sans avoir été initialisées)
- Résumé des warnings de compilation après la génération.
- Meilleure utilisation du multi processeur pour la compilation.
- Correction des problèmes d'accents dans le log de la compilation.
- Meilleure robustesse de NsDebug (plus de plantage sur les pointeurs incorrects)
- Amélioration de la génération C,
- Meilleure robustesse des applications grâce aux nouvelles options des compilateurs / générateur modernes.

## Améliorations de l'outil de développement.

Nous n'avons pas oublié le confort et l'efficacité des développeurs avec plusieurs évolutions qui concernent l'outils :

- Nouvelle boîte de modification des propriétés de tous les contrôles de la fenêtre (Couleurs, police, ancrage...)
- Sauvegarde centralisée des signets (bookmarks) et indication visuelle des librairies en contenant.
- Nouvelles polices plus lisibles dans l'outil
- Nouvelles icônes s'inspirant du « flat design »
- Nouvelles versions des Bases de Données.
- Meilleure cohérence des noms des nouveaux contrôles.
- Correction de bugs.
- Affichage plus rapide de la grille.
- Recherche plus facile dans la fenêtre de LOG.

A l'heure actuelle, si on fait un double-click sur un nom de fonction elle est incorporée dans le code NCL, désormais si on fait Ctrl-double-click on rentre dans la fonction.

## Fichier NSLIB.INI

- Possibilité de lire les fichiers texte avec diverses options de CR/LF CRLF
- Initialisation du métrique par défaut dans NS-CALC (LoEnglish/loMetric)

## Nouveautés dans les librairies

- Fonctionnalité de création de QR/CODE DataMatrix
- Nouvelle librairie de gestion de tableau dynamique de pointeur **nsdynCollec.ncl**
- Nouvelle librairie de chiffrage et hachage NSCRYPT.NCL
- NSMISC, fonction FGetSizeEx% renvoyant un int(8) pour lire les tailles de fichier > 2 gigas.
- NSWIN : Nouvelle API nsWriteConsole () pour écrire dans la console depuis une application graphique.

## Information : version 12

Dans la roadmap de ses produits, Nat System annonce la version 12 en G.A (General Availability) de ses ateliers de développement (NatStar, NS-DK et NatXtend), des Runtimes serveurs et Runtime utilisateurs ainsi que des produits associés pour juin 2023.

La version 12 est livrée en standard sur une nouvelle **cible 64 bits** en conformité avec les standards du marché. (\*)

Afin de ne pas pénaliser ses clients par l'acquisition d'une nouvelle cible 64 bits, Nat System a défini un nouveau modèle de licensing à partir de la version 11 de ses logiciels, basé sur le paiement d'une redevance annuelle de droits d'usage : Ateliers de développements, Runtimes utilisateurs (\*\*), Runtimes Serveurs, et Produits associés.

À cet effet, la livraison des versions 12 de nos logiciels fait l'objet d'un **nouveau contrat de licence et de maintenance (\*\*\*)**.

(\*) La version V12 intègre également une cible 32 bits

(\*\*) Pour rappel, l'ensemble des Runtime utilisateurs Nat System depuis la V10 fait l'objet d'une maintenance spécifique.

(\*\*\*) **La maintenance standard** Nat System est disponible sur la version courante V et la version V-1. Le support d'une version antérieure nécessite la souscription d'une extension de maintenance en complément de la maintenance standard.

Pour rappel, depuis la version 10 de ses produits, NAT SYSTEM propose à ses clients la cible de génération Java pour ses clients NS-DK ou NatStar.

## Installation

La version 12 est livrée sous forme de deux images distinctes. (Livraison possible par FTP, clef USB, CD ..) :

- La première image n'installe que le runtime nécessaire aux applications.
- La seconde image installe l'outil NS-DK et tous ses add-ons. L'image des outils de développement n'est plus

autonome. Elle nécessite l'image du runtime.

Pour installer un environnement de développement, il faut commencer par installer le runtime puis par-dessus les outils.

## Installation du runtime NS-DK

Le runtime doit être installé sur tous les postes utilisateurs d'une application NS-DK.

Les deux runtimes sont disponibles (32 et 64 bits). Il est possible de n'installer que la version 32 bits si vous continuez de développer en 32 bits.

