



Manuel d'utilisation Intégration de Services Web

Table des matières

Client Services Web	9
Le plugin WebServices	9
Services Web	10
Définitions	11
WSDL ou Web Services Description Language	11
SOAP	12
XML	13
Limitations	14
Installation	15
Exemple d'utilisation	16
Types, variables, fonctions et instructions NCL à utiliser avec les services Web	24
Champ ISNULL (Librairie NSVTYP)	25
Fonction <Prefix>_<OperationName> (Librairie NSVTYP)	26
Fonction <Prefix>_GET_ERRORMSG\$ (Librairie NSVTYP)	28
FONCTION VARARRAY_GET_ELEMENTPTR (Librairie NSVTYP)	29
Fonction VARARRAY_GET_NBDIMS (Librairie NSVTYP)	31
Fonction VARARRAY_GET_SIZES_PTR (Librairie NSVTYP)	32
FONCTION VARARRAY_ISNULL (Librairie NSVTYP)	33
Fonction VARBINARY_GET_PTR (Librairie NSVTYP)	34
Fonction VARBINARY_GET_SIZE (Librairie NSVTYP)	35
Fonction VARBINARY_ISNULL (Librairie NSVTYP)	36
Fonction VARSTRING_GET_PSTR (Librairie NSVTYP)	37
Fonction VARSTRING_GETLENGTH (Librairie NSVTYP)	38
Fonction VARSTRING_ISNULL (Librairie NSVTYP)	39
Instruction <Prefix>_INITIALIZE (Librairie NSVTYP)	40
Instruction <Prefix>_SET_ENDPOINT (Librairie NSVTYP)	41
Instruction <Prefix>_SET_PASSWORD (Librairie NSVTYP)	42
Instruction <Prefix>_SET_USERNAME (Librairie NSVTYP)	43
Instruction <Prefix>_TERMINATE (Librairie NSVTYP)	44
Instruction VARARRAY_GET_ARRAY (Librairie NSVTYP)	45
Instruction VARARRAY_GET_SIZES (Librairie NSVTYP)	46
Instruction VARARRAY_NEW (Librairie NSVTYP)	47

Instruction VARARRAY_NEW_1D (Librairie NSVTYP)	49
Instruction VARARRAY_NEW_2D (Librairie NSVTYP)	51
Instruction VARARRAY_SETNULL (Librairie NSVTYP).....	53
Instruction VARBINARY_GET (Librairie NSVTYP)	54
Instruction VARBINARY_NEW (Librairie NSVTYP).....	55
Instruction VARBINARY_NEW_SIZED (Librairie NSVTYP)	56
Instruction VARBINARY_SETNULL (Librairie NSVTYP)	57
Instruction VARPOOL_NEW (Librairie NSVTYP).....	58
Instruction VARSTRING_ADD_PSTR (Librairie NSVTYP).....	59
Instruction VARSTRING_ADD_VARSTRING (Librairie NSVTYP)	60
Instruction VARSTRING_GET (Librairie NSVTYP)	61
Instruction VARSTRING_NEW (Librairie NSVTYP)	62
Instruction VARSTRING_SETNULL (Librairie NSVTYP)	63
Segments VTYP_* (Librairie NSVTYP)	64
Types NULLABLE_* (Librairie NSVTYP).....	65
Fonctions et instructions de la librairie NSSOAP	67
SOAPHTTP	68
Fonction NS_SOAPHTTP_GET_HTTPCONNECTION (Librairie NSSOAP)	68
Instruction NS_SOAPHTTP_GET_RESPONSEHEADER (Librairie NSSOAP).....	69
Instruction NS_SOAPHTTP_SET_PASSWORD (Librairie NSSOAP)	70
Instruction NS_SOAPHTTP_SET_USERNAME (Librairie NSSOAP).....	71
Fonction NS_SOAPHTTP_NEW (Librairie NSSOAP)	72
Instruction NS_SOAPHTTP_DISPOSE (Librairie NSSOAP)	73
Instruction NS_SOAPHTTP_SET_PROXY (Librairie NSSOAP).....	74
Instruction NS_SOAPHTTP_SET_TIMEOUT (Librairie NSSOAP).....	75
Instruction NS_SOAPHTTP_SET_CONTENTTYPE (Librairie NSSOAP)	76
Instruction NS_SOAPHTTP_GET_RESPONSESTRING (Librairie NSSOAP).....	77
Fonction NS_SOAPHTTP_SENDREQUEST (Librairie NSSOAP)	79
SOAPCONFIG.....	81
Fonction NS_SOAPCONFIG_GET_SOAPVERSION (Librairie NSSOAP)	81
Instruction NS_SOAPCONFIG_SET_ENVELOPE_NS (Librairie NSSOAP).....	82
Instruction NS_SOAPCONFIG_SET_ENVELOPE_PREFIX (Librairie NSSOAP).....	83
Instruction NS_SOAPCONFIG_SET_ENCODING_NS (Librairie NSSOAP)	84
Instruction NS_SOAPCONFIG_SET_ENCODING_PREFIX (Librairie NSSOAP).....	85

Instruction NS_SOAPCONFIG_SET_SCHEMA_INST_NS (Librairie NSSOAP)	86
Instruction NS_SOAPCONFIG_SET_SCHEMA_PREFIX (Librairie NSSOAP)	87
Instruction NS_SOAPCONFIG_SET_SCHEMA_NS (Librairie NSSOAP)	88
Instruction NS_SOAPCONFIG_SET_SCHEMA_INST_PREFIX (Librairie NSSOAP)	89
SOAPMSGREQUEST	90
Fonction NS_SOAPMSGREQUEST_NEW (Librairie NSSOAP)	90
Instruction NS_SOAPMSGREQUEST_DISPOSE (Librairie NSSOAP)	91
Instruction NS_SOAPMSGREQUEST_SET_SOAPVERSION (Librairie NSSOAP)	92
Fonction NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (Librairie NSSOAP)	93
Fonction NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM (Librairie NSSOAP)	95
Fonction NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM (Librairie NSSOAP)	97
Fonction NS_SOAPMSGREQUEST_CREATE_NULLPARAM (Librairie NSSOAP)	99
Instruction NS_SOAPMSGREQUEST_ADDPARAM (Librairie NSSOAP)	100
Instruction NS_SOAPMSGREQUEST_ADD_HEADERBLOCK (Librairie NSSOAP)	101
Fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG (Librairie NSSOAP)	102
Instruction NS_SOAPMSGREQUEST_SERIALIZE (Librairie NSSOAP)	103
Instruction NS_SOAPMSGREQUEST_SET_MESSAGE_STYLE (Librairie NSSOAP)	105
SOAPMSGRESPONSE	106
Fonction NS_SOAPMSGRESPONSE_GET_PARAMETER (Librairie NSSOAP)	106
Fonction NS_SOAPMSGRESPONSE_GET_PARAMETER_NAMED (Librairie NSSOAP)	107
Fonction NS_SOAPMSGRESPONSE_GET_HEADERBLOCK (Librairie NSSOAP)	108
Fonction NS_SOAPMSGRESPONSE_GET_HEADERBLOCK_NAMED (Librairie NSSOAP)	109
SOAPPARAM	110
Instruction NS_SOAPPARAM_SET_TYPE (Librairie NSSOAP)	110
Fonction NS_SOAPPARAM_GET_PARAMETER_TYPE (Librairie NSSOAP)	112
Instruction NS_SOAPPARAM_ADD_ATTRIBUTE (Librairie NSSOAP)	113
Instruction NS_SOAPSIMPLEPARAM_SET_BASE64 (Librairie NSSOAP)	114
Instruction NS_SOAPSIMPLEPARAM_SET_BOOLEAN (Librairie NSSOAP)	116
Instruction NS_SOAPSIMPLEPARAM_SET_DOUBLE (Librairie NSSOAP)	117
Instruction NS_SOAPSIMPLEPARAM_SET_FLOAT (Librairie NSSOAP)	119
Instruction NS_SOAPSIMPLEPARAM_SET_HEX_BINARY (Librairie NSSOAP)	121
Instruction NS_SOAPSIMPLEPARAM_SET_INT (Librairie NSSOAP)	123
Instruction NS_SOAPSIMPLEPARAM_SET_STRING (Librairie NSSOAP)	125
Instruction NS_SOAPSIMPLEPARAM_SET_USE_EXPLICIT_TYPE (Librairie NSSOAP)	126

Fonction NS_SOAPSTRUCTPARAM_GET (Librairie NSSOAP)	127
Fonction NS_SOAPSIMPLEPARAM_GET_BOOLEAN (Librairie NSSOAP)	128
Fonction NS_SOAPSIMPLEPARAM_GET_DOUBLE (Librairie NSSOAP)	129
Fonction NS_SOAPSIMPLEPARAM_GET_FLOAT (Librairie NSSOAP)	130
Fonction NS_SOAPSIMPLEPARAM_GET_INT (Librairie NSSOAP)	131
Instruction NS_SOAPSIMPLEPARAM_GET_VALUE (Librairie NSSOAP)	132
Fonction NS_SOAPSIMPLEPARAM_GET_STRLGTH (Librairie NSSOAP)	134
Gestion des erreurs	136
Fonction NS_SOAP_ERROR (Librairie NSSOAP)	136
Fonction NS_SOAP_GET_DETAIL_ERROR (Librairie NSSOAP)	137
Constantes NS_SOAP_ERROR_ *% (Librairie NSSOAP)	138
Fonction NS_SOAP_GET_ERRORMSG (Librairie NSSOAP)	139
Utilisation des segments générés	141
Création	142
Holders	144
Utilisation des DynStr	145
Fixer une DynStr à NULL	145
Tester si une DynStr est à NULL	145
Utilisation des tableaux générés	147
Création d'un tableau	148
Exemple d'un tableau de DynStr	148
Exemple d'un tableau de dimension > 1	149
Parcours d'un tableau	150
Tableau de segments	151
Données binaires	153
Callback	155
Index	157

Ce chapitre présente le plugin WebServices intégré aux outils de développement Nat System permettant d'appeler des services Web.

CLIENT SERVICES WEB

Le plugin WebServices

Le plugin (ou composant enfichable) Webservices est un wizard qui interprète les documents WSDL et génère des librairies de fonctions permettant d'appeler les services web exposés dans ces documents WSDL.

L'appel des services Web se fait par le biais du protocole SOAP HTTP.

Le plugin Webservices est intégré avec trois bibliothèques NCL NSSOAP.NCL, NSVTYP.NCL, NSMISC.NCL et NSDYNSTR.NCL qu'il faudra installer pour assurer le bon fonctionnement.

Ce plugin ne permet pas de développer les services Web eux mêmes mais de les appeler.

Pour pouvoir développer des Services Web, Nat System propose un plug-in payant.

Services Web

Le terme Services Web décrit un ensemble d'applications modulaires autonomes et auto-descriptives qui peuvent être publiées, retrouvées et invoquées sur un réseau (l'Internet, un Intranet, ou extranet). Ces Services Web sont décrits par une interface standard, typiquement WSDL.

Vu par certains comme la prochaine génération, ou même le remplacement de l'EAI (Enterprise Application Integration), les services Web permettent les interactions entre différentes applications dans un environnement à configuration dispersée. Ces applications peuvent ainsi découvrir dynamiquement des services et s'y connecter sans qu'un accord préalable n'ait été établi entre elles.

Dans un contexte objet un service Web peut être une classe ou objet comportant des méthodes appelées opérations.

Définitions

WSDL ou Web Services Description Language

Le langage de description de services Web -Web Services Description Language ou WSDL- est un langage basé sur le XML employé pour définir des services Web et pour décrire comment y accéder

Un document WSDL est un document XML simple. Il contient l'ensemble des définitions pour définir un service Web.

SOAP

SOAP est un protocole léger pour l'échange d'information dans un environnement décentralisé et distribué. C'est un protocole basé sur XML.

SOAP peut potentiellement être employé en combinaison avec une variété d'autres protocoles; cependant, Nat System se limite au protocole HTTP.

SOAP a déjà 2 versions : la 1.1 couramment utilisée et la nouvelle 1.2 qui n'est pas encore supporté par ce module.

SOAP est un élément clé de l'architecture .NET de Microsoft pour les applications Internet.

XML

XML (eXtensible Markup Language) est un langage de balise qui a été élaboré pour décrire le contenu de documents et leurs données. Les données sont structurées en branches et nœuds de l'arbre. L'arbre à une racine et une seule. SOAP utilise ce langage pour le codage du message.

Le XML est sensible à la case. Pour plus d'information consulter la documentation XML.

Limitations

Le module Web Services respecte toutes les spécifications SOAP 1.1, avec cependant les limites suivantes :

- ne supporte que le protocole HTTP comme protocole de transport,
- ne supporte pas les messages SOAP avec pièce jointe,
- ne supporte pas l'Unicode.

Installation

Installez les fichiers NSSOAP.NCL, NSVTYP.NCL et NSDYNSTR.NCL en tant que service en exécutant les opérations suivantes :

1. Activez la commande Options \ Services. La boîte de dialogue Modify Services s'ouvre.
2. Sélectionnez NSSOAP, NSVTYP et NSDYNSTR dans la colonne Available et activez la flèche vers la droite. NSSOAP, NSVTYP et NSDYNSTR apparaissent dans la colonne Installed.
3. Cliquez sur le bouton Close.
4. Activez le menu Build/Set Configuration puis le bouton Gen ..., la fenêtre Generator apparaît.
5. Ajoutez dans les groupes (.DLL) et (.EXE), au champ Libraries : NSSOAP.LIB, NSVTYP.LIB.
6. Si vous utilisez un ancien espace de travail avec NatStar 5.00, il est nécessaire d'intégrer manuellement la commande correspondant aux services web. Activer Options / Products, dans le groupe Menus and Icons Configuration, saisir ADE_WEBS dans le champ DLL, IMPORTE dans le champ Name. Cliquez sur Insert, puis sur OK.

Exemple d'utilisation

Actuellement, presque la totalité des services web sont décrits par un document WSDL. Ces documents WSDL sont générés automatiquement par les outils de fabrication de services Web

Dans NatStar, il suffit de renseigner l'URL ou le chemin du WSDL pour que les fonctions nécessaires à l'appel de ces services soient générées dans une librairie que vous pouvez renommer ou supprimer mais dont le contenu, pour des raisons de cohérence et de sécurité est non modifiable.

Soit le fichier du service web de traduction BabelFish défini par le WSDL suivant :

```
http://www.xmethods.net/sd/2001/BabelFishService.wsdl
definitions xmlns:tns="http://www.xmethods.net/sd/BabelFishService.wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.xmethods.net/sd/BabelFishService.wsdl"
name="BabelFishService">
  <message name="BabelFishRequest">
    <part name="translationmode" type="xsd:string"/>
    <part name="sourcedata" type="xsd:string"/>
  </message>
  <message name="BabelFishResponse">
    <part name="return" type="xsd:string"/>
  </message>
  <portType name="BabelFishPortType">
    <operation name="BabelFish">
      <input message="tns:BabelFishRequest"/>
      <output message="tns:BabelFishResponse"/>
    </operation>
  </portType>
  <binding name="BabelFishBinding" type="tns:BabelFishPortType">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="BabelFish">
      <soap:operation soapAction="urn:xmethodsBabelFish#BabelFish"/>
      <input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:xmethodsBabelFish"/>
      </input>
      <output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:xmethodsBabelFish"/>
      </output>
    </operation>
  </binding>
  <service name="BabelFishService">
    <documentation>Translates text of up to 5k in length, between a variety
of languages.</documentation>
    <port name="BabelFishPort" binding="tns:BabelFishBinding">
      <soap:address
location="http://services.xmethods.net:80/perl/soaplite.cgi"/>
    </port>
```



```
</service>  
</definitions>
```

Ce fichier définit une opération ou méthode appelée BabelFish

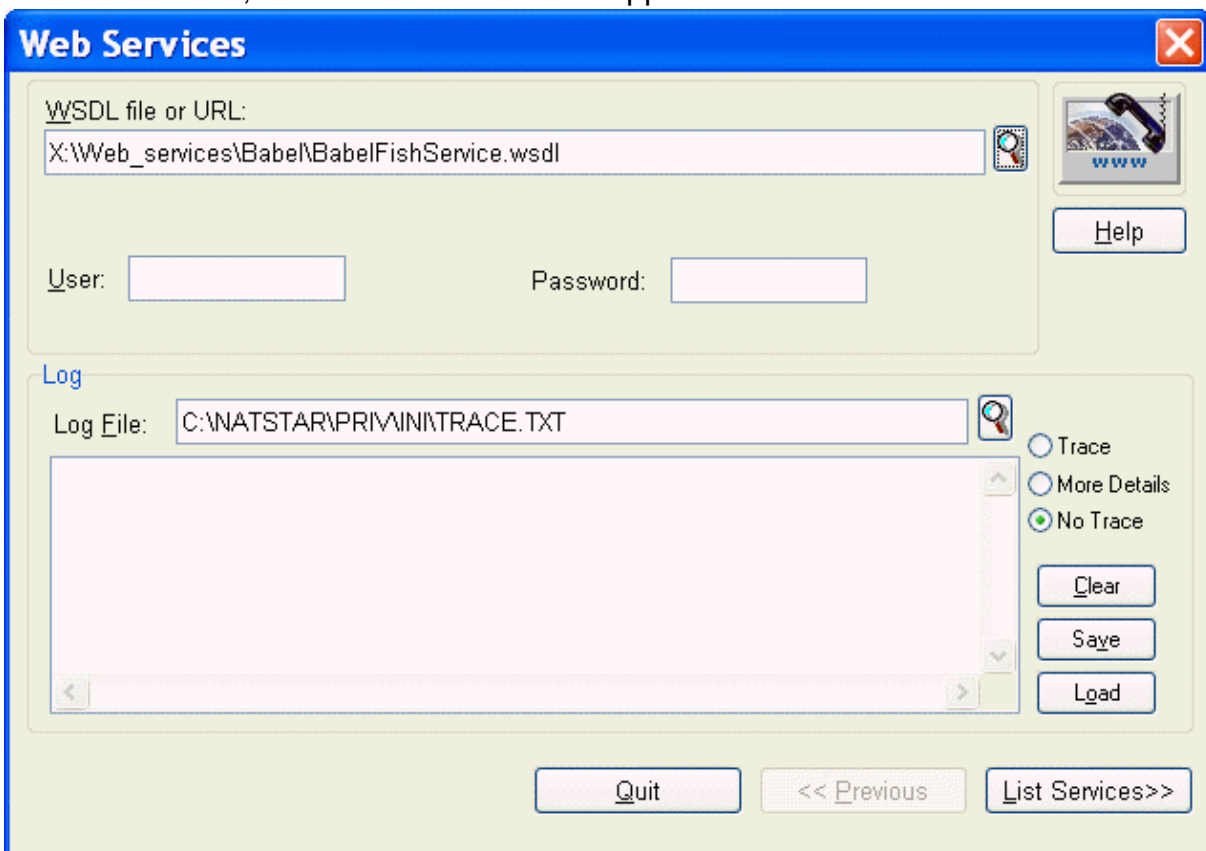
```
<operation name="BabelFish">
```

qui a comme paramètres deux chaînes translationmode et sourcedata et retourne une chaîne return.

Cette opération peut être appelée à l'adresse EndPoint.

Tous ces éléments ont été mis en gras dans le WSDL ci-dessus :

1. On commence par importer le document WSDL en activant le menu File \ Import Web Services, la fenêtre Web Services apparaît :



The image shows a 'Web Services' dialog box with a blue title bar. It contains several input fields and buttons. The 'WSDL file or URL:' field is filled with 'X:\Web_services\Babel\BabelFishService.wsdl'. Below it are 'User:' and 'Password:' fields. The 'Log' section has a 'Log File:' field with 'C:\NATSTAR\PRIV\NIN\TRACE.TXT'. To the right of the log file field are three radio buttons: 'Trace', 'More Details', and 'No Trace' (which is selected). Below the radio buttons are 'Clear', 'Save', and 'Load' buttons. At the bottom of the dialog are 'Quit', '<< Previous', and 'List Services>>' buttons. There is also a 'Help' button with a globe icon in the top right corner.

on saisit l'URL ou le chemin du WSDL au niveau du champ WSDL file or URL.

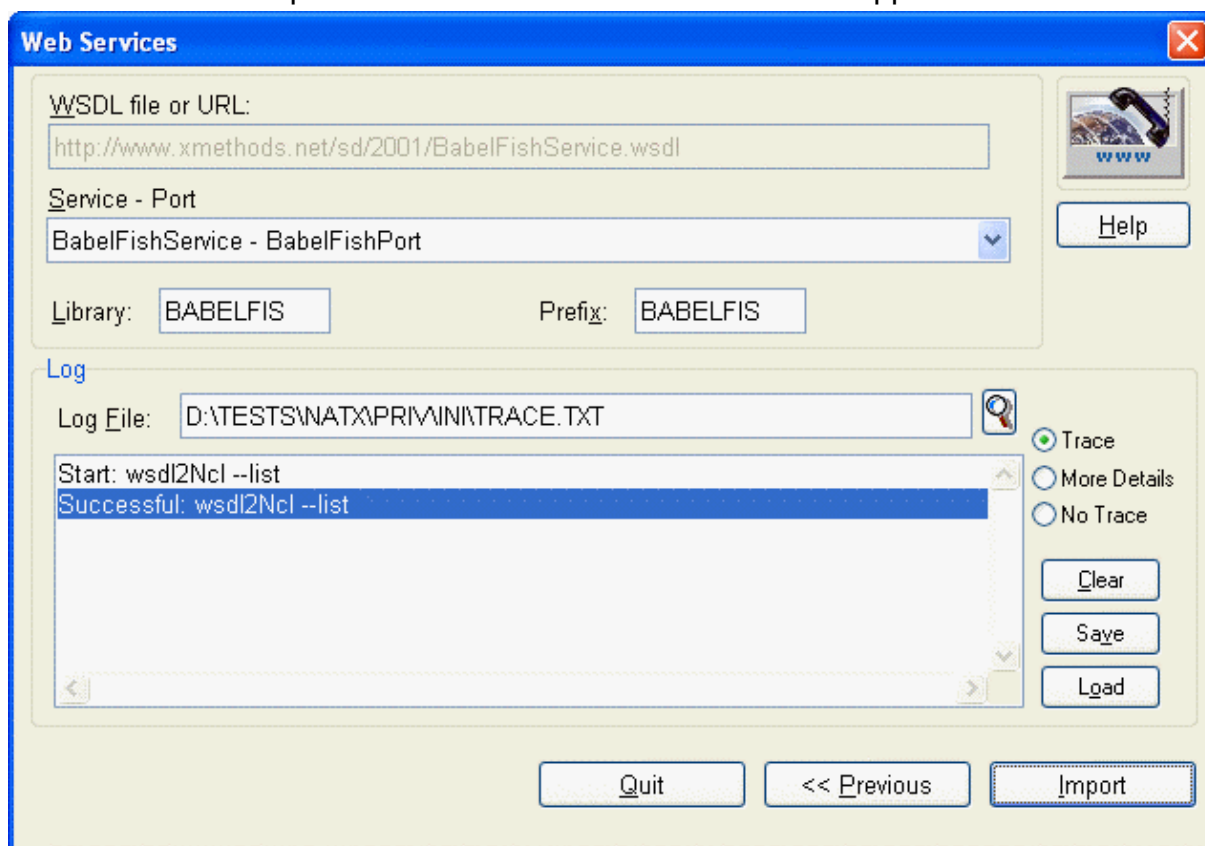
Les champs User et Password sont facultatifs et ne sont à renseigner que si le serveur fournissant le WSDL réclame un utilisateur pour des raisons de confidentialité ou de sécurité. Log file est le chemin où sera sauvegardé le fichier log que vous pouvez voir défiler dans la ListBox en dessous.