Le runtime doit également être installé sur les postes des développeurs NS-DK.

L'installation de l'outil propose :

- Le runtime NS-DK.
- Le runtime MSVC 2022 (dll vcruntime143.dll) : ce runtime Microsoft est nécessaire au bon fonctionnement du runtime NS-DK et cette version ou une version ultérieure compatible doit être présente sur le poste des utilisateurs ou des développeurs. La procédure d'installation propose son installation.

A partir de la version 12, le runtime inclut les fichiers PDB (Program DataBase file) correspondant aux DLL. Ces fichiers qui contiennent des informations sur les types et symboles utilisés lors de la compilation du runtime permettent lors d'un crash applicatif, la production d'une pile avec des informations détaillées correspondant au source C.

Avec les PDB, la taille du runtime est passé d'une vingtaine de Mo à 180 Mo.

Nous vous recommandons de déployer en production les fichiers PDB avec les DLL afin de profiter pleinement du nouveau mécanisme d'affichage de la pile lors d'un crash. Il est toutefois possible de ne déployer en production que les seules DLL sans leur fichier PDB : dans ce cas, la pile risque d'être peu exploitable.

## Installation de l'outil NS-DK

L'outil peut être installé en version 32 bits pour produire des applications 32 bits ou en 64 bits pour produire des applications 64 bits pour Windows.

L'installation de l'outil propose :

- NSDESIGN (Outil de développement).
- Java, JDK (Run Time java n'est utilisé que par le module d'import des services web).
- Samples (exemples de projets NS-DK).
- Pilotes de bases de données : DB2, Oracle, Sybase, ODBC, MS-SQL-Server, Informix, PostgreSQL

## À propos

En cliquant sur le menu Help/About vous pouvez immédiatement savoir quelle version vous utilisez :

- Pour la version officielle. Il s'affiche Version 12.0.1.

## Faciliter la migration vers le 64 bits.

### Nouvel entier INT(0)

Cet entier a la même taille qu'un pointeur. C'est-à-dire qu'il fera 4 octets en 32 bits, et 8 octets en 64 bits.

Cet entier peut être utilisé dans certains cas où une portabilité totale entre les entiers et les pointeurs est requise.

### Nouveau mot clé #null dans le NCL.

Si vous utilisez des pointeurs, il est plus "propre" de les initialiser à la valeur **#null** plutôt qu'avec 0, cela assure mieux la compatibilité, avec le 64 bits et avec la génération java et évite les warnings.

Il est possible de l'utiliser pour tester sa nullité.

## Faciliter la mise au point de vos applications à l'exécution.

### *Nouveau mécanisme de trace dans le fichier log.*

Le runtime NS-DK utilise log4cxx pour loguer les erreurs les plus importantes avec leur pile (uniquement sous Windows pour la pile).

Vous pouvez utiliser dans votre code applicatif la nouvelle librairie nclLogr.NCL (remplaçante de NSLOGR.NCL) pour ajouter vos propres logs.

La fonctionnalité NS\_TRACE est également envoyée par défaut vers ce nouveau logger. Elle devient obsolète. Nous vous recommandons l'utilisation de la librairie nclLogr.NCL qui permet :

- L'utilisation d'un des 6 niveaux de log : TRACE, DEBUG, INFO, WARN, ERROR, FATAL
- L'utilisation de plusieurs instances de logger qui permettent une activation différenciée des niveaux de log visibles et un filtrage efficace.

Ce mécanisme de log est compatible avec la production et fonctionne en mode serveur.

Il gère les accès concurrents, ce qui permet à plusieurs applications ou plusieurs utilisateurs d'alimenter en même temps le même fichier.

Nat System a prévu de renforcer au fil du temps l'utilisation de ce nouveau mécanisme pour loguer les informations les plus pertinentes du fonctionnement du runtime.

La librairie NSLOGR.NCL est remplacée par nclLogr.NCL qui met à disposition un jeu d'instruction plus simple et efficace.

Pour les utilisateurs de l'ancienne librairie NSLOGR.NCL, vous avez deux possibilités :

- Désactiver le nouveau mécanisme en positionnant initLogger à FALSE dans le fichier nslib.ini : votre application log comme auparavant.
- Supprimer votre initialisation NSLOGR.NCL, et laisser le runtime prendre en charge l'initialisation du mécanisme de log (il faudra peut-être renommer le fichier ini du logger et le compléter). Votre ancienne log vient s'ajouter à la nouvelle.