Les radio boutons Trace, More Trace et No Trace sont là pour déterminer la quantité d'informations à tracer au niveau du groupe Log. (Clear efface le fichier log, Save le sauvegarde, Load recharge le fichier log à partir du chemin préciser au niveau du champ Log File).

Le bouton List Services >> liste les services exposés dans le WSDL.

2. Appuyer sur List Services >> pour lancer l'interpréteur de WSDL.

3. Une fois l'interprétation terminée une nouvelle fenêtre apparaît :

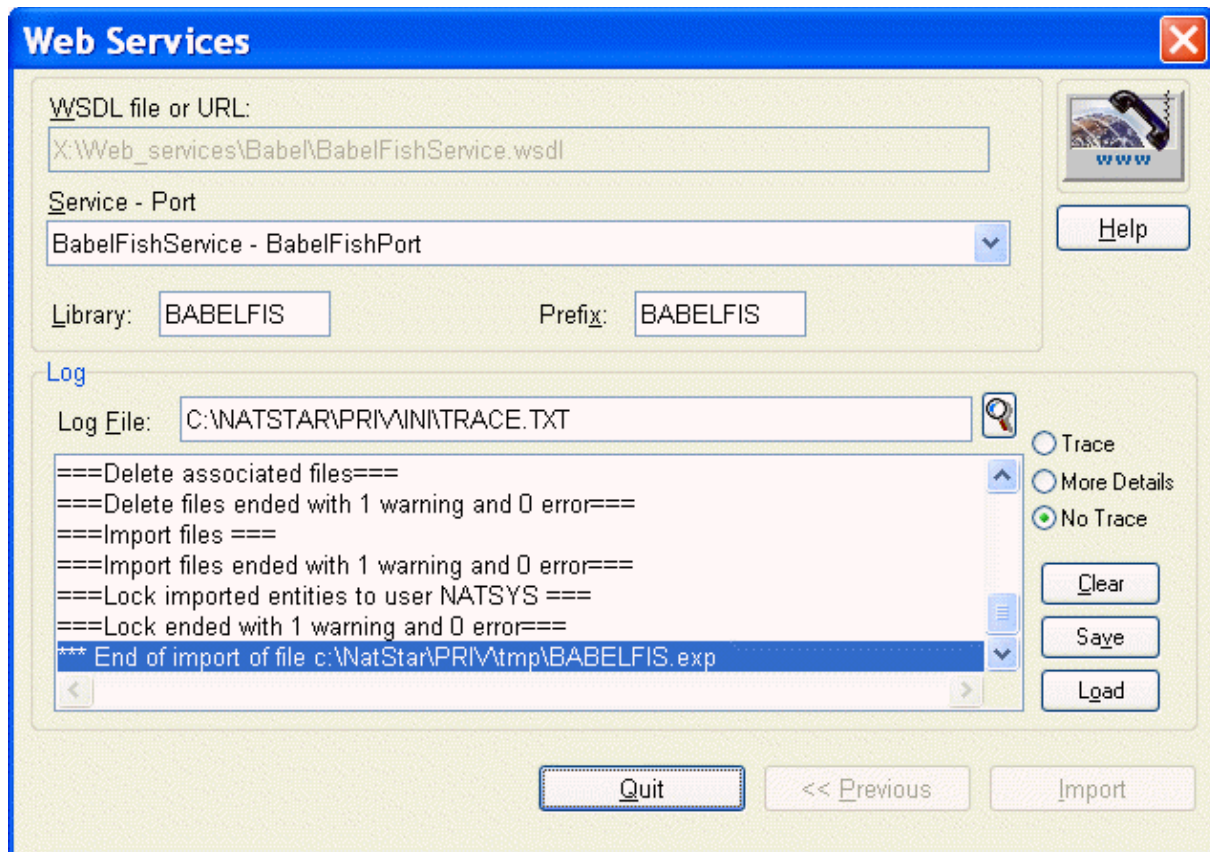


Le champ Service – Port liste les services et leurs ports déclarés dans le WSDL. En général, vous n'aurez qu'un seul choix.

Dans le champ Library, on vous propose un nom de bibliothèque que vous pouvez changer et un champ préfixe modifiable pour préfixer les fonctions/instructions générées.

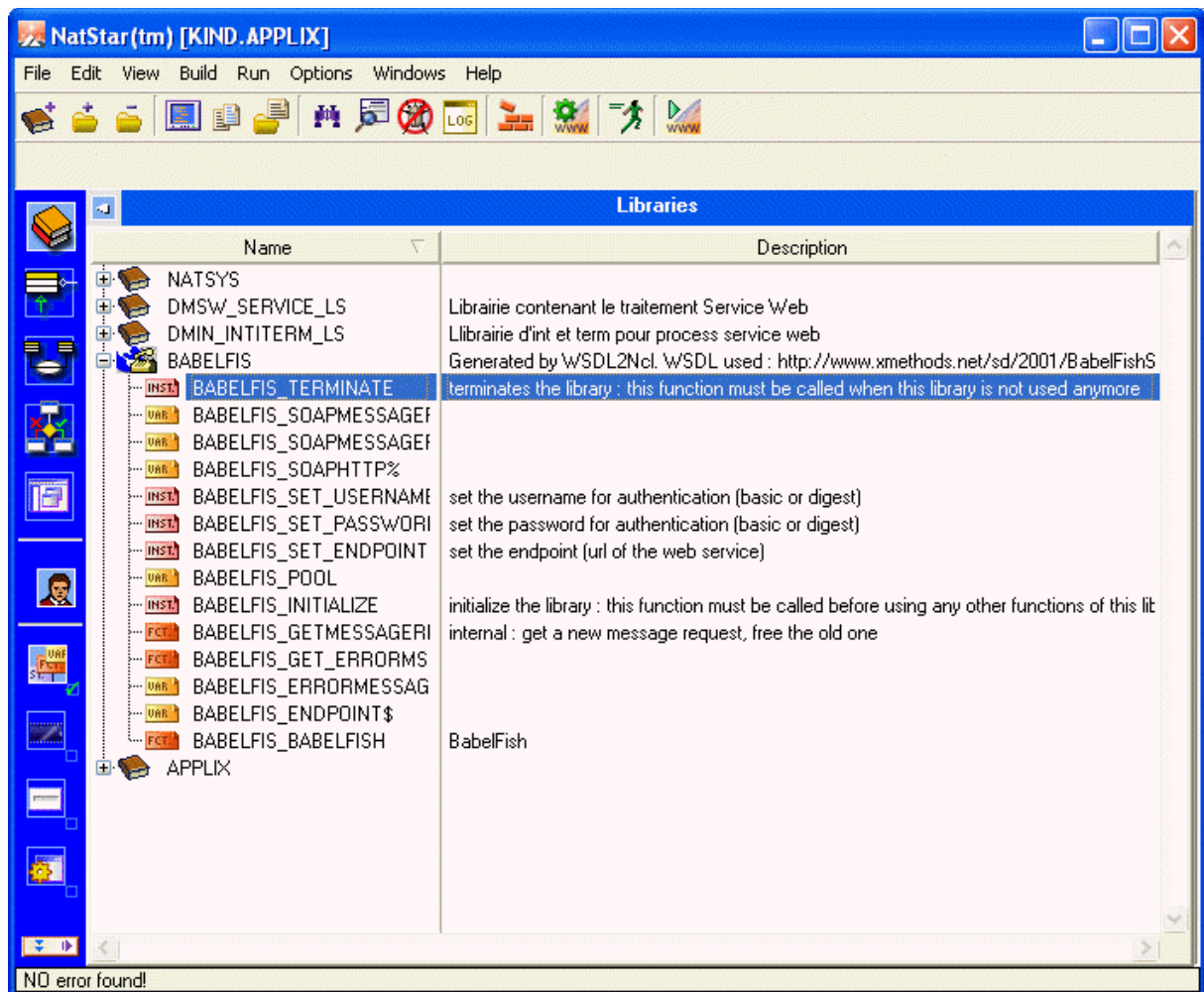
Le bouton Previous pour revenir à la fenêtre précédente et Import pour générer la bibliothèque.

4. Cliquer sur le bouton Import.
5. Vérifier dans le groupe Log qu'il n'y a pas d'erreurs.



6. Appuyer sur le bouton Quit.

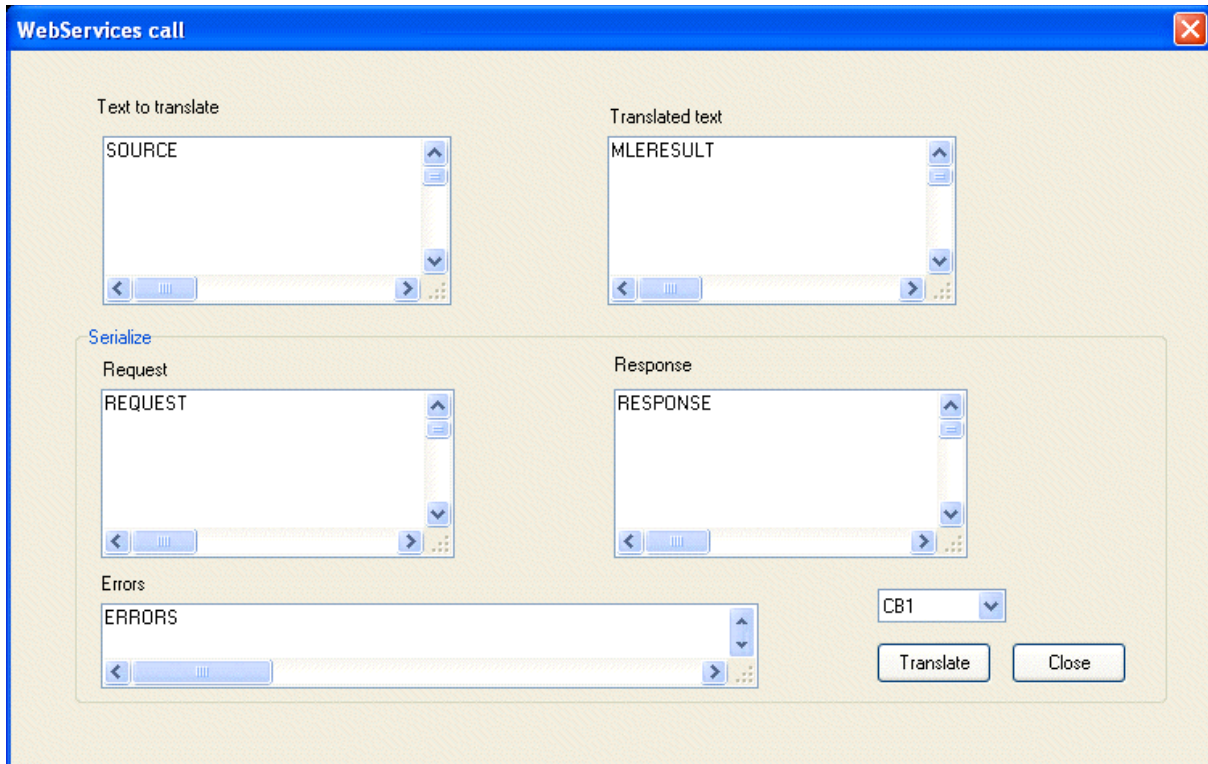
7. Sélectionner la vue des librairies ou rafraîchissez-la en appuyant sur [F7] vous verrez une nouvelle librairie de nom BabelFis si vous n'avez pas modifié le nom proposé par le wizard.



8. Dans cette librairie, il y a deux instructions et une fonction qui nous intéressent. BABELFIS_INITIALIZE pour initialiser l'appel, BABELFIS_BABELFIS pour l'appel au service web, et BABELFIS_TERMINATE pour libérer la mémoire

Une documentation brève de chaque fonction/instruction générée est disponible en faisant clic droit Modify/Definition.

9. Soit la fenêtre suivante à partir de laquelle on veut faire l'appel au service BabelFish :



The dialog box titled "WebServices call" contains five text areas for MLEs: "Text to translate" (containing "SOURCE"), "Translated text" (containing "MLERESULT"), "Request" (containing "REQUEST"), "Response" (containing "RESPONSE"), and "Errors" (containing "ERRORS"). There is a "Serialize" label above the "Request" and "Response" areas. A dropdown menu labeled "CB1" is positioned to the right of the "Errors" area. At the bottom right are "Translate" and "Close" buttons.

Cette fenêtre contient cinq MLE :

- MLERESULT, le résultat de la traduction,
- REQUEST, la sérialisation de la requête Soap en XML (utile pour des fins de réglage ou de debugging),
- RESPONSE, la sérialisation de la réponse Soap en XML (utile pour des fins de réglage ou de debugging),
- ERRORS, la MLE où on affichera les erreurs,
- un COMBO, qui contiendra les deux chaînes "en_fr" et "fr_en" respectivement pour la traduction de l'anglais vers le français et inversement,
- le bouton Translate qui fait l'appel dont voici le code :

```

LOCAL VARSTRING mode
LOCAL VARSTRING sourcedata
LOCAL VARSTRING result
LOCAL str$(20000), i%
LOCAL requestString$(20000)
LOCAL responseString$(20000)

DELETE FROM MLERESULT
str$ = ""
;Initializes the Web service CALL
BABELFIS_INITIALIZE
;creates "translationmode" string parameter with "en_fr" as a ;value
;<translationmode xsi:type="xsd:string">en_fr</translationmode>
VARSTRING_NEW BABELFIS_POOL,mode,CB1
;creates "sourcedata " string parameter with the contents of MLE SOURCE for
value
;<sourcedata xsi:type="xsd:string">Any text</sourcedata>
FOR i% = 0 TO linecount% (SOURCE) -1

```

```

    str$= str$ & source[i%]
ENDFOR
VARSTRING_NEW BABELFIS_POOL,sourcedata,str$
;when we generate a web service we look for operation (method) ;we asked
for we have a quick look at its parameters to ;determine want kind of
parameters we have to instantiate
if BABELFIS_BABELFISH (mode,sourcedata,result)
errors.text = BABELFIS_GET_ERRORMSG$
BABELFIS_TERMINATE
RETURN
Endif
;Optional serialization
NS_SOAPMSGREQUEST_SERIALIZE BABELFIS_SOAPMESSAGEREQUEST%, @requestString$,
SIZEOF requestString$
request.text = requestString$
NS_SOAPHTTP_GET_RESPONSESTRING BABELFIS_SOAPHTTP%, @responseString$, SIZEOF
responseString$
RESPONSE.text = responseString$

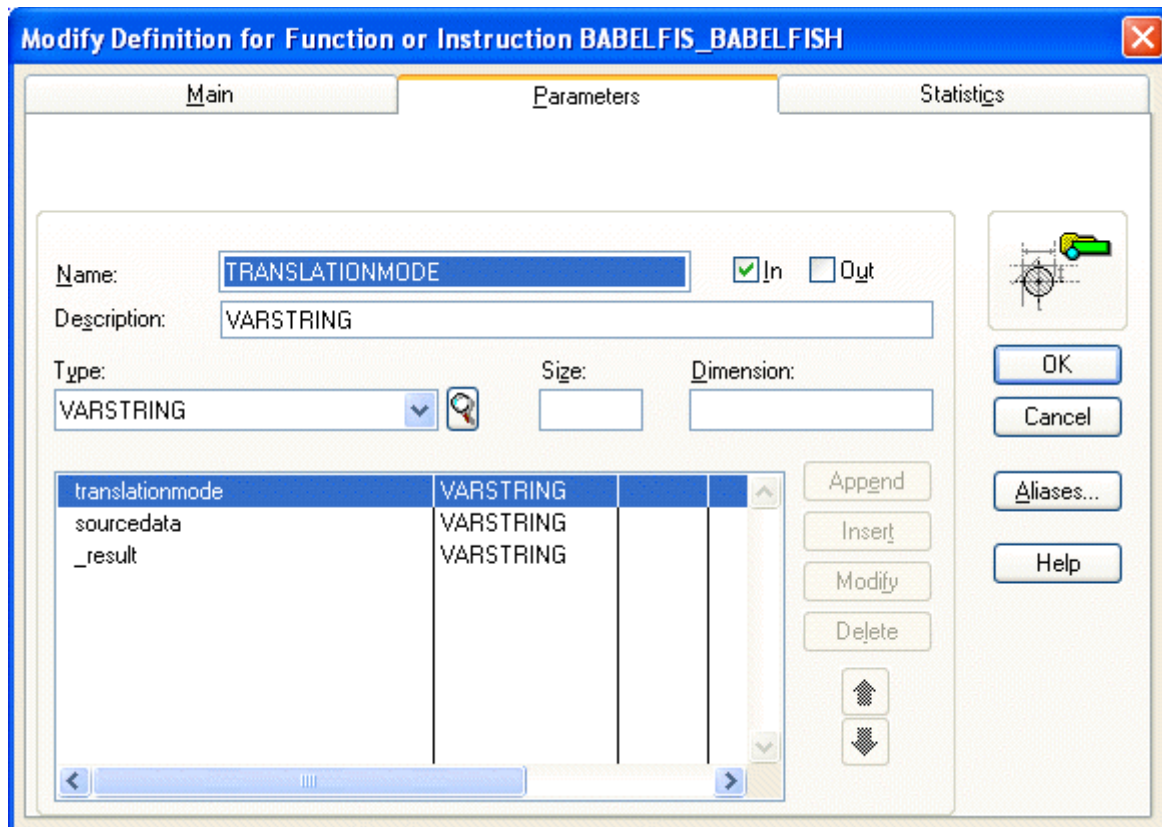
;Only when assign the value of the control we could go beyond ;the 255
character limit
MLERESULT.text = VTYP_PCSTRING(VARSTRING_GET_PSTR(result)).value
;Free the memory
BABELFIS_TERMINATE

```

La principale fonction dans ce code est BABELFIS_BABELFISH qui correspond à l'appel au service Web.

Comme on avait dit précédemment ce service prend deux chaînes paramètres en entrée et une chaîne en sortie.

Ces trois chaînes de type VARSTRING sont retrouvées au niveau du prototype de la fonction BABELFIS_BABELFISH.



Le Type varstring est un nouveau type de chaîne défini dans le fichier NSVTYP.NCL à utiliser avec les fonctions d'appel des services Web. Ce type de chaîne doit être initialisé par l'instruction VARSTRING_NEW. Pour la transformation inverse de Varstring en CSTRING on utilise la syntaxe suivante : TYP_PCSTRING(VARSTRING_GET_PSTR(result)).value

La sérialisation de la requête et de la réponse peut se faire avec NS_SOAPMSGREQUEST_SERIALIZE et NS_SOAPHTTP_GET_RESPONSESTRING du fichier NSSOAP.NCL

Types, variables, fonctions et instructions NCL à utiliser avec les services Web

Pour utiliser ces types et fonctions, le service NSVTYP.NCL doit être ajouté au workspace NatStar (NS-DK ou NatWeb).

Champ ISNULL (Librairie NSVTYP)

Un champ ISNULL de type INT(1) a été introduit pour chaque segment paramètre ISNULL pourrait être mis à TRUE% dans le cas où le paramètre représenté par le segment pourrait être NULL (paramètre optionnel comme détail de factures par exemple).

Fonction <Prefix>_<OperationName> (Librairie NSVTYP)

Fonction générée par le wizard appelant un service Web

Syntaxe	<Prefix>_<OperationName>	
Valeur retournée	INT(4)	Code d'erreur

1. <Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard
2. <OperationName> est le nom de l'opération ou méthode du Web Service
3. Un service Web peut exposer plusieurs opérations
4. Une documentation brève de chaque fonction/instruction générée est disponible en faisant clic droit "Additional/Description"

Exemple :

```
LOCAL VARSTRING mode
LOCAL VARSTRING sourcedata
LOCAL VARSTRING result
LOCAL str$(20000), i%
LOCAL requestString$(20000)
LOCAL responseString$(20000)

DELETE FROM MLERESULT
str$ = ""
BABELFIS_INITIALIZE
;creates "translationmode" string parameter with "en-fr" as a value
;<translationmode xsi:type="xsd:string">en_fr</translationmode>
VARSTRING_NEW BABELFIS_POOL,mode,CB1
;creates "sourcedata" string parameter with the contents of MLE SOURCE for
;value <sourcedata xsi:type="xsd:string">Any text</sourcedata>
FOR i% = 0 TO linecount%(SOURCE) -1
    str$= str$ & source[i%]
ENDFOR
VARSTRING_NEW BABELFIS_POOL,sourcedata,str$
;when we generate an web service we look for operation (methode) we asked
for
;we have a quick look at its parameters to determine want kind of
parameters ;we have to instanciate
;Pefix BABELFIS, OperationName BABELFISH
if BABELFIS_BABELFISH (mode,sourcedata,result)
    ; errors.text = BABELFIS_errorMessage$
    errors.text = BABELFIS_GET_ERRORMSG$
BABELFIS_TERMINATE
RETURN
endif
NS_SOAPMSGREQUEST_SERIALIZE BABELFIS_SOAPMESSAGEREQUEST%, @requestString$,
SIZEOF requestString$
request.text = requestString$
NS_SOAPHTTP_GET_RESPONSESTRING BABELFIS_SOAPHTTP%, @responseString$, SIZEOF
responseString$
RESPONSE.text = responseString$

;Only when assign the value of the control we could go beyond the 255
```

```
;character limit  
MLERESULT.text = VTYP_PCSTRING(VARSTRING_GET_PSTR(result)).value  
  
BABELFIS_TERMINATE
```

Voir aussi <Prefix> GET_ERRORMSG\$

Fonction <Prefix>_GET_ERRORMSG\$ (Librairie NSVTYP)

Fonction « accesseur » générée par le wizard retourne le message d'erreur

Syntaxe	<Prefix>_GET_ERRORMSG\$	
Valeur retournée	CSTRING(255)	le message d'erreur

On peut aussi utiliser directement la variable globale BABELFIS_errorMessage\$

Exemple :

```
...  
if BABELFIS_BABELFISH (mode,sourcedata,result)  
errors.text = BABELFIS_GET_ERRORMSG$  
BABELFIS_TERMINATE  
RETURN  
Endif  
...
```

Voir aussi <Prefix>_<OperationName>

FONCTION VARARRAY_GET_ELEMENTPTR (Librairie NSVTYP)

Retourne un pointeur vers un élément du tableau. Il est ensuite possible de modifier cet élément en utilisant les segments de cast VTYP_*.

Syntaxe	VARARRAY_GET_ELEMENTPTR (Varray, position[])		
Paramètres	Varray	VARARRAY	variable de type VARARRAY
	position	INTEGER	tableau indiquant la position de l'élément sur lequel on voudrait récupérer un pointeur. Si le tableau a 3 dimensions, position doit être un tableau à 3 éléments indiquant l'indice de l'élément dans les 3 dimensions.
Valeur retournée	POINTER	pointeur sur un élément du VARARRAY	

Il est possible de modifier la valeur de l'élément.