### *Sauvegarde de la pile après un crash dans le fichier log.*

A partir de la V12, Nat System a mis en place un mécanisme d'interception des crash applicatifs qui permet la récupération de la pile des appels au moment du crash.

Cette pile est envoyée dans la log.

Pour que les informations de la pile soient pertinentes, il est recommandé de compiler ses DLL applicatives avec des options qui provoquent la création de fichier PDB et de livrer ces fichiers PDB avec vos DLL applicatives en production.

Nat System suggère l'utilisation des options suivantes :

```
/c /W3 /Gs /Zip /DWIN32 /O2 /MD /diagnostics:column /analyze
```

## Faciliter la mise au point de vos applications dans l'outil NS-DESIGN

### *Amélioration de la vitesse d'affichage dans la fenêtre de LOG.*

Jusqu'alors, si on avait activé l'option permettant l'affichage des détails de la génération dans la fenêtre de log, il y avait une désynchronisation entre la fenêtre ligne de commande du compilateur, et la fenêtre de LOG de NS-DK.

Maintenant, la vitesse dans la fenêtre de LOG a été accélérée et les deux affichages sont synchrones. Ce qui diminue le temps de génération d'un projet.

### *Mise en avant des warnings C4700, (variables utilisées sans avoir été initialisées)*

Ces warnings, générés par le compilateur Microsoft, sont importants et doivent être pris en considération. Maintenant, une petite icône rouge apparaît devant la ligne incriminée du fichier.

```

ICompiling wdialog.c...
CL.EXE /c /W3 /Gs /Zip /DWIN32 /RTC1 /MDd /I D:\PRJ\PLIBSCPP\TRUNK\TEST\TRUC\GEN /FoD:\PRJ\PLIBSCPP\TRUNK\TES
Compilateur d'optimisation Microsoft (R) C/C++ version 19.34.31937 pour x86
Copyright (C) Microsoft Corporation. Tous droits réservés.

```

```
wdialog.c
```

```

% Compilation: D:\PRJ\PLIBSCPP\TRUNK\TEST\TRUC\GEN\wdialog.c(13719) : warning C4700: variable locale 'l' non initialisée utilisée

```

Si on double clique sur cette ligne, on ouvre le fichier C généré. Il est alors facile de retrouver le NCL correspondant.

### Résumé des warnings de compilation après la génération.

Il peut être intéressant d'avoir d'un coup d'œil un résumé des warning générés par le compilateur C.

```

----- Warning resume -----
Warning C4018 : '>' : incompatibilité signed/unsigned (2)
Warning C4022 : % : incompatibilité de pointeur pour le paramètre réel 7 (1)
Warning C4047 : 'fonction' : les niveaux d'indirection de 'NS_PTR *' et de 'NS_LONG *' sont différents (1)
Warning C4101 : % : variable locale non référencée (35)
Warning C4244 : '=' : conversion de 'NS_LONG' en 'NS_INT', perte possible de données (5)
% Warning C4700 : variable locale % non initialisée utilisée (1)

```

Voici un exemple de résumé d'un projet comportant de très nombreux warnings. (Dont un warning C4700 et 37 variables non référencées)

### Meilleure utilisation du multi processeur pour la compilation.

Une nouvelle option a été ajoutée au générateur. Elle permet, quand on utilise un compilateur récent (Visual Studio 2019 et supérieur) de lancer la compilation de plusieurs fichiers C en même temps.



La valeur 0 permet de compiler tous les fichiers C générés en même temps. Ce qui réduit le temps de compilation.

### Meilleure robustesse de NsDebug (plus de plantage sur les pointeurs incorrects)

La nouvelle architecture en C++ de nos produits permet une bien meilleure robustesse.

Dans cette optique il y avait un bug dans NS-DEBUG qui a pu être corrigé.

Si on faisait un watch d'une variable de type pointeur, et que ce pointeur n'était pas encore affecté ou utilisé, on pouvait avoir un plantage, car à l'évaluation, le pointeur référençait un emplacement mémoire interdit de lecture.

Maintenant, seul un message apparait indiquant que le pointeur n'est plus valide :

```
seg_personne(pt).nom$: <Access Violation Exception >
```

### Amélioration de la génération C.

Cette amélioration concerne principalement la gestion des sous/caractères uniques dans une chaîne de caractères.

Lors de l'utilisation de la syntaxe NCL qui permet d'isoler un caractère

```
myString$(5) = '*'
```

ou

```
ch = myString$(5)
```

### Nouveau mot clé #location du NCL pour les logs applicatives

Si, dans une librairie NCL ou depuis le code d'un événement, vous voulez mettre un log avec le nom de la fonction/instruction ou le nom du contrôle et de l'événement, le mot clef #location le fait automatiquement.

Exemple

```

instruction testLogFromLib
local u%
    u% = 10
    NSLogDebug "fr.natsystem.natware.purchases", #location && ": u=" && u%

```

```
endInstruction ; testLogFromLib
```

Affichera dans la log :

**DEBUG fr.natsystem.natware.purchases I.MYLIB.TESTLOGFROMLIB(145) : u= 10**

Et depuis l'événement EXECUTED du bouton PB\_TEST\_TRACE de la fenêtre WDIALOG

```
; EVENT executed
NSLogInfo "fr.natsystem.natware.purchases", #location
```

Affichera dans la log :

**INFO fr.natsystem.natware.purchases W.WDIALOG.PB\_TEST\_TRACE.EXECUTED(3)**

Les N° entre parenthèses correspondent aux N° de ligne dans la librairie et dans l'événement.

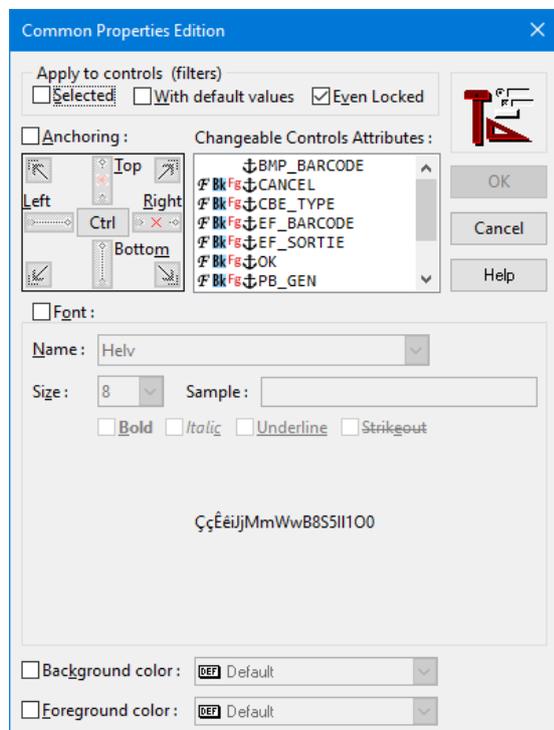
## Améliorations de l'outil de développement.

### *Nouvelle boîte de modification des propriétés de tous les contrôles de la fenêtre.*

Quand on clique sur la nouvelle icône :



Une nouvelle boîte de dialogue apparaît.



Cette boîte permet de modifier : pour tous le contrôles ou ceux sélectionnés (même ceux dans les group-box)

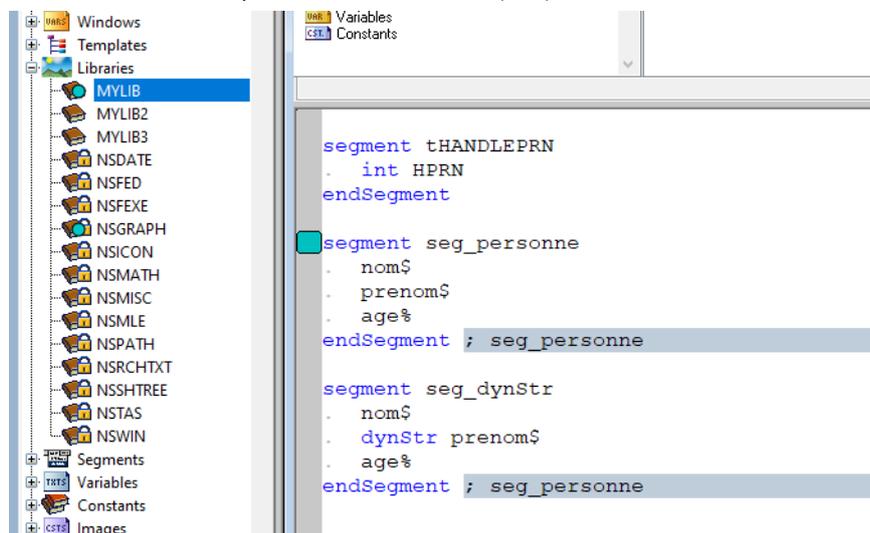
- La police de caractère,
- La couleur d'avant-plan
- La couleur d'arrière-plan
- L'ancrage.