Exemple :

```
...
LOCAL CSTRING array2[18]
LOCAL INTEGER size1[3]
LOCAL INTEGER position1[3]
LOCAL INTEGER position2
LOCAL POINTER ptr

array1[3][1][2] = 23
size1[0] = 5
size1[1] = 2
size1[2] = 3
VARARRAY_NEW pool,varray1, @array1, 3, size1, SIZEOF array1[0][0][0]

position1[0] = 3
position1[1] = 1
```

```
position1[2] = 2
ptr = VARARRAY_GET_ELEMENTPTR (varray1,position1)

if VTYP_PINT4(ptr).value = 23
...
ENDIF
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY GET ELEMENTPTR, VARARRAY GET ARRAY, VARARRAY GET SIZES PTR, VARARRAY GET SIZES, VARARRAY SETNULL, VARARRAY ISNULL

Fonction VARARRAY_GET_NBDIMS (Librairie NSVTYP)

Retourne le nombre de dimensions du tableau VARARRAY.

Syntaxe	VARARRAY_GET_NBDIMS (Varray)		
Paramètre	Varray	VARARRAY	variable de type VARARRAY
Valeur retournée	INT(4)	le nombre de dimensions du VARARRAY	

Exemple :

```
...
LOCAL VARARRAY varray1
LOCAL INT array1[3][3], size%

array1[2][2] = 23
VARARRAY_NEW_2D pool,varray1, @array1, 3, 3, SIZEOF array1[0][0]

size% = VARBINARY_GET_SIZE(varray1)
If size% = 2
...
Endif
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY GET ELEMENTPTR, VARARRAY GET ARRAY, VARARRAY GET SIZES_PTR, VARARRAY GET SIZES, VARARRAY SETNULL, VARARRAY ISNULL

Fonction `VARARRAY_GET_SIZES_PTR` (Librairie `NSVTYP`)

Retourne un pointeur sur le tableau d'INTEGER des tailles des dimensions.

Syntaxe	<code>VARARRAY_GET_SIZES_PTR</code> (<i>Varray</i>)		
Paramètre	Varray	VARARRAY	I variable de type VARARRAY
Valeur retournée	POINTER	pointeur sur un élément du VARARRAY	

Exemple :

```
...
LOCAL INTEGER size1[3]
LOCAL INTEGER size2[3]
LOCAL POINTER ptr, size%

array1[3][1][2] = 23
size1[0] = 5
size1[1] = 2
size1[2] = 3
VARARRAY_NEW pool, varray1, @array1, 3, size1, SIZEOF array1[0][0][0]

size% = VARBINARY_GET_SIZE(varray1)
ptr = VARARRAY_GET_SIZES_PTR (varray1)
mov ptr, @size2, size% * 4
if size2[0] = 5
...
endif
```

Voir aussi `<Prefix> INITIALIZE`, `<Prefix> TERMINATE`, `VARARRAY_GET_ELEMENTPTR`,
`VARARRAY_GET_ARRAY`, `VARARRAY_GET_SIZES`, `VARARRAY_SETNULL`, `VARARRAY_ISNULL`

FONCTION VARARRAY_ISNULL (Librairie NSVTYP)

Teste si la valeur du VARARRAY est à NULL.

Syntaxe	VARARRAY_ISNULL (<i>Varray</i>)		
Paramètre	Varray	VARARRAY	I variable de type VARARRAY
Valeur retournée	INT(1)	FALSE% ou TRUE%	

1. Le VARARRAY est invalide avant que VARARRAY_NEW ne soit appelé. Sa valeur n'est pas NULL.
2. Aucune des fonctions associées au VARARRAY ne doit être appelée (en dehors de VARARRAY_ISNULL) lorsque cette fonction a été appelée.

Exemple :

```
...
LOCAL AM_KEYWORDREQUEST keywordRequest
...
IF VARARRAY_ISNULL(productInfo.arrName) = FALSE%
...
ENDIF
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY GET ELEMENTPTR, VARARRAY GET ARRAY, VARARRAY GET SIZES PTR, VARARRAY GET SIZES, VARARRAY SETNULL

Fonction VARBINARY_GET_PTR (Librairie NSVTYP)

Retourne un pointeur vers le buffer sous-jacent.

Syntaxe	VARBINARY_GET_PTR (<i>Vbinary</i>)		
Paramètre	Vbinary	VARBINARY	I variable de type VARBINARY
Valeur retournée	POINTER	Pointeur sur le buffer sous-jacent	

Il est alors possible de manipuler directement le contenu du VARBINARY.

Exemple :

```
...  
VARBINARY_NEW_SIZED ROUND2BA_pool, _result, size1  
ptr2 = VARBINARY_GET_PTR(_result)  
...
```

Voir aussi *<Prefix> INITIALIZE*, *<Prefix> TERMINATE* , *VARBINARY_GET*, *VARBINARY_NEW_SIZED*, *VARBINARY_NEW*, *VARBINARY_GET*, *VARBINARY_SETNULL*, *VARBINARY_ISNULL*

Fonction VARBINARY_GET_SIZE (Librairie NSVTYP)

Retourne la taille du contenu du VARBINARY.

Syntaxe	VARBINARY_GET_SIZE (<i>Vbinary</i>)		
Paramètre	Vbinary	VARBINARY	I variable de type VARBINARY
Valeur retournée	INT(4)	taille du contenu du VARBINARY	

Exemple :

```
...
VARBINARY_NEW pool,vbinary1, @myString, LENGTH myString +1

length% = VARBINARY_GET_SIZE(vbinary1)
if (length% <> 477)
RESULTVARBINARY = "PROBLEM"
return DEFRET%
endif
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE , VARBINARY_GET, VARBINARY_NEW SIZED,
VARBINARY_NEW, VARBINARY_GET PTR, VARBINARY_SETNULL, VARBINARY_ISNULL

Fonction **VARBINARY_ISNULL** (Librairie NSVTYP)

Test si la valeur du VARBINARY est à « NULL ».

Syntaxe	VARBINARY_ISNULL (<i>Vbinary</i>)		
Paramètre	Vbinary	VARBINARY	I variable de type VARBINARY
Valeur retournée	INT(1)	FALSE% ou TRUE%	

1. Le VARBINARY est invalide avant que VARBINARY_NEW ne soit appelé. Sa valeur n'est pas NULL.
2. Aucune des fonctions associées au VARBINARY ne doit être appelée (en dehors de VARBINARY_ISNULL) lorsque cette fonction a été appelée.

Exemple :

```
...  
LOCAL AM_KEYWORDREQUEST keywordRequest  
...  
IF VARBINARY_ISNULL(productInfo.listName) = FALSE%  
...  
ENDIF  
...
```

Voir aussi *<Prefix> INITIALIZE*, *<Prefix> TERMINATE*, *VARBINARY_GET*, *VARBINARY_NEW SIZED*, *VARBINARY_NEW*, *VARBINARY_GET_PTR*, *VARBINARY_SETNULL*

Fonction VARSTRING_GET_PSTR (Librairie NSVTYP)

Permet de récupérer un pointeur vers la chaîne CSTRING sous-jacente.

Syntaxe	VARSTRING_GET_PSTR (<i>Vstring</i>)		
Paramètre	Vstring	VARSTRING	I variable de type VARSTRING
Valeur retournée	POINTER	Pointeur sur la chaîne cstring sous jacente	

1. La chaîne pointée NE doit PAS être modifiée
2. VARSTRING_GET_PSTR peut être utilisé avec le segment VTYP_PCSTRING pour récupérer sa valeur. VTYP_PCSTRING permet de "caster" le pointeur retourné par VARSTRING_GET_PSTR en un CSTRING.
3. Le champ valeur est limité à 65481 octets.

Exemple :

```
...
BABELFIS_INITIALIZE
VARSTRING_NEW BABELFIS_POOL,mode,CB1
str$= 'hi world'
VARSTRING_NEW BABELFIS_POOL,sourcedata,str$
if BABELFIS_BABELFISH (mode,sourcedata,result)
errors.text = BABELFIS_errorMessage$
endif
MLERESULT.text = VTYP_PCSTRING(VARSTRING_GET_PSTR(result)).value
BABELFIS_TERMINATE
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_ADD_VARSTRING, VARSTRING_ADD_PSTR, VARSTRING_SETNULL, VARSTRING_ISNULL

Fonction VARSTRING_GETLENGTH (Librairie NSVTYP)

Renvoie la longueur d'une chaîne VARSTRING.

Syntaxe	VARSTRING_GETLENGTH (<i>Vstring</i>)			
Paramètre	Vstring	VARSTRING		variable de type VARSTRING
Valeur retournée	INT(4)	longueur de la Varstring		

Exemple :

```
...
myString$= 'hello World'
VARSTRING_NEW pool,vstring1, myString$
length% = VARSTRING_GETLENGTH(vstring1)
...
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_SETNULL

Fonction VARSTRING_ISNULL (Librairie NSVTYP)

Teste si la valeur du VARSTRING est à NULL.

Syntaxe	VARSTRING_ISNULL (<i>Vstring</i>)		
Paramètre	Vstring	VARSTRING	I variable de type VARSTRING
Valeur retournée	INT(1)	FALSE% ou TRUE%	

1. Le VARSTRING est invalide avant que VARSTRING_NEW ne soit appelé. Sa valeur n'est pas NULL.
2. Aucune des fonctions associées au VARSTRING ne doit être appelée (en dehors de VARSTRING_ISNULL) lorsque cette fonction a été appelée.

Exemple :

```
...
LOCAL AM_KEYWORDREQUEST keywordRequest
...
AM_INITIALIZE
VARSTRING_NEW AM_pool,keywordRequest.keyword,ENTRYKEYWORD.TEXT
VARSTRING_NEW AM_pool,keywordRequest.page,"1"
VARSTRING_NEW AM_pool,keywordRequest.mode,"books"
VARSTRING_NEW AM_pool,keywordRequest.tag,"associates"
VARSTRING_NEW AM_pool,keywordRequest.type,"lite"
VARSTRING_SETNULL keywordRequest.locale
VARSTRING_SETNULL keywordRequest.price

res% = AM_KEYWORDSEARCHREQUEST(keywordRequest,productInfo)

IF res% <> 0
  AM_TERMINATE
  RETURN defret%
ENDIF

totalResults% = \
VTYP_PCSTRING(VARSTRING_GET_PSTR(productInfo.totalResults)).value

INSERT AT END "Total results " & totalResults% TO MLERESULT

IF VARSTRING_ISNULL(productInfo.listName) = FALSE%
  INSERT AT END "List name " & \
VTYP_PCSTRING(VARSTRING_GET_PSTR(productInfo.listName)).value TO MLERESULT
ENDIF
...
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_ADD VARSTRING, VARSTRING_ADD_PSTR, VARSTRING_SETNULL

Instruction <Prefix>_INITIALIZE (Librairie NSVTYP)

Instruction générée par le wizard. Initialise le mémoire commune ou pool, le client Soap HTTP, les variables globales et l'adresse du service Web.

Syntaxe	<Prefix>_INITIALIZE
---------	---------------------

1. <Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard.
2. Il est absolument nécessaire d'appeler cette instruction avant tout appel aux services Web.
3. Une documentation brève de chaque fonction/instruction générée est disponible en faisant un clic droit et en activant le menu "Additional/Description".

Exemple :

```
; INSTRUCTION INITIALIZE générée par le wizard
VARPOOL_NEW BABELFIS_pool
BABELFIS_soapHttp% = NS_SOAPHTTP_NEW
BABELFIS_soapMessageRequest% = 0
BABELFIS_soapMessageResponse% = 0
BABELFIS_endpoint$ = "http://services.xmethods.net:80/perl/soaplite.cgi"
```

Voir aussi <Prefix> TERMINATE

Instruction <Prefix>_SET_ENDPOINT (Librairie NSVTYP)

Positionne l'adresse URL du service web.

Syntaxe	<Prefix>_SET_ENDPOINT (endpoint\$)		
Paramètre	endpoint\$	CSTRING	adresse URL

1. <Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard.
2. Il n'est généralement pas nécessaire d'appeler cette instruction puisque l'adresse URL du service web est positionnée automatiquement par l'instruction <Prefix>_INITIALIZE.

Voir aussi <Prefix>_INITIALIZE

Instruction <Prefix>_SET_PASSWORD (Librairie NSVTYP)

Positionne le mot de passe pour l'authentification (basic ou digest) auprès du service web.

Syntaxe	<Prefix>_SETPASSWORD (password\$)		
Paramètre	password\$	CSTRING	mot de passe

<Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard.

Voir aussi <Prefix> SET USERNAME

Instruction <Prefix>_SET_USERNAME (Librairie NSVTYP)

Positionne le nom d'utilisateur pour l'authentification (basic ou digest) auprès du service web.

Syntaxe	<Prefix>_SET_USERNAME (username\$)		
Paramètre	username\$	CSTRING	nom d'utilisateur

<Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard.

Voir aussi <Prefix> SET PASSWORD

Instruction <Prefix>_TERMINATE (Librairie NSVTYP)

Instruction générée par le wizard. Libère la mémoire commune ou Pool et par conséquent toutes les variables créées dans le pool, le client Soap HTTP et l'objet requête.

Syntaxe	<Prefix>_TERMINATE
----------------	---------------------------------

1. <Prefix> est le préfixe que vous aviez validé lors de la génération de la bibliothèque par le wizard.
2. Il est absolument nécessaire d'appeler cette instruction à la fin de vos traitements sinon vous risquez d'avoir des fuites mémoire.
3. Une fois cette instruction appelée il ne faut plus utiliser les variables créées sur le pool
4. Une documentation brève de chaque fonction/instruction générée est disponible en faisant un clic droit pour activer le menu contextuel "Additional/Description"

Exemple :

```
; INSTRUCTION Terminate générée par le wizard
NS_SOAPHTTP_DISPOSE BABELFIS_soapHttp%
NS_SOAPMSGREQUEST_DISPOSE BABELFIS_soapMessageRequest%
VARPOOL_DISPOSE BABELFIS_pool
```

Voir aussi <Prefix>_INITIALIZE

Instruction `VARARRAY_GET_ARRAY` (Librairie NSVTYP)

Copie le contenu du `VARARRAY` dans un tableau NCL (ou un buffer quelconque).

Syntaxe	<code>VARARRAY_GET_ARRAY Varray, pBuffer, Size%</code>		
Paramètres	Varray	VARARRAY	variable de type VARARRAY
	pBuffer	POINTER	pointeur vers le tableau NCL
	Size%	INT(4)	taille du tableau NCL. Peut être facilement obtenu en utilisant <code>SIZEOF</code> sur le tableau ou buffer.

Exemple :

```

LOCAL VARARRAY inputFloatArrayVar1Dim
LOCAL VARARRAY resultVar1Dim
LOCAL NUM inputFloatArray(4) [3]
LOCAL NUM resultArray(4) [3]
...
VARARRAY_GET_ARRAY resultVar1Dim, @resultArray, SIZEOF resultArray
...

```

Voir aussi `<Prefix> INITIALIZE`, `<Prefix> TERMINATE`, `VARARRAY GET ELEMENTPTR`, `VARARRAY GET_ARRAY`, `VARARRAY GET_SIZES_PTR`, `VARARRAY GET_SIZES`, `VARARRAY SETNULL`, `VARARRAY_ISNULL`

Instruction `VARARRAY_GET_SIZES` (Librairie NSVTYP)

Copie le tableau des tailles des dimensions dans le buffer donné (généralement un tableau d'INTEGER suffisamment grand).

Syntaxe	<code>VARARRAY_GET_SIZES</code> <i>Varray, pArraySizes</i>		
Paramètres	Varray	VARARRAY	Variable de type VARARRAY
	pArraySizes	POINTER	Pointeur vers le tableau NCL

Exemple :

```
...  
LOCAL INTEGER size1[1]  
...  
VARARRAY_GET_SIZES inputFloatArray,@size1  
...
```

Voir aussi `<Prefix> INITIALIZE`, `<Prefix> TERMINATE`, `VARARRAY GET ELEMENTPTR`, `VARARRAY GET ARRAY`, `VARARRAY GET SIZES PTR`, `VARARRAY SETNULL`, `VARARRAY ISNULL`

Instruction VARARRAY_NEW (Librairie NSVTYP)

Crée un nouvel objet VARARRAY dans le Pool (mémoire commune) et l'initialise avec le contenu du tableau passée en paramètre.

Syntaxe	VARARRAY_NEW <i>Pool, Varray, pArray, nbDims, arDimensions[], elementSize%</i>			
Paramètres	Pool	VARPOOL	I	variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Varray	VARARRAY	I/O	variable de type VARARRAY à initialiser
	pArray	POINTER	I	pointeur vers le tableau sous-jacent
	nbDims	INT(4)	I	nombre de dimension du tableau ci-dessus
	arDimensions	INTEGER	I	tableau indiquant la taille de chacune des dimensions. Si le tableau a 3 dimensions, arDimensions doit être un tableau à 3 éléments indiquant la taille de chacune des trois dimensions.

	elementSize%	INT(4)	I	taille d'un élément du tableau. elementSize% peut être facilement obtenu en utilisant SIZEOF sur un élément quelconque du tableau
--	--------------	--------	---	--

1. VarArray représente un tableau à 1 ou plusieurs dimensions.
2. Ce paramètre est généré pour s'interfacer avec le type array de Soap mais cela reste transparent pour vous

Exemple :

```
LOCAL VARARRAY varray1
LOCAL INT array1[5][2][3]
LOCAL INTEGER size1[3]

array1[3][1][2] = 23
size1[0] = 5
size1[1] = 2
size1[2] = 3
VARARRAY_NEW pool,varray1, @array1, 3, size1, SIZEOF array1[0][0][0]
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY_NEW 1D, VARARRAY_NEW 2D

Instruction VARARRAY_NEW_1D (Librairie NSVTYP)

Crée un nouvel objet VARARRAY à une seule dimension dans le Pool (mémoire commune) et l'initialise avec le contenu du tableau passé en paramètre.

Syntaxe	VARARRAY_NEW_1D <i>Pool, Varray, pArray, Dims, elementSize%</i>			
Paramètres	pool	VARPOOL	I	variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Varray	VARARRAY	I/O	variable de type VARARRAY à initialiser
	pArray	POINTER	I	pointeur vers le tableau à 1 dimension sous-jacent
	Dims	INT(4)	I	dimension ou nombre d'éléments du tableau ci-dessus
	elementSize%	INT(4)	I	taille d'un élément du tableau. elementSize% peut être facilement obtenu en utilisant SIZEOF sur un élément quelconque du tableau

Ce paramètre est généré pour s'interfacer avec le type array de Soap mais cela reste transparent.

Exemple :

```
LOCAL VARARRAY varray1
LOCAL INT array1 [3]

array1[2] = 23
VARARRAY_NEW_1D pool,varray1, @array1, 3, SIZEOF array1[0]
...
```

Voir aussi <Prefix>_INITIALIZE, <Prefix>_TERMINATE, VARARRAY_NEW, VARARRAY_NEW_2D

Instruction VARARRAY_NEW_2D (Librairie NSVTYP)

Crée un nouvel objet VARARRAY à deux dimensions dans le pool (mémoire commune) et l'initialise avec le contenu du tableau passé en paramètre.

Syntaxe	VARARRAY_NEW_2D <i>Pool, Varray, pArray, Dims1, Dims2, elementSize%</i>			
Paramètres	pool	VARPOOL	I	variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Varray	VARARRAY	I/O	variable de type VARARRAY à initialiser
	pArray	POINTER	I	pointeur vers le tableau à 1 dimension sous-jacent
	Dims1	INT(4)	I	nombre d'éléments de la 1ère dimension du tableau ci-dessus
	Dims2	INT(4)	I	nombre d'éléments de la 2ème dimension du tableau ci-dessus
	elementSize%	INT(4)	I	taille d'un élément du tableau. elementSize% peut être facilement

			obtenu en utilisant sizeof sur un élément quelconque du tableau
--	--	--	--

Ce paramètre est généré pour s'interfacer avec le type array de Soap mais cela reste transparent.

Exemple :

```
LOCAL VARARRAY varray1
LOCAL INT array1[3][3]

array1[2][2] = 23
VARARRAY_NEW_2D pool,varray1, @array1, 3, 3, sizeof array1[0][0]
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY NEW, VARARRAY NEW 1D

Instruction VARARRAY_SETNULL (Librairie NSVTYP)

Positionne la valeur du VARARRAY à NULL.

Syntaxe	VARARRAY_SETNULL Varray		
Paramètre	Varray	VARARRAY	Variable de type VARARRAY

1. Aucune des fonctions associées au VARARRAY ne doit être appelée (en dehors de VARARRAY_ISNULL) lorsque cette fonction a été appelée.
2. Certains services web peuvent avoir des paramètres ou des champs de paramètres optionnels. Dans NatWeb ces champs seront en général indiqués par la mention [optional] c'est dans ces cas-là que le VARARRAY_SETNULL est à utiliser.

Exemple :

```
LOCAL AM_KEYWORDREQUEST keywordRequest
...
AM_INITIALIZE

VARARRAY_NEW AM_pool, keywordRequest.sort, "+daterank"
VARARRAY_SETNULL keywordRequest.arr
...
res% = AM_KEYWORDSEARCHREQUEST(keywordRequest, productInfo)
...
IF VARARRAY_ISNULL(productInfo.arrName) = FALSE%
...
ENDIF
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARARRAY GET ELEMENTPTR, VARARRAY GET ARRAY, VARARRAY GET SIZES PTR, VARARRAY GET SIZES, VARARRAY ISNULL

Instruction VARBINARY_GET (Librairie NSVTYP)

Copie le contenu d'un VARBINARY dans une chaîne CSTRING.

Syntaxe	VARBINARY_GET <i>Vbinary</i> , <i>pszStr\$</i> , <i>bufferSize%</i>		
Paramètres	Vstring	VARBINARY	variable de type VARSTRING dont on veut copier le contenu
	pszStr\$	POINTER	pointeur sur la chaîne CSTRING
	bufferSize%	INT(4)	taille de la chaîne CSTRING

Exemple :

```
...
VARBINARY_GET vbinary1,@myString2, length%

if (myString2 <> myString)
RESULTVARBINARY = "PROBLEM"
return DEFRET%
endif
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARBINARY GET SIZE, VARBINARY NEW SIZED, VARBINARY NEW, VARBINARY GET_PTR, VARBINARY SETNULL, VARBINARY ISNULL

Instruction VARBINARY_NEW (Librairie NSVTYP)

Crée un nouvel objet VARBINARY dans le Pool (mémoire commune) et l'initialise avec le contenu d'un buffer.

Syntaxe	VARBINARY_NEW <i>Pool, Vbinary, pBuffer\$, bufferSize%</i>			
Paramètres	pool	VARPOOL	I	variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Vbinary	VARBINARY	I/O	variable de type VARBINARY à initialiser
	pBuffer\$	POINTER	I	pointeur sur la chaîne cstring
	bufferSize%	INT(4)	I	taille de la chaîne cstring

1. Vbinary représente un buffer de taille quelconque.
2. Un Vbinary est généré pour les paramètres de type xs:hexBinary, SOAP_ENC:Base64 et xs:base64Binary de SOAP cependant vous n'avez pas à vous en préoccuper.
3. Une VARBINARY est allouée à partir du pool mémoire et détruite lorsque le pool lui-même est détruit.
4. Des fois des services nécessitent des paramètres complexes (segment NCL). Ces segments peuvent contenir des VARBINARY entre autres.

Exemple :

```
LOCAL VARBINARY vbinary1
LOCAL str$
str$= 'hello World'
ROUND2B_INITIALIZE
VARBINARY_NEW ROUND2B_pool,vbinary1, @Str$, LENGTH Str$ +1
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARBINARY GET, VARBINARY GET SIZE, VARBINARY NEW SIZED, VARBINARY GET PTR, VARBINARY SETNULL, VARBINARY ISNULL

Instruction VARBINARY_NEW_SIZED (Librairie NSVTYP)

Crée un nouvel objet VARBINARY d'une taille spécifiée. Le contenu de VARBINARY est alors indéterminé et pourra être modifié en utilisant VARBINARY_GET_PTR.

Syntaxe	VARBINARY_NEW_SIZED <i>pool, vbinary, Size%</i>			
Paramètres	pool	VARPOOL	I	variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Vbinary	VARBINARY	I/O	variable de type VARBINARY à allouer
	Size%	INT(4)	I	taille de la VARBINARY

Exemple :

```
...
VARBINARY_NEW_SIZED ROUND2BA_pool, _result, size1
ptr2 = VARBINARY_GET_PTR(_result)
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARBINARY_GET, VARBINARY_GET_SIZE, VARBINARY_NEW, VARBINARY_GET_PTR, VARBINARY_SETNULL, VARBINARY_ISNULL

Instruction VARBINARY_SETNULL (Librairie NSVTYP)

Positionne la valeur du VARBINARY à NULL.

Syntaxe	VARARRAY_SETNULL <i>Vbinary</i>		
Paramètre	Vbinary	VARBINARY	Variable de type VARBINARY

1. Aucune des fonctions associées au VARBINARY ne doit être appelée (en dehors de VARBINARY_ISNULL) lorsque cette fonction a été appelée.
2. Certains Services Web peuvent avoir des paramètres ou des champs de paramètres optionnels. Dans NatWeb ces champs seront en général indiqués par la mention [optional] c'est dans ces cas là que le VARBINARY_SETNULL est à utiliser.

Exemple :

```
LOCAL AM_KEYWORDREQUEST keywordRequest
...
AM_INITIALIZE

VARBINARY_NEW AM_pool, keywordRequest.sort, "+daterank"
VARBINARY_SETNULL keywordRequest.price
...
res% = AM_KEYWORDSEARCHREQUEST(keywordRequest, productInfo)
...
IF VARBINARY_ISNULL(productInfo.listName) = FALSE%
...
ENDIF
...
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE, VARBINARY GET, VARBINARY NEW SIZED, VARBINARY NEW, VARBINARY GET PTR, VARBINARY ISNULL

Instruction VARPOOL_NEW (Librairie NSVTYP)

Initialise le mémoire commune ou pool.

Syntaxe	VARPOOL_NEW <i>pool</i>		
Paramètre	pool	VARPOOL	c'est la variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel

1. L'instruction VARPOOL_NEW est appelée dans l'instruction générée par le wizard dont le nom est <NomService>_Initialize ou NomService est le préfixe que vous avez validé lors de la génération de la bibliothèque à partir du WSDL
2. Vous n'avez pas à appeler cette instruction.
3. Les types VARSTRING, VARBINARY, VARARRAY sont alloués à partir d'un pool mémoire créé en par VARPOOL_NEW et détruit par VARPOOL_DISPOSE.

Exemple :

```
; INSTRUCTION INITIALIZE
VARPOOL_NEW BABELFIS_pool
BABELFIS_soapHttp% = NS_SOAPHTTP_NEW
BABELFIS_soapMessageRequest% = 0
BABELFIS_soapMessageResponse% = 0
BABELFIS_endpoint$ = "http://services.xmethods.net:80/perl/soaplite.cgi"
```

Voir aussi <Prefix> INITIALIZE

Instruction **VARSTRING_ADD_PSTR** (Librairie NSVTYP)

Concatène une CSTRING à une VARSTRING.

Syntaxe	VARSTRING_ADD_PSTR <i>Vstring, szStr\$</i>			
Paramètres	Vstring	VARSTRING	I/O	variable de type VARSTRING à la quelle ont veut concaténer une CSTRING
	SzStr\$	CSTRING	I	chaîne qu'on concatène à la VARSTRING

Le type VARSTRING représente une chaîne de caractère. La longueur n'est pas limitée. VARSTRING est "immuable", les fonctions pour ajouter une chaîne à une autre créent en réalité une autre chaîne.

Exemple :

```
...
VARSTRING_NEW pool, vstring1, myString
VARSTRING_ADD_PSTR vstring1, "hhhhhhhhhh"
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_ADD_VARSTRING, VARSTRING_GET_PSTR, VARSTRING_SETNULL

Instruction **VARSTRING_ADD_VARSTRING** (Librairie NSVTYP)

Concatène deux VARSTRING. Le résultat se retrouve dans la première.

Syntaxe	VARSTRING_ADD_VARSTRING <i>Vstring, Vstring2</i>			
Paramètres	Vstring	VARSTRING	I/O	variable de type VARSTRING à laquelle on veut concaténer une autre VARSTRING
	Vstring2	VARSTRING	I	variable de type VARSTRING qu'on concatène à la 1ère VARSTRING

Le type VARSTRING représente une chaîne de caractère. La longueur n'est pas limitée. VARSTRING est "immuable", les fonctions pour ajouter une chaîne à une autre créent en réalité une autre chaîne.

Exemple :

```
...  
VARSTRING_NEW pool,vstring1, myString  
VARSTRING_NEW pool,vstring2, "ii"  
VARSTRING_ADD_VARSTRING vstring1,vstring2
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_SETNULL

Instruction VARSTRING_GET (Librairie NSVTYP)

Copie le contenu d'une VARSTRING dans une chaîne CSTRING.

Syntaxe	VARSTRING_GET <i>Vstring, pszStr\$, bufferSize%</i>		
Paramètres	Vstring	VARSTRING	variable de type VARSTRING dont on veut copier le contenu
	pszStr\$	POINTER	pointeur sur la chaîne cstring
	bufferSize%	INT(4)	taille de la chaîne cstring

Exemple :

```
...
LOCAL VARSTRING vstring1
LOCAL CSTRING myString2(500)
VARSTRING_NEW pool,vstring1, "Antoine"
VARSTRING_GET vstring1,@myString2,SIZEOF myString2
```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_ADD VARSTRING, VARSTRING_ADD_PSTR, VARSTRING_GET_PSTR, VARSTRING_SETNULL

Instruction VARSTRING_NEW (Librairie NSVTYP)

Crée un objet VARSTRING dans le pool et l'associe à la variable Varstring passée en paramètre.

Syntaxe	VARSTRING_NEW <i>Pool, Vstring,szStr\$</i>			
Paramètres	pool	VARPOOL		variable Globale de type VARPOOL générée par le wizard et dont le nom se termine par _pool qui est utilisée dans l'appel
	Vstring	VARSTRING	/O	variable de type VARSTRING à initialiser
	SzStr\$	CSTRING		chaîne avec laquelle on initialise la VARSTRING

1. Le type VARSTRING représente une chaîne de caractère. La longueur n'est pas limitée. VARSTRING est "immuable", les fonctions pour ajouter une chaîne à une autre créent en réalité une autre chaîne.
2. Elle est nullable contrairement à une CSTRING.
3. Une VARSTRING est allouée à partir du pool mémoire et détruite lorsque le pool lui-même est détruit.
4. Des fois des services nécessitent des paramètres complexes (segment NCL) Ces segments peuvent contenir des VARSTRING entre autres.

Exemple 1 :

```
LOCAL VARSTRING mode
LOCAL VARSTRING sourcedata
LOCAL str$(20000)

BABELFIS_INITIALIZE
VARSTRING_NEW BABELFIS_POOL,mode,'en-fr'
str$= 'hello World'
VARSTRING_NEW BABELFIS_POOL,sourcedata,str$
...
BABELFIS_TERMINATE
```

Exemple 2 :

```
LOCAL AM_KEYWORDREQUEST keywordRequest
...
AM_INITIALIZE
...
VARSTRING_NEW AM_pool,keywordRequest.mode,"books"
VARSTRING_SETNULL keywordRequest.locale
...
res% = AM_KEYWORDSEARCHREQUEST(keywordRequest,productInfo)
...
AM_TERMINATE
```

Voir aussi <Prefix> INITIALIZE, <Prefix> TERMINATE

Instruction VARSTRING_SETNULL (Librairie NSVTYP)

Positionne la valeur du VARSTRING à NULL.

Syntaxe	VARSTRING_SETNULL <i>Vstring</i>		
Paramètre	Vstring	VARSTRING	variable de type VARSTRING

1. Aucune des fonctions associées au VARSTRING ne doit être appelée (en dehors de VARSTRING_ISNULL) lorsque cette fonction a été appelée.
2. Certains services web peuvent avoir des paramètres ou des champs de paramètres optionnels. Dans NatWeb ces champs seront en général indiqués par la mention [optional] c'est dans ces cas là que le VARSTRING_SETNULL est à utiliser.

Exemple :

```

LOCAL AM_KEYWORDREQUEST keywordRequest
...
AM_INITIALIZE

VARSTRING_NEW AM_pool, keywordRequest.keyword, ENTRYKEYWORD.TEXT
VARSTRING_NEW AM_pool, keywordRequest.page, "1"
VARSTRING_NEW AM_pool, keywordRequest.mode, "books"
VARSTRING_NEW AM_pool, keywordRequest.tag, "associates"
VARSTRING_NEW AM_pool, keywordRequest.type, "lite"
VARSTRING_NEW AM_pool, keywordRequest.devttag, "your key"
VARSTRING_NEW AM_pool, keywordRequest.sort, "+daterank"
VARSTRING_SETNULL keywordRequest.locale
VARSTRING_SETNULL keywordRequest.price

res% = AM_KEYWORDSEARCHREQUEST(keywordRequest, productInfo)

IF res% <> 0
  AM_TERMINATE
  RETURN defret%
ENDIF

totalResults% = \
VTYP_PCSTRING(VARSTRING_GET_PSTR(productInfo.totalResults)).value

INSERT AT END "Total results " & totalResults% TO MLERESULT

IF VARSTRING_ISNULL(productInfo.listName) = FALSE%
  INSERT AT END "List name " & \
VTYP_PCSTRING(VARSTRING_GET_PSTR(productInfo.listName)).value TO MLERESULT
ENDIF
...

```

Voir aussi VARSTRING_NEW, VARSTRING_GETLENGTH, VARSTRING_ADD VARSTRING, VARSTRING_ADD PSTR, VARSTRING_ISNULL

Segments VTYP_* (Librairie NSVTYP)

Les segments VTYP_* permettent de récupérer la valeur pointée par un pointeur.

Syntaxe	VTYP_*(ptr).value			
Paramètre	ptr	POINTER	I	Pointeur sur un segment VTYPE

1. Les segments VTYP_* peuvent notamment être utilisés avec les fonctions. VARARRAY_GET_ELEMENTPTR et VARSTRING_GET_PSTR
2. Les segments VTYP_* peuvent avoir les valeurs suivantes :
 - a) VTYP_PCSTRING
 - b) VTYP_PINT1
 - c) VTYP_PINT2
 - d) VTYP_PINT4
 - e) VTYP_PNUM4
 - f) VTYP_PNUM8
 - g) VTYP_PPOINTER
3. le VTYP_PCSTRING est limité à 65481 octets

Exemple :

```
...
BABELFIS_BABELFISH (mode,sourcedata,result)
MLERESULT.text = VTYP_PCSTRING(VARSTRING_GET_PSTR(result)).value
...
ptr1 = VARARRAY_GET_ELEMENTPTR(_result, pos2)
I# = VTYP_PNUM4(ptr1).value
...
Ptr2 = VARARRAY_GET_ELEMENTPTR(_result, pos2)
I% = VTYP_PINT4(ptr2).value
...
```


Types NULLABLE_* (Librairie NSVTYP)

Assez souvent, certains champs de segments paramètres peuvent être optionnels. Pour cette raison de nouveaux types Nullable_* ont été introduits.

Syntaxe	NULLABLE_INT1
	NULLABLE_INT2
	NULLABLE_INT4
	NULLABLE_NUM4
	NULLABLE_NUM8
	NULLABLE_CSTRING255

1. Chacun de ces types a pour "méthodes" <Type>_ISNULL, <Type>_SETNULL, <Type>_SETVALUE et <Type>_GETVALUE.
2. Ces types ne seront pas générés.

Exemple 1 :

```
...
LOCAL NULLABLE_CSTRING255 myString

NULLABLE_CSTRING255_SETVALUE(myString,"hello world")
IF (NULLABLE_CSTRING255_ISNULL(myString))
    RETURN DEFRET%
ELSE
    Str$= NULLABLE_CSTRING255_GETVALUE(myString)
    NULLABLE_CSTRING255_SETNULL myString
ENDIF
...
```

Exemple 2 :

```
...
LOCAL NULLABLE_NUM8 myNum

NULLABLE_NUM8_SETVALUE(myNum, 3.14)
IF (NULLABLE_NUM8_ISNULL(myNum))
    RETURN DEFRET%
ELSE
    Str$= NULLABLE_NUM8_GETVALUE(myNum)
    NULLABLE_NUM8_SETNULL myNum
ENDIF
```


FONCTIONS ET INSTRUCTIONS DE LA LIBRAIRIE NSSOAP

Ce chapitre présente les fonctions et instructions de la librairie NSSOAP permettant d'appeler des services Web.

Pour des fins de mise au point ou « débogage », les fonctions suivantes sont particulièrement utiles : NS SOAPHTTP SET PROXY couplée à un espion HTTP NS SOAPMSGREQUEST SERIALIZE et NS SOAPHTTP GET RESPONSESTRING.

SOAPHTTP

Fonction **NS_SOAPHTTP_GET_HTTPCONNECTION** (Librairie NSSOAP)

Retourne un pointeur utilisable sur les fonctions et instructions NS_HTTP_*.

Syntaxe	NS_SOAPHTTP_GET_HTTPCONNECTION (<i>pSOAPHttp</i>)		
Paramètre	pSoapHttp	POINTER	pointeur sur l'objet SoapHttp
Valeur retournée	POINTER	handle de connexion pConnection des fonctions et instructions NS_HTTP_*.	

Cette fonction est utilisée notamment pour ajouter des champs dans le header HTTP, ou pour fixer le mot de passe et le login du proxy.

Exemple :

Le nom des deux ressources suivantes est déterminé par le préfixe saisi dans le plugin client WebServices de NatStar. Pour notre exemple, ce préfixe est TEST1 :

- TEST1_INITIALIZE est une instruction de type <Prefix>_INITIALIZE qui est générée par le wizard. Elle initialise la mémoire commune (ou pool), le client Soap HTTP, les variables globales et l'adresse du service Web.
- TEST1_SOAPHTTP est une variable globale également générée par le plugin WebServices qui correspond au handle de la connexion HTTP.

Exemple de fixation du proxy et de ses paramètres :

```
local POINTER http%

TEST1_INITIALIZE
http% = NS_SOAPHTTP_GET_HTTPCONNECTION(TEST1_SOAPHTTP%)

NS_HTTP_SET_PROXY http%, "http://localhost:8081"
NS_HTTP_SET_PROXYUSERNAME http%, "login"
NS_HTTP_SET_PROXYPASSWORD http%, "password"
[...]
```

Voir aussi Fonctions et instructions NS_HTTP_* (Librairie NSHTTP), NS_SOAPHTTP_SET_PASSWORD, NS_SOAPHTTP_GET_RESPONSEHEADER, NS_SOAPHTTP_DISPOSE, NS_SOAPHTTP_SET_PROXY, NS_SOAPHTTP_SET_TIMEOUT, NS_SOAPHTTP_SET_CONTENTTYPE, NS_SOAPHTTP_GET_RESPONSESTRING

Instruction **NS_SOAPHTTP_GET_RESPONSEHEADER** (Librairie NSSOAP)

Retourne le header de la réponse HTTP sous forme de chaîne de caractères.

Syntaxe	NS_SOAPHTTP_GET_RESPONSEHEADER (pSOAPHttP, pszResponseHeader, nBuffLen%)		
Paramètres	pSoapHttp	POINTER	pointeur sur l'objet SoapHttp
	pszResponseHeader	POINTER	header de réponse
	nBuffLen%	INT	longueur du buffer
Valeur renvoyée	CSTRING		

Cette instruction est à utiliser exclusivement pour le débogage.

Voir aussi NS_SOAPHTTP_GET_RESPONSESTRING, NS_SOAPHTTP_SET_PASSWORD,
NS_SOAPHTTP_SET_USERNAME, NS_SOAPHTTP_GET_HTTPCONNECTION, NS_SOAPHTTP_DISPOSE,
NS_SOAPHTTP_SET_PROXY, NS_SOAPHTTP_SET_TIMEOUT, NS_SOAPHTTP_SET_CONTENTTYPE

Instruction **NS_SOAPHTTP_SET_PASSWORD** (Librairie NSSOAP)

Positionne le mot de passe pour l'authentification (Basic ou Digest) auprès du service web.

Syntaxe	NS_SOAPHTTP_SET_PASSWORD (<i>pSOAPHttp</i> , <i>pszPassword\$</i>)			
Paramètres	<i>pSoapHttp</i>	POINTER		pointeur sur l'objet SoapHttp
	<i>pszPassword\$</i>	CSTRING		mot de passe utilisateur

Voir aussi NS_SOAPHTTP_SET_USERNAME , NS_SOAPHTTP_GET_HTTPCONNECTION,
NS_SOAPHTTP_GET_RESPONSEHEADER, NS_SOAPHTTP_DISPOSE, NS_SOAPHTTP_SET_PROXY,
NS_SOAPHTTP_SET_TIMEOUT, NS_SOAPHTTP_SET_CONTENTTYPE,
NS_SOAPHTTP_GET_RESPONSESTRING

Instruction **NS_SOAPHTTP_SET_USERNAME** (Librairie NSSOAP)

Positionne le nom d'utilisateur pour l'authentification (Basic ou Digest) auprès du service web.

Syntaxe	NS_SOAPHTTP_SET_USERNAME (<i>pSOAPHttp, pszUsername\$</i>)			
Paramètres	pSoapHttp	POINTER	I	pointeur sur l'objet SoapHttp
	pszUsername\$	CSTRING	I	nom d'utilisateur

Voir aussi NS_SOAPHTTP_SET_PASSWORD, NS_SOAPHTTP_GET_HTTPCONNECTION,
NS_SOAPHTTP_GET_RESPONSEHEADER, NS_SOAPHTTP_DISPOSE, NS_SOAPHTTP_SET_PROXY,
NS_SOAPHTTP_SET_TIMEOUT, NS_SOAPHTTP_SET_CONTENTTYPE,
NS_SOAPHTTP_GET_RESPONSESTRING

Fonction **NS_SOAPHTTP_NEW** (Librairie NSSOAP)

Crée un nouvel objet SOAPHTTP. SoapHttp se charge de l'émission, de la réception et permet de définir un proxy, et de modifier le timeout. L'objet doit être libéré avec NS_SOAPHTTP_DISPOSE.

Syntaxe	NS_SOAPHTTP_NEW
Valeur retournée	POINTER sur l'objet SoapHttp

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = \
NS_SOAPMSGREQUEST_NEW("m:GetQuote, {http://ws.cdyne.com/}")
...
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://ws.cdyne.com/delayedstockquote/delayedstockquote.asmx", "http://ws.c
dyne.com/GetQuote")
...
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
```

Voir aussi NS_SOAPHTTP_DISPOSE

Instruction **NS_SOAPHTTP_DISPOSE** (Librairie NSSOAP)

Libère la mémoire utilisée par un objet Soaphttp.

Syntaxe	NS_SOAPHTTP_DISPOSE
----------------	----------------------------

Exemple :

```
...  
soapHttp% = NS_SOAPHTTP_NEW  
...  
NS_SOAPHTTP_DISPOSE soapHttp%
```

Voir aussi **NS_SOAPHTTP_NEW**

Instruction **NS_SOAPHTTP_SET_PROXY** (Librairie NSSOAP)

Permet de spécifier un proxy pour les requêtes SOAP.

Syntaxe	NS_SOAPHTTP_SET_PROXY pSoapHttp, pszProxy\$		
Paramètres	pSoapHttp	POINTER	pointeur sur l'objet SoapHttp
	pszProxy\$	CSTRING	proxy à utiliser pour les requêtes SOAP

Le paramètre pszProxy\$ doit être de la forme http://xxx.xxx.xxx.xxx:Port où xxx.xxx.xxx.xxx est l'adresse IP du Proxy et Port son port d'écoute
ex : http://123.11.25.111:2080

Exemple :

```
...  
soapHttp% = NS_SOAPHTTP_NEW  
NS_SOAPHTTP_SET_PROXY soapHttp%, http://123.11.25.111:2080  
...
```

Voir aussi NS_SOAPHTTP_NEW, NS_SOAPHTTP_GET_RESPONSESTRING,
NS_SOAPHTTP_SET_PASSWORD, NS_SOAPHTTP_SET_USERNAME,
NS_SOAPHTTP_GET_HTTPCONNECTION, NS_SOAPHTTP_DISPOSE, NS_SOAPHTTP_SET_TIMEOUT,
NS_SOAPHTTP_SET_CONTENTTYPE

Instruction **NS_SOAPHTTP_SET_TIMEOUT** (Librairie NSSOAP)

Positionne le délai après lequel le client soap renvoie une erreur de timeOut.

Syntaxe	NS_SOAPHTTP_SET_TIMEOUT <i>pSoapHttp, nTimeout%</i>		
Paramètres	<i>pSoapHttp</i>	POINTER	pointeur sur l'objet SoapHttp
	<i>nTimeout%</i>	INT(4)	délai en millisecondes

Erreur possible : Non

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
I% = 5000
NS_SOAPHTTP_SET_TIMEOUT soapHttp%, I%
...
```

Voir *aussi* **NS_SOAPHTTP_SET_PROXY**, **NS_SOAPHTTP_NEW**,
NS_SOAPHTTP_GET_RESPONSESTRING, **NS_SOAPHTTP_SET_PASSWORD**,
NS_SOAPHTTP_SET_USERNAME, **NS_SOAPHTTP_GET_HTTPCONNECTION**, **NS_SOAPHTTP_DISPOSE**,
NS_SOAPHTTP_SET_CONTENTTYPE

Instruction **NS_SOAPHTTP_SET_CONTENTTYPE** (Librairie NSSOAP)

Positionne le Content-Type à utiliser pour la requête SOAP.

Syntaxe	NS_SOAPHTTP_SET_CONTENTTYPE <i>pSOAPHttp, pszContentType\$</i>		
Paramètres	<i>pSoapHttp</i>	POINTER	pointeur sur l'objet SoapHttp
	<i>pszContentType\$</i>	CSTRING	content-type

1. Le header Content-Type pour une requête ou réponse SOAP détermine le type MIME du message et le charset utilisé pour le corps XML de la requête ou de la réponse.
2. Par défaut le Content-Type est positionné à `text/xml; charset="utf-8`.
3. Certains serveurs SOAP n'acceptent que "text/xml".