On peut sélectionner les contrôles que l'on désire

### *Sauvegarde centralisée des signets (bookmarks) et indication visuelle des librairies en contenant.*

Avec NS-DESIGN, il est possible de placer des bookMarks (ou onglets) dans le code NCL. (Ctrl-F12 dans l'éditeur, ou menu contextuel)

Ces bookmarks sont maintenant sauves et une indication visuelle dans la liste de librairies permet de retrouver facilement les librairies NCL comportant des bookmark. (F12)



Sur cet exemple, on peut voir qu'il y a un signet sur la définition du segment SEG\_PERSONNE et que deux librairies NCL (MYLIB et NSGRAPH) ont des bookmarks ou signets.

Cette évolution permet de retrouver facilement l'endroit où l'on travaillait précédemment.

### *Nouvelles polices plus lisibles dans l'outil*

Dans l'outil NS-DESIGN de nombreuses fenêtres ont été remaniées afin de les rendre le texte plus lisible.

### *Nouvelles icônes s'inspirant du « flat design »*

Les icônes de NS-DK ont été changées pour correspondre davantage à la tendance « Flat Design » actuelle.

Meilleure cohérence des noms des nouveaux contrôles.

Lors de la création d'un nouveau contrôle dans une fenêtre, ou lors de sa copie, les noms générés sont maintenant normalisés.

(PB\_\*, EF\_\*, ST\_\* ...=

### *Affichage plus rapide de la grille.*

Si vous activez l'option "Display Grid" depuis NS-DESIGN, son affichage a été considérablement accéléré.

### *Les boîtes à onglets ne s'affichent plus en mode "Escalier"*

Cet ancien mode d'affichage a été abandonné pour la version 12.

### *Amélioration de la recherche dans la fenêtre de LOG.*

Quand on fait 'Ctrl-F' depuis fenêtre de log, une barre de recherche apparaît. Le champ de recherche est initialisé avec "Warning" par défaut. On peut lancer la recherche avec 'Entrée' ou F3. Ctrl-F3 permet une recherche vers le haut.

Ces raccourcis clavier sont actifs aussi si le focus est dans la fenêtre de log.

## Éditeur de NCL

La barre coulissante de la scroll-bar horizontale a été agrandie.

## Fonctionnalité de création de QR/CODE DataMatrix

Le composant Zint qui sert à créer tous types de barCode et QR Code, est livré et peut être appelé simplement. Voir l'exemple fourni.

## Nouveautés dans les bibliothèques

- Bibliothèque NSMISC.NCL
  - Fonction FGetSizeEx% renvoyant un INT(8) pour lire les tailles de fichier > 2 gigas.
  - Fonction getTmpDir\$, permettant de spécifier un répertoire temporaire spécifique à l'application courant.
  - Fonction cleanTmpDir\$, pour effacer le contenu de ce répertoire temporaire.
  - Fonctions concatPath1\$, concatPath2\$, concatPath3\$, concatPath4\$ pour concaténer simplement des répertoires et des noms de fichiers, sans se soucier des séparateurs intermédiaires.
- Bibliothèque NSMAP : Nouvelle API nsMaps\_Map\_Clear() pour vider une map sans la détruire.
- Bibliothèque NSWIN : Nouvelle API nsWriteLnConsole () pour écrire dans la console depuis une application graphique.
- Bibliothèque nsUnicode : Ajout d'une nouvelle fonction UTF8toAnsiEx\$ permettant de Convertir une chaîne UTF8 avec des caractères décomposés en une chaîne ANSI (précomposée)

## Nouvelle Bibliothèque nclLOGR

Cette bibliothèque permet de gérer la génération de fichier LOG, en utilisant toute la puissance de LOG4J/LOG4CXX Vous pouvez utiliser dans votre code applicatif la nouvelle bibliothèque nclLogr.NCL (remplaçante de NSLOGR.NCL) pour ajouter vos propres logs.