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
NS_SOAPHTTP_SET_CONTENTTYPE soapHttp%, 'text/xml'
```

Voir aussi *NS_SOAPHTTP_NEW*, *NS_SOAPHTTP_GET_RESPONSESTRING*,
NS_SOAPHTTP_SET_PASSWORD, *NS_SOAPHTTP_SET_USERNAME*,
NS_SOAPHTTP_GET_HTTPCONNECTION, *NS_SOAPHTTP_DISPOSE*,
NS_SOAPHTTP_SET_CONTENTTYPE, *NS_SOAPHTTP_SET_PROXY*

Instruction **NS_SOAPHTTP_GET_RESPONSESTRING** (Librairie NSSOAP)

Récupère la réponse SOAP complète sous forme d'une chaîne de caractères. Cela est utile notamment à des fins de tests.

Syntaxe	NS_SOAPHTTP_GET_RESPONSESTRING <i>pSOAPHttp, pszResponseString, nBuffLen%</i>		
Paramètres	<i>pSOAPHttp</i>	POINTER	pointeur sur un objet SoapHttp
	<i>pszResponseString</i>	POINTER	pointeur sur chaîne CSTRING
	<i>nBuffLen%</i>	INT(4)	taille de la chaîne CSTRING : utiliser SIZEOF.

Exemple :

```

LOCAL POINTER soapHttp%, POINTER soapMessageRequest%, POINTER soapConfig%,
POINTER nntp%
LOCAL POINTER soapStructParam%, POINTER soapSimple%
LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%, file%
LOCAL responseString$(4000), response$(4000), subject$
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("m:GetQuote, \
{http://ws.cdyne.com/}")
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_INST_NS
soapConfig%, "http://www.w3.org/2001/XMLSchema-instance"
NS_SOAPCONFIG_SET_SCHEMA_NS soapConfig%, "http://www.w3.org/2001/XMLSchema"
soapSimple% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "m:StockSymbol") NS
_SOAP_SIMPLEPARAM_SET_STRING(soapSimple%, "IBM")
NS_SOAP_SIMPLEPARAM_SET_USE_EXPLICIT_TYPE soapSimple%, false%
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, soapSimple%)
soapSimple% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "m:LicenseKey") NS
_SOAP_SIMPLEPARAM_SET_STRING(soapSimple%, "0")
NS_SOAP_SIMPLEPARAM_SET_USE_EXPLICIT_TYPE soapSimple%, false%
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, soapSimple%)
;for Tests purposes
NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
insert at END responseString$ to MAILBODY
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://ws.cdyne.com/delayedstockquote/delayedstockquote.asmx", "http://ws.c
dyne.com/GetQuote")
IF (NS_SOAP_ERROR <> 0)
ERRNO = NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
EXIT
ENDIF
;for Tests purposes
NS_SOAPHTTP_GET_RESPONSESTRING(soapHttp%, @response$, SIZEOF / response$)

```

```

insert at END response$ to S1
; this message here has a structure defined as follows
;<GetQuoteResult>
;    <StockSymbol>IBM</StockSymbol>
;    <LastTradeAmount>82.354</LastTradeAmount>
; etc.....
soapStructParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
soapSimpleParam% = NS_SOAPSTRUCTPARAM_GET(soapStructParam%,"StockSymbol",0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @subject$, SIZEOF /
subject$)
INSERT AT END subject$ TO S1
soapSimpleParam% =
NS_SOAPSTRUCTPARAM_GET(soapStructParam%,"LastTradeAmount",0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @subject$, SIZEOF /
subject$)
INSERT AT END subject$ TO S1
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)

```

Voir aussi NS SOAPHTTP NEW, NS SOAPHTTP SET PASSWORD, NS SOAPHTTP SET USERNAME,
NS SOAPHTTP GET HTTPCONNECTION, NS SOAPHTTP DISPOSE,
NS SOAPHTTP SET CONTENTTYPE, NS SOAPHTTP SET PROXY

Fonction NS_SOAPHTTP_SENDDREQUEST (Librairie NSSOAP)

Envoie une requête SOAP.

Syntaxe	NS_SOAPHTTP_SENDDREQUEST (<i>pSOAPHttp</i> , <i>pSOAPMessageRequest</i> , <i>pszUrlEndPoint</i> \$, <i>pszSoapAction</i> \$)		
Paramètres	PSOAPHttp	POINTER	pointeur sur un objet SoapHttp
	PSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest (message à envoyer)
	pszUrlEndPoint\$	CSTRING	URL du service web
	pszSoapAction\$	CSTRING	chaîne SOAPAction de la requête Soap (donnée fournie par le service web ou à partir du WSDL : attribut SOAPAction de l'élément <soap:operation>).
Valeur retournée	POINTER		

1. Le message à envoyer défini dans le paramètre PSOAPMessageRequest doit avoir été correctement créé et initialisé.
2. Cette fonction renvoie la réponse sous forme d'un objet SOAPMSGRESPONSE.
3. L' Endpoint est une donnée fournie par le Service Web ou récupérée à partir du WSDL : l'attribut location de l'élément <Soap :address>.

Exemple :

```
<port name="PDFCreatorPort" binding="tns:PDFCreatorBinding">
<soap:address
location="http://www.ammai.com/webservices/pdf/pdfcreator.php"/>
</port>
```

4. Le Header HTTP SOAPAction peut être utilisé pour indiquer le but de la requête SOAP. La valeur est une URI identifiant l'intention du message. SOAP ne met pas de restrictions sur le format ou la spécificité de cette URI. Un client HTTP doit utiliser ce champ header quand il émet une requête http SOAP. La présence et le contenu du champ header SOAPAction peuvent être utilisés par des serveurs comme des firewalls pour filtrer les requêtes. Si le champ SoapAction a une valeur de chaîne vide (""), cela indique que le but du message SOAP est indiqué par l'URI de la requête (pszUrlEndPoint\$).

Exemple :

```
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL CSTRING responseString$(10000)
```

```

LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%
LOCAL romulan$

soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("IntToRoman,{urn:Roman-\
IRoman}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"Int")
NS_SOAPSIMPLEPARAM_SET_VALUE(soapSimpleParam%,T1.TEXT)
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapSimpleParam%)
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%, "http://www.ebob42.com/cgi-
bin/Romulan.exe/soap/IRoman","urn:Roman-IRoman#IntToRoman")
IF (NS_SOAP_ERROR <> 0)
ERROR.TEXT = NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
EXIT
ENDIF
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @romulan$, SIZEOF romulan$)
ROMULAN.TEXT = romulan$
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)

```

Voir aussi NS_SOAPHTTP_NEW, NS_SOAPHTTP_SET_PASSWORD, NS_SOAPHTTP_SET_USERNAME,
NS_SOAPHTTP_GET_HTTPCONNECTION, NS_SOAPHTTP_DISPOSE,
NS_SOAPHTTP_SET_CONTENTTYPE, NS_SOAPHTTP_SET_PROXY, NS_SOAPHTTP_SET_TIMEOUT,
NS_SOAPHTTP_GET_RESPONSESTRING

SOAPCONFIG

Fonction **NS_SOAPCONFIG_GET_SOAPVERSION** (Librairie NSSOAP)

Récupère la version du message SOAP.

Syntaxe	NS_SOAPCONFIG_GET_SOAPVERSION (<i>pSOAPConfiguration</i>)		
Paramètre	pSOAPConfiguration	POINTER	pointeur sur une configuration Soap
Valeur retournée	INT(4)		

1. pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

2. La valeur renvoyée est soit NS_SOAPVERSION11% soit NS_SOAPVERSION12%.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("m:createPDF, /
http://www.ammai.com")
NS_SOAPMSGREQUEST_SET_SOAPVERSION soapMessageRequest%, NS_SOAPVERSION12%
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
;returnCode% = NS_SOAPVERSION12%
returnCode% = NS_SOAPCONFIG_GET_SOAPVERSION (soapConfig%)
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_ENVELOPE_NS** (Librairie NSSOAP)

Positionne le namespace de l'enveloppe du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_ENVELOPE_NS <i>pSOAPConfiguration</i> , <i>pszValue\$</i>		
Paramètres	<i>pSOAPConfiguration</i>	POINTER	pointeur sur une configuration Soap
	<i>pszValue\$</i>	CSTRING	namespace sous format http://WWW.W3.org/....

pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_ENVELOPE_NS
soapConfig%, "http://schemas.xmlsoap.org/soap/envelope"
NS_SOAPCONFIG_SET_ENVELOPE_PREFIX soapConfig%, "SOAP-ENV"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG, NS_SOAPCONFIG_GET_SOAPVERSION,
NS_SOAPCONFIG_SET_ENVELOPE_PREFIX, NS_SOAPCONFIG_SET_ENCODING_NS,
NS_SOAPCONFIG_SET_ENCODING_PREFIX, NS_SOAPCONFIG_SET_SCHEMA_INST_NS,
NS_SOAPCONFIG_SET_SCHEMA_INST_PREFIX, NS_SOAPCONFIG_SET_SCHEMA_NS,
NS_SOAPCONFIG_SET_SCHEMA_PREFIX

Instruction **NS_SOAPCONFIG_SET_ENVELOPE_PREFIX** (Librairie NSSOAP)

Positionne le préfixe du namespace de l'enveloppe du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_ENVELOPE_PREFIX <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	<i>pSOAPConfiguration</i>	POINTER	pointeur sur une configuration Soap
	<i>pszValue\$</i>	CSTRING	préfixe

pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_ENVELOPE_NS
soapConfig%, "http://schemas.xmlsoap.org/soap/envelope"
NS_SOAPCONFIG_SET_ENVELOPE_PREFIX soapConfig%, "SOAP-ENV"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_ENCODING_NS** (Librairie NSSOAP)

Positionne le namespace de l'encoding du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_ENCODING_NS <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	<i>pSOAPConfiguration</i>	POINTER	pointeur sur une configuration Soap
	<i>pszValue\$</i>	CSTRING	namespace sous format http://www.W3.org/....

pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_ENCODING_NS
soapConfig%, "http://schemas.xmlsoap.org/soap/encoding"
NS_SOAPCONFIG_SET_ENCODING_PREFIX soapConfig%, "SOAP-ENC"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_ENCODING_PREFIX** (Librairie NSSOAP)

Positionne le préfixe du namespace de l'encoding du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_ENCODING_PREFIX <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	pSOAPConfiguration	POINTER	pointeur sur une configuration Soap
	pszValue\$	CSTRING	préfixe du namespace

pSOAPConfiguration est récupéré avec la fonction
NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG(soapMessageRequest%)
NS_SOAPCONFIG_SET_ENCODING_NS
soapConfig%, "http://schemas.xmlsoap.org/soap/encoding"
NS_SOAPCONFIG_SET_ENCODING_PREFIX soapConfig%, "SOAP-ENC"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_SCHEMA_INST_NS** (Librairie **NSSOAP**)

Positionne le namespace du XMLSchema instance du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_SCHEMA_INST_NS <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	pSOAPConfiguration	POINTER	pointeur sur une configuration Soap
	pszValue\$	CSTRING	préfixe du namespace

pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_INST_NS
soapConfig%, "http://www.w3.org/2001/XMLSchema-instance"
...
```

Voir aussi **NS_SOAPMSGREQUEST_GET_SOAPCONFIG**

Instruction **NS_SOAPCONFIG_SET_SCHEMA_PREFIX** (Librairie NSSOAP)

Positionne le préfixe du namespace du XMLSchema du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_SCHEMA_PREFIX <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	pSOAPConfiguration	POINTER	pointeur sur une configuration Soap
	pszValue\$	CSTRING	préfixe du namespace.

pSOAPConfiguration est récupéré avec la fonction
NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_NS soapConfig%, "http://www.w3.org/2001/XMLSchema"
NS_SOAPCONFIG_SET_SCHEMA_PREFIX soapConfig%, "xsd"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_SCHEMA_NS** (Librairie NSSOAP)

Positionne le namespace du XMLSchema du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_SCHEMA_NS <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	<i>pSOAPConfiguration</i>	POINTER	pointeur sur une configuration Soap
	<i>pszValue\$</i>	CSTRING	namespace sous format http://www.W3.org/...

pSOAPConfiguration est récupéré avec la fonction NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_NS soapConfig%, "http://www.w3.org/2001/XMLSchema"
NS_SOAPCONFIG_SET_SCHEMA_PREFIX soapConfig%, "xsd"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Instruction **NS_SOAPCONFIG_SET_SCHEMA_INST_PREFIX** (Librairie NSSOAP)

Positionne le préfixe du namespace du XMLSchema instance du message SOAP.

Syntaxe	NS_SOAPCONFIG_SET_SCHEMA_INST_PREFIX <i>pSOAPConfiguration, pszValue\$</i>		
Paramètres	pSOAPConfiguration	POINTER	pointeur sur une configuration Soap
	pszValue\$	CSTRING	préfixe du namespace

pSOAPConfiguration est récupéré avec la fonction
NS_SOAPMSGREQUEST_GET_SOAPCONFIG.

Exemple :

```
...
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_INST_NS
soapConfig%, "http://www.w3.org/2001/XMLSchema-instance"
NS_SOAPCONFIG_SET_SCHEMA_INST_PREFIX soapConfig%, "xsi"
...
```

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG

SOAPMSGREQUEST

Fonction **NS_SOAPMSGREQUEST_NEW** (Librairie NSSOAP)

Crée un objet SOAPMSGREQUEST. Un tel objet doit être libéré par l'instruction `NS_SOAPMSGREQUEST_DISPOSE`.

Syntaxe	NS_SOAPMSGREQUEST_NEW (<i>pszMethodName\$</i>)		
Paramètre	<i>pszMethodName\$</i>	CSTRING	nom de la méthode de création
Valeur retournée	POINTER		

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("m:createPDF,
{http://www.ammai.com}")
username% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "username")
NS_SOAPSIMPLEPARAM_SET_STRING(username%, "aeltawil")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, username%)
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://www.ammai.com/webservices/pdf/pdfcreator.php", "http://www.ammai.com
/webservices/pdf/pdfcreator.php")
...
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
```

Voir aussi **NS_SOAPMSGREQUEST_DISPOSE**

Instruction **NS_SOAPMSGREQUEST_DISPOSE** (Librairie NSSOAP)

Libère un objet SOAPMSGREQUEST.

Syntaxe	NS_SOAPMSGREQUEST_DISPOSE <i>pSOAPMessageRequest</i>		
Paramètre	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Voir aussi **NS_SOAPMSGREQUEST_NEW**

Instruction **NS_SOAPMSGREQUEST_SET_SOAPVERSION** (Librairie NSSOAP)

Indique la version de SOAP à utiliser.

Syntaxe	NS_SOAPMSGREQUEST_SET_SOAPVERSION <i>pSOAPMessageRequest, soapVersion%</i>		
Paramètres	<i>pSOAPMessageRequest</i>	POINTER	pointeur sur un objet SoapMessageRequest
	<i>soapVersion%</i>	INT(4)	version de SOAP à utiliser

1. Le paramètre *soapVersion%* peut être fixé à **NS_SOAPVERSION11%** ou **NS_SOAPVERSION12%**. La valeur par défaut est **NS_SOAPVERSION11%**.
2. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
3. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW ("m:createPDF, \
{http://www.ammai.com}")
NS_SOAPMSGREQUEST_SET_SOAPVERSION soapMessageRequest%, NS_SOAPVERSION12%
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
returnCode% = NS_SOAPCONFIG_GET_SOAPVERSION (soapConfig%); returnCode% =
NS_SOAPVERSION12%
...
```

Voir aussi **NS_SOAPMSGREQUEST_GET_SOAPCONFIG**, **NS_SOAPMSGREQUEST_NEW**,
NS_SOAPMSGREQUEST_DISPOSE, **NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM**,
NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM, **NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM**,
NS_SOAPMSGREQUEST_CREATE_NULLPARAM, **NS_SOAPMSGREQUEST_ADD_HEADERBLOCK**,
NS_SOAPMSGREQUEST_SERIALIZE, **NS_SOAPMSGREQUEST_SET_MESSAGE_STYLE**

Fonction **NS_SOAPMSGREQUEST_CREATESIMPLEPARAM** (Librairie NSSOAP)

Crée un objet SIMPLEPARAM représentant les paramètres simples (du type chaîne de caractères, entier, date, données encodées en base64...).

Syntaxe	NS_SOAPMSGREQUEST_CREATESIMPLEPARAM (pSOAPMessageRequest, pszXMLQName\$)		
Paramètres	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest
	pszXMLQName\$	CSTRING	nom qualifié du paramètre
Valeur retournée	POINTER		

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("IntToRoman,{urn:Roman-Iroman}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATESIMPLEPARAM(soapMessageRequest%,"Int")
NS_SOAPSIMPLEPARAM_SET_VALUE(soapSimpleParam%,"5")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapSimpleParam%)
NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, sizeof
responseString$)
...
```

le résultat sérialise est :

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ns0:IntToRoman xmlns:ns0="urn:Roman-Iroman" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<Int>5</Int>
</ns0:IntToRoman>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Notez bien le paramètre <Int>5</Int>.

Il est également possible de créer un élément complètement qualifié

```
soapSimpleParam% = NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM  
(soapMessageRequest%, "t:Transaction, http://www.Natsys.com")
```

donne

```
<t:Transaction xmlns:t=" http://www.Natsys.com"> </t:Transaction>
```

Voir aussi *NS_SOAPMSGREQUEST_GET_SOAPCONFIG,* *NS_SOAPMSGREQUEST_NEW,*
NS_SOAPMSGREQUEST_DISPOSE, *NS_SOAPMSGREQUEST_SET_SOAPVERSION,*
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM, *NS_SOAPMSGREQUEST_CREATEARRAYPARAM,*
NS_SOAPMSGREQUEST_CREATENULLPARAM, *NS_SOAPMSGREQUEST_ADDHEADERBLOCK,*
NS_SOAPMSGREQUEST_SERIALIZE, *NS_SOAPMSGREQUEST_SET_MESSAGESTYLE*

Fonction **NS_SOAPMSGREQUEST_CREATESTRUCTPARAM** (Librairie NSSOAP)

Crée un objet STRUCTPARAM représentant une structure.

Syntaxe	NS_SOAPMSGREQUEST_CREATESTRUCTPARAM (pSOAPMessageRequest, pszXMLQName\$)		
Paramètres	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest
	pszXMLQName\$	CSTRING	nom qualifié du paramètre
Valeur retournée	POINTER	Pointeur sur un objet STRUCTPARAM	

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
; Links
soapStructParam% =
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(soapMessageRequest%, "Links")
; Links : EMAIL
linkRow% =
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(soapMessageRequest%, "LinkRow")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "URL")
NS_SOAP_SIMPLEPARAM_SET_STRING(soapSimpleParam%, "LIBSOAP.EMAIL")
NS_SOAP_STRUCTPARAM_ADD(linkRow%, soapSimpleParam%)
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "Target")
NS_SOAP_STRUCTPARAM_ADD(linkRow%, soapSimpleParam%)
NS_SOAP_STRUCTPARAM_ADD(soapStructParam%, linkRow%)
; Links : FORTUNE
linkRow% =
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(soapMessageRequest%, "LinkRow")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "URL")
NS_SOAP_SIMPLEPARAM_SET_STRING(soapSimpleParam%, "LIBSOAP.FORTUNE")
NS_SOAP_STRUCTPARAM_ADD(linkRow%, soapSimpleParam%)
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "Target")
NS_SOAP_STRUCTPARAM_ADD(linkRow%, soapSimpleParam%)
NS_SOAP_STRUCTPARAM_ADD(soapStructParam%, linkRow%)
...
```

Donne comme résultat :

```

<Links>
<LinkRow>
<URL xsi:type="xsd:string"> LIBSOAP.EMAIL </URL>
<Target> </Target>
</LinkRow>
<LinkRow>
<URL xsi:type="xsd:string"> LIBSOAP.FORTUNE </URL>
<Target> </Target>
</LinkRow>
</Links>

```

Voir ***aussi*** *NS SOAPMSGREQUEST_CREATEARRAYPARAM*, *NS SOAPMSGREQUEST_CREATENULLPARAM* ,
NS SOAPMSGREQUEST_SET_MESSAGESTYLE

Fonction **NS_SOAPMSGREQUEST_CREATEARRAYPARAM** (Librairie NSSOAP)

Crée un objet ARRAYPARAM représentant un tableau.

Syntaxe	NS_SOAPMSGREQUEST_CREATEARRAYPARAM (pSOAPMessageRequest, pszXMLQName\$)			
Paramètres	pSOAPMessageRequest	POINTER	I	pointeur sur un objet SoapMessageRequest
	pszXMLQName\$	CSTRING	I	nom qualifié du paramètre
Valeur retournée	POINTER	Pointeur sur un objet SoapArrayParameter		

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("CreateNavBar,{http://soap.xara.com/message/}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"Template")
NS_SOAPSIMPLEPARAM_SET_STRING(soapSimpleParam%,"Celtic")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapSimpleParam%)
; array of 2 elements string[8]
soapArrayParam% =
NS_SOAPMSGREQUEST_CREATEARRAYPARAM(soapMessageRequest%,"Text")
NS_SOAPARRAYPARAM_SET_ARRAYTYPE(soapArrayParam%,"string[8]")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"elem")
NS_SOAPSIMPLEPARAM_SET_STRING(soapSimpleParam%,"email")
NS_SOAPARRAYPARAM_ADD(soapArrayParam%,soapSimpleParam%)
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"elem")
NS_SOAPSIMPLEPARAM_SET_STRING(soapSimpleParam%,"fortune")
NS_SOAPARRAYPARAM_ADD(soapArrayParam%,soapSimpleParam%)
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapArrayParam%)
...
; Donne comme requête :
...
<SOAP-ENV:Body>
<ns0:CreateNavBar xmlns:ns0="http://soap.xara.com/message/" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<Template xsi:type="xsd:string"> Celtic </Template>
<Text SOAP-ENC:arrayType="string[8]">
<elem xsi:type="xsd:string"> email </elem>
<elem xsi:type="xsd:string"> fortune </elem>
```

```
</Text>  
...
```

Voir

aussi

NS SOAPMSGREQUEST_CREATESTRUCTPARAM,
NS SOAPMSGREQUEST_SET_MESSAGESTYLE

NS SOAPMSGREQUEST_CREATE_SIMPLEPARAM,
NS SOAPMSGREQUEST_CREATE_NULLPARAM,

Fonction **NS_SOAPMSGREQUEST_CREATENULLPARAM** (Librairie NSSOAP)

Crée un objet NULLPARAM représentant un paramètre vide.

Syntaxe	NS_SOAPMSGREQUEST_CREATENULLPARAM (<i>pSOAPMessageRequest</i> , <i>pszXMLQName\$</i>)		
Paramètres	<i>pSOAPMessageRequest</i>	POINTER	pointeur sur un objet SoapMessageRequest
	<i>pszXMLQName\$</i>	CSTRING	nom qualifié du paramètre
Valeur retournée	POINTER		

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Voir *aussi* [NS_SOAPMSGREQUEST_CREATESTRUCTPARAM](#), [NS_SOAPMSGREQUEST_CREATEARRAYPARAM](#),
[NS_SOAPMSGREQUEST_SET_MESSAGESTYLE](#)

Instruction **NS_SOAPMSGREQUEST_ADDPARAM** (Librairie NSSOAP)

Ajoute un paramètre à la requête.

Syntaxe	NS_SOAPMSGREQUEST_ADDPARAM <i>pSOAPMessageRequest, pParameter</i>		
Paramètres	<i>pSOAPMessageRequest</i>	POINTER	I pointeur sur un objet SoapMessageRequest
	<i>pParameter</i>	POINTER	I pointeur sur un objet SoapParameter (paramètre à ajouter)

1. Ce paramètre doit avoir été créé préalablement par NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM, NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM, NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM ou NS_SOAPMSGREQUEST_CREATE_NULLPARAM.
2. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
3. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("CreateNavBar,{http://soap.xara.com/message/}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"Template")
NS_SOAPSIMPLEPARAM_SET_STRING(soapSimpleParam%,"Celtic")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapSimpleParam%)
...
```

Voir **aussi** **NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM**,
NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM, **NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM**,
NS_SOAPMSGREQUEST_CREATE_NULLPARAM, **NS_SOAPMSGREQUEST_SET_MESSAGE_STYLE**

Instruction **NS_SOAPMSGREQUEST_ADDHEADERBLOCK** (Librairie NSSOAP)

Ajoute une entrée au header SOAP. Les headers SOAP pourront être utilisés pour gérer l'authentification, les transactions... mais sont encore peu utilisés actuellement.

Syntaxe	NS_SOAPMSGREQUEST_ADDHEADERBLOCK <i>pSOAPMessageRequest, pParameter, mustUnderstand%</i>		
Paramètres	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest
	pParameter	POINTER	pointeur sur un objet SoapParameter
	mustUnderstand%	INT(4)	gestion de l'entrée par le serveur SOAP

1. Le paramètre mustUnderstand indique si le serveur SOAP doit gérer ou non cette entrée.
2. Le header SOAP est un élément facultatif du message Soap. Si l'élément <Header> est présent il doit être le premier fils de l'élément <Envelope>.
3. Tous les fils immédiats de l'élément <Header> doivent être qualifiés par un namespace.
4. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
5. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
<SOAP-ENV:Header>
  <t:Transaction xmlns:t="some-URI" SOAP-ENV:mustUnderstand="1">
    5
  </t:Transaction>
</SOAP-ENV:Header>

soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("CreateNavBar,{http://soap.xara.com/message/}")
; a header element has to be fully qualified
soapSimpleParam% = NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM
(soapMessageRequest%, "t:Transaction")
NS_SOAPPARAM_ADD_ATTRIBUTE
soapSimpleParam%, "xmlns:t", "http://www.aelta.com"
NS_SOAPMSGREQUEST_ADDHEADERBLOCK
(soapMessageRequest%, soapSimpleParam%, true%)
```

Voir aussi NS_SOAPMSGREQUEST_NEW, NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM, NS_SOAPMSGREQUEST_SET_MESSAGESTYLE, NS_SOAPMSGREQUEST_GET_SOAPCONFIG

Fonction **NS_SOAPMSGREQUEST_GET_SOAPCONFIG** (Librairie NSSOAP)

Retourne un objet SOAPCONFIG. Un objet de ce type permet de positionner le préfixe de l'enveloppe SOAP.

Syntaxe	NS_SOAPMSGREQUEST_GET_SOAPCONFIG (<i>pSOAPMessageRequest</i>)		
Paramètre	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest
Valeur retournée	POINTER	Pointeur sur un Objet SoapConfiguration	

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
...
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = \
NS_SOAPMSGREQUEST_NEW("m:GetQuote,{http://ws.cdyne.com/}")
soapConfig% = NS_SOAPMSGREQUEST_GET_SOAPCONFIG (soapMessageRequest%)
NS_SOAPCONFIG_SET_SCHEMA_INST_NS
soapConfig%,"http://www.w3.org/2001/XMLSchema-instance"
NS_SOAPCONFIG_SET_SCHEMA_NS soapConfig%,"http://www.w3.org/2001/XMLSchema"
...
```

Résultat :

```
...
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
...
```

Voir aussi [NS_SOAPMSGREQUEST_NEW](#), [NS_SOAPMSGREQUEST_DISPOSE](#),
[NS_SOAPMSGREQUEST_SET_SOAPVERSION](#), [NS_SOAPMSGREQUEST_ADDHEADERBLOCK](#),
[NS_SOAPMSGREQUEST_SERIALIZE](#), [NS_SOAPMSGREQUEST_SET_MESSAGESTYLE](#)

Instruction **NS_SOAPMSGREQUEST_SERIALIZE** (Librairie NSSOAP)

Permet de sérialiser la requête SOAP dans une chaîne de caractères permettant ainsi de voir le contenu du message SOAP. Cela est utile pour les tests.

Syntaxe	NS_SOAPMSGREQUEST_SERIALIZE <i>pSOAPMessageRequest, pszBuffer, bufferSize%</i>		
Paramètres	pSOAPMessageRequest	POINTER	pointeur sur un objet SoapMessageRequest
	pszBuffer	POINTER	pointeur sur une chaîne CSTRING
	bufferSize%	INT(4)	taille de la chaîne CSTRING

1. Les noms qualifiés sont sous la forme "prefix:localname,uri". Le préfixe, de même que l'uri associé n'ont pas toujours à être précisés.
2. Pour tous les préfixes suivants, il n'est pas nécessaire de préciser l'uri correspondante (le client SOAP déduit l'uri correspondante selon le préfixe et la version de SOAP utilisée) : SOAP-ENV, env, SOAP-ENC, enc, xsi, xsd, xs.

Exemple :

```
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL CSTRING responseString$(10000)
LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%
LOCAL romulan$
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("IntToRoman,{urn:Roman-Iroman}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "Int")
NS_SOAPSIMPLEPARAM_SET_VALUE(soapSimpleParam%, T1.TEXT)
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, soapSimpleParam%)
NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
S1.TEXT = responseString$
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
Resultat
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ns0:IntToRoman xmlns:ns0="urn:Roman-Iroman" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<Int> 5 </Int>
</ns0:IntToRoman>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Voir aussi NS SOAPMSGREQUEST GET SOAPCONFIG, NS SOAPMSGREQUEST NEW,
NS SOAPMSGREQUEST DISPOSE, NS SOAPMSGREQUEST CREATESIMPLEPARAM,
NS SOAPMSGREQUEST CREATESTRUCTPARAM, NS SOAPMSGREQUEST CREATEARRAYPARAM,
NS SOAPMSGREQUEST CREATENULLPARAM, NS SOAPMSGREQUEST ADDHEADERBLOCK,
NS SOAPMSGREQUEST SET MESSAGESTYLE

Instruction **NS_SOAPMSGREQUEST_SET_MESSAGESTYLE** (Librairie NSSOAP)

Permet de définir le style d'encodage/pliage (RPC, Document ou Wrapped) de la requête SOAP.

Syntaxe	NS_SOAPMSGREQUEST_SET_MESSAGESTYLE <i>pSOAPMessageRequest, style\$</i>		
Paramètres	<i>pSOAPMessageRequest</i>	POINTER	pointeur sur un objet SoapMessageRequest
	<i>style\$</i>	CSTRING	Style d'encodage/pliage de la requête SOAP

Les valeurs possibles du paramètre *style\$* sont : "document", "rpc" ou "wrapped".

Voir aussi NS_SOAPMSGREQUEST_GET_SOAPCONFIG, NS_SOAPMSGREQUEST_NEW,
NS_SOAPMSGREQUEST_DISPOSE, NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM,
NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM, NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM,
NS_SOAPMSGREQUEST_CREATE_NULLPARAM, NS_SOAPMSGREQUEST_ADD_HEADERBLOCK,
NS_SOAPMSGREQUEST_SERIALIZE

SOAPMSGRESPONSE

Fonction **NS_SOAPMSGRESPONSE_GET_PARAMETER** (Librairie NSSOAP)

Retourne le nième paramètre de la réponse.

Syntaxe	NS_SOAPMSGRESPONSE_GET_PARAMETER (pSOAPMessageResponse, n%)		
Paramètres	pSOAPMessageResponse	POINTER	pointeur sur un objet SoapMessageResponse
	n%	INT(4)	rang du paramètre.
Valeur retournée	POINTER		

Le premier élément a 0 comme indice.

Exemple :

```
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL CSTRING responseString$(10000)
LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%
LOCAL romulan$
soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest% = NS_SOAPMSGREQUEST_NEW("IntToRoman, {urn:Roman-Iroman}")
soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "Int")
NS_SOAPSIMPLEPARAM_SET_VALUE(soapSimpleParam%, T1.TEXT)
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, soapSimpleParam%)
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%, "http://www.ebob42.com/cgi-
bin/Romulan.exe/soap/IRoman", "urn:Roman-Iroman#IntToRoman")
IF (NS_SOAP_ERROR <> 0)
ERROR.TEXT = NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
EXIT
ENDIF
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%, 0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @romulan$, SIZEOF romulan$)
ROMULAN.TEXT = romulan$
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
```

Voir aussi NS_SOAPSTRUCTPARAM GET, NS_SOAPMSGREQUEST GET SOAPCONFIG,
NS_SOAPMSGREQUEST NEW, NS_SOAPMSGREQUEST DISPOSE,
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM, NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM,
NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM, NS_SOAPMSGREQUEST_CREATE_NULLPARAM,
NS_SOAPMSGREQUEST_ADD_HEADERBLOCK, NS_SOAPMSGREQUEST_SERIALIZE

Fonction **NS_SOAPMSGRESPONSE_GET_PARAMETERNAMED** (Librairie NSSOAP)

Retourne le paramètre qui a pour nom local (sans préfixe) pszLocalName\$.

Syntaxe	NS_SOAPMSGRESPONSE_GET_PARAMETERNAMED (pSOAPMessageResponse, pszLocalName\$)		
Paramètres	pSOAPMessageResponse	POINTER	pointeur sur un objet SoapMessageResponse
	pszLocalName\$	CSTRING	nom de l'élément
Valeur retournée	POINTER sur un élément paramètre		

Exemple :

```
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL POINTER soapSimpleParam%, POINTER soapMessageResponse%
LOCAL response$(4000)
LOCAL i%, CONTROL ctrl

soapHttp% = NS_SOAPHTTP_NEW
soapMessageRequest%=
NS_SOAPMSGREQUEST_NEW("m:echoString,{http://soapinterop.org/}")

soapSimpleParam% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"inputString")
NS_SOAPSIMPLEPARAM_SET_STRING soapSimpleParam%,"Antoine"
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapSimpleParam%)

soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx","http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
EXIT
ENDIF

soapSimpleParam% =
NS_SOAPMSGRESPONSE_GET_PARAMETERNAMED(soapMessageResponse%,"return")
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @response$, SIZEOF
response$)
S1 = response$

NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
NS_SOAPHTTP_DISPOSE(soapHttp%)
```

Voir aussi [NS_SOAPSTRUCTPARAM_GET](#), [NS_SOAPSIMPLEPARAM_SET_STRING](#),
[NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM](#), [NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM](#),
[NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM](#), [NS_SOAPMSGREQUEST_CREATE_NULLPARAM](#)

Fonction **NS_SOAPMSGRESPONSE_GET_HEADERBLOCK** (Librairie **NSSOAP**)

Retourne le nième bloc header.

Syntaxe	NS_SOAPMSGRESPONSE_GET_HEADERBLOCK (pSOAPMessageResponse, n%)		
Paramètres	pSOAPMessageResponse	POINTER	I Pointeur sur un objet SoapMessageResponse
	n%	INT(4)	I Numéro du bloc header
Valeur retournée	POINTER		

Voir aussi NS_SOAPMSGRESPONSE_GET_PARAMETER, NS_SOAPSTRUCTPARAM_GET,
NS_SOAPMSGRESPONSE_GET_PARAMETERNAMED,
NS_SOAPMSGRESPONSE_GET_HEADERBLOCKNAMED

Fonction NS_SOAPMSGRESPONSE_GET_HEADERBLOCKNAMED (Librairie NSSOAP)

Retourne le bloc header qui a pour nom local pszLocalName\$.

Syntaxe	NS_SOAPMSGRESPONSE_GET_HEADERBLOCKNAMED (pSOAPMessageResponse, pszLocalName\$)			
Paramètres	pSOAPMessageResponse	POINTER	I	Pointeur sur un objet SoapMessageResponse
	pszLocalName\$	CSTRING	I	Nom local du bloc header
Valeur retournée	POINTER			

Voir aussi NS_SOAPMSGRESPONSE_GET_PARAMETER, NS_SOAPSTRUCTPARAM_GET,
NS_SOAPMSGRESPONSE_GET_PARAMETERNAMED, NS_SOAPMSGRESPONSE_GET_HEADERBLOCK

SOAPPARAM

Instruction **NS_SOAPPARAM_SET_TYPE** (Librairie NSSOAP)

Positionne le type du paramètre explicitement. Par défaut, le type est xsd:string

Syntaxe	NS_SOAPPARAM_SET_TYPE <i>pSOAPParameter</i> , <i>pszType</i> \$		
Paramètres	<i>pSOAPParameter</i>	POINTER	pointeur sur l'élément paramètre (élément XML)
	<i>pszType</i> \$	CSTRING	type du paramètre

pszType\$ peut contenir un élément complètement qualifié : préfixe, type et URI.

Exemple 1 :

```
...
password% =
NS_SOAPMSGREQUEST_CREATESIMPLEPARAM(soapMessageRequest%, "password")
NS_SOAPSIMPLEPARAM_SET_STRING(password%, "jtpi7qho")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, password%)
str$ = "hello world"
text% = NS_SOAPMSGREQUEST_CREATESIMPLEPARAM(soapMessageRequest%, "text")
NS_SOAPSIMPLEPARAM_SET_BASE64(text%, @str$, LENGTH str%)
NS_SOAPPARAM_SET_TYPE text%, "xsd:base64Binary"
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, text%)
font% = NS_SOAPMSGREQUEST_CREATESIMPLEPARAM(soapMessageRequest%, "font")
NS_SOAPSIMPLEPARAM_SET_STRING(font%, "Helvetica")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, font%)
fontsize% =
NS_SOAPMSGREQUEST_CREATESIMPLEPARAM(soapMessageRequest%, "fontsize")
NS_SOAPPARAM_SET_TYPE(fontsize%, "xsd:int")
NS_SOAPSIMPLEPARAM_SET_VALUE(fontsize%, "12")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, fontsize%)
...
```

donne :

```
...
<password xsi:type="xsd:string"> jtpi7qho </password>
<text xsi:type="xsd:base64Binary"> aGVsbG8gd29ybGQ= </text>
<font xsi:type="xsd:string"> Helvetica </font>
<fontsize xsi:type="xsd:int"> 12 </fontsize>
...
```

Exemple 2 :

```
...
soapStructParam% =\
```

```
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(soapMessageRequest%, "inputStruct")
NS_SOAPPARAM_SET_TYPE (soapStructParam%, "m0:SOAPStruct, {http://soapinterop.o
rg/xsd}")
soapSimpleParam% =\
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%, "varInt")
NS_SOAPSIMPLEPARAM_SET_INT soapSimpleParam%, 1962
NS_SOAPSTRUCTPARAM_ADD (soapStructParam%, soapSimpleParam%)
soapSimpleParam% =\
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%, "varString")
NS_SOAPSIMPLEPARAM_SET_STRING (soapSimpleParam%, "Nat Systems")
NS_SOAPSTRUCTPARAM_ADD (soapStructParam%, soapSimpleParam%)
soapSimpleParam% =\
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%, "varFloat")
NS_SOAPSIMPLEPARAM_SET_FLOAT soapSimpleParam%, 3.14159
NS_SOAPSTRUCTPARAM_ADD (soapStructParam%, soapSimpleParam%)

NS_SOAPMSGREQUEST_ADDPARAM (soapMessageRequest%, soapStructParam%)
NS_SOAPMSGREQUEST_SERIALIZE (soapMessageRequest%, @buffer$, sizeof buffer$)
...
```

donne :

```
...
<inputStruct xmlns:m0="http://soapinterop.org/xsd"
xsi:type="m0:SOAPStruct">
<varInt xsi:type="xsd:int"> 1962 </varInt>
<varString xsi:type="xsd:string"> Nat Systems </varString>
<varFloat xsi:type="xsd:float"> 3.14159012 </varFloat>
</inputStruct>
</m:echoStruct>
```

Voir aussi NS_SOAPPARAM_GET_PARAMETER_TYPE, NS_SOAPPARAM_ADD_ATTRIBUTE,
NS_SOAPSIMPLEPARAM_SET_BASE64, NS_SOAPSIMPLEPARAM_SET_BOOLEAN,
NS_SOAPSIMPLEPARAM_SET_DOUBLE, NS_SOAPSIMPLEPARAM_SET_FLOAT,
NS_SOAPSIMPLEPARAM_SET_HEXBINAR, NS_SOAPSIMPLEPARAM_SET_INT,
NS_SOAPPARAM_SET_TYPE NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE, NS_SOAP_ERROR,
NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM_SET_STRING

Fonction **NS_SOAPPARAM_GET_PARAMETERTYPE** (Librairie NSSOAP)

Récupère le type du paramètre.

Syntaxe	NS_SOAPPARAM_GET_PARAMETERTYPE (<i>pSOAPParameter</i>)		
Paramètre	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
Valeur retournée	INT(4)		

Les types de paramètre sont les suivants :

- 0 pour un tableau (array),
- 1 pour une structure,
- 2 pour un type simple.

Exemple :

```
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,soapArrayParam%)
INSERT AT END "Array"&&NS_SOAPPARAM_GET_PARAMETERTYPE \ (soapArrayParam%)
to S1
; Links
soapStructParam% =
NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(soapMessageRequest%,"Links")
INSERT AT END "structure"&&NS_SOAPPARAM_GET_PARAMETERTYPE
(soapStructParam%) to S1
```

Voir aussi **NS_SOAPPARAM_ADD_ATTRIBUTE,** **NS_SOAPSIMPLEPARAM_SET_BASE64,**
NS_SOAPSIMPLEPARAM_SET_BOOLEAN, **NS_SOAPSIMPLEPARAM_SET_DOUBLE,**
NS_SOAPSIMPLEPARAM_SET_FLOAT, **NS_SOAPSIMPLEPARAM_SET_HEXBINAR**
Y, **NS_SOAPSIMPLEPARAM_SET_INT,** **NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE,**
NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM_SET STRING

Instruction **NS_SOAPPARAM_ADD_ATTRIBUTE** (Librairie NSSOAP)

Rajoute un attribut au paramètre XML. L'élément XML peut faire partie indifféremment du Header Soap ou du body.

Syntaxe	NS_SOAPPARAM_ADD_ATTRIBUTE <i>pSOAPParameter, pszXMQName\$, pszValue\$</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	pszXMQName\$	CSTRING	nom de l'attribut XML
	pszValue\$	CSTRING	valeur de l'attribut XML

Exemple :

```
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("CreateNavBar,{http://soap.xara.com/message/}")
; a header element has to be fully qualified
soapSimpleParam% = NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM
(soapMessageRequest%, "t:Transaction")
NS_SOAPPARAM_ADD_ATTRIBUTE
soapSimpleParam%,"xmlns:t","http://www.aelta.com"
...
```

Donne :

```
<t:Transaction xmlns:t=" http://www.aelta.com" > 5 </t:Transaction>
```

Voir aussi NS_SOAPSIMPLEPARAM SET BASE64, NS_SOAPSIMPLEPARAM SET BOOLEAN,
NS_SOAPSIMPLEPARAM SET DOUBLE, NS_SOAPSIMPLEPARAM SET FLOAT,
NS_SOAPSIMPLEPARAM SET HEXBINARY, NS_SOAPSIMPLEPARAM SET_INT,
NS_SOAPPARAM GET PARAMETERTYPE, NS_SOAPSIMPLEPARAM SET_USEEXPLICITTYPE,
NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM SET STRING

Instruction NS_SOAPSIMPLEPARAM_SET_BASE64 (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML avec le résultat de la conversion en Base64 de la chaîne passée en paramètre et en forçant le type de l'élément à SOAP_ENC:Base64.

Syntaxe	NS_SOAPSIMPLEPARAM_SET_BASE64 <i>pSOAPParameter, pUnEncoded, Size%</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	pUnEncoded	POINTER	pointeur sur la chaîne
	Size%	INT(4)	taille de la chaîne

SOAP_ENC:Base64 et xsd:base64Binary sont équivalents.

Exemple :

```
...
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("m:echoBase64,{http://soapinterop.org/}")

inputBase64% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"inputBase64")
str$="hello world"
NS_SOAPSIMPLEPARAM_SET_BASE64(inputBase64%, @str$, LENGTH str$)
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,inputBase64%)

NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
IF NS_SOAP_ERROR
ERROR= "Serialize"&&NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
EXIT
ENDIF
@ctrl = MAILBODY
i% = ToMLE% (@responseString$, ctrl)

soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx","http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
EXIT
ENDIF
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
```

```
Size% = NS_SOAPSIMPLEPARAM_GET_Base64(soapSimpleParam%, @ret$, Sizeof ret$)
INSERT AT END "Size"&&Size%&&"ret$"&&ret$(0..size%-1) TO S1
; Hello World
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
...
la s rialisation de la requ te donne
...
<SOAP-ENV:Body>
<m:echoBase64 xmlns:m="http://soapinterop.org/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
; Hello World en Base64
<inputBase64 xsi:type="SOAP_ENC:Base64">aGVsbG8gd29ybGQ=</inputBase64>
</m:echoBase64>
</SOAP-ENV:Body>
...
```

Voir aussi NS_SOAPSIMPLEPARAM_SET_BOOLEAN, NS_SOAPSIMPLEPARAM_SET_DOUBLE,
NS_SOAPSIMPLEPARAM_SET_FLOAT, NS_SOAPSIMPLEPARAM_SET_HEXBINAR
Y, NS_SOAPSIMPLEPARAM_SET_INT, NS_SOAPPARAM_GET_PARAMETERTYPE,
NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE, NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG,
NS_SOAPSIMPLEPARAM_SET_STRING, NS_SOAPPARAM_ADD_ATTRIBUTE

Instruction **NS_SOAPSIMPLEPARAM_SET_BOOLEAN** (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML en forçant son type à xsd:boolean

Syntaxe	NS_SOAPSIMPLEPARAM_SET_BOOLEAN <i>pSOAPParameter, Value%</i>		
Paramètres	<i>pSOAPParameter</i>	POINTER	pointeur sur l'élément paramètre (élément XML)
	<i>Value%</i>	INT(1)	valeur de l'élément

Exemple :

```
...
inputBool% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"inputBoolean")
NS_SOAPSIMPLEPARAM_SET_BOOLEAN inputBool%,1
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,inputBool%)

NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
IF NS_SOAP_ERROR
ERROR= "Serialize"&&NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
EXIT
ENDIF
...
```

Donne :

```
...
<SOAP-ENV:Body>
<m:echoBoolean xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<inputBoolean xsi:type="xsd:boolean"> 1 </inputBoolean>
</m:echoBoolean>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPSIMPLEPARAM_SET_INT](#), [NS_SOAPPARAM_GET_PARAMETERTYPE](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#)

Instruction NS_SOAPSIMPLEPARAM_SET_DOUBLE (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML en forçant son type à xsd:double

Syntaxe	NS_SOAPSIMPLEPARAM_SET_DOUBLE <i>pSOAPParameter, Value#</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	Value#	NUM(4)	valeur de l'élément

Exemple :

```
Init event
LOCAL POINTER soapHttp%
soapHttp% = NS_SOAPHTTP_NEW
setdata soapHttp%

Terminate event
LOCAL POINTER soapHttp%
soapHttp% = getdata%
NS_SOAPHTTP_DISPOSE (soapHttp%)

Request
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL POINTER inputDouble%
LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%
LOCAL responseString$(4000)
LOCAL i%, n#(8) , m#(8)
LOCAL CONTROL ctrl
;NCL cannot hold this precision
n#=3.1415926535897932384626433832795
soapHttp% = getdata%

soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("m:echoDecimal,{http://soapinterop.org/}")

inputDouble% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%, "inputDecimal")
NS_SOAPSIMPLEPARAM_SET_DOUBLE inputDouble%, n#

NS_SOAPMSGREQUEST_ADDPARAM (soapMessageRequest%, inputDouble%)

NS_SOAPMSGREQUEST_SERIALIZE (soapMessageRequest%, @responseString$, SIZEOF
responseString$)
@ctrl = MAILBODY
i% = ToMLE% (@responseString$, ctrl)

soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST (soapHttp%,
soapMessageRequest%,
```

```

"http://mssoapinterop.org/asmx/simple.asmx","http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE (soapMessageRequest%)
EXIT
ENDIF

NS_SOAPHTTP_GET_RESPONSESTRING (soapHttp%, @responseString$, sizeof
responseString$)
@ctrl = S1
i% = ToMLE% (@responseString$, ctrl)
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER (soapMessageResponse%,0)
m# = NS_SOAPSIMPLEPARAM_GET_DOUBLE (soapSimpleParam%)
;NS_SOAPSIMPLEPARAM_GET_VALUE soapSimpleParam%, @n$, SIZEOF n$

INSERT AT END m# TO S1
NS_SOAPMSGREQUEST_DISPOSE (soapMessageRequest%)
Donne
<?xml version="1.0" encoding="utf-8" standalone="no"?>

<SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>
<m:echoDecimal xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<inputDecimal xsi:type="xsd:double"> 3.1415926535897931 </inputDecimal>
</m:echoDecimal>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Voir aussi [NS_SOAPSIMPLEPARAM SET BASE64](#), [NS_SOAPSIMPLEPARAM SET DOUBLE](#),
[NS_SOAPSIMPLEPARAM SET FLOAT](#), [NS_SOAPSIMPLEPARAM SET HEXBINARY](#),
[NS_SOAPSIMPLEPARAM SET_INT](#), [NS_SOAPPARAM GET PARAMETERTYPE](#),
[NS_SOAPSIMPLEPARAM SET USEEXPLICITYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM SET STRING](#), [NS_SOAPPARAM ADD ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM SET BOOLEAN](#)

Instruction NS_SOAPSIMPLEPARAM_SET_FLOAT (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML en forçant son type à xsd:float

Syntaxe	NS_SOAPSIMPLEPARAM_SET_FLOAT <i>pSOAPParameter</i> , <i>Value#</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	Value#	NUM(4)	valeur de l'élément

Exemple :

```
Init event
LOCAL POINTER soapHttp%
soapHttp% = NS_SOAPHTTP_NEW
setdata soapHttp%
Terminate event
LOCAL POINTER soapHttp%
soapHttp% = getdata%
NS_SOAPHTTP_DISPOSE (soapHttp%)
Request
LOCAL POINTER soapHttp%, POINTER soapMessageRequest%
LOCAL POINTER inputFloat%
LOCAL POINTER soapSimpleParam%
LOCAL POINTER soapMessageResponse%
LOCAL responseString$(4000)
LOCAL i%, n# , m#
LOCAL CONTROL ctrl

n#=3.1415
soapHttp% = getdata%

soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("m:echoFloat,{http://soapinterop.org/}")

inputFloat% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%,"inputFloat")
NS_SOAPSIMPLEPARAM_SET_FLOAT inputFloat%,n#
NS_SOAPMSGREQUEST_ADDPARAM (soapMessageRequest%,inputFloat%)

NS_SOAPMSGREQUEST_SERIALIZE (soapMessageRequest%, @responseString$, SIZEOF
responseString$)
@ctrl = MAILBODY
i% = ToMLE% (@responseString$, ctrl)

soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST (soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx","http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE (soapMessageRequest%)
EXIT
ENDIF

NS_SOAPHTTP_GET_RESPONSESTRING (soapHttp%, @responseString$, sizeof
```

```

responseString$)
@ctrl = S1
i% = ToMLE% (@responseString$, ctrl)
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
m# = NS_SOAPSIMPLEPARAM_GET_FLOAT (soapSimpleParam%)

INSERT AT END m# TO S1
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
Donne
<?xml version="1.0" encoding="utf-8" standalone="no"?>

<SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>
<m:echoFloat xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<inputFloat xsi:type="xsd:float"> 3.1415 </inputFloat>
</m:echoFloat>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Voir aussi NS_SOAPSIMPLEPARAM_SET_BASE64, NS_SOAPSIMPLEPARAM_SET_DOUBLE,
NS_SOAPSIMPLEPARAM_SET_HEXBINAR, NS_SOAPSIMPLEPARAM_SET_INT,
NS_SOAPPARAM_GET_PARAMETER, NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE,
NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM_SET_STRING,
NS_SOAPPARAM_ADD_ATTRIBUTE, NS_SOAPSIMPLEPARAM_SET_BOOLEAN

Instruction NS_SOAPSIMPLEPARAM_SET_HEXBINARY (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML avec le résultat de la conversion en Hexadécimal de la structure binaire passée en paramètre et en forçant le type de l'élément à xsd:hexBinary.