## Nouvelle bibliothèque NSDYNCOLLEC

La nouvelle bibliothèque NSDYNCOLLEC permet de gérer des collections de pointeurs. Cette collection peut croître dynamiquement, et possède de nombreuses fonctionnalités. (Tri, libération automatique...)

## Nouvelle bibliothèque NSCrypt

La nouvelle bibliothèque NSCRYPT permet le chiffrement de mot de passe et le hachage de fichiers.

## Nouveaux exemples disponibles

- NSDYNCOLLEC : Utilisation de la bibliothèque de gestion de tableaux dynamiques.
- NSBARCODE : démo de l'appel du composant Zint qui sert à créer tous types de barCode.
- NSCRYPT : Utilisation de la bibliothèque des fonctions de Hachage et de dérivation de clé.

## Exemples enrichis

- NSMAP : Utilisation de la nouvelle API : NSMAPS\_MAP\_CLEAR.

## Certifications

NS-DK 12.0 est certifié avec les derniers systèmes d'exploitation les plus performants :

- Windows Server 2019 et 2022
- Windows 10, 32 et 64 bits
- Windows 11 64 BITS

## Bulletins techniques

**NB:** Les bulletins techniques de NS-DK 8, NS-DK 10, et NS-DK 11, décrivant toutes les améliorations et extensions développées pour NS-DK. 8, 10 et 11 intégrées dans NS-DK 12.0 sont disponibles.

## Autres produits de Nat System

Par ailleurs Nat System propose aussi les offres et produits suivants :

- NatStar : Atelier permettant de concevoir efficacement des projets d'envergure et complexes en simplifiant les méthodes de développement pour une concentration sur le fonctionnel.
- NSA-Config: gestionnaire de sources de configuration et de cycle de vie pour NatStar.
- NatJet : Outil de développement Java/Ajax Plug-in Eclipse.
- NatXtend : Outil de transformation des applications NS-DK en applications Web JEE.

## Configurations supportées

Nous rappelons que Nat System garantit le support et la compatibilité des seuls environnements cités ci-dessous.

### *NS-DK 12. : environnement de développement 32 ou 64 bits*

**Les systèmes d'exploitation suivants sont supportés :**

- Windows 10, 32 et 64 bits.
- Windows 11.

**Les compilateurs supportés sont :**

- Microsoft Visual Studio 2022.

### *NS-DK 12.0 : environnement d'exécution (production)*

Dans cette partie, nous précisons les environnements d'exécution des applications produites avec NS-DK qui sont supportées.

### *Environnement graphique Windows : runtime 32 bits*

**Windows 10 et 11, Windows server 2019 et 2022**

Les pilotes de SGBD suivants sont supportés :

- DB2 11
- Oracle version 19c, 21c (XA et non XA)
- SQL Server 2019\*, 2022\*
- ODBC 3.51
- SAP ASE (Sybase) 16
- PostgreSQL 15

\* : avec le driver ODBC

Le support des SGBD suivants est possible sur demande :

- Maria DB 10.3
- MySQL 5.7 et 8.0
- PostgreSQL 13 et 14

### *Environnement graphique Windows : runtime 64 bits*

**Windows 10 et 11, Windows server 2019 et 2022**

Les pilotes de SGBD suivants sont supportés :

- DB2 11
- Oracle version 19c, 21c (XA et non XA)
- SQL Server 2019\*, 2022\*
- ODBC 3.51
- SAP ASE (Sybase) 16
- PostgreSQL 15

\* : avec le driver ODBC

Le support des SGBD suivants est possible sur demande :

- Maria DB 10.3

- MySQL 8.0
- PostgreSQL 13 et 14

## Environnement Serveur : runtime 32 bits et runtime 64 bits

### AIX 7.2

Les pilotes de SGBD suivants sont supportés :