Syntaxe	NS_SOAPSIMPLEPARAM_SET_HEXBINARY <i>pSOAPParameter, pUnEncoded, Size%</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	pUnEncoded	POINTER	adresse d'une structure binaire
	Size%	INT(4)	taille de la structure

Exemple :

```
...
Local INT J%(2)[5],K%(2)[5], Size%
J%[0] = 1962 ; Hexa AA07
J%[1] = 1963; Hexa AB07
J%[2] = 1964; Hexa AC07
J%[3] = 1965; Hexa AD07

...
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("m:echoHexBinary,{http://soapinterop.org/}")

inputHexBinary% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"inputHexBinary")

NS_SOAPSIMPLEPARAM_SET_HexBinary inputHexBinary%,@J%, sizeof J%
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,inputHexBinary%)
...
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx","http://soapinterop.org/")
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
Size% = NS_SOAPSIMPLEPARAM_GET_HexBinary(soapSimpleParam%, @K%, SIZEOF k%)
; Now K% has the same values of J%
INSERT AT END "K = "&k%[0]&k%[1]&k%[2]&k%[3] TO S1
...
```

Donne :

```

...
<SOAP-ENV:Body>
<m:echoHexBinary xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<inputHexBinary
xsi:type="xsd:hexBinary">AA07AB07AC07AD07F0F0</inputHexBinary>
</m:echoHexBinary>
</SOAP-ENV:Body>

```

Voir aussi NS_SOAPSIMPLEPARAM SET BASE64, NS_SOAPSIMPLEPARAM SET DOUBLE,
NS_SOAPSIMPLEPARAM SET FLOAT, NS_SOAPSIMPLEPARAM SET_INT,
NS_SOAPPARAM GET PARAMETERTYPE, NS_SOAPSIMPLEPARAM SET USEEXPLICITYPE,
NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM SET STRING,
NS_SOAPPARAM ADD ATTRIBUTE, NS_SOAPSIMPLEPARAM SET BOOLEAN

Instruction NS_SOAPSIMPLEPARAM_SET_INT (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML en forçant son type à xsd:int.

Syntaxe	NS_SOAPSIMPLEPARAM_SET_INT <i>pSOAPParameter, Value%</i>		
Paramètres	pSOAPParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	Value%	INT(4)	valeur de l'élément

Exemple :

```
Init event
LOCAL POINTER soapHttp%
soapHttp% = NS_SOAPHTTP_NEW
setdata soapHttp%
Terminate event
LOCAL POINTER soapHttp%
soapHttp% = getdata%
NS_SOAPHTTP_DISPOSE (soapHttp%)

Request
LOCAL POINTER soapHttp%, soapMessageRequest%, soapConfig%
LOCAL POINTER inputInteger%, soapSimpleParam%, soapMessageResponse%
LOCAL responseString$(4000)
LOCAL INTEGER j%
LOCAL i%, CONTROL ctrl
soapHttp% = getdata%
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("m:echoInteger,{http://soapinterop.org/}")

inputInteger% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM (soapMessageRequest%, "inputInteger")
NS_SOAPSIMPLEPARAM_SET_INT inputInteger%, 12
NS_SOAPMSGREQUEST_ADDPARAM (soapMessageRequest%, inputInteger%)

NS_SOAPMSGREQUEST_SERIALIZE (soapMessageRequest%, @responseString$, SIZEOF
responseString$)
@ctrl = MAILBODY
i% = ToMLE% (@responseString$, ctrl)

soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST (soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx", "http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE (soapMessageRequest%)
EXIT
ENDIF
soapSimpleParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER (soapMessageResponse%, 0)
j% = NS_SOAPSIMPLEPARAM_GET_INT (soapSimpleParam%)

S1 = j%
NS_SOAPMSGREQUEST_DISPOSE (soapMessageRequest%)
```

Donne :

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>

<SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>
<m:echoInteger xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
<inputInteger xsi:type="xsd:int"> 12 </inputInteger>
</m:echoInteger>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Voir aussi *NS_SOAPSIMPLEPARAM_SET_BASE64,* *NS_SOAPSIMPLEPARAM_SET_DOUBLE,*
NS_SOAPSIMPLEPARAM_SET_FLOAT, *NS_SOAPSIMPLEPARAM_SET_HEXBINARY,*
NS_SOAPPARAM_GET_PARAMETERTYPE, *NS_SOAPSIMPLEPARAM_SET_INT,*
NS_SOAPSIMPLEPARAM_SET_USEEXPLICITYPE, *NS_SOAP_ERROR,* *NS_SOAP_GET_ERRORMSG,*
NS_SOAPSIMPLEPARAM_SET_STRING, *NS_SOAPPARAM_ADD_ATTRIBUTE,*
NS_SOAPSIMPLEPARAM_SET_BOOLEAN

Instruction **NS_SOAPSIMPLEPARAM_SET_STRING** (Librairie NSSOAP)

Met à jour le contenu de l'élément paramètre XML en forçant son type à xsd:string.

Syntaxe	NS_SOAPSIMPLEPARAM_SET_STRING <i>pSOAPParameter</i> , <i>pszVal\$</i>		
Paramètres	<i>pSOAPParameter</i>	POINTER	pointeur sur l'élément paramètre (élément XML)
	<i>pszVal\$</i>	CSTRING	chaîne contenant la valeur du paramètre

Exemple 1 :

```
soapMessageRequest% =
NS_SOAPMSGREQUEST_NEW("SendMail,{urn:EmailIPortTypeInft-IemailService}")
fromAddress% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"FromAddress")
NS_SOAPSIMPLEPARAM_SET_STRING(fromAddress%,"test")
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,fromAddress%)
```

Exemple 2 :

```
...
inputString% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%,"inputString")
NS_SOAPSIMPLEPARAM_SET_STRING inputString%,"Antoine"
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%,inputString%)

NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
IF NS_SOAP_ERROR
ERROR= "Serialize"&&NS_SOAP_GET_ERRORMSG
NS_SOAPMSGREQUEST_DISPOSE(soapMessageRequest%)
EXIT
ENDIF
...
```

donne :

```
...
<SOAP-ENV:Body>
<m:echoBoolean xmlns:m="http://soapinterop.org/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<inputBoolean xsi:type="xsd:boolean"> 1 </inputBoolean>
</m:echoBoolean>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Voir aussi [NS_SOAPPARAM_SET_TYPE](#), [NS_SOAPPARAM_GET_PARAMETERTYPE](#),
[NS_SOAPPARAM_ADD_ATTRIBUTE](#), [NS_SOAPSIMPLEPARAM_SET_BASE64](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPSIMPLEPARAM_SET_INT](#), [NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE](#),
[NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#)

Instruction **NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE** (Librairie NSSOAP)

Permet d'activer ou de désactiver l'indication du type d'un paramètre de message Soap.

Par défaut, le type d'un élément paramètre est défini explicitement.

Exemple : `<inputString xsi:type="xsd:string">`. On peut si on le souhaite s'affranchir de cette définition en positionnant `NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE` à `False%` ce qui donnerait pour le même paramètre : `<inputString>`.

Syntaxe	NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE <i>pSOAPParameter, value%</i>		
Paramètres	pSOAPParameter	POINTER	I pointeur sur l'élément paramètre (élément XML)
	value%	INT(1)	I False% ou True% : active ou désactive la spécification explicite du type

Exemple :

```
soapSimple% =
NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(soapMessageRequest%, "m:StockSymbol")
NS_SOAPSIMPLEPARAM_SET_STRING(soapSimple%, "IBM")
NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE soapSimple%, false%
NS_SOAPMSGREQUEST_ADDPARAM(soapMessageRequest%, soapSimple%)
```

Donne comme sérialisation :

```
<m:StockSymbol> IBM </m:StockSymbol>
```

On remarque l'absence de type xsd.

Voir aussi NS_SOAPSIMPLEPARAM_SET_BASE64, NS_SOAPSIMPLEPARAM_SET_DOUBLE,
NS_SOAPSIMPLEPARAM_SET_FLOAT, NS_SOAPSIMPLEPARAM_SET_HEXBINAR,
NS_SOAPPARAM_GET_PARAMETERTYPE, NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG,
NS_SOAPSIMPLEPARAM_SET_STRING, NS_SOAPPARAM_ADD_ATTRIBUTE,
NS_SOAPSIMPLEPARAM_SET_BOOLEAN

Fonction **NS_SOAPSTRUCTPARAM_GET** (Librairie NSSOAP)

Retourne le descendant de rang n d'un paramètre structure. La recherche se fait sur le nom passé en paramètre.

Syntaxe	NS_SOAPSTRUCTPARAM_GET (<i>soapMessageResponse</i> %, <i>pszLocalName</i> %, <i>n</i> %)		
Paramètres	pSOAPMessageResponse	POINTER	pointeur sur un objet SoapMessageResponse
	pszLocalName%	CSTRING	nom de l'élément
	n%	INT(4)	rang de filiation
Valeur retournée	POINTER sur l'élément en question		

L'élément fils a 0 comme indice, le petits-fils 1 etc.

Exemple :

```
; this message here is a structure defined as follows
;<GetQuoteResult>
; <StockSymbol>IBM</StockSymbol>
; <LastTradeAmount>82.354</LastTradeAmount>
; etc.....
soapStructParam% = NS_SOAPMSGRESPONSE_GET_PARAMETER(soapMessageResponse%,0)
soapSimpleParam% = NS_SOAPSTRUCTPARAM_GET(soapStructParam%,"StockSymbol",0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @subject$, sizeof subject$)
INSERT AT END subject$ TO S1
soapSimpleParam% =
NS_SOAPSTRUCTPARAM_GET(soapStructParam%,"LastTradeAmount",0)
NS_SOAPSIMPLEPARAM_GET_STRING(soapSimpleParam%, @subject$, sizeof subject$)
INSERT AT END subject$ TO S1
```

Voir aussi NS_SOAPMSGRESPONSE_GET_PARAMETER, NS_SOAPSIMPLEPARAM_SET_STRING,
NS_SOAPSTRUCTPARAM_GET, NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM,
NS_SOAPMSGREQUEST_CREATE_STRUCTPARAM, NS_SOAPMSGREQUEST_CREATE_ARRAYPARAM,
NS_SOAPMSGREQUEST_CREATE_NULLPARAM, NS_SOAPMSGREQUEST_ADD_HEADERBLOCK,
NS_SOAPMSGREQUEST_SERIALIZE

Fonction **NS_SOAPSIMPLEPARAM_GET_BOOLEAN** (Librairie **NSSOAP**)

Lit le contenu de l'élément paramètre XML de type wsdl:boolean.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_BOOLEAN (<i>pSOAPSimpleParameter</i>)		
Paramètres	pSOAPSimpleParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
Valeur retournée	INT(1)	Le contenu de l'élément XML.	

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <facultatif>1</ facultatif >
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
Local facultatif%(1)
...
simpleParam1 = NS_SOAPSTRUCTPARAM_GET(structParam,"facultatif",0)
facultatif% = NS_SOAPSIMPLEPARAM_GET_BOOLEAN(simpleParam1)
  things_trace "la valeur de l'élément <facultatif> est " &
facultatif%
;** facultatif vaut 1 (true%)
```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPPARAM_GET_PARAMETERTYPE](#), [NS_SOAPSIMPLEPARAM_SET_INT](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_FLOAT](#),
[NS_SOAPSIMPLEPARAM_GET_INT](#),
[NS_SOAPSIMPLEPARAM_GET_DOUBLE](#), [NS_SOAPSIMPLEPARAM_GET_STRLGTH](#),
[NS_SOAPSIMPLEPARAM_GET_VALUE](#)

Fonction **NS_SOAPSIMPLEPARAM_GET_DOUBLE** (Librairie NSSOAP)

Lit le contenu de l'élément paramètre XML de type wsdl:double.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_DOUBLE (<i>pSOAPSsimpleParameter</i>)		
Paramètres	pSOAPSsimpleParameter	POINTER	I pointeur sur l'élément paramètre (élément XML)
Valeur retournée	NUM(8)	Le contenu de l'élément XML.	

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <prix>35.4</prix>
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
Local prix#
...
simpleParam1 = NS_SOAPSTRUCTPARAM_GET(structParam,"prix",0)
prix# = NS_SOAPSIMPLEPARAM_GET_DOUBLE(simpleParam1)
things_trace "le prix lu dans l'élément <prix> est " & prix#
;** prix vaut 35.4
```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPPARAM_GET_PARAMETERTYPE](#), [NS_SOAPSIMPLEPARAM_SET_INT](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_FLOAT](#),
[NS_SOAPSIMPLEPARAM_GET_INT](#),
[NS_SOAPSIMPLEPARAM_GET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_STRLGTH](#),
[NS_SOAPSIMPLEPARAM_GET_VALUE](#)

Fonction **NS_SOAPSIMPLEPARAM_GET_FLOAT** (Librairie **NSSOAP**)

Lit le contenu de l'élément paramètre XML de type wsdl:float.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_FLOAT (<i>pSOAPSimpleParameter</i>)		
Paramètres	pSOAPSimpleParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
Valeur retournée	NUM(4)	Le contenu de l'élément XML.	

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <prix>35.4</prix>
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
Local prix#(4)
...
simpleParam1 = NS_SOAPSTRUCTPARAM_GET(structParam,"prix",0)
prix# = NS_SOAPSIMPLEPARAM_GET_FLOAT(simpleParam1)
  things_trace "le prix lu dans l'élément <prix> est " & prix#
; ** prix vaut 35.4
```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPPARAM_GET_PARAMETERTYPE](#), [NS_SOAPSIMPLEPARAM_SET_INT](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_GET_INT](#), [NS_SOAPSIMPLEPARAM_GET_BOOLEAN](#),
[NS_SOAPSIMPLEPARAM_GET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_STRLGTH](#),
[NS_SOAPSIMPLEPARAM_GET_VALUE](#)

Fonction **NS_SOAPSIMPLEPARAM_GET_INT** (Librairie NSSOAP)

Lit le contenu de l'élément paramètre XML de type `wsd:int`.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_INT (<i>pSOAPSimpleParameter</i>)		
Paramètres	<code>pSOAPSimpleParameter</code>	<code>POINTER</code>	<code>I</code> pointeur sur l'élément paramètre (élément XML)
Valeur retournée	<code>INT(4)</code>	Le contenu de l'élément XML.	

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <age>35</age>
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
Local age%
...
simpleParam1 = NS_SOAPSTRUCTPARAM_GET(structParam,"age",0)
age% = NS_SOAPSIMPLEPARAM_GET_INT(simpleParam1)
  things_trace "l'âge lu dans l'élément <age> est " & age%
; ** age% vaut 35
```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPPARAM_GET_PARAMETERTYPE](#), [NS_SOAPSIMPLEPARAM_SET_INT](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_FLOAT](#),
[NS_SOAPSIMPLEPARAM_GET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_STRLGTH](#),
[NS_SOAPSIMPLEPARAM_GET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_GET_VALUE](#)

Instruction NS_SOAPSIMPLEPARAM_GET_VALUE (Librairie NSSOAP)

Permet de récupérer le pointeur et la taille de la chaîne correspondant à la valeur de l'élément pointé par le paramètre pSOAPSimpleParameter.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_VALUE <i>pSOAPSimpleParameter,</i> <i>pszValue, bufferSize%</i>		
Paramètres	pSOAPSimpleParameter	POINTER	pointeur sur l'élément paramètre (élément XML)
	pszValue	POINTER	valeur du pointeur
	bufferSize%	INT(4)	taille du buffer

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <contrat>
          <item>contrat 1</item>
          <item>contrat 2</item>
          <item>contrat 3</item>
        </contrat>
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
LOCAL POINTER simpleParam1
LOCAL INTEGER len1
LOCAL POINTER ptr1
LOCAL POINTER simpleParam4
LOCAL POINTER structArrayParam2
Local contrat$(1000)
...
IF (structArrayParam2<> 0) AND
(NS_SOAPPARAM_GET_PARAMETER_TYPE(structArrayParam2) <> NS_SOAPNULL%)
  size2 = NS_SOAPSTRUCTPARAM_SIZE_NAMED(structArrayParam2,"item")
  things_trace "nombre d'éléments <item> pour <contrat> : " &
```

```

size2
    @_result.contrat = YYWEBSEER_NEW_ArrayOf_xsd_strin(size2)
    FOR i2 = 0 TO size2-1
        pos2[0]=i2
        simpleParam4 =
NS_SOAPSTRUCTPARAM_GET(structArrayParam2,"item", 2)
        things_trace "pointeur sur l'élément <item> en
position" && 2 & " : " & simpleParam4
        IF (simpleParam4<> 0) AND
(NS_SOAPPARAM_GET_PARAMETERTYPE(simpleParam4) <> NS_SOAPNULL%)
            len1 =
NS_SOAPSIMPLEPARAM_GET_STRLGTH(simpleParam4)
            things_trace "longueur de la chaîne : "
& len1
; ** len1 vaut 8 (longueur de contrat 3)
            ptr1 =
ForceDSLlength(_result.contrat.r[i2],len1)
            things_trace "pointeur de la dynstr : "
& ptr1
            NS_SOAPSIMPLEPARAM_GET_VALUE
simpleParam4,ptr1,len1+1
            mov ptr1, @contrat$, len1 + 1
            things_trace "valeur de l'item de
contrat : " & contrat$ ; ** contrat vaut 'contrat 3'
            else
                SetDSNull(_result.contrat.r[i2])
            Endif
        endfor
    endif

```

Voir aussi [NS_SOAPSIMPLEPARAM_SET_BASE64](#), [NS_SOAPSIMPLEPARAM_SET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_SET_FLOAT](#), [NS_SOAPSIMPLEPARAM_SET_HEXBINAR](#),
[NS_SOAPPARAM_GET_PARAMETERTYPE](#), [NS_SOAPSIMPLEPARAM_SET_INT](#),
[NS_SOAPSIMPLEPARAM_SET_USEEXPLICITYPE](#), [NS_SOAP_ERROR](#), [NS_SOAP_GET_ERRORMSG](#),
[NS_SOAPSIMPLEPARAM_SET_STRING](#), [NS_SOAPPARAM_ADD_ATTRIBUTE](#),
[NS_SOAPSIMPLEPARAM_SET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_DOUBLE](#),
[NS_SOAPSIMPLEPARAM_GET_INT](#),
[NS_SOAPSIMPLEPARAM_GET_BOOLEAN](#), [NS_SOAPSIMPLEPARAM_GET_STRLGTH](#),
[NS_SOAPSIMPLEPARAM_GET_FLOAT](#)

Fonction NS_SOAPSIMPLEPARAM_GET_STRLGTH (Librairie NSSOAP)

Récupère la longueur de la chaîne de caractères pointée par le paramètre pSOAPSImpleParameter.

Syntaxe	NS_SOAPSIMPLEPARAM_GET_STRLGTH (<i>pSOAPSImpleParameter</i>)		
Paramètres	pSOAPSImpleParameter	POINTER	I pointeur sur l'élément paramètre (élément XML)
Valeur retournée	INT(4)		

Exemple :

Message soap :

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getDataResponse xmlns="http://testwebservice.xxx.xx">
      <getDataReturn>
        <contrat>
          <item>contrat 1</item>
          <item>contrat 2</item>
          <item>contrat 3</item>
        </contrat>
      </getDataReturn>
    </getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Code NCL :

```
LOCAL POINTER simpleParam1
LOCAL INTEGER len1
LOCAL POINTER ptr1
LOCAL POINTER simpleParam4
LOCAL POINTER structArrayParam2
Local contrat$(1000)
...
IF (structArrayParam2<> 0) AND
(NS_SOAPPARAM_GET_PARAMETERTYPE(structArrayParam2) <> NS_SOAPNULL%)
  size2 = NS_SOAPSTRUCTPARAM_SIZE_NAMED(structArrayParam2,"item")
  things_trace "nombre d'éléments <item> pour <contrat> : " &
size2
  @_result.contrat = YYWEBSE_NEW_ArrayOf_xsd_strin(size2)
  FOR i2 = 0 TO size2-1
    pos2[0]=i2
    simpleParam4 =
NS_SOAPSTRUCTPARAM_GET(structArrayParam2,"item", 2)
    things_trace "pointeur sur l'élément <item> en
position" && 2 & " : " & simpleParam4
    IF (simpleParam4<> 0) AND
```

```
(NS_SOAPPARAM_GET_PARAMETERTYPE(simpleParam4) <> NS_SOAPNULL%)
    len1 =
NS_SOAPSIMPLEPARAM_GET_STRLGTH(simpleParam4)
    things_trace "longueur de la chaîne : "
& len1      ;** len1 vaut 8 (longueur de contrat 3)
    ptr1 =
ForcedDSLength(_result.contrat.r[i2],len1)
    things_trace "pointeur de la dynstr : "
& ptr1
    NS_SOAPSIMPLEPARAM_GET_VALUE
simpleParam4,ptr1,len1+1
    mov ptr1, @contrat$, len1 + 1
    things_trace "valeur de l'item de
contrat : " & contrat$      ;** contrat vaut 'contrat 3'
    else
    SetDSNull(_result.contrat.r[i2])
    Endif
endfor
endif
```

Voir aussi NS_SOAPSIMPLEPARAM SET BASE64, NS_SOAPSIMPLEPARAM SET DOUBLE,
NS_SOAPSIMPLEPARAM SET FLOAT, NS_SOAPSIMPLEPARAM SET HEXBINARY,
NS_SOAPPARAM GET_PARAMETERTYPE, NS_SOAPSIMPLEPARAM SET_INT,
NS_SOAPSIMPLEPARAM SET_USEEXPLICITTYPE, NS_SOAP_ERROR, NS_SOAP_GET_ERRORMSG,
NS_SOAPSIMPLEPARAM SET STRING, NS_SOAPPARAM ADD_ATTRIBUTE,
NS_SOAPSIMPLEPARAM SET_BOOLEAN, NS_SOAPSIMPLEPARAM GET_FLOAT,
NS_SOAPSIMPLEPARAM GET_BOOLEAN, NS_SOAPSIMPLEPARAM GET_INT,
NS_SOAPSIMPLEPARAM GET_DOUBLE,
NS_SOAPSIMPLEPARAM GET_VALUE

Gestion des erreurs

Fonction **NS_SOAP_ERROR** (Librairie NSSOAP)

Détecte si une erreur a été produite par l'instruction NS SOAPMSGREQUEST SERIALIZE ou la fonction NS SOAPHTTP SENDREQUEST.

Syntaxe	NS_SOAP_ERROR	
Valeur retournée	INT(1)	TRUE% OU FALSE%

Exemple :

```
...
NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
IF NS_SOAP_ERROR
ERROR= "Serialize"&&NS_SOAP_GET_ERRORMSG
ENDIF
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx", "http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
ENDIF
...
```

Voir aussi NS_SOAPSIMPLEPARAM_SET_BASE64, NS_SOAPSIMPLEPARAM_SET_DOUBLE,
NS_SOAPSIMPLEPARAM_SET_FLOAT, NS_SOAPSIMPLEPARAM_SET_HEXBINAR,
NS_SOAPPARAM_GET_PARAMETERTYPE, NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE,
NS_SOAP_GET_ERRORMSG, NS_SOAPSIMPLEPARAM_SET_STRING,
NS_SOAPPARAM_ADD_ATTRIBUTE, NS_SOAPSIMPLEPARAM_SET_BOOLEAN

Fonction **NS_SOAP_GET_DETAIL_ERROR** (Librairie NSSOAP)

Récupère les détails de l'erreur produite par la fonction `NS_SOAPHTTP_SENDREQUEST`.

Syntaxe	NS_SOAP_GET_DETAIL_ERROR (<i>errKind</i>)		
Paramètre	errKind	INT(4)	type d'erreur retournée (une des constantes NS_SOAP_ERROR_*)
Valeur retournée	CSTRING	Chaîne de 255 caractères contenant le message d'erreur demandé.	

Exemple :

```
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx", "http://soapinterop.org/")
IF NS_SOAP_ERROR
    Errmsg$      = NS_SOAP_GET_DETAIL_ERROR (NS_SOAP_ERROR_MSG%)
    errCode      = NS_SOAP_GET_DETAIL_ERROR (NS_SOAP_ERROR_FAULT_CODE%)
    errString$   = NS_SOAP_GET_DETAIL_ERROR (NS_SOAP_ERROR_FAULT_STRING%)
    errActor$    = NS_SOAP_GET_DETAIL_ERROR (NS_SOAP_ERROR_FAULT_ACTOR%)
ENDIF
; remarque : ces codes retour sont valables pour la version 1.1 de Soap
```

Voir aussi `NS_SOAPHTTP_SENDREQUEST`, constantes `NS_SOAP_ERROR *`

Constantes NS_SOAP_ERROR_% (Librairie NSSOAP)

Constantes utilisées en paramètre avec la fonction NS SOAP GET DETAIL ERROR.

Syntaxe	Déclaration interne	Description
NS_SOAP_ERROR_MSG%	1	renvoie le code complet de l'erreur SOAP
NS_SOAP_ERROR_FAULT_CODE%	2	renvoie la partie "code" de l'erreur SOAP
NS_SOAP_ERROR_FAULT_STRING%	3	renvoie la partie "string" de l'erreur SOAP
NS_SOAP_ERROR_FAULT_ACTOR%	4	renvoie la partie "actor" de l'erreur SOAP

Voir aussi NS SOAP GET DETAIL ERROR

Fonction **NS_SOAP_GET_ERRORMSG** (Librairie NSSOAP)

Récupère le texte de l'erreur produite par l'instruction NS_SOAPMSGREQUEST_SERIALIZE ou la fonction NS_SOAPHTTP_SENDREQUEST.

Syntaxe	NS_SOAP_GET_ERRORMSG	
Valeur retournée	CSTRING	Chaîne de 255 caractères contenant le message d'erreur.

Exemple :

```
...
NS_SOAPMSGREQUEST_SERIALIZE(soapMessageRequest%, @responseString$, SIZEOF
responseString$)
IF NS_SOAP_ERROR
ERROR= "Serialize"&&NS_SOAP_GET_ERRORMSG
ENDIF
soapMessageResponse% = NS_SOAPHTTP_SENDREQUEST(soapHttp%,
soapMessageRequest%,
"http://mssoapinterop.org/asmx/simple.asmx", "http://soapinterop.org/")
IF NS_SOAP_ERROR
ERROR= NS_SOAP_GET_ERRORMSG
ENDIF
...
```

Voir aussi NS_SOAPSIMPLEPARAM_SET_BASE64, NS_SOAPSIMPLEPARAM_SET_DOUBLE,
NS_SOAPSIMPLEPARAM_SET_FLOAT, NS_SOAPSIMPLEPARAM_SET_HEXBINAR,
NS_SOAPPARAM_GET_PARAMETERTYPE, NS_SOAPSIMPLEPARAM_SET_USEEXPLICITTYPE,
NS_SOAP_ERROR, NS_SOAPSIMPLEPARAM_SET_STRING, NS_SOAPPARAM_ADD_ATTRIBUTE,
NS_SOAPSIMPLEPARAM_SET_BOOLEAN

UTILISATION DES SEGMENTS GENERES

La plupart des opérations des services webs prennent en paramètre des structures, ce qui correspond à des segments en NCL.

Des pointeurs typés sont utilisés systématiquement pour les segments, de façon à pouvoir gérer la valeur NULL.

Création

Pour chaque segment, une fonction permet de créer un pointeur typé vers ce segment. Elle initialise également le contenu du segment (DynStr à NULL, pointeurs typés à NULL, INT(4) à 0 etc ...)

Exemple :

La fonction suivante utilise un paramètre de type TB_SOAPStruct@ :

```
FUNCTION TB_echoStructAsSimpleTypes(TB_SOAPStruct @inputStruct, DYNSTR
@outputString, INT @outputInteger(4), NUM @outputFloat(4)) RETURN INT(4)
```

Le segment est défini de la façon suivante :

```
SEGMENT TB_SOAPStruct
DYNSTR _varString
INT varInt(4)
NUM varFloat(4)
ENDSEGMENT
```

et peut être créé avec la fonction :

```
FUNCTION TB_NEW_SOAPStruct RETURN TB_SOAPStruct@
```

Exemple d'utilisation :

```
LOCAL TB_SOAPSTRUCT@ struct
LOCAL dynstr outStringVar
LOCAL INT outputInt(4)
LOCAL NUM outputFloat(4)

@struct = TB_NEW_SOAPSTRUCT
struct._varString = "hello world"
struct.varInt = 999
struct.varFloat = 3.14159

TB_ECHOSTRUCTASSIMPLETYPES (struct,outStringVar,outputInt,outputFloat)
```

Ne pas oublier le @ lors de la création du pointeur typé.

Souvent, un segment peut contenir une référence vers un autre segment :

```
SEGMENT TEST5_SEG2Bean
;[nullable]
TEST5_SEG1Bean@ VARSEG
ENDSEGMENT
```

Dans ce cas, il est nécessaire d'instancier également le segment contenu :

```
LOCAL TEST5_SEG2Bean@ mySeg
@mySeg = TEST5_NEW_SEG2Bean
@mySeg.VARSEG = TEST5_NEW_SEG1Bean
mySeg.VARSEG.VARINT4 = 14
```

Il est possible de fixer à NULL un pointeur typé de la façon suivante :

```
@mySeg = 0
```

Holders

Des segments intermédiaires appelés holders sont utilisés pour les segments en paramètres out ou inout.

En effet, la fonction peut vouloir renvoyer la valeur NULL. Cela n'est pas possible sans passer par un segment intermédiaire.

Le holder contient simplement un élément nommé value.

Exemple :

```
SEGMENT TEST3_Holder_SEG1Bean
TEST3_SEG1Bean@ value
ENDSEGMENT

FUNCTION TEST3_MODIFYSEG(TEST3_Holder_SEG1Bean @PARAM) RETURN INT(4)

LOCAL TEST3_Holder_SEG1BEAN mySegHolder

@mySegHolder.value = TEST3_NEW_SEG1Bean
mySegHolder.value.VARINT4 = 45
mySegHolder.value.VARNUM4 = 22.32
mySegHolder.value._VARSTRING = "cstring"
mySegHolder.value.VARDYNSTR = "dynstr"

TEST3_MODIFYSEG(mySegHolder)
Pour fixer la valeur à NULL :
@mySegHolder.value = 0
Pour tester si la valeur est à NULL :
if @mySegHolder.value = 0
...
endif
```

Les holders sont également utilisés pour les tableaux de structures.

UTILISATION DES DYNSTR

Les services webs clients utilisent les DynStr et la possibilité qu'a une DynStr d'être positionnée à NULL.

Fixer une DynStr à NULL

Il est possible de fixer une DynStr à NULL lors de l'appel d'une fonction par exemple.

```
SetDSNull(mydynstr)
```

Tester si une DynStr est à NULL

```
if IsDSNull%(mydynstr) = TRUE%  
[...]  
endif
```


UTILISATION DES TABLEAUX GENERES

Les tableaux sont générés sous forme de segments.

Des fonctions permettent de créer le tableau et de déterminer la taille des dimensions du tableau.

Création d'un tableau

Exemple d'un tableau de DynStr

Déclaration dans la librairie correspondant au service web :

```
SEGMENT TEST2_ArrayOf_xsd_string
DYNSTR@ r[]
ENDSEGMENT

FUNCTION TEST2_NEW_ArrayOf_xsd_string(INTEGER size0) RETURN
TEST2_ArrayOf_xsd_string@

INSTRUCTION TEST2_SIZE_ArrayOf_xsd_string TEST2_ArrayOf_xsd_string @array,
INTEGER @sizes[1]
```

Utilisation :

```
LOCAL TEST2_ArrayOf_xsd_string@ array

@array = TEST2_NEW_ArrayOf_xsd_string(5)
array.r[0] = "addition"
array.r[1] = "substraction"
array.r[2] = "multiplication"
array.r[3] = "division"
array.r[4] = "modulus"
```

Exemple d'un tableau de dimension > 1

Déclaration dans la librairie correspondant au service web :

```
SEGMENT TB_ArrayOfString2D_D1
DYNSTR@ r[]
ENDSEGMENT

SEGMENT TB_ArrayOfString2D
TB_ArrayOfString2D_D1@ r[]
ENDSEGMENT

FUNCTION TB_NEW_ArrayOfString2D(INTEGER size0, INTEGER size1) RETURN
TB_ArrayOfString2D@

INSTRUCTION TB_SIZE_ArrayOfString2D TB_ArrayOfString2D @array, INTEGER
@sizes[2]
```

Utilisation :

```
local TB_ARRAYOFSTRING2D@ inputArray

@inputArray = TB_NEW_ARRAYOFSTRING2D(2,2)

inputArray.r[0].r[0] = "string[0,0]"
inputArray.r[0].r[1] = "string[0,1]"
inputArray.r[1].r[0] = "string[1,0]"
inputArray.r[1].r[1] = "string[1,1]"
```

Parcours d'un tableau

Lorsqu'une fonction correspondant à une opération d'un service web renvoie un tableau, un holder est utilisé, de façon à pouvoir positionner la valeur du tableau à NULL.

Pour parcourir le tableau il est nécessaire d'en connaître la taille. On utilise pour cela l'instruction `prefix_SIZE_arrayname` :

```
LOCAL XMET_ARRAYOFSERVICESUMMARY@ SERVICES
LOCAL XMET_Holder_ArrayOfServiceSumm servicesHolder
LOCAL INTEGER size@[1]
LOCAL INTEGER i%

IF XMET_GETALLSERVICESUMMARIES(servicesHolder) <> 0
  ERROR = XMET_GET_ERRORMSG$
  RETURN DEFRET%
ENDIF

@services = servicesHolder.value

XMET_SIZE_ARRAYOFSERVICESUMMAR services,size%

FOR i% = 0 TO size@[0]-1
  INSERT AT END services.r[i%].value.name TO LISTBOX
ENDFOR
```

Tableau de segments

Ce cas est particulier parce qu'on peut vouloir fixer à NULL certains éléments du tableau. Il faut donc passer par un holder.

```
GOOGLE_ResultElementArray@ resultElements

SEGMENT GOOGLE_ResultElementArray
GOOGLE_Holder_ResultElement@ r[]
ENDSEGMENT

FUNCTION GOOGLE_NEW_ResultElementArray(INTEGER size0) RETURN
GOOGLE_ResultElementArray@

INSTRUCTION GOOGLE_SIZE_ResultElementArray GOOGLE_ResultElementArray
@array, INTEGER @sizes[1]
Utilisation (result.resultElements est de type GOOGLE_ResultElementArray@):
for i% = result.startIndex-1 to result.endIndex-1
url = result.resultElements.r[i%].value.url
title = result.resultElements.r[i%].value.title
snippet = result.resultElements.r[i%].value.snippet

INSERT AT END url TO MLERESULT
INSERT AT END title TO MLERESULT
INSERT AT END snippet TO MLERESULT
INSERT AT END "-----" TO MLERESULT
endfor
On peut tester la nullité d'un élément de la façon suivante :
if @result.resultElements.r[i%].value = 0
...
endif
```


DONNEES BINAIRES

Les données binaires sont retournées sous la forme d'un VARBINARY.

Exemple :

```
FUNCTION BR_BrailleText(DYNSTR InText, NUM TextFontSize(4), VARBINARY  
@_result) RETURN INT(4)
```

```
BR_INITIALIZE
```

```
intext = ENTRYtext
```

```
if (BR_BRAILLETEXT (intext,100.0,result) <> 0)
```

```
ERRORTEXT = BR_GET_ERRORMSG$
```

```
BR_TERMINATE
```

```
return defret%
```

```
endif
```

```
if VARBINARY_SAVETO (result,"braille.jpg") = false%
```

```
ERRORTEXT = VARBINARY_GET_ERRORMSG
```

```
BR_TERMINATE
```

```
return defret%
```

```
endif
```

```
BR_TERMINATE
```


CALLBACK

L'instruction <prefix>_SET_CALLBACK_BEFORE_SEND permet de fixer un callback qui sera appelé juste avant l'appel du service web.

Le callback en question doit être une instruction sans paramètre.

Cela permet de modifier le message SOAP qui sera envoyé et ainsi de pallier certaines limites : il est ainsi possible d'ajouter des éléments au header SOAP.

Exemple d'instruction callback qui ajoute des éléments au header SOAP :

```
instruction callbackHeader
local pointer structParam
local pointer simpleParam1
  structParam = NS_SOAPMSGREQUEST_CREATESTRUCTPARAM(SS_soapMessageRequest%, \
  "MessengerHeader,{com.esendex.ems.soapinterface}")

  simpleParam1 =
  NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(SS_soapMessageRequest%, \
  "Username,{com.esendex.ems.soapinterface}")
  NS_SOAPSIMPLEPARAM_SET_STRING simpleParam1, MAINWND(WNDHANDLE).USERNAMEEF
  NS_SOAPSTRUCTPARAM_ADD structParam,simpleParam1

  simpleParam1 =
  NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(SS_soapMessageRequest%, \
  "Password,{com.esendex.ems.soapinterface}")
  NS_SOAPSIMPLEPARAM_SET_STRING simpleParam1,MAINWND(WNDHANDLE).PASSWORDDEF
  NS_SOAPSTRUCTPARAM_ADD structParam,simpleParam1

  simpleParam1 =
  NS_SOAPMSGREQUEST_CREATE_SIMPLEPARAM(SS_soapMessageRequest%, \
  "Account,{com.esendex.ems.soapinterface}")
  NS_SOAPSIMPLEPARAM_SET_STRING simpleParam1,MAINWND(WNDHANDLE).ACCOUNTTEF
  NS_SOAPSTRUCTPARAM_ADD structParam,simpleParam1

  NS_SOAPMSGREQUEST_ADDHEADERBLOCK(SS_soapMessageRequest%, \
  structParam, false%, "")
endinstruction
```

Utilisation :

```
local dynstr result

SS_INITIALIZE

SS_SET_CALLBACK_BEFORE_SEND @callbackHeader

if SS_SENDMESSAGE (EFRECIPIENT,EFMESSAGE,SS_MESSAGE_TYPE_TEXT,result) <> 0
  insert at end SS_GET_ERRORMSG$ to MLE
  SS_TERMINATE
  return defret%
endif
insert at end result to MLE
SS_TERMINATE
```

Pour supprimer le callback, faire :

```
SS_SET_CALLBACK_BEFORE_SEND 0
```

INDEX

<
 <Prefix>_<OperationName> 24
 C
 Callback 153
 D
 Données binaires 151
 G
 GET_ERRORMSG\$ 26
 H
 Holders 142
 I
 INITIALIZE 38
 ISNULL 23
 N
 NS_SOAP_ERROR 134
 NS_SOAP_ERROR_FAULT_ACTOR
 % 136
 NS_SOAP_ERROR_FAULT_CODE%
 136
 NS_SOAP_ERROR_FAULT_STRING
 % 136
 NS_SOAP_ERROR_MSG% 136
 NS_SOAP_GET_DETAIL_ERROR
 135
 NS_SOAP_GET_ERRORMSG 137
 NS_SOAPCONFIG_GET_SOAPVERS
 ION 79
 NS_SOAPCONFIG_SET_ENCODING
 _NS 82
 NS_SOAPCONFIG_SET_ENCODING
 _PREFIX 83
 NS_SOAPCONFIG_SET_ENVELOPE
 _NS 80
 NS_SOAPCONFIG_SET_ENVELOPE
 _PREFIX 81
 NS_SOAPCONFIG_SET_SCHEMA_I
 NST_NS 84
 NS_SOAPCONFIG_SET_SCHEMA_I
 NST_PREFIX 87
 NS_SOAPCONFIG_SET_SCHEMA_N
 S 86
 NS_SOAPCONFIG_SET_SCHEMA_P
 REFIX 85
 NS_SOAPHTTP_DISPOSE 71
 NS_SOAPHTTP_GET_HTTPCONNE
 CTION 66
 NS_SOAPHTTP_GET_RESPONSEH
 EADER 67
 NS_SOAPHTTP_GET_RESPONSEST
 RING 75
 NS_SOAPHTTP_NEW 70
 NS_SOAPHTTP_SENDREQUEST 77
 NS_SOAPHTTP_SET_CONTENTTYP
 E 74
 NS_SOAPHTTP_SET_PASSWORD
 68
 NS_SOAPHTTP_SET_PROXY 72
 NS_SOAPHTTP_SET_TIMEOUT 73
 NS_SOAPHTTP_SET_USERNAME 69
 NS_SOAPMSGREQUEST_ADDHEAD
 ERBLOCK 99
 NS_SOAPMSGREQUEST_ADDPARA
 M 98
 NS_SOAPMSGREQUEST_CREATEA
 RRAYPARAM 95
 NS_SOAPMSGREQUEST_CREATEN
 ULLPARAM 97
 NS_SOAPMSGREQUEST_CREATESE
 MPLEPARAM 91
 NS_SOAPMSGREQUEST_CREATESTR
 UCTPARAM 93
 NS_SOAPMSGREQUEST_DISPOSE
 89
 NS_SOAPMSGREQUEST_GET_SOA
 PCONFIG 100
 NS_SOAPMSGREQUEST_NEW 88
 NS_SOAPMSGREQUEST_SERIALIZ
 E 101
 NS_SOAPMSGREQUEST_SET_MES
 SAGESTYLE 103
 NS_SOAPMSGREQUEST_SET_SOA
 PVERSION 90
 NS_SOAPMSGRESPONSE_GET_HE
 ADERBLOCK 106

NS_SOAPMSGRESPONSE_GET_HE
ADERBLOCKNAMED 107
NS_SOAPMSGRESPONSE_GET_PA
RAMETER 104
NS_SOAPMSGRESPONSE_GET_PA
RAMETERNAMED 105
NS_SOAPPARAM_ADD_ATTRIBUTE
111
NS_SOAPPARAM_GET_PARAMETE
RTYPE 110
NS_SOAPPARAM_SET_TYPE 108
NS_SOAPSIMPLEPARAM_GET_BOO
LEAN 126
NS_SOAPSIMPLEPARAM_GET_DOU
BLE 127
NS_SOAPSIMPLEPARAM_GET_FLO
AT 128
NS_SOAPSIMPLEPARAM_GET_INT
129
NS_SOAPSIMPLEPARAM_GET_STR
LGTH 132
NS_SOAPSIMPLEPARAM_GET_VAL
UE 130
NS_SOAPSIMPLEPARAM_SET_BAS
E64 112
NS_SOAPSIMPLEPARAM_SET_BOO
LEAN 114
NS_SOAPSIMPLEPARAM_SET_DOU
BLE 115
NS_SOAPSIMPLEPARAM_SET_FLO
AT 117
NS_SOAPSIMPLEPARAM_SET_HEX
BINARY 119
NS_SOAPSIMPLEPARAM_SET_INT
121
NS_SOAPSIMPLEPARAM_SET_STRI
NG 123
NS_SOAPSIMPLEPARAM_SET_USE
EXPLICITTYPE 124
NS_SOAPSTRUCTPARAM_GET 125
NULLABLE_CSTRING255 63
NULLABLE_INT1 63
NULLABLE_INT2 63
NULLABLE_INT4 63
NULLABLE_NUM4 63
NULLABLE_NUM8 63

P

Parcours d un tableau 148

S

SET_ENDPOINT 39
SET_PASSWORD 40
SET_USERNAME 41

T

Tableau de segments 149
TERMINATE 42

U

Utilisation des DynStr 143
Utilisation des segments générés 139
Utilisation des tableaux générés 145

V

VARARRAY_GET_ARRAY 43
VARARRAY_GET_ELEMENTPTR 27
VARARRAY_GET_NBDIMS 29
VARARRAY_GET_SIZES 44
VARARRAY_GET_SIZES_PTR 30
VARARRAY_ISNULL 31
VARARRAY_NEW 45
VARARRAY_NEW_1D 47
VARARRAY_NEW_2D 49
VARARRAY_SETNULL 51
VARBINARY_GET 52
VARBINARY_GET_PTR 32
VARBINARY_GET_SIZE 33
VARBINARY_ISNULL 34
VARBINARY_NEW 53
VARBINARY_NEW_SIZED 54
VARBINARY_SETNULL 55
VARPOOL_NEW 56
VARSTRING_ADD_PSTR 57
VARSTRING_ADD_VARSTRING 58
VARSTRING_GET 59
VARSTRING_GET_PSTR 35
VARSTRING_GETLENGTH 36
VARSTRING_ISNULL 37
VARSTRING_NEW 60
VARSTRING_SETNULL 61
VTYP_PCSTRING 62
VTYP_PINT1 62
VTYP_PINT2 62
VTYP_PINT4 62
VTYP_PNUM4 62
VTYP_PNUM8 62
VTYP_PPOINTER 62