- Oracle version 19c et 21c (XA et non XA)
- SAP ASE (Sybase) 16
- PostgreSQL 15

Le support des SGBD suivants est possible sur demande :

- Maria DB 10.3
- MySQL 8.0
- PostgreSQL 13 et 14

Le compilateur supporté est :

- C++ compiler : IBM Open XL C/C++ 17.1 et +

### Linux Red Hat Enterprise 8.5

Les pilotes de SGBD suivants sont supportés :

- Oracle version 19c et 21c (XA et non XA)
- SAP ASE (Sybase) 16
- PostgreSQL 15

Le support des SGBD suivants est possible sur demande :

- Maria DB 10.3
- MySQL 8.0
- PostgreSQL 13 et 14

Le compilateur supporté est :

- GCC (GNU Compiler Collection) 8.5

## Support des produits tiers par leurs éditeurs respectifs

Nat System ne supporte plus les produits tiers au-delà de la fin de leur support standard par leurs éditeurs respectifs.

A titre informatif voici quelques dates de fin de support des logiciels les plus importants.

### Fin de vie des produits tiers :

- Oracle 11.2 fin de support standard en janvier 2015
- Oracle 12c fin de support standard en mars 2022
- Oracle 18 fin de support standard en juin 2021
- Fin de la phase principale de support pour Windows 2008 R2 serveur depuis 13/01/2015
- Fin de la phase principale de support pour Windows 2016 serveur depuis 11/01/2022
- Fin de support de SQL Server 2012 11 juillet 2017
- Fin de support de SQL Server 2014 9 juillet 2019
- Fin de support de SQL Server 2016 13 juillet 2021
- Fin de support de SQL Server 2017 11 octobre 2022
- Fin de support d'Aix 6.1 avril 2017.
- SAP ASE (Sybase) 15.x n'est plus supporté par SAP ASE

### Ne sont pas supportés par NS-DK 12.00 :

- Windows XP, Vista, 7, 8
- IBM AIX 5.\*, 6.\*, 7.1
- Sun Solaris
- HP UX
- SAP ASE (Sybase) 15
- SQL server 2012 à 2017
- Informix 9 et 11
- Oracle 11.\*, 12c et 18.c
- DB2 9, 10
- NS-DB et NSDBR
- Tuxedo 6.5, 11

## L'état technique d'une version

Chaque état technique de version passe successivement du statut " courant " au statut " transition " puis au statut " retiré ".

Un état technique débute à la mise en disponibilité générale d'une version majeure, mineure ou de maintenance et provoque :

L'affectation du statut " courante " à cette version du produit

L'affectation du statut " transition " à la précédente version " courante " du produit

L'affectation du statut " retirée " à la précédente version en " transition " du produit

Le tableau ci-dessous résume les prestations respectivement assurées pour les versions " courantes ", " en transition " et " retirées " d'un produit.

**Vente** : Etat technique disponible à la vente.

**Evolu.** : La couverture fonctionnelle du produit évolue avec les prochaines versions.

**Bugs** : L'analyse et la correction des bugs sont programmées dans les versions courantes ou futures en fonction des priorités.

**Montées de versions** : Le produit est porté sur les nouvelles versions des systèmes d'exploitation.

**Usage** : L'assistance (support téléphonique) est assurée par le centre de Hotline.

	Vente	Evolu.	Bugs P=0	BugsP=1,2,3	Montées versions	Usage
Courante	Oui	Oui (1)	Oui	Oui	Oui	Oui
Transit	Non	Non	Oui	Non	Non (2)	Oui
Retirée	Non	Non	Non	Non	Non	Non

(1) La version courante évolue avec le prochain état technique du produit.

(2) Dans certaines conditions, les montées de versions peuvent être symétriquement compatibles avec le précédent état technique en transition.

Nat System supporte une version de ses produits sur une durée minimum de trois ans. Une version est supportée sur une durée minimum de 1 an comme version courante puis sur une durée minimum de 1 an comme version de transition du produit.

La version 10.x sera retirée courant 2023

La version 11.x sera maintenue en transition jusqu'à la sortie de la version 13.0 (dénomination non contractuelle).

La version 12.0 remplace la version 11.x.



**NS-DK est une marque déposée par Nat System.  
Toutes les autres marques citées sont déposées par leurs auteurs.**